Exhibit 1008

01-16-01

A

TRANSMITTAL FOR U.S. PATENT APPLICATION UNDER 37 CFR §1.53(b)

Box Patent Application
ASSISTANT COMMISSIONER
OF PATENTS
Washington, D.C. 20231

Attorney's Docket: 066241.0117

Sir:

Transmitted herewith for filing is the original patent application of:

Inventor: Keith A. Lowery, et al.

For: METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING

__X__ **Pursuant to 37 C.F.R. § 1.213(a), Applicant(s) respectfully requests that the above-identified patent application not be published under 37 U.S.C. § 122(b). Applicant(s) hereby certifies that the invention disclosed in the above-identified application has not been and will not be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication at eighteen months after filing.**

Enclosed are:

(1) __89_ pages of Specification, Claims and Abstract;

(2) ___4_ pages of Executed Declaration and Power of Attorney;

(3) __11_ pages of Formal Drawings;

(4) ___1_ page of executed Verified Statement Claiming Small Entity Status under 37 C.F.R. §§ 1.9(f) and .1.27(c)—Small Business Concern;

(5) ___4_ pages of an Assignment of the invention to epicRealm Inc.;

(6) ___2_ pages of a separate cover sheet in compliance with 37 C.F.R. § 3.28 and § 3.31 is included with an Assignment recordal fee of $40.00 pursuant to 37 C.F.R. § 1.21(h);

(7) ___2_ pages of an Information Disclosure Statement with PTO-1449 Form and attached references (10 references);

(8) ___2_ pages of a Preliminary Amendment;

(8) ___1_ page of a Certificate of Mailing by Express Mail No.: EL759174439US; and

(9) ___2_ pages of a this Transmittal Form with attached patent application filing fee in the amount of $1,600.00.

FEE CALCULATION					FEE
	Number		Number Extra	Rate	Basic Fee $ 355.00
Total Claims	105	20	85	X $ 9 =	$ 765.00
Independent Claims	15	3	12	X $40 =	$ 480.00
				TOTAL FILING FEE =	$1,600.00

Enclosed is a check in the amount of $1,600.00 for the filing fee. Please charge any additional fees or credit any overpayment to Deposit Account No. 02-0384 of Baker Botts L.L.P. **A duplicate copy of this sheet is enclosed.**

BAKER BOTTS L.L.P.
Attorneys for Applicants

Matthew B. Talpis
Registration No. 45,152

Date: January 12, 2001

Attorney's Docket No.:
066241.0117

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Keith A. Lowery, et al.

Filing Date: January 12, 2001

Title: METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING

Box: Patent Application
Assistant Commissioner for Patents
Washington, D.C. 20231

Dear Sir:

CERTIFICATE OF MAILING BY EXPRESS MAIL

I hereby certify that the attached Filing Fee Transmittal (2 pages), Patent Application (89 pages), a check in the amount of $1,600.00, Declaration and Power of Attorney (4 pages), Formal Drawings (11 pages), Assignment Cover Sheet (2 pages), Assignment (4 pages), a check in the amount of $40.00, Information Disclosure Statement (2 pages), PTO-Form 1449 (1 page) (with 10 references), Preliminary Amendment (2 pages), Verified Statement (1 page), and a Baker Botts L.L.P. return receipt postcard (1 card) is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R. § 1.10 on this 12th day of January, 2001, addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.

Willie Jiles
Willie Jiles

Express Mail Receipt
No. 759174439US
Attorney Docket No.
066241.0117

Enclosed is a check in the amount of $1,600.00 for the filing fee. Please charge any additional fees or credit any overpayment to Deposit Account No. 02-0384 of Baker Botts L.L.P. **A duplicate copy of this sheet is enclosed.**

BAKER BOTTS L.L.P.
Attorneys for Applicants

Matthew B. Talpis
Registration No. 45,152

Date: January 12, 2001

0F-16-01 A

TRANSMITTAL FOR U.S. PATENT APPLICATION UNDER 37 CFR §1.53(b)

Box Patent Application
ASSISTANT COMMISSIONER
OF PATENTS
Washington, D.C. 20231

Attorney's Docket: 066241.0117

Sir:

Transmitted herewith for filing is the original patent application of:

Inventor: Keith A. Lowery, et al.

For: METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING

__X__ **Pursuant to 37 C.F.R. § 1.213(a), Applicant(s) respectfully requests that the above-identified patent application not be published under 37 U.S.C. § 122(b). Applicant(s) hereby certifies that the invention disclosed in the above-identified application has not been and will not be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication at eighteen months after filing.**

Enclosed are:

(1) __89_ pages of Specification, Claims and Abstract;

(2) ___4_ pages of Executed Declaration and Power of Attorney;

(3) __11_ pages of Formal Drawings;

(4) ___1_ page of executed Verified Statement Claiming Small Entity Status under 37 C.F.R. §§ 1.9(f) and .1.27(c)—Small Business Concern;

(5) ___4_ pages of an Assignment of the invention to epicRealm Inc.;

(6) ___2_ pages of a separate cover sheet in compliance with 37 C.F.R. § 3.28 and § 3.31 is included with an Assignment recordal fee of $40.00 pursuant to 37 C.F.R. § 1.21(h);

(7) ___2_ pages of an Information Disclosure Statement with PTO-1449 Form and attached references (10 references);

(8) ___2_ pages of a Preliminary Amendment;

(8) ___1_ page of a Certificate of Mailing by Express Mail No.: EL759174439US; and

(9) ___2_ pages of a this Transmittal Form with attached patent application filing fee in the amount of $1,600.00.

FEE CALCULATION					FEE
	Number		Number Extra	Rate	Basic Fee $ 355.00
Total Claims	105	20	85	X $ 9 =	$ 765.00
Independent Claims	15	3	12	X $40 =	$ 480.00
				TOTAL FILING FEE =	$1,600.00

Enclosed is a check in the amount of $1,600.00 for the filing fee. Please charge any additional fees or credit any overpayment to Deposit Account No. 02-0384 of Baker Botts L.L.P. **A duplicate copy of this sheet is enclosed.**

BAKER BOTTS L.L.P.
Attorneys for Applicants

Matthew B. Talpis
Registration No. 45,152

Date: January 12, 2001

Attorney's Docket No.:
066241.0117

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Keith A. Lowery, et al.

Filing Date: January 12, 2001

Title: METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING

Box: Patent Application
Assistant Commissioner for Patents
Washington, D.C. 20231

Dear Sir:

CERTIFICATE OF MAILING BY EXPRESS MAIL

I hereby certify that the attached Filing Fee Transmittal (2 pages), Patent Application (89 pages), a check in the amount of $1,600.00, Declaration and Power of Attorney (4 pages), Formal Drawings (11 pages), Assignment Cover Sheet (2 pages), Assignment (4 pages), a check in the amount of $40.00, Information Disclosure Statement (2 pages), PTO-Form 1449 (1 page) (with 10 references), Preliminary Amendment (2 pages), Verified Statement (1 page), and a Baker Botts L.L.P. return receipt postcard (1 card) is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R. § 1.10 on this 12th day of January, 2001, addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.

Willie Jiles
Willie Jiles

Express Mail Receipt
No. 759174439US
Attorney Docket No.
066241.0117

Enclosed is a check in the amount of $1,600.00 for the filing fee. Please charge any additional fees or credit any overpayment to Deposit Account No. 02-0384 of Baker Botts L.L.P. **A duplicate copy of this sheet is enclosed.**

BAKER BOTTS L.L.P.
Attorneys for Applicants

Matthew B. Talpis
Registration No. 45,152

Date: January 12, 2001

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Keith A. Lowery, et al.
Filing Date:	January 12, 2001
Title:	METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING

Assistant Commissioner
for Patents
Washington, DC 20231

Dear Sir:

PRELIMINARY AMENDMENT

Prior to the initial review of this patent application, please amend the application as follows:

IN THE SPECIFICATION

At page 65, line 22, please replace "29" with --30--.

REMARKS

No new matter has been added. The total number of claims has not been changed. Early and favorable acceptance of this Application is respectfully requested.

CONCLUSION

No other fees are believed to be due, however, the Commissioner is hereby authorized to charge any additional fees or credit any overpayments to Deposit Account No. 02-0384 of Baker Botts L.L.P.

Respectfully submitted,
BAKER BOTTS L.L.P.
Attorneys for Applicant

Matthew B. Talpis
Reg. No. 45,152

Correspondence Address:
Kevin J. Meek, Esq.
Baker Botts L.L.P.
2001 Ross Avenue, Suite 600
Dallas, Texas 75201
Phone: (214) 953-6680
Fax: (214) 661-4680

Date: January 12, 2001

METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING

TECHNICAL FIELD OF THE INVENTION

This invention relates in general to the field of data processing systems and, more particularly, to a method and system for dynamic distributed data caching.

BACKGROUND OF THE INVENTION

As computers have grown increasingly important in today's society, the importance of the Internet has also increased. As increasing numbers of users access the Internet, the need for efficient use of bandwidth has also increased. The increasing numbers of requests handled by the Internet are increasing the delay experienced by a user between generating a request and receiving a response to the request because of bandwidth limitations.

One traditional solution to decreasing bandwidth usage and decreasing the delay experienced by the user has involved caching previously requested content at the user's computer for faster retrieval. A related traditional solution has involved caching previously requested content for multiple users at a single cache server. Another traditional solution has involved increasing the bandwidth of the network connection between the Internet, the user and the web servers handling the requests. However, traditional solutions have often failed as the number of requests continue to increase and overload single cache servers and because of the expense associated with maintaining large numbers of high speed connections to the Internet. In addition, the traditional solutions have not utilized the "always-on" nature of newer broadband connections such as digital subscriber line and cable modems.

SUMMARY OF THE INVENTION

From the foregoing, it may be appreciated that a need has arisen for a method and system for dynamic distributed data caching to provide more efficient use of bandwidth.

According to one embodiment of the present invetion, a method for dynamic distributed data caching is provided. The method comprises providing a cache community comprising at least one peer. Each peer has an associated first content portion indicating content to be cached by the respective peer. The method further comprises allowing a client to join the cache community, updating a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community, and associating a respective second content portion with each peer based on the addition of the client. The second content portion is distinct from the first content portion.

According to another embodiment of the present invention, a system for dynamic distributed data caching is presented. The system comprises logic encoded on storage. The logic is operable to provide a cache community comprising at least one peer. Each peer has an associated first content portion indicating content to be cached by the respective peer and allow a client to join the cache community. The logic is further operable to update a peer list associated with the cache community to include the client. The peer list indicates the peers in the cache community. The logic is further operable to associate a respective second content portion with each peer based on the addition of the client. The second content portion is distinct from the first content portion.

According to a further embodiment of the present invention, a method for dynamic distributed data caching is presented. The method comprises determining that a first master associated with a cache community is non-operational, electing a second master to replace the first master in the cache community, and allocating at least one content portion based on the loss of the first master.

According to a yet another embodiment of the present invetion, a system for dynamic distributed data caching is presented. The system comprises logic encoded on storage. The logic is operable to determine that a first master associated with a cache community is non-operational, elect a second master to replace the first master in the cache community, and allocate at least one content portion based on the loss of the first master.

According to yet a further embodiment of the present invention, a method for dynamic distributed caching is presented. The method comprises requesting a list of cache communities from a cache server and determining whether at least one existing cache community exists. The method further comprises attempting to join a one of the existing cache communities when the existing cache communities are found and generating a new cache community when no existing cache communities are found.

According to another embodiment of the present invention, a system for dynamic distributed caching is presented. The system comprises logic encoded on storage. The logic is operable to request a list of cache communities from a cache server and determine whether at least one existing cache community exists. The logic is further operable to attempt to join a one of the existing cache communities when the existing cache communities are found and generate a new cache community when no existing cache communities are found.

According to a further embodiment of the present invention, a method for dynamic distributed data caching is presented. The method comprises generating a content request for requested content at a first peer in a cache community, determining a second peer associated with the requested content, the second peer being associated with the cache community, and retrieving, by the first peer, the requested content from the second peer.

According to yet another embodiment of the present invention, a system for dynamic distributed data caching is presented. The system comprises logic encoded on storage. The logic is operable to generate a content request for requested content at a first peer in a cache community, determine a second peer associated with the requested content, the second peer being associated with the cache community, and retrieve, by the first peer, the requested content from the second peer.

According to yet a further embodiment of the present invention, a method for dynamic distributed data caching is presented. The method comprises communicating a community request from a dynamic cache module to an administration module and receiving a community list from the administration module in response to the community request, the community list including a list of communities. The method further comprises generating a join request to attempt to join a one of the communities in the community list and receiving an allow message associated with the one of the communities. The method further comprises receiving a peer list associated with the one of the communities, receiving a content request, and storing content associated with the content request.

According to an additional embodiment of the present invention, a system for dynamic distributed data caching is presented. The system comprises logic encoded on storage. The logic is operable to communicate a community request from a dynamic cache module to an administration module and receive a community list from the administration module in response to the community request. The community list includes a list of communities. The logic is further operable to generate a join request to attempt to join a one of the communities in the community list and receive an allow message associated with the one of the communities. The logic is further operable to receive a peer list associated with the one of the communities, receive a content request, and store content associated with the content request.

According to a further additional embodiment of the present invention, a system for dynamic distributed data caching is presented. The system comprises means for providing a cache community comprising at least one peer. Each peer has an associated first content portion indicating content to be cached by the respective peer. The system further comprises means for allowing a client to join the cache community and means for updating a peer list associated with the cache community to include the client. The peer list indicates the peers in the cache community. The system further comprises means for associating a respective second content portion with each peer based on the addition of the client. The second content portion is distinct from the first content portion.

According to yet a further additional embodiment of the present invention, a system for dynamic distributed data caching is presented. The system comprises means for determining that a first master associated with a cache community is non-operational, means for electing a second master to replace the first master in the cache community, and means for allocating at least one content portion based on the loss of the first master.

According to yet another further additional embodiment of the present invention, a system for dynamic distributed caching is presented. The system comprises means for requesting a list of cache communities from a cache server and means for determining whether at least one existing cache community exists. The system further comprises means for attempting to join a one of the existing cache communities when the existing cache communities are found and means for generating a new cache community when no existing cache communities are found. According to another additional embodiment of the present invention, a system for dynamic distributed data caching is presented. The system comprises means for generating a content request for requested content at a first peer in a cache community and means for determining a second peer associated with the requested content. The second peer is associated with the cache community. The system further comprises means for retrieving, by the first peer, the requested content from the second peer.

According to yet a further additional embodiment of the present invention, a system for dynamic distributed data caching is presented. The system comprises means for communicating a community request from a dynamic cache module to an administration module and means for receiving a community list from the administration module in response to the community request. The community list includes a list of communities. The system further comprises means for generating a join request to attempt to join a one of the communities in the community list and means for receiving an allow message associated with the one of the communities. The system further comprises means for receiving a peer list associated with the one of the communities, means for receiving a content request, and means for storing content associated with the content request.

The present invention provides a number of technical advantages. Various embodiments of the present invention may provide all, some or none of these advantages. One such technical advantage is the capability to support a dynamic distributed caching system. In addition, the distributed caching system is supportable without the use of specialized hardware as standard personal computers may be used to support the distributed caching system. A further technical advantage is decreased utilization of expensive connections to the Internet and increased utilization of cheaper local area network connections and broadband connections, such as digital subscriber line and cable modems. By caching content at local machines on a local area network or on broadband connections to an Internet Service Provider, response time to requests for content is decreased by retrieving the content from local machines. Additional benefits may be realized by allowing more client machines to utilize a single connection to the Internet by decreasing the amount of bandwidth needed by particular client machines.

Another technical advantage is the capability to dynamically add and remove members from a distributed cache community. In contrast to traditional distributed caching systems, which have typically required a human administrator to add and remove members from the distributed caching system, the present invention provides the capability to dynamically add and remove members from the distributed cache community. Also, members may be added or removed from the cache community without the intervention of a human administrator. The present invention also reallocates the data to be cached by particular members of the distributed cache community based on the addition and subtraction of members to the distributed cache community.

BRIEF DESCRIPTION OF THE DRAWINGS

A better understanding of the present invention will be realized from the detailed description that follows, taken in conjunction with the accompanying drawings, in which:

FIGURE 1 is a block diagram illustrating a community cache system;

FIGURE 2 is a block diagram illustrating an exemplary community cache constructed according to the teachings of the present invention;

FIGURE 3 is a flowchart illustrating a method for community caching according to the teachings of the present invention;

FIGURE 4 is a flowchart illustrating a method for generating a community cache according to the teachings of the present invention;

FIGURE 5 is a diagram illustrating an exemplary distribution of cache shares according to the teachings of the present invention;

FIGURE 6 is a block diagram illustrating a dynamic caching system according to one embodiment of the system of FIGURE 1;

FIGURE 7 is a block diagram illustrating details of the dynamic cache application according to one embodiment of the present invention;

FIGURE 8 is a flow diagram illustrating a method for retrieving and caching content within a cache community according to one embodiment of the present invention;

FIGURE 9 is a flow chart illustrating a method for adding a client to the cache community according to one embodiment of the present invention;

FIGURE 10 is a flow chart illustrating a method for allowing the client to join the cache community according to one embodiment of the present invention;

FIGURE 11 is a flow chart illustrating a method for determining whether a member of the cache community has unexpectedly departed the cache community according to one embodiment of the present invention;

FIGURE 12 is a flow chart illustrating a method for gracefully removing the member from the cache community according to one embodiment of the present invention;

FIGURE 13 is a flow chart illustrating a method for gracefully removing a master from the cache community according to one embodiment of the present invention; and

FIGURE 14 is a flow chart illustrating a method for allowing the master to unexpectedly depart the cache community according to one embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

FIGURE 1 is a block diagram illustrating a community cache system 10. System 10 comprises one or more Internet Service Provider (ISP) clients 12, an ISP 14, an ISP caching community 15, a network 16, an Intranet caching community 18 and an origin server 19.

Client 12 comprises a processor 20, a computer readable memory 22, a computer readable storage device 24, a cache module 26 and a browser 30. Client 12 may be adapted to execute any of the well-known MS-DOS, PC-DOS, OS/2, UNIX, Linux, MAC-OS, mainframe, minicomputer and Windows operating systems or other operating systems. Processor 20 comprises any suitable general purpose or specialized electronic processing device, such as a central processing unit (CPU), operable to communicate with memory 22 and storage device 24, and further to execute cache module 26 and browser 30. Memory 22 comprises any suitable combination of transient or persistent memory operable to store cache module 26 and browser 30, and to communicate with processor 20. Storage device 24 comprises any suitable combination of optical, magnetic or other computer readable storage medium such as a floppy disk drive, a hard disk drive, a CD-ROM drive, a CD-RW drive, a magnetic tape drive or an optical drive. Storage device 24 may also represent multiple computer readable storage devices. Storage device 24 includes a cache portion 28.

Cache portion 28 comprises a portion of storage device 24 used by cache module 26 for caching data. Access to cache portion 28 may be controlled by cache module 26 so as to prevent user modification of data stored in cache portion 28. Cache portion 28 may comprise one or more directories, one or more logical partitions, one or more distinct physical devices and other suitable physical and logical elements.

Cache module 26 comprises a software application operable to manage cache portion 28 of storage device 24. Cache module 26 is operable to monitor the activities of browser 30 and to cache content items retrieved by browser 30. Cache module 26 is also operable to respond to content requests from browser 30 using content cached in cache portions 28 at clients 12 in community 15. In one embodiment, cache module 26 may use the Cache Array Routing Protocol (CARP) to determining the location of content within community 15. Cache module 26 is configurable such that limits may be placed on the size of cache portion 28 and the amount of processor time used on processor 20 by cache module 26. For example, a user associated with a client 12 may configure the cache module 26 associated with that client 12 to use only 5% of the storage space and no more than 10% of the processor time. For another example, a user associated with a client 12 may configure the cache module 26 associated with that client 12 to only provide caching services when the system is idle, such as when a screen saver is active or when processor usage by the user of the client 12 is below a particular threshold. In general, resource limits may be associated with cache module 26 such that cache module 26 is prevented from consuming more than a predetermined amount of the resource. The resources may comprise any of an amount of processor time on processor 20, an amount of bandwidth on link 13, an amount of storage space on storage 24, an amount of memory 22 and other computing resources associated with client 12. Cache module 26 is further operable to collect statistical information associated with link 13, browser 30, client 12, portion 28, cache module 26 and other elements in community 15.

Cache module 26 is further operable to encrypt data stored in cache portion 28. Cache module 26 may use any suitable symmetric and/or asymmetric encryption system for encrypting data in cache portion 28. For example, cache module 26 may use public-key/private-key encryption, the U.S. Data Encryption Standard (DES), the Twofish algorithm, the Blowfish algorithm and other suitable encryption systems. Encrypting data stored in cache portion 28 prevents a user associated with client 12 from unrestrictedly accessing and modifying cached content. Encryption also provides privacy as the user of any particular client 12 in community 15 is prevented from viewing the data retrieved by other users in community 15.

The increasing use of "always-on" Internet connections with large bandwidth capacities allows for the use of a distributed caching system using non-specialized equipment. Note that as used herein, an "always-on" connection is defined as a data connection between a client computer, such as a personal computer, and a network, such as the Internet, which operates without competing with other devices associated with a user of the client computer. In addition, an "always-on" connection as used herein may be off and may cycle between being on and off at unpredictable intervals. Stated another way, an "always-on" connection has the capability to be continuously active without interfering with other devices usable by the user associated with the client computer, but the "always-on" connection is not required to be literally "always-on". For example, a traditional analog modem is typically active only when a user of the modem explicitly activates the analog modem and prevents the use of the phone line by the user of the analog modem for voice communication when the analog modem is active. Typically, an analog modem is not considered as creating an "always-on" connection. For another example, a digital subscriber line (DSL) modem, a cable modem and a local area network (LAN) have the capability to be continuously active without interfering with the use of the phone line or the cable television while active. Typically, a DSL modem, a cable modem and a LAN would be considered as creating "always-on" connection.

The growing availability of "always-on" connections provides the opportunity to utilize the processing power and storage capacity of computers connected via "always-on" connections to the Internet or other networks. Many computers with "always-on" connections utilize only a small portion of their processing power and storage space. For example, many DSL and cable modem connected personal computers remain completely unused during the day when their owner is at work and this unused processing and storage capacity may be used to form a distributed caching system. In addition, even when being used, many computers with "always-on" connections may have significant unused storage and processing capability. System 10 provides the capability to harness the unused processing power and storage capacity of standard computers to create a distributed caching system.

In return for the use of processing power and storage capacity on computers to form the distributed caching system, a provider of system 10 may provide incentives to the users of clients 12. For example, an ISP, such as 14, may deploy cache modules 26 to subscribers of the ISP in return for a lower subscription cost. For another example, an ISP could use system 10 as part of a value-added service of faster internet service.

Each cache module 26 is further operable to generate a cache status message 27. Cache status message 27 comprises a indication of whether a particular cache module 26 is active or inactive. More specifically, cache status message 27 includes a "cache on" or a "cache off" indication. The "cache on" indication indicates that the associated cache module 26 has gone from an inactive to an active state and is requesting to join community 15. The "cache off" indication indicates that the associated cache module 26 is going from an active to an inactive state and is requesting removal from community 15. Cache status message 27 may also include an "active" indication. The active indication indicates that the associated cache module 26 is currently active, and caching content and handling requests for content from browsers 30. The active indication may operate as a heartbeat indicating that the associated cache module 26 is alive.

Each cache module 26 further comprises a distinct location table 29. Location table 29 comprises one or more indications of the location within community 15 of cached content. More specifically, location table 29 indicates which client 12 is responsible for caching which content. In general, table 29 may use any suitable indication for indicating which clients 12 to cache content at, such as IP addresses, domain names, portions of URLs or a hash value based on a content request from browser 30.

Cache modules 26 may be further operable to provide increased anonymity while surfing to users of clients 12. More specifically, cache module 26 may remove specific lines in HTTP requests, such as requests 32, sent to retrieve web pages from origin server 19. This feature assists in preventing the sending of certain personal information to origin servers 19. For example, browser 30 typically requests a web page from origin server 19 by generating a request 32. Request 32 may include a header portion that may include information about client 12 and the associated user, such as an email address and what web sites that the user has visited. Cache module 26 may intercept request 32 and remove parts of the header portion which are not required to retrieve web pages. In one embodiment, cache module 26 comprises a Microsoft Windows application. The cache module 26 is operable to collect statistics such as the number of cache hits, the number of requests received by cache module 26, the hit percentage, the amount of data cached, the maximum capacity of cache portion 28, and how long cache module 26 has been active. Cache module 26 may also allow a user to configure various aspects of cache module 26, such as the size of cache portion 28, enable or disable anonymous surfing and purge cache portion 28.

Browser 30 executes on client 12 and comprises any suitable Hypertext Transport Protocol (HTTP) client. In the disclosed embodiment, browser 30 comprises a web browser such as Internet Explorer® by Microsoft Corp. of Redmond, Washington, or Netscape Communicator by Netscape Communications Corp. Browser 30 is held in memory 22 and executed by processor 20. Browser 30 transmits and receives data over link 13. Each browser 30 is operable to generate one or more content requests 32. Each request 32 is generated by a particular browser 30 in response to a desire for information by a user (not shown) associated with that particular browser 30. Each request 32 comprises a request for one item of content from origin server 19. In the disclosed embodiment, request 32 comprises an HTTP request. The HTTP request comprises a Uniform Resource Locator (URL) portion identifying the requested item of content and a header portion containing further information about the HTTP request. The requested item of content may further comprise multiple elements of content, for example, a web page with multiple graphic elements therein, but the request is directed to retrieving the single content item associated with the URL while the remaining elements of content in the main content item are retrieved as a function of the single content item.

Content comprises static information and dynamic information. Static information typically does not change once the content has been created or does not change except slowly over time. Stated another way, static information comprises content that, when created, is expected to remain the same for an indeterminate amount of time. Static information may comprise, for example, text files and pre-existing Hypertext Markup Language (HTML) web pages. For example, a restaurant menu is the same regardless of the user who is accessing the menu because the menu is the same for all people. However, as chefs at the restaurant change over time, the menu may also change.

In contrast, dynamic information comprises content that is expected and designed to change. The dynamic content may change based on the data and criteria used for generating the dynamic content, such as a search result page. For example, a search for "mountain lions" will dynamically generate a search result page listing items related to "mountain lions." The dynamic content may also change based on information associated with the page, for example, a page in an online product catalog may change in response to pricing and availability changes, and dynamic content may further change for other suitable reasons. Dynamic information may comprise, for example, the output of Common Gateway Interface (CGI) scripts, search engine results, the output of Structured Query Language (SQL) searches and the SQL search itself, JavaScript, the output of JavaScript scripts, ActiveX controls and the output of ActiveX controls, Active Server Pages, applets and any other suitable dynamic content generation system. Often, the dynamic information will be expressed as a single HTML web page, but the information within the HTML web page has been generated dynamically based on some suitable criteria. For example, the result of a search using a search engine on the Internet returns different information based on the search terms provided by the user. The search results may also depend on one or more attributes associated with the search request, such as geographic location or the current date. Continuing the search engine example, when the user is searching for local businesses, the search results will vary not only on the search terms provided by the user, but also what "local" means with respect to the search request. For another example, a user searching for information about current events will want results that are tailored to the user's search terms, the user's location, and the current date. For yet another example, an online book retailer may provide price information and consumer reviews of books available from the online book retailer. The price information for a particular book may change unexpectedly in response to a sale or a sudden interest in that particular book. Also, the web page listing the book and the consumer reviews changes in response to a consumer entering a new review of the book.

Referring again to FIGURE 1, communications link 13 comprises a data communications link operable to communicate data between clients 12 and ISP 14. In the disclosed embodiment, link 13 comprises an "always-on" type connection, such as digital subscriber line (DSL) or a cable modem.

ISP 14 comprises a point of presence on network 16 for communicating data from clients 12 to remote locations. ISP 14 may also define the boundary of community 15.

Community 15 comprises a plurality of clients 12 at whom content items retrieved by browsers 30 may be cached in cache portions 28. Community 15 represents a group of clients 12 which cooperate to form a distributed caching system using cache module 26 and portion 28. Requests by browsers 30 within community 15 for content cached within community 15 do not need to be propagated over network 16 since the requested content is available within community 15.

Network 16 comprises any suitable data transport network, such as a Transmission Control Protocol/Internet Protocol (TCP/IP) network, an asynchronous transfer mode (ATM) network, an Ethernet network, or a frame-relay network. In the disclosed embodiment, network 16 comprises the Internet.

Community 18 represents an exemplary cache community based around a corporate Intranet. The distributed caching capabilities of system 10 are not limited to home computers. A corporation may utilize cache modules 26 in order to improve web performance for the corporation's employees and/or to decrease the amount of Internet connectivity the corporation purchases. The 10 megabit, 100 megabit, gigabit and faster LAN technologies used by corporations are well suited to the distributed cache of system 10. LAN technologies may be considered as providing an "always-on" connection within the scope of the present invention. Other collections of computers may also form cache communities, communities 15 and 18 represent two examples of possible cache communities.

Community 18 may comprise a corporate intranet having a communications interface 50, a LAN 52 and a plurality of intranet clients 54. Interface 50 comprises a communication interface between LAN 52 and Internet 16. For example, interface 50 may comprise a firewall, a router or other suitable communications interfaces. Interface 50 may also define the boundary of community 18. LAN 52 comprises any suitable local area network, such as Ethernet, ATM, or TCP/IP. Intranet clients 54 are similar to ISP clients 12 except that clients 54 are members of an Intranet. Community 18 operates similarly to community 15, except as otherwise noted.

Origin server 19 communicates data over network 16. Origin server 19 comprises any suitable hardware and/or software executing on a computer for receiving and responding to requests 32. Origin server 19 may comprise a single computer executing software or may comprise a plurality of computers each executing software. In the disclosed embodiment, origin server 19 comprises an HTTP server which may also be known as a web server. Origin server 19 may additionally support other protocols such as the file transfer protocol (FTP). Origin server 19 retrieves information from one or more data sources (not shown), such as a storage device coupled to server 19 or other origin servers, in response to requests 32. Origin server 19 is operable to retrieve static content, such as prewritten text files, images, and web pages, from the data source in response to requests 32. Origin server 19 is also operable to generate new, dynamic content, for example, by dynamically creating web pages based on content stored in the data source in response to requests 32. For example, origin server 19 may generate a new web page using a common gateway interface (CGI) script, generate a new web page from the result of a structured query language (SQL) request and perform other suitable content generation functions. Origin server 19 may also be operable to generate executable software, such as applications and applets, in response to requests for data. For example, origin server 19 may generate a Java applet in response to an appropriate request 32.

In operation, browser 30 generates request 32 for content. Operation of system 10 will be described with respected to cache community 15, however, it should be noted that cache community 18 operates similarly using clients 54. Cache module 26 then intercepts request 32 before request 32 is communicated to network 16. Cache module 26 examines request 32 to determine whether the requested content is available in community 15. If the requested content is available in community 15, cache module 26 retrieves the requested content from the appropriate storage portion 28 within community 15 and returns the requested information to the browser 30 which requested it. If the requested content is not available within community 15, then cache module 26 forwards request 32 over link 13 to ISP 14 for normal handling. Similarly, a request 32 generated by a browser on a client 54 is intercepted by cache module 26 to determine whether the requested content is available within community 18.

Cache module 26 may be configured to control the amount of processor power, storage space and bandwidth of a particular client 12 used by community 15. The client-by-client control of usage available to clients 12 allows for individual tailoring of community 15 to particular clients 12. The client-by-client control of usage also allows for different incentive plans for subscribers if ISP 14. For example, a subscriber to ISP 14 may have a second computer separate from the computer normally used by the subscriber. The subscriber with two computers could dedicate a large percentage of processor 20 and storage 24 to community 15 in exchange for ISP 14 providing a second IP address for the second computer over a DSL type link 13 for free. For another example, a subscriber of ISP 14 may dedicate increasing proportions of the bandwidth available over link 13 in return for decreasing subscription costs, such as 40% for a 40% discount or 50% for a 50% discount. Community 18, representing a corporate intranet, may allow for centralized control of the percentage of the processing power, storage and bandwidth used by community 18, such as by a corporate information technology (IT) department.

In one embodiment, cache module 26 may cache content using a conservative mode or an aggressive mode. When in the conservative mode, cache module 26 caches content received by browser 30 which is marked as cacheable. When in the aggressive mode, cache module 26 caches all content unless the content has been explicitly marked as non-cacheable. In general, by caching all content, unless the content is listed as non-cacheable, more content may be cached in comparison to conservative mode caching.

Cache modules 26 using aggressive mode caching may further include communicating with a data center. More specifically, cache module 26 may communicate with the data center to inform the data center of data cached by cache module 26.

Aggressive mode caching may use a content expiration protocol to avoid providing expired, but cached, content. The data expiration protocol may use data expiration commands to inform cache modules 24 that data at an origin server 19 has changed. Alternatively, a single cache module within a community 15, such as the master node discussed below, may communicate with the data center. By informing the data center of data cached within community 15, the data center can send data expiration commands to community 15 so that cache modules 24 can mark cached content as expired.

The data expiration command comprises any suitable message for expiring data stored by cache module 24. In one embodiment, the data expiration command comprises an Internet Cache Synchronization Protocol (ICSP) message. The ICSP message may expire any of a single web page, a plurality of web pages at a single web site, a plurality of web pages at a plurality of web sites, a plurality of sites within a single domain and one or more specific objects on a web page, such as an image. For example, the ICSP message may expire a single web page such as http://www.domain.com/web_page.html. The ICSP message may expire a plurality of web pages at a single web site such as http://www.domain.com/web_pages/* which would expire all web pages in the "web_pages" directory at the site www.domain.com. The ICSP message may expire a plurality of pages at a plurality of web sites such as http://*.domain.com/web_pages/* which would expire all web pages in the "web_pages" directory of each site in "domain.com". The ICSP message may expire a plurality of web sites such as http://*.domain.com which would expire all web pages on all sites in "domain.com". For another example, a single active server page (ASP) may result in many individual cached pages. A single ASP page may generate large numbers of individual cached pages because a single ASP page can dynamically create multiple different specific pages, such as http://www.domain.com/product_description.asp?category=__&product=__ may specify an ASP page that generates many specific product description pages based on a specified category and product. To cache module 24, product_description.asp?**category=5**&product=7, product_description.asp?category=5&**product=6** and product_description.asp?**category=3**&product=7 may specify three different cached pages. By expiring product_description.asp?category=*&product=7 ALL categories of product seven are expired, while not affecting product six related pages even though all of the pages were generated from a single ASP page.

In general, the data center may generate the data expiration command in response to a change in the content at origin server 19. The data center may also generate the data expiration command in response to the elapsing of a predetermined time period. ICSP supports the synchronization of cached content in community 15 with updated content available at origin server 19. Further details of ICSP are described in the patent application entitled "Method and Apparatus for Content Synchronization" by inventors Keith A. Lowery, et al., filed on June 8, 2000, serial number 09/590,760, which is incorporated herein by reference.

In addition, cache module 26 may provide a guaranteed click delivery capability. The guaranteed click delivery capability comprises the capability to regularly check whether a particular web page is available and to retrieve the web page when the web page becomes available. For example, a user of client 12 may attempt to retrieve a particular web page. The server providing that web page may be currently overloaded and unable to provide the requested web page. Cache module 26 may detect that the server is too busy, such as when the web site returns a "server too busy" error message, and then automatically attempt to retrieve the request web page from the busy server. For example, a busy server may comprise a server which is currently processing substantially all the requests 32 which the server is capable of handling. For another example, a busy server may comprise a server which is providing content and using substantially all of the bandwidth available to the server. In general, a busy server may comprise a server which is incapable of processing more requests 32 at a given time for one or more reasons. Cache module 26 may then display the retrieved web page in browser 30 or may abandon the attempt to retrieve the web page after a predetermined period of time has elapsed without successfully retrieving the requested web page. Cache module 26 may also ask the user whether the user wants cache module 26 to attempt to retrieve the requested web page from the busy server. Typically, cache module 26 would attempt to retrieve the requested web page from the busy server while the user retrieves and views other web pages from other origin servers 19. Stated another way, cache module 26 would attempt to retrieve the requested web page in the background while the user may also be performing other tasks.

Yet another capability of cache module 26 is the ability to provide a screen saver to a user associated with client 12. The screen saver displays a graphical representation of the user's response time to one or more origin servers 19. For example, the response time between client 12 and a particular web site may be displayed in a graphical manner. More specifically, the screen saver displays a solar system-like graph with client 12 generally in the center and the distance between client 12 and other web sites displayed based on the round-trip ping time between client 12 and the other web sites.

FIGURE 2 is a block diagram illustrating an exemplary community cache 100. Community 100 comprises a first client 102, a second client 104, a third client 106 and an ISP 108. In the exemplary embodiment of FIGURE 2, clients 102, 104 and 106 represent specific examples of clients 12 of FIGURE 1.

Clients 102, 104 and 106 communicate with ISP 108 over respective communication links 110. Client 102 comprises a browser 120, storage 122, and a cache module 124. Client 104 comprises a browser 140, storage 142, and a cache module 144. Client 106 comprises a browser 160, storage 162, and a cache module 164.

Browsers 120, 140 and 160 represent distinct examples of browsers 30 of FIGURE 1. Each of storage 122, 142, and 162 respectively support cache portions 126, 146, and 166. Storage 122, 142, and 162 represent distinct examples of storage 24 of FIGURE 1. Cache portions 126, 146, and 166 represent distinct examples of cache portions 28 of FIGURE 1. Cache modules 124, 144, and 164 support respective location tables 128, 148 and 168. Each of cache module 124, 144 and 164 are operable to generate respective cache status messages 130, 150 and 170. Cache modules 124, 144, and 164 represent distinct examples of cache modules 26 of FIGURE 1. Location tables 128, 148 and 168 represent distinct examples of location table 29 of FIGURE 1. Cache status messages 130, 150 and 170 represent distinct examples of cache status message 27 of FIGURE 1.

Location tables 128, 148, and 168 respectively comprise one or more indications of which client 102, 104 or 106 to cache content in response to requests 32 from browsers 120, 140, and 160. For example, location table 128 may indicate that content identified by URLs having a domain name beginning with A-D is cached at client 102, while domain names E-H are cached at client 104 and domain names H-Z are cached at client 106. For another example, location tables 128, 148 and 168 may indicate particular ranges of IP addresses to be cached at particular clients 102, 104 and 106. In general, tables 128, 148 and 168 may use any suitable indication for indicating which clients 102, 104 and 106 to cache content at, such as IP addresses, domain names, portions of URLs or a hash value based on request 32.

Cache status messages 130, 150 and 170 each comprise a message respectively generated by modules 124, 144 and 164 to indicate to other modules 124, 144 and 164 that the generating module is activating or deactivating its caching functionality. For example, when cache module 124 at client 102 is activated it may generate a cache status message 130 indicating caching is active at client 102.

Communications link 110 comprises any suitable data communications system. In the exemplary embodiment of FIGURE 3, communications link 110 comprises a "always-on" link, such as DSL or a cable modem, similar to link 13 in FIGURE 1. ISP 108 comprises an Internet service provider operable to communicate data between clients 102, 104 and 106 and network 16 similar to ISP 14 of FIGURE 1.

In operation, in one embodiment, community 100 may be formed by dynamically seeking out other active instances of cache module 26. Then, based on a set of performance heuristics, clients 12 are bonded together under favorable conditions. Cache module 26 may use dynamic affiliation algorithms to build and manage communities 100. More specifically, on startup, cache module 26 may communicate with a remote directory provider for assistance in finding other cache modules 26 with which to form a community 100. Using the assistance from the remote directory provider, the client may attempt to contact and possibly join a currently existing community 100. If no communities 100 are found or found communities 100 do not allow cache module 26, then cache module 26 may attempt to start its own cache community. Alternatively, if no remote directory is available, cache module 26 searches for communities 100 itself. For example, cache module 26 may send a broadcast and/or a multicast message looking for communities 100.

Each community 100 includes a master node and, optionally, one or more normal nodes. A master node comprises a cache module 26 on a particular client 12 which is responsible for supervising the addition and departure of clients from community 100. The master node receives data associated with the addition of a client 12 to community 100 and the departure of a client 12 from community 100 and communicates the data to the other members of community 100. Any cache module 26 may function as the master node. Any suitable method for electing the master node may be used by cache modules 26. For example, a cache module 26 which has been activated the longest may be selected as the master, with ties being resolved randomly. The departure of the master node causes the election of a new master node by the remaining members of community 100.

Community 100 handles the graceful and non-graceful departure of clients 12 from community 100. A graceful departure comprises an intentional departure of a client 12 from community 100. For example, a graceful departure may occur when a user deactivates cache module 26. A non-graceful departure comprises an unexpected departure of a client 12 from community 100. For example, a non-graceful departure may occur when a client 12 suddenly crashes and ceases operation. When an active cache module 26 shuts down, for example, the cache module 26 requests to leave community 100 and the request circulates through the remaining community members. The remaining community members would then discontinue forwarding requests to that client 12. In a non-graceful scenario, a managing peer (known as a Master) watches for dead peers and notifies the rest of a community if this condition is detected.

Similarly, the managing peer may depart gracefully or non-gracefully. A graceful departure of the managing peer comprises the managing peer informing community 100 that the managing peer is leaving community 100. An election is then held by the remaining members of the peer to select the new managing peer. When a non-graceful departure occurs, such as when the managing peer crashes, a cache module 26 may detect that the managing peer is not responding and call an election to create a new managing peer. In general, any suitable techniques may be used to handle the arrival and departure of cache modules 26 from community 100, and to create and maintain the managing peer.

FIGURE 3 is a flowchart illustrating a method for community caching. For increased clarity, the operation of the exemplary cache community 100 of FIGURE 2 will be described in association with FIGURE 3. The method begins at step 200 where browser 120 generates a request 32 for content. The URL portion of the request 32 identifies the requested content as "http://www.server.com/jobs/listing.html". Next, at step 202 module 124 intercepts request 32 generated by browser 120. Then, at step 204, cache module 128 determines the URL associated with request 32.

Proceeding to step 206, cache module 124 determines the location where the content associated with the URL associated with request 32 would be cached. More specifically, cache module 124 determines which of the storage portions 126, 146 or 166, would store the requested content based on information in location table 128. In the example of FIGURE 2, location table 128 indicates that the domain "server.com" would be cached at client 106. Next, at step 208, cache module 124 checks storage portion 166 for the requested content and at decisional step 210, determines whether the requested content has been cached. More specifically, cache module 124 queries cache module 164 to determine whether the content associated with the URL in request 32 has been cached in portion 166 at client 106. If cache module 164 replies that the requested content is cached in portion 166, then the YES branch of decisional step 210 is followed to step 212 where the requested content is retrieved from storage portion 166 and, at step 214, is displayed at browser 120.

If the requested content is not cached at portion 166, indicating that the requested content is not available within community 100, then the NO branch of decisional step 210 leads to step 216. At step 216, the requested content is retrieved from origin server 19 since the requested content is not cached within community 100. The requested content is then displayed on browser 120 at step 214 and, at step 218, a copy of the requested content is communicated to cache module 164 for storage in portion 166. Then, at step 220, the retrieved content is stored in portion 166 by module 164.

Returning to the start state of FIGURE 3, steps 230 and 232 are typically performed in parallel to the processing previously described for FIGURE 3. At step 230, modules 26 generate cache status messages 27 with an "active" indication and broadcast messages 27 to clients 12 (or 54) in community 15 (or 18). Message 27 with an "active" indication may be used by modules 26 to indicate that the associated cache module 26 is alive. Then, at step 232, modules 26 wait a predetermined period of time before broadcasting another message 27. Modules 26 wait so as not to flood community 15 (or 18) with messages 27 and waste bandwidth. In one embodiment, the predetermined wait period is five seconds.

Modules 26 which fail to broadcast the "active" indication may be removed from the cache community because the associated cache module 26 may have failed, other suitable actions may be taken in response to a failure to broadcast the "active" indication.

The method described by FIGURE 3 is shown to end after steps 214 and 220, this indicates that the processing for the request generated in step 200 has been completed. The method described by FIGURE 3 is expected to be repeated a plurality of times within the scope of the invention.

FIGURE 4 is a flowchart illustrating a method for generating a cache community, such as 15, 18 or 100. For increased clarity, the exemplary community 100 of FIGURE 2 is discussed in association with FIGURE 4. The method begins at step 300 where cache module 144 is initiated. Initiating module 144 may comprise activating module 144, such as by activating client 104. Next, at step 302, cache module 144 generates cache status message 150 with a "cache on" indication and broadcasts cache status message 150 to clients 102 and 106 in community 100. Message 150 may be broadcast using any suitable method supported by ISP 108 and links 110. For example, a TCP/IP broadcast or multicast message may be used to reach clients 102 and 106 associated with ISP 108. For another example and referring back to FIGURE 1, community 18 may use an Ethernet broadcast message to reach all clients 54 in community 18.

At step 303, a primary distribution of portions of content to be cached ("cache shares") is negotiated between modules 124, 144 and 164 within community 100. The primary distribution of cache shares may be determined by splitting up URLs alphabetically by domain name, by generating a unique hash value for each URL and distributing the hash values among clients 102, 104 and 106, by IP address range or by any other suitable method or combination of methods. The distribution method to use may be configured for modules 124, 144 and 164, such as by an administrator. The actual primary distribution of cache shares may comprise a simple fractional split between the members of a cache community, such as 15, 18 (FIGURE 1) or 100, or may also consider other elements. For example, the actual distribution may consider the speed of links 13 (FIGURE 1) or 110, the processing power of processors 20 (FIGURE 1) and/or the size of portions 28 (FIGURE 1), 126, 146 and 166. Historical and statistical information may be considered when determining the primary distribution. For example, a particular client, such as 12 or 54, which regularly reboots may receive a smaller a cache share than a client which has a historically high up time.

With respect to the example in FIGURE 2, modules 124 and 164 may have already negotiated cache shares based on an alphabetical distribution where URLs having domains starting with A-M are cached at client 102 and domains starting with N-Z are cached at client 106. Modules 124, 144 and 164 may now renegotiate cache shares such that A-H domains are cached at client 102, I-P domains are cached at client 104 and Q-Z domains are cached at client 106 using a simple fractional alphabetic distribution.

Proceeding to step 304, a secondary distribution may optionally be negotiated. For increased reliability, cache modules 26 may be configured to negotiate the secondary distribution. The secondary distribution represents a distribution of cache shares to decrease the impact of the failure of clients 12. The secondary distribution may be determined using any of the information and methods used in determining the primary distribution in step 303. For example, cache modules may negotiate a secondary distribution of cache shares such that multiple clients 12 are caching the same content so that if a particular client 12 fails, a backup of the cached content exists within the community.

FIGURE 5 is a diagram illustrating an exemplary distribution of cache shares. For increased clarity, FIGURE 5 is discussed in association with FIGURES 2 and 4. The primary distribution of cache shares described previously in associated with step 303 is shown in FIGURE 5 as "primary" where domains A-H were allocated to client 102, domains I-P were allocated to client 104 and domains Q-Z were allocated to client 106. An overlapping secondary distribution may be used to allocate portions of the primary distribution to other clients for increased reliability. For example, an overlapping secondary distribution may allocate I-L and Q-U to client 102, A-D and V-Z to client 104 and E-H and M-P to client 106 as shown in the "secondary" line of FIGURE 5. Alternatively, the secondary distribution may simply rotate cache shares, such as having A-H allocated to client 106, I-P allocated to client 102 and Q-Z allocated to client 104 and is shown as "alt. secondary" in FIGURE 5. Any suitable secondary distribution may be used depending on the configuration of cache modules 26 (FIGURE 1), 124, 144 and 164 (FIGURE 2). It should be noted that the cache shares shown in FIGURE 5 are provided for increased clarity and are merely exemplary, various other primary and secondary distributions of cache shares are usable within the scope of the invention.

In addition, tertiary and further distributions may be negotiated to provide multiple layers of backup within cache communities 15, 18 and 100 at step 304. The determination of how many layers of backup to provide may be based on the reliability of clients 12, 54 (FIGURE 1), 102, 104 and 106

(FIGURE 2), the delay associated with checking backup cache modules 26 versus the delay associated with retrieving the data from origin server 19, bandwidth considerations with community 15, 18 or 100 and other criteria.

Then, at step 305, once the primary, secondary and other distributions are determined, location tables 128, 148 and 168 are updated to reflect which client 102, 104 and 106 is caching which content.

Then, at step 306, any redistribution of previously cached content for supporting a newly negotiated distribution occurs. Redistribution may comprise actually copying cached content from one client, such as 12 or 54 (FIGURE 1), to another, or removing or expiring content no longer cached at a particular server. Simply removing the cached items may cause a request that could have been satisfied by the community cache to be forwarded to the origin server, but avoids the bandwidth usage associated with copying. The decision whether to copy or remove may be configured at modules 124, 144 and 164 and may consider bandwidth usage issues, the amount of content to copy and other criteria. For example, modules 124, 144 and 164 may be configured to perform a copy of cached content with the content comprises less than five megabytes of data and to perform a removal when more than five megabytes of content are involved.

Continuing the example of FIGURE 2, the distribution has changed such that domains starting with I-M, previously cached at client 102, and domains starting with N-P, previously cached at client 106, are now being cached at client 104. Thus, cached content may be moved or copied from clients 102 and 106 to portion 146 at client 104.

At state 308, modules 124, 144 and 164 cache content in portions 126, 146 and 166 and respond to requests 32 as described in FIGURE 3. At state 310, modules 124, 144 and 164 may also collect statistical data associated with clients 102, 104 and 106. Statistical data may comprise processor and/or bandwidth usage, such as with respect to the time of day, time between crashes or reboots, web surfing habits and any other suitable statistical information associated with clients 102, 104 and 106. The statistical data may be stored by modules 124, 144 and 164 for use in negotiating distributions and may also be communicated to a remote site. The statistical data may be communicated to the remote site for storage and use for marketing and advertising purposes. For example, a corporation may use the statistical information to learn about the surfing habits of its employees.

Then, at step 312, cache module 168 begins to deactivate and broadcasts cache status message 170 indicating "cache off" to clients 102 and 104. Modules 124 and 144 receiving cache status message 170 indicating a cache shutdown respond by renegotiating the cache distribution between the remaining active modules 124 and 144. More specifically, modules 124 and 144 may redistribute cached content such that domains starting with A-M are cached at client 102, while domains starting with N-Z are cached at client 104. Location tables 128 and 148 are updated to indicate the new content distribution. Next, at step 314, the module 164 clears portion 166. Depending on the configuration of modules 164, clearing portion 166 may include communicating cached content to the remaining clients 102 and 104.

System 10 provides the capability to create a distributed caching system using standard personal computers and the cache module software. Previous caching systems have typically been limited to caching at a single computer, such as the caching performed by web browsers, or have concentrated caching efforts at a single firewall or proxy server. The increasing adoption of "always-on", high-bandwidth Internet connections, such as DSL modems, cable modems and LANs, allows for system 10 to group standard personal computers to form a cache community in order to decrease response time for content requests. In addition, since the cache modules use the existing storage capacity of the client computers, additional hardware does not need to be purchased for the cache community. Also, corporations may benefit from forming community caches over their corporate intranet LANs in order to decrease usage of expensive wide-area-network connections, such as T-1 lines, to the Internet.

The use of community caches also provides benefits to Internet Service Providers (ISPs) by decreasing the amount of data transferred between the ISP and the Internet. Since the community caching takes place entirely on the subscriber side of the ISP's infrastructure, an increased number of subscribers can be supported by a given amount of Internet connection bandwidth. For example, if 25% of the content requests from subscribers are handled by the community cache, an approximately 25% decrease in the use of the ISP's public network connection is realized.

FIGURE 6 is a block diagram illustrating a dynamic caching system 400 comprising one embodiment of system 10. System 400 comprises a cache community 402, a client 404, a cache server 406, Internet 16, and origin server 19. Dynamic distributed caching system 400 provides the capability for allowing members to dynamically join and leave cache community 402 while continuing to provide a distributed caching system.

Community 402 comprises one or more peers 413. A peer 413 comprises a computer operable to execute logic and/or an application. Peers 413 further comprise a master 410 and a member 412. Community 402, similar to community 15, represents a group of peers 413 which cooperate to form a distributed caching system. Each community 402 includes one or more peers 413. More specifically, each community 402 includes one master 410 and zero or more members 412. A dynamic caching application 478 at each peer 413 provides functionality to support the distributed caching system 10.

Master 410 comprises a processor 420 and computer readable storage 422. Master 410 may be adapted to execute any of the well known MS-DOS, PC-DOS, OS/2, UNIX, Linux, MAC-OS, mainframe, minicomputer, Windows operating systems or other operating systems. Processor 420 comprises any suitable general purpose or specialized electronic or other processing device, such as a central processing unit (CPU), operable to communicate with storage 422 and execute applications stored in storage 422. Processor 420 may comprise multiple processors. Storage 422 comprises any suitable combination of transient and/or persistent memory operable to store member list 426 and dynamic caching application 428, and to communicate with processor 420. For example, storage 422 may comprise any suitable combination of optical and/or magnetic storage, such as a floppy disk drive, a hard disk drive, a CD-ROM drive or a CD-RW drive. Storage 422 may also represent multiple computer readable storage devices in any combination.

Master 410 is operable to generate an allow message 424 and maintain a peer list 426. Allow message 424 comprises a data message sent to client 404 to inform client 404 that client 404 is being allowed to join community 402. Allow message 424 may also be sent to client 404 to inform client 404 that entry to community 402 is denied.

Peer list 426 comprises a list of peers 413 in community 402. For example, peer list 426 may compromise a list of the Internet protocol (IP) addresses associated with peers 413.

Dynamic caching application 428 comprises software and/or hardware operable to support caching of data and other content within community 402. Application 428 is described in more detail in FIGURE 7.

Member 412 comprises a processor 430, computer readable storage 432 and dynamic caching application 428. Processor 430 comprises any suitable general purpose or specialized electronic processing device, such as a CPU, operable to communicate with storage 432. Processor 430 may comprise multiple processors. Storage 432 comprises any suitable combination of transient and/or persistent memory operable to store peer list 426 and dynamic caching application 428, and to communicate with processor 430. For example, storage 432 may comprise any suitable combination of optical or magnetic computer readable storage mediums such as a floppy disk drive, a hard disk drive, a CD-ROM drive, an optical drive, or a CD-RW drive. Storage 432 may also represent multiple computer readable storage devices in any combination.

Community 402 is comprised of a master 410 and zero or more members 412. Members 412 and master 410 operate to support the caching of content within community 402. More specifically, both master 410 and members 402 provide for a distributed caching system within community 402. In addition to the functionality provided by members 412, master 410 is further responsible for providing administrative support to community 402. In general, a member 412 is capable of being a master 410 as necessary and a member 412 may be designated as a master 410 at any time. Dynamic cache application 428, which executes on both master 410 and members 412, is typically fully capable of providing the functionality associated with being either a master 410 or a member 412.

Client 404 comprises a processor 440, computer readable storage 442 and dynamic caching cache application 428. Client 404 comprises a computer executing dynamic cache application 428 which has not yet joined a community 402. Processor 440 comprises any suitable general purpose or specialized electronic processing device, such as a CPU, operable to communicate with storage 442. Processor 440 may comprise multiple processors. Storage 442 comprises any suitable combination of transient and/or persistent memory operable to store member list 426 and dynamic caching application 428, and to communicate with processor 440. Storage 442 may further comprise any suitable combination of optical, magnetic or other computer readable storage mediums such as a floppy disk drive, a hard disk drive, a CD-ROM drive, an optical drive or a CD-RW drive. Storage 442 may also represent multiple computer readable storage devices in any combination.

Dynamic cache application 428 is operable to generate a community request 450 and a joined request 452. Community request 450 comprises a data message generated by dynamic cache application 428. Community request 450 requests the network location of communities 402 which client 404 may attempt to join. Community request 450 is communicating to cache server 406.

Join request 452 comprises a data message indicating that client 404 wishes to join a particular community 402. Join request 452 is communicated to a community 402.

Origin server 19 is further operable to provide content 460. Content 460 comprises static and/or dynamically generated information. In one embodiment, content 460 comprises one or more web pages, such as those formatted and linked using the hypertext markup language (HTML). For example, content 460 may comprise audio data, video data, text data, animations, applications, applets and other suitable content retrievable using the hypertext transport protocol (HTTP).

Cache server 406 comprises a processor 470, computer readable storage 472, an administration module 474, a community list 476, an expiration module 478, and is operable to handle an expiration message 480. Cache server 406 comprises one or more servers operable to provide content expiration and community tracking capabilities to system 400. More specifically, cache server 460 maintains a list of communities 402 which a client 404 may attempt to join. Processor 470 comprises any suitable general purpose or specialized electronic processing device, such as a CPU, operable to communicate with storage 472. Processor 470 may comprise multiple processors. Storage 472 comprises any suitable combination of transient and/or persistent memory operable to store member list 426 and dynamic caching application 428, and to communicate with processor 470. Storage 472 may further comprise any suitable combination of optical, magnetic or other computer readable storage mediums such as a floppy disk drive, a hard disk drive, a CD-ROM drive, an optical drive, or a CD-RW drive. Storage 472 may also represent multiple computer readable storage devices in any combination.

Administration module 474 comprises software and/or hardware operable to generate and maintain community list 476. Administration module 474 is further operable to respond to community request 450 from client 404.

Community list 476 comprises a list of communities 402 known to cache server 406 that client 404 may attempt to join.

In one embodiment, cache server 406 tracks communities 402 which are close to cache server 406 in terms of network distance. Typically, on a computer network, geographic location is not necessarily a good indicator of response time between that two nodes on a network. The latency between two network nodes is often a better indication of performance. The latency represents the sum of the time the packet has spent waiting to be forwarded out of a particular network node plus the transit time over the link connecting one network node to another node. The latency between two network locations may be used to determine the network distance between the two network locations. Other suitable indications may also be used to determine network distance, such as the number of hops or nodes between two nodes. Administration module 474 may impose an upper and/or a lower limit on the network distance from cache server 406 of communities 402 that may be listed in community list 476.

Expiration module 478 comprises software and/or hardware operable to receive and communicate data expiration messages 480 associated with content 460. Expiration message 480 comprises an indication that content 460 at origin server 19 has changed and that cached copies of content 460 should be marked as stale or out-of-date. Expiration module 478 operates to communicate expiration message 480 to communities 402 listed in community list 476.

In operation, dynamic cache application 428 is initiated at client 404. Dynamic cache application 428 then generates a community request 450. Community request 450 is a request for a list of communities 402 that client 404 may attempt to join. Community request 450 is communicated over Internet 60 to cache server 406. In one embodiment, the cache server 406 contacted by dynamic cache application 428 has been previously determined, such as a particular URL. Alternatively, cache server 406 may be specified by a user associated with client 404. Cache server 406 then receives community request 450 and returns community list 476 to client 404.

Dynamic cache application 428 then examines community list 476 and selects one of the communities 402 listed in community list 476 to attempt to join. For example, dynamic cache application 428 may select a community 402 which has the lowest latency from client 404. Alternatively, dynamic cache application 428 may select the first community 402 in list 476 or may select a community 402 randomly from list 476. In general, dynamic cache application 428 may select a community 402 to attempt to join using any suitable technique.

Once dynamic cache application 428 has selected a particular community 402 to attempt to join, dynamic cache application 420 generates join request 452 and communicates join request 452 to community 402. More specifically, join request 452 is communicated from client 404 to master 410 in community 402 since master 410 is responsible for handling the addition and removal of members 412 from community 402.

Master 410 receives join request 452 and determines whether to allow client 404 to become a member 412 of community 402. Master 410 may use any suitable criteria to determine whether to allow client 404 to join community 402. For example, master 410 may consider whether the addition of client 404 would exceed the maximum number of members 412 that may be in community 402 and/or whether the round trip transit time for data between client 404 and members 412 of community 402 is within a threshold limit. The round trip transmission time may comprise the total length of time a data package from master 410 or members 412 takes to travel to client 404 and return back to the originating master 410 or member 412.

Master 410 may attempt to keep round trip transmission times within a certain threshold so as to avoid degrading the caching performance of community 402. If master 410 determines that client 404 is appropriate to be a member of community 402, then dynamic cache application 428 at master 410 generates allow message 424 and communicates allow message 424 to client 404. When allow message 424 indicates that client 404 may join community 402, then dynamic cache application 428 also communicates peer list 426 along with allow message 424 to client 404. If allow message 424 indicates that client 404 is allowed to join community 402, then client 404 has joined community 402.

Master 410 then communicates to members 412 that a new member 412 is joining community 402 by communicating an updated peer list 426 which includes client 404. Dynamic cache application 428 then reallocates the content 460 to be cached by particular members 412 and master 410 so that the newly added client 404 is responsible for some subset of content 460 cached in community 402.

If master 410 determines that client 404 should not be allowed to join community 402, then dynamic cache application 428 may generate allow message 424 with the indication that client 404 has been denied entry to community 402. Alternatively, dynamic cache application 428 may simply ignore join request 452 and allow client 404 to determine that it has rejected from community 402 because no response to join request 452 has been received.

If client 404 receives an allow message 424 indicating denial of entry to community 402 or if client 404 does not receive a response from master 410 within a predetermined period of time, then dynamic cache application 428 will determine that client 404 has been denied entry to community

402. Dynamic cache application 428 will then select a new community 402 from community list 476 and communicate a join request 452 to the newly selected community 402.

When cache server 406 determines that content 460 has changed at origin server 19 or is informed that content 460 has changed at origin server 19, cache server 406 generates expiration message 480. Expiration message 480 is then sent to each master 410 for each community 402 listed in community list 476. Master 410 in community 402 then receives expiration message 480 and communicates expiration message 480 to members 412 so that expired, cached content 460 is marked as stale or expired. In one embodiment, expiration message 480 comprises an ICSP message. When expired content 460 is cached in community 402 and a request for the expired content 460 is received, a new copy of content 460 is retrieved from origin server 19.

FIGURE 7 is a block diagram illustrating details of dynamic cache application 428. Dynamic cache application 428 comprises a cache portion 500, an expiration portion 502, and a dynamic affiliation portion 504.

Cache portion 500 comprises software and/or hardware operable to receive any cached content 460. Cache portion 500 is responsible for managing and organizing the actual cache content on the computer readable storage, such as storage 422, 432, or 442. Cache portion 500 is further operable to maintain an allocation list 510 when cache application 428 is acting as a master 410.

Allocation list 510 comprises an indication of which peer 413 caches which content 460. In one embodiment, allocation list 510 comprises location table 148 and the cache shares previously described. In general, any suitable technique may be used for allocating content 468 among peers 413. For example, the CARP protocol may be used to manage distribution of content within community 402.

Expiration portion 502 comprises software and/or hardware operable to receive expiration messages 480 and mark content stored by cache portion 500 as expired. Alternatively, expiration portion 502 may request cache portion 500 to itself mark content as expired. Expiration portion 502 is further operable to propagate expiration message 480 to members 412 in member list 426 when dynamic cache application 428 is operating as a master 410.

Dynamic affiliation portion 504 comprises software and/or hardware operable to support the dynamic joining and departure of members 412 and master 410 from community 402. Dynamic affiliation portion 504 is further operable to generate and respond to various messages and requests. In particular, dynamic affiliation portion 504 is responsible for supporting allow message 424, community request 450, join request 452, probe message 456 and add master request 454. Dynamic affiliation portion 504 is further operable to provide a peer list message 520, a member status request 522, a nominate master message 524, an update master request 526, a remove peer message 528, a hold election message 530, a location request 550, a location response 554, a probe message 456 and a remove master request 527.

Peer list message 520 comprises a data communication to members 412 of community 402 to update peer list 426 to add or remove peers 413 from community 402. For example, peer list message 520 may include an updated peer list 426 from master 410 for use by members 412. For another example, peer list message 520 may include an instruction to a member 412 to add or delete a particular peer 413 from peer list 426.

Member status request 522 comprises data communicated from dynamic affiliation module 504 at master 410 to a member 412 to determine the status of the member 412. More specifically, member status request 522 is used by master 410 to determine whether a member 412 has unexpectedly left community 402, such as by crashing. In one embodiment, member status request 522 may comprise a "ping" type request, such as one based on the Internet Control Message Protocol (ICMP) used with the TCP/IP protocol.

Nominate master message 524 comprises data indicating that a particular peer 413 should be designated as master 410. Update master request 526 comprises data indicating to administration module 474 that community list 476 should be updated to indicate that a particular peer 413 or client 404 has become the master 410 of community 402.

Remove peer message 528 comprises a data message which indicates from master 410 to members 412 that a particular member 412 is leaving community 402 and should be removed from peer list 426. Hold election message 530 comprises a data message which indicates that a member 412 has detected the possible departure of master 410 from community 402 in that an election should be held among the remaining members 412 to determine a new master 410.

Location request 550 comprises a data message which indicates a request for the peer 413 which would cache content 460 requested by request 32. Location response 554 comprises a data message which indicates an indication of a peer 413 in community 402 which is responsible for caching the requested content. Probe message 456 comprises a data message which indicates a broadcast and/or multicast request to determine the existence of one or more communities 402.

Remove master request 527 comprises a data message which indicates a request by a master 410 that the cache server 406 remove the community 402 associated with the master 410. More specifically, remove master request 527 is used by master 410 to indicate the shutdown of a cache community 402.

FIGURE 8 is a flow diagram illustrating a method for retrieving and caching content within community 402. Initially, a request 32 for requested content 548 is generated by browser 30 in response to, for example, user input. Request 32 is received at cache portion 500 of dynamic cache application 428 at member 412A. Cache portion 500 at member 412A then determines whether the requested content 548 is available in cache portion 500 at member 412A. In general, content is considered "available" at a particular cache portion 500 when the content is stored and unexpired at the particular cache portion 500. Content may be considered unavailable if the content is stored, but marked as expired, or when the content is not stored at the cache portion 500. If requested content 548 is available at cache portion 500 at member 412A, then cache portion 500 returns requested content 548 to browser 30.

If requested content 548 is unavailable at cache portion 500 at member 412A, then cache portion 500 at member 412A generates location request 550 which is communicated to cache portion 500 at master 410. Cache portion 500 at master 410 examines allocation list 510 to determine which peer 413 would cache the requested content 460.

Cache portion 500 at master 410 determines where the requested content 460 is located and generates a location response 554 which is communicated to member 412A.

Cache portion 500 at member 412A then receives location response 554 and forwards request 32 to the appropriate peer

413. For example, the appropriate peer 413 may be member 412B. Cache portion 500 at member 412B receives request 32 and determines whether the requested content 548 is available and current in cache portion 500 at member 412B. If the requested content 548 is available and current in cache portion 500, then the requested content 548 is returned by member 412B to cache portion 500 at member 412A. Cache portion 500 at member 412A then provides requested content 548 to browser 30 for display to a user associated with member 412A.

If requested content 548 is not available at cache portion 500 at member 412B, then member 412B forwards request 32 to origin server 19. Origin server 19 then returns requested content 548 to cache portion 500 at member 412B. Member 412B then caches requested content 548 at cache portion 500 and forwards requested content 548 to cache portion 500 at member 412A. Cache portion 500 at member 412A then communicates requested content 548 to browser 30 for display to the user.

FIGURE 9 is a flow chart illustrating a method for adding a client 404 to community 402. The method begins at step 600, where client 404 sends community request 450 to administration module 474. Next, at step 602, client 404 determines whether a response has been received from administration module 474. More specifically, dynamic cache application 428 determines whether a community list 476 has been received from cache server 406.

If a response is received from administration module 474, then the YES branch of decisional step 602 leads to step 604. At step 604, dynamic cache application 428 examines community list 476. Then, at step 606, dynamic cache application 428 attempts to find a community 402 which comprises a "best fit" community for client 404 to join. More specifically, dynamic cache application 428 may evaluate various factors associated with communities 402 to determine which community 402 or community list 476 client 404 should join. For example, dynamic cache application 428 may determine the round trip transmission time between client 404 and community 402. The round trip transmission time may be useful in determining the overall performance of a community 402 and the performance of a community 402 with respect to client 404. Proceeding to decisional step 608, dynamic cache application 428 determines whether any communities 402 meet the best fit criteria. If one or more communities 402 meet the best fit criteria, then the YES branch of decisional step 608 leads to step 610. At step 610, dynamic cache application 428 sends join request 452 to one of the communities 402 which matched the best fit criteria in step 608. The particular community selected by dynamic cache application 428 may comprise the community 402 which most closely met the best fit criteria. Alternatively, dynamic cache application 428 may use other methods for selecting the first community 402 to contact.

Then, at step 612, dynamic cache application 428 at client 404 determines whether master 410 has accepted client 404 into community 402. Dynamic cache application 428 at client 404 may determine that master 410 has denied client 404 admittance to community 402 by receiving an allow message 424 indicating that client 404 is not allowed to join community 402 or by receiving no response from master 410 within a predetermined time period. If client 404 receives an allow message 424 indicating that client 404 is allowed to join community 402, then the YES branch of decisional step 612 leads to step 614. At step 614, client 404 is added to community 402 by master 410 and member list 426 is updated by master 410.

If master 410 denies client 404 entrance to community 402, then the NO branch of decisional step 612 leads to step 616. At step 616, dynamic cache application 428 at client 404 selects the next community 402 to attempt to join. Also, at step 616, the criteria for the best fit for client 404 may optionally be modified by dynamic cache application 428 so as to change or increase the number of communities 402 that client 404 may join. The method then proceeds to step 608 where communities 402 and community list 476 are evaluated against the best fit criteria which may have changed at step 616.

Returning to decisional step 608, if no communities 402 in community list 476 match the best fit criteria for client 404, then the NO branch of decisional step 608 leads to step 618. At step 618, dynamic cache application 428 at client 404 nominates itself as a master 410. When client 404 nominates itself as a master 410, client 404 creates a new cache community 402. Then, at step 620, add master 454 is sent from client 404 to administration module 474 at cache server 406. Add master 454 indicates to administration module 474 that a new cache community 402 is being generated with client 404 as master 410 and that the new community 402 should be added to community list 476.

Returning to decisional step 602, if client 404 does not receive a response from administration module 474, then the NO branch of decisional step 602 leads to step 622. At step 622, probe message 456 is sent from client 404 to a known port to one or more network nodes. More specifically, dynamic cache application 428 may be configured to listen at a particular location, such as a transmission control protocol/Internet protocol (TCP/IP) port, for clients 404 which wish to join community 402. This capability provides, for example, the capability to add members to a community 402 in the absence of a cache server 406 or when cache server 406 cannot be contacted. Proceeding to decisional step 624, if client 402 receives a response to probe message 456 from a master 410 and a community 402, then the YES branch of decisional 624 leads to step 610 where client 404 will attempt to join the responding community 402. If no response is received to probe message 456, then the NO branch of decisional step 624 leads to step 618 and client 404 will attempt to form its own community 402.

FIGURE 10 is a flow chart illustrating a method for allowing a client 404 to join a cache community 402. The method begins at step 900, where a master 410 in a community 402 listens for join request 452 from clients 404. Next, at step 902, dynamic affiliation portion 504 at master 410 evaluates join request 452 to determine whether client 404 will be allowed to join community 402. Master 410 may use any suitable criteria in determining whether to allow client 404 to join community 402, such as whether community 402 has reached a maximum number of peers 413 and the roundtrip transmission time between client 404 and community 402. Next, at decisional step 904, master 410 decides whether client 404 is allowed to join community 402. If client 404 is not allowed to join community 402, then the NO branch of decisional step 904 leads to 906 where join request 452 is ignored by master 410. Alternatively, master 410 may communicate an allow message 424 to client 404 with the indication that client 404 has been denied entry to community 402.

If client 404 is allowed to join community 402, then the YES branch of decisional step 904 leads to step 908. At step 908, peer list 426 is updated to include client 404. Proceeding to step 910, allow message 424 is communicated to client 404 along with the updated peer list 426. Then, at step 912, allocation list table 510 is updated to include client 404 and community 402. More specifically, the distribution of content 460 to be cached within community 402 is adjusted to include the additional capacity provided by client 404. Then, at step 914, update peer list message 520 is sent to members 412 by master 410 to inform members 412 to update peer list 426 at each of the members 412 to include client 404. The method then returns to step 900 where master 410 listens for additional join requests 452.

FIGURE 11 is a flow chart illustrating a method for determining whether a member 412 has unexpectedly departed community 402. The method begins at step 1000, where master 410 sends member status request 522 to a selected member 412. The member 412 to send member status request 522 to may be determined using any suitable method. For example, master 410 may sequentially traverse peer list 426 and send member status request 522 to each member 412 listed in peer list 426. Next, at step 1002, master 410 determines whether a response has been received to member status request 522. If a response is received then the YES branch of decisional step 1002 leads to step 1004. At step 1004, master 410 resets the time-out for the member 412 from which the response is received. More specifically, master 410 may have a predetermined time interval after which a particular member 412 is contacted again and the countdown to the expiration of that interval is reset. For example, a member 412 may be contacted once every minute. If no response is received from member 412, then the

NO branch of decisional step 1002 leads to step 1006. At step 1006, master 410 removes the member 412 which did not respond from peer list 426. Proceeding to step 1008, master 410 sends update peer list message 512 with the indication that the non-responding member 412 should be removed from the peer list 426 of other members 412. Then, at step 1010, master 410 updates allocation table 510 to account for the loss of the member 412.

FIGURE 12 is a flow chart illustrating a method for removing a member 412 from community 402. The method begins at step 1100, where master 410 listens for members which wish to leave community 402. Next, at step 1102, master 410 receives remove peer message 528. Remove peer message 528 indicates that a member 412 wishes to leave community 402. Remove peer message 528 is generated by the member 412 which desires to depart community 402 and is communicated to master 410. Then, at step 1104, master 410 removes the departing member 412 from peer list 426. Proceeding to step 1106, master 410 updates allocation table 510 to account for the departing member 412. Then, at step 1108, master 410 sends update peer list message 520 to the remaining members 412 in community 402 to update their peer lists 426 to remove the departing member 412. The departing member 412 is then removed from community 402 and the method returns to step 1100 where master 410 listens for further departing members.

FIGURE 13 is a flow chart illustrating a method for gracefully removing master 410 from community 402. The method begins at step 1200 where master 410 determines that it is going to leave community 402. Then, at decisional step 1201, master 410 determines if it is the only peer 413 in community 402. If master 410 is not the only peer 413 in community 402, then NO branch of decisional step 1201 leads to step 1202.

Next, at step 1202, master 410 selects a new master from members 412 in peer list 426. Master 410 may use any suitable method for determining who the new master 410 is to be. For example, master 410 may select the new master 410 based on the amount of bandwidth available at a member 412, the processing capability of member 412, or other suitable criteria. Then, at step 1204, master 410 removes itself from peer list 426. Proceeding to step 1206, master 410 sends nominate master message 524 to the selected member 412 whose become the new master 410 along with the updated peer list 426 which no longer includes the old master 410. Then, at decisional step 1208, the old master determines whether a response has been received from member 412 which is to become the new master 410. If no response is received from the member 412 which is to become the new master, then the NO branch of decisional step 1208 returns to step 1202 where the current master 410 will select a new master 410 from the other members 412 of community 402. If a response is received from member 412 which is to become the new master 410 then the YES branch of decisional step 1208 leads to step 1210. At step 1210, the existing master 410 shuts itself down. Next, at step 1212, the member 412 which was selected as the new master 410 becomes the master 410 and sends update peer list message 520 to the remaining members 412 of community 402. The update peer list message 520 indicates that the remaining members 412 of community 402 should update their peer lists 426 to indicate that the old master 410 is no longer in community 402 and that the new master 410 is the present master 410. Proceeding to step 1214, the new master 410 sends update master request 526 to cache server 406. Update master request 526 indicates to cache server 406 and administration module

474 that community list 476 should be updated to indicate that the new master 410 is the master for community 402.

Returning to decisional step 1201, if master 410 determines that it is the only peer 413 in community 402, then YES branch of decisional step 1201 leads to step 1220. At step 1220, master 410 sends remove master request 527 to cache server 406. Cache server 406 then removes community 402 associated with master 410. Master 410 then shuts down. The method then ends.

FIGURE 14 is a flow chart illustrating a method for allowing a master 410 to unexpectedly depart community 402. The method begins at step 1300, where a member 412 determines how much time has elapsed since the last member status request 522 was received by member 412. Then, at decisional step 1302, member 412 determines whether the time since the last member status request 522 was received at member 412 exceeds a predetermined threshold. For example, the predetermined threshold may be one minute since the last member status request 522 was received. If the time since the last member status request 522 was received does not exceed the predetermined threshold, then the NO branch of decisional step 1302 returns to step 1300 where member 412 checks the elapsed time again.

If the elapsed time exceeds the predetermined threshold, then the YES branch of decisional step 1302 leads to step 1304. At step 1304, member 412 generates hold election message 530 and communicates hold election message to members 412 in peer list 426. Member 412 generates hold election message 530 to inform other members 412 in community 402 that master 410 may have shut down and that an election for a new master 410 should be held. Proceeding to step 1306, member 412 which generated hold election message 530 may receive one or more responses from members 412 and master 410 in community 402. Then, at step 1308, member 412 which generated hold election message 530 determines whether any response has been received from master 410. If master 410 has responded to hold election message 530, then the YES branch of decisional step 1308 leads to step 1310 where member 412 aborts the attempt to elect a new master 410. Master 410 may respond because master 410 is still operating and was only temporarily unable to generate a member status request 522 for member 412 which generated hold election message 530. If master 410 was only temporarily unavailable, then it is unnecessary for a new master 410 to be elected for community 402.

