# EXHIBIT 12

US006928442B2

(12) **United States Patent**
**Farber et al.**

(10) **Patent No.: US 6,928,442 B2**
(45) **Date of Patent: Aug. 9, 2005**

(54) **ENFORCEMENT AND POLICING OF LICENSED CONTENT USING CONTENT-BASED IDENTIFIERS**

(75) Inventors: **David A. Farber**, Ojai, CA (US); **Ronald D. Lachman**, Northbrook, IL (US)

(73) Assignees: **Kinetech, Inc.**, Northbrook, IL (US); **Savvis, Inc.**, Town & Country, MO (US)

( * ) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/987,723**

(22) Filed: **Nov. 15, 2001**

(65) **Prior Publication Data**

US 2002/0052884 A1 May 2, 2002

**Related U.S. Application Data**

(63) Continuation of application No. 09/283,160, filed on Apr. 1, 1999, now Pat. No. 6,415,280, which is a division of application No. 08/960,079, filed on Oct. 24, 1997, now Pat. No. 5,978,791, which is a continuation of application No. 08/425,160, filed on Apr. 11, 1995, now abandoned.

(51) **Int. Cl.**[7] ..... **G06F 17/30**

(52) **U.S. Cl.** ..... **707/10**; 707/3; 707/101; 707/200; 709/203; 709/219; 709/229

(58) **Field of Search** ..... 707/3, 6, 9, 10, 707/101, 200; 709/203, 219, 229

(56) **References Cited**

U.S. PATENT DOCUMENTS

| | | | |
|---|---|---|---|
| 3,668,647 | A | 6/1972 | Evangelisti |
| 4,215,402 | A | 7/1980 | Mitchell |
| 4,290,105 | A | 9/1981 | Cichelli |
| 4,376,299 | A | 3/1983 | Rivest |
| 4,405,829 | A | 9/1983 | Rivest |
| 4,412,285 | A | 10/1983 | Neches |

(Continued)

FOREIGN PATENT DOCUMENTS

| | | |
|---|---|---|
| EP | 0592045 | 4/1994 |

OTHER PUBLICATIONS

Gwertzman, James, et al. "The Case for Geographical Push–Caching." Technical Report HU TR 34–94 (excerpt), Harvard University, DAS, Cambridge, MA 02138, 1994, 2 pgs.

Grigni, Michelangelo, et al. "Tight Bounds on Minimum Broadcasts Networks." SIAM Journal of Discrete Mathematics, vol. 4, No. 2, May 1991, pp. 207–222.

Devine, Robert. "Design and Implementation of DDH: A Distributed Dynamic Hashing Algorithm." In Proceedings of 4th International Conference on Foundations of Data Organizations and Algorithms, 1993, pp. 101–114.

Deering, Stephen, et al. "Multicast Routing in Datagram Internetworks and Extended LANs." ACM Transactions on Computer Systems, vol. 8, No. 2, May 1990, pp. 85–110.

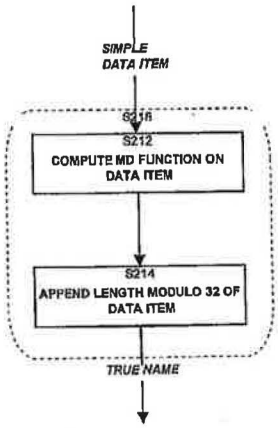
(Continued)

*Primary Examiner*—Luke S Wossum
*Assistant Examiner*—Khanh Pham
(74) *Attorney, Agent, or Firm*—Davidson Berquist Jackson & Gowdey, LLP

(57) **ABSTRACT**

Data files are distributed across a plurality of computers. The computers may form a network such as a content delivery network (CDN) or a peer-to-peer network. The network may operate as a TCP/IP network such as the Internet. Data files may represent may represent digital messages, images, videos or audio signals. For content—data items or files in the system—a name is obtained (or determined), where the name is based, at least in part, on a given function of the data in a data item or file. The given function may be a message digest or hash function, and it may be MD4, MD5, and SHA. A cony of a requested file is only provided to licensed (or authorized) parties. The system may check one or more computers for unauthorized or unlicensed content. Content is served based on a measure of availability of servers.

**56 Claims, 31 Drawing Sheets**



# ENFORCEMENT AND POLICING OF LICENSED CONTENT USING CONTENT-BASED IDENTIFIERS

This is a continuation of application Ser. No. 09/283,160, filed Apr. 1, 1999, now U.S. Pat. No. 6,415,280, which is a division of application Ser. No. 08/960,079, filed Oct. 24, 1997, now U.S. Pat. No. 5,978,791 filed Oct. 24, 2001 which is a continuation of Ser. No. 08/425,160, filed Apr. 11, 1995, now abandoned.

## BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates to data processing systems and, more particularly, to data processing systems wherein data items are identified by substantially unique identifiers which depend on all of the data in the data items and only on the data in the data items.

2. Background of the Invention

Data processing (DP) systems, computers, networks of computers, or the like, typically offer users and programs various ways to identify the data in the systems.

Users typically identify data in the data processing system by giving the data some form of name. For example, a typical operating system (OS) on a computer provides a file system in which data items are named by alphanumeric identifiers. Programs typically identify data in the data processing system using a location or address. For example, a program may identify a record in a file or database by using a record number which serves to locate that record.

In all but the most primitive operating systems, users and programs are able to create and use collections of named data items, these collections themselves being named by identifiers. These named collections can then, themselves, be made part of other named collections. For example, an OS may provide mechanisms to group files (data items) into directories (collections). These directories can then, themselves be made part of other directories. A data item may thus be identified relative to these nested directories using a sequence of names, or a so-called pathname, which defines a path through the directories to a particular data item (file or directory).

As another example, a database management system may group data records (data items) into tables and then group these tables into database files (collections). The complete address of any data record can then be specified using the database file name, the table name, and the record number of that data record.

Other examples of identifying data items include: identifying files in a network file system, identifying objects in an object-oriented database, identifying images in an image database, and identifying articles in a text database.

In general, the terms "data" and "data item" as used herein refer to sequences of bits. Thus a data item may be the contents of a file, a portion of a file, a page in memory, an object in an object-oriented program, a digital message, a digital scanned image, a part of a video or audio signal, or any other entity which can be represented by a sequence of bits. The term "data processing" herein refers to