If no master 410 responds in step 1308, then the NO branch decisional step 1308 leads to step 1312. At step 1312, member 412 generates a new peer list 426 based on responses from members 412 received in step 1306. A new peer list 426 may be generated so as to determine the remaining members of community 402. For example, a communications link connecting master 410 and one or more members 412 to other members 412 in community 402 may unexpectedly shut down. The unexpected loss of a communications link may force community 402 to divide into two separate communities 402 and result in the election of a new master 410 for one of the two new communities 402. Then, at step 1314, a new master 410 is selected from the remaining peers in 413 in the newly generated peer list 426 from step 1312. The new master 410 may be selected in any suitable manner, such as by selecting member 412 which detected that master 410 had ceased to respond. Next, at step 1316, the nominate master message 524 is communicated to member 412 which has been selected to be the new master 410 in step 1314 along with the newly generated peer list 426. Nominate master message 524 is generated by member 412 which originally detected that master 410 had ceased to respond in step 1302. Then, at decisional step 318, member 412 which originally detected that master 410 had ceased to respond determines whether the responses received from the newly nominated master. If no response is received from the newly nominated master, then the NO branch decisional step 1318 returns to step 1314 where another new master is selected from the remaining members 412. If a response is received then the YES branch decisional step 1318 leads to step 1320. At step 1320, the newly nominated master 410 sends update master request 526 to cache server 406 to update community list 476. The method then ends.

In one embodiment, dynamic affiliation portion 504 supports the addition and removal of members 412 and master 410 using the Dynamic Reef Protocol (DRP). In the DRP, community request 450 comprises a CRMSG_WAKEUP data message, add master request 454 comprises a CRMSG_ADDMASTER data message, update master request 526 comprises a CRMSG_REPLACEMASTER data message, remove master request 527 comprises a CRMSG_REMOVEMASTER data message, join request 452 comprises a CRMSG_REQUESTTOJOIN data message, nominate master message 524 comprises a CRMSG_NOMINATEMASTER data message, update peer list message 520 comprises a CRMSG_UPDATEPEERLIST data message, remove peer message 528 comprises a CRMSG_REMOVEPEER data message, hold election message 530 comprises a CRMSG_HOLDELECTION data message, and member status request 522 comprises a CRMSG_PEERPING data message. In addition, probe message 456 may use the CRMSG_WAKEUP data message. The DRP data messages may be organized using any suitable format. For example, a bit pattern may be used in a type field of a data message to indicate the type of data message being sent. For another example, the DRP may be built on top of or as part of the ICSP protocol.

By providing the capability to cache data in a cache community, system 400 provides the ability to cache large amounts of content to the user. Traditionally, caching of Internet content has been done only at an individual user's computer. This provides the benefits of caching data only after the user has retrieved the data from a remote web server. By providing the capability for computers to join a cache community, content stored in the cache community can be quickly provided to the members without having to retrieve the content from an origin server over a possibly slow Internet. As the use of high-speed connections to the Internet, such as Digital Subscriber Line (DSL) and cable modem technologies, increases, the increased responsiveness of a cache community using high-speed connections may be significant in comparison to the responsiveness of the Internet in general.

In addition, traditional distributed caching solutions have used static, predefined groups cache servers. In contrast, system 400 provides the capability to dynamically add and remove members from a distributed caching community. By allowing dynamic addition and subtraction of members from a distributed caching community, a distributed caching system is supported which does not require a centralized administrator. Also, system 400 supports creating a distributed caching system with heterogeneous computers that may be added and removed from the caching community in a dynamic manner.

Other changes, substitutions and alterations are also possible without departing from the spirit and scope of the present invention, as defined by the following claims.

WHAT IS CLAIMED IS:

1. A method for dynamic distributed data caching comprising:

providing a cache community comprising at least one peer, each peer having an associated first content portion indicating content to be cached by the respective peer;

allowing a client to join the cache community;

updating a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community; and

associating a respective second content portion with each peer based on the addition of the client, the second content portion being distinct from the first content portion.

2. The method for dynamic distributed data caching according to Claim 1 and further comprising:

receiving a join request from the client; and

determining whether to allow the client to join the cache community.

3. The method dynamic distributed data caching according to Claim 2, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

4. The method for dynamic distributed data caching according to Claim 1, wherein allowing the client to join the cache community comprises:

generating an allow message;

associating the peer list with the allow message; and

communicating the allow message to the client.

5. The method for dynamic distributed data caching according to Claim 4, wherein allowing the client to joint the cache community comprises:

generating an allow message comprising the peer list updated to include the client;

communicating the allow message to the client; and

communicating the allow message to at least one member associated with the cache community.

6. The method for dynamic distributed data caching according to Claim 4, wherein the allow message comprises a CRMSG_UPDATEPEERLIST data message.

7. The method for dynamic distributed caching according to Claim 4, wherein the peer list associated with the allow message comprises the updated peer list which includes the client.

8. The method for dynamic distributed data caching according to Claim 1, wherein the peer comprises a computer.

9. The method for dynamic distributed data caching according to Claim 1, wherein a one of the peers comprises a member.

10. The method for dynamic distributed data caching according to Claim 1, wherein a one of the peers comprises a master.

11. The method for dynamic distributed data caching according to Claim 1, wherein associating a respective second content portion comprises:

allocating respective second content portions to peers in the peer list; and

updating an allocation table to indicate the second content portion associated with the peers.

12. The method for dynamic distributed data caching according to Claim 11, wherein the second content portions are distinct.

13. The method for dynamic distributed data caching according to Claim 11, wherein at least two of the second content portions overlap.

14. The method for dynamic distributed data caching according to Claim 11, wherein the first and second content portions respectively comprise a plurality of internet protocol domain names.

15. The method for dynamic distributed data caching according to Claim 1 and further comprising removing the association between the first content portions and the peers.

16. A system for dynamic distributed data caching comprising:

logic encoded on storage and operable to:

provide a cache community comprising at least one peer, each peer having an associated first content portion indicating content to be cached by the respective peer;

allow a client to join the cache community;

update a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community; and

associate a respective second content portion with each peer based on the addition of the client, the second content portion being distinct from the first content portion.

17. The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to:

receive a join request from the client; and

determine whether to allow the client to join the cache community.

18. The system for dynamic distributed data caching according to Claim 17, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

19. The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to:

generate an allow message;

associate the peer list with the allow message;

communicate the allow message to the client.

20. The system for dynamic distributed data caching according to Claim 19, wherein the logic is further operable to:

generate an allow message comprising the peer list updated to include the client;

communicate the allow message to the client; and

communicate the allow message to at least one member associated with the cache community.

21. The system for dynamic distributed data caching according to Claim 19, wherein the allow message comprises a CRMSG_UPDATEPEERLIST data message.

22. The system for dynamic distributed caching according to Claim 19, wherein the peer list associated with the allow message comprises the updated peer list which includes the client.

23. The system for dynamic distributed data caching according to Claim 16, wherein the peer comprises a computer.

24. The system for dynamic distributed data caching according to Claim 16, wherein a one of the peers comprises a member.

25. The system for dynamic distributed data caching according to Claim 16, wherein a one of the peers comprises a master.

26. The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to:

allocate respective second content portions to peers in the peer list; and

update an allocation table to indicate the second content portion associated with the peers.

27. The system for dynamic distributed data caching according to Claim 26, wherein the second content portions are distinct.

28. The system for dynamic distributed data caching according to Claim 26, wherein at least two of the second content portions overlap.

29. The system for dynamic distributed data caching according to Claim 26, wherein the first and second content portions respectively comprise a plurality of internet protocol domain names.

29. The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to remove the association between the first content portions and the peers.

31. A method for dynamic distributed data caching comprising:

determining that a first master associated with a cache community is non-operational;

electing a second master to replace the first master in the cache community; and

allocating at least one content portion based on the loss of the first master.

32. The method for dynamic distributed data caching according to Claim 31, wherein determining that the first master is non-operational comprises:

determining the amount of time since a member status request message has been received;

determining whether the amount of time exceeds a threshold.

33. The method for dynamic distributed data caching according to Claim 32, wherein the member status request message comprises CRMSG_PEERPING data message.

34. The method for dynamic distributed data caching according to Claim 32, wherein the threshold comprises a predetermined time.

35. The method for dynamic distributed data caching according to Claim 31, wherein electing the second master comprises:

communicating a hold election message to peers associated with the cache community;

receiving at least one response to the hold election message;

selecting a one of the peers to be the second master;

communicating an update master message to a cache server.

36. The method for dynamic distributed data caching according to Claim 35, wherein the hold election message comprises a CRMSG_HOLDELECTION data message.

37. The method for dynamic distributed data caching according to Claim 35, wherein electing the second master further comprises aborting the election if a response is received from the first master.

38. The method for dynamic distributed data caching according to Claim 35, wherein the update master message comprises a CRMSG_REPLACEMASTER data message.

39. The method for dynamic distributed data caching according to Claim 31, wherein allocating at least one content portion comprises:

allocating respective content portions to peers in the cache community;

updating an allocation table at the second master to indicate the respective content portions associated with the respective peers.

40. The method for dynamic distributed caching according to Claim 39, wherein the content portions are distinct.

41. A system for dynamic distributed data caching comprising:

logic encoded on storage and operable to:

determine that a first master associated with a cache community is non-operational;

elect a second master to replace the first master in the cache community; and

allocate at least one content portion based on the loss of the first master.

42. The system for dynamic distributed data caching according to Claim 41, wherein the logic is further operable to:

determining the amount of time since a member status request message has been received; and

determining whether the amount of time exceeds a threshold.

43. The system for dynamic distributed data caching according to Claim 42, wherein the member status request message comprises CRMSG_PEERPING data message.

44. The system for dynamic distributed data caching according to Claim 45, wherein the threshold comprises a predetermined time.

45. The system for dynamic distributed data caching according to Claim 41, wherein the logic is further operable to:

communicate a hold election message to peers associated with the cache community;

receive at least one response to the hold election message;

select a one of the peers to be the second master; and

communicate an update master message to a cache server.

46. The system for dynamic distributed data caching according to Claim 45, wherein the hold election message comprises a CRMSG_HOLDELECTION data message.

47. The system for dynamic distributed data caching according to Claim 45, wherein the logic is further operable to abort the election if a response is received from the first master.

48. The system for dynamic distributed data caching according to Claim 45, wherein the update master message comprises a CRMSG_REPLACEMASTER data message.

49. The system for dynamic distributed data caching according to Claim 41, wherein the logic is further operable to:

allocate respective content portions to peers in the cache community; and

update an allocation table at the second master to indicate the respective content portions associated with the respective peers.

50. The system for dynamic distributed caching according to Claim 49, wherein the content portions are distinct.

51. A method for dynamic distributed caching comprising:

requesting a list of cache communities from a cache server;

determining whether at least one existing cache community exists;

attempting to join a one of the existing cache communities when the existing cache communities are found; and

generating a new cache community when no existing cache communities are found.

52. The method for dynamic distributed caching according to Claim 51, wherein requesting the list comprises:

generating a community request; and

communicating the community request to the cache server.

53. The method for dynamic distributed caching according to Claim 52 and further comprising determining whether a response has been received from the cache server within a threshold time.

54. The method for dynamic distributed caching according to Claim 51, wherein determining whether at least one existing cache community exists comprises:

generating a probe message; and

broadcasting the probe message.

55. The method for dynamic distributed data caching according to Claim 52, wherein the probe message comprises a CRMSG_WAKEUP data message.

56. The method for dynamic distributed data caching according to Claim 54, wherein the broadcast of the probe message is performed on a internet protocol broadcast address.

57. The method for dynamic distributed data caching according to Claim 54 and further comprising determining whether a response has been received from at least one existing cache community within a threshold time.

58. The method for dynamic distributed data caching according to Claim 54, wherein broadcasting the probe message comprises multicasting the probe message on an internet protocol multicast address.

59. The method for dynamic distributed data caching according to Claim 51, wherein attempting to join the one of the existing cache communities comprises:

generating a join request;

communicating the join request to the one of the existing cache communities.

60. The method for dynamic distributed data caching according to Claim 51, wherein generating the new cache community comprises:

generating an add master request; and

communicating the add master request to the cache server.

61. The method for dynamic distributed data caching according to Claim 60, wherein the add master request comprises a CRMSG_ADDMASTER data message.

62. The method for dynamic distributed data caching according to Claim 60 and further comprising listening for a join request.

63. A system for dynamic distributed caching comprising:

logic encoded on storage and operable to:

request a list of cache communities from a cache server;

determine whether at least one existing cache community exists;

attempt to join a one of the existing cache communities when the existing cache communities are found; and

generate a new cache community when no existing cache communities are found.

64. The system for dynamic distributed caching according to Claim 63, wherein the logic is further operable to:

generate a community request; and

communicate the community request to the cache server.

65. The system for dynamic distributed caching according to Claim 64, wherein the logic is further operable to determine whether a response has been received from the cache server within a threshold time.

66. The system for dynamic distributed caching according to Claim 63, wherein the logic is further operable to:

generate a probe message; and

broadcast the probe message.

67. The system for dynamic distributed data caching according to Claim 66, wherein the probe message comprises a CRMSG_WAKEUP data message.

68. The system for dynamic distributed data caching according to Claim 66, wherein the broadcast of the probe message is performed on an Internet protocol broadcast address.

69. The system for dynamic distributed data caching according to Claim 66, wherein the logic is further operable to determine whether a response has been received from at least one existing cache community within a threshold time.

70. The system for dynamic distributed data caching according to Claim 66, wherein the logic is further operable to multicast the probe message on an internet protocol multicast address.

71. The system for dynamic distributed data caching according to Claim 63, wherein the logic is further operable to:

generate a join request; and

communicate the join request to the one of the existing cache communities.

72. The system for dynamic distributed data caching according to Claim 63, wherein the logic is further operable to:

generating an add master request; and

communicating the add master request to the cache server.

73. The system for dynamic distributed data caching according to Claim 62, wherein the add master request comprises a CRMSG_ADDMASTER data message.

74. The system for dynamic distributed data caching according to Claim 72, wherein the logic is further operable to listen for a join request.

75. A method for dynamic distributed data caching comprising:

generating a content request for requested content at a first peer in a cache community;

determining a second peer associated with the requested content, the second peer being associated with the cache community; and

retrieving, by the first peer, the requested content from the second peer.

76. The method for dynamic distributed data caching according to Claim 75 and further comprising:

retrieving, by the second peer, the requested content from an origin server when the requested content is unavailable at the second peer; and

storing the requested content at the second peer.

77. The method for dynamic distributed data caching according to Claim 76, wherein the requested content is unavailable when the requested content has been expired.

78. The method for dynamic distributed data caching according to Claim 76, wherein the requested content is unavailable when the requested content is not stored at the second peer.

79. The method for dynamic distributed data caching according to Claim 75, wherein determining the location comprises:

generating, by a cache portion associated with the first peer, a location request;

communicating the location request to a master associated with the cache community; and

receiving a location response from the master, the location response indicating the second peer.

80. The method for dynamic distributed data caching according to Claim 79, wherein the cache portion comprises a software application.

81. The method for dynamic distributed data caching according to Claim 79 and further comprising forwarding the content request to the second peer.

82. The method for dynamic distributed data caching according to Claim 75, wherein the content request comprises a hypertext transport protocol request.

83. The method for dynamic distributed data caching according to Claim 75, wherein the content comprises a web page.

84. The method for dynamic distributed data caching according to Claim 75, wherein the first and second peers respectively comprises a member of the cache community.

85. A system for dynamic distributed data caching comprising:

logic encoded on storage and operable to:

generate a content request for requested content at a first peer in a cache community;

determine a second peer associated with the requested content, the second peer being associated with the cache community; and

retrieve, by the first peer, the requested content from the second peer.

86. The system for dynamic distributed data caching according to Claim 85, wherein the logic is operable to:

retrieve, by the second peer, the requested content from an origin server when the requested content is unavailable at the second peer; and

store the requested content at the second peer.

87. The system for dynamic distributed data caching according to Claim 86, wherein the requested content is unavailable when the requested content has been expired.

88. The system for dynamic distributed data caching according to Claim 86, wherein the requested content is unavailable when the requested content is not stored at the second peer.

89. The system for dynamic distributed data caching according to Claim 85, wherein the logic is operable to:

generate, by a cache portion associated with the first peer, a location request;

communicate the location request to a master associated with the cache community; and

receive a location response from the master, the location response indicating the second peer.

90. The system for dynamic distributed data caching according to Claim 89, wherein the cache portion comprises a software application.

91. The system for dynamic distributed data caching according to Claim 89, wherein the logic is operable to forward the content request to the second peer.

92. The system for dynamic distributed data caching according to Claim 85, wherein the content request comprises a hypertext transport protocol request.

93. The system for dynamic distributed data caching according to Claim 85, wherein the content comprises a web page.

94. The system for dynamic distributed data caching according to Claim 85, wherein the first and second peers respectively comprise a member of the cache community.

95. A method for dynamic distributed data caching comprising:

communicating a community request from a dynamic cache module to an administration module;

receiving a community list from the administration module in response to the community request, the community list including a list of communities;

generating a join request to attempt to join a one of the communities in the community list;

receiving an allow message associated with the one of the communities;

receiving a peer list associated with the one of the communities;

receiving a content request; and

storing content associated with the content request.

96. The method for dynamic distributed data caching according to Claim 95, wherein the community request comprises a CRMSG_WAKEUP data message.

97. The method for dynamic distributed data caching according to Claim 95, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

98. A system for dynamic distributed data caching comprising:

logic encoded on storage and operable to:

communicate a community request from a dynamic cache module to an administration module;

receive a community list from the administration module in response to the community request, the community list including a list of communities;

generate a join request to attempt to join a one of the communities in the community list;

receive an allow message associated with the one of the communities;

receive a peer list associated with the one of the communities;

receive a content request; and

store content associated with the content request.

99. The system for dynamic distributed data caching according to Claim 98, wherein the community request comprises a CRMSG_WAKEUP data message.

100. The system for dynamic distributed data caching according to Claim 98, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

101. A system for dynamic distributed data caching comprising:

means for providing a cache community comprising at least one peer, each peer having an associated first content portion indicating content to be cached by the respective peer;

means for allowing a client to join the cache community;

means for updating a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community; and

means for associating a respective second content portion with each peer based on the addition of the client, the second content portion being distinct from the first content portion.

102. A system for dynamic distributed data caching comprising:

means for determining that a first master associated with a cache community is non-operational;

means for electing a second master to replace the first master in the cache community; and

means for allocating at least one content portion based on the loss of the first master.

103. A system for dynamic distributed caching comprising:

means for requesting a list of cache communities from a cache server;

means for determining whether at least one existing cache community exists;

means for attempting to join a one of the existing cache communities when the existing cache communities are found; and

means for generating a new cache community when no existing cache communities are found.

104. A system for dynamic distributed data caching comprising:

means for generating a content request for requested content at a first peer in a cache community;

means for determining a second peer associated with the requested content, the second peer being associated with the cache community; and

means for retrieving, by the first peer, the requested content from the second peer.

105. A system for dynamic distributed data caching comprising:

means for communicating a community request from a dynamic cache module to an administration module;

means for receiving a community list from the administration module in response to the community request, the community list including a list of communities;

means for generating a join request to attempt to join a one of the communities in the community list;

means for receiving an allow message associated with the one of the communities;

means for receiving a peer list associated with the one of the communities;

means for receiving a content request; and

means for storing content associated with the content request.

METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING

ABSTRACT OF THE DISCLOSURE

A method and system for dynamic distributed data caching is presented. The method includes providing a cache community (402) comprising at least one peer (413). Each peer has an associated first content portion (511) indicating content to be cached by the respective peer. A client (404) may be allowed to join the cache community. A peer list (426) associated with the cache community is updated to include the client. The peer list indicates the peers in the cache community. A respective second content portion (511) is associated with each peer based on the addition of the client.

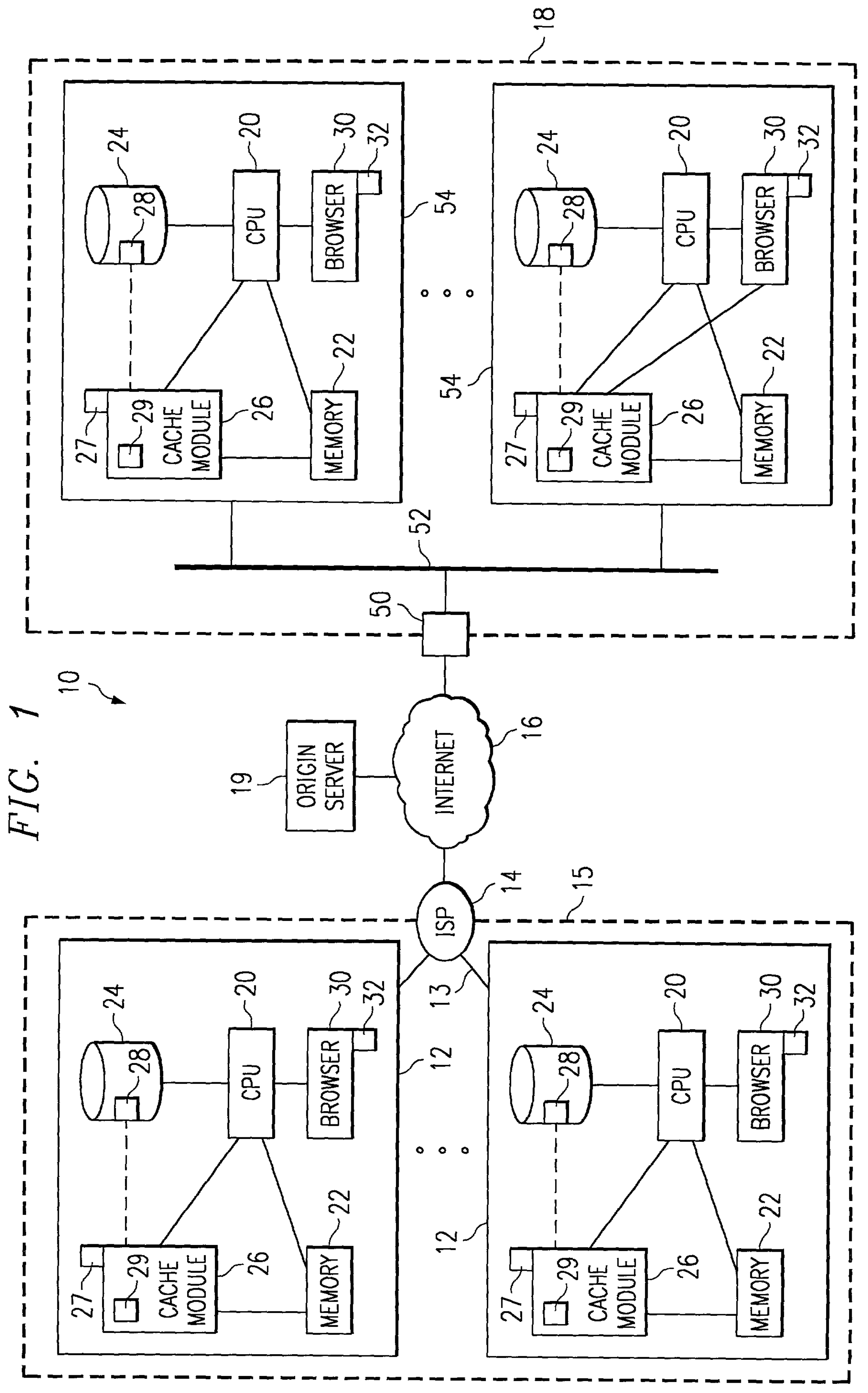


FIG. 1

10
18
24
28
20
30
32
CPU
BROWSER
54
27
29
CACHE MODULE
26
MEMORY
22
52
50
INTERNET
16
ORIGIN SERVER
19
ISP
14
15
13
12

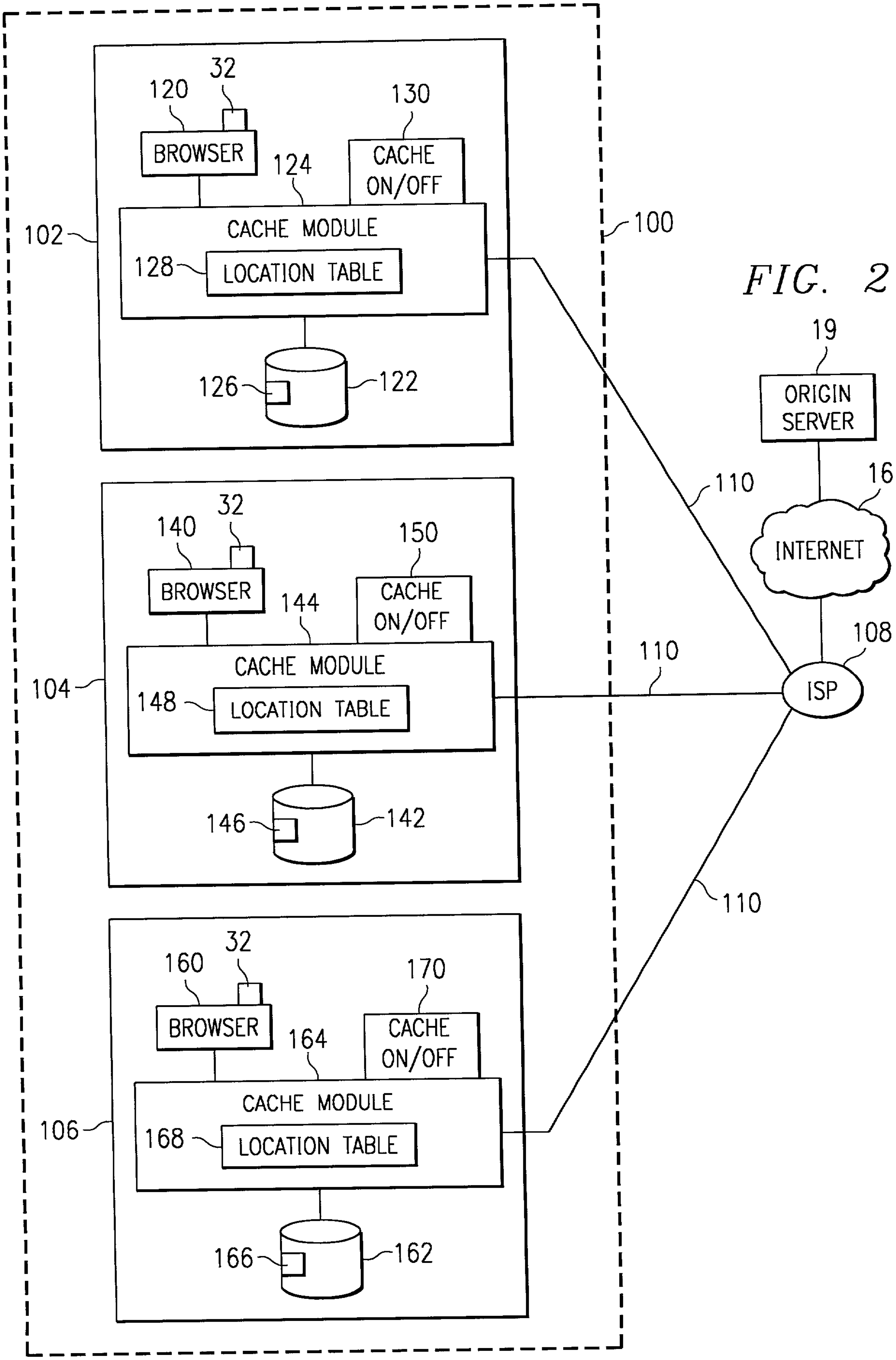

100
102
120
32
BROWSER
130
CACHE ON/OFF
124
CACHE MODULE
128
LOCATION TABLE
126
122
104
140
32
BROWSER
150
CACHE ON/OFF
144
CACHE MODULE
148
LOCATION TABLE
146
142
106
160
32
BROWSER
170
CACHE ON/OFF
164
CACHE MODULE
168
LOCATION TABLE
166
162
110
110
110
19
ORIGIN SERVER
16
INTERNET
108
ISP

FIG. 2

FIG. 3

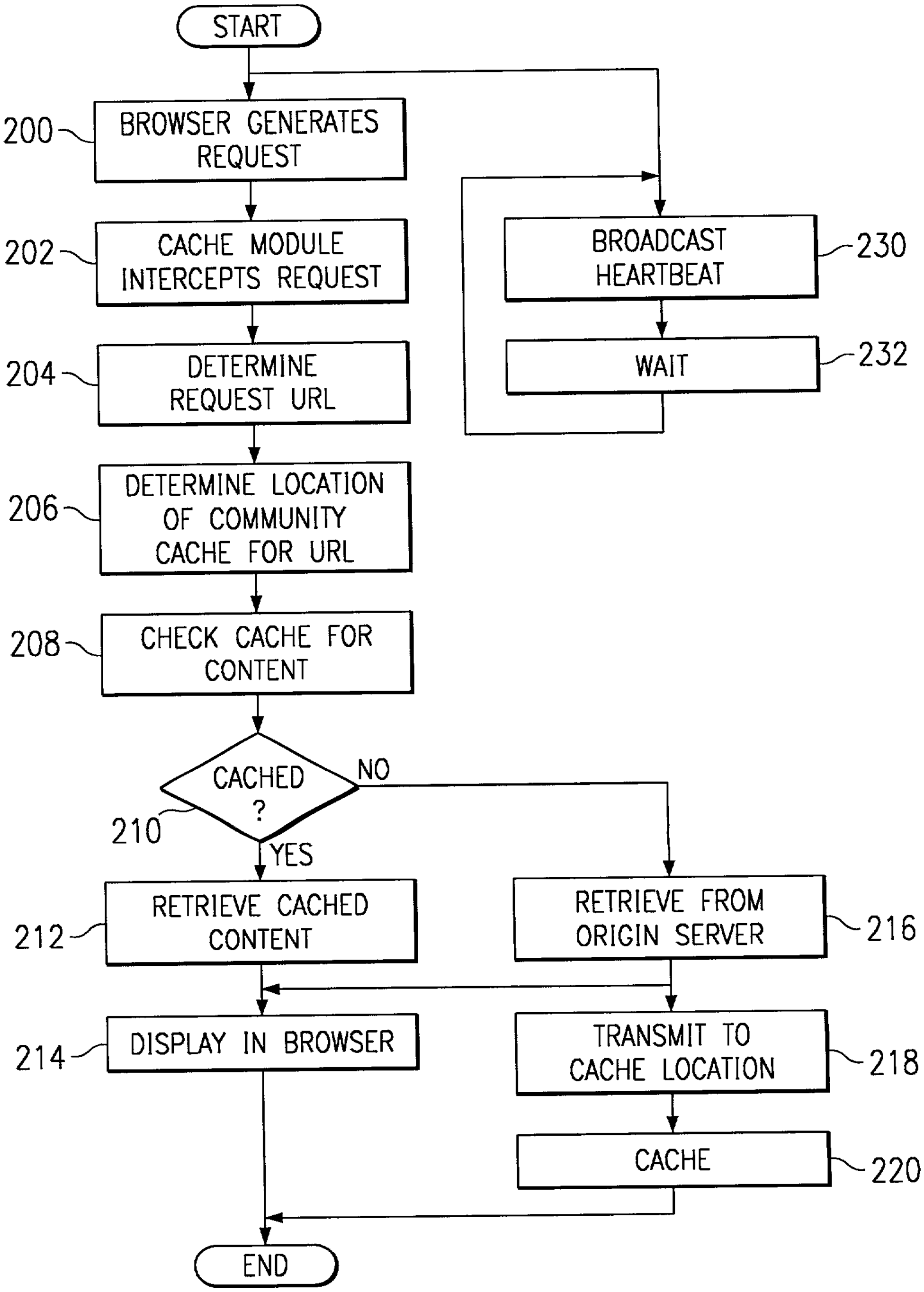

START
200
BROWSER GENERATES REQUEST
202
CACHE MODULE INTERCEPTS REQUEST
204
DETERMINE REQUEST URL
206
DETERMINE LOCATION OF COMMUNITY CACHE FOR URL
208
CHECK CACHE FOR CONTENT
CACHED ?
210
NO
YES
212
RETRIEVE CACHED CONTENT
214
DISPLAY IN BROWSER
END
BROADCAST HEARTBEAT
230
WAIT
232
RETRIEVE FROM ORIGIN SERVER
216
TRANSMIT TO CACHE LOCATION
218
CACHE
220

FIG. 4

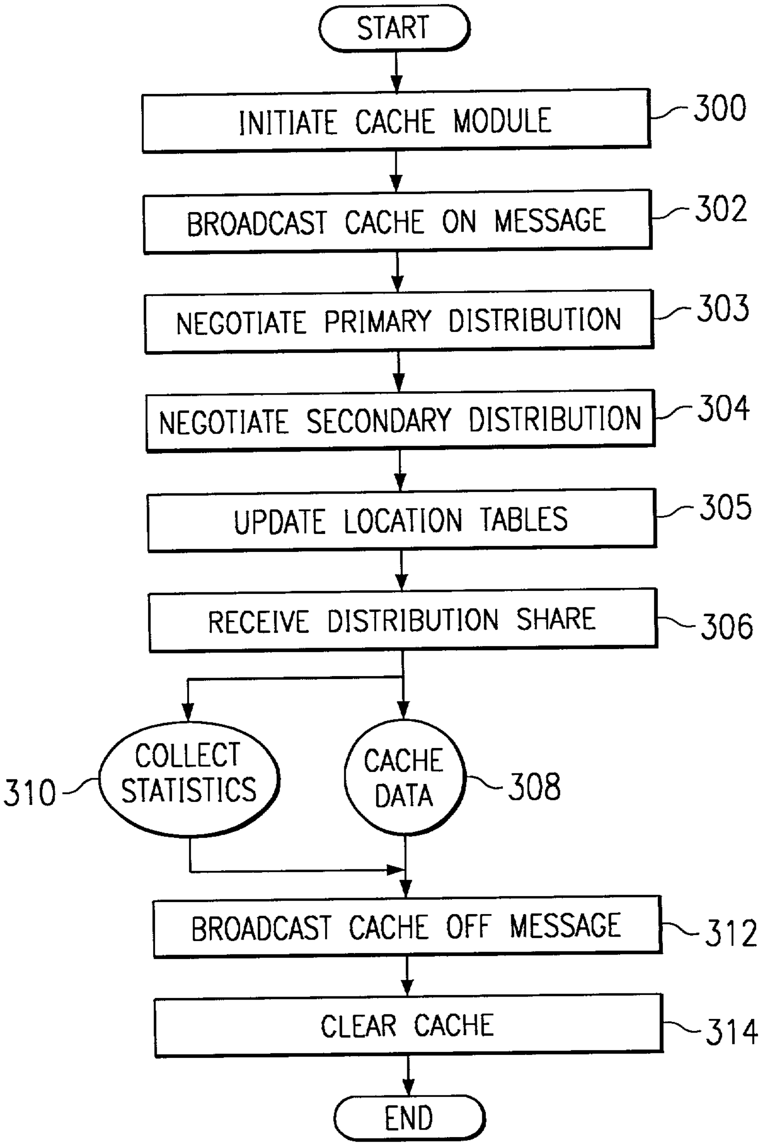

START
INITIATE CACHE MODULE
300
BROADCAST CACHE ON MESSAGE
302
NEGOTIATE PRIMARY DISTRIBUTION
303
NEGOTIATE SECONDARY DISTRIBUTION
304
UPDATE LOCATION TABLES
305
RECEIVE DISTRIBUTION SHARE
306
COLLECT STATISTICS
310
CACHE DATA
308
BROADCAST CACHE OFF MESSAGE
312
CLEAR CACHE
314
END

FIG. 5

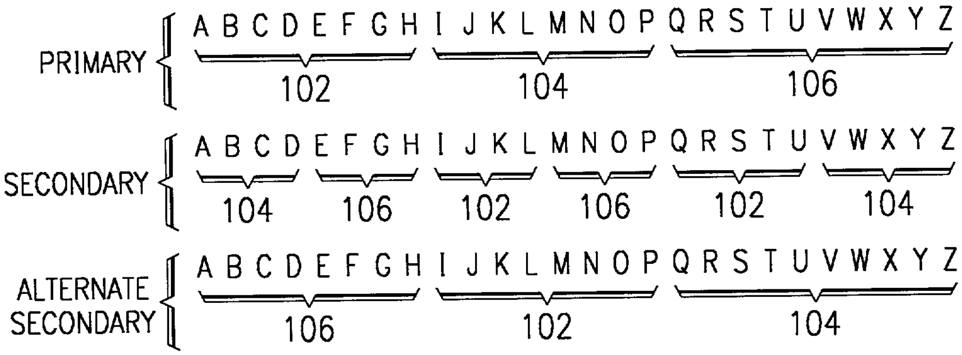

PRIMARY
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
102
104
106
SECONDARY
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
104
106
102
106
102
104
ALTERNATE SECONDARY
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
106
102
104

FIG. 6

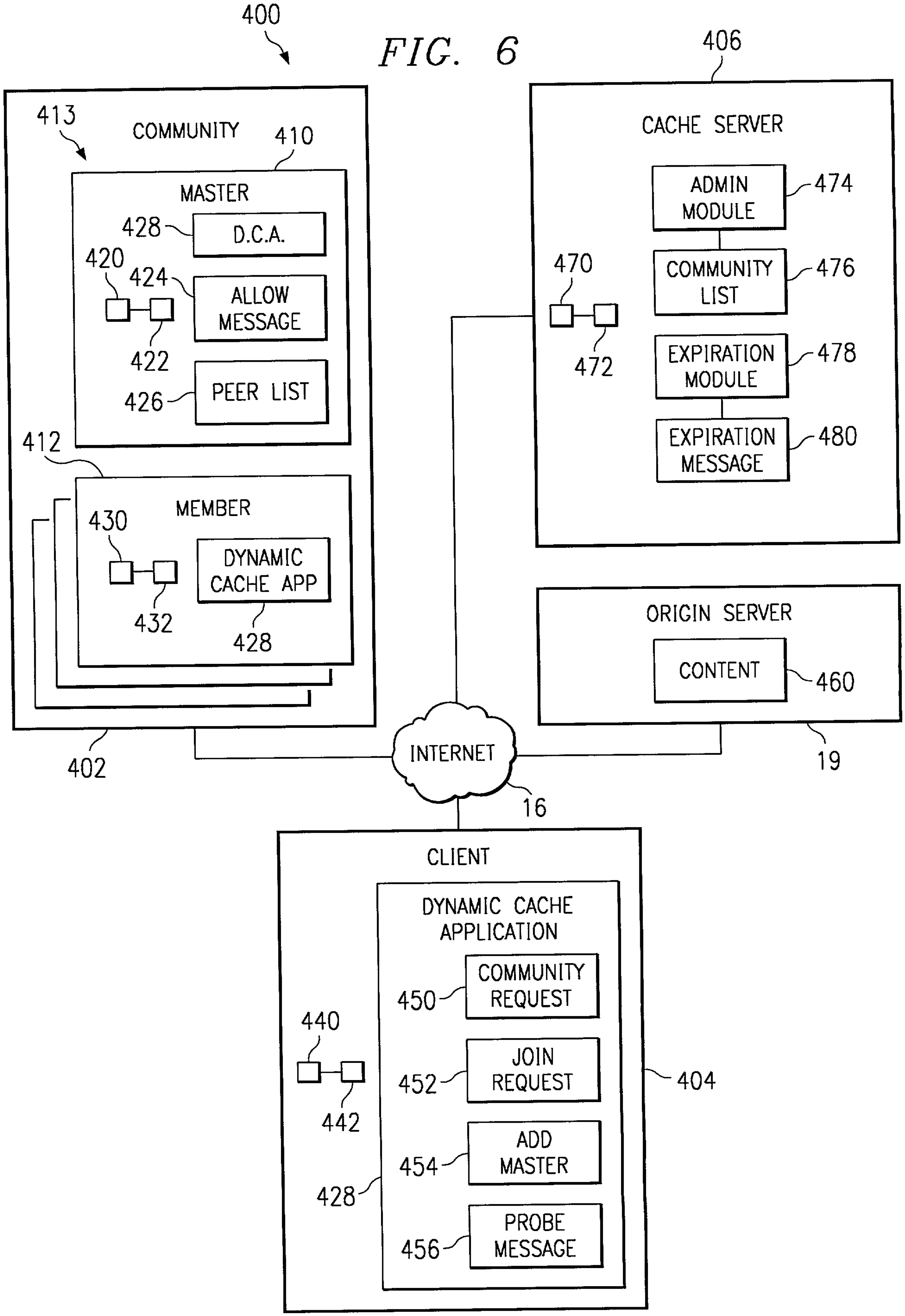
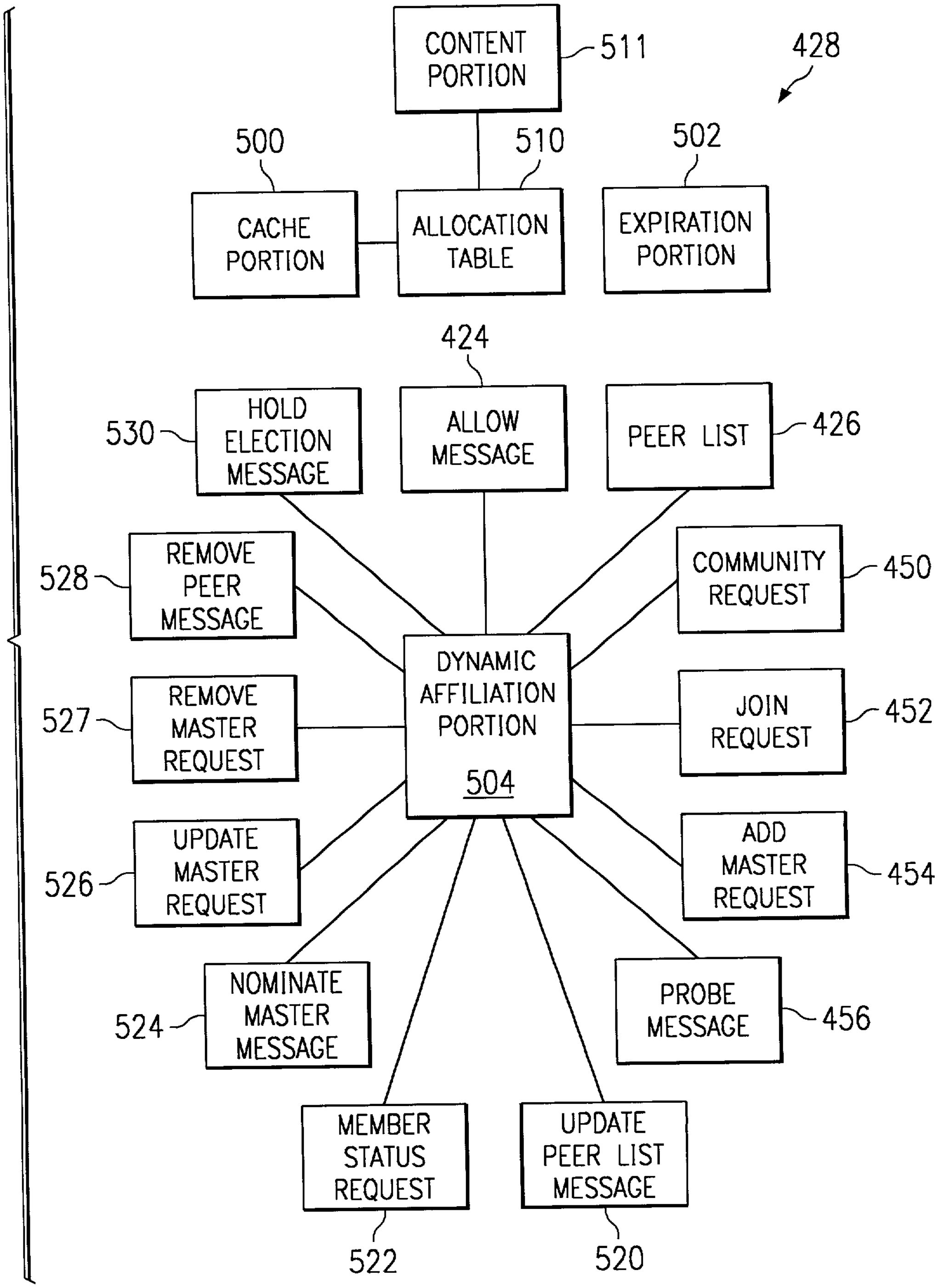

400
406
COMMUNITY
413
410
MASTER
428
D.C.A.
420
424
ALLOW MESSAGE
422
PEER LIST
426
412
MEMBER
430
DYNAMIC CACHE APP
432
428
402
CACHE SERVER
ADMIN MODULE
474
470
COMMUNITY LIST
476
472
EXPIRATION MODULE
478
EXPIRATION MESSAGE
480
ORIGIN SERVER
CONTENT
460
19
INTERNET
16
CLIENT
DYNAMIC CACHE APPLICATION
COMMUNITY REQUEST
450
440
JOIN REQUEST
452
404
442
ADD MASTER
454
428
PROBE MESSAGE
456

FIG. 7


CONTENT PORTION
511
428
500
510
502
CACHE PORTION
ALLOCATION TABLE
EXPIRATION PORTION
424
530
HOLD ELECTION MESSAGE
ALLOW MESSAGE
PEER LIST
426
528
REMOVE PEER MESSAGE
COMMUNITY REQUEST
450
DYNAMIC AFFILIATION PORTION
504
527
REMOVE MASTER REQUEST
JOIN REQUEST
452
526
UPDATE MASTER REQUEST
ADD MASTER REQUEST
454
524
NOMINATE MASTER MESSAGE
PROBE MESSAGE
456
MEMBER STATUS REQUEST
UPDATE PEER LIST MESSAGE
522
520

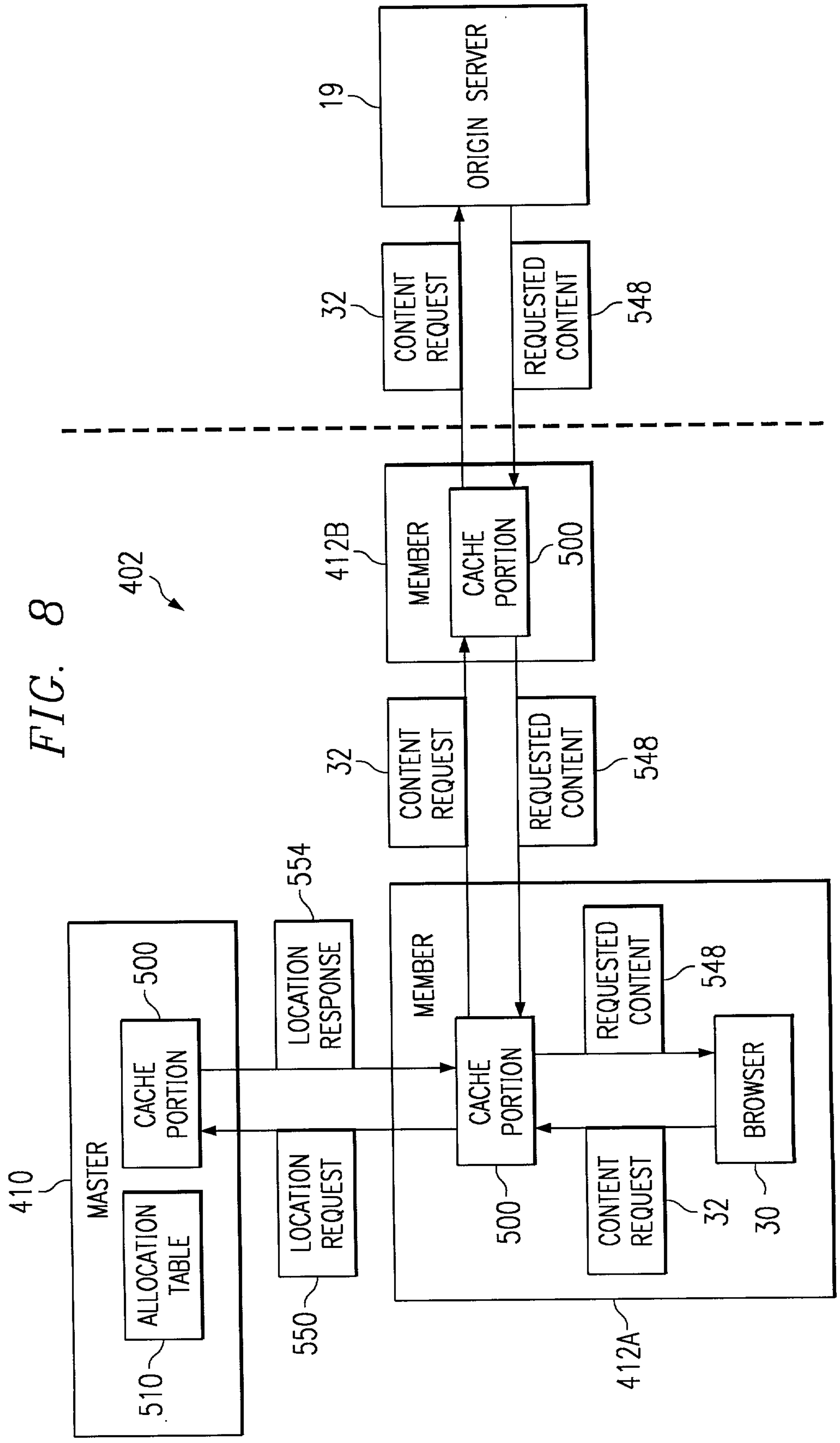


FIG. 8
402
410
MASTER
ALLOCATION TABLE
510
CACHE PORTION
500
LOCATION RESPONSE
554
LOCATION REQUEST
550
MEMBER
412A
CACHE PORTION
500
REQUESTED CONTENT
548
BROWSER
30
CONTENT REQUEST
32
32
CONTENT REQUEST
REQUESTED CONTENT
548
412B
MEMBER
CACHE PORTION
500
32
CONTENT REQUEST
REQUESTED CONTENT
548
19
ORIGIN SERVER

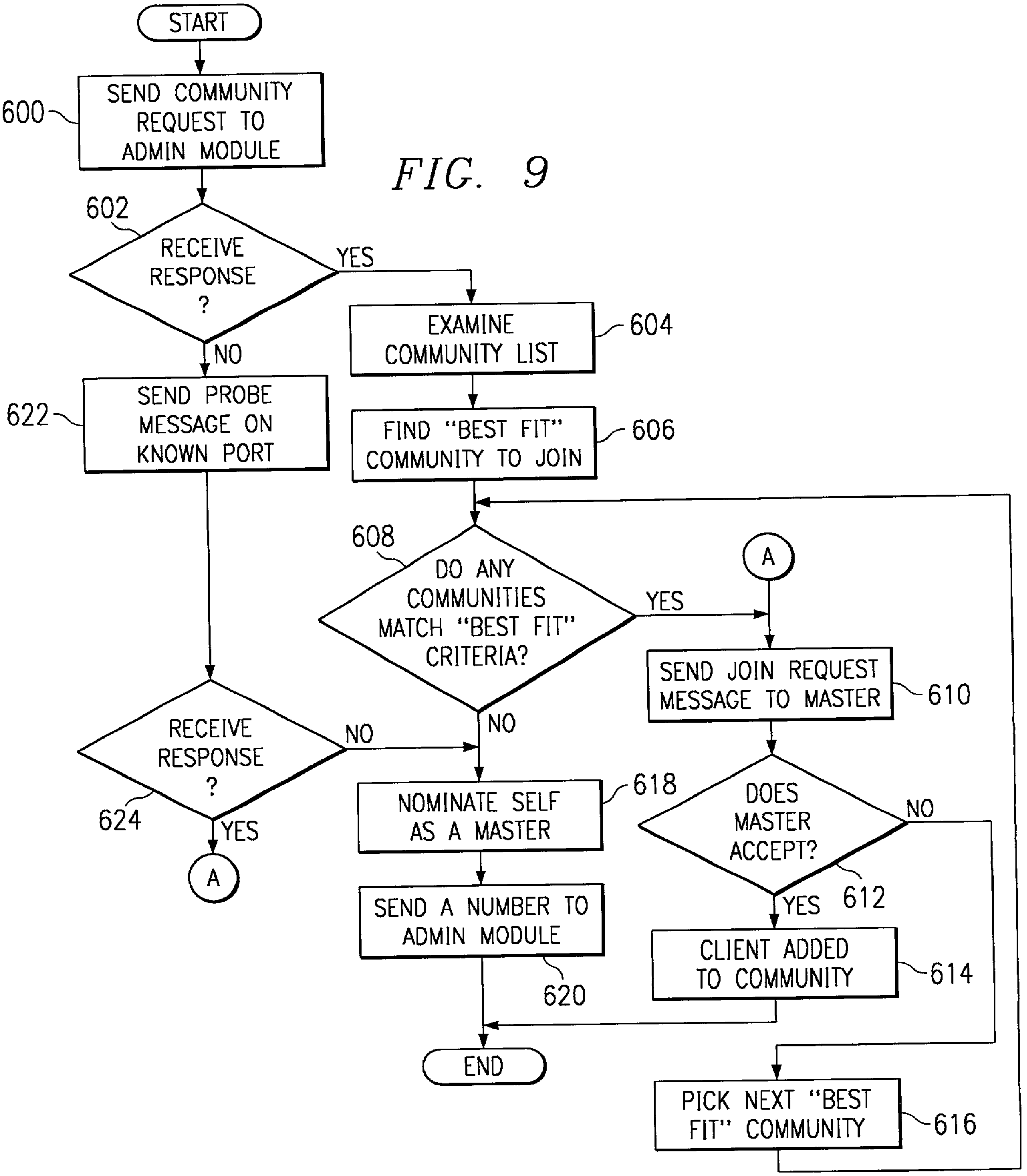

START
600
SEND COMMUNITY REQUEST TO ADMIN MODULE
602
RECEIVE RESPONSE ?
YES
NO
EXAMINE COMMUNITY LIST
604
FIND "BEST FIT" COMMUNITY TO JOIN
606
622
SEND PROBE MESSAGE ON KNOWN PORT
608
DO ANY COMMUNITIES MATCH "BEST FIT" CRITERIA?
YES
A
SEND JOIN REQUEST MESSAGE TO MASTER
610
NO
RECEIVE RESPONSE ?
NO
624
YES
A
NOMINATE SELF AS A MASTER
618
DOES MASTER ACCEPT?
NO
612
YES
SEND A NUMBER TO ADMIN MODULE
620
CLIENT ADDED TO COMMUNITY
614
END
PICK NEXT "BEST FIT" COMMUNITY
616

FIG. 9

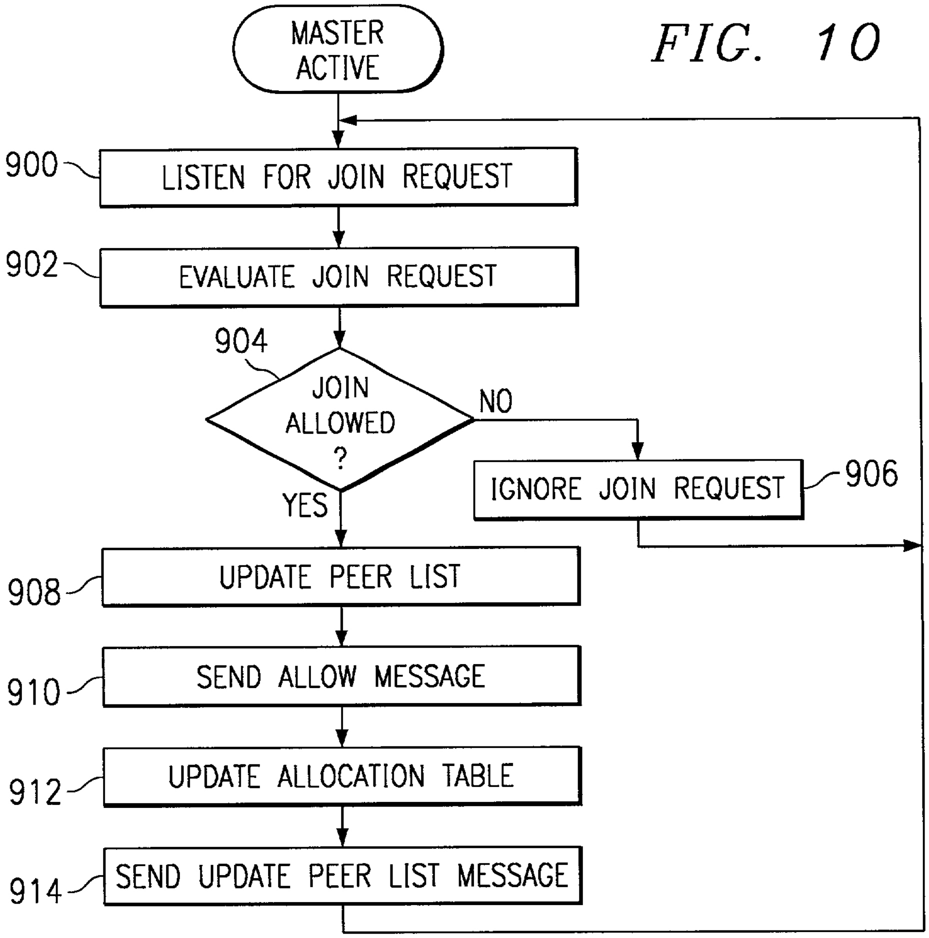

MASTER ACTIVE
900
LISTEN FOR JOIN REQUEST
902
EVALUATE JOIN REQUEST
904
JOIN ALLOWED ?
NO
IGNORE JOIN REQUEST
906
YES
908
UPDATE PEER LIST
910
SEND ALLOW MESSAGE
912
UPDATE ALLOCATION TABLE
914
SEND UPDATE PEER LIST MESSAGE

FIG. 10

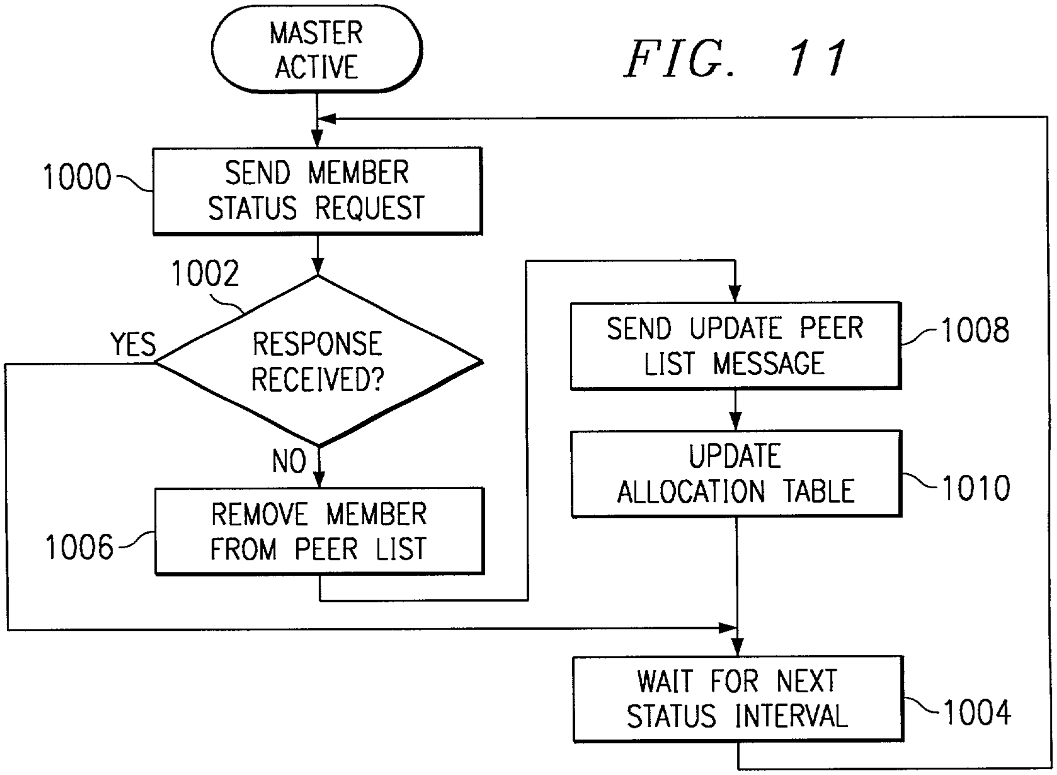

MASTER ACTIVE
1000
SEND MEMBER STATUS REQUEST
1002
RESPONSE RECEIVED?
YES
NO
1006
REMOVE MEMBER FROM PEER LIST
SEND UPDATE PEER LIST MESSAGE
1008
UPDATE ALLOCATION TABLE
1010
WAIT FOR NEXT STATUS INTERVAL
1004

FIG. 11

FIG. 12

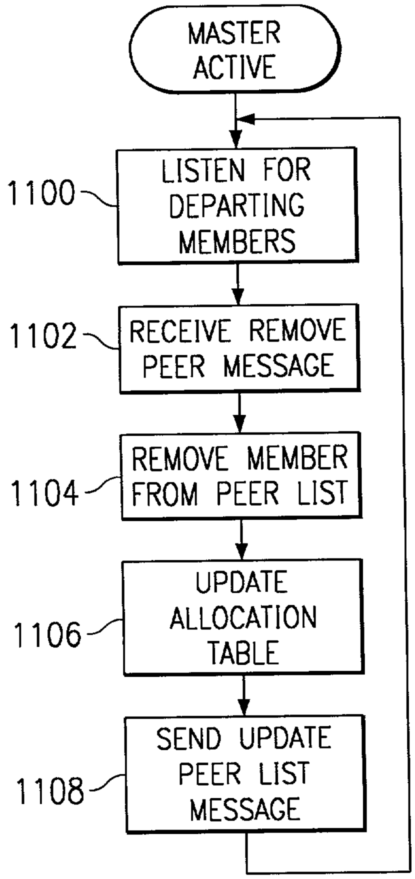

MASTER ACTIVE
1100
LISTEN FOR DEPARTING MEMBERS
1102
RECEIVE REMOVE PEER MESSAGE
1104
REMOVE MEMBER FROM PEER LIST
1106
UPDATE ALLOCATION TABLE
1108
SEND UPDATE PEER LIST MESSAGE

FIG. 13

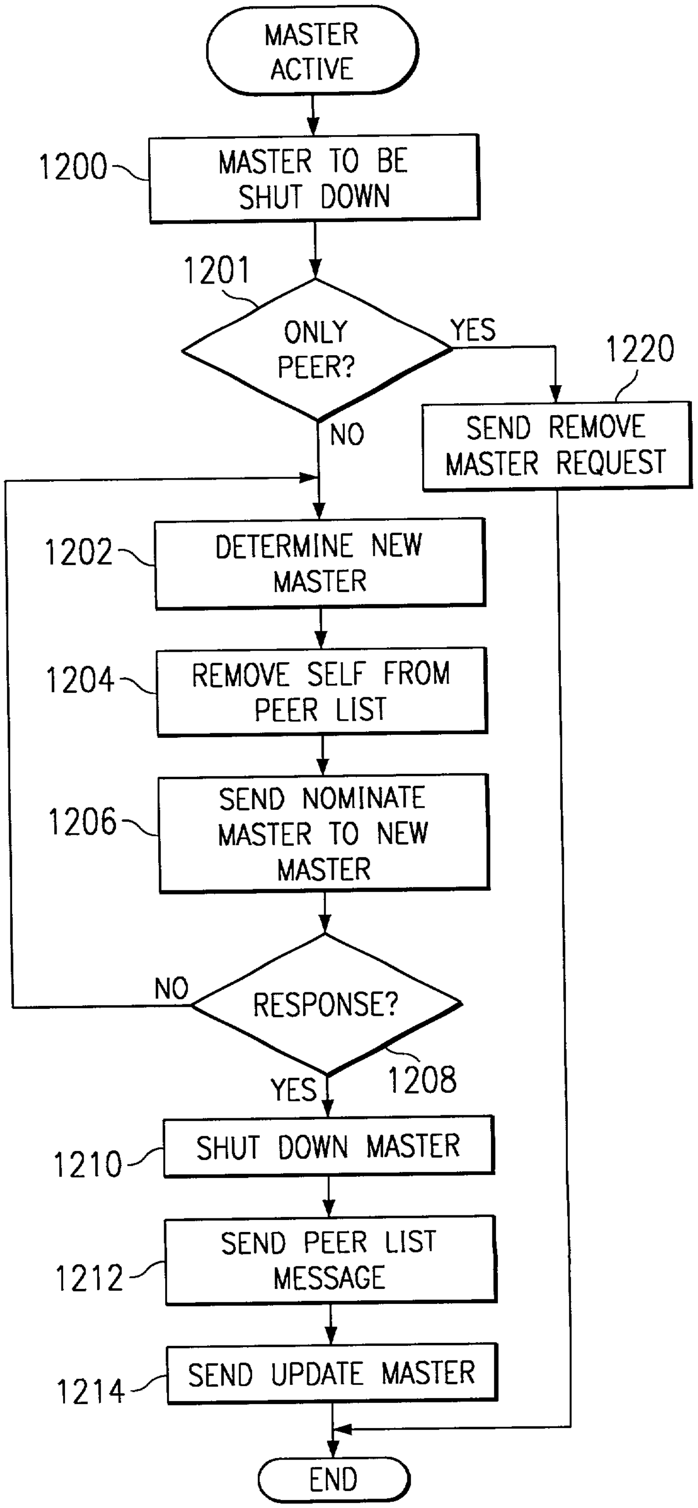
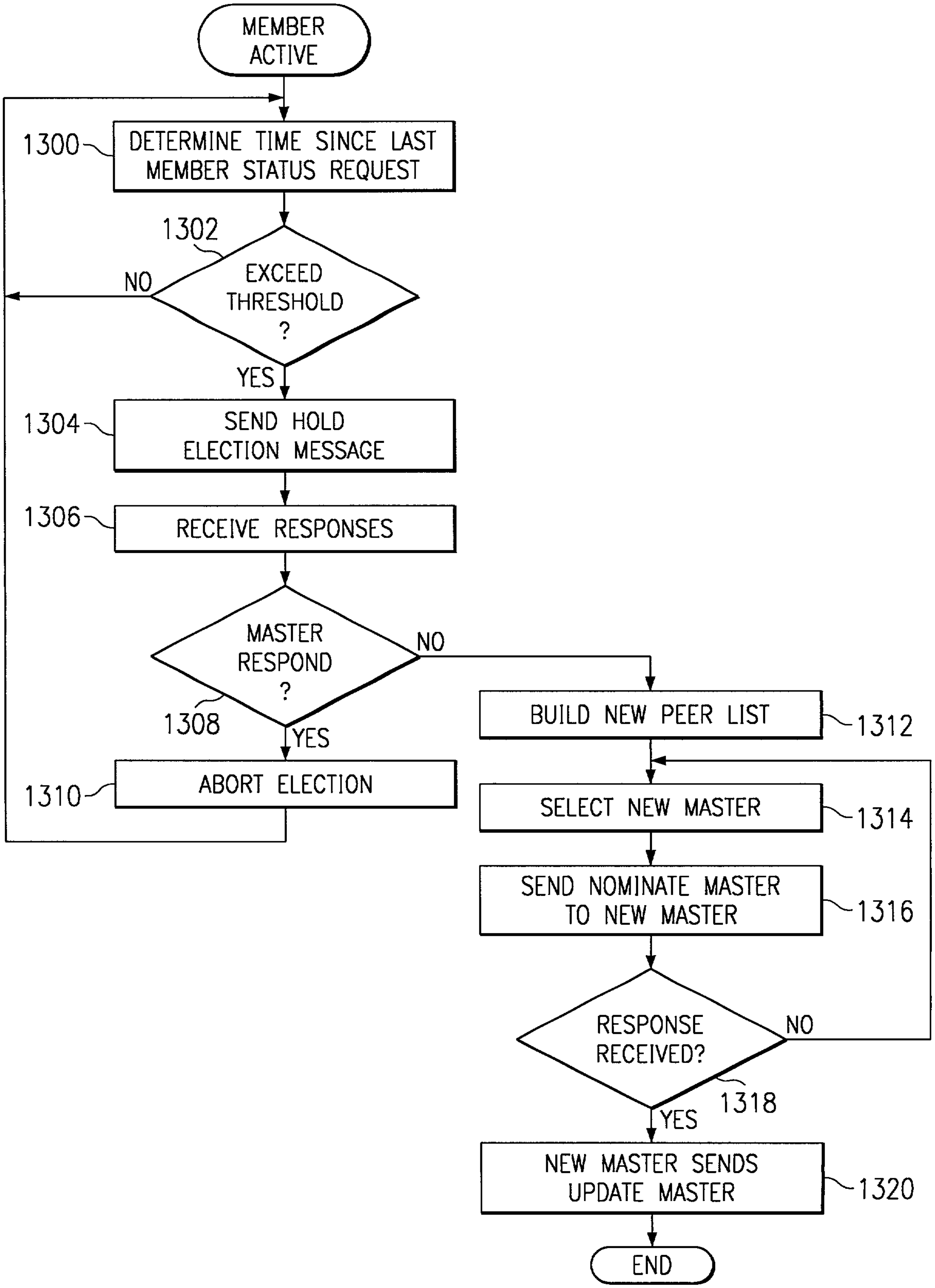

MASTER ACTIVE
1200
MASTER TO BE SHUT DOWN
1201
ONLY PEER?
YES
1220
SEND REMOVE MASTER REQUEST
NO
1202
DETERMINE NEW MASTER
1204
REMOVE SELF FROM PEER LIST
1206
SEND NOMINATE MASTER TO NEW MASTER
NO
RESPONSE?
1208
YES
1210
SHUT DOWN MASTER
1212
SEND PEER LIST MESSAGE
1214
SEND UPDATE MASTER
END

FIG. 14


MEMBER ACTIVE
1300
DETERMINE TIME SINCE LAST MEMBER STATUS REQUEST
1302
EXCEED THRESHOLD ?
NO
YES
1304
SEND HOLD ELECTION MESSAGE
1306
RECEIVE RESPONSES
MASTER RESPOND ?
1308
NO
YES
1310
ABORT ELECTION
BUILD NEW PEER LIST
1312
SELECT NEW MASTER
1314
SEND NOMINATE MASTER TO NEW MASTER
1316
RESPONSE RECEIVED?
NO
1318
YES
NEW MASTER SENDS UPDATE MASTER
1320
END

DECLARATION AND POWER OF ATTORNEY

As a below named inventor, I declare that:

My residence, post office address and citizenship are as stated below next to my name; that I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention or design entitled **METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING**, the specification of which is attached hereto;

that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above; and that I acknowledge the duty to disclose to the U.S. Patent and Trademark Office all information known to me to be material to patentability as defined in 37 C.F.R. § 1.56.

I hereby claim foreign priority benefits under 35 U.S.C. § 119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application(s) for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Number	Country	Date Filed	Priority Claimed

---NONE--

I hereby claim the benefit under 35 U.S.C. § 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application(s) in the manner provided by the first paragraph of 35 U.S.C. § 112, I acknowledge the duty to disclose to the U.S. Patent and Trademark Office all information known to me to be material to patentability as defined in 37 C.F.R. § 1.56 which became available between the filing date of the prior application(s) and the national or PCT international filing date of this application:

Application Serial Number	Date Filed	Status

---NONE---

I hereby appoint:

Jerry W. Mills	Reg. No. 23,005
Robert M. Chiaviello, Jr.	Reg. No. 32,461
Ann C. Livingston	Reg. No. 32,479
Thomas R. Felger	Reg. No. 28,842
Charles S. Fish	Reg. No. 35,870
Kevin J. Meek	Reg. No. 33,738
T. Murray Smith	Reg. No. 30,222
Barton E. Showalter	Reg. No. 38,302
David G. Wille	Reg. No. 38,363
Bradley P. Williams	Reg. No. 40,227
Terry J. Stalford	Reg. No. 39,522
Christopher W. Kennerly	Reg. No. 40,675
Harold E. Meier	Reg. No. 22,428
Douglas M. Kubehl	Reg. No. 41,915
Samir A. Bhavsar	Reg. No. 41,617
Thomas R. Nesbitt, Jr.	Reg. No. 22,075
James J. Maune	Reg. No. 26,946
Roger J. Fulghum	Reg. No. 39,678
Scott F. Partridge	Reg. No. 28,142
James B. Arpin	Reg. No. 33,470
Jay B. Johnson	Reg. No. 38,193
Robert W. Holland	Reg. No. 40,020
James L. Baudino	Reg. No. 43,486
Tara D. Knapp	Reg. No. 43,723
William R. Borchers	Reg. No. 44,549
Brian W. Oaks	Reg. No. 44,981
Luke K. Pedersen	Reg. No. 45,003
Matthew B. Talpis	Reg. No. 45,152
David M. Doyle	Reg. No. 43,596
Keiko Ichiye	Reg. No. 45,460
Jeffery D. Baxter	Reg. No. 45,560
Thomas A. Beaton	Reg. No. 46,543
Kurt M. Pankratz	Reg. No. 46,977
Brian E. Szymczak	Reg. No. 47,120
Thomas J. Frame	Reg. No. 47,232

Patent Agents:

Brian A. Dietzel	Reg. No. 44,656
Kevin R. Imes	Reg. No. 44,795

all of the firm of Baker Botts L.L.P., my attorneys with full power of substitution and revocation, to prosecute this application and to transact all business in the United States Patent and Trademark Office connected therewith, and to file and prosecute any international patent applications filed thereon before any international authorities.

Send Correspondence To:
Baker Botts L.L.P.
2001 Ross Avenue
Dallas, Texas 75201-2980

Direct Telephone Calls To:
Kevin J. Meek
at (214) 953-6680
Atty. Docket No. 066241.0117

I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Full name of the First Inventor: Keith A. Lowery

Inventor's signature:

Date: 1-5-01

Residence (City, County, State) Richardson, Dallas County, Texas

Citizenship: United States of America

Post Office Address: 1702 Drake Drive
Richardson, Texas 75081

Full name of the Second Inventor: Bryan S. Chin

Inventor's signature: Bryan S. Chin

Date: 1/5/01

Residence (City, County, State) Plano, Collin County, Texas

Citizenship: United States of America

Post Office Address: 4592 Spencer Drive
Plano, Texas 75024

Full name of the Third Inventor: David A. Consolver

Inventor's signature:

Date: 1/5/01

Residence (City, County, State) Arlington, Tarrant County, Texas

Citizenship: United States of America

Post Office Address: 6309 Woolwich Drive
Arlington, Texas 76001

Full name of the Fourth Inventor: Gregg A. DeMasters

Inventor's signature: Gregg A. DeMasters

Date: January 05, 2001

Residence (City, County, State) Plano, Collin County, Texas

Citizenship: United States of America

Post Office Address: 7301 Alma Drive, #624
Plano, Texas 75025

Applicant or Patentee: Keith A. Lowery, et al.
Attorney's Docket No.: 066241.0117
Serial or Patent No.: TBD

***Title:* METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING**

VERIFIED STATEMENT (DECLARATION) CLAIMING SMALL ENTITY STATUS (37 CFR 1.9(f) & 1.27(c)) -- SMALL BUSINESS CONCERN

I hereby declare that I am an official of the small business concern empowered to act on behalf of the concern identified below:

Name of Small Business Concern: epicRealm Inc.
Address of Small Business Concern: Palisades Central II, 2435 N. Central Expressway, Suite 300,
Richardson, Texas 75080

I hereby declare that the above-identified small business concern qualifies as a small business concern as defined in 13 CFR 121.12, and reproduced in 37 CFR 1.9(d), for purposes of paying reduced fees to the United States Patent and Trademark Office, in that the number of employees of the concern, including those of its affiliates, does not exceed 500 persons. For purposes of this statement, (1) the number of employees of the business concern is the average over the previous fiscal year of the concern of the persons employed on a full-time, part-time or temporary basis during each of the pay periods of the fiscal year, and (2) concerns are affiliates of each other when either, directly or indirectly, one concern controls or has the power to control the other, or a third party or parties controls or has the power to control both.

I hereby declare that rights under contract or law have been conveyed to and remain with the small business concern identified above with regard to the invention, entitled ***METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING*** by inventor, Keith A. Lowery, et al., described in the specification filed herewith.

If the rights held by the above-identified small business concern are not exclusive, each individual, concern or organization having rights in the invention is listed below, and no rights to the invention are held by any person, other than the inventor, who would not qualify as an independent inventor under 37 CFR 1.9(c) if that person made the invention, or by any concern which would not qualify as a small business concern under 37 CFR 1.9(d), or a nonprofit organization under 37 CFR 1.9(e):

----------------------------------NONE----------------------------------

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or my maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b)).

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

Name of Person Signing: Bradley A. Carl
Title of Person if other than owner: Vice President and General Counsel
Address of Person Signing: Palisades Central II, 2435 N. Central Expressway, Suite 300,
Richardson, Texas 75080

Signature:

Date: 12-21-00

DAL01:572584.1
066241.0117

PATENT APPLICATION SERIAL NO. ____________________

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

```
01/18/2001 HLE333   00000001 09759406

01 FC:201                        355.00 OP
02 FC:202                        480.00 OP
03 FC:203                        765.00 OP
```

PTO-1556
(5/87)

*U.S. GPO: 1999-459-082/19144

BEST AVAILABLE COPY

PATENT APPLICATION FEE DETERMINATION RECORD
Effective October 1, 2000

Application or Docket Number

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	105	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	105 minus 20=	* 85
INDEPENDENT CLAIMS	15 minus 3 =	* 12
MULTIPLE DEPENDENT CLAIM PRESENT	□	

* If the difference in column 1 is less than zero, enter "0" in column 2

SMALL ENTITY TYPE □		OR	OTHER THAN SMALL ENTITY	
RATE	FEE		RATE	FEE
BASIC FEE	355.00	OR	BASIC FEE	710.00
X$ 9=	765-00	OR	X$18=	
X40=	480-00	OR	X80=	
+135=		OR	+270=	
TOTAL	1600-00	OR	TOTAL	

CLAIMS AS AMENDED - PART II

		(Column 1)		(Column 2)	(Column 3)
AMENDMENT A		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
	Total	*	Minus	**	=
	Independent	*	Minus	***	=
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				□

SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

		(Column 1)		(Column 2)	(Column 3)
AMENDMENT B		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
	Total	*	Minus	**	=
	Independent	*	Minus	***	=
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				□

		OR		
RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

		(Column 1)		(Column 2)	(Column 3)
AMENDMENT C		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
	Total	*	Minus	**	=
	Independent	*	Minus	***	=
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				□

		OR		
RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
***If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE

*U.S. GPO: 2000-460-706/30103

CLAIMS ONLY

SERIAL NO. 09759406

FILING DATE 01/12/01

APPLICANT(S)

CLAIMS

	AS FILED		AFTER 1st AMENDMENT		AFTER 2nd AMENDMENT	
	IND.	DEP.	IND.	DEP.	IND.	DEP.
1	/					
2		/				
3		/				
4		/				
5		/				
6		/				
7		/				
8		/				
9		/				
10		/				
11		/				
12		/				
13		/				
14		/				
15		/				
16	/					
17		/				
18		/				
19		/				
20		/				
21		/				
22		/				
23		/				
24		/				
25		/				
26		/				
27		/				
28		/				
29		/				
30		/				
31	/					
32		/				
33		/				
34		/				
35		/				
36		/				
37		/				
38		/				
39		/				
40		/				
41	/					
42		/				
43		/				
44		/				
45		/				
46		/				
47		/				
48		/				
49		/				
50		/				
TOTAL IND.		⇩		⇩		⇩
TOTAL DEP.	⇦		⇦		⇦	
TOTAL CLAIMS						

	*		*		*	
	IND.	DEP.	IND.	DEP.	IND.	DEP.
51	/					
52		/				
53		/				
54		/				
55		/				
56		/				
57		/				
58		/				
59		/				
60		/				
61		/				
62		/				
63	/					
64		/				
65		/				
66		/				
67		/				
68		/				
69		/				
70		/				
71		/				
72		/				
73		/				
74		/				
75	/					
76		/				
77		/				
78		/				
79		/				
80		/				
81		/				
82		/				
83		/				
84		/				
85	/					
86		/				
87		/				
88		/				
89		/				
90		/				
91		/				
92		/				
93		/				
94		/				
95	/					
96		/				
97		/				
98	/					
99		/				
100		/				
TOTAL IND.		⇩		⇩		⇩
TOTAL DEP.	⇦		⇦		⇦	
TOTAL CLAIMS						

* *MAY BE USED FOR ADDITIONAL CLAIMS OR ADMENDMENTS*

FORM PTO-2022 (1-98)

U.S.DEPARTMENT OF COMMERCE
Patent and Trademark Office

*U.S. GPO: 1998-443-593/89152

CLAIMS ONLY

SERIAL NO.

FILING DATE

APPLICANT(S)

CLAIMS

	AS FILED		AFTER 1st AMENDMENT		AFTER 2nd AMENDMENT	
	IND.	DEP.	IND.	DEP.	IND.	DEP.
1	/					
2	/					
3	/					
4	/					
5	/					
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						
31						
32						
33						
34						
35						
36						
37						
38						
39						
40						
41						
42						
43						
44						
45						
46						
47						
48						
49						
50						
TOTAL IND.	15					
TOTAL DEP.	90					
TOTAL CLAIMS	105					

	*		*		*	
	IND.	DEP.	IND.	DEP.	IND.	DEP.
51						
52						
53						
54						
55						
56						
57						
58						
59						
60						
61						
62						
63						
64						
65						
66						
67						
68						
69						
70						
71						
72						
73						
74						
75						
76						
77						
78						
79						
80						
81						
82						
83						
84						
85						
86						
87						
88						
89						
90						
91						
92						
93						
94						
95						
96						
97						
98						
99						
100						
TOTAL IND.						
TOTAL DEP.						
TOTAL CLAIMS						

MAY BE USED FOR ADDITIONAL CLAIMS OR ADMENDMENTS

U.S.DEPARTMENT OF COMMERCE
Patent and Trademark Office

*U.S. GPO: 1998-443-593/89152

ATTORNEY DOCKET NO.:
066241.0117

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Keith A. Lowery, et al.
Filing Date:	January 12, 2001
Title:	METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING

Assistant Commissioner
for Patents
Washington, DC 20231

Dear Sir:

INFORMATION DISCLOSURE STATEMENT

Applicants respectfully request, pursuant to 37 C.F.R. §§ 1.56, 1.97 and 1.98, that the references listed on the attached PTO-1449 form be considered and cited in the examination of the above-identified patent application. Copies of the references are enclosed for the convenience of the Examiner. No representation is made that a search has been made, that the references are material to the patentability of the present application, or that the references qualify as prior art.

Applicants believe that this Information Disclosure Statement has been filed before the mailing date of the first Office Action in this case. Pursuant to 37 C.F.R. § 1.97(b), Applicants believe that no fee is due. The Commissioner is hereby authorized to charge any fees or credit any overpayments to Deposit Account No. 02-0384 of Baker Botts L.L.P.

Respectfully submitted,

Matthew B. Talpis
Reg. No. 45,152

<u>Correspondence Address</u>:
Kevin J. Meek, Esq.
Baker Botts L.L.P.
2001 Ross Avenue, Suite 600
Dallas, Texas 75201
Phone: (214) 953-6680
Fax: (214) 661-4680

Date: January 12, 2001

Page 1 of 1

PTO-1449 **Information Disclosure Citation in an Application**	Application No.	Applicant(s) Keith A. Lowery, et al.	
	Docket Number 066241.0117	Group Art Unit	Filing Date January 12, 2001

U.S. PATENT DOCUMENTS

		DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE
	A						
	B						
	C						
	D						
	E						

FOREIGN PATENT DOCUMENTS

		DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
							YES	NO
	F							

NON-PATENT DOCUMENTS

		DOCUMENT (Including Author, Title, Source, and Pertinent Pages)	DATE
	G	Author: John Borland, Article entitled "Net video not yet ready for prime time", printed from web site www.cnet.com on August 15, 2000.	February 5, 1999
	H	Author: Corey Grice, Article entitled "Start-up taps swapping tech to ease Web bottlenecks", printed from web site www.cnet.com on August 15, 2000.	August 11, 2000
	I	Author: VTrails.com. Product information regarding company's Full Duplex Packet Cascading technology printed from web site www.vtrails.com on August 15, 2000.	2000
	J	Author: Vinod Valloppillil and Keith W. Ross, Internet-Draft document entitled "Cache Array Routing Protocol v1.0", printed from web site www.globecom.net on November 14, 2000.	February 26, 1998
	K	Author: Microsoft Corporation. Information regarding the "Cache Array Routing Protocol (CARP) and Microsoft Proxy Server 2.0, printed from web site www.msdn.microsoft.com on October 9, 2000.	1997
	L	Author: Geek.com. Information news article entitled "Cash for your unused CPU cycles", printed from web site www.geek.com on June 30, 2000.	June 30, 2000
	M	Author: Dcypher.net. Answers to frequently answered questions v1.5, printed from web site www.dcypher.net on June 30, 2000.	April 10, 2000
	N	Author: Popular Power. Information regarding the company's Popular Power for Windows software, printed from web site www.popularpower.com on June 30, 2000.	None given.
	O	Author: ProcessTree Network. Information regarding their "for-pay" distributed processing network, printed from web site www.distributedscience.com on June 30, 2000.	None given.
	P	Author: Keith Schultz, Article entitled "Pushing content to the Internet's Edge", printed from web site www.internetweek.com on January 11, 2001.	December 4, 2000

EXAMINER	**DATE CONSIDERED**

EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP § 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to the applicant.

U.S. Patent and Trademark Office

#2

ATTORNEY DOCKET NO.:
066241.0117

PATENT APPLICATION
09/759,406

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Keith A. Lowery, et al.
Filing Date:	January 12, 2001
Serial No.:	09/759,406
Group Art Unit:	2185
Title:	METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING

Assistant Commissioner
for Patents
Washington, DC 20231

Dear Sir:

INFORMATION DISCLOSURE STATEMENT

Applicants respectfully request, pursuant to 37 C.F.R. §§ 1.56, 1.97 and 1.98, that the references listed on the attached PTO-1449 form be considered and cited in the examination of the above-identified patent application. Copies of the references are enclosed for the convenience of the Examiner. No representation is made that a search has been made, that the references are material to the patentability of the present application, or that the references qualify as prior art.

Applicants believe that this Information Disclosure Statement has been filed before the mailing date of the first Office Action in this case. Pursuant to 37 C.F.R. § 1.97(b), Applicants believe that no fee is due. The Commissioner is hereby authorized to charge any fees or credit any overpayments to Deposit Account No. 02-0384 of Baker Botts L.L.P.

Respectfully submitted,

Matthew B. Talpis
Reg. No. 45,152

Correspondence Address:
Kevin J. Meek, Esq.
Baker Botts L.L.P.
2001 Ross Avenue, Suite 600
Dallas, Texas 75201-2980
Phone: (214) 953-6680
Fax: (214) 661-4680

Date: April 19, 2001

Page 1 of 1

PTO-1449 **Information Disclosure Citation in an Application**	Application No. 09/759,406	Applicant(s) Keith A. Lowery, et al.	
	Docket Number 066241.0117	Group Art Unit 2185	Filing Date January 12, 2001

U.S. PATENT DOCUMENTS

		DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE
	A	6,026,474	02/15/00	Carter et al.	711	202	05/02/97
	B						
	C						
	D						
	E						

FOREIGN PATENT DOCUMENTS

		DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
							YES	NO
	F							

NON-PATENT DOCUMENTS

		DOCUMENT (Including Author, Title, Source, and Pertinent Pages)	DATE
	G		
	H		
	I		
	J		
	K		
	L		
	M		
	N		
	O		
	P		

EXAMINER	**DATE CONSIDERED**

EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP § 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to the applicant.

U.S. Patent and Trademark Office

04-23-01

GP/2185

ATTORNEY DOCKET NO.:
066241.0117

PATENT APPLICATION
09/759,406

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Keith A. Lowery, et al.
Filing Date:	January 12, 2001
Serial No.:	09/759,406
Group Art Unit:	2185
Title:	METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING

Box: Patent Application
Assistant Commissioner for Patents
Washington, D.C. 20231

Dear Sir:

CERTIFICATE OF MAILING BY EXPRESS MAIL

I hereby certify that the attached Information Disclosure Statement (1 page), PTO-Form 1449 (1 page with 1 reference), and a Baker Botts L.L.P. return receipt postcard (1 card) is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R. § 1.10 on this 19th day of April, 2001, addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.

Willie Jiles
Willie Jiles

Express Mail Receipt
No. 759178930US
Attorney Docket No.
066241.0117

DAL01:597462.1
066241.0117

#3

2185

PTO/SB/36 (4-01)
Approved for use through 10/31/2002. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

REQUEST TO RESCIND PREVIOUS NONPUBLICATION REQUEST 35 U.S.C. 122(b)(2)(B)(ii)

Application Number	09/759,406
Filing Date	January 12, 2001
First Named Inventor	Keith A. Lowery
Title	Method and System for Dynamic Distributed Data Caching
Atty Docket Number	066241.0117
Group Art Unit	2185
Examiner	unknown

I hereby **rescind** the previous request that the above-identified application not be published under 35 U.S.C. 122(b).

10 January 2002
Date

Signature

Matthew B. Talpis (Reg. No. 45,152)
Typed or printed name

This request must be signed in compliance with 37 CFR 1.33(b).

Note: Filing this rescission of a previous nonpublication request is considered the notice of a subsequent foreign or International filing required by 35 USC 122(b)(2)(B)(iii) and 37 CFR 1.213(c) if this rescission is filed no later than forty-five (45) days after the date of filing of such foreign or international application. See 37 CFR 1.137(f) if a notice of subsequent foreign or International filing required by 35 USC 122(b)(2)(B)(iii) and 37 CFR 1.213(c) is **not** filed within forty-five (45) days after the date of filing of the foreign or international application.

CERTIFICATE OF MAILING OR TRANSMISSION

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner For Patents, Box RCE, Washington, DC 20231, or facsimile transmitted to the U.S. Patent and Trademark Office on:

Name *(Print/Type)*	Elizabeth Turlo		
Signature		Date	10 January 2002

Burden Hour Statement: This collection of information is required by 37 CFR 1.213(b). The information is used by the public to rescind a previously filed request that an application not be published under 35 U.S.C. 122(b) (and the PTO to process that rescission). Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This form is estimated to take 6 minutes to complete. This time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.

UNITED STATES PATENT AND TRADEMARK OFFICE

COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
WASHINGTON, D.C. 20231
www.uspto.gov

APPLICATION NUMBER	FILING DATE	GRP ART UNIT	FIL FEE REC'D	ATTY.DOCKET.NO	DRAWINGS	TOT CLAIMS	IND CLAIMS
09/759,406	01/12/2001	2152	1600	066241.0117	11	105	15

CONFIRMATION NO. 2308

CORRECTED FILING RECEIPT

OC000000007547903

Baker Botts L.L.P.
2001 Rose Avenue
Dallas, TX 75201-2980

Date Mailed: 02/27/2002

Receipt is acknowledged of this nonprovisional Patent Application. It will be considered in its order and you will be notified as to the results of the examination. Be sure to provide the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION when inquiring about this application. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please write to the Office of Initial Patent Examination's Customer Service Center. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections (if appropriate).**

Applicant(s)

Keith A. Lowery, Richardson, TX;
Bryan S. Chin, Plano, TX;
David A. Consolver, Arlington, TX;
Gregg A. DeMasters, Plano, TX;

Assignment For Published Patent Application

epicRealm Inc.;

Domestic Priority data as claimed by applicant

Foreign Applications

If Required, Foreign Filing License Granted 02/23/2001

Projected Publication Date: Perfected

Non-Publication Request: No

Early Publication Request: No

**** SMALL ENTITY ****

Title

Method and system for dynamic distributed data caching

Preliminary Class
709

LICENSE FOR FOREIGN FILING UNDER
Title 35, United States Code, Section 184
Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Office of Export Administration, Department of Commerce (15 CFR 370.10 (j)); the Office of Foreign Assets Control, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

ATTORNEY DOCKET NO.:
066241.0117

PATENT APPLICATION
09/759,406

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Keith A. Lowery, et al.
Filing Date:	January 12, 2001
Serial No.:	09/759,406
Group Art Unit:	2152
Title:	METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING

Assistant Commissioner
for Patents
Washington, DC 20231

Dear Sir:

INFORMATION DISCLOSURE STATEMENT

Applicants respectfully request, pursuant to 37 C.F.R. §§ 1.56, 1.97 and 1.98, that the reference listed on the attached PTO-1449 form be considered and cited in the examination of the above-identified patent application. A copy of the reference is enclosed for the convenience of the Examiner. No representation is made that a search has been made, that the reference is material to the patentability of the present application, or that the reference qualifies as prior art.

Applicants believe that this Information Disclosure Statement has been filed before the mailing date of the first Office Action in this case. Pursuant to 37 C.F.R. § 1.97(b), Applicants believe that no fee is due. The Commissioner is hereby authorized to charge any fees or credit any overpayments to Deposit Account No. 02-0384 of Baker Botts L.L.P.

Respectfully submitted,

Matthew B. Talpis
Reg. No. 45,152

Correspondence Address:
Kevin J. Meek, Esq.
Baker Botts L.L.P.
2001 Ross Avenue, Suite 600
Dallas, Texas 75201-2980
Phone: (214) 953-6680
Fax: (214) 661-4680

Date: July 24, 2002

PTO-1449 Information Disclosure Citation in an Application	Application No. 09/759,406	Applicant(s) Keith A. Lowery, et al.	
	Docket Number 066241.0117	Group Art Unit 2152	Filing Date January 12, 2001

U.S. PATENT DOCUMENTS

		DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE
	A						
	B						
	C						
	D						
	E						

FOREIGN PATENT DOCUMENTS

		DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
							YES	NO
	F							

NON-PATENT DOCUMENTS

		DOCUMENT (Including Author, Title, Source, and Pertinent Pages)	DATE
	G	Patent Application Serial No. 09/759,392 filed January 12, 2001, entitled "*Method And System For Community Data Caching*," (Attorney's Docket 066241.0110), 78 total pages	1/12/2001
	H		
	I		
	J		
	K		
	L		
	M		
	N		
	O		
	P		

EXAMINER	**DATE CONSIDERED**

EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP § 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to the applicant.

U.S. Patent and Trademark Office

ATTORNEY DOCKET NO.: 07-26-02 PATENT APPLICATION Gp/2152
066241.0117 09/759,406

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Keith A. Lowery, et al.
Filing Date:	January 12, 2001
Serial No.:	09/759,406
Group Art Unit:	2152
Title:	METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING

#4

Assistant Commissioner for Patents
Washington, D.C. 20231

Dear Sir:

CERTIFICATE OF MAILING BY EXPRESS MAIL

I hereby certify that the attached Information Disclosure Statement (1 page), PTO-1449 (1 page with 1 attached reference), a Baker Botts L.L.P. return receipt postcard (1 card), and this Certificate of Mailing (1 page) is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R. § 1.10 on this 24th day of July, 2002, addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.

Willie Jiles
Willie Jiles

Express Mail Receipt
No. EL948546112US
Attorney Docket No.
066241.0117

ATTORNEY DOCKET NO.:
068241.0117

PATENT APPLICATION
09/759,406

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Keith A. Lowery, et al.
Filing Date:	January 12, 2001
Serial No.:	09/759,406
Group Art Unit:	2185
Title:	METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING

Assistant Commissioner
for Patents
Washington, DC 20231

Dear Sir:

INFORMATION DISCLOSURE STATEMENT

Applicants respectfully request, pursuant to 37 C.F.R. §§ 1.56, 1.97 and 1.98, that the references listed on the attached PTO-1449 form be considered and cited in the examination of the above-identified patent application. Copies of the references are enclosed for the convenience of the Examiner. No representation is made that a search has been made, that the references are material to the patentability of the present application, or that the references qualify as prior art.

Pursuant to 37 C.F.R. § 1.97(e)(1), Applicant hereby certifies that the item of information contained in this Information Disclosure Statement was cited in a communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of this Information Disclosure Statement.

Please charge any fees or credit any overpayment to Deposit Account No. 02-0384 of BAKER BOTTS L.L.P.

Respectfully submitted,

Matthew B. Talpis
Reg. No. 45,152

Correspondence Address:
Baker Botts L.L.P.
2001 Ross Avenue, Suite 600
Dallas, Texas 75201-2980
Phone: (214) 953-6680
Fax: (214) 661-4680

Date: 11-11-02

Page 1 of 1

PTO-1449 **Information Disclosure Citation in an Application**	Application No. 09/759,406	Applicant(s) Keith A. Lowery, et al.	
	D cket Number 066241.0117	Group Art Unit 2152	Filing Date January 12, 2001

U.S. PATENT DOCUMENTS

		DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE
	A						
	B						
	C						
	D						
	E						

FOREIGN PATENT DOCUMENTS

							TRANSLATION	
		DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	YES	NO
	F	EP 0993 163 A1	04/12/00	EPO	H04L	29/06	X	
	G	WO 98/22891	05/28/98	PCT	G06F	17/30	X	
	H							
	I							
	J							

NON-PATENT DOCUMENTS

		DOCUMENT (Including Author, Title, Source, and Pertinent Pages)	DATE
	K	Greg Barish, et al., "World Wide Web Caching: Trends and Techniques," XP-000949799, IEEE Communications Magazine, May 2000, pp. 178-185.	05/2000
	L	PCT/US 02-00886 Search Report, 6 pages.	10/2000
	M		
	N		
	O		
	P		
	Q		
	R		
	S		
	T		

EXAMINER	**DATE CONSIDERED**

EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP § 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to the applicant.

U.S. Patent and Trademark Office

11-13-02

2152

ATTORNEY DOCKET NO.:
066241.0117

PATENT APPLICATION
09/759,406 #5

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Keith A. Lowery, et al.
Filing Date:	January 12, 2001
Serial No.:	09/759,406
Group Art Unit:	2152
Title:	METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING

Assistant Commissioner for Patents
Washington, D.C. 20231

Dear Sir:

CERTIFICATE OF MAILING BY EXPRESS MAIL

I hereby certify that the attached Information Disclosure Statement (2 pages), PTO-1449 (1 page with 4 attached references), a Baker Botts L.L.P. return receipt postcard (1 card), and this Certificate of Mailing (1 page) is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R. § 1.10 on this 11th day of November, 2002, addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.

Willie Jiles
Willie Jiles

Express Mail Receipt
No. EV193174582US
Attorney Docket No.
066241.0117

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 993 163 A1

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
12.04.2000 Bulletin 2000/15

(51) Int. Cl.^{7}: **H04L 29/06**, G06F 17/30

(21) Application number: 99203169.0

(22) Date of filing: 28.09.1999

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 05.10.1998 US 166686

(71) Applicant:
Backweb Technologies Ltd.
Ramat Gan 52522 (IL)

(72) Inventors:
- Delaney, Hubert
Wilton CT 06897 (US)
- Ruppin, Adi
Ramat Gan 52584 (IS)
- Hass, Lior
Tel Aviv 62995 (IS)
- Faigon, Ofer
Jerusalem 96103 (IS)

(74) Representative:
Mertens, Hans Victor
van Exter Polak & Charlouis B.V.,
P.O. Box 3241
2280 GE Rijswijk (NL)

(54) **Distributed client-based data caching system and method**

(57) A system and method for enabling data package distribution to be performed by a plurality of peer clients connected to each other through a network, such as a LAN (local area network). Each peer client can obtain data packages from each other or from an external server. However, each peer client preferably obtains data packages from other peer clients, rather than obtaining data packages from the external server.

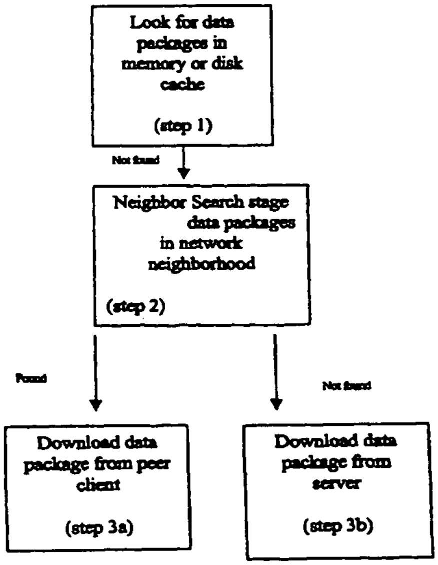

Look for data packages in memory or disk cache
(step 1)
Not found
Neighbor Search stage data packages in network neighborhood
(step 2)
Found
Not found
Download data package from peer client
(step 3a)
Download data package from server
(step 3b)

Fig. 1B

Printed by Xerox (UK) Business Services
2.16.7 (HRS)/3.6

Description

[0001] The present invention relates to a distributed client-based data caching system and method. Specifically, the system and method of the present invention enable data packages to be served to a client through a flexible, non-deterministic distributed system of peer clients which cache the data packages, in order to maximize efficiency and speed for serving the data package to the client.

[0002] Networks which connect two or more computers, such as the Internet or intranets, enable client computers to obtain data packages, such as documents, images, messages, data packages or other types of data, from remote storage media which are not installed on the client computer itself. Instead, these remote storage media we managed and operated through a remote computer, known as a server computer or simply as a "server" (in the same vein, the client computer is also often termed only a "client"). The advantage of such a system is that the client computer can potentially obtain data from any server on the network. The disadvantage of the system is the requirement for sufficient bandwidth on the network to enable data to be transmitted from the server to the client. Furthermore, if the load is not evenly distributed between servers on the network, one server may become overwhelmed with requests, thereby decreasing the speed and efficiency of retrieval. Thus, currently many networks cannot provide rapid and efficient data retrieval due to the heavy demands placed upon the available bandwidth.

[0003] Proxy servers are often installed to conserve bandwidth on an Internet connection or on connections to other LANs (local area networks). These proxy servers cache frequently accessed data, thereby reducing the load on the main server, and distributing demand for bandwidth more evenly across the network. Unfortunately, such proxy servers are typically expensive to maintain. Furthermore, proxy servers require dedicated computers to be installed and configured. Each computer on the LAN has to be separately configured in order to communicate with the proxy server. Such configuration is deterministic, such that each client must be configured to communicate with each proxy server separately. Thus, proxy servers have many drawbacks.

[0004] A more useful solution would enable Intranets to reap the benefits of the proxy server, without requiring dedicated machines and without requiring any special installation or configuration. Furthermore, such a solution would not be deterministic, such that each client could communicate with more than one server according to the load on each server, rather than according to the configuration of the client itself. Unfortunately, such a solution is not currently available.

[0005] Therefore, there is an unmet need for, and it would be highly useful to have, a distributed client-based data caching system and method which enable data to be stored and retrieved from a plurality of peer clients, or "caching entities", yet which does not require any special configuration or installation of separate servers.

[0006] The present invention is of a distributed client-based data caching system and method, which enable data to be served to a client through a flexible, non-deterministic distributed system of caching entities, in order to maximize efficiency and speed for serving the document to the client. The caching entities are peer clients which serve the data to each other, thereby reducing the amount of bandwidth required to obtain data from an external server.

[0007] According to the present invention, there is provided a method for distributing data packages across a network, the network featuring an external server for serving at least one data package, the external server being a dedicated server, the steps of the method being performed by a data processor, the method comprising the steps of: (a) providing a plurality of peer clients attached to the network and a list of data packages being stored by each of the plurality of peer clients, each data package on the list of data packages having an entry, the entry indicating a unique identifier for the data package and a location of the data package in at least one of the plurality of peer clients; (b) examining the list of data packages by a first peer client to find an entry for a data package; and (c) if the entry for the data package is present on the list of data packages of the first peer client, retrieving the data package from the location at another of the plurality of peer clients according to the entry for the data package.

[0008] Alternatively, the list of data packages is stored on the external server.

[0009] According to preferred embodiments of the present invention, the list of data packages is stored on at least the first peer client. Preferably, if alternatively the entry for the data package is absent from the list of data packages of the first peer client the method further comprises the steps of: (d) sending a request message for the data package by the first peer client to at least one other peer client; and (e) if a response message is received by the first peer client from the at least one other peer client retrieving the data package from the at least one other peer client by the first peer client.

[0010] Preferably, the request message and the response message are transmitted to the plurality of peer clients by broadcasting. Alternatively, the request message and the response message are transmitted to the plurality of peer clients by multicasting. Also alternatively, the request message and the response message are transmitted to the plurality of peer clients by polling each peer client individually.

[0011] Also alternatively and preferably, if the response message is not received from the at least one other peer client by the first peer client the method further comprises the step of: (f) obtaining the data package by the first peer client from the external server. Preferably, the method further comprises the step of sending a response message by the first peer client to the at least one other peer client substantially before the first peer client obtains the data package from the external server. More preferably, the list of data packages is stored on each of the plurality of peer clients, and the method further comprises the steps of: (g) receiving the response message from the first peer client by the at least one other peer client; and (h) altering the list of data packages being stored by the at least one other peer client for indicating the location of the data package according to the response message.

[0012] Alternatively, the list of data packages is stored on each of the plurality of peer clients, and the method further comprises the steps of: (g) receiving the response message from the first peer client by the at least one other peer client; and (h) altering the list of data packages being stored by the at least one other peer client for indicating the location of the data package according to a probabilistic function.

[0013] Preferably, the probabilistic function is performed according to a set of equations:

$$\text{New location} = \left\{ \begin{array}{ll} \text{Old location} & Po(x) = 1/(\text{generation}+1) \\ \text{New location} & Pn(x) = 1-1/(\text{generation}+1) \end{array} \right.$$

wherein Pn(x) is a probability that the new location is substituted for the old location, Po(x) is a probability that the old location is retained, and "generation" indicates how many times the location had been previously changed.

[0014] Also preferably, an upper limit is predetermined for a number of the plurality of peer clients served substantially simultaneously by the at least one other peer client, such that if a number of the plurality of peer clients served substantially simultaneously by the at least one other peer client is greater than the upper limit, the method further comprises the step of: (d) sending a busy message from the at least one other peer client to the first peer client.

[0015] Preferably, the external server is a Web server, and the plurality of peer clients is a plurality of Web browsers.

[0016] Also preferably, the external sewer is a BackWeb™ server, and the plurality of peer clients is a plurality of BackWeb™ clients.

[0017] Preferably, the unique identifier for the data package is an MD5 digest of the data package.

[0018] According to still other preferred embodiments of the present invention, the step of retrieving the data package is performed according to a protocol based on TCP/IP. Preferably, the protocol is HTTP. Alternatively and preferably, the protocol is FTP.

[0019] Hereinafter, the term "protocol based on TCP/IP" includes any such protocol, including but not limited to the HTTP (hypertext transfer protocol) and FTP (file transfer protocol) protocols.

[0020] Hereinafter, the term "data package" refers to any discrete, identifiable unit of data, including but not limited to documents, images, messages, data packages or any other type of data.

[0021] Hereinafter, the term "computing platform" refers to a particular computer hardware system or to a particular software operating system. Examples of such hardware systems include, but are not limited to, personal computers (PC), Apple Macintosh™ computers, mainframes, minicomputers and workstations, which are also non-limiting examples of data processors for operating a software application under an operating system. Examples of such software operating systems include, but are not limited to, UNIX, VMS, Linux, MacOS™, DOS, one of the Windows™ operating systems by Microsoft Corp. (USA), including Windows NT™, Windows 3.x™ (in which "x" is a version number, such as "Windows 3.1™"), Windows95™ and Windows98™ .

[0022] For the present invention, a software application could be written in a substantially suitable programming language, which could easily be selected by one of ordinary skill in the art. The programming language chosen should be compatible with the operating system according to which the software application is executed. Examples of suitable programming languages include, but are not limited to, C, C++ and Java.

[0023] Hereinafter, the term "broadcast' may also include "multicast" as well.

[0024] The invention is herein described, by way of example only, with reference to the accompanying drawings, wherein:

FIGS. 1A and 1B are schematic block diagrams of an exemplary basic system and method according to the present invention;

FIGS. 2A-2E are schematic block diagrams of an exemplary request/response protocol and method according to the present invention;

FIG. 3 is a schematic block diagram of an exemplary preferred data-flow diagram according to the present invention;

FIG. 4 is a flowchart of a method for operating the system of the present invention with Web browsers; and

FIGS. 5A and 5B are exemplary request and response messages according to the present invention.

[0025] The present invention is of a distributed client-based data caching system and method, which enable data to be sewed to a client through a flexible, non-deterministic distributed system of caching entities, in order to maximize efficiency and speed for serving the data to the client. The caching entities we peer clients which serve the data to each other, thereby reducing the amount of bandwidth required to obtain data from an external server.

[0026] The system and method of the present invention enable clients to share data packages among themselves across their local network neighborhood, for example within a LAN, thereby eliminating the need for a specialized proxy server. Furthermore, the network traffic is not significantly affected, since modem network architectures are well suited for peer-to-peer communications. Most currently operating networks have a star topology, using switching hubs, in which communication between two peers does not affect simultaneous communication among other nodes on the network. Thus, the system of the present invention overcomes the drawbacks of a proxy server, yet does not add significant loads to the traffic on the network itself.

[0027] For currently available client-server software applications known in the art, whenever a client requires a data package, the following algorithm is performed. First, the software application attempts to locate the data package locally on the memory or on the disk or disks of the client. Then, if the data package is not found locally, the software application retrieves the data package from the appropriate server.

[0028] By contrast the operation of the system of the present invention adds an intermediate step. For the present invention, if the data package is not found locally, an attempt is made to retrieve the data package from a peer client on the local network "neighborhood" before resorting to retrieving the data package from the server.

[0029] Thus, for the system of the present invention, every client actually functions as a caching proxy. Once a client requires a data package, it queries all the hosts, which are actually peer clients, on the local network for that data package. If no neighboring peer client has the data package, the client retrieves the data package from the external server as usual. However, if a neighboring client already has the required data package, the requesting client will download this data package from the peer client rather than from the external server.

[0030] The principles and operation of the distributed client-based data caching system according to the present invention may be better understood with reference to the drawings and the accompanying description.

[0031] Figure 1A is a schematic block diagram of an exemplary system according to the present invention, while Figure 1B is a flowchart of the operation of the system of Figure 1A. Figure 1A shows a system **10** which includes a plurality of peer clients **12** connected by a local network **14** of some type, for example a LAN, indicated by the heavier line in Figure 1A. Two peer clients **12**, labeled as "peer client 1" **20** and "peer client 2" **22**, are shown for the purposes of illustration only and without intending to be limiting in any way. Each peer client **12** is also connected to an external server **16** of some type by an external connection **18**. Although only one external server **16** is shown, a plurality of external servers could also be implemented. External server **16** is a dedicated server, in the sense that this server has a primary or at least a substantially significant role as a server for data packages. As shown for the purposes of illustration, external connection **18** only connects to local network **14** at one point, although multiple such external connections could also be implemented (not shown). In addition, external connection **18** could also optionally connect each peer client **12** directly to server **16** (not shown).

[0032] The operation of system **10** according to the present invention is illustrated with reference also to Figure 1B. In step 1, peer client **12**, such as peer client **12** looks for a data package in the local memory or disk cache of that particular peer client **12**. If the desired data package is not found on the local disk cache, then in step 2, peer client **12** queries any other peer client(s) **12** on local network **14** to determine whether any other peer client **12** has a particular data package. For example, peer client **20** could query peer client **22**, to determine whether peer client **22** has the desired data package. In step 3a, if peer client **22** has the desired data package, then peer client **20** obtains the data package from peer client **22**. Alternatively, as shown in step 3b, if peer client **22** does not have the desired data package, then peer client **20** obtains the data package from server **16** through external connection **18**. Thus, every peer client **12** is also potentially a server which is internal to local network **14**, and hence could be described as an "internal server" to distinguish peer client **12** from external server **16**.

[0033] Each peer client **12** could also be described as a "caching entity" and the data stored by each client for serving to other peer clients **12** as "cached data" or "cached data packages".

[0034] A number of different possible embodiments of the system of the present invention can be implemented, of which two illustrative embodiments are shown with reference to the Figures below. Briefly, Figures 2A-2D illustrate an exemplary embodiment of the system of the present invention for implementation with the software application of BackWeb^{TM} (BackWeb Technologies Ltd., Ramat Gan, Israel) on a local area network (LAN). Figures 4 and 5A-5B illustrate an exemplary embodiment of the system of the present invention for implementation with a Web browser software application on the Internet.

[0035] Figure 2A shows an exemplary local network **24** which featur s a plurality of peer clients **12** of which three are shown for the purposes of discussion only and without intending to be limiting in any way. For the purposes of discussion only, suppose a peer client 26, labeled "A", wishes to obtain four data packages "W", "X', "Y" and "Z". None of these data packages are local to peer client **26**, which must therefore obtain these data packages from either another peer client **12** as an internal server, or from an external server (not shown). Local area network **24** features two other peer clients **12**: peer client **28**, labeled "B", and peer client **30**, labeled "C". Peer client **26** must therefore first communicate a request to peer client **28** and peer client **30** to see if the desired data packages are available at either location, and then peer client **26** must obtain these data packages from peer client **28** or peer client **30** if the desired data packages are available.

[0036] Preferably, two protocols are used for communication between peer clients on a local area network (LAN), a *data package-exchange* protocol and a *control* protocol. Specifically, the data package exchange protocol is used to transfer data packages between peer clients, once the desired data package has been located, and is described in greater detail with respect to Figure 2B below. The *control* protocol enables each peer client to efficiently build and maintain tables which describe the location of available data packages across the local area network by exchanging messages.

[0037] Each peer client maintains two hash-tables which contain information about data package location: a local-data packages table and a network-data packages table. The local-data packages table is a hash-table of data packages which reside on the storage medium or media of the peer client itself. The network-data packages table is a hash-table of data packages which reside on the storage medium or media of other clients on the local network. This table contains the local area network address of the peer client on which each data package is being stored. The size of this hash-table is preferably limited in order to reduce memory consumption. More preferably, each entry in the table has a time-stamp, such that older entries are purged when the size of the table exceeds the upper permissible limit

[0038] In order to effectively identify the desired data package, preferably each data package has unique identifier or "fingerprint" associated with it. More preferably, this unique identifier is an MD5 digest of the content of the data package (for a description of the MD5 specification, which is an industry standard and would therefore be obvious to one of ordinary skill in the art, see "RFC 1321" at *http://ds.internic.net/rfc/rfc1321.txt).*

[0039] Once any peer client knows both the unique identifier and the location of the data package on the local network, that client can then proceed to download the data package. However, the peer-client may not know the location of the desired data package, in which case the client must follow a control protocol according to the present invention in order to determine the location of the desired data package and to enable the client to build these hash tables with respect to future attempts to locate a data package.

[0040] The control protocol is used to provide each client with knowledge about the locations of data packages across the local network. In the preferred implementation illustrated with respect to Figures 2A-2D, control messages are preferably sent and received as broadcast or multicast packets. Local area networks such as Ethernet networks support broadcast or multicast packets such that all peer clients on a local area network receive the broadcast or multicast packets. Effectively, a single packet can be sent to all peer clients by using broadcast or multicast, thereby reducing the amount of traffic on the network required as a result of transmitting the request message (see for example Chapter 12, "Broadcasting and Multicasting", of *TCP/IP Illustrated Volume*, by W. Richard Stevens, Addison-Wesley, 1994). However, optionally the system of the present invention could poll each peer client individually with a control message for that peer client, although this is not preferred since such individually addressed messages would consume excessive amounts of available bandwidth. In such a situation, preferably polling would be restricted to a certain group of peer clients as internal servers, in order to reduce the amount of traffic on the local area network.

[0041] For the preferred implementation in which broadcast or multicast is used, more preferably, the decision to select either IP multicast or broadcast is made according to the configuration set by the network administrator for the local area network. IP multicast is preferable in terms of load on the clients of the local network, but may not be supported on all platforms (operating systems). More preferably, the TTL or Time to Live may be configured. The TTL specifies the number of routers a packet can cross before being dropped. Configuring the TTL enables data package sharing to be expanded across subnet boundaries.

[0042] As shown with respect to Figure 2B, the control protocol of the present invention preferably operates as follows. In step 1, peer client"A" from Figure 2A looks for a data package on the local storage medium or media. In step 2, since the data package was not found locally on the medium or media of peer client "A", peer client "A" must download the data package and therefore preferably multicasts (or alternatively broadcasts) a *request* message. A request message preferably contains a protocol identifying version number (PVN) for the control protocol of the present invention and a list of MD5 digests of the needed data packages, as shown in Figure 2C.

[0043] Optionally and preferably, if more than one data package is desired, a list of requested data packages is included in the request message rather than a single MD5 digest, in order to reduce the total number of request messages on the network.

[0044] In step 3, the neighboring clients, shown as peer clients "B" and "C" in Figure 2A, receive this request message and search for the requested data package in their local-data packages hash-table. A peer client which does not find the data package locally does not reply, as shown in step 4a. Otherwise, in step 4b the peer client sends a *response* message, preferably after waiting a short random time interval to determine whether another peer client will respond first More preferably, th peer client does not distribute the response message if another client responded previously, in order to reduce unnecessary traffic on the local area network. Also more preferably, the peer client distributes the response message by broadcast or multicast.

[0045] For example, as shown in Figure 2A, if peer client "A" requests a data package "W", peer client "B" would reply with the response message, since peer client "B" has the data package stored locally. By contrast, peer client "C" would not reply with a response message, since peer client"C" does not have data package "W" stored locally. On the other hand, if peer client "A" requests a data package "X", both peer client "B" and peer client "C" could respond. In this situation, preferably only peer client "B" or peer client "C" would respond, depending on which peer client had the shorter random interval for waiting before sending the response message.

[0046] More preferably, responses are sent only for data packages with yet unknown locations. For example, suppose client "A" requests data packages "W", "X", "Y" and "Z". Client "B" has data packages "W", "X" and "Y", and is the first to reply, with a reply message indicating possession of data packages "W", "X" and "Y". Suppose another client "C" has data packages "X", "Y" and "Z". Since it replied after client "B", the response message from client "C" will only indicate possession of data package "Z" because this is the only data package with an as yet unknown location.

[0047] A response message optionally contains the identifying PVN, the list of MD5 digests of data packages that were found and a TCP port number, as shown in Figure 2D. The port number identities on which TCP port the responding peer client is waiting for data package requests. Alternatively, the response message optionally contains other indicators which enable the requesting client to retrieve one or more data packages from the responding peer. Preferably, response messages are also be broadcast for data packages which are currently being downloaded from an external server, for reasons described in greater detail below.

[0048] In step 5, the peer client downloads the data package or data packages. In principle, according to a relatively simple embodiment of the present invention, at this stage the requesting client either receives a reply and downloads the data packages from the client that replied; or, if a reply is not received within a certain period of time, proceeds to download these data packages from an external server. If the peer client is downloading a data package from another peer client as an internal server, the data package-exchange protocol is used to obtain the data package. The data package-exchange protocol is based on some appropriate peer-to-peer communication protocol, including but not limited to the HTTP protocol (see RFC-2068, "Hypertext Transfer Protocol - HTTP/1.1", available from http://ds.internic.net/rfc/rfc2068.txt as of September 23, 1998).

[0049] Preferably, a more complex implementation is employed, since such a simple implementation may cause multiple clients to fetch the same data packages from the external server simultaneously. This situation would arise if several peer clients need to download the same data packages at approximately the same time, which is a very probable scenario for push clients for which content delivery is triggered by an external server, since none of these clients would receive a response to its request. Instead, the other clients would still be downloading the data package when the new client request is broadcast such that none of them would be ready to serve these data packages. Thus, many or even all of the clients would attempt to retrieve the data package from the external server and not from another peer client, thereby increasing the amount of traffic on the network and reducing the efficiency of operation of the system of the present invention.

[0050] Preferably, the problem is solved by notifying other clients when a first client is downloading the data package from the external server, even if the process of retrieving the data package is not yet complete. In this preferred embodiment, the first client which requires the data package obtains the data package from the external server. Other clients which require the data package will then download it from the first client even if the first client is still in the process of retrieving the data package from the external server. The preferred embodiment of the method of the present invention is described in greater detail with regard to Figure 2E.

[0051] In step 1, the requesting client again transmits the request, again preferably by broadcasting or multicasting, and then waits for a response. If no response is received within a certain period of time, in step 2 the client transmits a *response* message as if replying to its own request, indicating that this client either has the data package, or in this case, that the client is retrieving the data package. In step 3, the client retrieves the data package from the external server.

[0052] In step 4, other clients create an entry in their network data packages hash table, indicating the location of the client which will be serving the data package. Thus, preferably only a single client accesses the external server for any given data package.

[0053] If a request is sent for multiple data packages, but a response is received indicating the location of only some of the data packages at a neighboring peer client or clients, the client first obtains these data packages from the neighboring peer client or clients. Next the client then transmits the response message for the rest of the data packages, and proceeds to obtain the remaining data package or data packages from the external server. Thus, the client only obtains the data package or data packages from the external server which are not available locally, rather than obtaining all of the data packages from the external server, thereby reducing network traffic.

[0054] According to preferred embodiments of the present invention, preferably the process of downloading data package from peer clients is optimized to reduce the amount of time required for downloading, the load on each individual client and the overall network traffic. Such optimization is performed as follows.

[0055] First, preferably the *exit degree* of each client is bound, such that each client is only able to serve a fixed, limited number of other clients simultaneously. More preferably, the default limit is three other clients, for example, or some another appropriate number which is preferably configured by the user or by the network administrator. If an additional client attempts to download a data package from a client which is already serving the maximum number of other clients will receive a "busy" message. This feature limits the load on each individual client.

[0056] Also preferably, the present invention is able to optimize the selection of the best client from which the data package should be obtained. For example, if client "A" had already downloaded a larger portion of the required data package than client "B", transferring the data package from client "A" is more optimal. Such clients are preferentially selected to serve data packages, since these clients will be able to serve the data package after a shorter time period has elapsed. Such preferential selection occurs by shortening the time period for waiting before these clients respond, thereby increasing the likelihood that they will serve the data packages. For this reason, the client preferably calculates the random delay before responding such that the delay is inversely proportional to the percentage of the data package which has been already downloaded. In addition, the random delay is preferably proportional to the number of clients being served at the moment, in order to decrease the likelihood of overloading already busy clients.

[0057] In addition, according to other preferred embodiments of the present invention, preferably the entries of the locations of data packages in the network data packages table are updated according to a probabilistic function. Such a function is preferred in order to prevent all of the clients from registering a single client as the server for any particular data package, for example. When different clients respond, usually at different times, indicating they have a specific data package, the remaining clients listening across the network update the entry for this data package in their network data packages table, by adding the IP address, or some other type of address according to the addressing system employed by the network, of the client which can serve the data package to this table. In a simple implementation, the clients would store only the last advertised location of each data package, and therefore many or all clients might attempt to obtain the data package from a single client as the internal server, thereby overloading that client.

[0058] To avoid this situation, preferably the following probabilistic algorithm is used to determine the particular client address which is stored in the network data packages table. Each time a new client transmits a response message, indicating that this client is able to serve a particular data package, the probability that the new IP address of the new client is substituted for the old IP address is calculated according to the following equations:

$$\text{New IP address} = \begin{cases} \text{Old IP address} & Po(x) = 1/(\text{generation}+1) \\ \text{New IP address} & Pn(x) = 1-1/(\text{generation}+1) \end{cases}$$

wherein Pn(x) is the probability that anew IP address is substituted for the old IP address, Po(x) is the probability that the old IP address is retained, and "generation" is a number indicating how many times this address had been previously changed.

[0059] For example, if client "A" responds indicating it has data package "X", then initially all other peer clients store the IP address of client "A" as the location of data package "X". If client "B" then broadcasts a response also indicating that client "B" has data package "X", then the probability that any one client now changes the IP address for the location of data package "X" is 50%. In other words, about half of the clients should now point to client "A" and about half should point to client "B".

[0060] Such a substantially even distribution of load across multiple clients should produce data-flow with a tree-shaped topology, as shown in Figure 3, rather than a random topology, thus optimizing the average download time and the load on the serving clients.

[0061] Furthermore, if any client requests a particular data package during the period required by client"A" for downloading that package, preferably client "A" sends a broadcast or multicast message indicating that the package is in the process of being downloaded. Therefore, preferably only a single client "B" polls client "A" for each data package, for example. Other clients preferably automatically receive any responses from that polling action though the broadcast or multicast transmission, and thus will not be forced to poll for themselves.

[0062] The polling (request/response) traffic is optimized since there is usually no need to transmit both a request and a response for each data package needed by each client. Such optimization is possible since each client preferably receives substantially all of the request/response communication of all the other clients and "remembers" the location of the data packages in the network-data packages table.

[0063] As previously described, the actual process for receiving a data package from an internal server is performed according to the data package exchange protocol, by using the HTTP protocol or some other suitable peer-to-peer communication protocol. The data package transfer software application of the present invention preferably features a timer, for detection of an aborted transfer or a very slow data package transfer, for example. The timer determines when such a transfer has timed out. If a time-out occurs, the requesting client preferably repeats the whole process. If the transfer remains unsuccessful after a plurality of attempts, the client preferably ceases to attempt to transfer the data package from the peer client as the internal server, and instead transfers the data package or data packages directly from the external server.

[0064] Again, as described previously, if a requested data package has not yet finished being downloaded by a peer client the requesting client receives a message indicating that the data package is not ready, as well as an indication of the fraction of the data package already downloaded. The requesting client continues polling the serving client until the data package download is complete. If the download becomes substantially slower or is otherwise interrupted or terminated for a long period of time, the requesting client behaves as if a time-out occurred.

[0065] According to additional preferred features of the present invention, substantially automatic detection of peer clients is supported. Such automatic detection enables each peer client to detect the presence of other peer clients on the network. If such peer clients are not found, preferably the system of the present invention is disabled, since the operation of the system as described above would only prolong the time period required to download a data package if no other peer clients are available.

[0066] Preferably, the amount of bandwidth on the local area network which is consumed by each peer client serving data packages to other clients is limited, to avoid over-burdening any specific host. This limit is preferably configurable by the user or by the network administrator.

[0067] Furthermore, in order to protect peer clients from unauthorized access of local storage media through the system of the present invention, certain security features are preferably included. For example, preferably only data packages identified in the hash tables are able to be transferred from the client. Thus, transmitted data packages are preferably only data packages which were intended to be served to the peer clients, such that malicious users preferably cannot use the system of the present invention to obtain "random" data packages from the storage media of a peer client. Data packages are more preferably only referenced by their unique identifier, such as their 128-bit MD5 digest, such that a data package is only able to be downloaded from a client if the intended recipient knows this digest. Thus, the name of a data package alone is preferably not sufficient information to permit retrieval of the data package from a peer client.

[0068] According to another embodiment of the present invention, the system of the present invention is also applicable to Web browsers, FTP clients, and other software applications involving client-server data-transfer. As described with reference to Figures 4 and 5A-5B, another exemplary embodiment of the present invention is used for caching Web content.

[0069] In step 1 of Figure 4, a Web browser being operated by a client computer requests a specific data package. First the Web browser looks at the local cache, as is known to one of ordinary skill in the art. If the data package is found in the local cache, then that data package is retrieved from the local cache. Otherwise, the Web browser issues a message requesting this data package, preferably by using broadcast or multicast message transmission. The data package is preferably uniquely defined by a unique identifier. More preferably, the unique identifier is the URL of the data package, or alternatively and preferably a combination of the URL of the data package and timestamp, or by any other suitable unique identifier.

[0070] For optimization, if more than one data package is required, the Web browser preferably transmits one request message containing the list of needed data packages, thereby reducing the total network traffic across the network. Such a situation may arise if, for example, the Web browser had just parsed an HTML (hypertext mark-up language) document, or Web page, which contains many links to follow. Preferably and optionally, each request message contains an identifying "magic number", which may contain the protocol version (PVN). For instance: "V1.0". As shown in Figure 5A, the request message includes the list of URL's or other unique identifiers to identify the data package or data packages being requested, which is similar in function to the list of MD5 digests described previously for request messages, and a unique identifier identifying the request message, shown as "REQ".

[0071] In step 2, other Web browsers across the network listen to detect request messages of this type. These Web browsers, which are peer clients for this embodiment of the present invention, receive this request message and check their own cache for the requested URL. If the requested URL is found in the local cache of a Web browser, that Web browser preferably waits a random interval and then preferably transmits a *response* message indicating it has the required data package (or data packages). Preferably, the message is broadcast or multicast. More preferably, that Web browser does not reply if another Web browser had replied first. A reply message is preferably sent by a particular Web browser even if the requested URL is still being downloaded by that Web browser.

[0072] In step 3, if no response to an issued request message is received within a certain amount of time, for example 5 seconds, then the process is preferably timed out. In this case, the Web browser preferably no longer attempts to obtain the URL from another Web browser, and the URL is obtained from the regular Web server using regular HTTP protocol. Before starting to download the data package from the regular Web server, the Web browser preferably transmits a response message indicating that this particular Web browser is downloading the data package.

[0073] On the other hand, if a response message is received, the Web browser obtains the URL from the other Web browser which indicated that it had the URL in the local cache. Preferably, Web browsers across the network record the URLs and the address from which the response message originated for future use, such that these Web browsers would be able to download the URL at a future time without first transmitting the request message.

[0074] Once the Web browser is able to locate a data package on a neighboring Web browser, the Web browser attempts to download the data package. The downloading process is performed with a suitable data-transfer protocol, such as HTTP or FTP. If a time-out or other failure occurs during the processing of data package transfer, the receiving Web browser preferably performs substantially the entire procedure more than once. More preferably, the number of permitted attempts to retry the transfer is configurable. If the process fails after these attempts have been performed, preferably the Web browser transfers the required data package or data packages from the regular Web server.

[0075] According to preferred features of this embodiment of the present invention, data package downloading is well distributed, such that the Web browsers do not obtain a data package from only a single Web browser, but rather obtain the data package from a plurality of Web browsers. Such distribution is maintained as follows.

[0076] First, preferably the number of simultaneous data package transfers from a single Web browser is limited. If this number is exceeded, the Web browser transmits a "busy" message to other Web browsers attempting to transfer the data package. Next, preferably once a Web browser receives a message giving the location of a particular data package, the corresponding entry in the hash table for that data package is not altered every time another response message is received pertaining to this data package. The hash table is preferably altered by subsequent messages in a probabilistic manner, such that the probability that any particular entry is updated to indicate a new location of a data package is equal to 1/(generation+1), where 'generation' counts the number of times a response message was received for that data package.

[0077] For example, if Web browser "A" transmits a response message indicating that data package "X" is on the local cache, then initially all of the neighboring Web browsers have an entry in the hash table indicating that Web browser "A" is the location of data package "X". If Web browser "B" then transmits a response message for data package "X", then each Web browser preferably now alters the entry in the hash table to indicate a new location of data package "X" with a probability of about fifty percent, such that about fifty percent of the Web browsers now have an entry indicating that the data package is available from Web browser "A" and such that about fifty percent of the Web browsers now have an entry indicating that the data package is available from Web browser "B". Thus, a good load distribution can be achieved.

[0078] The random delay (mentioned in step 2 above) chosen by a browser is proportional to the number of currently served browsers, or the number of browsers currently downloading data packages from that browser, and inversely proportional to the amount of the data package already downloaded by it. This way the browsers more eligibl to download from are more likely to be chosen by other browsers to serve these data packages.

[0079] While the invention has been described with respect to a limited number of embodiments, it will be appreciated that many variations, modifications and other applications of the invention may be made.

Claims

1. A method for distributing data packages across a network, the network featuring an external server for serving at least one data package, the external server being a dedicated server, the steps of the method being performed by a data processor, the method comprising the steps of:

   (a) providing a plurality of peer clients attached to the network and providing a list of data packages, said data packages being stored by each of said plurality of peer clients, each data package of said data packages having an entry in said list, said entry indicating a unique identifier for said data package and a location of said data package in at least one of said plurality of peer clients;
   (b) examining said list of data packages by a first peer client to find an entry for a required data package; and
   (c) if said entry for said data package is present on said list of data packages of said first peer client, retrieving said data package from said location at another of said plurality of peer clients according to said entry for said data package.

2. The method of claim 1, wherein said list of data packages is stored on at least said first peer client.

3. The method of claim 2, wherein alternatively said entry for said data package is absent from said list of data packages of said first peer client, the method further comprising the steps of:

   (d) sending a request message for said data package by said first peer client to at least one other peer client; and
   (e) if a response message is received by said first peer client from said at least one other peer client, retrieving said data package from said at least one other peer client by said first peer client.

4. The method of claim 3, the method further comprising the step of:

   (f) altering said list of data packages being stored by at least said first peer client for indicating said location of said data package according to said response message.

5. The method of claim 4, wherein if said response message is not received from said at least one other peer client by said first peer client, the method further comprises the step of:

   (g) obtaining said data package by said first peer client from the external server.

6. The method of claim 5, further comprising the step of sending a response message by said first peer client to said at least one other peer client substantially before said first peer client obtains said data package from the external server.

7. The method of claim 6, wherein said list of data packages is stored on each of said plurality of peer clients, the method further comprising the steps of:

   (h) receiving said response message from said first peer client by said at least one other peer client; and
   (i) altering said list of data packages being stored by said at least one other peer client for indicating said location of said data package according to said response message.

8. The method of claim 5, wherein said list of data packages is stored on each of said plurality of peer clients, the method further comprising the steps of:

   (h) receiving said response message from said first peer client by said at least one other peer client; and
   (i) altering said list of data packages being stored by said at least one other peer client for indicating said location of said data package according to a probabilistic function.

9. The method of claim 1, wherein an upper limit is predetermined for a number of said plurality of peer clients served substantially simultaneously by said at least one other peer client, such that if a number of said plurality of peer clients served substantially simultaneously by said at least one other peer client is greater than said upper limit, the method further comprises the step of:

   (d) sending a busy message from said at least one other peer client to said first peer client.

10. The method of claim 1, wherein the external server is a BackWeb™ server, and said plurality of peer clients is a plurality of BackWeb™ clients.

11. A system for distributing data packages across a network according to a list of the data packages, the system comprising:

   (a) an external server for serving at least one data package, said external server being attached to the network; and
   (b) a plurality of peer clients attached to the network, the data packages being stored by each of said plurality of peer clients, each data package of said data packages having an entry in the list, said entry indicating a unique identifier for said data package and a location of said data package in at least one of said plurality of peer clients, such that each peer client retrieves a data package according to the list, each peer client first attempting to retrieve said data package from another of said plurality of peer clients, and alternatively retrieving said data package from said external server if said data packag is not available from another of said plurality of peer clients.

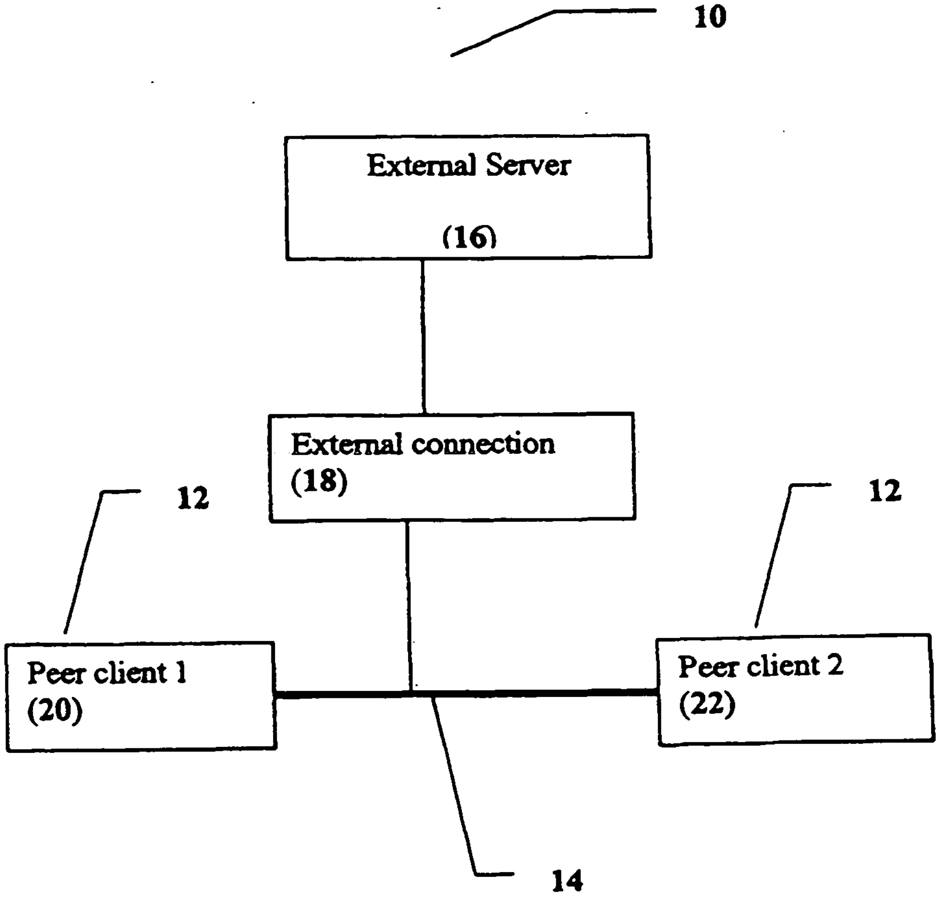

10
External Server
(16)
External connection
(18)
12
12
Peer client 1
(20)
Peer client 2
(22)
14

Fig. 1A

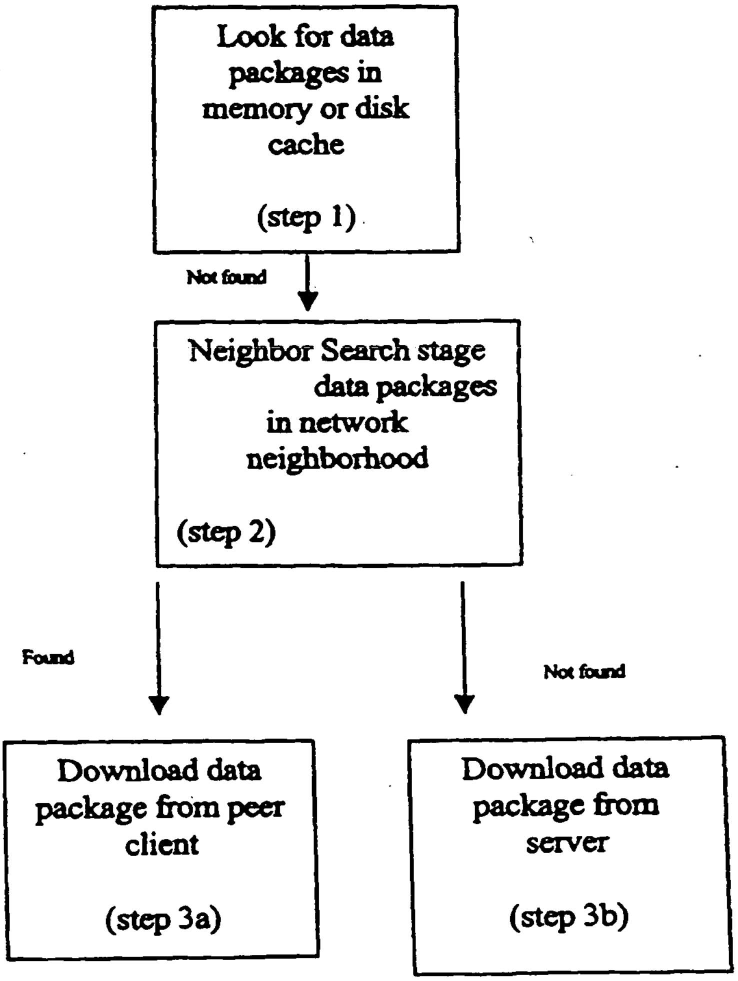

Look for data packages in memory or disk cache
(step 1)
Not found
Neighbor Search stage data packages in network neighborhood
(step 2)
Found
Not found
Download data package from peer client
(step 3a)
Download data package from server
(step 3b)

Fig. 1B

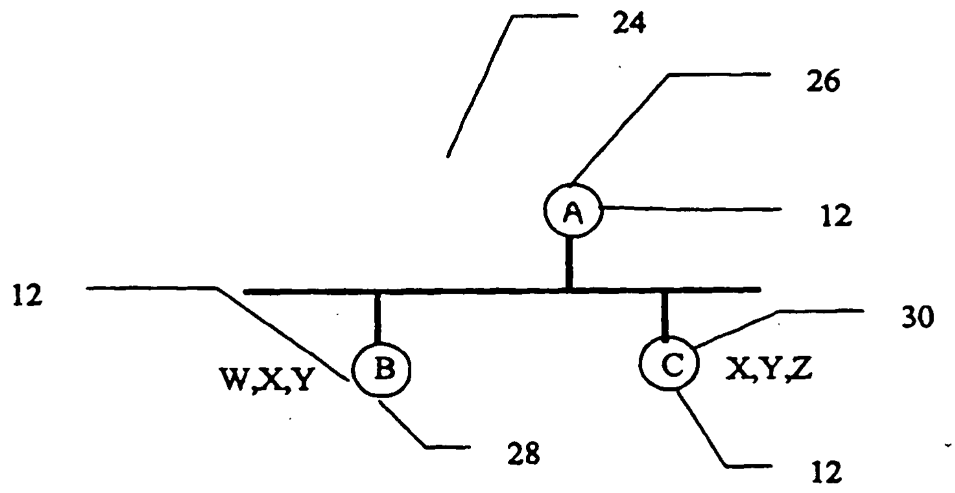


Fig. 2A

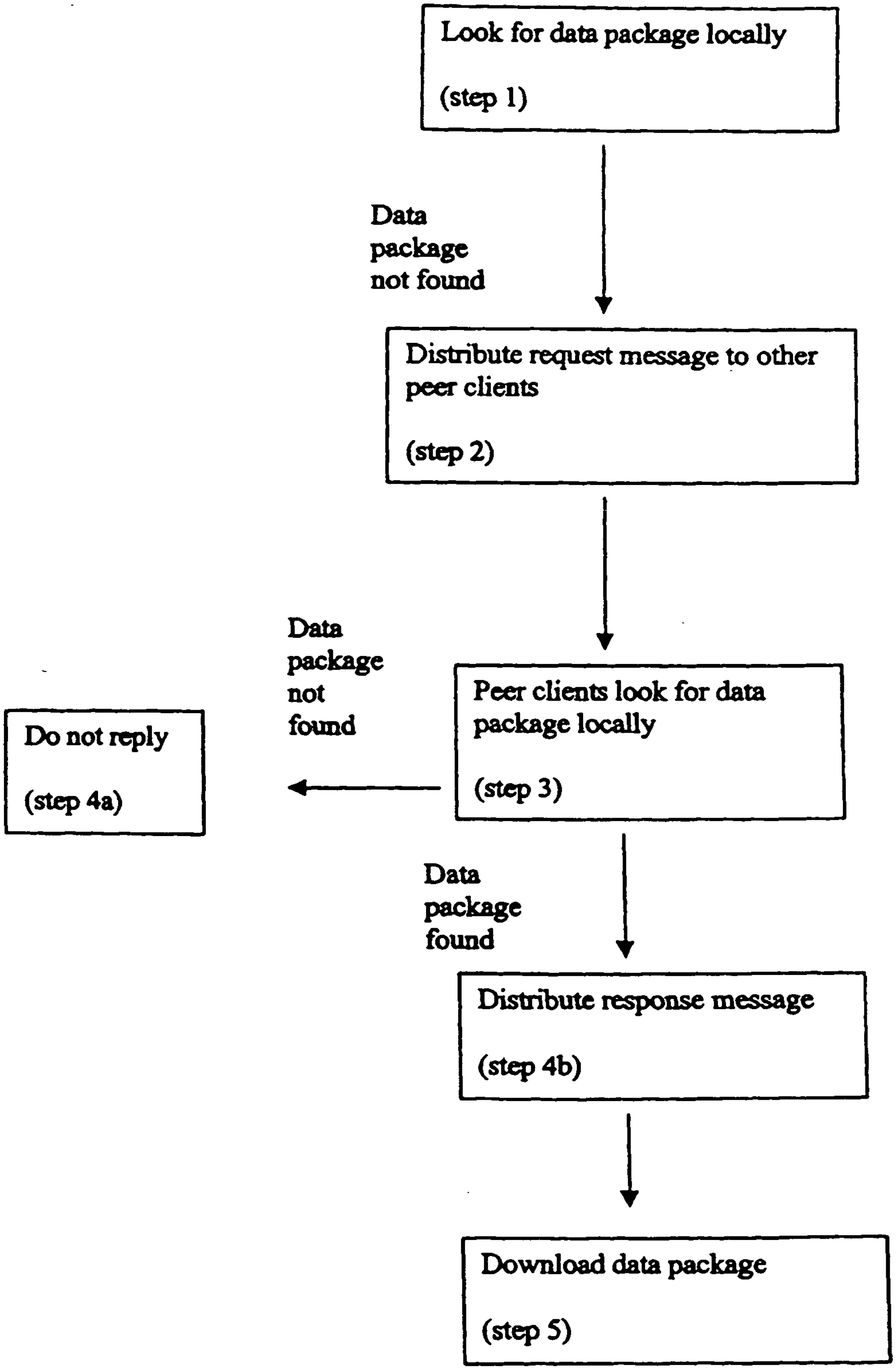

Look for data package locally
(step 1)
Data package not found
Distribute request message to other peer clients
(step 2)
Data package not found
Do not reply
(step 4a)
Peer clients look for data package locally
(step 3)
Data package found
Distribute response message
(step 4b)
Download data package
(step 5)

Fig. 2B

PVN	Digest1, Digest2,...,

Fig. 2C

PVN	Digest1, Digest2,..., DigestN	TCP Port

Fig. 2D

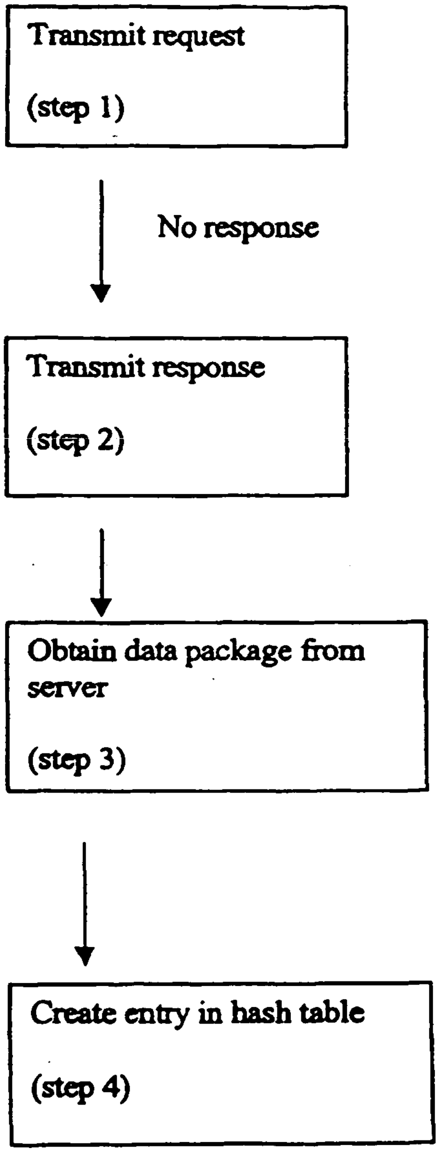

Transmit request
(step 1)
No response
Transmit response
(step 2)
Obtain data package from server
(step 3)
Create entry in hash table
(step 4)

Fig. 2E

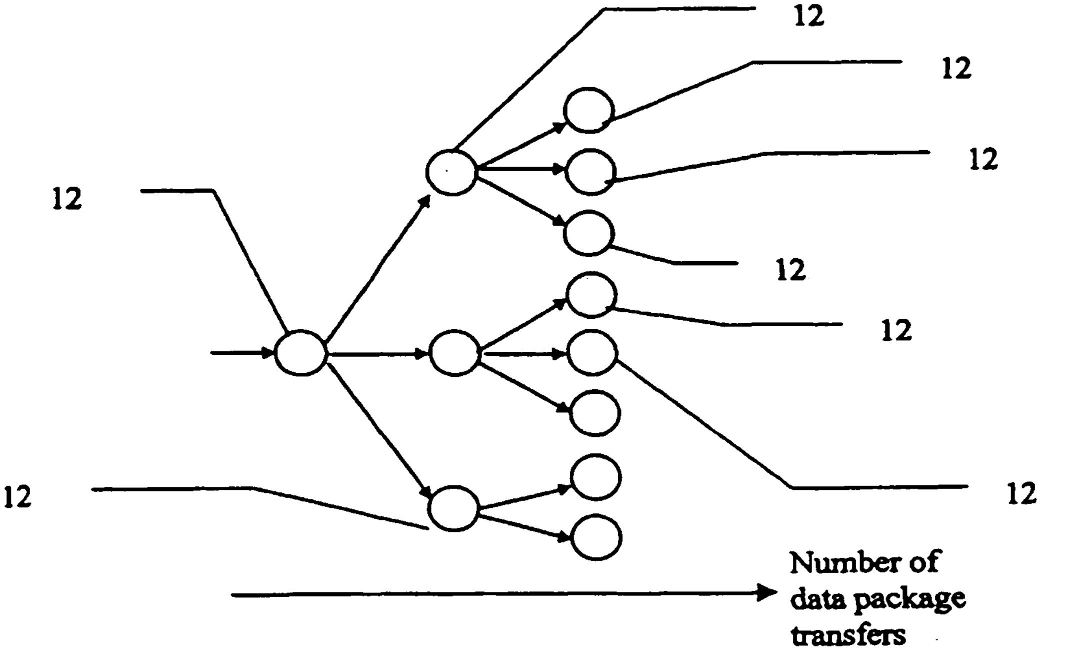


Fig. 3

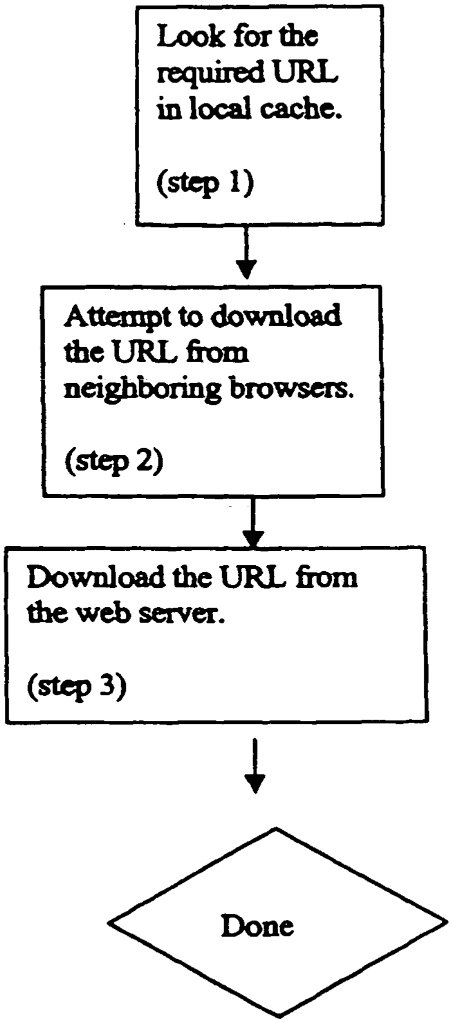

Look for the required URL in local cache.
(step 1)
Attempt to download the URL from neighboring browsers.
(step 2)
Download the URL from the web server.
(step 3)
Done

Fig. 4

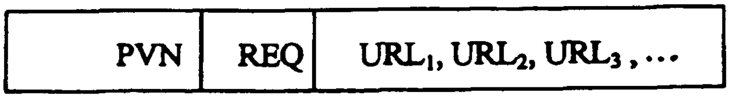


Fig. 5A

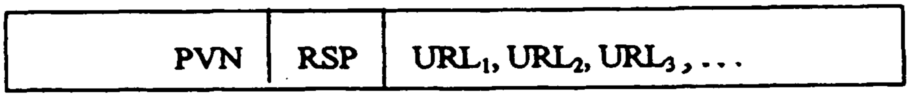


Fig. 5B

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 99 20 3169

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
X	EP 0 837 584 A (AT & T CORP) 22 April 1998 (1998-04-22) * abstract *	1,2,11	H04L29/06 G06F17/30
Y	* column 1, line 21 - column 3, line 19 * * column 10, line 35 - line 41 *	3-7	
A	* figure 1 * ---	8-10	
Y	GB 2 294 132 A (MARCONI GEC LTD) 17 April 1996 (1996-04-17) * abstract * * page 1, paragraph 5 - page 2, paragraph 3 * * page 5, line 16 - page 6, line 30 *	3-7	
A	* page 12, paragraph 4 * ---	8-10	
A	US 5 249 290 A (HEIZER ISAAC J) 28 September 1993 (1993-09-28) * column 5, line 66 - column 6, line 2 * ---	9	
P,A	US 5 864 854 A (BOYLE DOUGLAS B) 26 January 1999 (1999-01-26) * abstract * * column 4, line 25 - line 57 * -----	8	**TECHNICAL FIELDS SEARCHED (Int.Cl.7)** G06G G06F H04L

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	31 January 2000	Blanco Cardona, P

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 99 20 3169

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

31-01-2000

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0837584 A	22-04-1998	US 5787470 A CA 2216346 A JP 10254753 A	28-07-1998 18-04-1998 25-09-1998
GB 2294132 A	17-04-1996	NONE	
US 5249290 A	28-09-1993	NONE	
US 5864854 A	26-01-1999	NONE	

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

THIS PAGE BLANK (USPTO)

THIS PAGE BLANK (USPTO)

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :
G06F 17/30, 12/08

A1

(11) International Publication Number: **WO 98/22891**

(43) International Publication Date: 28 May 1998 (28.05.98)

(21) International Application Number: PCT/US97/21459

(22) International Filing Date: 21 November 1997 (21.11.97)

(30) Priority Data:

08/754,481	22 November 1996 (22.11.96)	US
08/827,534	28 March 1997 (28.03.97)	US
08/848,971	2 May 1997 (02.05.97)	US

(71) Applicant: MANGOSOFT CORPORATION [US/US]; Suite 190, 1500 West Park Drive, Westborough, MA 01581 (US).

(72) Inventors: CARTER, John, B.; 414 South Douglas Street, Salt Lake City, UT 84102 (US). DAVIS, Scott, H.; 136 Riverbend Road, Groton, MA 01450 (US). DIETTERICH, Daniel, J.; 4 Cedar Terrac, Acton, MA 01720 (US). FRANK, Steven, J.; 6 Tiffany Trail, Hopkinton, MA 01748 (US). LEE, Hsin, H.; 7 Francine Road, Acton, MA 01720 (US).

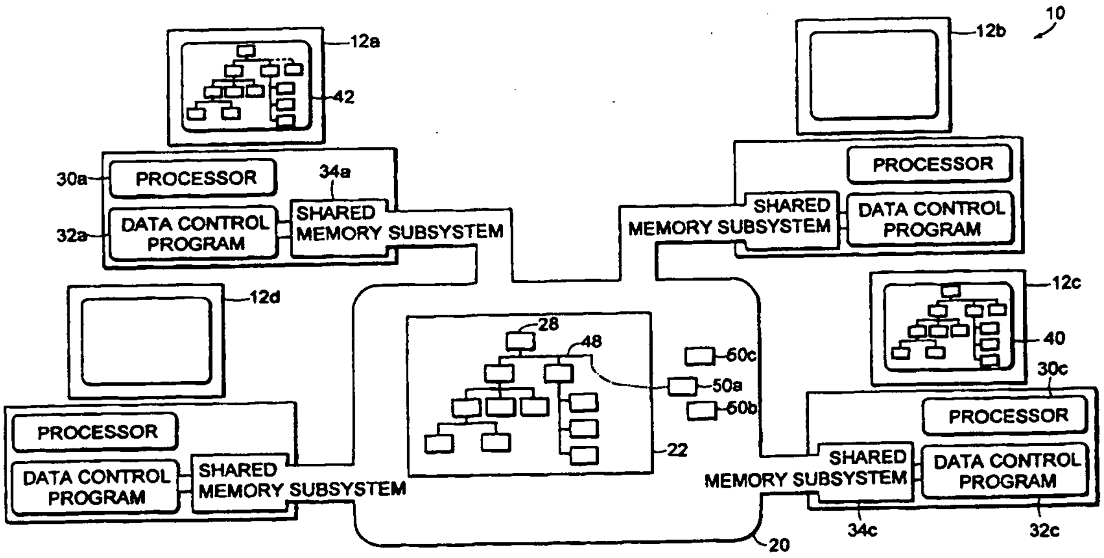
(74) Agent: LANZA, John, D.; Testa, Hurwitz & Thibeault, LLP, High Street Tower, 125 High Street, Boston, MA 02110 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published

With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: SHARED CLIENT-SIDE WEB CACHING USING GLOBALLY ADDRESSABLE MEMORY



(57) Abstract

A shared client-side Web cache is provided by implementing a file system shared between nodes. Each browser application stores cached data in files stored in a globally addressable data store. Since the file system is a shared one, the client-side Web caches are also shared.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

SHARED CLIENT-SIDE WEB CACHING USING GLOBALLY ADDRESSABLE MEMORY

Cross-Reference to Related Application

This application is a continuation-in-part of co-pending U.S. patent applications serial number 08/754,481, filed November 22, 1996, and serial number 08/827,534, filed March 28, 1997 and bearing attorney docket number CLC-002, both of which are incorporated herein by reference.

Technical Field

The present invention relates in general to a shared client-side Web cache and, more specifically, to a shared client-side Web cache maintained within a globally addressable memory.

Background Information

Computer based structured storage systems, such as computer file systems and database systems, have been remarkably successful at providing users with quick and facile access to enormous amounts of data. The importance of these structured storage systems in today's commerce is difficult to exaggerate. For example, structured storage systems have allowed businesses to generate and maintain enormous stores of persistent data that the company can modify and update over the course of years. For many companies, this persistent data is a valuable capital asset that is employed each day to perform the company's core operations. The data can be, for example, computer files (e.g., source code, wordprocessing documents, etc.), database records and information (e.g., information on employees, customers, and/or products), or information and data downloaded from the Internet.

A typical computer-based structured storage system for data downloaded from the Internet is simply a file system on a user's terminal, that is, the user's PC, PC-Compatible, Apple Macintosh, or other workstation on which the user executes a program allowing access to the Internet. In these systems, downloaded data is cached in the file system on the terminal.

While this eliminates delays for the user when retrieving data previously accessed by the user, users do not benefit from the caches of other users.

This drawback is partially solved by inserting a central server, commonly referred to as a proxy server, between a group of users and target data to be downloaded. The proxy server provides a degree of sharing between individual users caches, because the proxy server caches data accessed by the entire group of users. Stored data may include cached copies of Web pages, image files, JAVA applets, and ActiveX controls. The proxy server manages each user's Internet connection and, as mentioned above, provides a degree of data caching between users of the system, since the proxy server may cache data downloaded by a first user. If that data remains in the proxy server's cache when a second user requests it, the proxy server can supply the data to the second user.

Success of proxy servers has been limited by drawbacks associated with their centralized nature. The centralized cache technique suffers from several drawbacks. The first is that providing a centralized cache having storage capability equal to the sum of each user's individual Web cache is expensive. Another drawback from which centralized Web caches suffer stems, to some extent, from the first drawback mentioned above. Because the size of a centralized Web cache is generally smaller than the size of the sum of all of the individual users' Web caches, data may be swapped out of the centralized Web cache even though it is cached locally in an individual user's Web cache. This can lead to situations in which a second user accesses, for example, a Web page previously accessed by a first user. Although the Web page still resides in the first user's Web cache, it has been removed from the centralized Web cache due to space constraints. As a result, the second user must then fetch the Web page over the Internet, incurring delays associated with the Internet transfer.

Yet a third drawback of these types of systems is that the proxy server introduces additional delays due to processing constraints. That is, if the proxy server is fetching data for a first user, a data fetch for a second user must wait for the proxy server to complete the data fetch for the first user. Also, any failure of the proxy server to maintain proper operation, such as a power failure, hardware failure, or other such system failure, will disable the individual user's ability to access the Internet or cause the shared data to be lost.

Another attempt to optimize Web access involves prefetching Web pages. For example, when a user downloads a Web page, the system may also download Web pages which are linked to that page. Should the user traverse a link to one of the prefetched Web pages, the system is able to display the new page to the user with little or no delay. These systems, however, do not allow users to share Web caches and have an additional drawback that network traffic is increased. For example, these systems must prefetch multiple pages to provide an end-user with quick access to only one page, and the user may select a page that is not linked to the previous page viewed by the user that, therefore, has not been prefetched.

Summary of the Invention

The present invention relates to a system and method that provides a shared client-side Web cache. The Web cache is shared by a particular group of users. The shared Web cache is stored in a globally addressable data store that allows each user to access the shared Web cache as if it were accessing a traditional, local Web cache. Since the globally addressable data store is distributed over each user's terminal, the size of the cache can increase with each user that is added to the system. Also, since data is replicated on a user's local terminal whenever the user accesses it, data cached in the shared Web cache is automatically replicated, providing a degree of fault tolerance and also insuring that data is stored on the nodes most likely to use the data. The shared cache benefits client users by allowing them to access cached Web pages regardless of whether the pages were cached by one or more different users, and regardless of the method used to load the page into a cache, i.e. whether the page was prefetched or cached in a traditional manner.

In one aspect, the invention relates to a method for locally caching and sharing downloaded data between a plurality of client nodes. The method begins by providing a number of client nodes that are interconnected by a network. Each node hosts a control program, in some embodiments as an Internet browser or file transfer program, that stores and retrieves cached data. Each control program accesses a globally addressable data store which provides persistent storage of data, and each instance of the control program employs the data store as a memory device for storing and retrieving cached data. The data cached by the control program instantiations can include any type of downloaded data. In some embodiments, the downloaded data represents HTML files, fragments of HTML files, ActiveX controls, JAVA applets, or image files. In one particular embodiment, the globally addressable data store is a distributed shared memory system.

In another aspect, the invention relates to a system for caching and sharing downloaded data. The system includes one or more client nodes and a network. The client nodes are each connected to the network and each client node includes a control program that manipulates caches of downloaded data. Each instance of the control program accesses a globally addressable data store formed by the memory elements present on each individual node. Each node stores and retrieves data from the globally addressable data store.

Brief Description of the Drawings

This invention is pointed out with particularity in the appended claims. The above and further advantages of this invention may be better understood by referring to the following description taken in conjunction with the accompanying drawings, in which:

FIG. 1 is a conceptual block diagram of a distributed addressable shared memory structured data storage system according to the invention.

FIG. 2 is a diagram of one possible embodiment of the system of FIG. 1, namely a distributed addressable shared memory file system providing storage for computer files such as source code files, wordprocessing documents files, etc.

FIG. 3 is a graphical representation of the organization of directory entries and associated file descriptors (also known as "Inodes"), suitable for use with the file system of FIG. 2.

FIG. 4 is a diagram of an Inode suitable for use with the file system of FIG. 2.

FIG. 5 illustrates a distributed shared memory computer network.

FIG. 6 is a functional block diagram that illustrates in more detail one distributed shared memory computer network of the type shown in FIG. 5.

FIG. 7 illustrates in more detail a shared memory subsystem suitable for practice with the network illustrated in FIG. 6.

FIG. 8 is a functional block diagram of one shared memory subsystem according to the invention.

FIG. 9 illustrates a directory page that can be provided by a shared memory subsystem of the type depicted in FIG. 8.

FIG. 10 illustrates a directory that can be distributed within a shared memory and formed of directory pages of the type illustrated in FIG. 9.

FIG. 11 illustrates in functional block diagram form a system that employs a directory according to FIG. 10 for tracking portions of a distributed shared memory.

FIG. 12 depicts a typical system for accessing data using the Internet.

Description

A network system 10 according to the invention includes a plurality of network nodes that access a memory space storing a structured store of data, such as a structured file system or a database. Each of the nodes includes at least a data control program which accesses and manages the structured store of data. The structured store of data may be stored in an addressable shared memory or the structured store may be stored in a more traditional fashion. For example, each node may be responsible for storing a particular element or elements of the structured store of data. In such an embodiment, the data control program can access a desired portion of the structured store using a globally unique identifier. The underlying system would translate the identifier into one or more commands for accessing the desired data, including network transfer commands. In another embodiment, the structured store of data is stored in an addressable shared memory space, which allows the nodes to transparently access portions of the structured store using standard memory access commands.

The system 10 can be a file system, a database system, a Web server, an object repository system, or any other structured storage system that maintains an organized set of data. In one disclosed embodiment, the system 10 is a file system that maintains various computer files. However, this is just one embodiment of the invention that is provided for illustrative purposes. The invention can be employed to provide any one of a plurality of structured storage systems (e.g., database system, Web page system, Intranet, etc.). The invention is not to be limited to the file system or other particular embodiments described herein.

Referring to FIG. 1, a network system 10 according to the invention includes a plurality of network nodes 12a-12d and an addressable shared memory space 20 that has a portion 22 for storing a structured store of data 28. Each of the nodes 12a-12d can include several sub-elements. For example, node 12a includes a processor 30a, a data control program 32a, and a shared memory subsystem 34a. In the disclosed embodiment, two of the nodes, 12a and 12c, include monitors that provide displays 40 and 42 graphically depicting the structured store of data 28 within the addressable shared memory space 20. The addressable shared memory space 20 interconnects each of the network nodes 12a-12d and provides each node 12a-12d with access to the structured store of data 28 contained within the addressable shared memory space 20.

A system 10 according to the invention can provide, among other things, each network node 12a-12d with shared control over the structured store of data 28 and, therefore, the system 10 can distribute control of the data store across the nodes of the network. To this end, each node of the system 10, such as node 12a, includes a data control program 32a that interfaces to a shared memory subsystem 34a. The data control program 32a can operate as a structured storage system, such as a file system, that is adapted to maintain a structured store of data and to employ the shared memory system as an addressable memory device that can store a structured store of data. At the direction of the data control program 32a, the shared memory subsystem 34a can access and store data within the addressable shared memory space 20. These cooperating elements provide a structured storage system that has a distributed architecture and thereby achieves greater fault tolerance, reliability, and flexibility than known structured storage systems that rely on centralized control and centralized servers. Accordingly, the invention can provide computer networks with distributively controlled and readily scaled file systems, database systems, Web page systems, object repositories, data caching systems, or any other structured storage system.

Still referring to FIG. 1, the system 10 of the invention maintains within the addressable shared memory space 20 a structured store of data 28. Each of the nodes 12a-12d can access the addressable shared memory space 20 through the shared memory subsystems 34a-34d. Each of the shared memory subsystems 34a-34d provides its node with access to the addressable shared memory space 20. The shared memory subsystems 34a-34d coordinate each of the respective node's memory access operations to provide access to the desired data and maintain data coherency within the addressable shared memory space 20. This allows the interconnected nodes 12a-12d to employ the addressable shared memory space 20 as a space for storing and retrieving data. At least a portion of the addressable shared memory space 20 is supported by a physical memory system that provides persistent storage of data. For example, a portion of the addressable shared memory space 20 can be assigned or mapped to one or more hard disk drives that are on the network or associated with one or more of the network nodes 12a-12d as local hard disk storage for those particular nodes. Accordingly, FIG. 1 illustrates that systems of the invention have shared memory subsystems providing the network nodes with access to an addressable shared memory space, wherein at least a portion of that space is assigned to at least a portion of one or more of the persistent storage memory devices (e.g., hard disks) to allow the nodes addressably to store and retrieve data to and from the one or more persistent storage memory devices. A preferred embodiment of such an addressable shared memory space is described in the commonly-owned U.S. patent application serial number 08/754,481 filed November 22, 1996, and incorporated by reference above.

Therefore, one realization of the present invention is that each of the nodes 12a-12d can employ its respective shared memory subsystem as a memory device that provides persistent data storage.

Each of the data control programs 32a-32d is a software module that couples to the respective shared memory subsystem 34a-34d in a way that operates similarly to an interface between a conventional data storage program and a local memory device. For example, the data control program 32a can stream data to, and collect data from, the shared memory subsystem 34a. Because the shared memory subsystems coordinate the memory accesses to the addressable shared memory space 20, each of the data control programs is relieved from having to manage and coordinate its activities with the other data control programs on the network or from having to manage and coordinate its activities with one or more central servers. Accordingly, each of the data control programs 32a-32d can be a peer incarnation (i.e., an instance) residing on a different one of the network nodes 12a-12d and can treat the respective shared memory subsystem 34a-34d as a local memory device such as a local hard disk.

One or more of the data control programs 32a-32d can provide a graphical user interface 42 that graphically depicts the structured store of data 28 contained within the addressable shared memory space 20. The graphical user interface 42 allows a user at a node, for example at node 12a, to insert data objects graphically within the structured store of data 28. To this end, the data control program 32a can generate a set of commands that will present a stream of data to the shared memory subsystem 34a and the shared memory subsystem 34a will employ the data stream to store an object within the structured store of data 28. Similarly, the other shared memory subsystems 34b-34d can provide information to their respective nodes that is indicative of this change to the structured store of data 28. Accordingly, as shown depicted in FIG. 1 for node 12c only for simplicity, that node (which includes a graphical user interface 40) reflects the change to the structured store of data 28 affected by the data control program 32a of the node 12a. In particular, the graphical user interface 40 of the node 12c can depict to a user that an object is being placed within the structured store of data 28. For example, the addressable shared memory space 20 also contains the data objects 50a-50c which can be placed within the structured data store 28 to become part of that structured data store. As illustrated, a system user at node 12a can direct object 50a to be inserted at a set location within the data store 28. The data control program 32a then directs the shared memory subsystem 34a to place the object 50a within the data store 28 at the proper location. Moreover, the shared memory subsystem 34c on node 12c detects the change within the data store 28 and reflects that change within the graphical user interface 40.

Referring now to FIG. 2, a structured file system 60 is a particular embodiment according to the invention that employs the properties of the addressable shared memory space 20 to implement what looks to all network nodes like a coherent, single file system when in fact it spans all network nodes coupled to the addressable shared memory space 20.

The file system 60 of FIG. 2 differs from known physical and distributed file systems in a variety of ways. In contrast to known physical file systems which map a file organization onto disk blocks, the file system 60 according to the invention manages the mapping of a directory and file structure onto a distributed addressable shared memory system 20 which has at least a portion of its addressable space mapped or assigned to at least a portion of one or more persistent storage devices (e.g., hard disks) on the network. Unlike known distributed file systems, the file system 60 of the invention employs peer nodes, each of which have an incarnation or instance of the same data control program. Also, unlike known file systems generally, the file system 60 of the invention: maintains data coherence among network nodes; automatically replicates data for redundancy and fault tolerance; automatically and dynamically migrates data to account for varying network usage and traffic patterns; and provides a variety of other advantages and advances, some of which are disclosed in the commonly-owned U.S. patent application serial number 08/754,481 filed November 22, 1996, and incorporated by reference above.

Still referring to FIG. 2, the file system 60 resides in part within the addressable shared memory space 20, and includes a structured store of data 62, a super root 64, file sets 66-74, directory entry 80, and file or document 82. Two network nodes 84 and 86 are shown accessing the addressable shared memory space 20 (in the manner described previously with reference to FIG. 1) via the logical drives 90 and 94. Application programs 92 and 96 executing on the nodes interact with the data control programs (not shown in FIG. 2 but shown in FIG. 1 as 32a-32d) and cause the data control programs in the nodes to access the logical drives 90 and 94. In the disclosed embodiment, the logical drives are DOS devices that "connect to" the fileset directories via Installable File System drivers associated with the file system 60.

The file system 60 supports one global file system per addressable shared memory space 20 shared by all of the network nodes. This global file system is organized into one or more independent collections of files, depicted as the filesets 66-74. A fileset can be thought as logically equivalent to a traditional file system partition. It is a collection of files organized hierarchically as a directory tree structure rooted in a root directory. The non-leaf nodes in the tree are the directories 80, and the leaves in the tree are regular files 82 or empty directories. Sub-directory trees within a fileset can overlap by linking a file to multiple directories.

A benefit of breaking up the file system 60 into filesets 66-74 is that it provides more flexible file system management for users of the system 60. As the file system 60 grows into very large sizes (e.g., hundreds of nodes with thousands of gigabits of storage), it is desirable to have the files organized into groups of management entities such that management actions can be independently applied to individual groups without affecting the operation of the others.

The filesets in the addressable shared memory space 20 are described and enumerated in a common structure, the root 64 of which provides the starting point to locate the filesets in the addressable shared memory space 20. The root 64 can be stored in a static and well-known memory location in the addressable shared memory space 20, and it can be accessed via a distributed shared memory system program interface. When a node is accessing a fileset for the first time, it first looks up the root 64 to determine the identifier associated with the fileset, e.g., the shared memory address used to access the fileset. Once it has determined the identifier, the node can access the root directory of the fileset. From the root directory, it then can traverse the entire fileset directory tree to locate the desired file. Filesets used by the file system 60 are described in greater detail below under the heading "Fileset."

Referring to FIG. 3, in the disclosed embodiment of the file system 60 according to the invention, a directory 126 (such as the directory 80 of FIG. 2) is accessed by starting at a directory Inode or descriptor 128 containing an address that points to a directory entries stream descriptor 130. This descriptor 130 is a pointer to a block of data containing directory entries for files File 1 through File 3. The directory entry for File 1 has a number of entries; one of the entries is a string containing the name of the file and another entry is the address of the Inodes and stream descriptors 132. The stream descriptors for File 1 are used to locate and retrieve the various 4 kilobyte pages in the addressable shared memory space 20 that constitute File 1. Other files are retrieved and constructed from the addressable shared memory space 20 in the same fashion. The directories used by the file system 60 are described in greater detail below under the heading "Directory."

In the embodiment of the file system 60 disclosed in FIG. 4, a file 98 (such as the file 82 of FIG. 2) is represented by one or more shared pages of data 100, 102, 104, 106, and 108 in the addressable shared memory space 20. Each file 98 has a file Inode or descriptor 110 that includes various file attributes 112. The file descriptor 110 contains an address that points to a data stream descriptor 114, and the data stream itself includes one or more addresses 116, 118, 120, 122, and 124 that point to particular pages in the virtual addressable shared memory space 20. In the disclosed embodiment, a page is the atomic unit in the addressable shared memory space 20, and it contains up to 4 kilobytes of data. Even if the entire 4 kbytes is not needed, an entire page is used. This is illustrated by the page 108 that only contains about 2 kbytes of data. The files used by the file system 60 are described in greater detail below under the heading "Files."

FILESET

The filesets are the basic unit for the file system 60. Each fileset is identified with a name having up to 255 characters. The file system 60 exports a set of fileset level operations that allow an administrator to manage the filesets through the following type of actions.

Fileset Creation

This operation creates a new fileset. The fileset is initially created with one file, the empty root directory. A default fileset is created automatically at the initialization of the addressable shared memory space 20.

Fileset Deletion

This operation deletes a fileset. All files in the fileset are removed, and all shared memory space allocated to the files in the fileset is discarded and the backing physical storage freed for new storage. The file system 60 will only allow deletion of a fileset until there are no open handles to file data stream in the fileset. In order to ready a fileset for deletion, the fileset must be "shutdown" by putting it off-line.

Fileset Enumeration

This operation enumerates a specific fileset, or all the filesets, in the addressable shared memory space 20.

Fileset Control

This operation performs fileset level control routines such as setting fileset attributes.

Mount Export Control

Directory are attached to local devices, i.e. "mounted" using parameters stored in the Windows NT registry, or some other similar central storage area for such information. When first started up, the data control program 60 accesses the central storage and determines which filesets should be mounted. The data control program creates a file object representing each fileset identified by the entries in the central storage. In some embodiments an API may be provided which allows the data control program 60 to dynamically mount and unmount filesets by making appropriate API calls.

The users of the file system 60 are not aware of the shared memory "logical volume," but rather view each fileset as a volume (or partition in the sense of a traditional physical file system). The Win32 GetVolumeInformation is used to get information on the fileset (more precisely, on the logical device on which the fileset is attached to). Because all the filesets share the same pool of the storage in the addressable shared memory space 20, the total volume size returned to the user for each fileset is the current aggregate storage capacity in the addressable shared memory space 20. The same approach is taken for the total free space information, and the aggregate value of the addressable shared memory space 20 is returned for each fileset.

DIRECTORY

Directory entry scanning is one of the most frequently performed operations by user applications. It is also may be the most visible operation in terms of performance. Consequently, much attention is directed to making the directory scan efficient and the WindowsNT Files System (NTFS) duplicates sufficient file Inode information in the directory entry such that a read directory operation can be satisfied by scanning and reading the directory entries without going out to read the information from the file Inodes. The problem with this scheme is that the doubly stored file metadata, such as the file time stamps and file size, can be updated quite frequently, making the metadata update more expensive. However, this overhead is considered acceptable in face of the performance gained in directory scan operations.

The file system 60 adopts the same philosophy of providing efficient directory scanning by duplicating file Inode information in directory entries. Each directory entry contains sufficient information to satisfy the Win32 query file information requests. The file Inode is stored with the file stream descriptors on a separate page. The Inode is located via a pointer in the directory entry.

The file system's directory entries are stored in the directory file's directory entry data stream. To maximize space utilization, each directory entry is allocated on the first available free space in a page that can hold the entire entry. The length of the entry varies depending on the length of the file's primary name. The following information is part of the directory entry: creation time; change time; last write time; last accessed time; pointers to stream descriptor; pointer to parent directory Inode; MS-DOS type file attributes; and MS-DOS style file name (8.3 naming convention). For average file name lengths, a page contains up to about 30 entries. All the file information in the directory entry is also contained in the file Inode, except for the file primary name and MS-DOS file name. The file primary names and associated short names are only stored in the directory entries. This makes the Inode size fixed.

When a file information is modified (except for file names), the Inode is updated in the context of the update transaction and therefore always contains the most up-to-date information. The associated directory entry change is lazily flushed to reduce the cost of double updating. This means the Inode updates are either flushed or recoverable, but not the corresponding directory entry updates. If the directory entry gets out of synch with the Inode (when the Inode change is successfully flushed but not the directory change), the entry is updated the next time the Inode is updated. In order to facilitate synchronization of directory updates, the directory entries (Inodes) can not span multiple pages. FIG. 3 illustrates the organization of directory entries and associated Inodes.

FILES

A file of the file system 60 comprises streams of data and the file system metadata to describe the file. Files are described in the file system 60 by objects called Inodes. The Inode is a data structure that stores the file metadata. It represents the file in the file system 60.

A data stream is a logically contiguous stream of bytes. It can be the data stored by applications or the internal information stored by the file system 60. The data streams are mapped onto pages allocated from the addressable shared memory space 20 for storage. The file system 60 segments a data stream into a sequence of 4 kilobyte segments, each segment corresponding to a page. The file system 60 maintains two pieces of size information per data stream: the number of bytes in the data stream; and the allocation size in number of pages. The byte-stream to segment/page mapping information is part of the file metadata and is stored in a structure called data stream descriptor. See FIG. 4.

Users' requests for data are specified in terms of range of bytes and the position of the starting byte measured by its offset from the beginning of the data stream, byte position zero. The file system 60 maps the offset into the page containing the starting byte and the intra-page offset from the beginning of the page.

Every file of the file system 60 has at least two data streams: the default data stream; and the Access Control List (ACL) stream. Each file may optionally have other data streams. The ACL stream is used to store the security Access Control Lists set on the file. Each data stream is individually named so that the user can create or open access to a specific data stream. The name of the default data stream is assumed to be the primary name of the file. To access a data stream, the user of the file system 60 must first open a file handle to the desired data stream by name. If the file name is used then the handle to the default data stream is opened. This open file handle represents the data stream in all the file system services that operates on the data stream.

The file system 60 exports a set of services to operate at the file level. The input to the services are the file object handle (Inode) or the data stream object handle, and the operation specific parameters, including the desired portions of the data stream in byte positions.

Open files are represented by data stream objects (or just file objects). Users access files using these file objects, identified to the users through file handles. A file handle is a 32-bit entity representing an instance of an open file stream. For example, WindowsNT creates the file object and returns a file handle to the users in response to the user request for file creation or file open. The file system 60 initializes a pointer to a file control block. Multiple file objects point to the same file control block and each file control block maintains separate stream objects for each open context. Externally, the file handle is opaque to the users. Multiple opens can be issued against the same file. When the user closes a file, the file object and the associated file handle is removed.

The file system 60 maps file streams into sequences of segments which become progressively larger; each segment corresponds to one or more pages. The file system 60 attempts to reserve contiguous pages for data streams but only allocates real backing storage on an as needed basis, usually as a result of a file extension requested by writing beyond the data stream allocation size. When a file extension request is received, the file system 60 rounds the extension size in number of bytes up to a multiple of 4 kilobytes to make it an integer number of pages, and requests pages for actual allocation. The number of 4 kilobyte pages allocated by the file system depends on the number of file extension requests made. The file system 60 allocate one 4 kilobyte page for the first extension request, two 4 kilobyte pages for the second request, four 4 kilobyte pages for the third extension request, and so on. The newly allocated pages are zero filled. By reserving contiguous pages, the file system 60 can reduce the amount of bookkeeping information on the byte offset to page mapping. The file system 60 reserves (sometimes much) larger than requested memory space for a file, and substantiates the storage by allocating backing storage page by page.

Four kilobyte allocation segments are chosen to reduce the unused storage space and yet provide a reasonable allocation size for usual file extensions. Since allocation is an expensive operation (most likely involving distributed operations), smaller allocation size is not efficient. Larger allocation size would lead to inefficient space utilization, or additional complexity to manage unused space. A 4 kilobyte segment also maps naturally to a page, simplifying the data stream segment to page mapping. Although an analogy could be made with the NTFS's allocation policy of 4 kilobyte clusters (segment) size for large disks to speed up allocation and reduce fragmentation, such analogy is not completely valid because the actual on-disk allocation segment size depends greatly on the local disk size and the physical file systems.

Similar to the NTFS, which controls the allocation of each disk partition and therefore can quickly determine the free volume space available for allocation, the file system 60 requests the total available space information and uses this information to quickly determine whether to proceed with the allocation processing. If the total available space is less than the required allocation size, the request is denied immediately. Otherwise, the file system 60 will proceed to allocate the pages to satisfy the request. The fact that the file system 60 can proceed with the allocation does not guarantee that the allocation will succeed, because the actual total available space may change constantly.

The file system 60 takes advantage of the page level replication capability of the underlying distributed addressable shared memory system 20 disclosed in the U.S. patent application incorporated by reference above. Page level replication allows the system to provide file replication. The data streams of a replicated file are backed by pages, which are themselves replicated. In this way, data streams are replicated automatically without intervention of the file system 60. The extra space consumed by the multiple replicas is not reflected in the file (data stream) sizes. The stream allocation size still reports the total allocation size in pages required for one replica. The pages backing temporary files, however, are not replicated.

FILE ACCESS AND RESOURCE SHARING - LOCKING

The shared memory provides the distribution mechanism for resource sharing among peer nodes running the file system 60 software. Each instance of the file system 60 on each network node views the shared memory resources (i.e., pages) as being shared with other local or remote threads. The file system 60 needs a way to implement high level, file system locks to provide consistent resource sharing. Any concurrency control structure can be used to implement locks, such as lock objects or semaphores. In database applications, locking may also be achieved by implementing concurrency control structures associated with database indices or keys. In file system applications access to files or directories may be controlled. Another example of file system locks is Byte Range Locking, which provides the users the ability to coordinate shared access to files. A byte range lock is a lock set on a range of bytes of a file. Coordinated shared access to a file can be accomplished by taking locks on the desired byte ranges. In general, the high level file system lock works in the following fashion: (a) a file system resource is to be shared by each file system 60 instance, and the access to the resource is coordinated by a locking protocol using a lock object data structure that represents the high level lock to coordinate the shared resource, and it is the value of the data structure that represents the current state of the lock; (b) to access the resource, the instance at each node must be able to look at the state (or value) of the lock data structure, and if it is "free," modify it so that it becomes "busy," but if it is "busy," then it has to wait to become "free," and there could be intermediate states between "free" and "busy" (i.e., more than two lock states), but in any event, in this byte range locking example, a lock is a description of a certain byte range being shared/exclusively locked by some thread of the file system 60, and a conflicting new byte range lock request that falls in or overlaps the already locked byte range will be denied or the requester may block (depending on how the request was made); and (c) access to or modification of the lock data structure by each node's instance needs to be serialized so that it in turn can then be used to coordinate high level resource sharing.

The locking features and capabilities of the shared memory engine described in the U.S. patent application serial no. 08/754,481, incorporated by reference above, allow the file system 60 to coordinate access to pages. The engine can also be used to coordinate access to resources, but in the case of complex high level resource locking such as Byte Range Locking, using the engine's locking features and capabilities directly to provide locks may be too costly for the following reasons: (a) each byte range lock would require a page representing the lock, and since the number of byte range locks can be large, the cost in terms of page consumption may be too high; and (b) the engine locks only provide two lock states (i.e., shared and exclusive), and high level file system locks may require more lock states.

The file system 60 of the invention implements the file system locking using the engine locking as a primitive to provide serialization to access and update the lock data structures. To read a lock structure, the file system 60 takes a shared lock on the data structure's page using the engine locking features and capabilities before it reads the page to prevent the data structure being modified. To modify the lock structure, it sets a exclusive lock on the page. The page lock is taken and released as soon as the lock structure value is read or modified.

With the serialization provided by the page locking and the page invalidation notification, the file system 60 implements the high level locks in the following way: (a) to take a file system lock (FS lock), the file system 60 sets a shared lock on the FS lock page and reads the page and then examines the lock structure; (b) if the lock structure indicates the resource is unlocked or locked in compatible lock mode, then the file system 60 requests to exclusively lock the page, and this guarantees only one file system 60 node instance can modify the lock data structure, and if the request succeeds then the file system 60 write maps the lock page and then changes the lock structure to set the lock and unlocks the page and sets page access to none; and

(c) if the resource is locked in incompatible lock mode, the file system 60 unlocks the page but retains the page read mapped, and it then puts itself (the current thread) in a queue and waits for a system event notifying that the lock value has changed, and when the lock value does change then the file system 60 thread gets notified and repeats the step (a) above. The file system 60 implements the notification using a signal primitive. The file system 60 threads waiting for a lock are blocked on a system event. When the page containing the lock changes, a signal is sent to each blocked file system 60 thread. Each blocked file system 60 threads then wakes up and repeats step (a). FS locks are stored in volatile pages.

FILE ACCESS AND RESOURCE SHARING - BYTE RANGE LOCKING

Byte Range Locking is a file system locking service exported to the users through the Win32 LockFile() and LockFileEx() API. It allows simultaneous access to different non-overlapping regions of a file data stream by multiple users. To access the data stream, the user locks the region (byte range) of the file to gain exclusive or shared read access to the region.

The file system 60 supports byte range locking for each individual data stream of the file. The following Win32-style byte range locking behavior is supported: (a) locking a region of a file is used to acquire shared or exclusive access to the specified region of the file, and the file system 60 will track byte range locks by file handle, therefore file handles provide a way to identify uniquely the owner of the lock; (b) locking a region that goes beyond the current end-of-file position is not an error; (c) locking a portion of a file for exclusive access denies all other processes both read and write access to the specified region of the file, and locking a portion of a file for shared access denies all other processes write access to the specified region of the file but allows other processes to read the locked region, and this means that the file system 60 must check byte range locks set on the data stream not only for lock requests but for every read or write access; (d) if an exclusive lock is requested for a region that is already locked either shared or exclusively by other threads, the request blocks or fails immediately depending on the calling option specified.; and (e) locks may not overlap an existing locked region of the file.

For each byte range lock, the file system 60 creates a byte range lock record to represent the lock. The record contains the following information: (a) byte range; (b) lock mode (shared or exclusive); (c) process identification; and (d) a Win32 lock key value.

The file system 60 regards the file byte ranges as resources with controlled access. For each byte range lock record, the file system 60 creates a file system lock (as discussed above) to coordinate the access to the byte range "resource." A compatible byte range lock request (share lock) translates into taking read lock on the file system lock associated with the byte range record. An exclusive byte range lock request is mapped to taking write lock on the file system lock.

Using the file system locking mechanism discussed above, lock requests waiting on the page containing the desired byte range will be notified when the page content changes.

Addressable Shared Memory Space

Having described the invention and various embodiments thereof in some detail, a more detailed description is now provided of the addressable shared memory space that is disclosed in the commonly-owned U.S. patent application serial number 08/754,481 filed November 22, 1996, and incorporated by reference above. All of the information provided below is contained in that patent application.

The addressable shared memory system disclosed in the U.S. patent application incorporated by reference is an "engine" that can create and manage a virtual memory space that can be shared by each computer on a network and can span the storage space of each memory device connected to the network. Accordingly, all data stored on the network can be stored within the virtual memory space and the actual physical location of the data can be in any of the memory devices connected to the network.

More specifically, the engine or system can create or receive, a global address signal that represents a portion, for example 4k bytes, of the virtual memory space. The global address signal can be decoupled from, i.e. unrelated to, the physical and virtual address spaces of the underlying computer hardware, to provide support for a memory space large enough to span each volatile and persistent memory device connected to the system. For example, systems of the invention can operate on 32-bit computers, but can employ global address signals that can be 128 bits wide. Accordingly, the virtual memory space spans 2^{128} bytes, which is much larger than the 2^{32} address space supported by the underlying computer hardware. Such an address space can be large enough to provide a separate address for every byte of data storage on the network, including all RAM, disk and tape storage.

For such a large virtual memory space, typically only a small portion is storing data at any time. Accordingly, the system includes a directory manager that tracks those portions of the virtual memory space that are in use. The system provides physical memory storage for each portion of the virtual memory space in use by mapping each such portion to a physical memory device, such as a RAM memory or a hard-drive. Optionally, the mapping includes a level of indirection that facilitates data migration, fault-tolerant operation, and load balancing.

By allowing each computer to monitor and track which portions of the virtual memory space are in use, each computer can share the memory space. This allows the networked computers to appear to have a single memory, and therefore can allow application programs running on different computers to communicate using techniques currently employed to communicate between applications running on the same machine.

In one aspect, the invention of the above-identified, incorporated-by-reference U.S. patent application can be understood to include computer systems having a addressable shared memory space. The systems can comprise a data network that carries data signals representative of computer readable information a persistent memory device that couples to the data network and that provides persistent data storage, and plural computers that each have an interface that couples to the data network, for accessing the data network to exchange data signals therewith. Moreover, each of the computers can include a shared memory subsystem for mapping a portion of the addressable memory space to a portion of the persistent storage to provide addressable persistent storage for data signals.

In a system that distributes the storage across the memory devices of the network, the persistent memory device will be understood to include a plurality of local persistent memory devices that each couple to a respective one of the plural computers. To this same end, the system can also include a distributor for mapping portions of the addressable memory space across the plurality of local persistent memory devices and a disk directory manager for tracking the mapped portions of the addressable memory space to provide information representative of the local persistent memory device that stores that portion of the addressable memory space mapped thereon.

The systems can also include a cache system for operating one of the local persistent memory devices as a cache memory for cache storing data signals associated with recently accessed portions of the addressable memory space. Further the system can include a migration controller for selectively moving portions of the addressable memory space between the local persistent memory devices of the plural computers. The migration controller can determine and respond to data access patterns, resource demands or any other criteria or heuristic suitable for practice with the invention. Accordingly, the migration controller can balance the loads on the network, and move data to nodes from which it is commonly accessed. The cache controller can be a software program running on a host computer to provide a software managed RAM and disk cache. The RAM can be any volatile memory including SRAM, DRAM or any other volatile memory. The disk can be any persistent memory including any disk, RAID, tape or other device that provides persistent data storage.

The systems can also include a coherent replication controller for generating a copy, or select number of copies, of a portion of the addressable memory space maintained in the local persistent memory device of a first computer and for storing the copy in the local persistent memory device of a second computer. The coherent replication controller can maintain the coherency of the copies to provide coherent data replication.

The systems can also be understood to provide integrated control of data stored in volatile memory and in persistent memory. In such systems a volatile memory device has volatile storage for data signals, and the shared memory subsystem includes an element, typically a software module, for mapping a portion of the addressable memory space to a portion of the volatile storage. In these systems the volatile memory device can be comprised of a plurality of local volatile memory devices each coupled to a respective one of the plural computers, and the persistent memory device can be comprised of a plurality of local persistent memory devices each coupled to a respective one of the plural computers.

In these systems, a directory manager can track the mapped portions of the addressable memory space, and can include two sub-components; a disk directory manager for tracking portions of the addressable memory space mapped to the local persistent memory devices, and a RAM directory manager for tracking portions of the addressable memory space mapped to the local volatile memory devices. Optionally, a RAM cache system can operate one of the local volatile memory devices as a cache memory for cache storing data signals associated with recently accessed portions of the addressable memory space.

The systems can include additional elements including a paging element for remapping a portion of the addressable memory space between one of the local volatile memory devices and one of the local persistent memory devices; a policy controller for determining a resource available signal representative of storage available on each of the plural computers and, a paging element that remaps the portion of addressable memory space from a memory device of a first computer to a memory device of a second computer, responsive to the resource available signal; and a migration controller for moving portions of addressable memory space between the local volatile memory devices of the plural computers.

Optionally, the systems can include a hierarchy manager for organizing the plural computers into a set of hierarchical groups wherein each group includes at least one of the plural computers. Each the group can include a group memory manager for migrating portions of addressable memory space as a function of the hierarchical groups.

The system can maintain coherency between copied portions of the memory space by including a coherent replication controller for generating a coherent copy of a portion of addressable memory space.

The system can generate or receive global address signals. Accordingly the systems can include an address generator for generating a global address signal representative of a portion of addressable memory space. The address generator can include a spanning unit for generating global address signals as a function of a storage capacity associated with the persistent memory devices, to provide global address signals capable of logically addressing the storage capacity of the persistent memory devices.

In distributed systems, the directory manager can be a distributed directory manager for storing within the distributed memory space, a directory signal representative of a storage location of a portion of the addressable memory space. The distributed directory manager can include a directory page generator for allocating a portion of the addressable memory space and for storing therein an entry signal representative of a portion of the directory signal. The directory page generator optionally includes a range generator for generating a range signal representative of a portion of the addressable memory space, and for generating the entry signal responsive to the range signal, to provide an entry signal representative of a portion of the directory signal that corresponds to the portion of the addressable memory space. Moreover, the distributed directory manager can include a linking system for linking the directory pages to form a hierarchical data structure of the linked directory pages as well as a range linking system for linking the directory pages, as a function of the range signal, to form a hierarchical data structure of linked directory pages.

As the data stored by the system can be homeless, in that the data has no fixed physical home, but can migrate, as resources and other factors dictate, between the memory devices of the network, a computer system according to the invention can include a directory page generator that has a node selector for generating a responsible node signal representative of a select one of the plural computers having location information for a portion of the shared address space. This provides a level of indirection that decouples the directory from the physical storage location of the data. Accordingly, the directory needs only to identify the node, or other device, that tracks the physical location of the data. This way, each time data migrates between physical storage locations, the directory does not have to be updated, since the node tracking the location of the data has not changed and still provides the physical location information.

Accordingly, the system can include page generators that generate directory pages that carry information representative of a location monitor, such as a responsible computer node, that tracks a data storage location, to provide a directory structure for tracking homeless data. Moreover, the directory itself can be stored as pages within the virtual memory space. Therefore, the data storage location can store information representative of a directory page, to store the directory structure as pages of homeless data.

In another aspect, the invention of the above-identified, incorporated-by-reference U.S. patent application can be understood as methods for providing a computer system having a addressable shared memory space. The method can include the steps of providing a network for carrying data signals representative of computer readable information, providing a hard-disk, coupled to the network, and having persistent storage for data signals, providing plural computers, each having an interface, coupled to the data network, for exchanging data signals between the plural computers, and assigning a portion of the addressable memory space to a portion of the persistent storage of the hard disk to provide addressable persistent storage for data signals.

Turning now to the drawings related to the addressable shared memory system or engine of the above-identified, incorporated-by-reference U.S. patent application, FIG. 5 illustrates a computer network 10 that provides a shared memory that spans the memory space of each node of the depicted computer network 210.

Specifically, FIG. 5 illustrates a computer network 210 that includes a plurality of nodes 212a-212c, each having a CPU 214, an operating system 216, an optional private memory device 218, and a shared memory subsystem 220. As further depicted in by FIG. 5, each node 212a-212c connects via the shared memory subsystem 220 to a virtual shared memory 222. As will be explained in greater detail hereinafter, by providing the shared memory subsystem 220 that allows the node 212a-212c to access the virtual shared memory 222, the computer network 210 enables network nodes 212a-212c to communicate and share functionality using the same techniques employed by applications when communicating between applications running on the same machine. These techniques can employ object linking and embedding, dynamic link libraries, class registering, and other such techniques. Accordingly, the nodes 212 can employ the virtual shared memory 222 to exchange data and objects between application programs running on the different nodes 212 of the network 210.

In the embodiment depicted in FIG. 5, each node 212 can be a conventional computer system such as a commercially available IBM PC compatible computer system. The processor 214 can be any processor unit suitable for performing the data processing for that computer system. The operating system 216 can be any commercially available or proprietary operating system that includes, or can access, functions for accessing the local memory of the computer system and networking.

The private memory device 218 can be any computer memory device suitable for storing data signals representative of computer readable information. The private memory provides the node with local storage that can be kept inaccessible to the other nodes on the network. Typically the private memory device 218 includes a RAM, or a portion of a RAM memory, for temporarily storing data and application programs and for providing the processor 214 with memory storage for executing programs. The private memory device 18 can also include persistent memory storage, typically a hard disk unit or a portion of a hard disk unit, for the persistent storage of data.

The shared memory subsystem 220 depicted in FIG. 5 is an embodiment of the invention that couples between the operating system 216 and the virtual shared memory 222 and forms an interface between the operating system 216 and the virtual shared memory to allow the operating system 216 to access the virtual shared memory 222. The depicted shared memory subsystem 220 is a software module that operates as a stand-alone distributed shared memory engine. The depicted system is illustrative and other systems of the invention can be realized as shared memory subsystems that can be embedded into an application program, or be implemented as an embedded code of a hardware device. Other such applications can be practiced without departing from the scope of the invention.

The depicted virtual shared memory 222 illustrates a virtual shared memory that is accessible by each of the nodes 212a-212c via the shared memory subsystem 220. The virtual shared memory 222 can map to devices that provide physical storage for computer readable data, depicted in FIG. 5 as a plurality of pages 224a-224d. In one embodiment, the pages form portions of the shared memory space and divide the address space of the shared memory into page addressable memory spaces. For example the address space can be paged into 4K byte sections. In other embodiments alternative granularity can be employed to manager the shared memory space. Each node 212a-212c through the shared memory subsystem 220 can access each page 224a-224d stored in the virtual shared memory 222. Each page 224a-224d represents a unique entry of computer data stored within the virtual shared memory 222. Each page 224a-224d is accessible to each one of the nodes 212a-212c, and alternatively, each node can store additional pages of data within the virtual shared memory 222. Each newly stored page of data can be accessible to each of the other nodes 212a-212c. Accordingly, the virtual shared memory

222 provides a system for sharing and communicating data between each node 212 of the computer network 210.

FIG. 6 illustrates in functional block diagram form a computer network 230 that has a distributed shared memory. In this embodiment, each node 212a-212c has a memory subsystem 232 that connects between the operating system 216 and the two local memory devices, the RAM 234 and the disk 236, and that further couples to a network 238 that couples to each of the depicted nodes 212a, 212b and 212c and to a network memory device 226.

More particularly, FIG. 6 illustrates a distributed shared memory network 30 that includes a plurality of nodes 212a-212c, each including a processing unit 214, an operating system 216, a memory subsystem 232, a RAM 234, and a disk 236. FIG. 6 further depicts a computer network system 38 that connects between the nodes 212a-212c and the network memory device 226. The network 238 provides a network communication system across these elements.

The illustrated memory subsystems 232a-232c that connect between the operating system 216a-216c, the memory elements 234a-234c, 236a-236c, and the network 238, encapsulate the local memories of each of the nodes to provide an abstraction of a shared virtual memory system that spans across each of the nodes 212a-212c on the network 238. The memory subsystems 232a-232c can be software modules that act as distributors to map portions of the addressable memory space across the depicted memory devices. The memory subsystems further track the data stored in the local memory of each node 212 and further operate network connections with network 238 for transferring data between the nodes 212a-212c. In this way, the memory subsystems 232a-232c access and control each memory element on the network 238 to perform memory access operations that are transparent to the operating system 216. Accordingly, the operating system 216 interfaces with the memory subsystem 232 as an interface to a global memory space that spans each node 212a-212c on the network 238.

FIG. 6 further depicts that the system 230 provides a distributed shared memory that includes persistent storage for portions of the distributed memory. In particular, the depicted embodiment includes a memory subsystem, such as subsystem 232a, that interfaces to a persistent memory device, depicted as the disk 236a. The subsystem 232a can operate the persistent memory device to provide persistent storage for portions of the distributed shared memory space. As illustrated, each persistent memory device 236 depicted in FIG. 6 has a portion of the addressable memory space mapped onto it. For example, device 236a has the portions of the addressable memory space, C_o, C_d, C_g, mapped onto it, and provides persistent storage for data signals stored in those ranges of addresses.

Accordingly, the subsystem 232a can provide integrated control of persistent storage devices and electronic memory to allow the distributed shared memory space to span across both types of storage devices, and to allow portions of the distributed shared memory to move between persistent and electronic memory depending on predetermined conditions, such as recent usage.

In one optional embodiment, the nodes of the network are organized into a hierarchy of groups. In this embodiment, the memory subsystems 232a-232c can include a hierarchy manager that provides hierarchical control for the distribution of data. This includes controlling the migration controller, and policy controller, which are discussed in detail below, to perform hierarchical data migration and load balancing, such that data migrates primarily between computers of the same group, and passes to other groups in hierarchical order. Resource distribution is similarly managed.

FIG. 7 illustrates in more detail one shared memory subsystem 240 according to the invention. FIG. 7 depicts a shared memory subsystem 240, that includes an interface 242, a DSM directory manager 244, a memory controller 246, a local disk cache controller 248, and a local RAM cache controller 250. FIG. 7 further depicts the network 254, an optional consumer of the DSM system, depicted as the service 258, the operating system 216, a disk driver 260, a disk element 262 and a RAM element 264.

The shared memory subsystem 240 depicted in FIG. 7 can encapsulate the memory management operations of the network node 212 to provide a virtual shared memory that can span across each node that connects into the network 254. Accordingly, each local node 212 views the network as a set of nodes that are each connected to a large shared computer memory.

The depicted interface 242 provides an entry point for the local node to access the shared memory space of the computer network. The interface 242 can couple directly to the operating system 216, to a distributed service utility such as the depicted DSM file system 258, to a distributed user-level service utility, or alternatively to any combination thereof.

The depicted interface 242 provides an API that is a memory oriented API. Thus, the illustrated interface 242 can export a set of interfaces that provide low-level control of the distributed memory. As illustrated in FIG. 7, the interface 242 exports the API to the operating system 216 or to the optional DSM service 258. The operating system 216 or the service employs the interface 242 to request standard memory management techniques, such as reading and writing from portions of the memory space. These portions of the memory space can be the pages as described above which can be 4K byte portions of the shared memory space, or other units of memory, such as objects or segments. Each page can be located within the shared memory space which is designated by a global address signal for that page of memory. The system can receive address signals from an application program or, optionally, can include a global address generator that generates the address signals. The address generator can include a spanning module that generates address signals for a memory space that spans the storage capacity of the network.

Accordingly, in one embodiment, the interface 242 receives requests to manipulate pages of the shared memory space. To this end, the interface 242 can comprise a software module that includes a library of functions that can be called by services, the OS 216, or other caller, or device. The function calls provide the OS 216 with an API of high level memory oriented services, such as read data, write data, and allocate memory. The implementation of the functions can include a set of calls to controls that operate the directory manager 244, and the local memory controller 246. Accordingly, the interface 242 can be a set of high level memory function calls to interface to the low-level functional elements of shared memory subsystem 240.

FIG. 7 further depicts a DSM directory manager 244 that couples to the interface 242. The interface 242 passes request signals that represent requests to implement memory operations such as allocating a portion of memory, locking a portion of memory, mapping a portion of memory, or some other such memory function. The directory manager 244 manages a directory that can include mappings than can span across each memory device connected to the network

238 depicted in FIG. 6, including each RAM and disk element accessible by the network. The directory manager 244 stores a global directory structure that provides a map of the global address space. In one embodiment as will be explained in greater detail hereinafter, the directory manager 244 provides a global directory that maps between global address signals and responsible nodes on the network. A responsible node stores information regarding the location and attributes of data associated with a respective global address, and optionally stores a copy of that page's data. Consequently, the directory manager 244 tracks information for accessing any address location within the virtual address space.

The control of the distributed shared memory can be coordinated by the directory manager 244 and the memory controller 246. The directory manager 244 maintains a directory structure that can operate on a global address received from the interface 242 and identify, for that address, a node on the network that is responsible for maintaining the page associated with that address of the shared memory space. Once the directory manager 244 identifies which node is responsible for maintaining a particular address, the directory manager 244 can identify a node that stores information for locating a copy of the page, and make the call to the memory controller 246 of that node and pass to that node's memory controller the memory request provided by the memory interface 242. Accordingly, the depicted directory manager 244 is responsible for managing a directory structure that identifies for each page of the shared memory space a responsible node that tracks the physical location of the data stored in the respective page. Thus, the directory, rather than directly providing the location of the page, can optionally identify a responsible node, or other device, that tracks the location of the page. This indirection facilitates maintenance of the directory as pages migrate between nodes.

The memory controller 246 performs the low level memory access functions that physically store data within the memory elements connected to the network. In the depicted embodiment, the directory manager 244 of a first node can pass a memory access request through the interface 242, to the network module of the OS 216, and across the network 254 to a second node that the directory manager 244 identifies as the responsible node for the given address. The directory manager 244 can then query the responsible node to determine the attributes and the current owner node of the memory page that is associated with the respective global address. The owner of the respective page is the network node that has control over the memory storage element on which the data of the associated page is stored. The memory controller 246 of the owner can access, through the OS 216 of that node or through any interface, the memory of the owner node to access the data of the page that is physically stored on that owner node.

In particular, as depicted in FIG. 7, the directory manager 244 couples to the network module 252 which couples to the network 254. The directory manager can transmit to the network module 252 a command and associated data that directs the network interface 252 to pass a data signal to the owner node. The owner node receives the memory request across network 254 and through network module 252 that passes the memory request to the interface 242 of that owner node. The interface 242 couples to the memory controller 246 and can pass the memory request to the local memory controller of that owner node for operating the local storage elements, such as the disk or RAM elements, to perform the requested memory operation.

Once the owner node has performed the requested memory operation, such as reading a page of data, the memory subsystem 240 of the owner node can then transfer the page of data, or a copy of the page of data, via the network 254 to the node that originally requested access to that portion of the shared memory. The page of data is transferred via the network 254 to the network module 252 of the requesting node and the shared memory subsystem 240 operates the memory controller 246 to store in the local memory of the requesting node a copy of the accessed data.

Accordingly, in one embodiment of the invention, when a first node accesses a page of the shared memory space which is not stored locally on that node, the directory manager 244 identifies a node that has a copy of the data stored in that page and moves a copy of that data into the local memory of the requesting node. The local memory storage, both volatile and persistent, of the requesting node therefore becomes a cache for pages that have been requested by that local node. This embodiment is depicted FIG. 7 which depicts a memory controller that has a local disk cache controller 248 and a local RAM cache controller 250. Both of these local cache controllers can provide to the operating system 216, or other consumer pages of the shared memory space that are cache stored in the local memory of the node, including local persistent memory and local volatile memory.

The shared memory subsystem can include a coherent replication controller that maintains coherency between cached pages by employing a coherence through invalidation process, a coherence through migration process or other coherence process suitable for practice with the present invention. The coherent replication controller can automatically generate a copy of the data stored in each page and can store the copy in a memory device that is separate from the memory device of the original copy. This provides for fault tolerant operation, as the failure of any one memory device will not result in the loss of data. The coherent replication controller can be a software model that monitors all copies of pages kept in volatile memory and made available for writing. The controller can employ any of the coherency techniques named above, and can store tables of location information that identifies the location information for all generated copies.

FIG. 8 illustrates in greater detail one embodiment of a shared memory subsystem according to the invention. The shared memory subsystem 270 depicted in FIG. 8 includes a remote operations element 274, a local RAM cache 276, a RAM copyset 278, a global RAM directory 280, a disk copyset 282, a global disk directory 284, a configuration manager 288, a policy element 290, and a local disk cache 94. FIG. 8 further depicts a network element 304, a physical memory 300, shared data element 302, a physical file system 298, which is part of the operating system 216, a configuration service 308, a diagnostic service 310, and a memory access request 312. The depicted subsystem 270 can be a computer program that couples to the physical memory, file system, and network system of the host node, or can be electrical circuit card assemblies that interface to the host node, or can be a combination of programs and circuit card assemblies.

The flow scheduler 272 depicted in FIG. 8 can orchestrate the controls provided by an API of the subsystem 270. In one embodiment, the flow scheduler 272 can be a state machine that monitors and responds to the requests 312 and remote requests through network 304 which can be instructions for memory operations and which can include signals representative of the global addresses being operated on. These memory operation requests 312 can act as op-codes for primitive operations on one or more global addresses. They can be read and write requests, or other memory operations. Alternatively, the flow scheduler 272 can be a program, such as an interpreter, that provides an execution environment and can map these op-codes into control flow programs called applets. The applets can be independent executable programs that employ both environment services, such as threading, synchronization, and buffer management, and the elements depicted in FIG. 8. The API is capable of being called from both external clients, like a distributed shared memory file system, as well as recursively by the applets and the other elements 274-294 of the subsystem 270. Each element can provide a level of encapsulation to the management of a particular resource or aspect of the system. To this end, each element can export an API consisting of functions to be employed by the applets. This structure is illustrated in FIG. 8. Accordingly, the flow scheduler 272 can provide an environment to load and execute applets. The applets are dispatched by the flow scheduler 272 on a per op-code basis and can perform the control flow for sequential or parallel execution of an element to implement the op-code on the specified global address, such as a read or write operation. Optionally, the flow scheduler 272 can include an element to change dynamically the applet at run time as well as execute applets in parallel and in interpreted mode.

The depicted shared memory subsystem 270 includes a bifurcated directory manager that includes the global RAM directory 280 and the global disk directory 284. The global RAM directory 280 is a directory manager that tracks information that can provide the location of pages that are stored in the volatile memory, typically RAM, of the network nodes. The global disk directory 284 is a global disk directory manager that manages a directory structure that tracks information that can provide the location of pages that are stored on persistent memory devices. Together, the global RAM directory 280 and the global disk directory 284 provide the shared memory subsystem 270 with integrated directory management for pages that are stored in persistent storage and volatile memory.

In one embodiment a paging element can operate the RAM and disk directory managers to remap portions of the addressable memory space between one of the volatile memories and one of the persistent memories. In the shared memory system, this allows the paging element to remap pages from the volatile memory of one node to a disk memory of another node. Accordingly, the RAM directory manager passes control of that page to the disk directory manager which can then treat the page as any other page of data. This allows for improved load balancing, by removing data from RAM memory, and storing it in the disk devices, under the control of the disk directory manager.

The local memory controller of the subsystem 270 is provided by the local RAM cache 276 and the local disk cache 294. The local RAM cache 276 which couples to the physical memory 300 of the local node can access, as described above, the virtual memory space of the local node to access data that is physically stored within the RAM memory 300. Similarly, the local disk cache 294 couples to the persistent storage device 298 and can access a physical location that maintains in the local persistent storage data of the distributed shared memory.

FIG. 8 also depicts a remote operations element 274 that couples between the network 304 and the flow scheduler 272. The remote operations element 274 negotiates the transfer of data across the network 304 for moving portions of the data stored in the shared memory space between the nodes of the network. The remote operations element 274 can also request services from remote peers, i.e. invalidate to help maintain coherency or for other reasons.

FIG. 8 also depicts a policy element 290 that can be a software module that acts as a controller to determine the availability of resources, such as printer capabilities, hard-disk space, available RAM and other such resources. The policy controller can employ any of the suitable heuristics to direct the elements, such as the paging controller, disk directory manager, and other elements to dynamically distribute the available resources.

FIG. 8 further depicts a memory subsystem 270 that includes a RAM copyset 278 and a disk copyset 282. These copysets can manage copies of pages that are cached at a single node. The disk copyset 282 can maintain information on copies of pages that are stored in the local disk cache, which can be the local persistent memory. Similarly, the RAM copyset 278 can maintain information on copies of pages that are stored in the local RAM cache which can be the local RAM. These copysets encapsulate indexing and storage of copyset data that can be employed by applets or other executing code for purposes of maintaining the coherency of data stored in the shared memory space. The copyset elements can maintain copyset data that identifies the pages cached by the host node. Further, the copyset can identify the other nodes on the network that maintain a copy of that page, and can further identify for each page which of these nodes is the owner node, wherein the owner node can be a node which has write privileges to the page being accessed. The copysets themselves can be stored in pages of the distributed shared memory space.

The local RAM cache 276 provides storage for memory pages and their attributes. In one embodiment, the local RAM cache 276 provides a global address index for accessing the cached pages of the distributed memory and the attributes based on that page. In this embodiment, the local ram cache 276 provides the index by storing in memory a list of each global address cached in the local RAM. With each listed global address, the index provides a pointer into a buffer memory and to the location of the page data. Optionally, with each listed global address, the index can further provide attribute information including a version tag representative of the version of the data, a dirty bit representative of whether the RAM cached data is a copy of the data held on disk, or whether the RAM cached data has been modified but not yet flushed to disk, a volatile bit to indicate if the page is backed by backing store in persistent memory, and other such attribute information useful for managing the coherency of the stored data.

In the embodiment depicted in FIG. 8, the memory subsystem 270 provides the node access to the distributed memory space by the coordinated operation of the directory manager that includes the global RAM directory 280 and the global disk directory 284, the cache controller that includes the local RAM cache and the local disk cache elements 276 and 294, and the copyset elements which include the RAM copyset 278 and the disk copyset 282.

The directory manager provides a directory structure that indexes the shared address space. Continuing with the example of a paged shared address space, the directory manager of the subsystem 270 allows the host node to access, by global addresses, pages of the shared memory space.

FIGS. 9 and 10 illustrate one example of a directory structure that provides access to the shared memory space. FIG. 9 depicts a directory page 320 that includes a page header 322, directory entries 324 and 326, wherein each directory entry includes a range field 330, a responsible node field 332, and an address field 334. The directory pages can be generated by a directory page generator that can be a software module controlled by the directory manager. It will be understood that the directory manager can generate multiple directories, including one for the Global disk and one for the Global RAM directories. The depicted directory page 320 can be a page of the global address space, such as a 4K byte portion of the shared address space.

Therefore, the directory page can be stored in the distributed shared memory space just as the other pages to which the directory pages provide access.

As further depicted in FIG. 9, each directory page 120 includes a page header 322 that includes attribute information for that page header, which is typically metadata for the directory page, and further includes directory entries such as the depicted directory entries, 324 and 326, which provide an index into a portion of the shared address space wherein that portion can be one or more pages, including all the pages of the distributed shared memory space. The depicted directory page 320 includes directory entries that index a selected range of global addresses of the shared memory space. To this end, the directory generator can include a range generator so that each directory entry can include a range field 330 that describes the start of a range of addresses that that entry locates.

Accordingly, each directory page 320 can include a plurality of directory entries, such as entries 324 and 326, that can subdivide the address space into a subset of address ranges. For example, the depicted directory page 320 includes two directory entries 324 and 326. The directory entries 324 and 326 can, for example, subdivide the address space into two sub-portions. In this example, the start address range of the directory entry 324 could be the base address of the address space, and the start address range of the directory entry 326 could be the address for the upper half of the memory space. Accordingly, the directory entry 324 provides an index for pages stored in the address space between the base address and up to the mid-point of the memory space and, in complement thereto, the directory entry 326 provides an index to pages stored in the address space that ranges from the mid-point of the address space to the highest address.

FIG. 9 further depicts a directory page 320 that includes, in each directory entry, a responsible node field 332 and the child page global address field 334. These fields 332, 334 provide further location information for the data stored in pages within the address range identified in field 330.

FIG. 10 depicts a directory 340 formed from directory pages similar to those depicted in FIG. 9. FIG. 10 depicts that the directory 340 includes directory pages 342, 350-354, and 360-

366. FIG. 10 further depicts that the directory 340 provides location information to the pages of the distributed shared memory space depicted in FIG. 10 as pages 370-384.

The directory page 342 depicted in FIG. 10 acts like a root directory page and can be located at a static address that is known to each node coupled to the distributed address space. The root directory page 342 includes three directory entries 344, 346, and 348. Each directory entry depicted in FIG. 10 has directory entries similar to those depicted in FIG. 9. For example, directory entry 344 includes a variable Co which represents the address range field 330, a variable Nj representative of the field 332, and a variable Cs representative of the field 334. The depicted root directory page 342 subdivides the address space into three ranges illustrated as an address range that extends between the address Co and Cd, a second address range that extends between the address Cd and Cg, and a third address range that extends between Cg and the highest memory location of the address space.

As further depicted in FIG. 10, each directory entry 344, 346, and 348 points to a subordinate directory page, depicted as directory pages 350, 352, and 354, each of which further subdivides the address range index by the associated directory entry of the root directory 342. In FIG. 9, this subdivision process continues as each of the directory pages 350, 352, and 354 each again have directory entries that locate subordinate directory pages including the depicted examples of directory pages 360, 362, 364, and 366.

The depicted example of directory pages 360, 362, 364, and 366 are each leaf entries. The leaf entries contain directory entries such as the directory entries 356 and 358 of the leaf entry 360, that store a range field 330 and the responsible node field 332. These leaf entries identify an address and a responsible node for the page in the distributed memory space that is being accessed, such as the depicted pages 370-384. For example, as depicted in FIG. 10, the leaf entry 356 points to the page 370 that corresponds to the range field 330 of the leaf entry 356, which for a leaf entry is the page being accessed. In this way, the directory structure 340 provides location information for pages stored in the distributed address space.

In the depicted embodiment of FIG. 10, a node selector can select a responsible node for each page, as described above, so that the leaf entry 356 provides information of the address and responsible node of the page being located. Accordingly, this directory tracks ownership and responsibility for data, to provide a level of indirection between the directory and the physical location of the data. During a memory access operation, the memory subsystem 270 passes to the responsible node indicated in the leaf entry 356 the address of the page being accessed. The shared memory subsystem of that node can identify a node that stores a copy of the page being accessed, including the owner node. This identification of a node having a copy can be performed by the RAM copyset or disk copyset of the responsible node. The node having a copy stored in its local physical memory, such as the owner node, can employ its local cache elements, including the local RAM cache and local disk cache to the identify from the global address signal a physical location of the data stored in the page being accessed. The cache element can employ the operating system of the owner node to access the memory device that maintains that physical location in order that the data stored in the page can be accessed. For a read-memory operation, or for other similar operations, the data read from the physical memory of the owner node can be passed via the network to the memory subsystem of the node requesting the read and subsequently stored into the virtual memory space of the requesting node for use by that node.

With reference again to FIG. 10, it can be seen that the depicted directory structure 340 comprises a hierarchical structure. To this end, the directory structure 340 provides a structure that continually subdivides the memory space into smaller and smaller sections. Further, each section is represented by directory pages of the same structure, but indexes address spaces of different sizes. As pages are created or deleted, a linker inserts or deletes the pages from the directory. In one embodiment, the linker is a software module for linking data structures. The linker can operate responsive to the address ranges to provide the depicted hierarchical structure. Accordingly, the depicted directory 340 provides a scaleable directory for the shared address space. Moreover, the directory pages are stored in the distributed address space and maintained by the distributed shared memory system. A root for the directory can be stored in known locations to allow for bootstrap of the system. Consequently, commonly used pages are copied and distributed, and rarely used pages are shuffled off to disk. Similarly, directory pages will migrate to those nodes that access them most, providing a degree of self-organization that reduces network traffic.

FIG. 11 depicts the directory of FIG. 10 being employed by a system according to the invention. In particular FIG. 11 depicts a system 400 that includes two nodes, 406a and 406b, a directory structure 340, and a pair of local memories having volatile memory devices 264a and 264b, and persistent memory devices 262a and 262b. Depicted node 406a includes an address consumer 408a, a global address 410a, and interface 242a, a directory manager 244a and a memory controller 246a. Node 406b has corresponding elements. The nodes are connected by the network 254. The directory 340 has a root page, directory pages A-F, and pages 1-5.

Each node 406a and 406b operates as discussed above. The depicted address consumers 408a and 408b can be an application program, file system, hardware device or any other such element that requests access to the virtual memory. In operation, the address consumers 408a and 408b request an address, or range of addresses, and the directory manager can include a global address generator that provides the consumer with the requested address, or a pointer to the requested address. As addresses get generated, the respective directory managers 244a and 244b generate directory pages and store the pages in the directory structure 340. As depicted, the directory structure 340 tracks the portions of the address space being employed by the system 400, and physical storage for each page is provided within the local memories.

As shown in FIG. 11, the data associated with the directory pages are distributively stored across the two local memories and duplicate copies can exist. As described above and now illustrated in FIG. 11, the data can move between different local memories and also move, or page, between volatile and persistent storage. The data movement can be responsive to data requests made by memory users like application programs, or by operation of the migration controller described above. As also described above, the movement of data between different memory locations can occur without requiring changes to the directory 340. This is achieved by providing a directory 340 that is decoupled from the physical location of the data by employing a pointer to a responsible node that tracks the data storage location. Accordingly, although the data storage location can change, the responsible node can remain constant, thereby avoiding any need to change the directory 340.

Shared Client-Side Web Cache

The distributed, structured file system described above provides a number of advantages that are not readily apparent. For example, the file system 60 may be used to provide an efficient, shared client-side Web cache.

FIG. 12 shows a typical embodiment of a system for accessing the global information network commonly known as the Internet. The user generally uses a terminal 400 to interact with the network. The terminal 400 may be a UNIX workstation, an Apple Macintosh, an IBM PC or PC clone, or any other hardware device capable of running a browser program such as Netscape Navigator or Microsoft Internet Explorer. The user requests data from a variety of other computers connected to the global information network. One particular type of file that may be requested is a file written in Hypertext Markup Language (HTML), commonly referred to as "Web" pages. Although the invention will be explained in relation to HTML files, it should be emphasized that it is applicable to any form of downloaded data, such as JAVA applets and ActiveX controls.

HTML files generally include "tags" that indicate to the browser application executing on the terminal 400 that special action should be taken. For example, a tag may indicate to the browser: that a graphics file should be displayed at a particular point in the document (typical formats include GIF, JPEG, MPEG, and MOV files); that certain text should be centered, bolded, or otherwise formatted before it is displayed on the screen of the terminal 400; that the background of a document should be shaded or have a particular pattern when displayed on the screen of the terminal 400; or that a different HTML file should be loaded in place of the HTML file the browser is currently displaying.

The user's terminal 400 is usually connected to a mass storage device 408. The mass storage device 408 may be a hard disk drive, random access memory, or any other media that allows data to be written to it. The browser program executing on the terminal 400 uses the mass storage device 408 to store recently accessed Web pages . The cache provides a mapping between Uniform Resource Locator addresses and locally cached files.

For example, Internet Explorer, manufactured by Microsoft Corporation of Redmond, Washington, caches downloaded ActiveX objects, Web pages, Web page fragments, JAVA applets, image files, and other downloaded data. The cache is instantiated as a COM Name Space extension object. This object supports the OLE IPERSIST API for storage access and provides a mapping between URL addresses and locally cached files. Another example is Netscape Navigator, manufactured by Netscape Communications Corporation of Mountain View, California, which also caches Web pages, Web page fragments, JAVA applets, image files, and other downloaded data within a directory which is specified as a parameter in the browser. The browser stores downloaded data as separate files in the directory and the file names are browser generated handles that are unique per client.

The distributed file system 60 described above allows user 400 and user 420 to share their Internet browser caches. This is particularly advantageous if user 400 and user 420 are employees of the same corporation or have some other close working relationship that is facilitated by sharing their Internet Web caches. For example, if user 400 has accessed a Web page 402 or Web site that user 420 subsequently needs to access, the Web page 402 or Web site will appear to be cached by the browser application used by user 420. This eliminates the delay associated with retrieving the Web page 402 or Web site over the network. Also, the distributed file system 60 allows the size of the shared Web cache to increase as the number of users sharing the cache increases. This results from the distributed file system 60 utilizing local memory on each of the users' terminals. Accordingly, much larger Web cache sizes can be achieved than with traditional methods such as proxy servers.

Since the browser application caches are files that are stored by the browsers, it is clear from the above description that files stored in the distributed file system described above can be shared between both users; however, the browser application used by each user must be directed to cache downloaded data in the same cache file. In these embodiments, an object may be provided which redirects a users cache read or cache right to or from its local cache to the shared cache file.

In operation, the user of terminal 400 loads data from Web server. Although this example will use a Web page 402 as the data that is downloaded, it should be emphasized that the data can be any form of data accessible over a network, and, in particular, HTML files, HTML file fragments, JAVA applets, ActiveX controls, or image files. The user of terminal 400 experiences some delay while the HTML file is transmitted by the Web server to the user's terminal 400. In order to avoid a similar delay should the user access the same Web page 402 again, the browser application caches the downloaded HTML page on the mass storage device 408 associated with the terminal 400 by writing the downloaded data representing the Web page 402 to a file. In traditional systems, when the user of terminal 420 desires to load the same Web page 402, the same delay is encountered while Web server transmits the Web page 402 to the terminal 420 and the browser application executed on the terminal 420 caches the HTML Web page 402 on the associated mass storage device 428.

Using the distributed file system 60 described above allows the client-side Web cache, i.e., the caches on mass storage devices 408, 428, to be shared by both users. Thus, when the user of terminal 400 downloads the HTML page and caches it on the mass storage device 408 associated with its terminal 400, a cache file is created in the distributed file system 60. If the user of terminal 400 desires to access that same Web page 402 again, the browser application executing on terminal 400 retrieves the Web page 402 from its cache file instead of fetching the Web page 402 from the Web server. Similarly, if the user of terminal 420 desires to access the Web page 402 cached by the first user, the browser application executing on the terminal 420 accesses the cache file created by its browser. Because the cache file is shared between the users, the browser application executing on terminal 420 will find the Web page 402 in the cache file and display it without requiring the Web server to transmit it a second time.

For example, Internet Explorer uses two distinct client-side caches. Both caches are stored in folders visible within the system root. One cache stores ActiveX controls, and this cache is a standard file system directory. The second cache is for Web pages, Web page fragments, and other downloaded data. Since Internet explorer stores downloaded data using the data's file name, the browser application executing on terminal 400 and terminal 420 will each attempt to access the same file in the same directory. The directories and the files are shared by terminal 400 and terminal 420 for systems using the shared file system 60 described above, the browser applications executing on those terminals share their caches.

Another example is Netscape Navigator. Netscape Navigator uses a directory specified within the browser as its cache, but file names within the cache file are browser generated handles which are unique for each browser. That is, the same file may be named differently depending on whether the browser application executing on terminal 400 stored the file or the browser application executing on terminal 420 stored the file. This problem may be solved by providing a translation between stored file handles and files to redirect file handle requests to the appropriate file. Such a translation can be effected by, in one embodiment, a table mapping browser file handles to URL addresses.

As noted above, the distributed file system 60 need not be implemented using a distributed shared memory. For example, the underlying memory may be unstructured. In another example, the mass storage devices 408, 428 may be connected by a physical bus and appropriate hardware may be implemented that insures the data written to one of the mass storage devices is replicated on the other.

Variations, modifications, and other implementations of what is described herein will occur to those of ordinary skill in the art without departing from the spirit and the scope of the invention as claimed. Accordingly, the invention is to be defined not by the preceding illustrative description but instead by the spirit and scope of the following claims.

CLAIMS

What is claimed is:

1. A method for locally caching and sharing downloaded data between a plurality of client nodes, the method comprising:

(a) providing a plurality of client nodes interconnected by a network;

(b) storing on each node an instance of a control program for manipulating caches of downloaded data to provide multiple, distributed instances of the control program;

(c) interfacing each instance of the control program to a globally addressable data store that provides persistent storage of data;

(d) operating each instance of said control program to employ the globally addressable data store as a memory device; and

(e) storing data downloaded by any one of the plurality of nodes in the globally addressable data store.

2. The method of claim 1 wherein step (e) comprises storing HTML files downloaded by any one of the plurality of nodes in the globally addressable data store.

3. The method of claim 1 wherein step (e) comprises storing ActiveX controls downloaded by any one of the plurality of nodes in the globally addressable data store.

4. The method of claim 1 wherein step (e) comprises storing fragments of downloaded HTML files in the globally addressable data store.

5. The method of claim 1 wherein step (e) comprises storing JAVA applets downloaded by any one of the plurality of nodes in the globally addressable data store.

6. The method of claim 1 wherein step (e) comprises image files downloaded by any one of the plurality of nodes in the globally addressable data store.

7. The method of claim 1 further comprising the step of:

(f) retrieving downloaded data stored in the globally addressable data store by a different one of the plurality of nodes.

8. The method of claim 1 further comprising the step of associating a URL address with data stored in the globally addressable data store relating to the data accessed at the URL address.

9. A method for locally caching and sharing downloaded data between a plurality of client nodes, the method comprising:

(a) providing a plurality of client nodes interconnected by a network;

(b) storing on each node an instance of a control program for manipulating caches of downloaded data to provide multiple, distributed instances of the control program;

(c) interfacing each instance of the control program to a distributed shared memory that provides persistent storage of data;

(d) operating each instance of said control program to employ the distributed shared memory as a memory device; and

(e) storing data downloaded by any one of the plurality of nodes in the distributed shared memory.

10. The method of claim 9 wherein step (e) comprises storing HTML files downloaded by any one of the plurality of nodes in the distributed shared memory.

11. The method of claim 9 wherein step (e) comprises storing ActiveX controls downloaded by any one of the plurality of nodes in the distributed shared memory.

12. The method of claim 9 wherein step (e) comprises storing fragments of downloaded HTML files in the distributed shared memory.

13. The method of claim 9 wherein step (e) comprises storing JAVA applets downloaded by any one of the plurality of nodes in the distributed shared memory.

14. The method of claim 9 wherein step (e) comprises image files downloaded by any one of the plurality of nodes in the distributed shared memory.

15. The method of claim 9 further comprising the step of:

(f) retrieving downloaded data stored in the distributed shared memory by a different one of the plurality of nodes.

16. The method of claim 9 further comprising the step of associating a URL address with data stored in the distributed shared memory relating to the data accessed at the URL address.

1/12

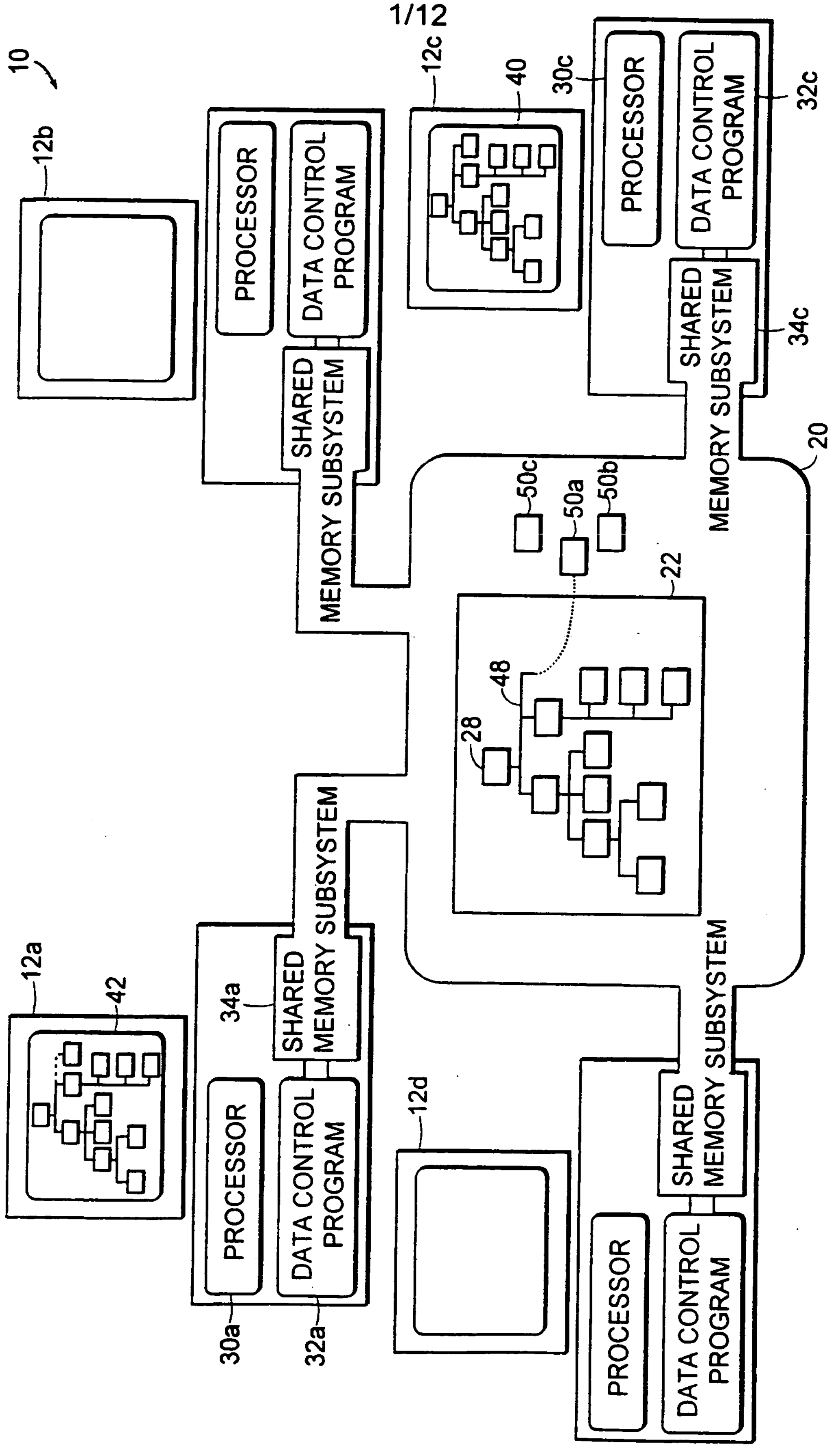

10
12b
PROCESSOR
DATA CONTROL PROGRAM
SHARED MEMORY SUBSYSTEM
12c
40
30c
PROCESSOR
DATA CONTROL PROGRAM
32c
SHARED MEMORY SUBSYSTEM
34c
20
50c
50a
50b
22
48
28
12a
42
34a
SHARED MEMORY SUBSYSTEM
PROCESSOR
DATA CONTROL PROGRAM
30a
32a
12d
SHARED MEMORY SUBSYSTEM
PROCESSOR
DATA CONTROL PROGRAM

FIG. 1

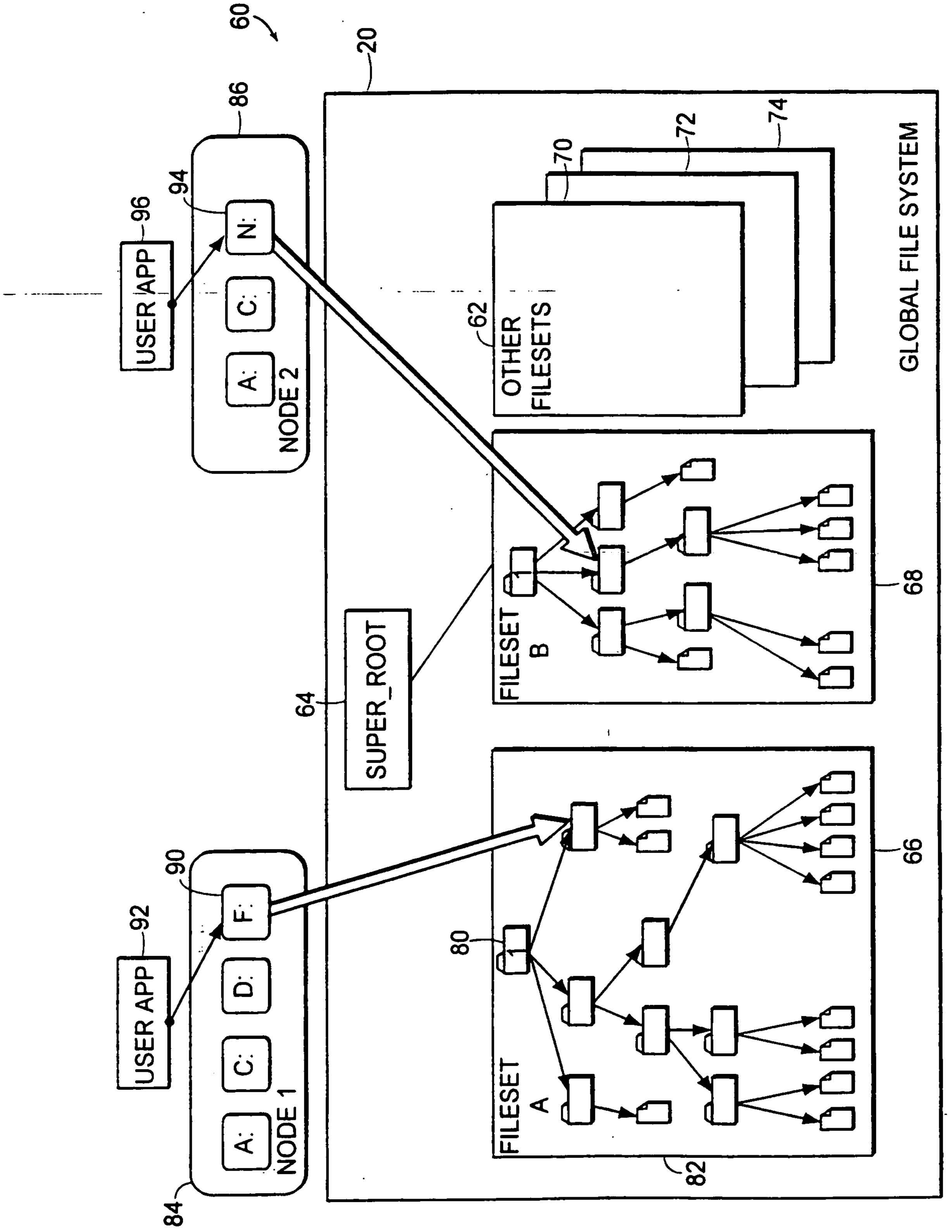

60
20
86
94
96
USER APP
N:
C:
A:
NODE 2
70
72
74
OTHER FILESETS
62
GLOBAL FILE SYSTEM
SUPER_ROOT
64
FILESET B
68
90
92
USER APP
F:
D:
C:
A:
NODE 1
84
80
FILESET A
66
82

FIG. 2

3/12

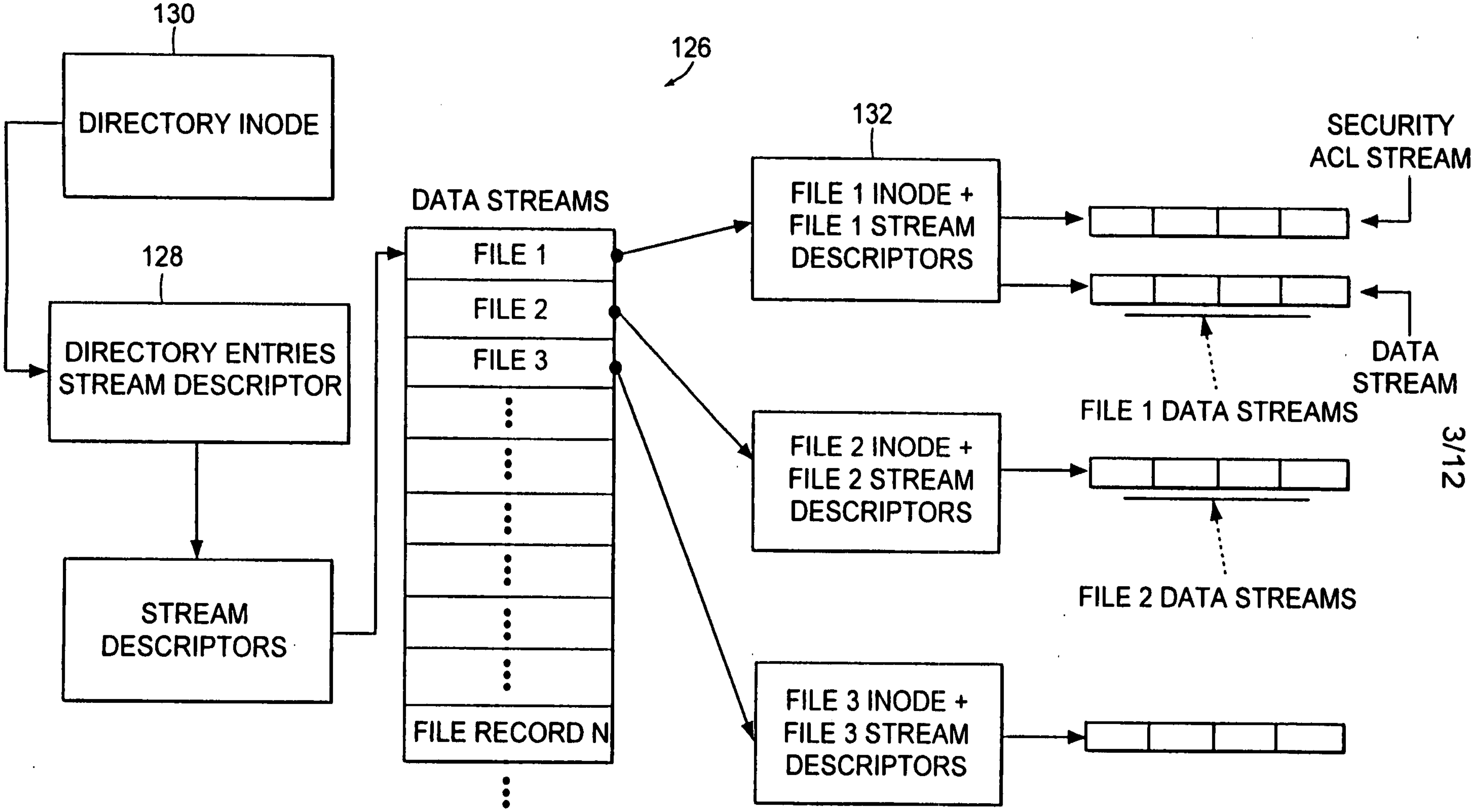

130
DIRECTORY INODE
128
DIRECTORY ENTRIES STREAM DESCRIPTOR
STREAM DESCRIPTORS
126
DATA STREAMS
FILE 1
FILE 2
FILE 3
FILE RECORD N
132
FILE 1 INODE + FILE 1 STREAM DESCRIPTORS
FILE 2 INODE + FILE 2 STREAM DESCRIPTORS
FILE 3 INODE + FILE 3 STREAM DESCRIPTORS
SECURITY ACL STREAM
DATA STREAM
FILE 1 DATA STREAMS
FILE 2 DATA STREAMS

FIG. 3

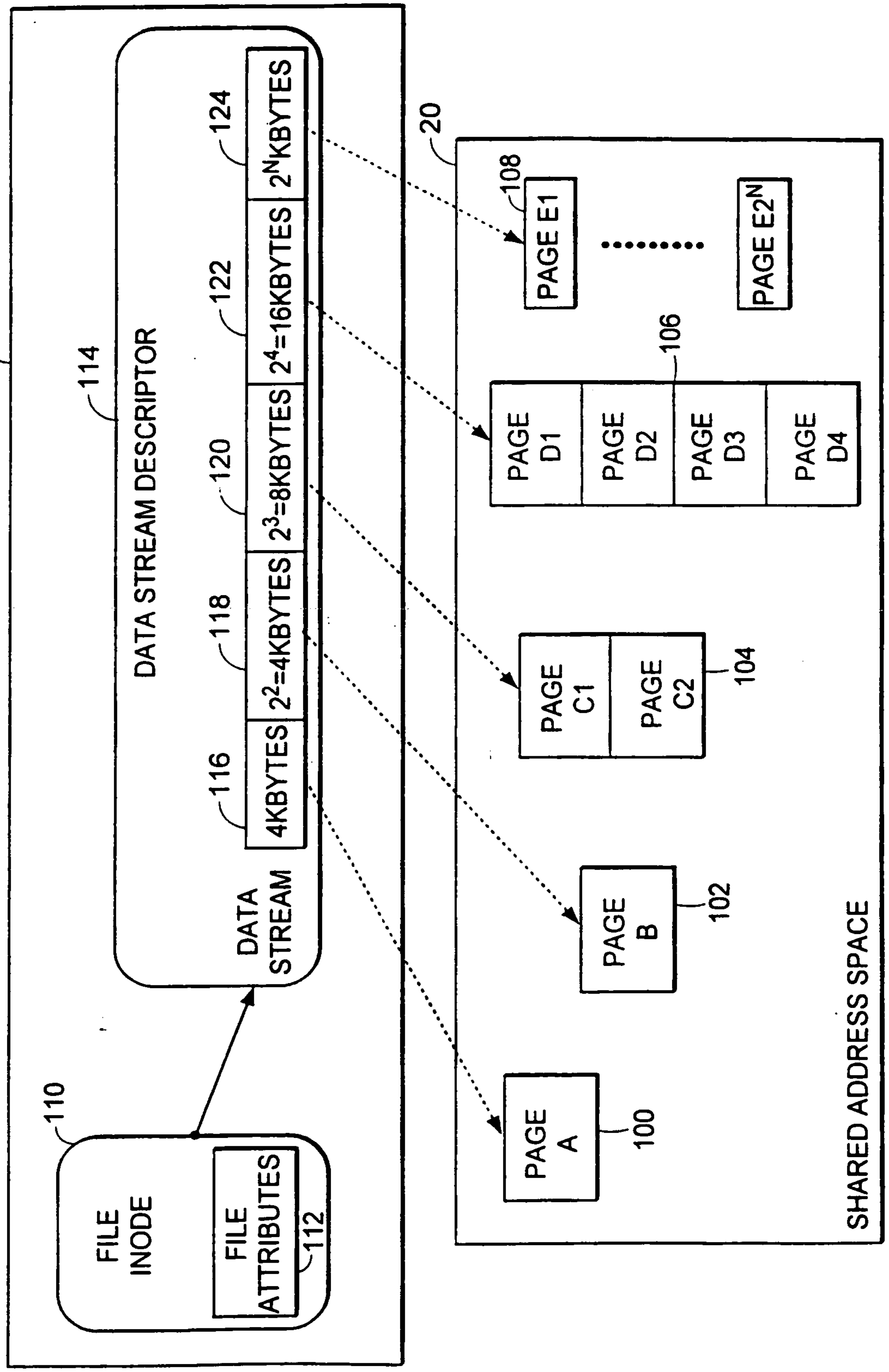

98
114
DATA STREAM DESCRIPTOR
DATA STREAM
116
4KBYTES
118
2^2=4KBYTES
120
2^3=8KBYTES
122
2^4=16KBYTES
124
2^NKBYTES
110
FILE INODE
FILE ATTRIBUTES
112
20
108
PAGE E1
PAGE E2^N
106
PAGE D1
PAGE D2
PAGE D3
PAGE D4
PAGE C1
PAGE C2
104
PAGE B
102
PAGE A
100
SHARED ADDRESS SPACE

FIG. 4

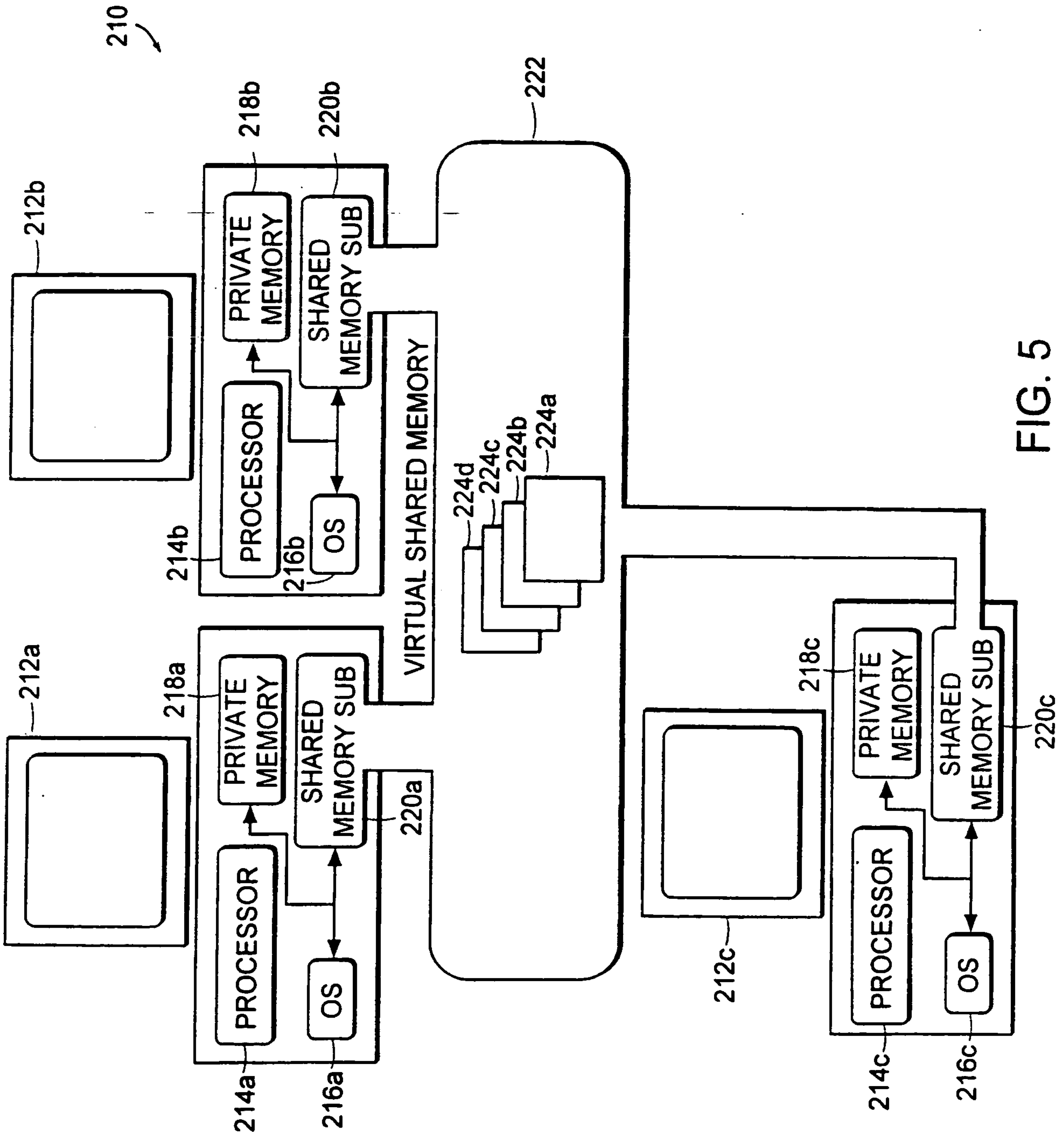

210
212a
212b
212c
PROCESSOR
PRIVATE MEMORY
SHARED MEMORY SUB
OS
214a
214b
214c
216a
216b
216c
218a
218b
218c
220a
220b
220c
222
VIRTUAL SHARED MEMORY
224a
224b
224c
224d

FIG. 5

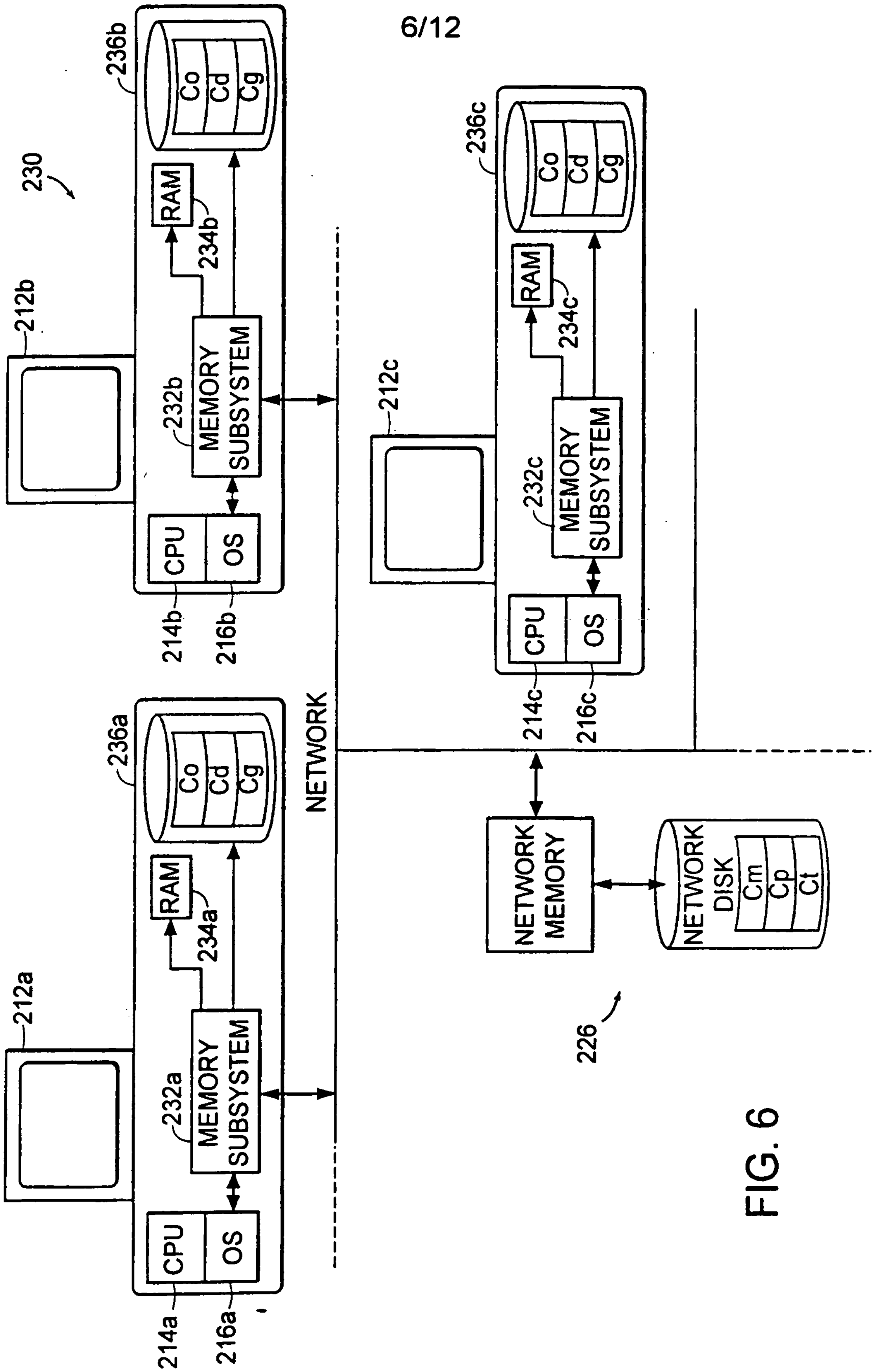

230
212b
236b
Co
Cd
Cg
RAM
234b
232b
MEMORY SUBSYSTEM
CPU
OS
214b
216b
212c
236c
Co
Cd
Cg
RAM
234c
232c
MEMORY SUBSYSTEM
CPU
OS
214c
216c
NETWORK
212a
236a
Co
Cd
Cg
RAM
234a
232a
MEMORY SUBSYSTEM
CPU
OS
214a
216a
NETWORK MEMORY
NETWORK DISK
Cm
Cp
Ct
226

FIG. 6

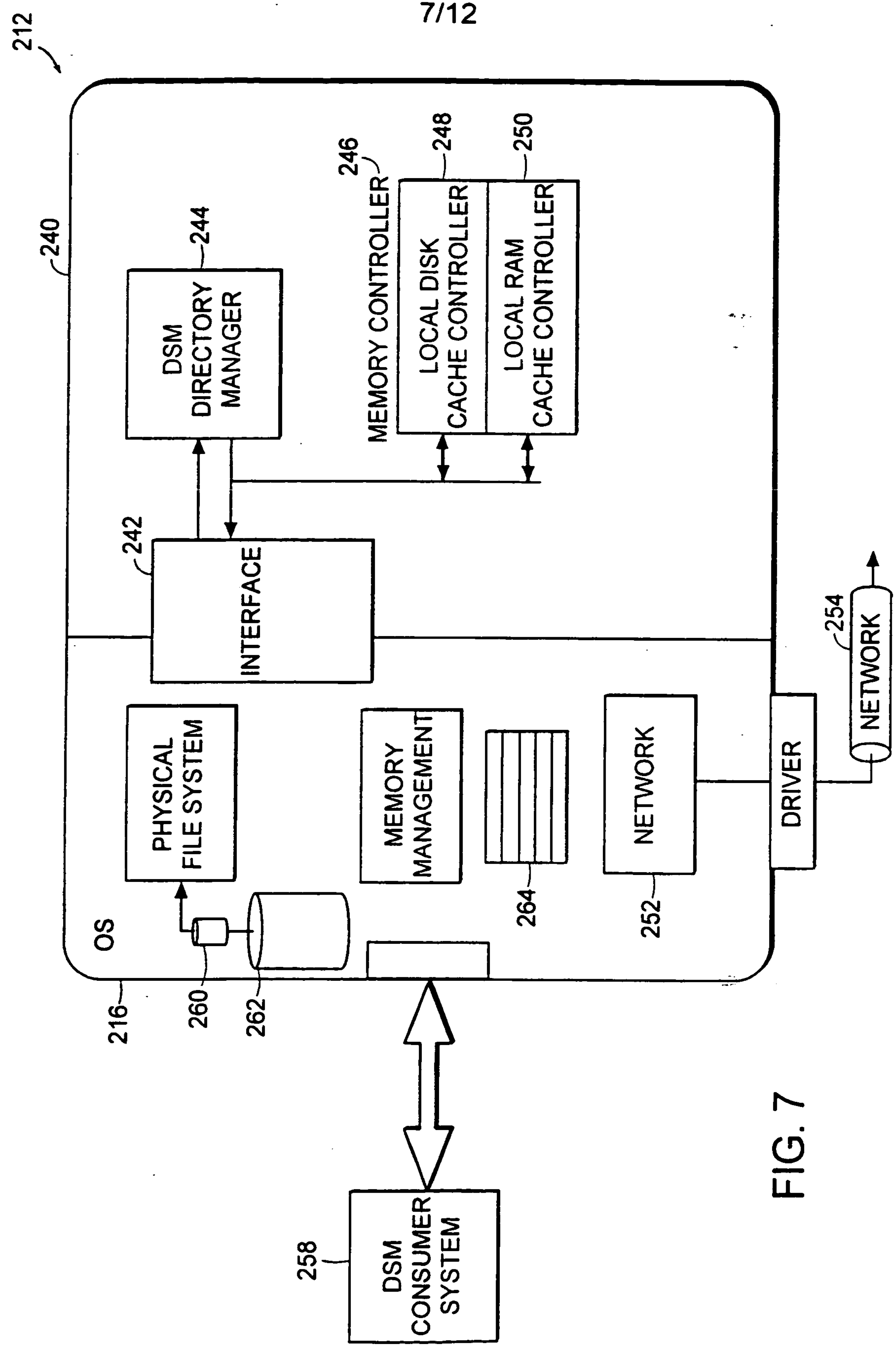

212
240
244
DSM DIRECTORY MANAGER
246
MEMORY CONTROLLER
248
LOCAL DISK CACHE CONTROLLER
250
LOCAL RAM CACHE CONTROLLER
242
INTERFACE
PHYSICAL FILE SYSTEM
MEMORY MANAGEMENT
NETWORK
DRIVER
254
NETWORK
OS
264
252
216
260
262
258
DSM CONSUMER SYSTEM

FIG. 7

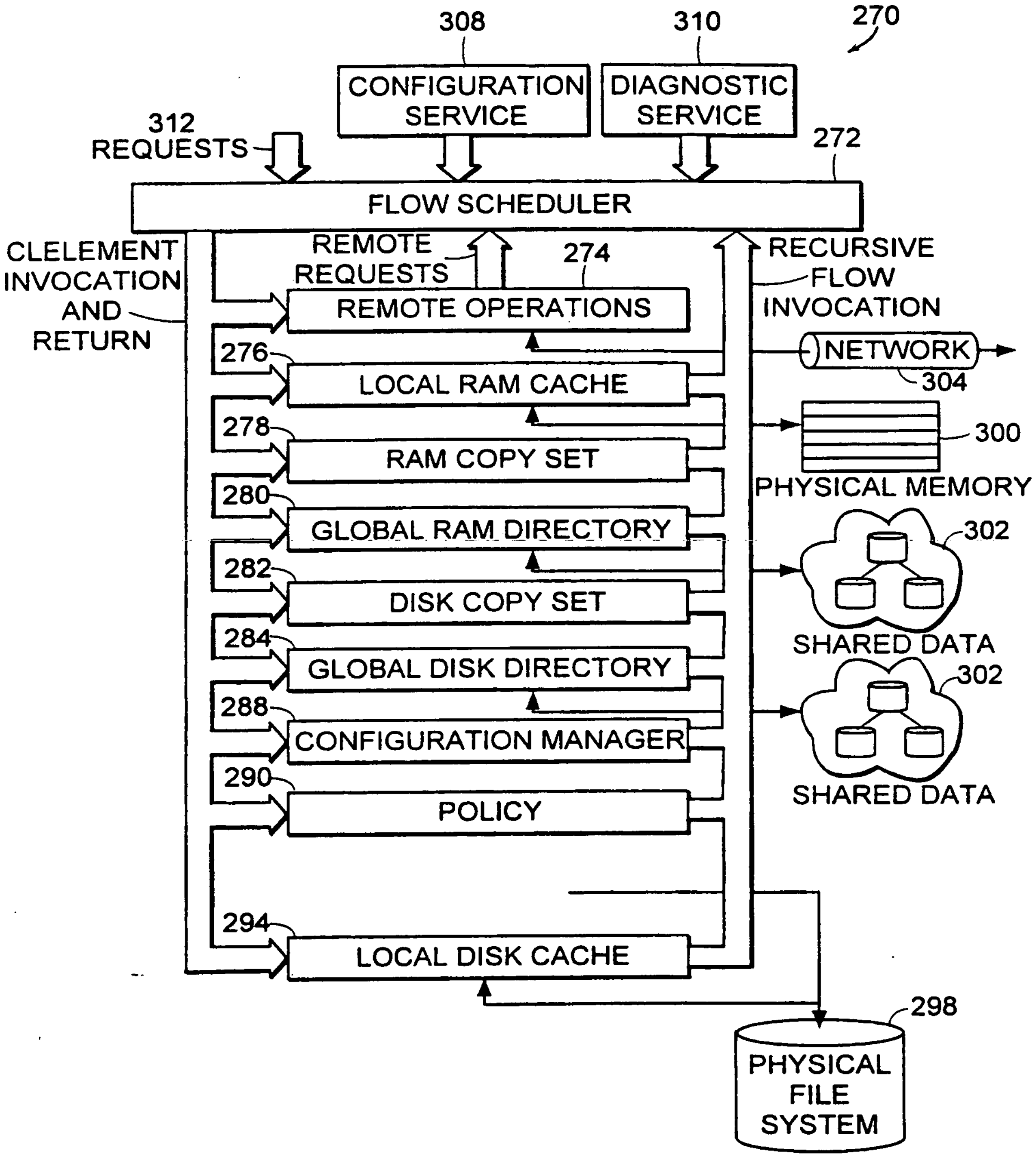

308
310
270
CONFIGURATION SERVICE
DIAGNOSTIC SERVICE
312
REQUESTS
272
FLOW SCHEDULER
CLELEMENT INVOCATION AND RETURN
REMOTE REQUESTS
274
RECURSIVE FLOW INVOCATION
REMOTE OPERATIONS
276
LOCAL RAM CACHE
NETWORK
304
278
RAM COPY SET
300
PHYSICAL MEMORY
280
GLOBAL RAM DIRECTORY
302
282
DISK COPY SET
SHARED DATA
284
GLOBAL DISK DIRECTORY
302
288
CONFIGURATION MANAGER
290
POLICY
SHARED DATA
294
LOCAL DISK CACHE
298
PHYSICAL FILE SYSTEM

FIG. 8

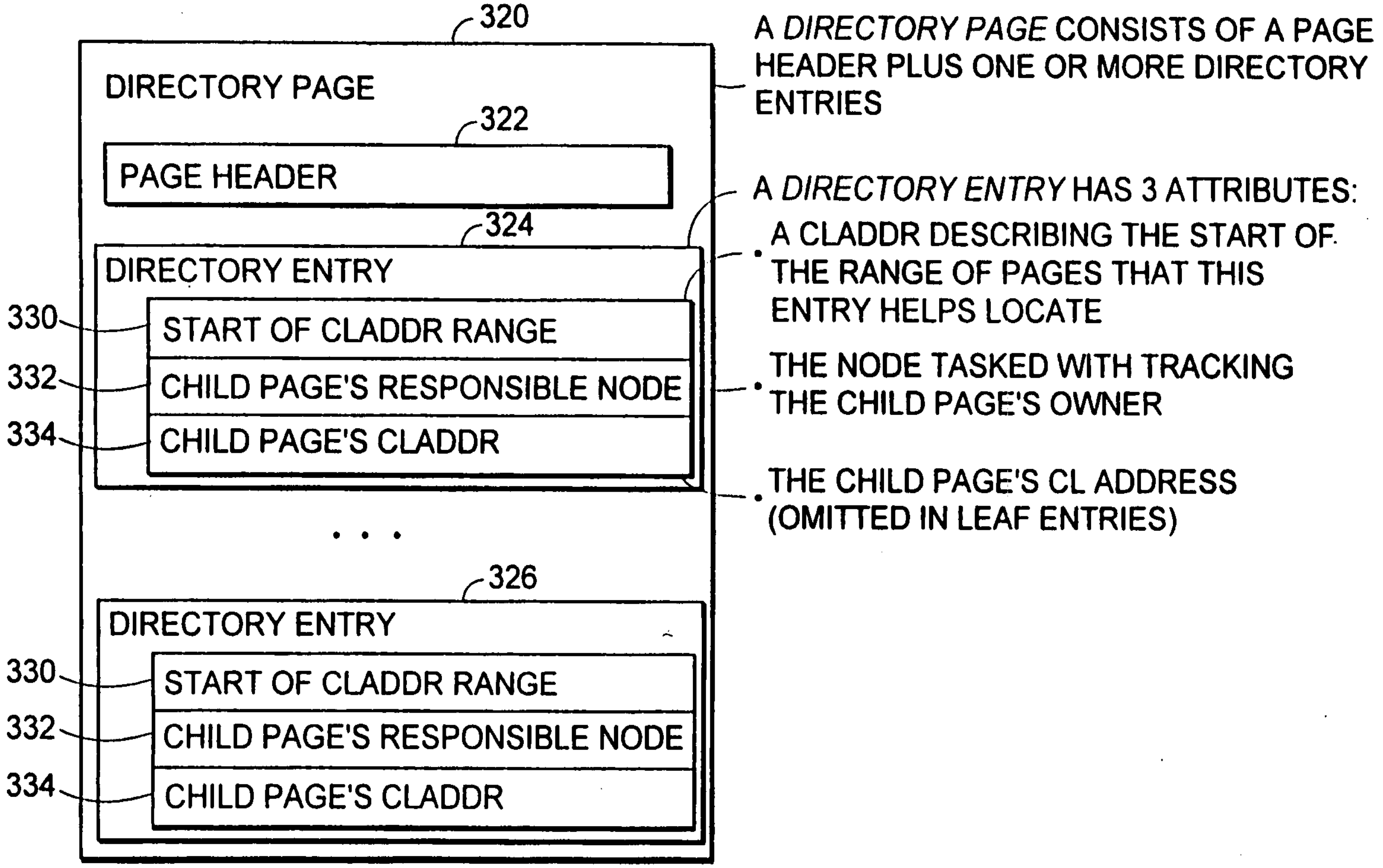

320
DIRECTORY PAGE
322
PAGE HEADER
324
DIRECTORY ENTRY
330
START OF CLADDR RANGE
332
CHILD PAGE'S RESPONSIBLE NODE
334
CHILD PAGE'S CLADDR
. . .
326
DIRECTORY ENTRY
330
START OF CLADDR RANGE
332
CHILD PAGE'S RESPONSIBLE NODE
334
CHILD PAGE'S CLADDR
A *DIRECTORY PAGE* CONSISTS OF A PAGE HEADER PLUS ONE OR MORE DIRECTORY ENTRIES
A *DIRECTORY ENTRY* HAS 3 ATTRIBUTES:
• A CLADDR DESCRIBING THE START OF THE RANGE OF PAGES THAT THIS ENTRY HELPS LOCATE
• THE NODE TASKED WITH TRACKING THE CHILD PAGE'S OWNER
• THE CHILD PAGE'S CL ADDRESS (OMITTED IN LEAF ENTRIES)

FIG. 9

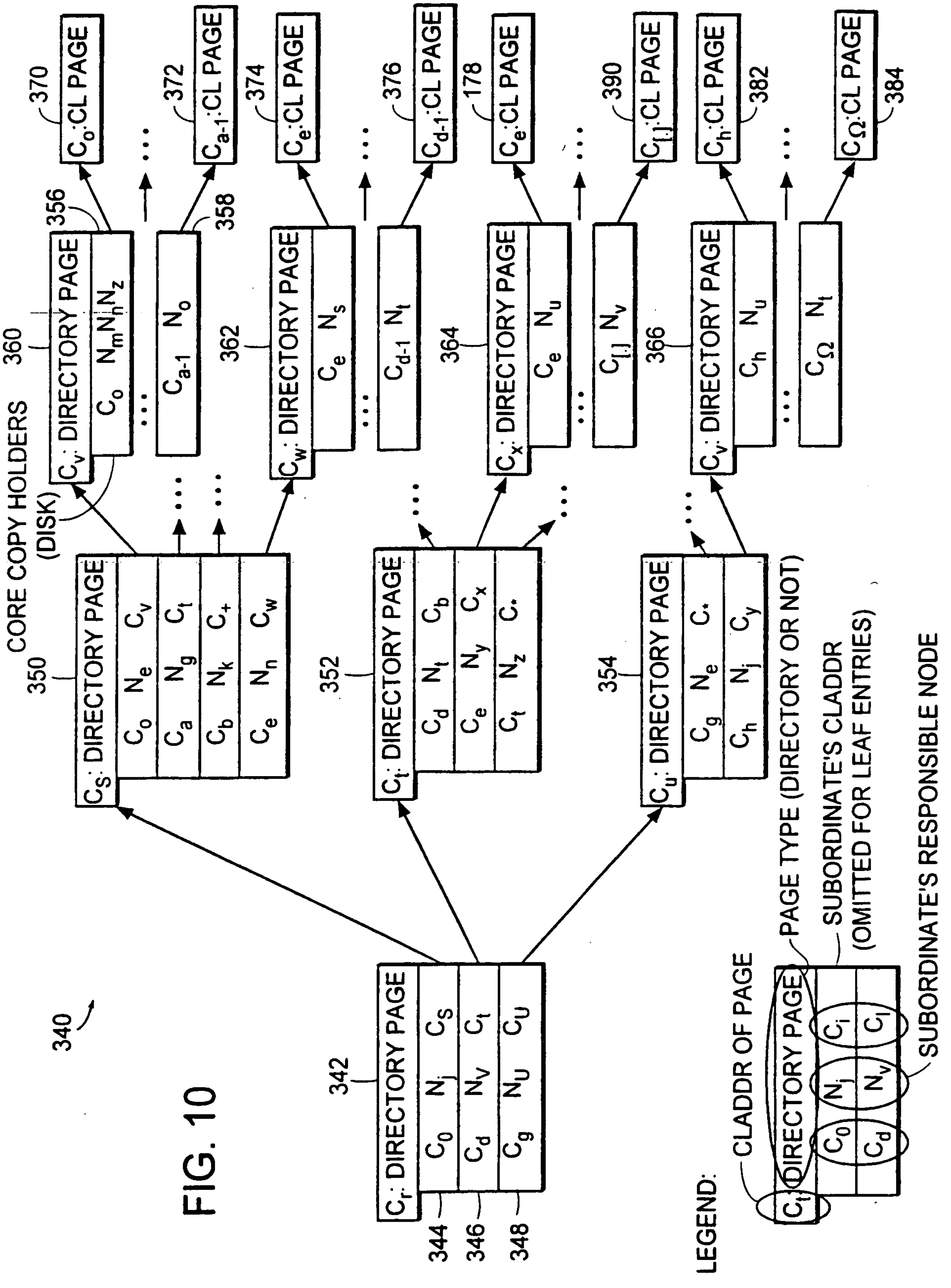


FIG. 10

SUBSTITUTE SHEET (RULE 26)

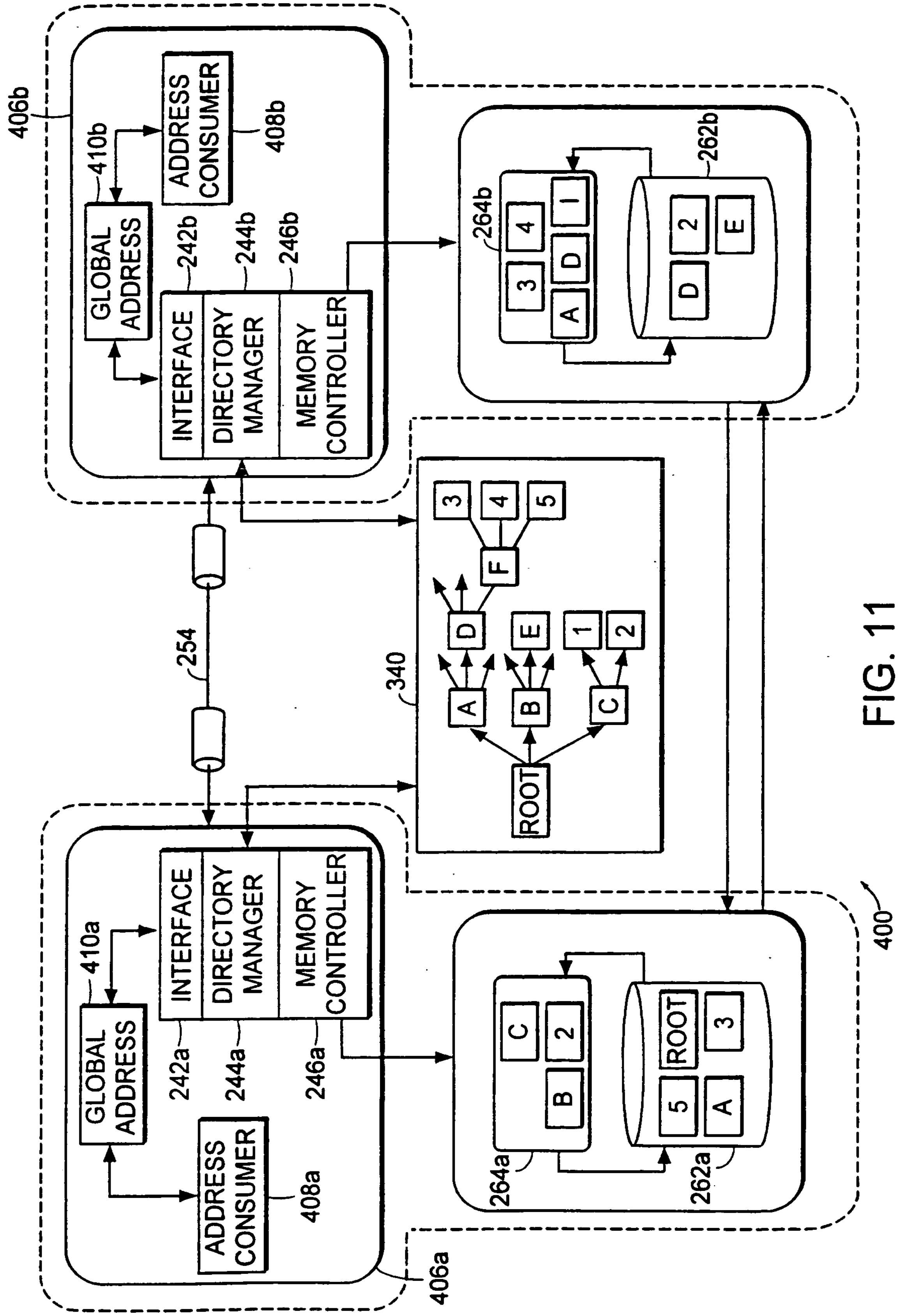

406b
410b
ADDRESS CONSUMER
408b
GLOBAL ADDRESS
242b
244b
246b
INTERFACE
DIRECTORY MANAGER
MEMORY CONTROLLER
264b
262b
3
4
A
D
I
D
2
E
254
340
3
4
5
F
D
E
1
2
A
B
C
ROOT
400
406a
410a
GLOBAL ADDRESS
INTERFACE
DIRECTORY MANAGER
MEMORY CONTROLLER
242a
244a
246a
ADDRESS CONSUMER
408a
C
2
B
ROOT
3
5
A
264a
262a

FIG. 11

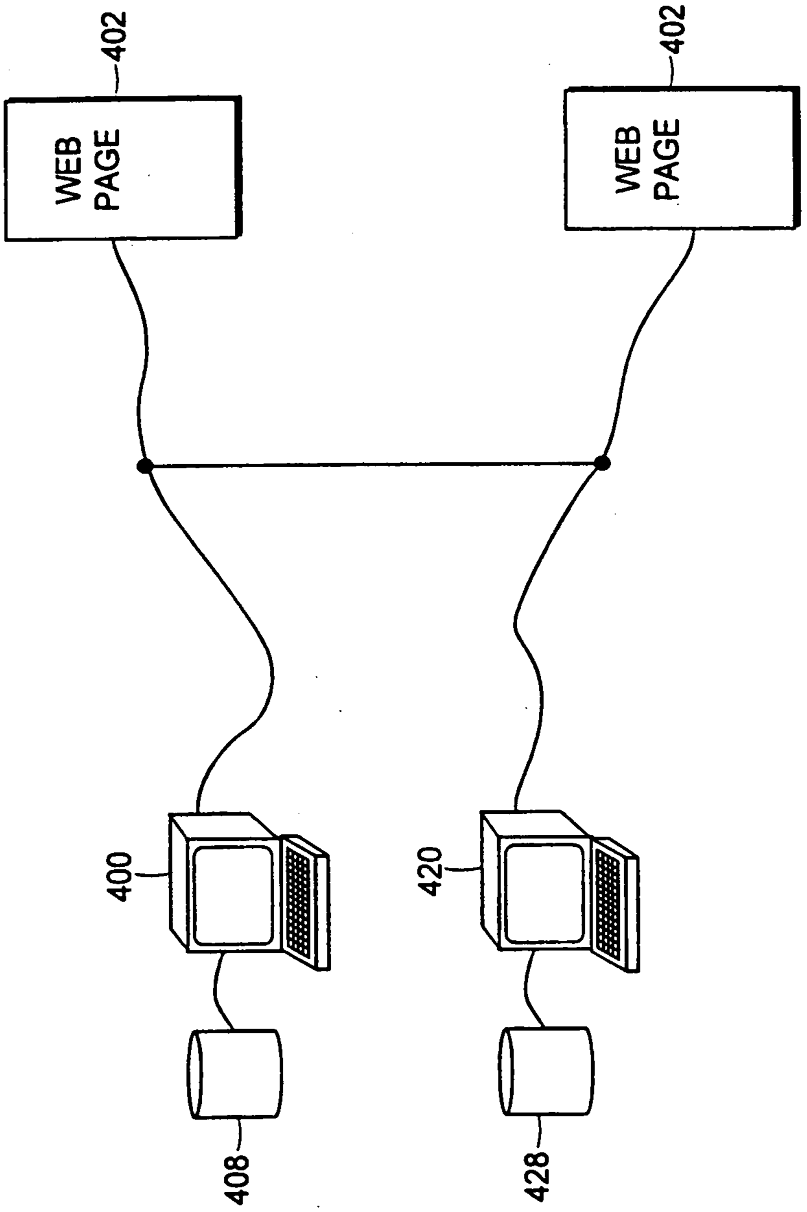


FIG. 12

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 97/21459

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06F17/30 G06F12/08

According to International Patent Classification(IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	HUBER J V JR ET AL: "PPFS: A HIGH PERFORMANCE PORTABLE PARALLEL FILE SYSTEM" PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON SUPERCOMPUTING, BARCELONA, JULY 3 - 7, 1995, 3 July 1995, ASSOCIATION FOR COMPUTING MACHINERY, pages 385-394, XP000546303	1,9
Y	see page 387, paragraph 4 - page 388; figure 1 --- -/--	2-8, 10-16

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

1

Date of the actual completion of the international search 15 April 1998	Date of mailing of the international search report 23/04/1998
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Michel, T

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 97/21459

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
Y	RAGHAVAN G K ET AL: "A DOMAIN MODEL OF WWW BROWSERS" PROCEEDINGS OF SOUTHEASTCON, BRINGING TOGETHER EDUCATION, SCIENCE AND TECHNOLOGY TAMPA, APR. 11 - 14, 1996, 11 April 1996, INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, pages 436-439, XP000631747 see the whole document ---	2-8, 10-16
A	ANDERSON T E ET AL: "SERVERLESS NETWORK FILE SYSTEMS" OPERATING SYSTEMS REVIEW (SIGOPS), vol. 29, no. 5, 1 December 1995, pages 109-126, XP000584821 see page 112, paragraph 3.1 - page 115; figure 2 ---	1,10
A	WO 95 25306 A (UNIV STANFORD) 21 September 1995 see the whole document ---	1,10
A	"JAVA DYNAMIC CLASS LOADER" IBM TECHNICAL DISCLOSURE BULLETIN, vol. 39, no. 11, November 1996, page 107/108 XP000679837 see the whole document -----	2-8, 10-16

INTERNATIONAL SEARCH REPORT

Information on patent family members

Inte. .ional Application No
PCT/US 97/21459

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9525306 A	21-09-95	NONE	

Form PCT/ISA/210 (patent family annex) (July 1992)

04-21-03

2152

ATTORNEY DOCKET NO.:
066241.0117

PATENT APPLICATION
09/759,406

#6

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Keith A. Lowery, et al.
Filing Date:	January 12, 2001
Serial No.:	09/759,406
Group Art Unit:	2152
Title:	METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING

"EXPRESS MAIL"

Express Mailing Label No. EV 193178933 US
Date of Deposit: April 17, 2003

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.

Levado Hamilton

Honorable
Assistant Commissioner for Patents
Washington, D.C. 20231

Dear Sir:

INFORMATION DISCLOSURE STATEMENT

Applicants respectfully request, pursuant to 37 C.F.R. §§ 1.56, 1.97 and 1.98, that the references listed on the attached PTO-1449 form be considered and cited in the examination of the above-identified patent application. Copies of the references are enclosed for the convenience of the Examiner. No representation is made that a search has been made, that the references are material to the patentability of the present application, or that the references qualify as prior art.

Pursuant to 37 C.F.R. § 1.97(e)(1), Applicants hereby certify that the items of information contained in this Information Disclosure Statement were cited in a communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of this Information Disclosure Statement.

Please charge any fees or credit any overpayment to Deposit Account No. 02-0384 of BAKER BOTTS L.L.P.

Respectfully submitted,

BAKER BOTTS L.L.P.
Attorneys for Applicants

Keiko Ichiye

Keiko Ichiye
Reg. No. 45,460

Date: April 17, 2003

Correspondence Address
2001 Ross Avenue
Dallas, Texas 75201-2980
Tel. 214.953.6494

PTO-1449 **Information Disclosure Citation in an Application**	Application No. 09/759,406	Applicant(s) Keith A. Lowery, et al.	
	Docket Number 066241.0117	Group Art Unit 2152	Filing Date January 12, 2001

U.S. PATENT DOCUMENTS

		DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE
	A						
	B						
	C						
	D						
	E						

FOREIGN PATENT DOCUMENTS

		DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
							YES	NO
	F							
	G							
	H							
	I							
	J							

NON-PATENT DOCUMENTS

		DOCUMENT (Including Author, Title, Source, and Pertinent Pages)	DATE
	K	PCT/US 02-00886 Search Report, 10 pages.	Mar. 27, 2003
	L	Inohara, et al., "*Self-Organizing Cooperative WWW Caching*," © 1998 IEEE (pp. 74-83).	1998
	M	Zhang, et al., "*Adaptive Web Caching*," Proceedings of the 1997 NLANR Web Cache Workshop, April 25, 1997 (9 pages).	Apr. 25, 1997
	N	Michel, et al., "*Adaptive Web Caching: Towards a New Global Caching Architecture*," © 1998 Elsevier Science B.V., Computer Networks and ISDN Systems 30 (1998) (pp. 2169-2177).	1998
	O	Karger, et al., "*Web Caching with Consistent Hashing*," © 1999 Elsevier Science B.V., Computer Networks 31 (1999)(pp. 1203-1213).	1999
	P		
	Q		
	R		
	S		
	T		

EXAMINER	**DATE CONSIDERED**

EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP § 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to the applicant.

U.S. Patent and Trademark Office

6

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/759,406	01/12/2001	Keith A. Lowery	066241.0117	2308

7590 08/05/2004

Baker Botts L.L.P.
2001 Rose Avenue
Dallas, TX 75201-2980

EXAMINER	
WON, MICHAEL YOUNG	
ART UNIT	**PAPER NUMBER**
2155	

DATE MAILED: 08/05/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

17

Office Action Summary	**Application No.** 09/759,406	**Applicant(s)** LOWERY ET AL.	
	Examiner Michael Y Won	**Art Unit** 2155	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _12 January 2001_.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1-30,95-101 and 105_ is/are pending in the application.

4a) Of the above claim(s) _31-94 and 102-104_ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1-30,95-101 and 105_ is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)	4) ☒ Interview Summary (PTO-413) Paper No(s)/Mail Date. _attached_ .
2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)	5) □ Notice of Informal Patent Application (PTO-152)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _1/12/01, 4/19/01, 7/24/02, 11/11/02_.	6) □ Other: _____.

DETAILED ACTION

1. Claims 1-105 have been examined and are pending with this action.

Election/Restrictions

2. Restriction to one of the following inventions is required under 35 U.S.C. 121:

I. Claims 1-30, 95-100, 101, and 105, drawn to Multi-computer data transferring via shared memory, classified in class 709, subclass 213.

II. Claims 31-50 and 102, drawn to Master/slave computer controlling, classified in class 709, subclass 208.

III. Claims 51-74 and 103, drawn to Accessing a remote server, classified in class 709, subclass 219.

IV. Claims 75-91 and 104, drawn to Computer to computer direct memory accessing, classified in class 709, subclass 216.

3. Because these inventions are distinct for the reasons given above and have acquired a separate status in the art as shown by their different classification, restriction for examination purposes as indicated is proper.

4. During a telephone conversation with Keiko Ichiye (Reg. No. 45,460) on July 28, 2004, a provisional election was made with traverse to prosecute the invention of group I, claims 1-30, 95-100, 101, and 105. Applicant in replying to this Office action must make affirmation of this election. Claims 31-94 and 102-103 are withdrawn from further consideration by the examiner, 37 CFR 1.142(b), as being drawn to a non-elected invention.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claims 1-30, 95-101, and 105 are rejected under 35 U.S.C. 102(e) as being anticipated by Christensen et al. (US 6,330,605 B1).

INDEPENDENT:

As per claims 1, 16, and 101, Christensen teaches of a method and a system for dynamic distributed data caching comprising logic and means for: providing a cache community (see col.2, lines 60-63) comprising at least one peer (see col.3, lines 3-6), each peer having associated first content portion indicating content to be cached (see col.1, lines 37-51) by the respective peer (see col.3, lines 17-18); allowing a client (see Fig.4 and col.6, lines 57-62) to join the cache community (see col.8, lines 39-40); updating peer list (see col.8, lines 46-51) associated with the cache community to include the client, the peer list indicating the peers the cache community (see col.5, lines 58-65 and col.6, lines 40-51); and associating a respective second content portion with each peer based on the addition the client, the second content portion being distinct from the first content portion (see col.7, lines 47-53 and col.8, lines 4-7).

As per claims 95, 98 and 105, Christensen teaches a method and a system for dynamic distributed data caching comprising logic and means for: communicating a community request from a dynamic cache module to an administration module (see col.6, lines 26-36); receiving a community list (see col.8, lines 46-51) from the administration module in response the community request, the community list including a list of communities (see col.6, lines 37-43); generating a join request to attempt to join a one of the communities in the community list (see col.8, lines 39-43); receiving an allow message associated with the one of the communities (see col.6, lines 23-28); receiving a peer list associated with the one the communities (see col.8, line 46-51);

receiving a content request (see col.1, lines 37-51 and col.2, line 65 to col.3, line 13); and storing content associated with the content request (see col.1, lines 37-51).

DEPENDENT:

As per claims 2 and 17, Christensen teaches of further comprising: receiving a join request from the client (see col.8, lines 39-41); and determining whether allow the client join the cache community (implicit: see col.8, lines 41-43).

As per claims 3, 18, 97, and 100, Christensen further teaches wherein the join request comprises a CRMSG_REQUESTTOJOIN data message (see col.8, lines 39-43).

As per claims 4 and 19, Christensen further teaches wherein allowing the client to join the cache community comprises: generating an allow message (see col.6, lines 23-28); associating the peer with the allow message (see col.6, lines 40-51); and communicating the allow message to the client (implicit: see col.6, lines 23-28).

As per claims 5 and 20, Christensen further teaches wherein allowing the client to join the cache community comprises: generating an allow message comprising the peer list updated to include the clients (see col.5, lines 58-65 and col.6, lines 23-28); communicating the allow message to the client (implicit: see col.6, lines 23-28); and communicating the allow message to at least one member associated with the cache community (see col.6, lines 34-36 and col.7, lines 47-53).

As per claims 6 and 21, Christensen further teaches wherein the allow message comprises a CRMSG_UPDATEPEERLIST data message (see col.5, lines 58-65 and col.6, lines 40-43).

As per claims 7 and 22, Christensen further teaches wherein peer list associated with the allow message comprises updated peer which includes the client (see col.6, lines 23-36 and col.7, lines 47-53).

As per claims 8 and 23, Christensen further teaches wherein the peer comprises a computer (see col.6, lines 57-62).

As per claims 9 and 24, Christensen further teaches wherein a one of the peers comprises a member (see col.7, lines 47-49).

As per claims 10 and 25, Christensen further teaches wherein one of the peers comprises a master (see col.6, lines 37-40).

As per claims 11 and 26, Christensen further teaches wherein associating a respective second content portion comprises: allocating respective second content portions peers in the peer list (see col.5, lines 58-65 and col.7, lines 47-49); and updating an allocation table indicate the second content portion associated with the peers (see col.7, lines 47-49).

As per claims 12 and 27, Christensen further teaches wherein the second content portions are distinct (see col.21, lines 26-32).

As per claims 13 and 28, Christensen further teaches wherein at least two of the second content portions overlap (see col.21, lines 32-40).

As per claims 14 and 29, Christensen further teaches wherein the first and second content portions respectively comprise a plurality Internet Protocol domain names (see col.1, line 64 to col.2, line 5).

As per claims 15 and 30, Christensen teaches of further comprising removing the association between the first content portions and the peers (see col.9, lines 58-63).

As per claims 96 and 99, further teaches wherein the community request comprises a CRMSG_WAKEUP data message (see col.6, lines 28-36).

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael Y Won whose telephone number is 703-605-4241. The examiner can normally be reached on M-Th: 6AM-3PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hosain T Alam can be reached on 703-308-6662. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Michael Y Won

July 28, 2004

HOSAIN ALAM
UPERVISORY PATENT EXAMINER

Notice of References Cited	Application/Control No. 09/759,406	Applicant(s)/Patent Under Reexamination LOWERY ET AL.	
	Examiner Michael Y Won	Art Unit 2155	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-5,778,185 A	07-1998	Gregerson et al.	709/226
	B	US-5,864,854 A	01-1999	Boyle, Douglas B.	707/10
	C	US-5,940,838 A	08-1999	Schmuck et al.	707/200
	D	US-5,968,176 A	10-1999	Nessett et al.	713/201
	E	US-5,987,477 A	11-1999	Schmuck et al.	707/201
	F	US-5,988,847 A	11-1999	McLaughlin et al.	700/9
	G	US-6,006,254 A	12-1999	Waters et al.	709/205
	H	US-6,098,064 A	08-2000	Pirolli et al.	707/2
	I	US-6,112,279 A	08-2000	Wang, Zheng	711/119
	J	US-6,167,438 A	12-2000	Yates et al.	709/216
	K	US-6,205,481 B1	03-2001	Heddaya et al.	709/226
	L	US-6,487,583 B1	11-2002	Harvey et al.	709/204
	M	US-6,542,926 B2	04-2003	Zalewski et al.	709/213

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

PTO-1449 **Information Disclosure Citation in an Application**	Application No.	Applicant(s) Keith A. Lowery, et al.	
	Docket Number 066241.0117	Group Art Unit	Filing Date January 12, 2001

U.S. PATENT DOCUMENTS

		DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE
	A						
	B						
	C						
	D						
	E						

FOREIGN PATENT DOCUMENTS

		DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
							YES	NO
	F							

NON-PATENT DOCUMENTS

		DOCUMENT (Including Author, Title, Source, and Pertinent Pages)	DATE
	G	Author: John Borland, Article entitled "Net video not yet ready for prime time", printed from web site www.cnet.com on August 15, 2000.	February 5, 1999
	H	Author: Corey Grice, Article entitled "Start-up taps swapping tech to ease Web bottlenecks", printed from web site www.cnet.com on August 15, 2000.	August 11, 2000
	I	Author: VTrails.com. Product information regarding company's Full Duplex Packet Cascading technology printed from web site www.vtrails.com on August 15, 2000.	2000
	J	Author: Vinod Valloppillil and Keith W. Ross, Internet-Draft document entitled "Cache Array Routing Protocol v1.0", printed from web site www.globecom.net on November 14, 2000.	February 26, 1998
	K	Author: Microsoft Corporation. Information regarding the "Cache Array Routing Protocol (CARP) and Microsoft Proxy Server 2.0, printed from web site www.msdn.microsoft.com on October 9, 2000.	1997
	L	Author: Geek.com. Information news article entitled "Cash for your unused CPU cycles", printed from web site www.geek.com on June 30, 2000.	June 30, 2000
	M	Author: Dcypher.net. Answers to frequently answered questions v1.5, printed from web site www.dcypher.net on June 30, 2000.	April 10, 2000
	N	Author: Popular Power. Information regarding the company's Popular Power for Windows software, printed from web site www.popularpower.com on June 30, 2000.	None given.
	O	Author: ProcessTree Network. Information regarding their "for-pay" distributed processing network, printed from web site www.distributedscience.com on June 30, 2000.	None given.
	P	Author: Keith Schultz, Article entitled "Pushing content to the Internet's Edge", printed from web site www.internetweek.com on January 11, 2001.	December 4, 2000

EXAMINER	DATE CONSIDERED 7/21/04

EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP § 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to the applicant.

U.S. Patent and Trademark Office

Page 1 of 1

PTO-1449 **Information Disclosure Citation in an Application**	Application No. 09/759,406	Applicant(s) Keith A. Lowery, et al.	
	Docket Number 066241.0117	Group Art Unit 2185	Filing Date January 12, 2001

U.S. PATENT DOCUMENTS

		DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE
mw	A	6,026,474	02/15/00	Carter et al.	711	202	05/02/97
	B						
	C						
	D						
	E						

FOREIGN PATENT DOCUMENTS

		DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
							YES	NO
	F							

NON-PATENT DOCUMENTS

		DOCUMENT (Including Author, Title, Source, and Pertinent Pages)	DATE
	G		
	H		
	I		
	J		
	K		
	L		
	M		
	N		
	O		
	P		

EXAMINER	DATE CONSIDERED 7/21/04

EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP § 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to the applicant.

U.S. Patent and Trademark Office

DAL01:597455.1
066241.0117

Page 1 of 1

PTO-1449 **Information Disclosure Citation in an Application**	**Application No.** 09/759,406	**Applicant(s)** Keith A. Lowery, et al.	
	Docket Number 066241.0117	**Group Art Unit** 2152	**Filing Date** January 12, 2001

U.S. PATENT DOCUMENTS

		DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE
	A						
	B						
	C						
	D						
	E						

FOREIGN PATENT DOCUMENTS

		DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
							YES	NO
	F							

NON-PATENT DOCUMENTS

		DOCUMENT (Including Author, Title, Source, and Pertinent Pages)	DATE
	G	Patent Application Serial No. 09/759,392 filed January 12, 2001, entitled "*Method And System For Community Data Caching*," (Attorney's Docket 066241.0110), 78 total pages	1/12/2001
	H		
	I		
	J		
	K		
	L		
	M		
	N		
	O		
	P		

EXAMINER	DATE CONSIDERED 7/21/04

EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP § 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to the applicant.

U.S. Patent and Trademark Office

Page 1 of 1

PTO-1449 **Information Disclosure Citation in an Application**	Application No. 09/759,406	Applicant(s) Keith A. Lowery, et al.	
	D cket Number 066241.0117	Group Art Unit 2152	Filing Date January 12, 2001

U.S. PATENT DOCUMENTS

		DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE
	A						
	B						
	C						
	D						
	E						

FOREIGN PATENT DOCUMENTS

		DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
							YES	NO
mw	F	EP 0993 163 A1	04/12/00	EPO	H04L	29/06	X	
mw	G	WO 98/22891	05/28/98	PCT	G06F	17/30	X	
	H							
	I							
	J							

NON-PATENT DOCUMENTS

		DOCUMENT (Including Author, Title, Source, and Pertinent Pages)	DATE
mw	K	Greg Barish, et al., "World Wide Web Caching: Trends and Techniques," XP-000949799, IEEE Communications Magazine, May 2000, pp. 178-185.	05/2000
mw	L	PCT/US 02-00886 Search Report, 6 pages.	10/2000
	M		
	N		
	O		
	P		
	Q		
	R		
	S		
	T		

EXAMINER	DATE CONSIDERED 7/21/04

EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP § 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to the applicant.

U.S. Patent and Trademark Office

DAL01:704443.1

PTO-1449 **Information Disclosure Citation in an Application**	Application No. 09/759,406	Applicant(s) Keith A. Lowery, et al.
	Docket Number 066241.0117	Group Art Unit 2152
		Filing Date January 12, 2001

U.S. PATENT DOCUMENTS

		DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS	FILING DATE
	A						
	B						
	C						
	D						
	E						

FOREIGN PATENT DOCUMENTS

		DOCUMENT NO.	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
							YES	NO
	F							
	G							
	H							
	I							
	J							

NON-PATENT DOCUMENTS

		DOCUMENT (Including Author, Title, Source, and Pertinent Pages)	DATE
nw	K	PCT/US 02-00886 Search Report, 10 pages.	Mar. 27, 2003
nw	L	Inohara, et al., "*Self-Organizing Cooperative WWW Caching,*" © 1998 IEEE (pp. 74-83).	1998
nw	M	Zhang, et al., "*Adaptive Web Caching,*" Proceedings of the 1997 NLANR Web Cache Workshop, April 25, 1997 (9 pages).	Apr. 25, 1997
nw	N	Michel, et al., "*Adaptive Web Caching: Towards a New Global Caching Architecture,*" © 1998 Elsevier Science B.V., Computer Networks and ISDN Systems 30 (1998) (pp. 2169-2177).	1998
nw	O	Karger, et al., "*Web Caching with Consistent Hashing,*" © 1999 Elsevier Science B.V., Computer Networks 31 (1999)(pp. 1203-1213).	1999
	P		
	Q		
	R		
	S		
	T		

EXAMINER	**DATE CONSIDERED** 7/21/04

EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP § 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to the applicant.

U.S. Patent and Trademark Office

Interview Summary	**Application No.** 09/759,406	**Applicant(s)** LOWERY ET AL.	
	Examiner Michael Y Won	**Art Unit** 2155	

All participants (applicant, applicant's representative, PTO personnel):

(1) *Michael Y Won*. (3)_____.

(2) *Keiko Ichiye (Applicant's Representative)*. (4)_____.

Date of Interview: *28 July 2004*.

Type: a)☒ Telephonic b)□ Video Conference
c)□ Personal [copy given to: 1)□ applicant 2)□ applicant's representative]

Exhibit shown or demonstration conducted: d)□ Yes e)□ No.
If Yes, brief description: _____.

Claim(s) discussed: *1-105*.

Identification of prior art discussed: *none*.

Agreement with respect to the claims f)☒ was reached. g)□ was not reached. h)□ N/A.

Substance of Interview including description of the general nature of what was agreed to if an agreement was reached, or any other comments: *Examiner called Ms. Ichiye on July 22, regarding a restriction and asked to elect from the different groups. Ms. Ichiye called back on July 28, 2004 and left a message electing group one*.

(A fuller description, if necessary, and a copy of the amendments which the examiner agreed would render the claims allowable, if available, must be attached. Also, where no copy of the amendments that would render the claims allowable is available, a summary thereof must be attached.)

THE FORMAL WRITTEN REPLY TO THE LAST OFFICE ACTION MUST INCLUDE THE SUBSTANCE OF THE INTERVIEW. (See MPEP Section 713.04). If a reply to the last Office action has already been filed, APPLICANT IS GIVEN ONE MONTH FROM THIS INTERVIEW DATE, OR THE MAILING DATE OF THIS INTERVIEW SUMMARY FORM, WHICHEVER IS LATER, TO FILE A STATEMENT OF THE SUBSTANCE OF THE INTERVIEW. See Summary of Record of Interview requirements on reverse side or on attached sheet.

Examiner Note: You must sign this form unless it is an Attachment to a signed Office action.

Examiner's signature, if required

Summary of Record of Interview Requirements

Manual of Patent Examining Procedure (MPEP), Section 713.04, Substance of Interview Must be Made of Record
A complete written statement as to the substance of any face-to-face, video conference, or telephone interview with regard to an application must be made of record in the application whether or not an agreement with the examiner was reached at the interview.

Title 37 Code of Federal Regulations (CFR) § 1.133 Interviews
Paragraph (b)
In every instance where reconsideration is requested in view of an interview with an examiner, a complete written statement of the reasons presented at the interview as warranting favorable action must be filed by the applicant. An interview does not remove the necessity for reply to Office action as specified in §§ 1.111, 1.135. (35 U.S.C. 132)

37 CFR §1.2 Business to be transacted in writing.
All business with the Patent or Trademark Office should be transacted in writing. The personal attendance of applicants or their attorneys or agents at the Patent and Trademark Office is unnecessary. The action of the Patent and Trademark Office will be based exclusively on the written record in the Office. No attention will be paid to any alleged oral promise, stipulation, or understanding in relation to which there is disagreement or doubt.

The action of the Patent and Trademark Office cannot be based exclusively on the written record in the Office if that record is itself incomplete through the failure to record the substance of interviews.

It is the responsibility of the applicant or the attorney or agent to make the substance of an interview of record in the application file, unless the examiner indicates he or she will do so. It is the examiner's responsibility to see that such a record is made and to correct material inaccuracies which bear directly on the question of patentability.

Examiners must complete an Interview Summary Form for each interview held where a matter of substance has been discussed during the interview by checking the appropriate boxes and filling in the blanks. Discussions regarding only procedural matters, directed solely to restriction requirements for which interview recordation is otherwise provided for in Section 812.01 of the Manual of Patent Examining Procedure, or pointing out typographical errors or unreadable script in Office actions or the like, are excluded from the interview recordation procedures below. Where the substance of an interview is completely recorded in an Examiners Amendment, no separate Interview Summary Record is required.

The Interview Summary Form shall be given an appropriate Paper No., placed in the right hand portion of the file, and listed on the "Contents" section of the file wrapper. In a personal interview, a duplicate of the Form is given to the applicant (or attorney or agent) at the conclusion of the interview. In the case of a telephone or video-conference interview, the copy is mailed to the applicant's correspondence address either with or prior to the next official communication. If additional correspondence from the examiner is not likely before an allowance or if other circumstances dictate, the Form should be mailed promptly after the interview rather than with the next official communication.

The Form provides for recordation of the following information:
- Application Number (Series Code and Serial Number)
- Name of applicant
- Name of examiner
- Date of interview
- Type of interview (telephonic, video-conference, or personal)
- Name of participant(s) (applicant, attorney or agent, examiner, other PTO personnel, etc.)
- An indication whether or not an exhibit was shown or a demonstration conducted
- An identification of the specific prior art discussed
- An indication whether an agreement was reached and if so, a description of the general nature of the agreement (may be by attachment of a copy of amendments or claims agreed as being allowable). Note: Agreement as to allowability is tentative and does not restrict further action by the examiner to the contrary.
- The signature of the examiner who conducted the interview (if Form is not an attachment to a signed Office action)

It is desirable that the examiner orally remind the applicant of his or her obligation to record the substance of the interview of each case. It should be noted, however, that the Interview Summary Form will not normally be considered a complete and proper recordation of the interview unless it includes, or is supplemented by the applicant or the examiner to include, all of the applicable items required below concerning the substance of the interview.

A complete and proper recordation of the substance of any interview should include at least the following applicable items:
1) A brief description of the nature of any exhibit shown or any demonstration conducted,
2) an identification of the claims discussed,
3) an identification of the specific prior art discussed,
4) an identification of the principal proposed amendments of a substantive nature discussed, unless these are already described on the Interview Summary Form completed by the Examiner,
5) a brief identification of the general thrust of the principal arguments presented to the examiner,
   (The identification of arguments need not be lengthy or elaborate. A verbatim or highly detailed description of the arguments is not required. The identification of the arguments is sufficient if the general nature or thrust of the principal arguments made to the examiner can be understood in the context of the application file. Of course, the applicant may desire to emphasize and fully describe those arguments which he or she feels were or might be persuasive to the examiner.)
6) a general indication of any other pertinent matters discussed, and
7) if appropriate, the general results or outcome of the interview unless already described in the Interview Summary Form completed by the examiner.

Examiners are expected to carefully review the applicant's record of the substance of an interview. If the record is not complete and accurate, the examiner will give the applicant an extendable one month time period to correct the record.

Examiner to Check for Accuracy

If the claims are allowable for other reasons of record, the examiner should send a letter setting forth the examiner's version of the statement attributed to him or her. If the record is complete and accurate, the examiner should place the indication, "Interview Record OK" on the paper recording the substance of the interview along with the date and the examiner's initials.

Index of Claims	Application No. 09/759,406	Applicant(s) LOWERY ET AL.
	Examiner Michael Y Won	Art Unit 2155

√	Rejected	–	(Through numeral) Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

Claim		Date								
Final	Original	7 28 04								
	1	✓								
	2	↑								
	3	│								
	4	│								
	5	│								
	6	│								
	7	│								
	8	│								
	9	│								
	10	│								
	11	│								
	12	│								
	13	│								
	14	│								
	15	│								
	16	│								
	17	│								
	18	│								
	19	│								
	20	│								
	21	│								
	22	│								
	23	│								
	24	│								
	25	│								
	26	│								
	27	│								
	28	│								
	29	↓								
	30	✓								
	~~31~~									
	~~32~~									
	~~33~~									
	~~34~~									
	~~35~~									
	~~36~~									
	~~37~~									
	~~38~~									
	~~39~~									
	~~40~~									
	~~41~~									
	~~42~~									
	~~43~~									
	~~44~~									
	~~45~~									
	~~46~~									
	~~47~~									
	~~48~~									
	~~49~~									
	~~50~~									

Claim		Date								
Final	Original	7 28 04								
	~~51~~									
	~~52~~									
	~~53~~									
	~~54~~									
	~~55~~									
	~~56~~									
	~~57~~									
	~~58~~									
	~~59~~									
	~~60~~									
	~~61~~									
	~~62~~									
	~~63~~									
	~~64~~									
	~~65~~									
	~~66~~									
	~~67~~									
	~~68~~									
	~~69~~									
	~~70~~									
	~~71~~									
	~~72~~									
	~~73~~									
	~~74~~									
	~~75~~									
	~~76~~									
	~~77~~									
	~~78~~									
	~~79~~									
	~~80~~									
	~~81~~									
	~~82~~									
	~~83~~									
	~~84~~									
	~~85~~									
	~~86~~									
	~~87~~									
	~~88~~									
	~~89~~									
	~~90~~									
	~~91~~									
	~~92~~									
	~~93~~									
	~~94~~									
	95	✓								
	96	↑								
	97	│								
	98	│								
	99	↓								
	100	✓								

Claim		Date								
Final	Original	7 28 04								
	101	✓								
	~~102~~									
	~~103~~									
	~~104~~									
	105	✓								
	106									
	107									
	108									
	109									
	110									
	111									
	112									
	113									
	114									
	115									
	116									
	117									
	118									
	119									
	120									
	121									
	122									
	123									
	124									
	125									
	126									
	127									
	128									
	129									
	130									
	131									
	132									
	133									
	134									
	135									
	136									
	137									
	138									
	139									
	140									
	141									
	142									
	143									
	144									
	145									
	146									
	147									
	148									
	149									
	150									

UNITED STATES PATENT AND TRADEMARK OFFICE

COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
WASHINGTON, D.C. 20231
www.uspto.gov

Bib Data Sheet

CONFIRMATION NO. 2308

SERIAL NUMBER 09/759,406	**FILING DATE** 01/12/2001 **RULE**	**CLASS** 709	**GROUP ART UNIT** 2152	**ATTORNEY DOCKET NO.** 066241.0117

APPLICANTS

Keith A. Lowery, Richardson, TX;
Bryan S. Chin, Plano, TX;
David A. Consolver, Arlington, TX;
Gregg A. DeMasters, Plano, TX;

** **CONTINUING DATA** ***************************** yw, none

** **FOREIGN APPLICATIONS** ************************ yw, none

IF REQUIRED, FOREIGN FILING LICENSE GRANTED ** SMALL ENTITY **
** 02/23/2001

Foreign Priority claimed	□ yes ☑ no	**STATE OR COUNTRY** TX	**SHEETS DRAWING** 11	**TOTAL CLAIMS** 105	**INDEPENDENT CLAIMS** 15
35 USC 119 (a-d) conditions met	□ yes □ no □ Met after Allowance				
Verified and Acknowledged	Examiner's Signature yw Initials				

ADDRESS

Baker Botts L.L.P.
2001 Rose Avenue
Dallas ,TX 75201-2980

TITLE

Method and system for dynamic distributed data caching

FILING FEE RECEIVED 1600	FEES: Authority has been given in Paper No. ________ to charge/credit DEPOSIT ACCOUNT No. ________ for following:	□ All Fees □ 1.16 Fees (Filing) □ 1.17 Fees (Processing Ext. of time) □ 1.18 Fees (Issue) □ Other ____________ □ Credit

BEST AVAILABLE COPY

Search Notes	Application No. 09/759,406	Applicant(s) LOWERY ET AL.	
	Examiner Michael Y Won	**Art Unit** 2155	

SEARCHED

Class	Subclass	Date	Examiner
709	213 217	7/21/04	mw

INTERFERENCE SEARCHED

Class	Subclass	Date	Examiner

SEARCH NOTES (INCLUDING SEARCH STRATEGY)

	DATE	EXMR
EAST USPAT USPG-PUB DERWENT EPO, JPO	7/21/04	mw
NPL - SEARCH IEEE	7/21/04	mw

11-08-04

2155 JFW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Keith A. Lowery, et al.
Serial No.:	09/759,406
Filing Date:	January 12, 2001
Group Art Unit:	2155
Examiner:	Michael Won
Title:	METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING

MAIL STOP AMENDMENT

Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

CERTIFICATE OF MAILING
BY EXPRESS MAIL

I hereby certify that this communication is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" under 37 C.F.R. § 1.10 on the date indicated below and is addressed to Mail Stop Amendment, Commissioner For Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Willie Jiles
Willie Jiles

Date: November 5, 2004.

Exp. Mail Receipt No. EV473956929US

RESPONSE PURSUANT TO 37 C.F.R. § 1.111

In response to the Office Action mailed August 5, 2004 (the "Office Action"), Applicants respectfully request the Examiner to reconsider the rejection of the claims in view of the following Amendments thereto and the comments as set forth below. Please amend the Application as follows.

AMENDMENTS TO THE CLAIMS

Please amend the claims as follows:

1. (Currently Amended) A method for dynamic distributed data caching comprising:

providing a cache community on a first side of a point of presence, the cache community comprising at least one peer, each peer having an associated first content portion indicating content obtained from a second side of the point of presence to be cached by the respective peer;

allowing a client to join the cache community;

updating a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community; and

associating a respective second content portion with each peer based on the addition of the client, the second content portion being distinct from the first content portion.

2. (Original) The method for dynamic distributed data caching according to Claim 1 and further comprising:

receiving a join request from the client; and

determining whether to allow the client to join the cache community.

3. (Original) The method dynamic distributed data caching according to Claim 2, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

4. (Original) The method for dynamic distributed data caching according to Claim 1, wherein allowing the client to join the cache community comprises:

generating an allow message;

associating the peer list with the allow message; and

communicating the allow message to the client.

5. (Original) The method for dynamic distributed data caching according to Claim 4, wherein allowing the client to joint the cache community comprises:

generating an allow message comprising the peer list updated to include the client;

communicating the allow message to the client; and

communicating the allow message to at least one member associated with the cache community.

6. (Original) The method for dynamic distributed data caching according to Claim 4, wherein the allow message comprises a CRMSG_UPDATEPEERLIST data message.

7. (Original) The method for dynamic distributed caching according to Claim 4, wherein the peer list associated with the allow message comprises the updated peer list which includes the client.

8. (Currently Amended) The method for dynamic distributed data caching according to Claim 1, wherein the point of presence is an ISP. ~~the peer comprises a computer.~~

9. (Original) The method for dynamic distributed data caching according to Claim 1, wherein a one of the peers comprises a member.

10. (Original) The method for dynamic distributed data caching according to Claim 1, wherein a one of the peers comprises a master.

11. (Original) The method for dynamic distributed data caching according to Claim 1, wherein associating a respective second content portion comprises:

allocating respective second content portions to peers in the peer list; and

updating an allocation table to indicate the second content portion associated with the peers.

12. (Original) The method for dynamic distributed data caching according to Claim 11, wherein the second content portions are distinct.

13. (Original) The method for dynamic distributed data caching according to Claim 11, wherein at least two of the second content portions overlap.

14. (Original) The method for dynamic distributed data caching according to Claim 11, wherein the first and second content portions respectively comprise a plurality of internet protocol domain names.

15. (Original) The method for dynamic distributed data caching according to Claim 1 and further comprising removing the association between the first content portions and the peers.

16. (Currently Amended) A system for dynamic distributed data caching comprising:

logic encoded on storage and operable to:

provide a cache community on a first side of a point of presence, the cache community comprising at least one peer, each peer having an associated first content portion indicating content obtained from a second side of the point of presence to be cached by the respective peer;

allow a client to join the cache community;

update a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community; and

associate a respective second content portion with each peer based on the addition of the client, the second content portion being distinct from the first content portion.

17. (Original) The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to:

receive a join request from the client; and

determine whether to allow the client to join the cache community.

18. (Original) The system for dynamic distributed data caching according to Claim 17, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

19. (Original) The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to:

generate an allow message;

associate the peer list with the allow message;

communicate the allow message to the client.

20. (Original) The system for dynamic distributed data caching according to Claim 19, wherein the logic is further operable to:

generate an allow message comprising the peer list updated to include the client;

communicate the allow message to the client; and

communicate the allow message to at least one member associated with the cache community.

21. (Original) The system for dynamic distributed data caching according to Claim 19, wherein the allow message comprises a CRMSG_UPDATEPEERLIST data message.

22. (Original) The system for dynamic distributed caching according to Claim 19, wherein the peer list associated with the allow message comprises the updated peer list which includes the client.

23. (Currently Amended) The system for dynamic distributed data caching according to Claim 16, wherein the point of presence is an ISP. ~~the peer comprises a computer.~~

24. (Original) The system for dynamic distributed data caching according to Claim 16, wherein a one of the peers comprises a member.

25. (Original) The system for dynamic distributed data caching according to Claim 16, wherein a one of the peers comprises a master.

26. (Original) The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to:

allocate respective second content portions to peers in the peer list; and

update an allocation table to indicate the second content portion associated with the peers.

27. (Original) The system for dynamic distributed data caching according to Claim 26, wherein the second content portions are distinct.

28. (Original) The system for dynamic distributed data caching according to Claim 26, wherein at least two of the second content portions overlap.

29. (Original) The system for dynamic distributed data caching according to Claim 26, wherein the first and second content portions respectively comprise a plurality of internet protocol domain names.

30. (Original) The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to remove the association between the first content portions and the peers.

31. (Canceled)

32. (Canceled)

33. (Canceled)

34. (Canceled)

35. (Canceled)

36. (Canceled)

37. (Canceled)

38. (Canceled)

39. (Canceled)

40. (Canceled)

41. (Canceled)

42. (Canceled)

43. (Canceled)

44. (Canceled)

45. (Canceled)

46. (Canceled)

47. (Canceled)

48. (Canceled)

49. (Canceled)

50. (Canceled)

51. (Canceled)

52. (Canceled)

53. (Canceled)

54. (Canceled)

55. (Canceled)

56. (Canceled)

57. (Canceled)

58. (Canceled)

59. (Canceled)

60. (Canceled)

61. (Canceled)

62. (Canceled)

63. (Canceled)

64. (Canceled)

65. (Canceled)

66. (Canceled)

67. (Canceled)

68. (Canceled)

69. (Canceled)

70. (Canceled)

71. (Canceled)

72. (Canceled)

73. (Canceled)

74. (Canceled)

75. (Canceled)

76. (Canceled)

77. (Canceled)

78. (Canceled)

79. (Canceled)

80. (Canceled)

81. (Canceled)

82. (Canceled)

83. (Canceled)

84. (Canceled)

85. (Canceled)

86. (Canceled)

87. (Canceled)

88. (Canceled)

89. (Canceled)

90. (Canceled)

91. (Canceled)

92. (Canceled)

93. (Canceled)

94. (Canceled)

95. (Original) A method for dynamic distributed data caching comprising:

communicating a community request from a dynamic cache module to an administration module;

receiving a community list from the administration module in response to the community request, the community list including a list of communities;

generating a join request to attempt to join a one of the communities in the community list;

receiving an allow message associated with the one of the communities;

receiving a peer list associated with the one of the communities;

receiving a content request; and

storing content associated with the content request.

96. (Original) The method for dynamic distributed data caching according to Claim 95, wherein the community request comprises a CRMSG_WAKEUP data message.

97. (Original) The method for dynamic distributed data caching according to Claim 95, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

98. (Original) A system for dynamic distributed data caching comprising:

logic encoded on storage and operable to:

communicate a community request from a dynamic cache module to an administration module;

receive a community list from the administration module in response to the community request, the community list including a list of communities;

generate a join request to attempt to join a one of the communities in the community list;

receive an allow message associated with the one of the communities;

receive a peer list associated with the one of the communities;

receive a content request; and

store content associated with the content request.

99. (Original) The system for dynamic distributed data caching according to Claim 98, wherein the community request comprises a CRMSG_WAKEUP data message.

100. (Original) The system for dynamic distributed data caching according to Claim 98, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

101. (Currently Amended) A system for dynamic distributed data caching comprising:

means for providing a cache community on a first side of a point of presence, the cache community comprising at least one peer, each peer having an associated first content portion indicating content obtained from a second side of the point of presence to be cached by the respective peer;

means for allowing a client to join the cache community;

means for updating a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community; and

means for associating a respective second content portion with each peer based on the addition of the client, the second content portion being distinct from the first content portion.

102. (Canceled)

103. (Canceled)

104. (Canceled)

105. (Original) A system for dynamic distributed data caching comprising:

means for communicating a community request from a dynamic cache module to an administration module;

means for receiving a community list from the administration module in response to the community request, the community list including a list of communities;

means for generating a join request to attempt to join a one of the communities in the community list;

means for receiving an allow message associated with the one of the communities;

means for receiving a peer list associated with the one of the communities;

means for receiving a content request; and

means for storing content associated with the content request.

REMARKS

This Application has been carefully reviewed in light of the Office Action mailed August 5, 2004. In order to advance prosecution of this case, Applicants amend Claims 1, 8, 16, 23 and 101, and cancel Claims 31-94 and 102-104 without prejudice or disclaimer. The amendments are not considered to be narrowing or necessary for patentability. Applicants respectfully request reconsideration and favorable action in this case.

Confirmation of Election and Interview Summary

Pursuant to a telephone conversation between the Examiner and Ms. Keiko Ichiye on July 28, 2004, Applicants hereby confirm the election without traverse to prosecute Group I, Claims 1-30, 95-101 and 105, which are drawn to "multi-computer data transferring via shared memory, classified in class 709, subclass 213." Claims 31-94 and 102-104 are hereby withdrawn from the instant application without prejudice or disclaimer.

Section 102(e) Rejections

Claims 1-30, 95-101 and 105 were rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,330,605 B1 to Christensen et al. (hereinafter "*Christensen*"). Applicants respectfully traverse these rejections for the reasons discussed below.

With respect to independent Claim 1, as amended, Applicants submit that *Christensen* does not disclose, teach or suggest at least "providing a cache community on a first side of a point of presence, the cache community comprising at least one peer, each peer having an associated first content portion indicating content obtained from a second side of the point of presence to be cached by the respective peer."

Christensen is directed toward a proxy cache cluster ("PCC") "applicable to a reverse proxy environment." (*Christensen*, col. 1, lines 49-50). *Christensen's* PCC "front-ends the [web] servers of a service provider to increase the availability of services offered by the provider." (*Id.*, col. 5, lines 25-29). In essence, the web servers have "helpers" to help manage and cache various data that a user tries to access via the Internet. (*See, e.g.*, FIGS. 1 and 2 of *Christensen*). Conversely, Applicants' cache community is applicable to what *Christensen* refers to as a "forward proxy" environment. This cache community is separate from the web servers that host the information and, in essence, exist on the "other side" of the Internet remote from the web servers. A user of a personal computer (*e.g.*, a peer) does not have to access the web server via the Internet to become part of the cache community. Thus, the cache community exists on one side of a point of presence, while the web server exists on the other side of the point of presence.

For at least this reason, *Christensen* does not anticipate amended Claim 1. Reconsideration and favorable action are respectfully requested.

With respect to independent Claims 16 and 101, Applicants submit that independent Claims 16 and 101 are allowable for reasons analogous to those above in conjunction with amended Claim 1. Reconsideration and favorable action are respectfully requested.

Dependent Claims 2-15 depend from independent Claim 1 and dependent Claims 17-30 depend from independent Claim 16, and are also not anticipated by *Christensen* because they include the limitations of their respective base claim, which are shown above to be allowable, as well as additional limitations that further distinguish *Christensen*. Therefore, Applicants respectfully request that the rejection of Claims 2-15 and 17-30 be withdrawn.

With respect to independent Claim 95, Applicants submit that *Christensen* does not disclose, teach or suggest at least "receiving a community list from the administration module in response to the community request, the community list including a list of communities."

Christensen's PCC "comprises a group of processor/memory mechanisms (PMMs) that cooperately interact as a system . . . of proxy cache servers." (*Christensen*, col. 3, lines 3-6). "The operational status of a PMM includes joining, up, down and leaving states." (*Id.*, col. 8, lines 29-40). A PMM in *Christensen* simply does not receive a community list (*i.e.*, a PCC list) including a list of communities (*i.e.*, PCCs) from which to choose. There is only the ability in *Christensen* to join one PCC. For at least this reason, *Christensen* does not anticipate independent Claim 95. Reconsideration and favorable action are respectfully requested.

With respect to independent Claims 98 and 105, Applicants submit that independent Claims 98 and 105 are allowable for reasons analogous to those above in conjunction with amended Claim 95. Reconsideration and favorable action are respectfully requested.

Dependent Claims 96-97 depend from independent Claim 95 and dependent Claims 99-100 depend from independent Claim 98, and are also not anticipated by *Christensen* because they include the limitations of their respective base claim, which are shown above to be allowable, as well as additional limitations that further distinguish *Christensen*. Therefore, Applicants respectfully request that the rejection of Claims 96-97 and 99-100 be withdrawn.

CONCLUSION

Applicants have made an earnest attempt to place this case in condition for allowance. For the foregoing reasons, and for other reasons clearly apparent, Applicants respectfully request full allowance of all pending Claims. If the Examiner feels that a telephone conference or an interview would advance prosecution of this Application in any manner, the undersigned attorney for Applicants stands ready to conduct such a conference at the convenience of the Examiner.

No fees are believed to be due, however, the Commissioner is hereby authorized to charge any additional fees or credit any overpayments to Deposit Account No. 02-0384 of Baker Botts L.L.P.

Respectfully submitted,
BAKER BOTTS L.L.P.
Attorneys for Applicants

Keiko Ichiye
Reg. No. 45,460

Correspondence Address:
Baker Botts L.L.P.
2001 Ross Avenue, Suite 600
Dallas, Texas 75201

Date: November 5, 2004

BEST AVAILABLE COPY

PATENT APPLICATION FEE DETERMINATION RECORD

Effective October 1, 2000

Application or Docket Number

09/759,406

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	105	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	105 minus 20=	* 85
INDEPENDENT CLAIMS	15 minus 3 =	* 12
MULTIPLE DEPENDENT CLAIM PRESENT		□

* If the difference in column 1 is less than zero, enter "0" in column 2

SMALL ENTITY TYPE □			OTHER THAN SMALL ENTITY	
RATE	FEE	OR	RATE	FEE
BASIC FEE	355.00	OR	BASIC FEE	710.00
X$ 9=	765.00	OR	X$18=	
X40=	480.00	OR	X80=	
+135=		OR	+270=	
TOTAL	1600.00	OR	TOTAL	

CLAIMS AS AMENDED - PART II

11-5-04

AMENDMENT A	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 38	Minus	** 105	=
Independent	* 6	Minus	*** 15	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				□

SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT B	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				□

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				□

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
***If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

*U.S. GPO: 2000-460-706/30103

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/759,406	01/12/2001	Keith A. Lowery	066241.0117	2308

7590 03/01/2005

Baker Botts L.L.P.
2001 Rose Avenue
Dallas, TX 75201-2980

EXAMINER	
WON, MICHAEL YOUNG	
ART UNIT	**PAPER NUMBER**
2155	

DATE MAILED: 03/01/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/759,406	**Applicant(s)** LOWERY ET AL.	
	Examiner Michael Y Won	**Art Unit** 2155	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *05 November 2004*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-30,95-101 and 105* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-30,95-101 and 105* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

1. Claims 1, 8, 16, 23, and 101 have been amended. Claims 31-94 and 102-104 have been cancelled.

2. Claims 1-30, 95-101, and 105 have been examined and are pending with this action.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3. Claims **95-100** and **105** are rejected under 35 U.S.C. 102(e) as being anticipated by Christensen et al. (US 6,330,605 B1).

As per *claims 95, 98 and 105*, Christensen teaches a method and a system for dynamic distributed data caching comprising logic and means for: communicating a community request from a dynamic cache module to an administration module (see col.6, lines 26-36); receiving a community list (see col.8, lines 46-51) from the administration module in response the community request (see col.6, lines 37-43), the community list including a list of communities (see col.11, lines 58-67); generating a join request to attempt to join a one of the communities in the community list (see col.8, lines 39-43); receiving an allow message associated with the one of the communities (see col.6, lines 23-28); receiving a peer list associated with the one the communities (see col.8, line 46-51); receiving a content request (see col.1, lines 37-51 and col.2, line 65 to col.3, line 13); and storing content associated with the content request (see col.1, lines 37-51).

As per *claims 96 and 99*, Christensen further teaches wherein the community request comprises a CRMSG_WAKEUP data message (see col.6, lines 28-36).

As per *claims 97 and 100*, Christensen further teaches wherein the join request comprises a CRMSG_REQUESTTOJOIN data message (see col.8, lines 39-43).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims **1-30** and **101** are rejected under 35 U.S.C. 103(a) as being unpatentable over Christensen et al. (US 6,330,605 B1) in view of McCanne (US 6,785,704 B1).

As per claims 1, 16, and 101, Christensen teaches of a method and a system for dynamic distributed data caching comprising logic and means for: providing a cache community (see col.2, lines 60-63) comprising at least one peer (see col.3, lines 3-6), each peer having associated first content portion indicating content obtained from a second side of the point of presence (see col.1, lines 43-49) to be cached (see col.1, lines 37-51) by the respective peer (see col.3, lines 17-18); allowing a client (see Fig.4 and col.6, lines 57-62) to join the cache community (see col.8, lines 39-40); updating peer list (see col.8, lines 46-51) associated with the cache community to include the client, the peer list indicating the peers the cache community (see col.5, lines 58-65 and col.6, lines 40-51); and associating a respective second content portion with each peer based on the addition the client, the second content portion being distinct from the first content portion (see col.7, lines 47-53 and col.8, lines 4-7).

Christensen does not explicitly teach wherein the providing is performed on a first side of a point of presence. McCanne teaches wherein the providing is performed on a first side of a point of presence (see Fig.2 and col.6, lines 33-44). It would have been obvious to a person of ordinary skill in the art at the time the invention was made to employ the teachings of McCanne within the system of Christensen by implementing providing a cache community on a first side of a point of presence within the dynamic distributed data caching method and a system because McCanne teaches by servicing requests locally, improves “response time”, reduces “wide-area bandwidth consumption”, and relaxes “load on the production server” (see col.9, lines 61-64).

As per claims 2 and 17, Christensen teaches of further comprising: receiving a join request from the client (see col.8, lines 39-41); and determining whether allow the client join the cache community (implicit: see col.8, lines 41-43).

As per claims 3, 18, 97, and 100, Christensen further teaches wherein the join request comprises a CRMSG_REQUESTTOJOIN data message (see col.8, lines 39-43).

As per claims 4 and 19, Christensen further teaches wherein allowing the client to join the cache community comprises: generating an allow message (see col.6, lines 23-28); associating the peer with the allow message (see col.6, lines 40-51); and communicating the allow message to the client (implicit: see col.6, lines 23-28).

As per claims 5 and 20, Christensen further teaches wherein allowing the client to join the cache community comprises: generating an allow message comprising the peer list updated to include the clients (see col.5, lines 58-65 and col.6, lines 23-28); communicating the allow message to the client (implicit: see col.6, lines 23-28); and communicating the allow message to at least one member associated with the cache community (see col.6, lines 34-36 and col.7, lines 47-53).

As per claims 6 and 21, Christensen further teaches wherein the allow message comprises a CRMSG_UPDATEPEERLIST data message (see col.5, lines 58-65 and col.6, lines 40-43).

As per claims 7 and 22, Christensen further teaches wherein peer list associated with the allow message comprises updated peer which includes the client (see col.6, lines 23-36 and col.7, lines 47-53).

As per claims 8 and 23, Christensen does not explicitly teach wherein the point of presence is an ISP. McCanne teaches wherein the point of presence is an ISP (see Fig.2).

As per claims 9 and 24, Christensen further teaches wherein a one of the peers comprises a member (see col.7, lines 47-49).

As per claims 10 and 25, Christensen further teaches wherein one of the peers comprises a master (see col.6, lines 37-40).

As per claims 11 and 26, Christensen further teaches wherein associating a respective second content portion comprises: allocating respective second content portions peers in the peer list (see col.5, lines 58-65 and col.7, lines 47-49); and updating an allocation table indicate the second content portion associated with the peers (see col.7, lines 47-49).

As per claims 12 and 27, Christensen further teaches wherein the second content portions are distinct (see col.21, lines 26-32).

As per claims 13 and 28, Christensen further teaches wherein at least two of the second content portions overlap (see col.21, lines 32-40).

As per claims 14 and 29, Christensen further teaches wherein the first and second content portions respectively comprise a plurality Internet Protocol domain names (see col.1, line 64 to col.2, line 5).

As per claims 15 and 30, Christensen teaches of further comprising removing the association between the first content portions and the peers (see col.9, lines 58-63).

Response to Arguments

5. Applicant's arguments with respect to claims 1, 16, and 101, specifically regarding that the Christensen (US 6,330,605) reference teaches an embodiment that is different that the presently claimed invention, have been considered but are moot in view of the new ground(s) of rejection. The examiner concurs that Christensen explicitly teaches of a cache community that exists in the "reverse proxy" environment, however, the newly discovered McCanne (US 6,785,704) teaches of a cache community embodied in the "forward proxy" and an obvious motivation for such. Therefore, claims 1-30 and claim 101 have been finally rejected.

6. With respect to the arguments regarding claims 95, 98, and 105, additional reference location has been provided to clearly teach the limitation "the community list including a list of communities". Clearly Christensen teaches of a list addresses and their load rating with various services (communities). Therefore, claims 95-100 and claim 105 have been finally rejected.

Conclusion

7. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP

§ 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael Y Won whose telephone number is 571-272-3993. The examiner can normally be reached on M-Th: 7AM-5PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hosain T Alam can be reached on 571-272-3978. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Michael Won

February 25, 2005

HOSAIN ALAM
SUPERVISORY PATENT EXAMINER

Notice of References Cited

Application/Control No. 09/759,406	Applicant(s)/Patent Under Reexamination LOWERY ET AL.		
Examiner Michael Y Won	Art Unit 2155	Page 1 of 1	

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-6,725,261 B1	04-2004	Novaes et al.	709/220
	B	US-6,785,704 B1	08-2004	McCanne, Steve	718/105
	C	US-			
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Index of Claims	**Application No.** 09/759,406	**Applicant(s)** LOWERY ET AL.
	Examiner Michael Y Won	**Art Unit** 2155

√	**Rejected**		–	**(Through numeral) Cancelled**		N	**Non-Elected**		A	**Appeal**
=	**Allowed**		÷	**Restricted**		I	**Interference**		O	**Objected**

Claim		Date								
Final	Original	2/25/05								
	1	√								
	2	↑								
	3									
	4									
	5									
	6									
	7									
	8									
	9									
	10									
	11									
	12									
	13									
	14									
	15									
	16									
	17									
	18									
	19									
	20									
	21									
	22									
	23									
	24									
	25									
	26									
	27									
	28									
	29	↓								
	30	√								
	~~31~~									
	32	↑								
	33									
	34									
	35									
	36									
	37									
	38									
	39									
	40									
	41									
	42									
	43									
	44									
	45									
	46									
	47									
	48									
	49									
	~~50~~	↓								

Claim		Date								
Final	Original	2/25/05								
	~~51~~	↑								
	52									
	53									
	54									
	55									
	56									
	57									
	58									
	59									
	60									
	61									
	62									
	63									
	64									
	65									
	66									
	67									
	68									
	69									
	70									
	71									
	72									
	73									
	74									
	75									
	76									
	77									
	78									
	79									
	80									
	81									
	82									
	83									
	84									
	85									
	86									
	87									
	88									
	89									
	90									
	91									
	92									
	93	↓								
	~~94~~									
	95	√								
	96	↑								
	97									
	98									
	99	↓								
	100	√								

Claim		Date								
Final	Original	2/25/05								
	101	√								
	~~102~~	↑								
	103									
	~~104~~	↓								
	105	√								
	106									
	107									
	108									
	109									
	110									
	111									
	112									
	113									
	114									
	115									
	116									
	117									
	118									
	119									
	120									
	121									
	122									
	123									
	124									
	125									
	126									
	127									
	128									
	129									
	130									
	131									
	132									
	133									
	134									
	135									
	136									
	137									
	138									
	139									
	140									
	141									
	142									
	143									
	144									
	145									
	146									
	147									
	148									
	149									
	150									

Search Notes	Application No. 09/759,406	Applicant(s) LOWERY ET AL.	
	Examiner Michael Y Won	Art Unit 2155	

SEARCHED

Class	Subclass	Date	Examiner
709	201, 212, 217	2/25/2005	MW

SEARCH NOTES (INCLUDING SEARCH STRATEGY)

	DATE	EXMR
EAST: USPAT; USPG-PUB; DERWENT; EPO; JPO	2/25/2005	MW

INTERFERENCE SEARCHED

Class	Subclass	Date	Examiner

ATTORNEY DOCKET NO.
066241.0117

05/02/05

PATENT APPLICATION
09/759,406

AF
TJW

1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Keith A. Lowery, et al.
Serial No.:	09/759,406
Filing Date:	January 12, 2001
Group Art Unit:	2155
Examiner:	Michael Won
Title:	**METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING**

MAIL STOP AF

Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

EXPRESS MAIL CERTIFICATE
NO.: EV 323318926 US

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR §1.10 on the date indicated above and is addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Name: Willie Jiles
Willie Jiles

Date: April 29, 2005

EV323318926US

RESPONSE PURSUANT TO 37 C.F.R. § 1.116

In response to the Final Office Action mailed March 1, 2005 (the "Office Action"), Applicants respectfully request the Examiner to reconsider the rejection of the claims in view of the following Amendments thereto and the comments as set forth below. Please amend the Application as follows.

AMENDMENTS TO THE CLAIMS

Please amend the claims as follows:

1. (Previously Presented) A method for dynamic distributed data caching comprising:

providing a cache community on a first side of a point of presence, the cache community comprising at least one peer, each peer having an associated first content portion indicating content obtained from a second side of the point of presence to be cached by the respective peer;

allowing a client to join the cache community;

updating a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community; and

associating a respective second content portion with each peer based on the addition of the client, the second content portion being distinct from the first content portion.

2. (Original) The method for dynamic distributed data caching according to Claim 1 and further comprising:

receiving a join request from the client; and

determining whether to allow the client to join the cache community.

3. (Original) The method dynamic distributed data caching according to Claim 2, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

4. (Original) The method for dynamic distributed data caching according to Claim 1, wherein allowing the client to join the cache community comprises:

generating an allow message;

associating the peer list with the allow message; and

communicating the allow message to the client.

5. (Original) The method for dynamic distributed data caching according to Claim 4, wherein allowing the client to joint the cache community comprises:

generating an allow message comprising the peer list updated to include the client;

communicating the allow message to the client; and

communicating the allow message to at least one member associated with the cache community.

6. (Original) The method for dynamic distributed data caching according to Claim 4, wherein the allow message comprises a CRMSG_UPDATEPEERLIST data message.

7. (Original) The method for dynamic distributed caching according to Claim 4, wherein the peer list associated with the allow message comprises the updated peer list which includes the client.

8. (Previously Presented) The method for dynamic distributed data caching according to Claim 1, wherein the point of presence is an ISP.

9. (Original) The method for dynamic distributed data caching according to Claim 1, wherein a one of the peers comprises a member.

10. (Original) The method for dynamic distributed data caching according to Claim 1, wherein a one of the peers comprises a master.

11. (Original) The method for dynamic distributed data caching according to Claim 1, wherein associating a respective second content portion comprises:

allocating respective second content portions to peers in the peer list; and

updating an allocation table to indicate the second content portion associated with the peers.

12. (Original) The method for dynamic distributed data caching according to Claim 11, wherein the second content portions are distinct.

13. (Original) The method for dynamic distributed data caching according to Claim 11, wherein at least two of the second content portions overlap.

14. (Original) The method for dynamic distributed data caching according to Claim 11, wherein the first and second content portions respectively comprise a plurality of internet protocol domain names.

15. (Original) The method for dynamic distributed data caching according to Claim 1 and further comprising removing the association between the first content portions and the peers.

16. (Previously Presented) A system for dynamic distributed data caching comprising:

logic encoded on storage and operable to:

provide a cache community on a first side of a point of presence, the cache community comprising at least one peer, each peer having an associated first content portion indicating content obtained from a second side of the point of presence to be cached by the respective peer;

allow a client to join the cache community;

update a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community; and

associate a respective second content portion with each peer based on the addition of the client, the second content portion being distinct from the first content portion.

17. (Original) The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to:

receive a join request from the client; and

determine whether to allow the client to join the cache community.

18. (Original) The system for dynamic distributed data caching according to Claim 17, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

19. (Original) The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to:

generate an allow message;

associate the peer list with the allow message;

communicate the allow message to the client.

20. (Original) The system for dynamic distributed data caching according to Claim 19, wherein the logic is further operable to:

generate an allow message comprising the peer list updated to include the client;

communicate the allow message to the client; and

communicate the allow message to at least one member associated with the cache community.

21. (Original) The system for dynamic distributed data caching according to Claim 19, wherein the allow message comprises a CRMSG_UPDATEPEERLIST data message.

22. (Original) The system for dynamic distributed caching according to Claim 19, wherein the peer list associated with the allow message comprises the updated peer list which includes the client.

23. (Previously Presented) The system for dynamic distributed data caching according to Claim 16, wherein the point of presence is an ISP.

24. (Original) The system for dynamic distributed data caching according to Claim 16, wherein a one of the peers comprises a member.

25. (Original) The system for dynamic distributed data caching according to Claim 16, wherein a one of the peers comprises a master.

26. (Original) The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to:

allocate respective second content portions to peers in the peer list; and

update an allocation table to indicate the second content portion associated with the peers.

27. (Original) The system for dynamic distributed data caching according to Claim 26, wherein the second content portions are distinct.

28. (Original) The system for dynamic distributed data caching according to Claim 26, wherein at least two of the second content portions overlap.

29. (Original) The system for dynamic distributed data caching according to Claim 26, wherein the first and second content portions respectively comprise a plurality of internet protocol domain names.

30. (Original) The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to remove the association between the first content portions and the peers.

31. (Canceled)

32. (Canceled)

33. (Canceled)

34. (Canceled)

35. (Canceled)

36. (Canceled)

37. (Canceled)

38. (Canceled)

39. (Canceled)

40. (Canceled)

41. (Canceled)

42. (Canceled)

43. (Canceled)

44. (Canceled)

45. (Canceled)

46. (Canceled)

47. (Canceled)

48. (Canceled)

49. (Canceled)

50. (Canceled)

51. (Canceled)

52. (Canceled)

53. (Canceled)

54. (Canceled)

55. (Canceled)

56. (Canceled)

57. (Canceled)

58. (Canceled)

59. (Canceled)

60. (Canceled)

61. (Canceled)

62. (Canceled)

63. (Canceled)

64. (Canceled)

65. (Canceled)

66. (Canceled)

67. (Canceled)

68. (Canceled)

69. (Canceled)

70. (Canceled)

71. (Canceled)

72. (Canceled)

73. (Canceled)

74. (Canceled)

75. (Canceled)

76. (Canceled)

77. (Canceled)

78. (Canceled)

79. (Canceled)

80. (Canceled)

81. (Canceled)

82. (Canceled)

83. (Canceled)

84. (Canceled)

85. (Canceled)

86. (Canceled)

87. (Canceled)

88. (Canceled)

89. (Canceled)

90. (Canceled)

91. (Canceled)

92. (Canceled)

93. (Canceled)

94. (Canceled)

95. (Currently Amended) A method for dynamic distributed data caching comprising:

communicating a community request from a dynamic cache module to an administration module;

receiving a community list from the administration module in response to the community request, the community list including a list of communities;

selecting one of the communities to attempt to join;

generating a join request to attempt to join the selected community; ~~a one of the communities in the community list;~~

receiving an allow message associated with the one of the communities;

receiving a peer list associated with the one of the communities;

receiving a content request; and

storing content associated with the content request.

96. (Original) The method for dynamic distributed data caching according to Claim 95, wherein the community request comprises a CRMSG_WAKEUP data message.

97. (Original) The method for dynamic distributed data caching according to Claim 95, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

98. (Currently Amended) A system for dynamic distributed data caching comprising:

logic encoded on storage and operable to:

communicate a community request from a dynamic cache module to an administration module;

receive a community list from the administration module in response to the community request, the community list including a list of communities;

select one of the communities to attempt to join;

generate a join request to attempt to join the selected community; ~~a one of the communities in the community list;~~

receive an allow message associated with the one of the communities;

receive a peer list associated with the one of the communities;

receive a content request; and

store content associated with the content request.

99. (Original) The system for dynamic distributed data caching according to Claim 98, wherein the community request comprises a CRMSG_WAKEUP data message.

100. (Original) The system for dynamic distributed data caching according to Claim 98, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

101. (Previously Presented) A system for dynamic distributed data caching comprising:

means for providing a cache community on a first side of a point of presence, the cache community comprising at least one peer, each peer having an associated first content portion indicating content obtained from a second side of the point of presence to be cached by the respective peer;

means for allowing a client to join the cache community;

means for updating a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community; and

means for associating a respective second content portion with each peer based on the addition of the client, the second content portion being distinct from the first content portion.

102. (Canceled)

103. (Canceled)

104. (Canceled)

105. (Currently Amended) A system for dynamic distributed data caching comprising:

means for communicating a community request from a dynamic cache module to an administration module;

means for receiving a community list from the administration module in response to the community request, the community list including a list of communities;

means for selecting one of the communities to attempt to join;

means for generating a join request to attempt to join the selected community; ~~a one of the communities in the community list;~~

means for receiving an allow message associated with the one of the communities;

means for receiving a peer list associated with the one of the communities;

means for receiving a content request; and

means for storing content associated with the content request.

REMARKS

This Application has been carefully reviewed in light of the Office Action mailed March 1, 2005. In order to advance prosecution of this case, Applicants amend Claims 95, 98 and 105. The amendments are not considered to be narrowing or necessary for patentability. Applicants respectfully request reconsideration and favorable action in this case.

Section 102(e) Rejections

The Office Action rejects Claims 95-100 and 105 under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,330,605 B1 issued to Christensen et al. (hereinafter "*Christensen*"). Applicants respectfully traverse these rejections for the reasons discussed below.

With respect to independent Claim 95, as amended, Applicants submit that *Christensen* does not disclose, teach or suggest at least "communicating a community request from a dynamic cache module to an administration module, receiving a community list from the administration module in response to the community request, the community list including a list of communities, [and] selecting one of the communities to attempt to join."

Christensen's PCC "comprises a group of processor/memory mechanisms (PMMs) that cooperately interact as a system . . . of proxy cache servers." (*Christensen*, col. 3, lines 3-6). A PMM in *Christensen* does not communicate a community request from a dynamic cache module to an administration module, but is merely "configured with a unique identifier (ID), a network address and PCC configuration software to enable participation in the clustering process." (*Christensen*, col. 6, lines 18-21). "[T]he PMM 'listens' for a mechanism notifying the PMM that it is a member of a PCC 300." (*Id.*, at col. 6, lines 24-25). A PMM also does not receive a community list (*i.e.*, a PCC list) including a list of communities (*i.e.*, PCCs), and selecting one of the communities to attempt to join. Although *Christensen* arguably has more than one PCC, there is certainly no PCC list from which a PMM can select one to attempt to join. A PMM is more of a passive component that is assigned a PCC by a PCC coordinator. (*See, e.g., Christensen*, col. 6, lines 37-40). For at least this reason, *Christensen* does not anticipate amended independent Claim 95. Reconsideration and favorable action are respectfully requested.

With respect to independent Claims 98 and 105, as amended, Applicants submit that independent Claims 98 and 105 are allowable for reasons analogous to those above in conjunction with amended Claim 95. Reconsideration and favorable action are respectfully requested.

Dependent Claims 96-97 depend from independent Claim 95 and dependent Claims 99-100 depend from independent Claim 98, and are also not anticipated by *Christensen* because they include the limitations of their respective base claim, which are shown above to be allowable, as well as additional limitations that further distinguish *Christensen*. Therefore, Applicants respectfully request that the rejection of Claims 96-97 and 99-100 be withdrawn.

Section 103(a) Rejections

The Office Action rejects Claims 1-30 and 101 under 35 U.S.C. 103(a) as being unpatentable over *Christensen* in view of U.S. Patent No. 6,785,704 B1 issued to McCanne (hereinafter "*McCanne*"). Applicants respectfully traverse these rejections for the reasons discussed below.

To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations." MPEP § 2143.

First, Applicants respectfully submit that there is no suggestion or motivation, in either *Christensen* or *McCanne* or in the knowledge generally available to one of ordinary skill in the art, to combine *Christensen* and *McCanne* as proposed by the Examiner. Applicants note that the Examiner has not provided the required evidence of a suggestion to combine *Christensen* and *McCanne.* As mandated by the Federal Circuit, "[t]he factual inquiry whether to [modify] references must be thorough and searching." *In re Sang-Su Lee,* 277 F.3d 1338, 1343 (Fed. Cir. 2002). Any "conclusory statements . . . do not adequately address the issue of motivation to combine." *Id.* The Examiner simply states at page 4 of the Office Action that "[i]t would have been obvious to a person of ordinary skill in the art at the time the invention was made to employ the teachings of McCanne within the system of Christensen by implementing providing a cache community on a first side of a point of presence within the dynamic distributed data caching method and a system because McCanne teaches by servicing requests locally, improves 'response time', reduces 'wide-area bandwidth consumption', and relaxes 'load on the production server' (see col.9, lines 61-

64)." The statement merely presents perceived advantages of the combination. In fact, most proxy caching systems have these general advantages. Thus, there is no "factual inquiry" in the Examiner's statement. The Examiner has not compared and analyzed the two systems of *Christensen* and *McCanne* to come up with technical reasoning as to why one skilled in the relevant art would be motivated to combine these two references. Therefore, this reasoning surely cannot be said to be "thorough and searching."

Furthermore, Applicants submit that one having ordinary skill in the art would not be motivated to combine *Christensen* and *McCanne* because *Christensen* is directed towards a reverse proxy environment, while *McCanne* is directed towards a forward proxy environment. As indicated at col. 1, lines 34-37 of *Christensen*, "[a] proxy cache server ("proxy") may be used to accelerate client access to the Internet ("forward proxy") or to accelerate Internet access to a web server ("reverse proxy")." Forward proxy environments and reverse proxy environments require different hardware and/or software to carry out their functions.

Therefore, Applicants respectfully submit that there is no suggestion or motivation, in either *Christensen* or *McCanne* or in the knowledge generally available to one of ordinary skill in the art, to combine *Christensen* and *McCanne.* For at least this reason, a *prima facie* case of obviousness has not been established, and the *Christensen- McCanne* combination does not render Claim 1 obvious. Reconsideration and favorable action are respectfully requested.

With respect to independent Claims 16 and 101, Applicants submit that independent Claims 16 and 101 are allowable for reasons analogous to those above in conjunction with Claim 1. Reconsideration and favorable action are respectfully requested.

Dependent Claims 2-15 depend from independent Claim 1 and dependent Claims 17-30 depend from independent Claim 16, and are also not rendered obvious by the *Christensen-McCanne* combination because they include the limitations of their respective base claim, which are shown above to be allowable, as well as additional limitations that further distinguish *Christensen* and *McCanne.* Therefore, Applicants respectfully request that the rejection of Claims 2-15 and 17-30 be withdrawn.

CONCLUSION

Applicants have made an earnest attempt to place this case in condition for allowance. For the foregoing reasons, and for other apparent reasons, Applicants respectfully request full allowance of all pending Claims.

If the Examiner feels that a telephone conference or an interview would advance prosecution of this Application in any manner, the undersigned attorney for Applicants stands ready to conduct such a conference at the convenience of the Examiner.

No fees are believed to be due, however, the Commissioner is hereby authorized to charge any additional fees or credit any overpayments to Deposit Account No. 02-0384 of Baker Botts L.L.P.

Respectfully submitted,
BAKER BOTTS L.L.P.
Attorneys for Applicants

Thomas A. Beaton
Reg. No. 46,543

Date: 4/28/05

CORRESPONDENCE ADDRESS:

Customer Number:	**05073**
Attorney Docket Number:	066241.0117

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/759,406	01/12/2001	Keith A. Lowery	066241.0117	2308

7590 05/19/2005

Baker Botts L.L.P.
2001 Rose Avenue
Dallas, TX 75201-2980

EXAMINER	
WON, MICHAEL YOUNG	
ART UNIT	PAPER NUMBER
2155	

DATE MAILED: 05/19/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Advisory Action Before the Filing of an Appeal Brief	**Application No.** *09/759,406*	**Applicant(s)** LOWERY ET AL.
	Examiner Michael Y. Won	**Art Unit** 2155

--The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

THE REPLY FILED 29 April 2005 FAILS TO PLACE THIS APPLICATION IN CONDITION FOR ALLOWANCE.

1. ☒ The reply was filed after a final rejection, but prior to or on the same day as filing a Notice of Appeal. To avoid abandonment of this application, applicant must timely file one of the following replies: (1) an amendment, affidavit, or other evidence, which places the application in condition for allowance; (2) a Notice of Appeal (with appeal fee) in compliance with 37 CFR 41.31; or (3) a Request for Continued Examination (RCE) in compliance with 37 CFR 1.114. The reply must be filed within one of the following time periods:

   a) ☐ The period for reply expires ______months from the mailing date of the final rejection.

   b) ☒ The period for reply expires on: (1) the mailing date of this Advisory Action, or (2) the date set forth in the final rejection, whichever is later. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of the final rejection.

   Examiner Note: If box 1 is checked, check either box (a) or (b). ONLY CHECK BOX (b) WHEN THE FIRST REPLY WAS FILED WITHIN TWO MONTHS OF THE FINAL REJECTION. See MPEP 706.07(f).

Extensions of time may be obtained under 37 CFR 1.136(a). The date on which the petition under 37 CFR 1.136(a) and the appropriate extension fee have been filed is the date for purposes of determining the period of extension and the corresponding amount of the fee. The appropriate extension fee under 37 CFR 1.17(a) is calculated from: (1) the expiration date of the shortened statutory period for reply originally set in the final Office action; or (2) as set forth in (b) above, if checked. Any reply received by the Office later than three months after the mailing date of the final rejection, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

NOTICE OF APPEAL

2. ☐ The Notice of Appeal was filed on _____. A brief in compliance with 37 CFR 41.37 must be filed within two months of the date of filing the Notice of Appeal (37 CFR 41.37(a)), or any extension thereof (37 CFR 41.37(e)), to avoid dismissal of the appeal. Since a Notice of Appeal has been filed, any reply must be filed within the time period set forth in 37 CFR 41.37(a).

AMENDMENTS

3. ☒ The proposed amendment(s) filed after a final rejection, but prior to the date of filing a brief, will not be entered because

   (a)☒ They raise new issues that would require further consideration and/or search (see NOTE below);

   (b)☐ They raise the issue of new matter (see NOTE below);

   (c)☐ They are not deemed to place the application in better form for appeal by materially reducing or simplifying the issues for appeal; and/or

   (d)☐ They present additional claims without canceling a corresponding number of finally rejected claims.

   NOTE: *Amendments to claims 95, 98, and 105 would require further searching and/or consideration.*. (See 37 CFR 1.116 and 41.33(a)).

4. ☐ The amendments are not in compliance with 37 CFR 1.121. See attached Notice of Non-Compliant Amendment (PTOL-324).

5. ☐ Applicant's reply has overcome the following rejection(s): _____.

6. ☐ Newly proposed or amended claim(s) _____ would be allowable if submitted in a separate, timely filed amendment canceling the non-allowable claim(s).

7. ☒ For purposes of appeal, the proposed amendment(s): a) ☒ will not be entered, or b) ☐ will be entered and an explanation of how the new or amended claims would be rejected is provided below or appended.

   The status of the claim(s) is (or will be) as follows:
   Claim(s) allowed: _____.
   Claim(s) objected to: _____.
   Claim(s) rejected: *1-30,95-101 and 105*.
   Claim(s) withdrawn from consideration: _____.

AFFIDAVIT OR OTHER EVIDENCE

8. ☐ The affidavit or other evidence filed after a final action, but before or on the date of filing a Notice of Appeal will not be entered because applicant failed to provide a showing of good and sufficient reasons why the affidavit or other evidence is necessary and was not earlier presented. See 37 CFR 1.116(e).

9. ☐ The affidavit or other evidence filed after the date of filing a Notice of Appeal, but prior to the date of filing a brief, will not be entered because the affidavit or other evidence failed to overcome all rejections under appeal and/or appellant fails to provide a showing a good and sufficient reasons why it is necessary and was not earlier presented. See 37 CFR 41.33(d)(1).

10. ☐ The affidavit or other evidence is entered. An explanation of the status of the claims after entry is below or attached.

REQUEST FOR RECONSIDERATION/OTHER

11. ☐ The request for reconsideration has been considered but does NOT place the application in condition for allowance because: _____.

12. ☐ Note the attached Information Disclosure Statement(s). (PTO/SB/08 or PTO-1449) Paper No(s). _____

13. ☐ Other: _____.

Bharat Barot

BHARAT BAROT
PRIMARY EXAMINER

ATTORNEY DOCKET NO.
066241.0117

05/02/05

PATENT APPLICATION
09/759,406

AF
IFW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Keith A. Lowery, et al.
Serial No.:	09/759,406
Filing Date:	January 12, 2001
Group Art Unit:	2155
Examiner:	Michael Won
Title:	**METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING**

DO NOT ENTER -myw

MAIL STOP AF

Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

EXPRESS MAIL CERTIFICATE
NO.: EV 323318926 US

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR §1.10 on the date indicated above and is addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Name: Willie Jiles
Willie Jiles

Date: April 29, 2005

EV323318926US

RESPONSE PURSUANT TO 37 C.F.R. § 1.116

In response to the Final Office Action mailed March 1, 2005 (the "Office Action"), Applicants respectfully request the Examiner to reconsider the rejection of the claims in view of the following Amendments thereto and the comments as set forth below. Please amend the Application as follows.

RCE

PTO/SB/30 (10-01)
Approved for use through 10/31/2002. OMB 0651-0031
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

REQUEST FOR CONTINUED EXAMINATION (RCE) TRANSMITTAL

Address to:
Mail Stop RCE
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Application Number	09/759,406
Filing Date	January 12, 2001
First Named Inventor	Lowery, et al.
Art Unit	2155
Examiner Name	M. Y. Won
Attorney Docket Number	066241.0117

This is a Request for Continued Examination (RCE) under 37 C.F.R. 1.114 of the above-identified application.
Request for Continued Examination (RCE) practice under 37 CFR 1.114 does not apply to any utility or plant application filed prior to June 8, 1995, or to any design application. See Instruction Sheet for RCEs (not to be submitted to the USPTO) on page 2.

1. Submission required under 37 C.F.R. § 1.114
   a. ☒ Previously submitted
      i. ☒ Consider the amendment(s)/reply under 37 C.F.R. 1.116 previously filed on 04/29/05.
      (Any unentered amendment(s) referred to above will be entered).
      ii. ☐ Consider the arguments in the Appeal Brief or Reply Brief previously filed on ______
      iii. ☐ Other ______
   b. ☐ Enclosed
      i. ☐ Amendment/Reply
      ii. ☐ Affidavit(s)/Declaration(s)
      iii. ☐ Information Disclosure Statement (IDS)
      iv. ☐ Other ______
2. Miscellaneous
   a. ☐ Suspension of action on the above-identified application is requested under 37 C.F.R. 1.103(c) for a period of ______ months. (Period of suspension shall not exceed 3 months; Fee under 37 C.F.R. 1.117(i) required)
   b. ☐ Other ______
3. Fees The RCE fee under 37 C.F.R. 1.17(e) is required by 37 C.F.R 1.114 when the RCE is filed.
   a. ☐ The Director is hereby authorized to charge the following fees, or credit any overpayments, to Deposit Account No.______
      i. ☐ RCE fee required under 37 C.F.R. 1.17(e)
      ii. ☐ Extension of time fee (37 C.F.R. 1.136 and 1.17)
      iii. ☐ Other ______
   b. ☒ Check in the amount of $ 395.00 enclosed
   c. ☐ Payment by credit card (Form PTO-2038 enclosed)
   WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

06/01/2005 SSITHIB1 00000048 09759406
01 FC:2801 395.00 OP

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT REQUIRED

Name (Print / Type)	Thomas A. Beaton	Registration No. (Attorney/Agent)	46543
Signature		Date	5/27/05

CERTIFICATE OF MAILING OR TRANSMISSION

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop RCE, Commissioner for Patents, P.O. Box 1490, Alexandria, Virginia 22313-1450 , or facsimile transmitted to the U.S. Patent and Trademark Office shown below.

Name (Print / Type)	Willie Jiles		
Signature	Willie Jiles	Date	May 27, 2005

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND Fees and Completed Forms to the following address: Assistant Commissioner for Patents, Box RCE, Washington, DC 20231.

DAL01:854380.1

40

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/759,406	01/12/2001	Keith A. Lowery	066241.0117	2308

7590 06/21/2005

Baker Botts L.L.P.
2001 Rose Avenue
Dallas, TX 75201-2980

EXAMINER	
WON, MICHAEL YOUNG	
ART UNIT	PAPER NUMBER
2155	

DATE MAILED: 06/21/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	09/759,406	LOWERY ET AL.
	Examiner	Art Unit
	Michael Y. Won	2155

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 31 May 2005.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-30,95-101 and 105 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-30,95-101 and 105 is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

Response to Amendment

1. This office action is in response to the amendment filed April 29, 2005 and RCE filed May 31, 2005.

2. Applicants' amendment and arguments have been fully considered but they are deemed to be moot in view of the new grounds of rejection.

3. Claims 95, 98, and 105 have been amended.

4. Claims 1-30, 95-101, and 105 have been examined and are pending with this action.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

5. Claims 1, 16, and 101 are rejected under 35 U.S.C. 102(b) as being anticipated by Boyle (US 5,864,854 A).

As per ***claims 1, 16, and 101***, Boyle teaches of a method and a system for dynamic distributed data caching comprising logic and means for: providing a cache community on a first side of a point of presence, the cache community comprising at least one peer, each peer having associated first content portion (see col.3, lines 58-60: "entry") indicating content obtained from a second side of the point of presence to be cached by the respective peer (see abstract; Fig.5, steps 102→106→120→122; and col.1, line 64-col.2, line 9); allowing a client to join the cache community (see col.2, lines 5-9); updating peer list associated with the cache community to include the client (see col.3, lines 21-31, 35-38, and 55-57 and col.6, lines 55-58), the peer list indicating the peers in the cache community (see Fig.3 and col.4, lines 38-40); and associating a respective second content portion with each peer based on the addition of the client (see col.4, lines 9-13), the second content portion being distinct from the first content portion (inherent: see col.3, line 60-col.4, line 17).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 2-15 and 17-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Boyle (US 5,864,854 A) in view of Maggenti et al. (US 6,477,150 B1).

As per ***claims 2 and 17***, Boyle does not explicitly teach of further comprising: receiving a join request from the client; and determining whether allow the client join the cache community. Maggenti teaches of receiving a join request from the client (see col.10, lines 24-29; col.18, lines 36-37; and col.26, lines 65-66); and determining whether to allow the client join the cache community (see col.5, lines 34-37; col.12, lines 3-7; and col.31, lines 49-51).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to employ the teachings of Maggenti within the system of Boyle by implementing receiving a join request from the client and determining whether to allow the client join the cache community within the method and a system for dynamic distributed data caching because respectively, Boyle teaches that the "performance of each client typically improves as the group grows to include more clients, since data items can be obtained more quickly from another client in the group than from a server" (see col.2, lines 5-9) and teaches that the number of members of groups is "preferably selected such that the expected peak number of requests for any data item is not significantly delay by queuing (see col.5, lines 58-61). Therefore Boyle welcomes additional members, but not at the expense of losing performance.

As per ***claims 3 and 18***, Maggenti further teaches wherein the join request comprises a CRMSG_REQUESTTOJOIN data message (implicit: see col.10, lines 24-29; col.18, lines 36-37; and col.26, lines 65-66).

As per ***claims 4 and 19***, Boyle does not explicitly teach wherein allowing the client to join the cache community comprises: generating an allow message; associating the peer with the allow message; and communicating the allow message to the client. Maggenti teaches of generating an allow message (see col.12, lines 3-7 and col.31, lines 49-51); associating the peer with the allow message (inherent); and communicating the allow message to the client (see col.12, lines 3-7 and col.31, lines 49-51).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to employ the teachings of Maggenti within the system of Boyle by implementing generating an allow message; associating the peer with the allow message; and communicating the allow message to the client within the method and a system for dynamic distributed data caching because Boyle teaches that the number of members of groups is "preferably selected such that the expected peak number of requests for any data item is not significantly delay by queuing (see col.5, lines 58-61) and by generating, associating, and communicating and acknowledgment is relayed back so that the client device can be notified.

As per ***claims 5 and 20***, Maggenti further teaches wherein allowing the client to join the cache community comprises: generating an allow message comprising the peer list updated to include the clients (see col.17, lines 47-52 and col.20, lines 8-14); communicating the allow message to the client (see claim 4 and 19 rejection above: redundant limitation); and communicating the allow message to at least one member associated with the cache community (see col.12, lines 16-28).

As per ***claims 6 and 21***, Maggenti further teaches wherein the allow message comprises a CRMSG_UPDATEPEERLIST data message (implicit: see col.12, lines 16-20).

As per ***claims 7 and 22***, Maggenti further teaches wherein peer list associated with the allow message comprises updated peer which includes the client (see col.12, lines 16-20).

As per ***claims 8 and 23***, Boyle further teaches wherein the point of presence is an ISP (implicit: see Fig.1 and col.2, lines 48-58).

As per ***claims 9 and 24***, Boyle further teaches wherein a one of the peers comprises a member (see col.3, lines 58-61).

As per ***claims 10 and 25***, Boyle does not explicitly teach wherein one of the peers comprises a master. Maggenti teaches of a master (see col.3, lines 58-65: "communication manager (CM)").

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to employ the teachings of Maggenti within the system of Boyle by implementing a master within the method and a system for dynamic distributed data caching because such an implementation provides centralized control for administering the functionalities.

As per ***claims 11 and 26***, Boyle further teaches wherein associating a respective second content portion comprises: allocating respective second content portions peers in the peer list (see Fig.2 and Fig.3); and updating an allocation table indicate the second content portion associated with the peers (see col.8, lines 59-62).

As per ***claims 12 and 27***, Boyle further teaches wherein the second content portions are distinct (implicit: see col.3, line 60-col.4, line 17: all the field may be distinct).

As per ***claims 13 and 28***, Boyle further teaches wherein at least two of the second content portions overlap (implicit: see col.3, line 60-col.4, line 17: some of the fields may be the same).

As per ***claims 14 and 29***, Boyle further teaches wherein the first and second content portions respectively comprise a plurality Internet Protocol domain names (see col.3, lines 60-48).

As per ***claims 15 and 30***, Boyle teaches of further comprising removing the association between the first content portions and the peers (inherent).

7. Claims 95-100 and 105 are rejected under 35 U.S.C. 103(a) as being unpatentable over Maggenti et al. (US 6,477,150 B1) in view of Boyle (US 5,864,854 A).

As per ***claims 95, 98 and 105***, Maggenti teaches a method and a system comprising logic and means for: communicating a community request from a module to an administration module (see col.3, lines 55-63); receiving a community list from the administration module in response the community request, the community list including a list of communities (see col.5, lines 38-49); selecting one of the communities to attempt to join (see col.7, lines 43-46); generating a join request to attempt to join the selected communities (see col.10, lines 24-29; col.18, lines 36-37; and col.26, lines 65-

66); receiving an allow message associated with the one of the communities (see col.6, lines 64-66; col.12, lines 3-7; col.31, lines 49-51; and col.42, lines 48-50); receiving a peer list associated with the one the communities (see col.17, lines 47-52 and col.20, lines 8-14); receiving a content request (see col.3, lines 55-63 and col.4, lines 9-12); and storing content associated with the content request (see col.11, lines 20-23).

Maggenti does not explicitly teach that the method and system is employed for dynamic distributed data caching and wherein the module is a dynamic cache module. Boyle teaches of dynamic distributed data caching and wherein the module is a dynamic cache module (see col.1, line 64 to col.2, line 2). It would have been obvious to a person of ordinary skill in the art at the time the invention was made to employ the teachings of Boyle within the system of Maggenti by implementing dynamic distributed data caching and a dynamic cache module because Boyle teaches that such implementation minimizes requests for data items outside each group and minimizes the service load on servers having popular data items" (see col.2, lines 14-17).

As per ***claims 96 and 99***, Maggenti further teaches wherein the community request comprises a CRMSG_WAKEUP data message (implicit: see col.13, lines 63-65).

As per ***claims 97 and 100***, Maggenti further teaches wherein the join request comprises a CRMSG_REQUESTTOJOIN data message (implicit: see col.10, lines 24-29; col.18, lines 36-37; and col.26, lines 65-66).

Response to Arguments

8. In response to the argument filed April 29, 2005 with respect to the amended claims 95, 98, and 105, specifically the limitation "communicating a community request from a dynamic cache module to an administration module, receiving a community list from the administration module in response to the community request, the community list including a list of communities, and selecting one of the communities to attempt to join", the combinational teachings of *Maggenti* et al. (US 6,477,150 B1) and *Boyle* (US 5,864,854 A) have been newly discovered to teach this limitation. See new grounds of rejection above.

With respect to the arguments of claims 1, 16, and 101, the combination of *Christensen* et la. (US 6,330,605 B1) and *McCane* (US 6,785,704 B1) are no longer relied upon to teach the limitations. Claims 1, 16, and 101 is anticipated by *Boyle* (US 5,864,854 A) and therefore rejected. See new grounds of rejection above.

For the reasons above, claims 1-30, 95-101, and 105 are rejected.

Conclusion

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael Y. Won whose telephone number is 571-272-3993. The examiner can normally be reached on M-Th: 7AM-5PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Saleh Najjar can be reached on 571-272-4006. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Michael Won

June 14, 2005

SALEH NAJJAR
PRIMARY EXAMINER

[redacted]

Notice of References Cited	Application/Control No. 09/759,406	Applicant(s)/Patent Under Reexamination LOWERY ET AL.
	Examiner Michael Y. Won	Art Unit 2155 — Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-5,701,427 A	12-1997	Lathrop, Alan	709/237
	B	US-6,065,102 A	05-2000	Peters et al.	711/151
	C	US-6,122,629 A	09-2000	Walker et al.	707/8
	D	US-6,477,150 B1	11-2002	Maggenti et al.	370/312
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

[redacted]

Index of Claims

Application/Control No.	Applicant(s)/Patent under Reexamination
09/759,406	LOWERY ET AL.
Examiner	**Art Unit**
Michael Y. Won	2155

√	Rejected		–	(Through numeral) Cancelled		N	Non-Elected		A	Appeal
=	Allowed		÷	Restricted		I	Interference		O	Objected

Claim		Date								
Final	Original	6/15/05								
	1	√								
	2	↑								
	3									
	4									
	5									
	6									
	7									
	8									
	9									
	10									
	11									
	12									
	13									
	14									
	15									
	16									
	17									
	18									
	19									
	20									
	21									
	22									
	23									
	24									
	25									
	26									
	27									
	28									
	29	↓								
	30	√								
	31	↑								
	32									
	33									
	34									
	35									
	36									
	37									
	38									
	39									
	40									
	41									
	42									
	43									
	44									
	45									
	46									
	47									
	48									
	49	↓								
	50									

Claim		Date								
Final	Original	6/15/05								
	51	↑								
	52									
	53									
	54									
	55									
	56									
	57									
	58									
	59									
	60									
	61									
	62									
	63									
	64									
	65									
	66									
	67									
	68									
	69									
	70									
	71									
	72									
	73									
	74									
	75									
	76									
	77									
	78									
	79									
	80									
	81									
	82									
	83									
	84									
	85									
	86									
	87									
	88									
	89									
	90									
	91									
	92									
	93									
	94	↓								
	95	√								
	96	↑								
	97									
	98									
	99	↓								
	100	√								

Claim		Date								
Final	Original	6/15/05								
	101	√								
	102	↑								
	103									
	104	↓								
	105	√								
	106									
	107									
	108									
	109									
	110									
	111									
	112									
	113									
	114									
	115									
	116									
	117									
	118									
	119									
	120									
	121									
	122									
	123									
	124									
	125									
	126									
	127									
	128									
	129									
	130									
	131									
	132									
	133									
	134									
	135									
	136									
	137									
	138									
	139									
	140									
	141									
	142									
	143									
	144									
	145									
	146									
	147									
	148									
	149									
	150									

U.S. Patent and Trademark Office

Part of Paper No. 20050615

Search Notes	Application/Control No. 09/759,406	Applicant(s)/Patent under Reexamination LOWERY ET AL.	
	Examiner Michael Y. Won	Art Unit 2155	

SEARCHED

Class	Subclass	Date	Examiner
709	212	6/15/2005	MW
707	8, 10	6/15/2005	MW
711	151	6/15/2005	MW

INTERFERENCE SEARCHED

Class	Subclass	Date	Examiner

SEARCH NOTES (INCLUDING SEARCH STRATEGY)

	DATE	EXMR
EAST: USPAT; USPG-PUB; DERWENT; EPO; JPO	6/15/2005	MW

9-8-05

[illegible] 2155

ATTORNEY DOCKET NO.:
066241.0117

PATENT APPLICATION
09/759,406

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Keith A. Lowery, et al.
Serial No.:	09/759,406
Filing Date:	January 12, 2001
Group Art Unit:	2155
Examiner:	Michael Won
Title:	**METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING**

MAIL STOP AMENDMENT

Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

EXPRESS MAIL CERTIFICATE
NO.: EV 322272595 US

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR §1.10 on the date indicated above and is addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Name: Willie Jiles
Willie Jiles

Date: September 7, 2005

EV322272595US

RESPONSE PURSUANT TO 37 C.F.R. § 1.111

In response to the Office Action mailed June 21, 2005 (the "Office Action"), Applicants respectfully request that the Examiner reconsider the rejection of the claims in view of the remarks set forth below.

AMENDMENTS TO THE CLAIMS

Please amend the claims as follows:

1. (Previously Presented) A method for dynamic distributed data caching comprising:

providing a cache community on a first side of a point of presence, the cache community comprising at least one peer, each peer having an associated first content portion indicating content obtained from a second side of the point of presence to be cached by the respective peer;

allowing a client to join the cache community;

updating a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community; and

associating a respective second content portion with each peer based on the addition of the client, the second content portion being distinct from the first content portion.

2. (Original) The method for dynamic distributed data caching according to Claim 1 and further comprising:

receiving a join request from the client; and

determining whether to allow the client to join the cache community.

3. (Original) The method dynamic distributed data caching according to Claim 2, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

4. (Original) The method for dynamic distributed data caching according to Claim 1, wherein allowing the client to join the cache community comprises:

generating an allow message;

associating the peer list with the allow message; and

communicating the allow message to the client.

5. (Original) The method for dynamic distributed data caching according to Claim 4, wherein allowing the client to joint the cache community comprises:

generating an allow message comprising the peer list updated to include the client;

communicating the allow message to the client; and

communicating the allow message to at least one member associated with the cache community.

6. (Original) The method for dynamic distributed data caching according to Claim 4, wherein the allow message comprises a CRMSG_UPDATEPEERLIST data message.

7. (Original) The method for dynamic distributed caching according to Claim 4, wherein the peer list associated with the allow message comprises the updated peer list which includes the client.

8. (Previously Presented) The method for dynamic distributed data caching according to Claim 1, wherein the point of presence is an ISP.

9. (Original) The method for dynamic distributed data caching according to Claim 1, wherein a one of the peers comprises a member.

10. (Original) The method for dynamic distributed data caching according to Claim 1, wherein a one of the peers comprises a master.

11. (Original) The method for dynamic distributed data caching according to Claim 1, wherein associating a respective second content portion comprises:

allocating respective second content portions to peers in the peer list; and

updating an allocation table to indicate the second content portion associated with the peers.

12. (Original) The method for dynamic distributed data caching according to Claim 11, wherein the second content portions are distinct.

13. (Original) The method for dynamic distributed data caching according to Claim 11, wherein at least two of the second content portions overlap.

14. (Original) The method for dynamic distributed data caching according to Claim 11, wherein the first and second content portions respectively comprise a plurality of internet protocol domain names.

15. (Original) The method for dynamic distributed data caching according to Claim 1 and further comprising removing the association between the first content portions and the peers.

16. (Previously Presented) A system for dynamic distributed data caching comprising:

logic encoded on storage and operable to:

provide a cache community on a first side of a point of presence, the cache community comprising at least one peer, each peer having an associated first content portion indicating content obtained from a second side of the point of presence to be cached by the respective peer;

allow a client to join the cache community;

update a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community; and

associate a respective second content portion with each peer based on the addition of the client, the second content portion being distinct from the first content portion.

17. (Original) The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to:

receive a join request from the client; and

determine whether to allow the client to join the cache community.

18. (Original) The system for dynamic distributed data caching according to Claim 17, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

19. (Original) The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to:

generate an allow message;

associate the peer list with the allow message;

communicate the allow message to the client.

20. (Original) The system for dynamic distributed data caching according to Claim 19, wherein the logic is further operable to:

generate an allow message comprising the peer list updated to include the client;

communicate the allow message to the client; and

communicate the allow message to at least one member associated with the cache community.

21. (Original) The system for dynamic distributed data caching according to Claim 19, wherein the allow message comprises a CRMSG_UPDATEPEERLIST data message.

22. (Original) The system for dynamic distributed caching according to Claim 19, wherein the peer list associated with the allow message comprises the updated peer list which includes the client.

23. (Previously Presented) The system for dynamic distributed data caching according to Claim 16, wherein the point of presence is an ISP.

24. (Original) The system for dynamic distributed data caching according to Claim 16, wherein a one of the peers comprises a member.

25. (Original) The system for dynamic distributed data caching according to Claim 16, wherein a one of the peers comprises a master.

26. (Original) The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to:

allocate respective second content portions to peers in the peer list; and

update an allocation table to indicate the second content portion associated with the peers.

27. (Original) The system for dynamic distributed data caching according to Claim 26, wherein the second content portions are distinct.

28. (Original) The system for dynamic distributed data caching according to Claim 26, wherein at least two of the second content portions overlap.

29. (Original) The system for dynamic distributed data caching according to Claim 26, wherein the first and second content portions respectively comprise a plurality of internet protocol domain names.

30. (Original) The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to remove the association between the first content portions and the peers.

31. (Canceled)
32. (Canceled)
33. (Canceled)
34. (Canceled)
35. (Canceled)
36. (Canceled)
37. (Canceled)
38. (Canceled)
39. (Canceled)
40. (Canceled)
41. (Canceled)
42. (Canceled)
43. (Canceled)

44. (Canceled)

45. (Canceled)

46. (Canceled)

47. (Canceled)

48. (Canceled)

49. (Canceled)

50. (Canceled)

51. (Canceled)

52. (Canceled)

53. (Canceled)

54. (Canceled)

55. (Canceled)

56. (Canceled)

57. (Canceled)

58. (Canceled)

59. (Canceled)

60. (Canceled)

61. (Canceled)

62. (Canceled)

63. (Canceled)

64. (Canceled)

65. (Canceled)

66. (Canceled)

67. (Canceled)

68. (Canceled)

69. (Canceled)

70. (Canceled)

71. (Canceled)

72. (Canceled)

73. (Canceled)

74. (Canceled)

75. (Canceled)

76. (Canceled)

77. (Canceled)

78. (Canceled)

79. (Canceled)

80. (Canceled)

81. (Canceled)

82. (Canceled)

83. (Canceled)

84. (Canceled)

85. (Canceled)

86. (Canceled)

87. (Canceled)

88. (Canceled)

89. (Canceled)

90. (Canceled)

91. (Canceled)

92. (Canceled)

93. (Canceled)

94. (Canceled)

95. (Previously Presented) A method for dynamic distributed data caching comprising:

communicating a community request from a dynamic cache module to an administration module;

receiving a community list from the administration module in response to the community request, the community list including a list of communities;

selecting one of the communities to attempt to join;

generating a join request to attempt to join the selected community;

receiving an allow message associated with the one of the communities;

receiving a peer list associated with the one of the communities;

receiving a content request; and

storing content associated with the content request.

96. (Original) The method for dynamic distributed data caching according to Claim 95, wherein the community request comprises a CRMSG_WAKEUP data message.

97. (Original) The method for dynamic distributed data caching according to Claim 95, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

98. (Previously Presented) A system for dynamic distributed data caching comprising:

logic encoded on storage and operable to:

communicate a community request from a dynamic cache module to an administration module;

receive a community list from the administration module in response to the community request, the community list including a list of communities;

select one of the communities to attempt to join;

generate a join request to attempt to join the selected community;

receive an allow message associated with the one of the communities;

receive a peer list associated with the one of the communities;

receive a content request; and

store content associated with the content request.

99. (Original) The system for dynamic distributed data caching according to Claim 98, wherein the community request comprises a CRMSG_WAKEUP data message.

100. (Original) The system for dynamic distributed data caching according to Claim 98, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

101. (Previously Presented) A system for dynamic distributed data caching comprising:

means for providing a cache community on a first side of a point of presence, the cache community comprising at least one peer, each peer having an associated first content portion indicating content obtained from a second side of the point of presence to be cached by the respective peer;

means for allowing a client to join the cache community;

means for updating a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community; and

means for associating a respective second content portion with each peer based on the addition of the client, the second content portion being distinct from the first content portion.

102. (Canceled)

103. (Canceled)

104. (Canceled)

105. (Previously Presented) A system for dynamic distributed data caching comprising:

means for communicating a community request from a dynamic cache module to an administration module;

means for receiving a community list from the administration module in response to the community request, the community list including a list of communities;

means for selecting one of the communities to attempt to join;

means for generating a join request to attempt to join the selected community;

means for receiving an allow message associated with the one of the communities;

means for receiving a peer list associated with the one of the communities;

means for receiving a content request; and

means for storing content associated with the content request.

REMARKS

This Application has been carefully reviewed in light of the Office Action mailed June 21, 2005. At the time of the Office Action, Claims 1-30, 95-101, and 105 were pending in the present application. The Examiner rejected Claims 1-30, 95-101, and 105. Applicants respectfully request reconsideration and allowance of all pending claims.

Section 102(b) Rejections

Claims 1, 16 and 101 were rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,864,854 issued to Boyle ("*Boyle*"). Applicants respectfully traverse these rejections for the reasons discussed below.

In order to establish a *prima facie* case of anticipation, all the elements of the claimed invention must be found within a single prior art reference. *Dewey & Almy Chemical Co. v. Mimex*, 124 F.2d 986, 52 USPQ 138 (2d Cir. 1942). Applicants respectfully submit that each and every element of Claims 1, 16, and 101 are not found within the *Boyle* reference.

Claim 1 recites:

> A method for dynamic distributed data caching comprising:
>
> providing a cache community on a first side of a point of presence, the cache community comprising at least one peer, each peer having an associated first content portion indicating content obtained from a second side of the point of presence to be cached by the respective peer;
>
> allowing a client to join the cache community;
>
> updating a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community; and
>
> associating a respective second content portion with each peer based on the addition of the client, the second content portion being distinct from the first content portion.

Applicants submit that *Boyle* fails to teach, suggest, or disclose each and every element of Claim 1. For example, *Boyle* fails to teach, suggest, or disclose "updating a peer list associated with the cache community to include the client." Instead, *Boyle* discloses a group cache look-up table. Rather than "indicating peers in the cache community" as required by Claim 1, Boyle's group cache look-up table "includes a plurality of entries . . . with <u>each entry corresponding to a data item</u> cached within [the] group." Col. 3, ll. 52-55 (emphasis added); *see also* col. 3, ll. 24-26 ("The group cache look-up table includes an entry corresponding to each data item cached by any member of the group."). As such, Boyle's group cache look-up table is not a peer list as required by Claim 1.

Boyle also fails to teach, suggest, or disclose "associating a respective second content portion with each peer based on the addition of the client." Instead, the portion of *Boyle* relied upon by the Examiner as disclosing this element actually discloses a list of clients at which a particular data item is cached. Col. 4, ll. 7-8. "If the data item is cached by more than one client . . . , the list [] includes a client identifier for each client caching the data item." Col. 4, ll. 9-11. This list, however, does not associate a respective content portion with each peer based on the addition of a client. It merely indicates which clients have cached a particular data item. Therefore, for at least these reasons, Applicants submit that the rejection of Claim 1 is improper and respectfully request that the rejection of Claim 1 be withdrawn.

Similar to Claim 1, Claim 16 recites a system for dynamic distributed data caching comprising logic encoded on storage and operable to "update a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community" and "associate a respective second content portion with each peer based on the addition of the client." Likewise, Claim 101 recites "means for updating a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community" and "means for associating a respective second content portion with each peer based on the addition of the client." Therefore, for reasons analogous to those discussed above with regard to Claim 1, Applicants submit that Claims 16 and 101 are also allowable, and respectfully request that the rejections of Claims 16 and 101 be withdrawn.

<u>Section 103(a) Rejections</u>

Claims 2-15, 17-30 were rejected under 35 U.S.C. § 103(a) as being unpatentable over *Boyle* in view of U.S. Patent No. 6,477,150 issued to Maggenti et al. ("*Maggenti*"). Applicants respectfully traverse these rejections for the reasons discussed below.

Claims 2-15 and 17-30 each depend, directly or indirectly, from Claims 1 and 16. Therefore, Applicants submit that Claims 2-15 and 17-30 are also patentable over the cited references, for example, for the same reasons discussed above with regard to Claims 1 and 16, and respectfully request that the rejections of Claims 2-15 and 17-30 be withdrawn.

Claims 95-100 and 105 were rejected under 35 U.S.C. § 103(a) as being unpatentable over *Maggenti* in view of *Boyle*. Applicants respectfully traverse these rejections for the reasons discussed below.

In order to establish a *prima facie* case of obviousness of a claimed invention, all claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 U.S.P.Q. 580 (C.C.P.A. 1974). Furthermore, the teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, not in Applicant's disclosure. *In re Vaeck*, 947 F.2d 488, 20 U.S.P.Q.2d 1438 (Fed. Cir. 1991). Applicants respectfully submit that each and every element of Claims 95-100, and 105 are not found within the references cited by the Examiner and that there is no motivation to combine the references as suggested by the Examiner.

Generally, *Maggenti* is directed to a system and method for providing group communications services, otherwise known as point-to-multipoint communications or push-to-talk communications, in an existing communication system. *Boyle*, on the other hand, is directed to a system and method for maintaining a shared cache look-up table in an information system network. Although *Boyle* states that "group cache look-up tables minimize requests for data items outside each group and minimize the service load on servers having popular data items" (col. 2, ll. 14-17), there would be no motivation to combine the teachings of *Boyle* with the teachings of *Maggenti* as *Maggenti* does not involve caching or content requests. Instead, *Maggenti* involves implementing a push-to-talk communications system where content is neither requested nor cached. As such, there is clearly no motivation to combine the *Maggenti* and *Boyle* references as suggested by the Examiner.

Even if there were a motivation to combine the *Maggenti* and *Boyle* references as suggested by the Examiner, the *Maggenti-Boyle* combination still fails to teach, suggest, or disclose each and every element of Claims 95-100 and 104.

Claim 95 recites:

> A method for dynamic distributed data caching comprising:
> communicating a community request from a dynamic cache module to an administration module;
> receiving a community list from the administration module in response to the community request, the community list including a list of communities;
> selecting one of the communities to attempt to join;

> generating a join request to attempt to join the selected community;
> receiving an allow message associated with the one of the communities;
> receiving a peer list associated with the one of the communities;
> receiving a content request; and
> storing content associated with the content request.

The *Maggenti-Boyle* combination suggested by the Examiner fails to teach, suggest, or disclose each of these elements. For example, the *Maggenti-Boyle* combination fails to teach, suggest, or disclose "receiving a community list from the administration module in response to the community request." Instead, *Maggenti* discloses a system in which communication devices never communicate a community request to an administration module and never receive a community list in response. The communication devices of *Maggenti* are merely "equipped with a means for requesting a transmission privilege from a [communications manager]." Col. 5, ll. 38-46. This communications manager maintains a net database and "manages the real-time and administrative operation of nets" (col. 5, ll. 38-46); however, the communication manager itself never transmits a community list (*e.g.*, the net database) to the communication devices. Because of this, the communications devices never receive a community list from an administration module in response to a community request as required by Claim 95.

The *Maggenti-Boyle* combination also fails to teach, suggest, or disclose "receiving a peer list associated with the one of the communities." In addition to defining the set of nets known to the communication manager, the net database of *Maggenti* "also lists the defined members of each net." Col. 17, ll. 47-50. However, as discussed above, the communications manager of *Maggenti* never transmits this net database to the communication devices that are part of the various nets. Therefore, the communication devices of *Maggenti* also fail to receive a peer list associated with the one of the communities as required by Claim 95.

The *Maggenti-Boyle* combination also fails to teach, suggest, or disclose "receiving a content request." Instead, *Maggenti* discloses that each communication device may request a transmission privilege, and that if the transmission privilege is granted, the communication device may transmit information to other net members. A request for a transmission privilege, however, is not a content request. In fact, nowhere in *Maggenti* is it disclosed that content may requested by a communication device. At best, *Maggenti* discloses that a communication device may request to transmit content. This, however, is not "receiving a content request" as required by Claim 95.

Similarly, the *Maggenti-Boyle* combination suggest by the Examiner also fails to teach, suggest, or disclose "storing content associated with the content request." Instead, the portion of *Maggenti* relied upon by the Examiner as disclosing this element, actually discloses that each communication device "maintains a database for storing information pertaining to group communications" such as "a list of nets in which the [communication device] is able to join." Col. 11, ll. 20-23. This "information pertaining to group communications," however, is not content associated with a content request, as required by Claim 95. Therefore, for at least this reason, as well as the others discussed above, Applicants submit that the rejection of Claim 95 is improper, and respectfully request that the rejection of Claim 95 be withdrawn.

Similar to Claim 95, Claim 98 recites a system for dynamic distributed data caching comprising logic encoded on storage and operable to "receive a community list from the administration module in response to the community request," "receive a peer list associated with the one of the communities," "receive a content request," and "store content associated with the content request." Likewise, Claim 105 recites "means for receiving a community list from the administration module in response to the community request," "means for receiving a peer list associated with the one of the communities," "means for receiving a content request," and "means for storing content associated with the content request." Therefore, for reasons analogous to those discussed above with regard to Claim 95, Applicants submit that Claims 98 and 105 are also allowable, and respectfully request that the rejections of Claims 98 and 105 be withdrawn.

Claims 96, 97, 99, and 100 each depend, directly or indirectly, from Claims 95 and 98. Therefore, Applicants submit that Claims 96, 97, 99, and 100 are also patentable over the cited references, for example, for the same reasons discussed above with regard to Claims 95 and 98, and respectfully request that the rejections of Claims 96, 97, 99, and 100 be withdrawn.

CONCLUSION

Applicants have made an earnest attempt to place this case in condition for allowance. For the foregoing reasons, and for other apparent reasons, Applicants respectfully request full allowance of all pending Claims. If the Examiner feels that a telephone conference would advance prosecution of this Application in any manner, the undersigned attorney for Applicants stands ready to conduct such a conference at the convenience of the Examiner.

Applicants believe no fee is due. However, should there be a fee discrepancy, the Commissioner is hereby authorized to charge any required fees or credit any overpayments to Deposit Account No. 02-0384 of Baker Botts L.L.P.

Respectfully submitted,

BAKER BOTTS L.L.P.
Attorneys for Applicants

Thomas A. Beaton
Reg. No. 46,543
Phone: 214-953-6464

Date: September 7, 2005

CORRESPONDENCE ADDRESS:

Customer Number: **05073**
Attorney Docket Number: 066241.0117

BEST AVAILABLE COPY

PATENT APPLICATION FEE DETERMINATION RECORD
Effective October 1, 2000

Application or Docket Number
09/759,406

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	105	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	105 minus 20=	* 85
INDEPENDENT CLAIMS	15 minus 3 =	* 12
MULTIPLE DEPENDENT CLAIM PRESENT		□

* If the difference in column 1 is less than zero, enter "0" in column 2

SMALL ENTITY TYPE □ OR OTHER THAN SMALL ENTITY

RATE	FEE		RATE	FEE
BASIC FEE	355.00	OR	BASIC FEE	710.00
X$ 9=	765.00	OR	X$18=	
X40=	480.00	OR	X80=	
+135=		OR	+270=	
TOTAL	1600.00	OR	TOTAL	

CLAIMS AS AMENDED - PART II

11-5-04

AMENDMENT A	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 38	Minus	** 105	=
Independent	* 6	Minus	*** 15	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				□

SMALL ENTITY OR OTHER THAN SMALL ENTITY

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

9-7-05

AMENDMENT B	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 38	Minus	** 105	=
Independent	* 6	Minus	*** 15	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				□

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				□

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
***If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE
*U.S. GPO: 2000-460-706/30103

UNITED STATES PATENT AND TRADEMARK OFFICE

A

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/759,406	01/12/2001	Keith A. Lowery	066241.0117	2308

7590 11/25/2005

Baker Botts L.L.P.
2001 Rose Avenue
Dallas, TX 75201-2980

EXAMINER	
WON, MICHAEL YOUNG	
ART UNIT	**PAPER NUMBER**
2155	

DATE MAILED: 11/25/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/759,406	**Applicant(s)** LOWERY ET AL.	
	Examiner Michael Y. Won	**Art Unit** 2155	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _07 September 2005_.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1-30,95-101 and 105_ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1-30,95-101 and 105_ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This office action is in response to the amendment filed September 7, 2005.

2. Claims 1-30, 95-101, and 105 have been examined and are pending with this action.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

3. Claims 1, 16, and 101 are rejected under 35 U.S.C. 102(b) as being anticipated by Boyle (US 5,864,854 A).

As per ***claims 1, 16, and 101***, Boyle teaches of a method and a system for dynamic distributed data caching comprising logic and means for: providing a cache community on a first side of a point of presence, the cache community comprising at least one peer, each peer having associated first content portion (see col.3, lines 58-60: "entry") indicating content obtained from a second side of the point of presence to be cached by the respective peer (see abstract; Fig.5, steps 102→106→120→122; and col.1, line 64-col.2, line 9); allowing a client to join the cache community (see col.2, lines

5-9); updating peer list associated with the cache community to include the client (see col.3, lines 21-31, 35-38, and 55-60 and col.6, lines 55-58), the peer list indicating the peers in the cache community (see Fig.3 and col.4, lines 38-40); and associating a respective second content portion with each peer based on the addition of the client (see col.4, lines 9-17; col.6, lines 55-58; and col.8, lines 22-26 & 59-62), the second content portion being distinct from the first content portion (inherent: see col.3, line 60-col.4, line 17).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 2-15 and 17-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Boyle (US 5,864,854 A) in view of Maggenti et al. (US 6,477,150 B1).

As per ***claims 2 and 17***, Boyle does not explicitly teach of further comprising: receiving a join request from the client; and determining whether allow the client join the cache community. Maggenti teaches of receiving a join request from the client (see col.10, lines 24-29; col.18, lines 36-37; and col.26, lines 65-66); and determining whether to allow the client join the cache community (see col.5, lines 34-37; col.12, lines 3-7; and col.31, lines 49-51).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to employ the teachings of Maggenti within the system of Boyle by implementing receiving a join request from the client and determining whether to allow the client join the cache community within the method and a system for dynamic distributed data caching because respectively, Boyle teaches that the "performance of each client typically improves as the group grows to include more clients, since data items can be obtained more quickly from another client in the group than from a server" (see col.2, lines 5-9) and teaches that the number of members of groups is "preferably selected such that the expected peak number of requests for any data item is not significantly delay by queuing (see col.5, lines 58-61). Therefore Boyle welcomes additional members, but not at the expense of losing performance.

As per ***claims 3 and 18***, Maggenti further teaches wherein the join request comprises a CRMSG_REQUESTTOJOIN data message (implicit: see col.10, lines 24-29; col.18, lines 36-37; and col.26, lines 65-66).

As per ***claims 4 and 19***, Boyle does not explicitly teach wherein allowing the client to join the cache community comprises: generating an allow message; associating the peer with the allow message; and communicating the allow message to the client. Maggenti teaches of generating an allow message (see col.12, lines 3-7 and col.31, lines 49-51); associating the peer with the allow message (inherent); and communicating the allow message to the client (see col.12, lines 3-7 and col.31, lines 49-51).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to employ the teachings of Maggenti within the system of Boyle by implementing generating an allow message; associating the peer with the allow message; and communicating the allow message to the client within the method and a system for dynamic distributed data caching because Boyle teaches that the number of members of groups is "preferably selected such that the expected peak number of requests for any data item is not significantly delay by queuing (see col.5, lines 58-61) and by generating, associating, and communicating and acknowledgment is relayed back so that the client device can be notified.

As per ***claims 5 and 20***, Maggenti further teaches wherein allowing the client to join the cache community comprises: generating an allow message comprising the peer list updated to include the clients (see col.17, lines 47-52 and col.20, lines 8-14); communicating the allow message to the client (see claim 4 and 19 rejection above: redundant limitation); and communicating the allow message to at least one member associated with the cache community (see col.12, lines 16-28).

As per ***claims 6 and 21***, Maggenti further teaches wherein the allow message comprises a CRMSG_UPDATEPEERLIST data message (implicit: see col.12, lines 16-20).

As per ***claims 7 and 22***, Maggenti further teaches wherein peer list associated with the allow message comprises updated peer which includes the client (see col.12, lines 16-20).

As per ***claims 8 and 23***, Boyle further teaches wherein the point of presence is an ISP (implicit: see Fig.1 and col.2, lines 48-58).

As per ***claims 9 and 24***, Boyle further teaches wherein a one of the peers comprises a member (see col.3, lines 58-61).

As per ***claims 10 and 25***, Boyle does not explicitly teach wherein one of the peers comprises a master. Maggenti teaches of a master (see col.3, lines 58-65: "communication manager (CM)").

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to employ the teachings of Maggenti within the system of Boyle by implementing a master within the method and a system for dynamic distributed data caching because such an implementation provides centralized control for administering the functionalities.

As per ***claims 11 and 26***, Boyle further teaches wherein associating a respective second content portion comprises: allocating respective second content portions peers in the peer list (see Fig.2 and Fig.3); and updating an allocation table indicate the second content portion associated with the peers (see col.8, lines 59-62).

As per ***claims 12 and 27***, Boyle further teaches wherein the second content portions are distinct (implicit: see col.3, line 60-col.4, line 17: all the field may be distinct).

As per ***claims 13 and 28***, Boyle further teaches wherein at least two of the second content portions overlap (implicit: see col.3, line 60-col.4, line 17: some of the fields may be the same).

As per ***claims 14 and 29***, Boyle further teaches wherein the first and second content portions respectively comprise a plurality Internet Protocol domain names (see col.3, lines 60-48).

As per ***claims 15 and 30***, Boyle teaches of further comprising removing the association between the first content portions and the peers (inherent).

5. Claims 95-100 and 105 are rejected under 35 U.S.C. 103(a) as being unpatentable over Maggenti et al. (US 6,477,150 B1) in view of Boyle (US 5,864,854 A).

As per ***claims 95, 98 and 105***, Maggenti teaches a method and a system comprising logic and means for: communicating a community request from a module to an administration module (see col.3, lines 55-63); receiving a community list from the administration module in response the community request, the community list including a list of communities (see col.5, lines 38-49 and col.12, lines 16-20); selecting one of the communities to attempt to join (see col.7, lines 43-46); generating a join request to attempt to join the selected communities (see col.10, lines 24-29; col.18, lines 36-37; and col.26, lines 65-66); receiving an allow message associated with the one of the communities (see col.6, lines 64-66; col.12, lines 3-7; col.31, lines 49-51; and col.42, lines 48-50); receiving a peer list associated with the one the communities (see col.17, lines 47-52 and col.20, lines 8-14); receiving a content request (see col.3, lines 55-63; col.4, lines 9-12; and col.12, lines 16-20); and storing content associated with the content request (see col.11, lines 20-23).

Maggenti does not explicitly teach that the method and system is employed for dynamic distributed data caching and wherein the module is a dynamic cache module. Boyle teaches of dynamic distributed data caching and wherein the module is a dynamic cache module (see col.1, line 64 to col.2, line 2).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to employ the teachings of Boyle within the system of Maggenti by implementing dynamic distributed data caching and a dynamic cache module because Boyle teaches that such implementation "minimizes requests for data items outside each group and minimizes the service load on servers having popular data items" (see col.2, lines 14-17).

As per ***claims 96 and 99***, Maggenti further teaches wherein the community request comprises a CRMSG_WAKEUP data message (implicit: see col.13, lines 63-65).

As per ***claims 97 and 100***, Maggenti further teaches wherein the join request comprises a CRMSG_REQUESTTOJOIN data message (implicit: see col.10, lines 24-29; col.18, lines 36-37; and col.26, lines 65-66).

Response to Arguments

6. Applicant's arguments filed September 7, 2005 have been fully considered but they are not persuasive. See reason below.

A. In response to the argument regarding claims 1, 16, and 101, specifically that Boyle fails to teach, suggest, or disclose, "updating a peer list associated with the cache community to include the client", the applicant(s) based the argument on the basis that because Boyle teaches "the group cache look-up table includes an entry corresponding to each data item cached by any member of the group" that the cache look-up table is not a peer list. Boyle further teaches that this entry corresponding to each data item includes "a client identifier field" (see col.3, lines 58-60). In column 6 lines 55-58, Boyle still further teach of updating this "client identifier field... to indicate an additional client caching the data". Therefore, Boyle clearly teaches the limitation of "updating a peer list associated with the cache community to include the client".

B. In response to the argument regarding claims 1, 16, and 101, specifically that Boyle fails to teach, suggest, or disclose, "associating a respective second content portion with each peer based on the addition of the client", Boyle teaches of updating the client identifier field (see col.6, lines 55-58) and updating based on adding of a client (see col.8, lines 59-62). Therefore, since Boyle teaches of associating content portion with each peer (see col.4, lines 9-11 and col.8, lines 22-26), Boyle in combination will all the reference locations explicitly teach "associating a respective second content portion with each peer based on the addition of the client".

C. For the same reasons above, claims 2-15 and 17-30 remain rejected.

D. In response to applicant's argument regarding the combination of Boyle into Maggenti, that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so **found either in the references themselves** or **in the knowledge generally available to one of ordinary skill in the art**. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988) and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992).

In this case, Boyle teaches the use of caches with a group cache look-up table because this "minimizes requests for data items outside each group and minimizes the service load on servers having popular data items" (see Boyle: col.2, lines 14-17) and Maggenti teaches that each CD (communication device) is "capable of generating data packets suitable for transmission over a data network such as the Internet" (see Maggenti: col.3, lines 55-58) and further teaches that each CD has a "group-list" (see col.12, lines 16-17). Therefore, since Maggenti teaches of a cache incorporated by the CD (see col.31, line 65), it would have been obvious to incorporate the teachings of Boyle within the system of Maggenti to incorporate the benefits claimed.

E. In response to the argument regarding claims 95, 98, and 105, specifically that Maggenti-Boyle fails to teach, suggest, or disclose, "receiving a community list from the administration module in response the community request", clearly this limitation is taught by Maggenti. The applicant(s) seems to equate that the assertion of the communication manager "never transmits this net database" to the communication devices "fail to receive a peer list". There is no support for the assertion of the communication manager "never transmits this net database". In column 5 lines 41-46, Maggenti teach that the communication manager "manages the real-time and administrative operation of nets" including "distribution of net membership and registration lists". Maggenti further teaches in column 12 lines 17-20 that the CM provides the CD with updates of its group-list. For the reasons above, this limitation is explicitly taught.

F. In response to the argument regarding claims 95, 98, and 105, specifically that Maggenti-Boyle fails to teach, suggest, or disclose, "receiving a peer list associated with the one the communities", the applicant(s) again argue, "Maggenti never transmits this net database to the communication device". For the reasons above, Maggenti explicitly teaches this element.

G. In response to the argument regarding claims 95, 98, and 105, specifically that Maggenti-Boyle fails to teach, suggest, or disclose, "receiving a content request", the cited reference locations explicitly teach this limitation. Maggenti teaches in column 3 lines 55-63 that the CD is suitable for generating data packets suitable for transmission over the Internet and that the CM processes data packets and distributes between CDs. Clearly, the request and response of data packets is implicit. Maggenti teaches in column 4 lines 9-12, the substance of the data packets. Furthermore, Maggenti teaches of requesting for update content from the CM (see col.12, lines16-20).

H. In response to the argument regarding claims 95, 98, and 105, specifically that Maggenti-Boyle fails to teach, suggest, or disclose, "storing content associated with the content request", clearly Maggenti teaches of storing the requested content of "group-list" (see col.11, lines 20-23).

l. For the same reasons above, claims 96, 97, 99, and 100 remain rejected.

Conclusion

7. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael Y. Won whose telephone number is 571-272-3993. The examiner can normally be reached on M-Th: 7AM-5PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Saleh Najjar can be reached on 571-272-4006. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Michael Won

November 15, 2005

SALEH NAJJAR
SUPERVISORY PATENT EXAMINER

Search Notes	**Application/Control No.** 09/759,406	**Applicant(s)/Patent under Reexamination** LOWERY ET AL.	
	Examiner Michael Y. Won	**Art Unit** 2155	

SEARCHED

Class	Subclass	Date	Examiner
709	212	11/15/2005	MW

INTERFERENCE SEARCHED

Class	Subclass	Date	Examiner

SEARCH NOTES (INCLUDING SEARCH STRATEGY)

	DATE	EXMR
EAST: USPAT; USPG-PUB; DERWENT; EPO; JPO	11/15/2005	MW

Index of Claims	Application/Control No. 09/759,406	Applicant(s)/Patent under Reexamination LOWERY ET AL.	
	Examiner Michael Y. Won	Art Unit 2155	

√	Rejected	–	(Through numeral) Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

Claim		Date							
Final	Original	11/15/05							
	1	√							
	2								
	3								
	4								
	5								
	6								
	7								
	8								
	9								
	10								
	11								
	12								
	13								
	14								
	15								
	16								
	17								
	18								
	19								
	20								
	21								
	22								
	23								
	24								
	25								
	26								
	27								
	28								
	29								
	30	√							
	~~31~~								
	32								
	33								
	34								
	35								
	36								
	37								
	38								
	39								
	40								
	41								
	42								
	43								
	44								
	45								
	46								
	47								
	48								
	49								
	~~50~~								

Claim		Date							
Final	Original	11/15/05							
	~~51~~								
	52								
	53								
	54								
	55								
	56								
	57								
	58								
	59								
	60								
	61								
	62								
	63								
	64								
	65								
	66								
	67								
	68								
	69								
	70								
	71								
	72								
	73								
	74								
	75								
	76								
	77								
	78								
	79								
	80								
	81								
	82								
	83								
	84								
	85								
	86								
	87								
	88								
	89								
	90								
	91								
	92								
	93								
	~~94~~								
	95	√							
	96								
	97								
	98								
	99								
	100	√							

Claim		Date							
Final	Original	11/15/05							
	101	√							
	~~102~~								
	~~103~~								
	~~104~~								
	105	√							
	106								
	107								
	108								
	109								
	110								
	111								
	112								
	113								
	114								
	115								
	116								
	117								
	118								
	119								
	120								
	121								
	122								
	123								
	124								
	125								
	126								
	127								
	128								
	129								
	130								
	131								
	132								
	133								
	134								
	135								
	136								
	137								
	138								
	139								
	140								
	141								
	142								
	143								
	144								
	145								
	146								
	147								
	148								
	149								
	150								

ATTORNEY DOCKET NO.
066241.0117

PATENT APPLICATION
09/759,406

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Keith A. Lowery, et al.
Serial No.: 09/759,406
Filing Date: January 12, 2001
Group Art Unit: 2155
Examiner: Michael Y. Won
Title: METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED CACHING

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

CERTIFICATE OF MAILING BY EXPRESS MAIL

I hereby certify that the attached Response to Examiner's Final Action is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R. §1.10 on this 20th day of January 2006, addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Willie Jiles
Willie Jiles

Express Mail Receipt
No. EV 733635969 US

01-23-06

AF

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Keith A. Lowery, et al.
Serial No.:	09/759,406
Filing Date:	January 12, 2001
Group Art Unit:	2155
Examiner:	Michael Won
Title:	METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING

Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Dear Sir:

RESPONSE TO EXAMINER'S FINAL ACTION

In response to the Final Action mailed November 25, 2005, Applicant respectfully requests the Examiner to reconsider the rejection of the claims and the objections to the Application in view of the following amendments and/or comments as set forth below.

IN THE CLAIMS

1. (Currently Amended) A method for dynamic distributed data caching comprising:

providing a cache community on a first side of a point of presence, the cache community comprising at least one peer, each peer having an associated first content portion indicating content obtained from a second side of the point of presence to be cached by the respective peer;

allowing a client to join the cache community;

updating a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community; ~~and~~

associating a ~~respective~~ second content portion corresponding to the client with each peer based on ~~the addition~~ joinder of the client, the second content portion being distinct from the first content portion;

allocating the first content portion and the second content portion among the peers in the cache community in response to allowing the client to join the community.

2. (Original) The method for dynamic distributed data caching according to Claim 1 and further comprising:

receiving a join request from the client; and

determining whether to allow the client to join the cache community.

3. (Original) The method dynamic distributed data caching according to Claim 2, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

4. (Original) The method for dynamic distributed data caching according to Claim 1, wherein allowing the client to join the cache community comprises:

generating an allow message;

associating the peer list with the allow message; and

communicating the allow message to the client.

5. (Original) The method for dynamic distributed data caching according to Claim 4, wherein allowing the client to joint the cache community comprises:

generating an allow message comprising the peer list updated to include the client;

communicating the allow message to the client; and

communicating the allow message to at least one member associated with the cache community.

6. (Original) The method for dynamic distributed data caching according to Claim 4, wherein the allow message comprises a CRMSG_UPDATEPEERLIST data message.

7. (Original) The method for dynamic distributed caching according to Claim 4, wherein the peer list associated with the allow message comprises the updated peer list which includes the client.

8. (Previously Presented) The method for dynamic distributed data caching according to Claim 1, wherein the point of presence is an ISP.

9. (Original) The method for dynamic distributed data caching according to Claim 1, wherein a one of the peers comprises a member.

10. (Original) The method for dynamic distributed data caching according to Claim 1, wherein a one of the peers comprises a master.

11. (Original) The method for dynamic distributed data caching according to Claim 1, wherein associating a respective second content portion comprises:

allocating respective second content portions to peers in the peer list; and

updating an allocation table to indicate the second content portion associated with the peers.

12. (Original) The method for dynamic distributed data caching according to Claim 11, wherein the second content portions are distinct.

13. (Original) The method for dynamic distributed data caching according to Claim 11, wherein at least two of the second content portions overlap.

14. (Original) The method for dynamic distributed data caching according to Claim 11, wherein the first and second content portions respectively comprise a plurality of internet protocol domain names.

15. (Original) The method for dynamic distributed data caching according to Claim 1 and further comprising removing the association between the first content portions and the peers.

16. (Currently Amended) A system for dynamic distributed data caching comprising:

logic encoded on storage and operable to:

provide a cache community on a first side of a point of presence, the cache community comprising at least one peer, each peer having an associated first content portion indicating content obtained from a second side of the point of presence to be cached by the respective peer;

allow a client to join the cache community;

update a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community; ~~and~~

associate a ~~respective~~ second content portion corresponding to the client with each peer based on ~~the addition~~ joinder of the client, the second content portion being distinct from the first content portion;

allocate the first content portion and the second content portion among the peers in the cache community in response to allowing the client to join the community.

17. (Original) The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to:

receive a join request from the client; and

determine whether to allow the client to join the cache community.

18. (Original) The system for dynamic distributed data caching according to Claim 17, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

19. (Original) The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to:

generate an allow message;

associate the peer list with the allow message;

communicate the allow message to the client.

20. (Original) The system for dynamic distributed data caching according to Claim 19, wherein the logic is further operable to:

generate an allow message comprising the peer list updated to include the client;

communicate the allow message to the client; and

communicate the allow message to at least one member associated with the cache community.

21. (Original) The system for dynamic distributed data caching according to Claim 19, wherein the allow message comprises a CRMSG_UPDATEPEERLIST data message.

22. (Original) The system for dynamic distributed caching according to Claim 19, wherein the peer list associated with the allow message comprises the updated peer list which includes the client.

23. (Previously Presented) The system for dynamic distributed data caching according to Claim 16, wherein the point of presence is an ISP.

24. (Original) The system for dynamic distributed data caching according to Claim 16, wherein a one of the peers comprises a member.

25. (Original) The system for dynamic distributed data caching according to Claim 16, wherein a one of the peers comprises a master.

26. (Original) The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to:

allocate respective second content portions to peers in the peer list; and

update an allocation table to indicate the second content portion associated with the peers.

27. (Original) The system for dynamic distributed data caching according to Claim 26, wherein the second content portions are distinct.

28. (Original) The system for dynamic distributed data caching according to Claim 26, wherein at least two of the second content portions overlap.

29. (Original) The system for dynamic distributed data caching according to Claim 26, wherein the first and second content portions respectively comprise a plurality of internet protocol domain names.

30. (Original) The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to remove the association between the first content portions and the peers.

31. (Canceled)

32. (Canceled)

33. (Canceled)

34. (Canceled)

35. (Canceled)

36. (Canceled)

37. (Canceled)

38. (Canceled)

39. (Canceled)

40. (Canceled)

41. (Canceled)

42. (Canceled)

43. (Canceled)

44. (Canceled)

45. (Canceled)

46. (Canceled)

47. (Canceled)

48. (Canceled)

49. (Canceled)

50. (Canceled)

51. (Canceled)

52. (Canceled)

53. (Canceled)

54. (Canceled)

55. (Canceled)

56. (Canceled)

57. (Canceled)

58. (Canceled)

59. (Canceled)

60. (Canceled)

61. (Canceled)

62. (Canceled)

63. (Canceled)

64. (Canceled)

65. (Canceled)

66. (Canceled)

67. (Canceled)

68. (Canceled)

69. (Canceled)

70. (Canceled)

71. (Canceled)

72. (Canceled)

73. (Canceled)

74. (Canceled)

75. (Canceled)

76. (Canceled)

77. (Canceled)

78. (Canceled)

79. (Canceled)

80. (Canceled)

81. (Canceled)

82. (Canceled)

83. (Canceled)

84. (Canceled)

85. (Canceled)

86. (Canceled)

87. (Canceled)

88. (Canceled)

89. (Canceled)

90. (Canceled)

91. (Canceled)

92. (Canceled)

93. (Canceled)

94. (Canceled)

95. (Currently Amended) A method for dynamic distributed data caching comprising:

communicating a community request ~~from a dynamic cache module~~ to an administration module;

receiving a community list from the administration module in response to the community request, the community list including a list of communities;

selecting one of the communities to attempt to join;

generating a join request to attempt to join the selected ~~community~~ one of the communities;

receiving an allow message associated with the selected one of the communities;

receiving a peer list associated with the selected one of the communities;

receiving ~~a content request~~ content allocated from peers in the peer list in response to joining the selected one of the communities; and

providing ~~storing~~ content ~~associated with the content request~~ for allocation to peers in the peer list in response to joining the selected one of the communities.

96. (Original) The method for dynamic distributed data caching according to Claim 95, wherein the community request comprises a CRMSG_WAKEUP data message.

97. (Original) The method for dynamic distributed data caching according to Claim 95, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

98. (Currently Amended) A system for dynamic distributed data caching comprising:

logic encoded on storage and operable to:

communicate a community request ~~from a dynamic cache module~~ to an administration module;

receive a community list from the administration module in response to the community request, the community list including a list of communities;

select one of the communities to attempt to join;

generate a join request to attempt to join the selected ~~community~~ one of the communities;

receive an allow message associated with the selected one of the communities;

receive a peer list associated with the selected one of the communities;

receive ~~a content request~~ content allocated from peers in the peer list in response to joining the selected one of the communities; and

provide ~~storing~~ content ~~associated with the content request~~ for allocation to peers in the peer list in response to joining the selected one of the communities.

99. (Original) The system for dynamic distributed data caching according to Claim 98, wherein the community request comprises a CRMSG_WAKEUP data message.

100. (Original) The system for dynamic distributed data caching according to Claim 98, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

101. (Currently Amended) A system for dynamic distributed data caching comprising:

means for providing a cache community on a first side of a point of presence, the cache community comprising at least one peer, each peer having an associated first content portion indicating content obtained from a second side of the point of presence to be cached by the respective peer;

means for allowing a client to join the cache community;

means for updating a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community; ~~and~~

means for associating a ~~respective~~ second content portion corresponding to the client with each peer based on ~~the addition~~ joinder of the client, the second content portion being distinct from the first content portion;

means for allocating the first content portion and the second content portion among the peers in the cache community in response to allowing the client to join the community.

102. (Canceled)

103. (Canceled)

104. (Canceled)

105. (Currently Amended) A system for dynamic distributed data caching comprising:

means for communicating a community request ~~from a dynamic cache module~~ to an administration module;

means for receiving a community list from the administration module in response to the community request, the community list including a list of communities;

means for selecting one of the communities to attempt to join;

means for generating a join request to attempt to join the selected ~~community~~ one of the communities;

means for receiving an allow message associated with the selected one of the communities;

means for receiving a peer list associated with the selected one of the communities;

means for receiving ~~a content request~~ content allocated from peers in the peer list in response to joining the selected one of the communities; and

means for providing ~~storing~~ content ~~associated with the content request~~ for allocation to peers in the peer list in response to joining the selected one of the communities.

REMARKS

This Application has been carefully reviewed in light of the Final Action mailed November 25, 2005. Applicant respectfully requests reconsideration and favorable action in this Application.

Claims 1, 16, and 101 stand rejected under 35 U.S.C. §102(b) as being anticipated by Boyle. Independent Claims 1, 16, and 101 recite in general an ability to allocate a first content portion and a second content portion among peers in a cache community in response to allowing a client to join the community. By contrast, the Boyle patent merely discloses the use of a group look-up table that can be distributed among clients in a group. The Boyle patent does not disclose a capability to allocate cache contents upon adding a new client to a community as provided by the claimed invention. Support for the above recitation can be found at page 44, lines 13-19, of Applicant's specification. Therefore, Applicant respectfully submits that Claims 1, 16, and 101 are not anticipated by the Boyle patent.

Claims 2-15 and 17-30 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Boyle in view of Maggenti, et al. Independent Claim 1, from which Claims 2-15 depend, and Independent Claim 16, from which Claims 17-30 depend, have been shown above to be patentably distinct from the Boyle patent. Moreover, the Maggenti, et al. patent does not include any additional disclosure combinable with the Boyle patent that would be material to patentability of these claims. Therefore, Applicant respectfully submits that Claims 2-15 and 17-30 are patentably distinct over the proposed Boyle - Maggenti, et al. combination.

Claims 95-100 and 105 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Maggenti in view of Boyle. Applicant respectfully traverses this rejection.

The Maggenti, et al. patent is directed to a system and method for providing group communications services, otherwise known as point-to-multipoint communications or push-to-talk communications, in an existing communication system. The Boyle patent, on the other hand, is directed to a system and method for maintaining a shared cache look-up table in an information system network. There would be no motivation to combine the teachings of the Boyle patent with the teachings of Maggenti, et al. patent as the Maggenti, et al. patent does not involve caching or content requests. Instead, the Maggenti, et al. patent involves implementing a push-to-talk communications system where content is neither requested nor cached. The only mention of caching in the Maggenti, et al. patent is with respect to mappings between prior PTA announcements and media streams. As such, there is clearly no motivation to combine the Maggenti, et al. patent and the Boyle patent as proposed by the Examiner.

Even if there were a motivation to combine the Maggenti, et al. and Boyle patents as proposed by the Examiner, the proposed Maggenti, et al. - Boyle combination still fails to teach, suggest, or disclose each and every element of the claimed invention. Independent Claims 95, 98, and 105 recite in general an ability to receive content allocated from peers in the peer list in response to joining the selected one of the communities; and provide content for allocation to peers in the peer list in response to joining the selected one of the communities. By contrast, neither Boyle patent nor the Maggenti, et al. patent provide for content allocation upon joinder into a community as required in the claimed invention. Support for the above recitation can be found at page 44, lines 13-19, of Applicants specification. Therefore, Applicant respectfully submits that Claims 95-100 and 105 are patentably distinct from the proposed Maggenti, et al. - Boyle combination.

This Response to Examiner's Final Action is necessary to address the grounds of rejection and interpretation of the cited art in support thereof. This Response to Examiner's Final Action could not have been presented earlier as the Examiner has only now provided the current interpretation of the cited art in support of the claim rejections.

CONCLUSION

Applicant has now made an earnest attempt to place this Application in condition for allowance. For the foregoing reasons, and for other reasons clearly apparent, Applicant respectfully requests full allowance of all pending claims.

If the Examiner feels that a telephone conference would advance prosecution of this Application in any manner, the undersigned attorney for Applicant stands ready to conduct such a conference at the convenience of the Examiner.

The Commissioner is hereby authorized to charge any required fees or credit any overpayments to Deposit Account No. 02-0384 of BAKER BOTTS L.L.P.

Respectfully submitted,
BAKER BOTTS L.L.P.
Attorneys for Applicant

Charles S. Fish
Reg. No. 35,871

January 20, 2006

CORRESPONDENCE ADDRESS:
2001 Ross Avenue, Suite 600
Dallas, TX 75201-2980
(214) 953-6507
Customer Number: 05073

~~BEST AVAILABLE COPY~~

PATENT APPLICATION FEE DETERMINATION RECORD

Effective October 1, 2000

Application or Docket Number

09/759,406

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	105	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	105 minus 20=	* 85
INDEPENDENT CLAIMS	15 minus 3 =	* 12
MULTIPLE DEPENDENT CLAIM PRESENT		□

* If the difference in column 1 is less than zero, enter "0" in column 2

SMALL ENTITY TYPE □ OR OTHER THAN SMALL ENTITY

RATE	FEE		RATE	FEE
BASIC FEE	355.00	OR	BASIC FEE	710.00
X$ 9=	765.00	OR	X$18=	
X40=	480.00	OR	X80=	
+135=		OR	+270=	
TOTAL	1600.00	OR	TOTAL	

4-29-05

CLAIMS AS AMENDED - PART II

11-5-04

AMENDMENT A		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
	Total	* 38	Minus	** 105	= —
	Independent	* 6	Minus	*** 15	= —
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				□

SMALL ENTITY OR OTHER THAN SMALL ENTITY

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

9-7-05

AMENDMENT B		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
	Total	* 38	Minus	** 105	= —
	Independent	* 6	Minus	*** 15	= —
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				□

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

1-20-06

AMENDMENT C		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
	Total	*	Minus	**	=
	Independent	* 6	Minus	***	=
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				□

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/759,406	01/12/2001	Keith A. Lowery	066241.0117	2308

7590 02/08/2006

Baker Botts L.L.P.
2001 Rose Avenue
Dallas, TX 75201-2980

EXAMINER
WON, MICHAEL YOUNG

ART UNIT	PAPER NUMBER
2155	

DATE MAILED: 02/08/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Advisory Action Before the Filing of an Appeal Brief	**Application No.** *09/759,406*	**Applicant(s)** LOWERY ET AL.	
	Examiner Michael Y. Won	**Art Unit** 2155	

--The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

THE REPLY FILED 20 January 2006 FAILS TO PLACE THIS APPLICATION IN CONDITION FOR ALLOWANCE.

1. ☒ The reply was filed after a final rejection, but prior to or on the same day as filing a Notice of Appeal. To avoid abandonment of this application, applicant must timely file one of the following replies: (1) an amendment, affidavit, or other evidence, which places the application in condition for allowance; (2) a Notice of Appeal (with appeal fee) in compliance with 37 CFR 41.31; or (3) a Request for Continued Examination (RCE) in compliance with 37 CFR 1.114. The reply must be filed within one of the following time periods:

a) ☐ The period for reply expires ______months from the mailing date of the final rejection.

b) ☒ The period for reply expires on: (1) the mailing date of this Advisory Action, or (2) the date set forth in the final rejection, whichever is later. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of the final rejection.

Examiner Note: If box 1 is checked, check either box (a) or (b). ONLY CHECK BOX (b) WHEN THE FIRST REPLY WAS FILED WITHIN TWO MONTHS OF THE FINAL REJECTION. See MPEP 706.07(f).

Extensions of time may be obtained under 37 CFR 1.136(a). The date on which the petition under 37 CFR 1.136(a) and the appropriate extension fee have been filed is the date for purposes of determining the period of extension and the corresponding amount of the fee. The appropriate extension fee under 37 CFR 1.17(a) is calculated from: (1) the expiration date of the shortened statutory period for reply originally set in the final Office action; or (2) as set forth in (b) above, if checked. Any reply received by the Office later than three months after the mailing date of the final rejection, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

NOTICE OF APPEAL

2. ☐ The Notice of Appeal was filed on _____. A brief in compliance with 37 CFR 41.37 must be filed within two months of the date of filing the Notice of Appeal (37 CFR 41.37(a)), or any extension thereof (37 CFR 41.37(e)), to avoid dismissal of the appeal. Since a Notice of Appeal has been filed, any reply must be filed within the time period set forth in 37 CFR 41.37(a).

AMENDMENTS

3. ☒ The proposed amendment(s) filed after a final rejection, but prior to the date of filing a brief, will not be entered because

(a)☒ They raise new issues that would require further consideration and/or search (see NOTE below);

(b)☐ They raise the issue of new matter (see NOTE below);

(c)☐ They are not deemed to place the application in better form for appeal by materially reducing or simplifying the issues for appeal; and/or

(d)☐ They present additional claims without canceling a corresponding number of finally rejected claims.

NOTE: *See Continuation Sheet*. (See 37 CFR 1.116 and 41.33(a)).

4. ☐ The amendments are not in compliance with 37 CFR 1.121. See attached Notice of Non-Compliant Amendment (PTOL-324).

5. ☐ Applicant's reply has overcome the following rejection(s): _____.

6. ☐ Newly proposed or amended claim(s) _____ would be allowable if submitted in a separate, timely filed amendment canceling the non-allowable claim(s).

7. ☒ For purposes of appeal, the proposed amendment(s): a) ☒ will not be entered, or b) ☐ will be entered and an explanation of how the new or amended claims would be rejected is provided below or appended.

The status of the claim(s) is (or will be) as follows:
Claim(s) allowed: _____.
Claim(s) objected to: _____.
Claim(s) rejected: *1-30,95-101 and 105*.
Claim(s) withdrawn from consideration: _____.

AFFIDAVIT OR OTHER EVIDENCE

8. ☐ The affidavit or other evidence filed after a final action, but before or on the date of filing a Notice of Appeal will not be entered because applicant failed to provide a showing of good and sufficient reasons why the affidavit or other evidence is necessary and was not earlier presented. See 37 CFR 1.116(e).

9. ☐ The affidavit or other evidence filed after the date of filing a Notice of Appeal, but prior to the date of filing a brief, will not be entered because the affidavit or other evidence failed to overcome all rejections under appeal and/or appellant fails to provide a showing a good and sufficient reasons why it is necessary and was not earlier presented. See 37 CFR 41.33(d)(1).

10. ☐ The affidavit or other evidence is entered. An explanation of the status of the claims after entry is below or attached.

REQUEST FOR RECONSIDERATION/OTHER

11. ☐ The request for reconsideration has been considered but does NOT place the application in condition for allowance because: _____.

12. ☐ Note the attached Information Disclosure Statement(s). (PTO/SB/08 or PTO-1449) Paper No(s). _____

13. ☐ Other: _____.

SALEH NAJJAR
SUPERVISORY PATENT EXAMINER

Continuation of 3. NOTE: The amended claim language would require further consideration and/or searching.

ATTORNEY DOCKET NO.
066241.0117

DO NOT ENTER -myw

PATENT APPLICATION
09/759,406

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Keith A. Lowery, et al.
Serial No.:	09/759,406
Filing Date:	January 12, 2001
Group Art Unit:	2155
Examiner:	Michael Y. Won
Title:	METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED CACHING

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

CERTIFICATE OF MAILING BY EXPRESS MAIL

I hereby certify that the attached Response to Examiner's Final Action is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R. §1.10 on this 20th day of January 2006, addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Willie Jiles
Willie Jiles

Express Mail Receipt
No. EV 733635969 US

ATTORNEY DOCKET NO.
066241.0117

PATENT APPLICATION
09/759,406

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Keith A. Lowery, et al.
Serial No.:	09/759,406
Filing Date:	January 12, 2001
Group Art Unit:	2155
Examiner:	Michael Y. Won
Title:	METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED CACHING

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

CERTIFICATE OF MAILING BY EXPRESS MAIL

I hereby certify that the attached Request for Continued Examination with check is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R. §1.10 on this 27th day of February 2006, addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Willie Jiles

Willie Jiles

Express Mail Receipt
No. EV 732504043 US

3-1-06

RCE
IFW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Keith A. Lowery, et al.
Serial No.:	09/759,406
Filing Date:	January 12, 2001
Group Art Unit:	2155
Examiner:	Michael Won
Title:	METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING

Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Dear Sir:

REQUEST FOR CONTINUED EXAMINATION

In response to the Advisory Action mailed February 8, 2006, Applicant respectfully requests continued examination of this Application to allow the Examiner to reconsider the rejection of the claims and the objections to the Application in view of the following amendments and/or comments as set forth below.

IN THE CLAIMS

1. (Currently Amended) A method for dynamic distributed data caching comprising:

providing a cache community on a first side of a point of presence, the cache community comprising at least one peer, each peer having an associated first content portion indicating content obtained from a second side of the point of presence to be cached by the respective peer;

allowing a client to join the cache community;

updating a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community; ~~and~~

associating a ~~respective~~ second content portion corresponding to the client with each peer based on ~~the addition~~ joinder of the client, the second content portion being distinct from the first content portion;

allocating the first content portion and the second content portion among the peers in the cache community in response to allowing the client to join the community.

2. (Original) The method for dynamic distributed data caching according to Claim 1 and further comprising:

receiving a join request from the client; and

determining whether to allow the client to join the cache community.

3. (Original) The method dynamic distributed data caching according to Claim 2, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

4. (Original) The method for dynamic distributed data caching according to Claim 1, wherein allowing the client to join the cache community comprises:

generating an allow message;

associating the peer list with the allow message; and

communicating the allow message to the client.

5. (Original) The method for dynamic distributed data caching according to Claim 4, wherein allowing the client to joint the cache community comprises:

generating an allow message comprising the peer list updated to include the client;

communicating the allow message to the client; and

communicating the allow message to at least one member associated with the cache community.

6. (Original) The method for dynamic distributed data caching according to Claim 4, wherein the allow message comprises a CRMSG_UPDATEPEERLIST data message.

7. (Original) The method for dynamic distributed caching according to Claim 4, wherein the peer list associated with the allow message comprises the updated peer list which includes the client.

8. (Previously Presented) The method for dynamic distributed data caching according to Claim 1, wherein the point of presence is an ISP.

9. (Original) The method for dynamic distributed data caching according to Claim 1, wherein a one of the peers comprises a member.

10. (Original) The method for dynamic distributed data caching according to Claim 1, wherein a one of the peers comprises a master.

11. (Original) The method for dynamic distributed data caching according to Claim 1, wherein associating a respective second content portion comprises:

allocating respective second content portions to peers in the peer list; and

updating an allocation table to indicate the second content portion associated with the peers.

12. (Original) The method for dynamic distributed data caching according to Claim 11, wherein the second content portions are distinct.

13. (Original) The method for dynamic distributed data caching according to Claim 11, wherein at least two of the second content portions overlap.

14. (Original) The method for dynamic distributed data caching according to Claim 11, wherein the first and second content portions respectively comprise a plurality of internet protocol domain names.

15. (Original) The method for dynamic distributed data caching according to Claim 1 and further comprising removing the association between the first content portions and the peers.

16. (Currently Amended) A system for dynamic distributed data caching comprising:

logic encoded on storage and operable to:

provide a cache community on a first side of a point of presence, the cache community comprising at least one peer, each peer having an associated first content portion indicating content obtained from a second side of the point of presence to be cached by the respective peer;

allow a client to join the cache community;

update a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community; ~~and~~

associate a ~~respective~~ second content portion corresponding to the client with each peer based on ~~the addition~~ joinder of the client, the second content portion being distinct from the first content portion;

allocate the first content portion and the second content portion among the peers in the cache community in response to allowing the client to join the community.

17. (Original) The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to:

receive a join request from the client; and

determine whether to allow the client to join the cache community.

18. (Original) The system for dynamic distributed data caching according to Claim 17, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

19. (Original) The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to:

generate an allow message;

associate the peer list with the allow message;

communicate the allow message to the client.

20. (Original) The system for dynamic distributed data caching according to Claim 19, wherein the logic is further operable to:

generate an allow message comprising the peer list updated to include the client;

communicate the allow message to the client; and

communicate the allow message to at least one member associated with the cache community.

21. (Original) The system for dynamic distributed data caching according to Claim 19, wherein the allow message comprises a CRMSG_UPDATEPEERLIST data message.

22. (Original) The system for dynamic distributed caching according to Claim 19, wherein the peer list associated with the allow message comprises the updated peer list which includes the client.

23. (Previously Presented) The system for dynamic distributed data caching according to Claim 16, wherein the point of presence is an ISP.

24. (Original) The system for dynamic distributed data caching according to Claim 16, wherein a one of the peers comprises a member.

25. (Original) The system for dynamic distributed data caching according to Claim 16, wherein a one of the peers comprises a master.

26. (Original) The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to:

allocate respective second content portions to peers in the peer list; and

update an allocation table to indicate the second content portion associated with the peers.

27. (Original) The system for dynamic distributed data caching according to Claim 26, wherein the second content portions are distinct.

28. (Original) The system for dynamic distributed data caching according to Claim 26, wherein at least two of the second content portions overlap.

29. (Original) The system for dynamic distributed data caching according to Claim 26, wherein the first and second content portions respectively comprise a plurality of internet protocol domain names.

30. (Original) The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to remove the association between the first content portions and the peers.

31. (Canceled)

32. (Canceled)

33. (Canceled)

34. (Canceled)

35. (Canceled)

36. (Canceled)

37. (Canceled)

38. (Canceled)

39. (Canceled)

40. (Canceled)

41. (Canceled)

42. (Canceled)

43. (Canceled)

44. (Canceled)

45. (Canceled)

46. (Canceled)

47. (Canceled)

48. (Canceled)

49. (Canceled)

50. (Canceled)

51. (Canceled)

52. (Canceled)

53. (Canceled)

54. (Canceled)

55. (Canceled)

56. (Canceled)

57. (Canceled)

58. (Canceled)

59. (Canceled)

60. (Canceled)

61. (Canceled)

62. (Canceled)

63. (Canceled)

64. (Canceled)

65. (Canceled)

66. (Canceled)

67. (Canceled)

68. (Canceled)

69. (Canceled)

70. (Canceled)

71. (Canceled)

72. (Canceled)

73. (Canceled)

74. (Canceled)

75. (Canceled)

76. (Canceled)

77. (Canceled)

78. (Canceled)

79. (Canceled)

80. (Canceled)

81. (Canceled)

82. (Canceled)

83. (Canceled)

84. (Canceled)

85. (Canceled)

86. (Canceled)

87. (Canceled)

88. (Canceled)

89. (Canceled)

90. (Canceled)

91. (Canceled)

92. (Canceled)

93. (Canceled)

94. (Canceled)

95. (Currently Amended) A method for dynamic distributed data caching comprising:

communicating a community request ~~from a dynamic cache module~~ to an administration module;

receiving a community list from the administration module in response to the community request, the community list including a list of communities;

selecting one of the communities to attempt to join;

generating a join request to attempt to join the selected ~~community~~ one of the communities;

receiving an allow message associated with the selected one of the communities;

receiving a peer list associated with the selected one of the communities;

receiving ~~a content request~~ content allocated from peers in the peer list in response to joining the selected one of the communities; and

providing ~~storing~~ content ~~associated with the content request~~ for allocation to peers in the peer list in response to joining the selected one of the communities.

96. (Original) The method for dynamic distributed data caching according to Claim 95, wherein the community request comprises a CRMSG_WAKEUP data message.

97. (Original) The method for dynamic distributed data caching according to Claim 95, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

98. (Currently Amended) A system for dynamic distributed data caching comprising:

logic encoded on storage and operable to:

communicate a community request ~~from a dynamic cache module~~ to an administration module;

receive a community list from the administration module in response to the community request, the community list including a list of communities;

select one of the communities to attempt to join;

generate a join request to attempt to join the selected ~~community~~ one of the communities;

receive an allow message associated with the selected one of the communities;

receive a peer list associated with the selected one of the communities;

receive ~~a content request~~ content allocated from peers in the peer list in response to joining the selected one of the communities; and

provide ~~storing~~ content ~~associated with the content request~~ for allocation to peers in the peer list in response to joining the selected one of the communities.

99. (Original) The system for dynamic distributed data caching according to Claim 98, wherein the community request comprises a CRMSG_WAKEUP data message.

100. (Original) The system for dynamic distributed data caching according to Claim 98, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

101. (Currently Amended) A system for dynamic distributed data caching comprising:

means for providing a cache community on a first side of a point of presence, the cache community comprising at least one peer, each peer having an associated first content portion indicating content obtained from a second side of the point of presence to be cached by the respective peer;

means for allowing a client to join the cache community;

means for updating a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community; ~~and~~

means for associating a ~~respective~~ second content portion corresponding to the client with each peer based on ~~the addition~~ joinder of the client, the second content portion being distinct from the first content portion;

means for allocating the first content portion and the second content portion among the peers in the cache community in response to allowing the client to join the community.

102. (Canceled)

103. (Canceled)

104. (Canceled)

105. (Currently Amended) A system for dynamic distributed data caching comprising:

means for communicating a community request ~~from a dynamic cache module~~ to an administration module;

means for receiving a community list from the administration module in response to the community request, the community list including a list of communities;

means for selecting one of the communities to attempt to join;

means for generating a join request to attempt to join the selected ~~community~~ one of the communities;

means for receiving an allow message associated with the selected one of the communities;

means for receiving a peer list associated with the selected one of the communities;

means for receiving ~~a content request~~ content allocated from peers in the peer list in response to joining the selected one of the communities; and

means for providing ~~storing~~ content ~~associated with the content request~~ for allocation to peers in the peer list in response to joining the selected one of the communities.

REMARKS

This Application has been carefully reviewed in light of the Advisory Action mailed February 8, 2006. In order to advance prosecution of this Application, Claims 1, 16, 95, 98, and 101 have been amended. Applicant respectfully requests reconsideration and favorable action in this Application.

The Examiner issued a Final Action on November 25, 2005. Applicant submitted a Response to Examiner's Final Action on January 20, 2006. The Examiner issued an Advisory Action on February 8, 2006 stating that the Response to Examiner's Final Action would not be entered because it raised new issues requiring further searching and consideration. Applicant respectfully requests continued examination of this Application so that the Response to Examiner's Final Action can be entered and considered by the Examiner pursuant to this Request for Continued Examination. For the convenience of the Examiner, the amendments made to the claims in the Response to Examiner's Final Action and the accompanying comments are repeated herein.

Claims 1, 16, and 101 stand rejected under 35 U.S.C. §102(b) as being anticipated by Boyle. Independent Claims 1, 16, and 101 recite in general an ability to allocate a first content portion and a second content portion among peers in a cache community in response to allowing a client to join the community. By contrast, the Boyle patent merely discloses the use of a group look-up table that can be distributed among clients in a group. The Boyle patent does not disclose a capability to allocate cache contents upon adding a new client to a community as provided by the claimed invention. Support for the above recitation can be found at page 44, lines 13-19, of Applicant's specification. Therefore, Applicant respectfully submits that Claims 1, 16, and 101 are not anticipated by the Boyle patent.

Claims 2-15 and 17-30 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Boyle in view of Maggenti, et al. Independent Claim 1, from which Claims 2-15 depend, and Independent Claim 16, from which Claims 17-30 depend, have been shown above to be patentably distinct from the Boyle patent. Moreover, the Maggenti, et al. patent does not include any additional disclosure combinable with the Boyle patent that would be material to patentability of these claims. Therefore, Applicant respectfully submits that Claims 2-15 and 17-30 are patentably distinct over the proposed Boyle - Maggenti, et al. combination.

Claims 95-100 and 105 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Maggenti in view of Boyle. Applicant respectfully traverses this rejection.

The Maggenti, et al. patent is directed to a system and method for providing group communications services, otherwise known as point-to-multipoint communications or push-to-talk communications, in an existing communication system. The Boyle patent, on the other hand, is directed to a system and method for maintaining a shared cache look-up table in an information system network. There would be no motivation to combine the teachings of the Boyle patent with the teachings of Maggenti, et al. patent as the Maggenti, et al. patent does not involve caching or content requests. Instead, the Maggenti, et al. patent involves implementing a push-to-talk communications system where content is neither requested nor cached. The only mention of caching in the Maggenti, et al. patent is with respect to mappings between prior PTA announcements and media streams. As such, there is clearly no motivation to combine the Maggenti, et al. patent and the Boyle patent as proposed by the Examiner.

Even if there were a motivation to combine the Maggenti, et al. and Boyle patents as proposed by the Examiner, the proposed Maggenti, et al. - Boyle combination still fails to teach, suggest, or disclose each and every element of the claimed invention. Independent Claims 95, 98, and 105 recite in general an ability to receive content allocated from peers in the peer list in response to joining the selected one of the communities; and provide content for allocation to peers in the peer list in response to joining the selected one of the communities. By contrast, neither Boyle patent nor the Maggenti, et al. patent provide for content allocation upon joinder into a community as required in the claimed invention. Support for the above recitation can be found at page 44, lines 13-19, of Applicants specification. Therefore, Applicant respectfully submits that Claims 95-100 and 105 are patentably distinct from the proposed Maggenti, et al. - Boyle combination.

Attached herewith is a check in an amount of $790.00 made payable to the "Commissioner of Patents and Trademarks" to satisfy the request for continued examination fee of 37 C.F.R. §1.17(e).

CONCLUSION

Applicant has now made an earnest attempt to place this Application in condition for allowance. For the foregoing reasons, and for other reasons clearly apparent, Applicant respectfully requests full allowance of all pending claims.

If the Examiner feels that a telephone conference would advance prosecution of this Application in any manner, the undersigned attorney for Applicant stands ready to conduct such a conference at the convenience of the Examiner.

The Commissioner is hereby authorized to charge any required fees or credit any overpayments to Deposit Account No. 02-0384 of BAKER BOTTS L.L.P.

Respectfully submitted,
BAKER BOTTS L.L.P.
Attorneys for Applicant

Charles S. Fish
Reg. No. 35,870

February 27, 2006

CORRESPONDENCE ADDRESS:
2001 Ross Avenue, Suite 600
Dallas, TX 75201-2980
(214) 953-6507
Customer Number: 05073

PATENT APPLICATION FEE DETERMINATION RECORD

Effective October 1, 2004

Application or Docket Number

09/759406

CLAIMS AS FILED - PART I

SMALL ENTITY TYPE □ OR OTHER THAN SMALL ENTITY

FOR 02/27/06	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY RATE	FEE		OTHER THAN SMALL ENTITY RATE	FEE
BASIC FEE					OR		790.00
TOTAL CLAIMS	38 minus 105	* 0	x$11=		OR	x$22=	
INDEPENDENT CLAIMS	6 minus 15	* 0	x41=		OR	x82=	
MULTIPLE DEPENDENT CLAIM PRESENT			+135=		OR	+270=	
			TOTAL	0	OR	TOTAL	

* If the difference in column 1 is less than zero, enter "0" in column 2

CLAIMS AS AMENDED - PART II

AMENDMENT A	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE	ADDITIONAL FEE		OTHER THAN SMALL ENTITY RATE	ADDITIONAL FEE
Total	*	Minus	**	=	x$11=		OR	x$22=	
Independent	*	Minus	***	=	x41=		OR	x82=	
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM					+135=		OR	+270=	
					TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT B	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE	ADDITIONAL FEE		OTHER THAN SMALL ENTITY RATE	ADDITIONAL FEE
Total	*	Minus	**	=	x$11=		OR	x$22=	
Independent	*	Minus	***	=	x41=		OR	x82=	
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM					+135=		OR	+270=	
					TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE	ADDITIONAL FEE		OTHER THAN SMALL ENTITY RATE	ADDITIONAL FEE
Total	*	Minus	**	=	x$11=		OR	x$22=	
Independent	*	Minus	***	=	x41=		OR	x82=	
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM					+135=		OR	+270=	
					TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

*U.S. Government Printing Office: 1997 - 430-571/69194

PATENT APPLICATION FEE DETERMINATION RECORD

Effective October 1, 2000

Application or Docket Number

09/759,406

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	105	
FOR 01/12/01	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	105 minus 20=	* 85
INDEPENDENT CLAIMS	15 minus 3 =	* 12
MULTIPLE DEPENDENT CLAIM PRESENT		□

* If the difference in column 1 is less than zero, enter "0" in column 2

SMALL ENTITY TYPE □ OR OTHER THAN SMALL ENTITY

RATE	FEE		RATE	FEE
BASIC FEE	355.00	OR	BASIC FEE	710.00
X$ 9=	765.—	OR	X$18=	
X40=	480.—	OR	X80=	
+135=		OR	+270=	
TOTAL	1600.00	OR	TOTAL	

4-29-05 CLAIMS AS AMENDED - PART II

11-5-04

AMENDMENT A

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 38	Minus	** 105	= —
Independent	* 6	Minus	*** 15	= —
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				□

SMALL ENTITY OR OTHER THAN SMALL ENTITY

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE	0	OR	TOTAL ADDIT. FEE	

9-7-05

AMENDMENT B

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 38	Minus	** 105	= —
Independent	* 6	Minus	*** 15	= —
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				□

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE	0	OR	TOTAL ADDIT. FEE	

1-20-06

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 38	Minus	** 105	= 0
Independent	* 6	Minus	*** 15	= 0
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				□

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE	0	OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

M PTO-875
.8/00)

Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE
*U.S. GPO: [illegible]

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	15	709/203,205,215,216,248.ccls. and cach$5 with (community or cluster) and (sharing and updating) with (content or data or cache)	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	OFF	2006/05/11 14:34
S1	7	709/217,213.ccls. and dynamic$5 with distributed with cach$5 and peer and (community or group) and peer	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2004/07/21 14:41
S2	24	709/217,213.ccls. and dynamic$5 with cach$5 and peer and (community or group)	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2004/07/21 14:26
S3	25	709/217,213.ccls. and dynamic$5 with cach$5 and peer$9 and (community or group)	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2004/07/21 14:44
S4	11	709/217,213.ccls. and dynamic$5 with cach$5 and peer$9 and (community or group) and join$5	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2004/07/21 14:29
S5	3	dynamic$5 with distributed with cach$5 and (community or group) and peer and join$3 and updat$3 and peer with list	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2004/07/21 14:45
S6	0	709/713,714,715,716,717.ccls. and cach$5.ab. and (community or group) and peer$9	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2004/07/21 14:44
S7	275	cach$5.ab. and (community or group) and peer$9	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2004/07/21 14:43
S8	0	709/713,714,715,716,717.ccls. and cach$5 and (community or group) and peer$9	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2004/07/21 14:44
S9	184	dynamic$5 and cach$5.ab. and peer$9 and (community or group)	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2004/07/21 14:45
S10	7	dynamic$5 and distributed and cach$5.ab. and (community or group) and peer and join$3 and updat$3 and peer with list	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2004/07/21 14:46

EAST Search History

S121	7	(request adj2 join or request$4join or join adj2 request) and (community or group) and (wakeup or wake adj up) with message	USPAT; EPO; DERWENT	OR	ON	2005/06/15 12:00
S122	3	community and join with request and allocat$8 with peer with content and list	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	OFF	2006/05/11 13:10
S123	3	group and join with request and allocat$8 with peer with content and list	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	OFF	2006/05/11 13:10
S124	0	cluster and join with request and allocat$8 with peer with content and list	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	OFF	2006/05/11 13:11
S125	21	cluster and join with request and (shar$6 or synchoniz$8 or distribut$8 or allocat$8) with (peer or member) with content and list	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	OFF	2006/05/11 13:12

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/759,406	01/12/2001	Keith A. Lowery	066241.0117	2308

7590 05/16/2006

Baker Botts L.L.P.
2001 Rose Avenue
Dallas, TX 75201-2980

EXAMINER	
WON, MICHAEL YOUNG	

ART UNIT	PAPER NUMBER
2155	

DATE MAILED: 05/16/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

6

Office Action Summary	**Application No.** 09/759,406	**Applicant(s)** LOWERY ET AL.	
	Examiner Michael Y. Won	**Art Unit** 2155	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 27 February 2006.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-30,95-101 and 105* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-30,95-101 and 105* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

1. This action is in response to the Request For Continued Examination and the amendment filed February 27, 2006.

2. Claims 1, 16, 95, 98, 101, and 105 have been amended.

3. Claims 1-30, 95-101, and 105 have been examined and are pending with this action.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

4. Claims 1, 16, and 101 are rejected under 35 U.S.C. 102(b) as being anticipated by Boyle (US 5,864,854 A).

As per ***claim 1***, Boyle teaches of a method for dynamic distributed data caching comprising:

providing a cache community on a first side of a point of presence, the cache community comprising at least one peer, each peer having associated first content portion (see col.3, lines 58-60: "entry") indicating content obtained from a second side of the point of presence to be cached by the respective peer (see abstract; Fig.5, steps 102→106→120→122; and col.1, line 64-col.2, line 9);

allowing a client to join the cache community (see col.2, lines 5-9: "group grows to include more clients");

updating peer list associated with the cache community to include the client (see col.3, lines 21-31, 35-38, and 55-60 and col.6, lines 55-58), the peer list indicating the peers in the cache community (see Fig.3 and col.4, lines 38-40);

associating a second content portion corresponding to the client with each peer based on joiner of the client (see col.4, lines 7-11), the second content portion being distinct from the first content portion (see col.3, line 60-col.4, line 11);

allocating the first content portion (see col.3, lines 42-44) and the second content portion among the peers in the cache community (see col.2, lines 40-44: "distributes the data"; col.3, lines 19-26: "to distribute replication of data items"; col.4, lines 18-24: "to obtain a copy of the data item"; and col.5, lines 3-11), in response to allowing the client to join the community (implicit: see col.3, line 44).

As per ***claim 16***, Boyle teaches a system for dynamic distributed data caching comprising:

logic encoded on storage and operable to (see col.11, lines 13-17):

provide a cache community on a first side of a point of presence, the cache community comprising at least one peer, each peer having associated first content portion (see col.3, lines 58-60: "entry") indicating content obtained from a second side of the point of presence to be cached by the respective peer (see abstract; Fig.5, steps 102→106→120→122; and col.1, line 64-col.2, line 9);

allow a client to join the cache community (see col.2, lines 5-9: "group grows to include more clients");

update peer list associated with the cache community to include the client (see col.3, lines 21-31, 35-38, and 55-60 and col.6, lines 55-58), the peer list indicating the peers in the cache community (see Fig.3 and col.4, lines 38-40);

associate a second content portion corresponding to the client with each peer based on joiner of the client (see col.4, lines 7-11), the second content portion being distinct from the first content portion (see col.3, line 60-col.4, line 11);

allocate the first content portion (see col.3, lines 42-44) and the second content portion among the peers in the cache community (see col.2, lines 40-44: "distributes the data"; col.3, lines 19-26: "to distribute replication of data items"; col.4, lines 18-24: "to obtain a copy of the data item"; and col.5, lines 3-11), in response to allowing the client to join the community (implicit: see col.3, line 44).

As per ***claim 101***, Boyle teaches a system for dynamic distributed data caching comprising:

means for providing a cache community on a first side of a point of presence, the cache community comprising at least one peer, each peer having associated first content portion (see col.3, lines 58-60: "entry") indicating content obtained from a second side of the point of presence to be cached by the respective peer (see abstract; Fig.5, steps 102→106→120→122; and col.1, line 64-col.2, line 9);

means for allowing a client to join the cache community (see col.2, lines 5-9: "group grows to include more clients");

means for updating peer list associated with the cache community to include the client (see col.3, lines 21-31, 35-38, and 55-60 and col.6, lines 55-58), the peer list indicating the peers in the cache community (see Fig.3 and col.4, lines 38-40);

means for associating a second content portion corresponding to the client with each peer based on joiner of the client (see col.4, lines 7-11), the second content portion being distinct from the first content portion (see col.3, line 60-col.4, line 11);

allocating the first content portion (see col.3, lines 42-44) and the second content portion among the peers in the cache community (see col.2, lines 40-44: "distributes the data"; col.3, lines 19-26: "to distribute replication of data items"; col.4, lines 18-24: "to obtain a copy of the data item"; and col.5, lines 3-11), in response to allowing the client to join the community (implicit: see col.3, line 44).

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States

only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claims 95-100 and 105 are rejected under 35 U.S.C. 102(e) as being anticipated by Maggenti et al. (US 6,477,150 B1).

INDEPENDENT:

As per ***claim 95***, Maggenti teaches a method for dynamic distributed data caching comprising:

communicating a community request to an administration module (see col.3, lines 55-63);

receiving a community list from the administration module in response the community request, the community list including a list of communities (see col.5, lines 38-49 and col.12, lines 16-20);

selecting one of the communities to attempt to join (see col.7, lines 43-46);

generating a join request to attempt to join the selected one of the communities (see col.10, lines 24-29; col.18, lines 36-37; and col.26, lines 65-66);

receiving an allow message associated with the selected one of the communities (see col.6, lines 64-66; col.12; lines 3-7; col.31, lines 49-51; and col.42, lines 48-50);

receiving a peer list associated with the selected one the communities (see col.17, lines 47-52 and col.20, lines 8-14);

receiving content allocated from the peers in the peer list in response to joining the selected one of the communities (see col.2, lines 26-38: "enables data packets from the data network to be distributed to various net members" and col.3, lines 55-63); and providing content for allocation to peers in the peer list in response to joining the selected on of the communities (see col.2, lines 26-38: "enables data packets from the data network to be distributed to various net members" and col.3, lines 55-63).

As per ***claim 98***, Maggenti teaches a system comprising:

logic encoded on storage and operable to (see col.15, lines 44-46):

communicate a community request to an administration module (see col.3, lines 55-63);

receive a community list from the administration module in response the community request, the community list including a list of communities (see col.5, lines 38-49 and col.12, lines 16-20);

select one of the communities to attempt to join (see col.7, lines 43-46);

generate a join request to attempt to join the selected one of the communities (see col.10, lines 24-29; col.18, lines 36-37; and col.26, lines 65-66);

receive an allow message associated with the selected one of the communities (see col.6, lines 64-66; col.12, lines 3-7; col.31, lines 49-51; and col.42, lines 48-50);

receive a peer list associated with the selected one the communities (see col.17, lines 47-52 and col.20, lines 8-14);

receive content allocated from the peers in the peer list in response to joining the selected one of the communities (see col.2, lines 26-38: "enables data packets from the data network to be distributed to various net members" and col.3, lines 55-63); and provide content for allocation to peers in the peer list in response to joining the selected on of the communities (see col.2, lines 26-38: "enables data packets from the data network to be distributed to various net members" and col.3, lines 55-63).

As per ***claim 105***, Maggenti teaches a system for dynamic distributed data caching comprising:

means for communicating a community request to an administration module (see col.3, lines 55-63);

means for receiving a community list from the administration module in response the community request, the community list including a list of communities (see col.5, lines 38-49 and col.12, lines 16-20);

means for selecting one of the communities to attempt to join (see col.7, lines 43-46);

means for generating a join request to attempt to join the selected one of the communities (see col.10, lines 24-29; col.18, lines 36-37; and col.26, lines 65-66);

means for receiving an allow message associated with the selected one of the communities (see col.6, lines 64-66; col.12, lines 3-7; col.31, lines 49-51; and col.42, lines 48-50);

means for receiving a peer list associated with the selected one the communities (see col.17, lines 47-52 and col.20, lines 8-14);

means for receiving content allocated from the peers in the peer list in response to joining the selected one of the communities (see col.2, lines 26-38: "enables data packets from the data network to be distributed to various net members" and col.3, lines 55-63); and

means for providing content for allocation to peers in the peer list in response to joining the selected on of the communities (see col.2, lines 26-38: "enables data packets from the data network to be distributed to various net members" and col.3, lines 55-63).

DEPENDENT:

As per ***claims 96 and 99***, Maggenti further teaches wherein the community request comprises a CRMSG_WAKEUP data message (implicit: see col.13, lines 63-65).

As per ***claims 97 and 100***, Maggenti further teaches wherein the join request comprises a CRMSG_REQUESTTOJOIN data message (implicit: see col.10, lines 24-29; col.18, lines 36-37; and col.26, lines 65-66).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 2-15 and 17-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Boyle (US 5,864,854 A) in view of Maggenti et al. (US 6,477,150 B1).

As per ***claims 2 and 17***, Boyle does not explicitly teach of further comprising: receiving a join request from the client; and determining whether to allow the client join the cache community.

Maggenti teaches of receiving a join request from the client (see col.10, lines 24-29; col.18, lines 36-37; and col.26, lines 65-66); and determining whether to allow the client join the community (see col.5, lines 34-37; col.12, lines 3-7; and col.31, lines 49-51).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to employ the teachings of Maggenti within the method and system of Boyle by implementing receiving a join request from the client and determining whether to allow the client join the community within the method and a system for dynamic distributed data caching because Boyle teaches that the "performance of each client typically improves as the group grows to include more clients, since data items can be obtained more quickly from another client in the group than from a server" (see col.2, lines 5-9) and teaches that the number of members of groups is "preferably selected such that the expected peak number of requests for any data item is not significantly delay by queuing (see col.5, lines 58-61).

As per ***claims 3 and 18***, Boyle does not explicitly teach wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

Maggenti further teaches wherein the join request comprises a CRMSG_REQUESTTOJOIN data message (implicit: see col.10, lines 24-29; col.18, lines 36-37; and col.26, lines 65-66).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to employ the teachings of Maggenti within the method and system of Boyle by implementing receiving any join request message from the client within the method and a system for dynamic distributed data caching because such implementation allows the community to know that a client wants to join before a client has been joined.

As per ***claims 4 and 19***, Boyle does not explicitly teach wherein allowing the client to join the cache community comprises: generating an allow message; associating the peer with the allow message; and communicating the allow message to the client.

Maggenti teaches of generating an allow message (see col.12, lines 3-7 and col.31, lines 49-51); associating the peer with the allow message (inherent); and communicating the allow message to the client (see col.12, lines 3-7 and col.31, lines 49-51).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to employ the teachings of Maggenti within the method and system of Boyle by implementing generating an allow message; associating the peer with the allow message; and communicating the allow message to the client within the method and a system for dynamic distributed data caching because Boyle teaches that the number of members of groups is "preferably selected such that the expected peak number of requests for any data item is not significantly delay by queuing (see col.5, lines 58-61) and by generating, associating, and communicating and acknowledgment is relayed back so that the client device can be notified.

As per ***claims 5 and 20***, Boyle does not explicitly teach wherein allowing the client to join the cache community comprises: generating an allow message comprising the peer list updated to include the clients; communicating the allow message to the client; and communicating the allow message to at least one member associated with the cache community.

Maggenti further teaches wherein allowing the client to join the cache community comprises: generating an allow message comprising the peer list updated to include the clients (see col.17, lines 47-52 and col.20, lines 8-14); communicating the allow message to the client (see claim 4 and 19 rejection above: redundant limitation); and communicating the allow message to at least one member associated with the cache community (see col.12, lines 16-28).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to employ the teachings of Maggenti within the method and system of Boyle by implementing generating an allow message comprising the peer list updated to include the clients; communicating the allow message to the client; and communicating the allow message to at least one member associated with the cache community within the method and a system for dynamic distributed data caching because by generating and communicating the allow message notifies every community device that a client has been joined.

As per ***claims 6 and 21***, Boyle does not explicitly teach wherein the allow message comprises a CRMSG_UPDATEPEERLIST data message.

Maggenti further teaches wherein the allow message comprises a CRMSG_UPDATEPEERLIST data message (implicit: see col.12, lines 16-20).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to employ the teachings of Maggenti within the method and system of Boyle by implementing an allow message comprising CRMSG_UPDATEPEERLIST data message within the method and a system for dynamic distributed data caching because such notification notifies every community device that a client has been joined and updates it the look-up table.

As per ***claims 7 and 22***, Boyle does not explicitly teach wherein peer list associated with the allow message comprises updated peer which includes the client

Maggenti further teaches wherein peer list associated with the allow message comprises updated peer which includes the client (see col.12, lines 16-20).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to employ the teachings of Maggenti within the method and system of Boyle by implementing a peer list associated with the allow message comprising updated peer which includes the client within the method and a system for dynamic distributed data caching because such notification notifies every community device that a client has been joined and updates it the look-up table.

As per ***claims 8 and 23***, Boyle further teaches wherein the point of presence is an ISP (implicit: see Fig.1 and col.2, lines 48-58).

As per ***claims 9 and 24***, Boyle further teaches wherein a one of the peers comprises a member (see col.3, lines 58-61).

As per ***claims 10 and 25***, Boyle does not explicitly teach wherein one of the peers comprises a master.

Maggenti teaches of a master (see col.3, lines 58-65: "communication manager (CM)").

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to employ the teachings of Maggenti within the method and system of Boyle by implementing a master within the method and a system for dynamic distributed data caching because such an implementation provides centralized control for administering the functionalities.

As per ***claims 11 and 26***, Boyle further teaches wherein associating a respective second content portion comprises: allocating respective second content portions peers in the peer list (see Fig.2 and Fig.3); and updating an allocation table indicate the second content portion associated with the peers (see col.8, lines 59-62).

As per ***claims 12 and 27***, Boyle further teaches wherein the second content portions are distinct (implicit: see col.3, line 60-col.4, line 17: all the field may be distinct).

As per ***claims 13 and 28***, Boyle further teaches wherein at least two of the second content portions overlap (implicit: see col.3, line 60-col.4, line 17: some of the fields may be the same).

As per ***claims 14 and 29***, Boyle further teaches wherein the first and second content portions respectively comprise a plurality Internet Protocol domain names (see col.3, lines 60-48).

As per ***claims 15 and 30***, Boyle teaches of further comprising removing the association between the first content portions and the peers (inherent).

Response to Arguments

7. Applicant's arguments filed February 27, 2006 have been fully considered but they are not persuasive. See Below.

With regards to claims 1, 16, and 101, the applicant(s) argue that the claimed invention is patentable because U.S. Patent 5,864,854 (*Boyle*) does not teach the amended claim limitation of "*allocate the first content portion and the second content portion among the peers in the cache community, in response to allowing the client to join the community*".

In response to the argument, the examiner has sited several locations in *Boyle* teaching this limitation (see rejection above).

With regard to claims 95-100 and 105, the applicant(s) argue that there is no motivation to combine the references U.S. Patent 6,477,150 (*Maggenti* et al.) and *Boyle*. The applicant(s) further argue that even if a motivation existed, "neither Boyle nor the Maggenti, et al. patent provide for content allocation upon joiner into a community as required in the claimed invention"

In response to the argument no motivation to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988) and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, *Boyle* suggests other clients can join the cache community to enhance performance (see column 2, lines 5-9).

Furthermore, it has been concluded with respect to the amendment that *Maggenti* alone teaches all the limitations of claim 95-100 and 105 (see rejection above). The amendment deleted the limitation "dynamic cache module" in which *Boyle* was relied upon to teach this limitation. The recitation "dynamic distributed data caching" which is recited in the preamble has not been given patentable weight. A preamble is generally not accorded any patentable weight where it merely recites the purpose of a process or the intended use of a structure, and where the body of the claim does not depend on the preamble for completeness but, instead, the process steps or structural limitations are able to stand alone. See *In re Hirao*, 535 F.2d 67, 190 USPQ 15 (CCPA 1976) and *Kropa v. Robie*, 187 F.2d 150, 152, 88 USPQ 478, 481 (CCPA 1951).

Maggenti clearly teaches of allocating contents upon all members of the community (see rejection above).

Conclusion

8. For the reason above claim 1-30, 95-101, and 105 remain rejected.

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael Y. Won whose telephone number is 571-272-3993. The examiner can normally be reached on M-Th: 7AM-5PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Saleh Najjar can be reached on 571-272-4006. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Michael Won

May 11, 2006

Index of Claims

Application/Control No. 09/759,406	**Applicant(s)/Patent under Reexamination** LOWERY ET AL.	
Examiner Michael Y. Won	**Art Unit** 2155	

√	**Rejected**		–	**(Through numeral) Cancelled**		N	**Non-Elected**		A	**Appeal**
=	**Allowed**		÷	**Restricted**		I	**Interference**		O	**Objected**

Claim		Date								
Final	Original	5/11/06								
	1	√								
	2	↑								
	3									
	4									
	5									
	6									
	7									
	8									
	9									
	10									
	11									
	12									
	13									
	14									
	15									
	16									
	17									
	18									
	19									
	20									
	21									
	22									
	23									
	24									
	25									
	26									
	27									
	28									
	29	↓								
	30	√								
	~~31~~									
	32	↑								
	33									
	34									
	35									
	36									
	37									
	38									
	39									
	40									
	41									
	42									
	43									
	44									
	45									
	46									
	47									
	48									
	49	↓								
	~~50~~									
	~~51~~									
	52	↑								
	53									
	54									
	55									
	56									
	57									
	58									
	59									
	60									
	61									
	62									
	63									
	64									
	65									
	66									
	67									
	68									
	69									
	70									
	71									
	72									
	73									
	74									
	75									
	76									
	77									
	78									
	79									
	80									
	81									
	82									
	83									
	84									
	85									
	86									
	87									
	88									
	89									
	90									
	91									
	92									
	93	↓								
	~~94~~									
	95	√								
	96	↑								
	97									
	98									
	99	↓								
	100	√								
	101	√								
	~~102~~									
	~~103~~									
	~~104~~									
	105	√								
	106									
	107									
	108									
	109									
	110									
	111									
	112									
	113									
	114									
	115									
	116									
	117									
	118									
	119									
	120									
	121									
	122									
	123									
	124									
	125									
	126									
	127									
	128									
	129									
	130									
	131									
	132									
	133									
	134									
	135									
	136									
	137									
	138									
	139									
	140									
	141									
	142									
	143									
	144									
	145									
	146									
	147									
	148									
	149									
	150									

U.S. Patent and Trademark Office

Part of Paper No. 20060511

Search Notes	**Application/Control No.** 09/759,406	**Applicant(s)/Patent under Reexamination** LOWERY ET AL.
	Examiner Michael Y. Won	**Art Unit** 2155

SEARCHED			
Class	Subclass	Date	Examiner
709	203, 205, 215, 216, 248	5/11/2006	MW

SEARCH NOTES (INCLUDING SEARCH STRATEGY)		
	DATE	EXMR
EAST: USPAT; USPG-PUB; DERWENT; EPO; JPO	5/11/2006	MW

INTERFERENCE SEARCHED			
Class	Subclass	Date	Examiner

U.S. Patent and Trademark Office

Part of Paper No. 20060511

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Keith A. Lowery et al.
Serial No.:	09/759,406
Filing Date:	January 12, 2001
Confirmation No.:	2308
Group Art Unit:	2155
Examiner:	Michael Won
Title:	METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING

Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Dear Sir:

RESPONSE TO EXAMINER'S ACTION

In response to the Official Action mailed May 16, 2006, Applicants respectfully request the Examiner to reconsider the rejection of the claims and the objections to the Application in view of the following amendments and/or comments as set forth below.

IN THE CLAIMS

1. (Previously Presented) A method for dynamic distributed data caching comprising:

providing a cache community on a first side of a point of presence, the cache community comprising at least one peer, the cache community being associated with ~~each peer having an associated first content portion indicating~~ content obtained from a second side of the point of presence, the content being ~~to be~~ cached by the ~~respective~~ at least one peer;

allowing a client to join the cache community;

updating a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community;

associating ~~a second content portion corresponding to the client with each peer~~ the content with the client based on joinder of the client~~, the second content portion being distinct from the first content portion~~;

re-allocating the cache storage of the content ~~allocating the first content portion and the second content portion~~ among the peers in the cache community in response to allowing the client to join the community.

2. (Original) The method for dynamic distributed data caching according to Claim 1 and further comprising:

receiving a join request from the client; and

determining whether to allow the client to join the cache community.

3. (Original) The method dynamic distributed data caching according to Claim 2, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

4. (Original) The method for dynamic distributed data caching according to Claim 1, wherein allowing the client to join the cache community comprises:

generating an allow message;

associating the peer list with the allow message; and

communicating the allow message to the client.

5. (Original) The method for dynamic distributed data caching according to Claim 4, wherein allowing the client to joint the cache community comprises:

generating an allow message comprising the peer list updated to include the client;

communicating the allow message to the client; and

communicating the allow message to at least one member associated with the cache community.

6. (Original) The method for dynamic distributed data caching according to Claim 4, wherein the allow message comprises a CRMSG_UPDATEPEERLIST data message.

7. (Original) The method for dynamic distributed caching according to Claim 4, wherein the peer list associated with the allow message comprises the updated peer list which includes the client.

8. (Previously Presented) The method for dynamic distributed data caching according to Claim 1, wherein the point of presence is an ISP.

9. (Original) The method for dynamic distributed data caching according to Claim 1, wherein a one of the peers comprises a member.

10. (Original) The method for dynamic distributed data caching according to Claim 1, wherein a one of the peers comprises a master.

11. (Currently Amended) The method for dynamic distributed data caching according to Claim 1, wherein associating ~~a respective second content portion~~ the content with the client comprises:

allocating ~~respective second~~ the content ~~portions~~ to peers in the peer list; and

updating an allocation table to indicate the ~~second~~ content ~~portion~~ associated with the peers.

12. (Canceled).

13. (Currently Amended) The method for dynamic distributed data caching according to Claim 11, wherein ~~at least two of the second content portions overlap~~ the content is cached at more than one peer.

14. (Currently Amended) The method for dynamic distributed data caching according to Claim 11, wherein the ~~first and second~~ content ~~portions respectively comprise~~ includes a plurality of internet protocol domain names.

15. (Currently Amended) The method for dynamic distributed data caching according to Claim 1 and further comprising removing the association between the ~~first~~ content ~~portions~~ and the peers.

16. (Currently Amended) A system for dynamic distributed data caching comprising:

logic encoded on storage and operable to:

provide a cache community on a first side of a point of presence, the cache community comprising at least one peer, the cache community being associated with ~~each peer having an associated first content portion indicating~~ content obtained from a second side of the point of presence, the content being ~~to be~~ cached by the ~~respective~~ at least one peer;

allow a client to join the cache community;

update a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community;

associate the content with ~~a second content portion corresponding to~~ the client ~~with each peer~~ based on joinder of the client~~, the second content portion being distinct from the first content portion~~;

re-allocate the cache storage of the content ~~allocate the first content portion and the second content portion~~ among the peers in the cache community in response to allowing the client to join the community.

17. (Original) The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to:

receive a join request from the client; and

determine whether to allow the client to join the cache community.

18. (Original) The system for dynamic distributed data caching according to Claim 17, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

19. (Original) The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to:

generate an allow message;

associate the peer list with the allow message;

communicate the allow message to the client.

20. (Original) The system for dynamic distributed data caching according to Claim 19, wherein the logic is further operable to:

generate an allow message comprising the peer list updated to include the client;

communicate the allow message to the client; and

communicate the allow message to at least one member associated with the cache community.

21. (Original) The system for dynamic distributed data caching according to Claim 19, wherein the allow message comprises a CRMSG_UPDATEPEERLIST data message.

22. (Original) The system for dynamic distributed caching according to Claim 19, wherein the peer list associated with the allow message comprises the updated peer list which includes the client.

23. (Previously Presented) The system for dynamic distributed data caching according to Claim 16, wherein the point of presence is an ISP.

24. (Original) The system for dynamic distributed data caching according to Claim 16, wherein a one of the peers comprises a member.

25. (Original) The system for dynamic distributed data caching according to Claim 16, wherein a one of the peers comprises a master.

26. (Currently Amended) The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to:

allocate ~~respective second content portions~~ the content to peers in the peer list; and

update an allocation table to indicate the ~~second~~ content ~~portion~~ associated with the peers.

27. (Canceled).

28. (Currently Amended) The system for dynamic distributed data caching according to Claim 26, wherein ~~at least two of the second content portions overlap~~ the content is cached at more than one peer.

29. (Currently Amended) The system for dynamic distributed data caching according to Claim 26, wherein the ~~first and second~~ content ~~portions respectively comprise~~ includes a plurality of internet protocol domain names.

30. (Currently Amended) The system for dynamic distributed data caching according to Claim 16, wherein the logic is further operable to remove the association between the ~~first~~ content ~~portions~~ and the peers.

31. (Canceled)

32. (Canceled)

33. (Canceled)

34. (Canceled)

35. (Canceled)

36. (Canceled)

37. (Canceled)

38. (Canceled)

39. (Canceled)

40. (Canceled)

41. (Canceled)

42. (Canceled)

43. (Canceled)

44. (Canceled)

45. (Canceled)

46. (Canceled)

47. (Canceled)

48. (Canceled)

49. (Canceled)

50. (Canceled)

51. (Canceled)

52. (Canceled)

53. (Canceled)

54. (Canceled)

55. (Canceled)

56. (Canceled)

57. (Canceled)

58. (Canceled)

59. (Canceled)

60. (Canceled)

61. (Canceled)

62. (Canceled)

63. (Canceled)

64. (Canceled)

65. (Canceled)

66. (Canceled)

67. (Canceled)

68. (Canceled)

69. (Canceled)

70. (Canceled)

71. (Canceled)

72. (Canceled)

73. (Canceled)

74. (Canceled)

75. (Canceled)

76. (Canceled)

77. (Canceled)

78. (Canceled)

79. (Canceled)

80. (Canceled)

81. (Canceled)

82. (Canceled)

83. (Canceled)

84. (Canceled)

85. (Canceled)

86. (Canceled)

87. (Canceled)

88. (Canceled)

89. (Canceled)

90. (Canceled)

91. (Canceled)

92. (Canceled)

93. (Canceled)

94. (Canceled)

95. (Currently Amended) A method for dynamic distributed data caching comprising:

communicating a community request to an administration module;

receiving a community list from the administration module in response to the community request, the community list including a list of communities;

selecting one of the communities to attempt to join;

generating a join request to attempt to join the selected one of the communities;

receiving an allow message associated with the selected one of the communities;

receiving a peer list associated with the selected one of the communities;

receiving content allocated ~~from peers~~ for storage in caches of peers in the peer list for cache storage re-allocation in response to joining the selected one of the communities; and

providing content for cache storage re-allocation ~~allocation~~ to peers in the peer list in response to joining the selected one of the communities.

96. (Original) The method for dynamic distributed data caching according to Claim 95, wherein the community request comprises a CRMSG_WAKEUP data message.

97. (Original) The method for dynamic distributed data caching according to Claim 95, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

98. (Currently Amended) A system for dynamic distributed data caching comprising:

logic encoded on storage and operable to:

communicate a community request to an administration module;

receive a community list from the administration module in response to the community request, the community list including a list of communities;

select one of the communities to attempt to join;

generate a join request to attempt to join the selected one of the communities;

receive an allow message associated with the selected one of the communities;

receive a peer list associated with the selected one of the communities;

receive content allocated ~~from peers~~ for storage in caches of peers in the peer list for cache storage re-allocation in response to joining the selected one of the communities; and

provide content for cache storage re-allocation ~~allocation~~ to peers in the peer list in response to joining the selected one of the communities.

99. (Original) The system for dynamic distributed data caching according to Claim 98, wherein the community request comprises a CRMSG_WAKEUP data message.

100. (Original) The system for dynamic distributed data caching according to Claim 98, wherein the join request comprises a CRMSG_REQUESTTOJOIN data message.

101. (Previously presented) A system for dynamic distributed data caching comprising:

means for providing a cache community on a first side of a point of presence, the cache community comprising at least one peer, the cache community being associated with ~~each peer having an associated first content portion indicating~~ content obtained from a second side of the point of presence, the content being ~~to be~~ cached by the ~~respective~~ at least one peer;

means for allowing a client to join the cache community;

means for updating a peer list associated with the cache community to include the client, the peer list indicating the peers in the cache community;

means for associating ~~a second content portion corresponding to~~ the content with the client ~~with each peer~~ based on joinder of the client~~, the second content portion being distinct from the first content portion~~;

means for re-allocating the cache storage of the content ~~allocating the first content portion and the second content portion~~ among the peers in the cache community in response to allowing the client to join the community.

102. (Canceled)

103. (Canceled)

104. (Canceled)

105. (Previously Presented) A system for dynamic distributed data caching comprising:

means for communicating a community request to an administration module;

means for receiving a community list from the administration module in response to the community request, the community list including a list of communities;

means for selecting one of the communities to attempt to join;

means for generating a join request to attempt to join the selected one of the communities;

means for receiving an allow message associated with the selected one of the communities;

means for receiving a peer list associated with the selected one of the communities;

means for receiving content allocated ~~from peers~~ for storage in caches of peers in the peer list for cache storage re-allocation in response to joining the selected one of the communities; and

means for providing content for cache storage re-allocation ~~allocation~~ to peers in the peer list in response to joining the selected one of the communities.

Please cancel Claims 12 and 27 as indicated above without prejudice or disclaimer.

REMARKS

This Application has been carefully reviewed in light of the Official Action mailed May 16, 2006. In order to advance prosecution of this Application, Claims 1, 11, 13-16, 26, 28-30, 95, 98, and 101 have been amended and Claims 12 and 27 have been canceled without prejudice or disclaimer. Applicant respectfully requests reconsideration and favorable action in this Application.

Claims 1, 16, and 101 are rejected under 35 U.S.C. §102(b) as being anticipated by Boyle. Independent Claims 1, 16, and 101 recite in general an ability to re-allocating the cache storage of the content among the peers in the cache community in response to allowing the client to join the community. By contrast, the Boyle patent merely discloses the use of a group look-up table that can be used to distribute a requested data item to a requesting client in the group. The Boyle patent does not disclose a capability to re-allocate the cache storage of content upon adding a new client to a community as provided by the claimed invention. Thus, the Boyle patent only provides disclosure for distribution of requested data but not how to allocate the cache storage of data as provided in the claimed invention. Support for the above recitation can be found at page 44, lines 13-19, of Applicant's specification. Therefore, Applicant respectfully submits that Claims 1, 16, and 101 are not anticipated by the Boyle patent.

Claims 95-100 and 105 are rejected under 35 U.S.C. § 102(e) as being anticipated by Maggenti, et al. Independent Claims 95, 98, and 105 recite in general an ability to receive content allocated for storage in caches of peers in the peer list for cache storage re-allocation in response to joining the selected one of the communities and provide content for cache storage re-allocation to peers in the peer list in response to joining the selected one of the communities. By contrast, the Maggenti, et al. patent merely discloses a communication manager for distribution of data packets in general to communication devices and does not provide for cache storage re-allocation upon joining a community as required in the claimed invention. Support for the above recitation can be found at page 44, lines 13-19, of Applicants specification. Therefore, Applicant respectfully submits that Claims 95-100 and 105 are not anticipated by the Maggenti, et al. patent.

Claims 2-15 and 17-30 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Boyle in view of Maggenti, et al. Independent Claim 1, from which Claims 2-15 depend, and Independent Claim 16, from which Claims 17-30 depend, have been shown above to be patentably distinct from the Boyle patent. Moreover, the Maggenti, et al. patent does not include any additional disclosure combinable with the Boyle patent that would be material to patentability of these claims. Therefore, Applicant respectfully submit s that Claims 2-15 and 17-30 are patentably distinct over the proposed Boyle - Maggenti, et al. combination.

CONCLUSION

Applicant has now made an earnest attempt to place this Application in condition for allowance. For the foregoing reasons, and for other reasons clearly apparent, Applicant respectfully requests full allowance of all pending claims.

If the Examiner feels that a telephone conference would advance prosecution of this Application in any manner, the undersigned attorney for Applicant stands ready to conduct such a conference at the convenience of the Examiner.

The Commissioner is hereby authorized to charge any required fees or credit any overpayments associated with this Application to Deposit Account No. 02-0384 of BAKER BOTTS L.L.P.

Respectfully submitted,
BAKER BOTTS L.L.P.
Attorneys for Applicants

Charles S. Fish
Reg. No. 35,870

August 16, 2006

CORRESPONDENCE ADDRESS:
2001 Ross Avenue, Suite 600
Dallas, TX 75201-2980
(214) 953-6507
Customer Number: 05073

Electronic Acknowledgement Receipt

EFS ID:	1156762
Application Number:	09759406
Confirmation Number:	2308
Title of Invention:	Method and system for dynamic distributed data caching
First Named Inventor:	Keith A. Lowery
Correspondence Address:	Baker Botts L.L.P. - 2001 Rose Avenue - Dallas TX 75201-2980 US 2149536680 -
Filer:	Charles Spaulding Fish/Judy Baggett
Filer Authorized By:	Charles Spaulding Fish
Attorney Docket Number:	066241.0117
Receipt Date:	16-AUG-2006
Filing Date:	12-JAN-2001
Time Stamp:	13:04:45
Application Type:	Utility
International Application Number:	

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)	Multi Part	Pages
1		117res.pdf	499063	yes	18

	Multipart Description		
	Doc Desc	**Start**	**End**
	Amendment - After Non-Final Rejection	1	1
	Claims	2	15
	Applicant Arguments/Remarks Made in an Amendment	16	18

Warnings:

Information:

Total Files Size (in bytes):	499063

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

PATENT APPLICATION FEE DETERMINATION RECORD

Effective October 1, 2004

Application or Docket Number

09/759406

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)	SMALL ENTITY TYPE □		OR	OTHER THAN SMALL ENTITY	
FOR 02/27/06	NUMBER FILED	NUMBER EXTRA	RATE	FEE		RATE	FEE
BASIC FEE					OR		790.00
TOTAL CLAIMS	38 minus 105	* 0	x$11=		OR	x$22=	
INDEPENDENT CLAIMS	6 minus 15	* 0	x41=		OR	x82=	
MULTIPLE DEPENDENT CLAIM PRESENT			+135=		OR	+270=	
			TOTAL	0	OR	TOTAL	

* If the difference in column 1 is less than zero, enter "0" in column 2

CLAIMS AS AMENDED - PART II

8-16-06

AMENDMENT A	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE	ADDITIONAL FEE	OR	OTHER THAN SMALL ENTITY RATE	ADDITIONAL FEE
Total	* 35	Minus	** 38	= —	x$11=		OR	x$22=	
Independent	* 6	Minus	*** 6	= —	x41=		OR	x82=	
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM					+135=		OR	+270=	
					TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT B	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE	ADDITIONAL FEE	OR	RATE	ADDITIONAL FEE
Total	*	Minus	**	=	x$11=		OR	x$22=	
Independent	*	Minus	***	=	x41=		OR	x82=	
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM					+135=		OR	+270=	
					TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE	ADDITIONAL FEE	OR	RATE	ADDITIONAL FEE
Total	*	Minus	**	=	x$11=		OR	x$22=	
Independent	*	Minus	***	=	x41=		OR	x82=	
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM					+135=		OR	+270=	
					TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

*U.S. Government Printing Office: 1997 - 430-571/69194

Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	13	napster.as.	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2006/10/18 10:24
L2	3	napster.as. and cach$5	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2006/10/18 10:26
L3	1	napster.as. and (allocat$5 or re-allocat$5 or reallocat$5)	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2006/10/18 10:26
L5	11	(709/203,214.ccls. or 711/119,126, 130,135.ccls.) and cache$5 adj3 (group or community or farm or cluster) and joining	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2006/10/18 11:35
L6	6	(709/203,214.ccls. or 711/119,126, 130,135.ccls.) and cache$5 adj3 (group or community or farm or cluster) and joining and (reallocat$5 or re-allocat$5 or re-assign$5 or reassign$5 or redistribut$5 or re-distribut$5)	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2006/10/18 11:37
S1	7	709/217,213.ccls. and dynamic$5 with distributed with cach$5 and peer and (community or group) and peer	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2006/10/18 10:24
S2	24	709/217,213.ccls. and dynamic$5 with cach$5 and peer and (community or group)	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2004/07/21 14:26
S3	25	709/217,213.ccls. and dynamic$5 with cach$5 and peer$9 and (community or group)	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2004/07/21 14:44
S4	11	709/217,213.ccls. and dynamic$5 with cach$5 and peer$9 and (community or group) and join$5	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2004/07/21 14:29
S5	3	dynamic$5 with distributed with cach$5 and (community or group) and peer and join$3 and updat$3 and peer with list	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2004/07/21 14:45
S6	0	709/713,714,715,716,717.ccls. and cach$5.ab. and (community or group) and peer$9	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2004/07/21 14:44

EAST Search History

S7	275	cach$5.ab. and (community or group) and peer$9	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/21 14:43

S8	0	709/713,714,715,716,717.ccls. and cach$5 and (community or group) and peer$9	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/21 14:44

S9	184	dynamic$5 and cach$5.ab. and peer$9 and (community or group)	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/21 14:45

S10	7	dynamic$5 and distributed and cach$5.ab. and (community or group) and peer and join$3 and updat$3 and peer with list	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/21 14:46

S11	48	dynamic$5 and distributed and cach$5.ab. and (community or group) and peer and join$3 and updat$3 and list	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/21 14:46

S12	22	dynamic with distributed with cach$5 and (community or group) and peer$9	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/21 15:10

S13	281	dynamic and distributed with cach$5 and (community or group) and peer$9	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/21 15:10

S14	138	dynamic$5 and distributed with cach$5 and (community or group) and peer$9 and join$5 and updat$5 and member	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/21 15:13

S15	92	dynamic$5 and distributed with cach$5 and (community or group) and peer$9 and join$5 and updat$5 and member and internet and (http or (hypertext adj transfer adj protocol))	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/21 15:29

S16	12	cach$5.ab. and distributed and dynamic$5 and (community or group) and peer$9 and join$5 and updat$5 and member and internet and (http or (hypertext adj transfer adj protocol))	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/21 15:15

S17	56	dynamic$5 and distributed with cach$5 and (community or group) and peer$9 and join$5 with request and updat$5 and member and internet and (http or (hypertext adj transfer adj protocol))	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/22 10:22

S18	10	cach$3 adj community	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2004/07/22 10:35
S19	3	709/214,216.ccls. and cach$3 and community and member$5 and join$5 and shar$3	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2004/07/22 10:37
S20	13	cach$3.ab. and community and member$5 and join$5 and shar$3 and distributed and dynamic$5	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2004/07/22 10:38
S21	12	cach$3.ab. and community and member$5 and join$5 and shar$3 and distributed and dynamic$5 and internet	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2004/07/22 10:46
S22	136	(p2p or peer-to-peer or (peer adj to adj peer)) and cach$3 and community and member$5 and join$5 and shar$3 and distributed and dynamic$5 and internet	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2004/07/22 10:49
S23	270	(p2p or peer-to-peer or (peer adj to adj peer)) and cach$3 and member$5 and join$5 and shar$3 and distributed and dynamic$5 and internet	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2004/07/22 11:42
S24	1181	((dynamic adj reef adj protocol) or drp)	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2004/07/22 11:43
S25	2	((dynamic adj reef adj protocol) or drp) and member and join$5 and process$3 adj power	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2004/07/22 11:44
S26	58	((dynamic adj reef adj protocol) or drp) and cach$3	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2004/07/22 11:44
S27	12	((dynamic adj reef adj protocol) or drp) and cach$3 and join$3	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2004/07/22 11:45
S28	3	((dynamic adj reef adj protocol) or drp) and cach$3.bsum. and join$3	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2004/07/22 11:48
S29	2	((dynamic adj reef adj protocol) or drp) and $wakeup with message	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2004/07/22 11:48

EAST Search History

S30	17	709/212.ccls. and cach$3 and group and join$4	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/22 12:37

S31	14	709/212.ccls. and cach$3 and group and join$4 and distributed	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/22 12:38

S32	0	709/212.ccls. and cach$3 and group and client with join$4 and distributed	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/22 12:39

S33	10	709/212.ccls. and cach$3 and group and join$4 and distributed and client	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/22 12:41

S34	24	cach$3 and group and client with join$4 and distributed and peer$ and member$5 and community	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/22 13:28

S35	13	cach$3 adj server and group and client with join$4 and distributed and peer$ and member$5 and community	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/22 13:05

S36	28	cach$3 adj server and client and distributed and peer$ and member$5 and community	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/22 13:06

S37	2	(cach$3 adj server).ab. and cache with community	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/22 13:07

S38	24	cach$3 and group and client with join$4 and distributed and peer$ and member$5 and community and shar$3	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/22 13:29

S39	79	cach$3 and group and client with join$4 and distributed and peer$ and shar$3	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/22 13:36

S40	47	cach$3 with communit$5 and cach$3 with server	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/22 13:39

S41	44	cach$3 with communit$5 and cach$3 with server and local$5	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/22 13:39

EAST Search History

S42	34	cach$3 with communit$5 and cach$3 with server and local$5 and distributed	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/22 13:40

S43	9	cach$3 adj3 communit$5 and cach$3 adj2 server and local$5 and distributed	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/22 13:40

S44	324	distributed with client with cach$3	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/26 13:22

S45	324	(distributed with client with cach$3)". bsum"	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/26 13:22

S46	111	(distributed with client with cach$3). bsum.	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/26 13:22

S47	72	(distributed with client with cach$3). bsum. and group	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/26 13:28

S48	20	(distributed with client with cach$3). bsum. and group and peer	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/26 13:24

S49	26	(distributed with client with cach$3). bsum. and group and cach$3 and expir$6	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/26 13:27

S50	72	(distributed with client with cach$3). bsum. and group and cach$5 and request$3	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2004/07/26 13:29

S51	2	cach$5 adj2 community with client adj2 join$5	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2005/02/23 14:25

S52	0	forward adj proxy and cach$6 with community	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2005/02/23 14:37

S53	63	cach$6 with community	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2005/02/23 14:43

EAST Search History

S54	11	cach$6 with community with client	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2005/02/23 14:39
S55	11	client with community with cach$5	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2005/02/23 14:39
S56	12	cach$6 with community and join	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2005/02/23 14:45
S57	12	cach$6 with community and join$5	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2005/02/23 14:48
S58	280	cach$6 and community and join$5 and (p2p or "peer-to-peer" or peer adj2 peer)	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2005/02/23 14:50
S59	106	cach$6 and community and join$5 and (p2p or "peer-to-peer" or peer adj2 peer) and (point adj2 presence or pop)	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2005/02/23 14:50
S60	19	cach$6 and community and request with join$5 and (p2p or "peer-to-peer" or peer adj2 peer) and (point adj2 presence or pop)	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2005/02/23 14:51
S61	27	cach$6 and community and join$5 and (p2p or "peer-to-peer" or peer adj2 peer) and (point adj2 presence or pop) and updat$5 with list	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2005/02/23 14:52
S62	103	cach$6 and community and join$5 and (p2p or "peer-to-peer" or peer adj2 peer) and (point adj2 presence or pop) and updat$5	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2005/02/23 14:53
S63	77	cach$6 and community and join$5 and leav$5 and (p2p or "peer-to-peer" or peer adj2 peer) and (point adj2 presence or pop) and updat$5	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2005/02/23 14:53
S64	53	cach$6 and community and join$5 and leav$5 and (p2p or "peer-to-peer" or peer adj2 peer) and (point adj2 presence or pop) and updat$5	USPAT	OR	ON	2005/02/23 14:53
S65	151	dynamic$5 and distributed with cach$5 and (community or group) and peer$9 and join$5 and updat$5 and member	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2005/02/25 08:16

S66	149	S65 and member$5 and shar$5 and request$5	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2005/02/25 08:18

S67	11	709/217,213.ccls. and dynamic$5 with cach$5 and peer$9 and (community or group) and join$5	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2005/02/25 08:17

S68	9	S67 and member$5 and shar$5 and request$5	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2005/02/25 08:17

S69	149	S65 and member$5 and shar$5 and request$5	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2005/02/25 08:20

S70	40	S65 and member$5 and shar$5 and request$5	USPAT	OR	ON	2005/02/25 08:21
S71	28	S65 and member$5 and shar$5 and request$5 and client	USPAT	OR	ON	2005/02/25 08:21

S72	0	"crmsg_requesttojoin"	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2005/02/25 09:06

S73	2	"$requesttojoin"	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2005/02/25 09:06

S74	0	cach$5 and communicty and join	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2005/02/25 09:43

S75	45	cach$5 and community and join$5 and shar$5 and request$3 with join$5 and ((internet adj service adj provider) or isp) and peer$10 and list and updat$3	US-PGPUB;
			USPAT;
			EPO;
			DERWENT
				OR	ON	2005/06/15 11:43

S76	9	cach$5 and community and join$5 and shar$5 and request$3 with join$5 and ((internet adj service adj provider) or isp) and peer$10 and list and updat$3	USPAT	OR	ON	2005/02/25 09:46
S77	102	cach$5 and community and join$5 and shar$5 and peer$10 and list and updat$3 and internet	USPAT	OR	ON	2005/02/25 09:46
S78	25	cach$5 and community and join$5 and shar$5 and peer$10 and list and updat$3 and internet and request$3 with join$5	USPAT	OR	ON	2005/02/25 09:47

EAST Search History

S79	1	cach$5.ab. and community and join$5 and shar$5 and peer$10 and list and updat$3 and internet and request$3 with join$5	USPAT	OR	ON	2005/02/25 09:48
S80	5	cach$5 and community and client with join$5 and shar$5 and peer$10 and list and updat$3 and internet and request$3 with join$5	USPAT	OR	ON	2005/02/25 09:48
S81	24	cach$5 and community and client and join$5 and shar$5 and peer$10 and list and updat$3 and internet and request$3 with join$5	USPAT	OR	ON	2005/02/25 09:55
S82	36	cach$5 and group and client and join$5 and shar$5 and peer$10 and list and updat$3 and internet and request$3 with join$5	USPAT	OR	ON	2005/02/25 10:07
S83	38	cach$5 and client and join$5 and shar$5 and peer$10 and list and updat$3 and internet and request$3 with join$5	USPAT	OR	ON	2005/02/25 09:57
S84	0	point adj "of" adj presence and cache with (community or group) and peer	USPAT	OR	ON	2005/02/25 10:09
S85	0	point adj "of" adj presence	USPAT	OR	ON	2005/02/25 10:09
S86	3270	point adj2 presence	USPAT	OR	ON	2005/02/25 10:09
S87	2	point adj2 presence and cache with (community or group) and peer	USPAT	OR	ON	2005/02/25 10:10
S88	1	point adj2 presence and cache with (community or group) and peer and join$5	USPAT	OR	ON	2005/02/25 10:29
S89	1	point adj2 presence and cache with (community or group) and peer and join$5 and shar$5 and request$5	USPAT	OR	ON	2005/02/25 10:30
S90	0	point adj2 presence and cache with (community or group) and peer and join$5 and shar$5 and request$5 and updat$5 and list$5	USPAT	OR	ON	2005/02/25 10:30
S91	1	point adj2 presence and cache with (community or group) and peer and join$5 and shar$5 and request$5 and updat$5	USPAT	OR	ON	2005/02/25 10:37
S92	1	point adj2 presence and cache with (community or group) and peer$5 and join$5 and shar$5 and request$5 and updat$5	USPAT	OR	ON	2005/02/25 10:42
S93	2	point adj2 presence and cache with (community or group) and peer$12	USPAT	OR	ON	2005/02/25 10:42

EAST Search History

S94	0	point adj2 presence and cach$5 with (community or group) with client and peer$12	USPAT	OR	ON	2005/02/25 10:43
S95	8	cach$5 with (community or group) with client and peer$12	USPAT	OR	ON	2005/02/25 10:45
S96	319	cach$5 with client and peer$12	USPAT	OR	ON	2005/02/25 10:46
S97	24	cach$5 with client and peer$12 and join$5 with (community or group)	USPAT	OR	ON	2005/02/25 10:46
S98	14	cache adj2 community	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2005/06/14 14:48
S99	7	cache adj2 community	USPAT; EPO; DERWENT	OR	ON	2005/06/14 14:35
S100	1	(client adj side or client-side) adj cach$7 and community	USPAT; EPO; DERWENT	OR	ON	2005/06/14 14:36
S101	214	client with cach$7 and community	USPAT; EPO; DERWENT	OR	ON	2005/06/14 14:45
S102	4	client with cach$7 with community	USPAT; EPO; DERWENT	OR	ON	2005/06/14 14:35
S103	74	(client adj side or client-side) adj cach$7	USPAT; EPO; DERWENT	OR	ON	2005/06/14 14:36
S104	4	(client adj side or client-side) adj cach$7 and client with join$5	USPAT; EPO; DERWENT	OR	ON	2005/06/14 14:37
S105	6	(client adj side or client-side) adj cach$7 and join$5 and plural$5 with client	USPAT; EPO; DERWENT	OR	ON	2005/06/14 14:40
S106	8	cach$7 adj3 (community or group) and join$5 and plural$5 with client	USPAT; EPO; DERWENT	OR	ON	2005/06/14 14:45
S107	5	client with cach$7 and community and join$4 with request	USPAT; EPO; DERWENT	OR	ON	2005/06/14 14:45
S108	14	cach$7 adj2 community	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2005/06/14 14:49
S109	0	cient with cach$7 adj2 community	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2005/06/14 14:49

S110	0	cient with cach$7 adj2 (community or group)	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2005/06/14 14:49
S111	75	(cach$5 and client and (community or group)).ab.	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2005/06/14 14:59
S112	2	(cach$5 and client and (community or group)).ab. and request with client with join$4	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2005/06/14 15:00
S113	507	cach$5 with (community or group) and join$5 with (community or group)	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2005/06/15 11:46
S114	13	cach$5 with (community or group) and join$5 with (community or group) and (wakeup or wake adj up) and (request adj2 join or request$4join)	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2005/06/15 11:52
S115	141	cach$5 adj2 (community or group) and join$5 with (community or group)	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2005/06/15 11:46
S116	53	cach$5 adj2 (community or group) and join$5 with (community or group)	USPAT; EPO; DERWENT	OR	ON	2005/06/15 11:46
S117	44	join$5 with (community or group) and (wakeup or wake adj up) and (request adj2 join or request$4join or join adj2 request)	US-PGPUB; USPAT; EPO; DERWENT	OR	ON	2005/06/15 11:53
S118	6	join$5 with (community or group) and (wakeup or wake adj up) and (request adj2 join or request$4join or join adj2 request)	USPAT; EPO; DERWENT	OR	ON	2005/06/15 11:54
S119	29	(request adj2 join or request$4join or join adj2 request) and (community or group) and alive	USPAT; EPO; DERWENT	OR	ON	2005/06/15 11:58
S120	12	(request adj2 join or request$4join or join adj2 request) and (community or group) and (wakeup or wake adj up)	USPAT; EPO; DERWENT	OR	ON	2005/06/15 12:00
S121	7	(request adj2 join or request$4join or join adj2 request) and (community or group) and (wakeup or wake adj up) with message	USPAT; EPO; DERWENT	OR	ON	2005/06/15 12:00
S122	3	community and join with request and allocat$8 with peer with content and list	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	OFF	2006/05/11 13:10

S123	3	group and join with request and allocat$8 with peer with content and list	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	OFF	2006/05/11 13:10
S124	0	cluster and join with request and allocat$8 with peer with content and list	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	OFF	2006/05/11 13:11
S125	21	cluster and join with request and (shar$6 or synchoniz$8 or distribut$8 or allocat$8) with (peer or member) with content and list	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	OFF	2006/05/11 13:12
S126	15	709/203,205,215,216,248.ccls. and cach$5 with (community or cluster) and (sharing and updating) with (content or data or cache)	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	OFF	2006/05/11 14:34
S127	34	cache with cluster and (reallocat$5 or re-allocat$5) with content	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2006/10/17 14:02
S128	10	cache with cluster and (reallocat$5 or re-allocat$5) with content	USPAT	OR	ON	2006/10/17 14:03
S129	24	cache with cluster and (reallocat$5 or re-allocat$5) with content not S128	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2006/10/17 14:02
S130	5	(reallocat$5 or re-allocat$5) with cache with content	USPAT	OR	ON	2006/10/17 14:05
S131	1	(reallocat$5 or re-allocat$5) with cache with content and (plural$5 adj3 cache or cache adj3 (community or network or group))	USPAT	OR	ON	2006/10/17 14:04
S132	15	(reallocat$5 or re-allocat$5) with content and cluster	USPAT	OR	ON	2006/10/17 14:08
S133	15	(reallocat$5 or re-allocat$5) with storage with content	USPAT	OR	ON	2006/10/17 14:26
S134	11	(reallocat$5 or re-allocat$5) with storage with content and cache	USPAT	OR	ON	2006/10/17 14:23
S135	0	(reallocat$5 or re-allocat$5) with content with join$5 and cluster	USPAT	OR	ON	2006/10/17 14:24
S136	11	(reallocat$5 or re-allocat$5) with storage with content and cache	USPAT	OR	ON	2006/10/17 14:28
S137	51	((cluster with cache) or plural$5 adj3 cache) and (re-allocat$5 or reallocat$5 or re-distribut$5 or redistribut$5 or reassign$5 or re-assign$5) with content with (cache or storage)	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2006/10/17 14:32

EAST Search History

S138	12	((cluster with cache) or plural$5 adj3 cache) and (re-allocat$5 or reallocat$5 or re-distribut$5 or redistribut$5 or reassign$5 or re-assign$5) with content with (cache or storage)	USPAT	OR	ON	2006/10/17 14:37
S139	18	((cluster with cache) or plural$5 adj3 cache) and (re-allocat$5 or reallocat$5 or re-distribut$5 or redistribut$5 or reassign$5 or re-assign$5) with content	USPAT	OR	ON	2006/10/17 14:37
S140	6	((cluster with cache) or plural$5 adj3 cache) and (re-allocat$5 or reallocat$5 or re-distribut$5 or redistribut$5 or reassign$5 or re-assign$5) with content not S138	USPAT	OR	ON	2006/10/17 14:44
S141	6	((cluster with cache) or plural$5 adj3 cache or cache adj2 group or cache adj2 community) and (re-allocat$5 or reallocat$5 or re-distribut$5 or redistribut$5 or reassign$5 or re-assign$5) with content not S138	USPAT	OR	ON	2006/10/17 14:44
S142	8	cache with (join$5 or added) and (re-allocat$5 or reallocat$5 or re-distribut$5 or redistribut$5 or reassign$5 or re-assign$5) with content not S138	USPAT	OR	ON	2006/10/17 14:46
S143	58	(cache.ab. or cache.bsum.) and (re-allocat$5 or reallocat$5 or re-distribut$5 or redistribut$5 or reassign$5 or re-assign$5) with content	USPAT	OR	ON	2006/10/17 14:56
S144	562136	cache with (farm or cluster) and Join$5 and (reallocat$5 or re-allocat$5) with storage or memory	USPAT	OR	ON	2006/10/17 14:57
S145	4	cache with (farm or cluster) and Join$5 and (reallocat$5 or re-allocat$5) with (storage or memory)	USPAT	OR	ON	2006/10/17 15:00
S146	10	cache with Join$5 and (reallocat$5 or re-allocat$5) with (storage or memory)	USPAT	OR	ON	2006/10/17 15:02
S147	20	709/213,214,215,216.ccls. and (reallocat$5 or re-allocat$5) with (storage or memory)	USPAT	OR	ON	2006/10/17 15:03
S148	9	709/213,214,215,216.ccls. and cache and (reallocat$5 or re-allocat$5) with (storage or memory)	USPAT	OR	ON	2006/10/17 15:03

EAST Search History

S149	0	(re-allocating or reallocating) adj3 cache adj3 content	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2006/10/17 17:22
S150	0	(re-allocating or reallocating) adj3 cached adj3 content	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2006/10/17 17:23
S151	0	(re-allocating or reallocating) with cached with content	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2006/10/17 17:24
S152	4	(re-allocat$5 or reallocat$5) with cached with content	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2006/10/17 17:52
S153	0	(allocat$5 or allocat$5) with cached with content with join$5 and cluster	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2006/10/17 17:27
S154	0	(allocat$5 or allocat$5) with cached with content with join$5 and (cluster or group or community or farm)	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2006/10/17 17:27
S155	0	(re-allocat$5 or reallocat$5 or re-distribut$5 or redistribut$5 or reassign$5 or re-assign$5) with cached with content with join$5 and (cluster or group or community or farm)	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2006/10/17 17:27
S156	5	(re-allocat$5 or reallocat$5 or re-distribut$5 or redistribut$5 or reassign$5 or re-assign$5) with cached with content	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2006/10/17 17:52
S157	194	(allocat$5 or distribut$5 or assign$5 or re-allocat$5 or reallocat$5 or re-distribut$5 or redistribut$5 or reassign$5 or re-assign$5) with cached with content	US-PGPUB; USPAT; EPO; JPO; DERWENT	OR	ON	2006/10/17 17:53
S158	47	(allocat$5 or distribut$5 or assign$5 or re-allocat$5 or reallocat$5 or re-distribut$5 or redistribut$5 or reassign$5 or re-assign$5) with cached with content	USPAT	OR	ON	2006/10/17 17:53
S159	0	(allocat$5 or distribut$5 or assign$5 or re-allocat$5 or reallocat$5 or re-distribut$5 or redistribut$5 or reassign$5 or re-assign$5) with cached with content with join$5	USPAT	OR	ON	2006/10/17 17:54

PATENT APPLICATION FEE DETERMINATION RECORD
Effective October 1, 2004

Application or Docket Number
09/759406

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)	SMALL ENTITY TYPE □		OR	OTHER THAN SMALL ENTITY	
FOR 02/27/06	NUMBER FILED	NUMBER EXTRA	RATE	FEE		RATE	FEE
BASIC FEE					OR		790.00
TOTAL CLAIMS	38 minus /05	* 0	x$11=		OR	x$22=	
INDEPENDENT CLAIMS	6 minus /5	* 0	x41=		OR	x82=	
MULTIPLE DEPENDENT CLAIM PRESENT			+135=		OR	+270=	
			TOTAL	0	OR	TOTAL	

* If the difference in column 1 is less than zero, enter "0" in column 2

CLAIMS AS AMENDED - PART II

AMENDMENT A — 8-16-06

	(Column 1)		(Column 2)	(Column 3)	SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
Total	* 35	Minus	** 38	=	x$11=		OR	x$22=	
Independent	* 6	Minus	*** 6	=	x41=		OR	x82=	
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM					+135=		OR	+270=	
					TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT B — 10-23-06

	(Column 1)		(Column 2)	(Column 3)					
	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
Total	* 36	Minus	** 38	=	x$11=		OR	x$22=	
Independent	* 6	Minus	*** 6	=	x41=		OR	x82=	
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM					+135=		OR	+270=	
					TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C

	(Column 1)		(Column 2)	(Column 3)					
	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
Total	*	Minus	**	=	x$11=		OR	x$22=	
Independent	*	Minus	***	=	x41=		OR	x82=	
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM					+135=		OR	+270=	
					TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

FORM PTO-875 (Rev. 8/87) *U.S. Government Printing Office: 1997 - [illegible] Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

7590 10/24/2006

Baker Botts L.L.P.
2001 Rose Avenue
Dallas, TX 75201-2980

EXAMINER	
WON, MICHAEL YOUNG	
ART UNIT	PAPER NUMBER
2155	

DATE MAILED: 10/24/2006

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/759,406	01/12/2001	Keith A. Lowery	066241.0117	2308

TITLE OF INVENTION: METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	YES	$700	$300	$0	$1000	01/24/2007

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.

If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

A. If the status is the same, pay the TOTAL FEE(S) DUE shown above.

B. If the status above is to be removed, check box 5b on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and twice the amount of the ISSUE FEE shown above, or

If the SMALL ENTITY is shown as NO:

A. Pay TOTAL FEE(S) DUE shown above, or

B. If applicant claimed SMALL ENTITY status before, or is now claiming SMALL ENTITY status, check box 5a on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and 1/2 the ISSUE FEE shown above.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail
Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

7590 10/24/2006

Baker Botts L.L.P.
2001 Rose Avenue
Dallas, TX 75201-2980

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)

(Signature)

(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/759,406	01/12/2001	Keith A. Lowery	066241.0117	2308

TITLE OF INVENTION: METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	YES	$700	$300	$0	$1000	01/24/2007

EXAMINER	ART UNIT	CLASS-SUBCLASS
WON, MICHAEL YOUNG	2155	709-214000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

□ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.

□ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list

(1) the names of up to 3 registered patent attorneys or agents OR, alternatively,

(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 ____________________

2 ____________________

3 ____________________

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE

(B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : □ Individual □ Corporation or other private group entity □ Government

4a. The following fee(s) are submitted:

□ Issue Fee

□ Publication Fee (No small entity discount permitted)

□ Advance Order - # of Copies ____________

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**

□ A check is enclosed.

□ Payment by credit card. Form PTO-2038 is attached.

□ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number ________ (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)

□ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27.

□ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature ____________________ Date ____________________

Typed or printed name ____________________ Registration No. ____________________

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/759,406	01/12/2001	Keith A. Lowery	066241.0117	2308

7590 10/24/2006

Baker Botts L.L.P.
2001 Rose Avenue
Dallas, TX 75201-2980

EXAMINER	
WON, MICHAEL YOUNG	
ART UNIT	PAPER NUMBER
2155	

DATE MAILED: 10/24/2006

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)

(application filed on or after May 29, 2000)

The Patent Term Adjustment to date is 875 day(s). If the issue fee is paid on the date that is three months after the mailing date of this notice and the patent issues on the Tuesday before the date that is 28 weeks (six and a half months) after the mailing date of this notice, the Patent Term Adjustment will be 875 day(s).

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

Notice of Allowability	**Application No.** 09/759,406	**Applicant(s)** LOWERY ET AL.
	Examiner Michael Y. Won	**Art Unit** 2155

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--
All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *the amendment filed August 16, 2006*.

2. ☒ The allowed claim(s) is/are *1-11,13-26,28-30,95-101 and 105 (renumbered 1-36)*.

3. □ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) □ All b) □ Some* c) □ None of the:
   1. □ Certified copies of the priority documents have been received.
   2. □ Certified copies of the priority documents have been received in Application No. _____ .
   3. □ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. □ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

5. □ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   (a) □ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
   1) □ hereto or 2) □ to Paper No./Mail Date _____.
   (b) □ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6. □ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)
1. ☒ Notice of References Cited (PTO-892)
2. □ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. □ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date _____
4. □ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. □ Notice of Informal Patent Application
6. □ Interview Summary (PTO-413), Paper No./Mail Date _____ .
7. □ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. □ Other _____.

SALEH NAJJAR
SUPERVISORY PATENT EXAMINER

DETAILED ACTION

1. This action is in response to the amendment filed August 16, 2006 after the Non-Final office action filed on May 16, 2006.

2. Claims 1, 11, 13-16, 26, 28-30, 95, 98, 101, and 105 have been amended and claims 12 and 27 have been cancelled.

Allowable Subject Matter

3. Claims 1-11, 13-26, 28-30, 95-101, and 105 are allowable over prior art of record and in light of applicants' amendments and arguments presented in amendment filed August 16, 2006.

4. The following is an examiner's statement of reasons for allowance:

The prior art of record does not disclose, teach, or suggest neither singly nor in combination the claimed limitation of "re-allocating the cache storage of the content among the peers in the cache community, in response to allowing the client to join the community" as recited in claims 1, 16, 101 and similarly recited in claims 95, 98 and 105.

5. Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably

accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael Y. Won whose telephone number is 571-272-3993. The examiner can normally be reached on M-Th: 7AM-5PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Saleh Najjar can be reached on 571-272-4006. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Michael Won
October 18, 2006

SALEH NAJJAR
SUPERVISORY PATENT EXAMINER

Notice of References Cited	Application/Control No. 09/759,406	Applicant(s)/Patent Under Reexamination LOWERY ET AL.	
	Examiner Michael Y. Won	Art Unit 2155	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-4,603,382	07-1986	Cole et al.	710/56
*	B	US-5,522,045	05-1996	Sandberg, Jonathan	709/215
*	C	US-5,537,572	07-1996	Michelsen et al.	711/135
*	D	US-6,167,490	12-2000	Levy et al.	711/148
*	E	US-6,199,179	03-2001	Kauffman et al.	714/26
*	F	US-6,263,302	07-2001	Hellestrand et al.	703/17
*	G	US-6,330,605	12-2001	Christensen et al.	709/226
*	H	US-2002/0026560	02-2002	JORDAN et al.	711/120
*	I	US-2002/0103972	08-2002	Satran et al.	711/119
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Issue Classification	Application/Control No. 09/759,406	Applicant(s)/Patent under Reexamination LOWERY ET AL.
	Examiner Michael Y. Won	Art Unit 2155

ISSUE CLASSIFICATION

ORIGINAL	
CLASS	SUBCLASS
709	214

INTERNATIONAL CLASSIFICATION									
CLAIMED					NON-CLAIMED				
G	06	F	15	/167					/
				/					/
				/					/
				/					/
				/					/
				/					/
				/					/

CROSS REFERENCES

CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)			
709	203	217	219	226
707	10	200		
710	56			
711	119	126	130	135

10/18/06
(Assistant Examiner) (Date)

Laurina Johnson 10-23-06
(Legal Instruments Examiner) (Date)

SALEH NAJJAR
SUPERVISORY PATENT EXAMINER

10-23-06
(Primary Examiner) (Date)

Total Claims Allowed: 36

O.G. Print Claim(s)	O.G. Print Fig.
1	10

□ **Claims renumbered in the same order as presented by applicant** □ CPA □ T.D. □ R.1.47

Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
1	1		31		61		91		121		151		181
2	2		32		62		92		122		152		182
3	3		33		63		93		123		153		183
4	4		34		64		94		124		154		184
5	5		35		65	29	95		125		155		185
6	6		36		66	30	96		126		156		186
7	7		37		67	31	97		127		157		187
8	8		38		68	32	98		128		158		188
9	9		39		69	33	99		129		159		189
10	10		40		70	34	100		130		160		190
11	11		41		71	35	101		131		161		191
	12		42		72		102		132		162		192
12	13		43		73		103		133		163		193
13	14		44		74		104		134		164		194
14	15		45		75	36	105		135		165		195
15	16		46		76		106		136		166		196
16	17		47		77		107		137		167		197
17	18		48		78		108		138		168		198
18	19		49		79		109		139		169		199
19	20		50		80		110		140		170		200
20	21		51		81		111		141		171		201
21	22		52		82		112		142		172		202
22	23		53		83		113		143		173		203
23	24		54		84		114		144		174		204
24	25		55		85		115		145		175		205
25	26		56		86		116		146		176		206
	27		57		87		117		147		177		207
26	28		58		88		118		148		178		208
27	29		59		89		119		149		179		209
28	30		60		90		120		150		180		210

Search Notes	**Application/Control No.** 09/759,406	**Applicant(s)/Patent under Reexamination** LOWERY ET AL.	
	Examiner Michael Y. Won	**Art Unit** 2155	

SEARCHED

Class	Subclass	Date	Examiner
709	213, 217	7/21/2004	MW
709	201, 212, 217	2/25/2005	MW
709	212	6/15/2005	MW
707	8, 10	6/15/2005	MW
711	151	6/15/2005	MW
709	212	11/15/2005	MW
709	203, 205, 215, 216, 248	5/11/2006	MW

INTERFERENCE SEARCHED

Class	Subclass	Date	Examiner
709	**203, 214**	10/18/2006	**MW**
711	119, 126, 130, 135	10/18/06	MW

SEARCH NOTES
(INCLUDING SEARCH STRATEGY)

	DATE	EXMR
EAST: USPAT; USPG-PUB; DERWENT; EPO; JPO	7/21/2006	MW
NPL SEARCH: IEEE	7/21/2006	MW
EAST: USPAT; USPG-PUB; DERWENT; EPO; JPO	2/25/2005	MW
EAST: USPAT; USPG-PUB; DERWENT; EPO; JPO	6/15/2005	MW
EAST: USPAT; USPG-PUB; DERWENT; EPO; JPO	11/15/2005	MW
EAST: USPAT; USPG-PUB; DERWENT; EPO; JPO	5/11/2006	MW
EAST: USPAT; USPG-PUB; DERWENT; EPO; JPO	10/17/2006	**MW**
EAST: USPAT; USPG-PUB; DERWENT; EPO; JPO	10/18/2006	MW

Index of Claims	Application/Control No. 09/759,406	Applicant(s)/Patent under Reexamination LOWERY ET AL.	
	Examiner Michael Y. Won	Art Unit 2155	

√	Rejected	–	(Through numeral) Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

Claim		Date								
Final	Original	10/18/06								
1	1	=								
2	2	↑								
3	3									
4	4									
5	5									
6	6									
7	7									
8	8									
9	9									
10	10	↓								
11	11	=								
	~~12~~									
12	13	=								
13	14	↑								
14	15									
15	16									
16	17									
17	18									
18	19									
19	20									
20	21									
21	22									
22	23									
23	24									
24	25	↓								
25	26	=								
	~~27~~									
16	28	=								
27	29	=								
28	30	=								
	~~31~~	↑								
	32									
	33									
	34									
	35									
	36									
	37									
	38									
	39									
	40									
	41									
	42									
	43									
	44									
	45									
	46									
	47									
	48									
	49									
	~~50~~	↓								

Claim		Date								
Final	Original	10/18/06								
	~~51~~	↑								
	52									
	53									
	54									
	55									
	56									
	57									
	58									
	59									
	60									
	61									
	62									
	63									
	64									
	65									
	66									
	67									
	68									
	69									
	70									
	71									
	72									
	73									
	74									
	75									
	76									
	77									
	78									
	79									
	80									
	81									
	82									
	83									
	84									
	85									
	86									
	87									
	88									
	89									
	90									
	91									
	92									
	93									
	~~94~~	↓								
29	95	=								
30	96	↑								
31	97									
32	98									
33	99	↓								
34	100	=								

Claim		Date								
Final	Original	10/18/06								
35	101	=								
	~~102~~									
	~~103~~									
	~~104~~									
36	105	=								
	106									
	107									
	108									
	109									
	110									
	111									
	112									
	113									
	114									
	115									
	116									
	117									
	118									
	119									
	120									
	121									
	122									
	123									
	124									
	125									
	126									
	127									
	128									
	129									
	130									
	131									
	132									
	133									
	134									
	135									
	136									
	137									
	138									
	139									
	140									
	141									
	142									
	143									
	144									
	145									
	146									
	147									
	148									
	149									
	150									

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

7590 10/24/2006

Baker Botts L.L.P.
2001 Rose Avenue
Dallas, TX 75201-2980

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)

(Signature)

(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/759,406	01/12/2001	Keith A. Lowery	066241.0117	2308

TITLE OF INVENTION: METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	YES	$700	$300	$0	$1000	01/24/2007

EXAMINER	ART UNIT	CLASS-SUBCLASS
WON, MICHAEL YOUNG	2155	709-214000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

□ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.

□ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list

(1) the names of up to 3 registered patent attorneys or agents OR, alternatively,

(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 Baker Botts L.L.P.
2 ________
3 ________

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE: EpicRealm Licensing LLC

(B) RESIDENCE: (CITY and STATE OR COUNTRY): Dallas, Texas

Please check the appropriate assignee category or categories (will not be printed on the patent) : □ Individual ☒ Corporation or other private group entity □ Government

4a. The following fee(s) are submitted:

☒ Issue Fee
☒ Publication Fee (No small entity discount permitted)
□ Advance Order - # of Copies ________

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**

□ A check is enclosed.
□ Payment by credit card. Form PTO-2038 is attached.
☒ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number 02-0384 (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)

□ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27. □ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature ________ Date 23 Jan 07

Typed or printed name Charles S. Fish Registration No. 35,870

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Keith A. Lowery et al
Serial No.:	09/759,406
Filed:	January 12, 2001
Group No.:	2155
Examiner:	Michael Young Won
Notice of Allowance Mailed:	October 24, 2006
Confirmation No.:	2308
Title:	**METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING**

Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

RESPONSE TO STATEMENT OF REASONS FOR ALLOWANCE

Pursuant to 37 C.F.R. § 1.104, Applicants respectfully issue a statement commenting on the Examiner's reasons for allowance. Applicants respectfully disagree with the Examiner's reasons for allowance to the extent that they are inconsistent with applicable case law, statutes, and regulations.

Furthermore, Applicants do not admit to any characterization or limitation of the claims or to any characterization of a reference by the Examiner, particularly any that are inconsistent with the language of the claims considered in their entirety and including all of their constituent limitations.

Respectfully submitted,
BAKER BOTTS L.L.P.
Attorneys for Applicants

Charles S. Fish
Registration No. 35,870

Date: 23 Jan 07
2001 Ross Avenue, Suite 600
Dallas, TX 75201-2980
(214) 953-6507

Customer No.: **05073**

Electronic Patent Application Fee Transmittal

Application Number:	09759406
Filing Date:	12-Jan-2001
Title of Invention:	METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING
First Named Inventor/Applicant Name:	Keith A. Lowery
Filer:	Barton E. Showalter/Anthony T. Smith
Attorney Docket Number:	066241.0117
Filed as Small Entity	

Utility Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Utility Appl issue fee	2501	1	700	700
Publ. Fee- early, voluntary, or normal	1504	1	300	300

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Extension-of-Time:				
Miscellaneous:				
	Total in USD ($)			**1000**

Electronic Acknowledgement Receipt

EFS ID:	1457651
Application Number:	09759406
International Application Number:	
Confirmation Number:	2308
Title of Invention:	METHOD AND SYSTEM FOR DYNAMIC DISTRIBUTED DATA CACHING
First Named Inventor/Applicant Name:	Keith A. Lowery
Correspondence Address:	Baker Botts L.L.P. - 2001 Rose Avenue - Dallas TX 75201-2980 US 2149536680 -
Filer:	Barton E. Showalter/Anthony T. Smith
Filer Authorized By:	Barton E. Showalter
Attorney Docket Number:	066241.0117
Receipt Date:	23-JAN-2007
Filing Date:	12-JAN-2001
Time Stamp:	11:25:51
Application Type:	Utility

Payment information:

Submitted with Payment	yes
Payment was successfully received in RAM	$1000
RAM confirmation Number	1530

Deposit Account	020384
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows: Charge any Additional Fees required under 37 C.F.R. Section 1.16 and 1.17	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)	Multi Part /.zip	Pages (if appl.)
1	Issue Fee Payment (PTO-85B)	0662410117IssueFee.pdf	93053	no	1
Warnings:					
Information:					
2	Post Allowance Communication - Incoming	0662410117RespReasAllow.pdf	33744	no	1
Warnings:					
Information:					
3	Fee Worksheet (PTO-06)	fee-info.pdf	8311	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	135108		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	ISSUE DATE	PATENT NO.	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/759,406	03/06/2007	7188145	066241.0117	2308

7590 02/14/2007

Baker Botts L.L.P.
2001 Rose Avenue
Dallas, TX 75201-2980

ISSUE NOTIFICATION

The projected patent number and issue date are specified above.

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment is 875 day(s). Any patent to issue from the above-identified application will include an indication of the adjustment on the front page.

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at (571)-272-4200.

APPLICANT(s) (Please see PAIR WEB site http://pair.uspto.gov for additional applicants):

Keith A. Lowery, Richardson, TX;
Bryan S. Chin, Plano, TX;
David A. Consolver, Arlington, TX;
Gregg A. DeMasters, Plano, TX;

IR103 (Rev. 11/05)

AO 120 (Rev. 08/10)

TO: **Mail Stop 8 Director of the U.S. Patent and Trademark Office P.O. Box 1450 Alexandria, VA 22313-1450**	**REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court for the District of Delaware on the following

□ Trademarks or ☑ Patents. (□ the patent action involves 35 U.S.C. § 292.):

DOCKET NO.	DATE FILED 2/1/2013	U.S. DISTRICT COURT for the District of Delaware
PLAINTIFF PARALLEL NETWORKS, LLC		DEFENDANT TURBINE, INC.
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 US 7,188,145 B2	3/6/2007	Parallel Networks, LLC
2 US 7,730, 262 B2	6/1/2010	Parallel Networks, LLC
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY □ Amendment □ Answer □ Cross Bill □ Other Pleading	
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

AO 120 (Rev. 08/10)

TO: Mail Stop 8 Director of the U.S. Patent and Trademark Office P.O. Box 1450 Alexandria, VA 22313-1450	**REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court for the District of Delaware on the following

□ Trademarks or ☑ Patents. (□ the patent action involves 35 U.S.C. § 292.):

DOCKET NO.	DATE FILED	U.S. DISTRICT COURT
	2/1/2013	for the District of Delaware

PLAINTIFF	DEFENDANT
PARALLEL NETWORKS, LLC	WIZARDS OF THE COAST LLC

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 US 7,188,145 B2	3/6/2007	Parallel Networks, LLC
2 US 7,730, 262 B2	6/1/2010	Parallel Networks, LLC
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY □ Amendment □ Answer □ Cross Bill □ Other Pleading

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

AO 120 (Rev. 08/10)

TO: Mail Stop 8 Director of the U.S. Patent and Trademark Office P.O. Box 1450 Alexandria, VA 22313-1450	**REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court for the District of Delaware on the following

□ Trademarks or ☑ Patents. (□ the patent action involves 35 U.S.C. § 292.):

DOCKET NO.	DATE FILED 2/1/2013	U.S. DISTRICT COURT for the District of Delaware
PLAINTIFF PARALLEL NETWORKS, LLC		DEFENDANT RIOT GAMES, INC.

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 US 7,188,145 B2	3/6/2007	Parallel Networks, LLC
2 US 7,730, 262 B2	6/1/2010	Parallel Networks, LLC
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY □ Amendment □ Answer □ Cross Bill □ Other Pleading

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

AO 120 (Rev. 08/10)

TO: **Mail Stop 8** **Director of the U.S. Patent and Trademark Office** **P.O. Box 1450** **Alexandria, VA 22313-1450**	**REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court for the District of Delaware on the following

□ Trademarks or ☑ Patents. (□ the patent action involves 35 U.S.C. § 292.):

DOCKET NO.	DATE FILED 2/1/2013	U.S. DISTRICT COURT for the District of Delaware
PLAINTIFF PARALLEL NETWORKS, LLC		DEFENDANT PERFECT WORLD ENTERTAINMENT INC.

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 US 7,188,145 B2	3/6/2007	Parallel Networks, LLC
2 US 7,730, 262 B2	6/1/2010	Parallel Networks, LLC
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY □ Amendment □ Answer □ Cross Bill □ Other Pleading	
PATENT OR TRADEMARK NO.	**DATE OF PATENT OR TRADEMARK**	**HOLDER OF PATENT OR TRADEMARK**
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

AO 120 (Rev. 08/10)

TO: **Mail Stop 8** **Director of the U.S. Patent and Trademark Office** **P.O. Box 1450** **Alexandria, VA 22313-1450**	**REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court for the District of Delaware on the following

□ Trademarks or ☑ Patents. (□ the patent action involves 35 U.S.C. § 292.):

DOCKET NO.	DATE FILED 2/1/2013	U.S. DISTRICT COURT for the District of Delaware
PLAINTIFF PARALLEL NETWORKS, LLC		DEFENDANT OUTSPARK, INC.

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 US 7,188,145 B2	3/6/2007	Parallel Networks, LLC
2 US 7,730, 262 B2	6/1/2010	Parallel Networks, LLC
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY □ Amendment □ Answer □ Cross Bill □ Other Pleading

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

AO 120 (Rev. 08/10)

TO: **Mail Stop 8** **Director of the U.S. Patent and Trademark Office** **P.O. Box 1450** **Alexandria, VA 22313-1450**	**REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court for the District of Delaware on the following

□ Trademarks or ☑ Patents. (□ the patent action involves 35 U.S.C. § 292.):

DOCKET NO.	DATE FILED 2/1/2013	U.S. DISTRICT COURT for the District of Delaware
PLAINTIFF PARALLEL NETWORKS, LLC		DEFENDANT ONNET USA, INC.

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 US 7,188,145 B2	3/6/2007	Parallel Networks, LLC
2 US 7,730, 262 B2	6/1/2010	Parallel Networks, LLC
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY □ Amendment □ Answer □ Cross Bill □ Other Pleading	
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

AO 120 (Rev. 08/10)

TO: **Mail Stop 8** **Director of the U.S. Patent and Trademark Office** **P.O. Box 1450** **Alexandria, VA 22313-1450**	**REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court for the District of Delaware on the following

□ Trademarks or ☑ Patents. (□ the patent action involves 35 U.S.C. § 292.):

DOCKET NO.	DATE FILED 2/1/2013	U.S. DISTRICT COURT for the District of Delaware
PLAINTIFF PARALLEL NETWORKS, LLC		DEFENDANT FUNCOM INC.

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 US 7,188,145 B2	3/6/2007	Parallel Networks, LLC
2 US 7,730, 262 B2	6/1/2010	Parallel Networks, LLC
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY □ Amendment □ Answer □ Cross Bill □ Other Pleading	
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

AO 120 (Rev. 08/10)

TO: **Mail Stop 8** **Director of the U.S. Patent and Trademark Office** **P.O. Box 1450** **Alexandria, VA 22313-1450**	**REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court for the District of Delaware on the following

□ Trademarks or ☑ Patents. (□ the patent action involves 35 U.S.C. § 292.):

DOCKET NO.	DATE FILED 2/1/2013	U.S. DISTRICT COURT for the District of Delaware
PLAINTIFF PARALLEL NETWORKS, LLC		DEFENDANT NEXON AMERICA INC.

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 US 7,188,145 B2	3/6/2007	Parallel Networks, LLC
2 US 7,730, 262 B2	6/1/2010	Parallel Networks, LLC
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY □ Amendment □ Answer □ Cross Bill □ Other Pleading

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

AO 120 (Rev. 08/10)

TO: **Mail Stop 8 Director of the U.S. Patent and Trademark Office P.O. Box 1450 Alexandria, VA 22313-1450**	**REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court for the District of Delaware on the following

□ Trademarks or ☑ Patents. (□ the patent action involves 35 U.S.C. § 292.):

DOCKET NO.	DATE FILED 2/1/2013	U.S. DISTRICT COURT for the District of Delaware
PLAINTIFF PARALLEL NETWORKS, LLC		DEFENDANT KOG GAMES, INC.

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 US 7,188,145 B2	3/6/2007	Parallel Networks, LLC
2 US 7,730, 262 B2	6/1/2010	Parallel Networks, LLC
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY □ Amendment □ Answer □ Cross Bill □ Other Pleading	
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

AO 120 (Rev. 08/10)

TO: **Mail Stop 8** **Director of the U.S. Patent and Trademark Office** **P.O. Box 1450** **Alexandria, VA 22313-1450**	**REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court for the District of Delaware on the following

□ Trademarks or ☑ Patents. (□ the patent action involves 35 U.S.C. § 292.):

DOCKET NO.	DATE FILED	U.S. DISTRICT COURT
	2/1/2013	for the District of Delaware

PLAINTIFF	DEFENDANT
PARALLEL NETWORKS, LLC	ELECTRONIC ARTS, INC.

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 US 7,188,145 B2	3/6/2007	Parallel Networks, LLC
2 US 7,730, 262 B2	6/1/2010	Parallel Networks, LLC
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY
	□ Amendment □ Answer □ Cross Bill □ Other Pleading

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

AO 120 (Rev. 3/04)

TO: **Mail Stop 8** **Director of the U.S. Patent and Trademark Office** **P.O. Box 1450** **Alexandria, VA 22313-1450**	**REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court Delaware on the following Patents or Trademarks:

DOCKET NO. 13cv185-RGA	DATE FILED 2/1/2013	U.S. DISTRICT COURT
PLAINTIFF Parallel Networks LLC		DEFENDANT Wizards of the Coast LLC
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 US 7,188,145 B2	3/6/2007	Parallel Networks LLC
2 US 7,730,262 B2	6/1/2010	Parallel Networks LLC
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY □ Amendment Answer Cross Bill Other Pleading	
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

See attached Notice of Dismissal

CLERK JOHN A. CERINO, CLERK OF COURT	(BY) DEPUTY CLERK	DATE 4/11/2013

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

AO 120 (Rev. 3/04)

TO: **Mail Stop 8** **Director of the U.S. Patent and Trademark Office** **P.O. Box 1450** **Alexandria, VA 22313-1450**	**REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court Delaware on the following Patents or Trademarks:

DOCKET NO. 13cv176-RGA	DATE FILED 2/1/2013	U.S. DISTRICT COURT	
PLAINTIFF Parallel Networks LLC		DEFENDANT En Masse Entertainment Inc.	

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 US 7,188,145 B2	3/6/2007	Parallel Networks LLC
2 US 7,730,262 B2	6/1/2010	Parallel Networks LLC
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY □ Amendment Answer Cross Bill Other Pleading	
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

See attached Notice of Dismissal

CLERK	(BY) DEPUTY CLERK	DATE
JOHN A. CERINO, CLERK OF COURT		4/11/2013

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

✎ AO 120 (Rev. 3/04)

TO: **Mail Stop 8** **Director of the U.S. Patent and Trademark Office** **P.O. Box 1450** **Alexandria, VA 22313-1450**	**REPORT ON THE** **FILING OR DETERMINATION OF AN** **ACTION REGARDING A PATENT OR** **TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court Delaware on the following Patents or Trademarks:

DOCKET NO.	DATE FILED	U.S. DISTRICT COURT
13cv175-RGA	2/1/2013	

PLAINTIFF	DEFENDANT
Parallel Networks LLC	Electronic Arts Inc.

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 US 7,188,145 B2	3/6/2007	Parallel Networks LLC
2 US 7,730,262 B2	6/1/2010	Parallel Networks LLC
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY □ Amendment Answer Cross Bill Other Pleading	
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

See attached Notice of Dismissal

CLERK	(BY) DEPUTY CLERK	DATE
JOHN A. CERINO, CLERK OF COURT		4/11/2013

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

AO 120 (Rev. 3/04)

TO: **Mail Stop 8** **Director of the U.S. Patent and Trademark Office** **P.O. Box 1450** **Alexandria, VA 22313-1450**	**REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court Delaware on the following X Patents or Trademarks:

DOCKET NO. 13-177-RGA	DATE FILED 2/1/2013	U.S. DISTRICT COURT DISTRICT OF DELAWARE
PLAINTIFF Parallel Networks LLC		DEFENDANT Funcom Inc.

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 US 7,188,145 B2	3/6/2007	Parallel Networks LLC
2 US 7,730,262 B2	6/1/2010	Parallel Networks LLC
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY □ Amendment □ Answer □ Cross Bill □ Other Pleading	
PATENT OR TRADEMARK NO.	**DATE OF PATENT OR TRADEMARK**	**HOLDER OF PATENT OR TRADEMARK**
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

See attached Order of Dismissal

CLERK	(BY) DEPUTY CLERK	DATE
JOHN A. CERINO, CLERK OF COURT		4/9/2013

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

AO 120 (Rev. 08/10)

TO: **Mail Stop 8** **Director of the U.S. Patent and Trademark Office** **P.O. Box 1450** **Alexandria, VA 22313-1450**	**REPORT ON THE** **FILING OR DETERMINATION OF AN** **ACTION REGARDING A PATENT OR** **TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court for the District of Delaware on the following

□ Trademarks or ☑ Patents. (□ the patent action involves 35 U.S.C. § 292.):

DOCKET NO.	DATE FILED 5/21/2013	U.S. DISTRICT COURT for the District of Delaware

PLAINTIFF PARALLEL NETWORKS, LLC	DEFENDANT KONTIKI, INC.

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 US 7,188,145 B2	3/6/2007	Parallel Networks, LLC
2 US 7,730,262 B2	6/1/2010	Parallel Networks, LLC
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY □ Amendment □ Answer □ Cross Bill □ Other Pleading

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

AO 120 (Rev. 08/10)

TO: **Mail Stop 8** **Director of the U.S. Patent and Trademark Office** **P.O. Box 1450** **Alexandria, VA 22313-1450**	**REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court for the District of Delaware on the following

□ Trademarks or ☑ Patents. (□ the patent action involves 35 U.S.C. § 292.):

DOCKET NO.	DATE FILED 5/9/2013	U.S. DISTRICT COURT for the District of Delaware
PLAINTIFF PARALLEL NETWORKS, LLC		DEFENDANT RELOADED GAMES, INC.

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 US 7,188,145 B2	3/6/2007	Parallel Networks, LLC
2 US 7,730,262 B2	6/1/2010	Parallel Networks, LLC
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY □ Amendment □ Answer □ Cross Bill □ Other Pleading

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

AO 120 (Rev. 08/10)

TO: **Mail Stop 8** **Director of the U.S. Patent and Trademark Office** **P.O. Box 1450** **Alexandria, VA 22313-1450**	**REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court for the District of Delaware on the following

□ Trademarks or ☑ Patents. (□ the patent action involves 35 U.S.C. § 292.):

DOCKET NO.	DATE FILED 5/9/2013	U.S. DISTRICT COURT for the District of Delaware
PLAINTIFF PARALLEL NETWORKS, LLC		DEFENDANT SG INTERACTIVE, INC.

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 US 7,188,145 B2	3/6/2007	Parallel Networks, LLC
2 US 7,730,262 B2	6/1/2010	Parallel Networks, LLC
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY □ Amendment □ Answer □ Cross Bill □ Other Pleading	
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

AO 120 (Rev. 08/10)

TO: Mail Stop 8 Director of the U.S. Patent and Trademark Office P.O. Box 1450 Alexandria, VA 22313-1450	**REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court for the District of Delaware on the following

□ Trademarks or ☑ Patents. (□ the patent action involves 35 U.S.C. § 292.):

DOCKET NO.	DATE FILED 5/9/2013	U.S. DISTRICT COURT for the District of Delaware
PLAINTIFF PARALLEL NETWORKS, LLC		DEFENDANT BLIZZARD ENTERTAINMENT, INC.

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 US 7,188,145 B2	3/6/2007	Parallel Networks, LLC
2 US 7,730,262 B2	6/1/2010	Parallel Networks, LLC
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY □ Amendment □ Answer □ Cross Bill □ Other Pleading	
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

AO 120 (Rev. 08/10)

TO: **Mail Stop 8** **Director of the U.S. Patent and Trademark Office** **P.O. Box 1450** **Alexandria, VA 22313-1450**	**REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court for the District of Delaware on the following

□ Trademarks or ☑ Patents. (□ the patent action involves 35 U.S.C. § 292.):

DOCKET NO.	DATE FILED 5/6/2013	U.S. DISTRICT COURT for the District of Delaware

PLAINTIFF PARALLEL NETWORKS, LLC	DEFENDANT SPOTIFY USA INC.

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 US 7,188,145 B2	3/6/2007	Parallel Networks, LLC
2 US 7,730,262 B2	6/1/2010	Parallel Networks, LLC
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY □ Amendment □ Answer □ Cross Bill □ Other Pleading	
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

AO 120 (Rev. 08/10)

TO: **Mail Stop 8** **Director of the U.S. Patent and Trademark Office** **P.O. Box 1450** **Alexandria, VA 22313-1450**	**REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court for the District of Delaware on the following

□ Trademarks or ☑ Patents. (□ the patent action involves 35 U.S.C. § 292.):

DOCKET NO.	DATE FILED	U.S. DISTRICT COURT
	7/10/2013	for the District of Delaware

PLAINTIFF	DEFENDANT
PARALLEL NETWORKS, LLC	NCSOFT WEST CORPORATION

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 US 7,188,145 B2	3/6/2007	Parallel Networks, LLC
2 US 7,730,262 B2	6/1/2010	Parallel Networks, LLC
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY □ Amendment □ Answer □ Cross Bill □ Other Pleading	
PATENT OR TRADEMARK NO.	**DATE OF PATENT OR TRADEMARK**	**HOLDER OF PATENT OR TRADEMARK**
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy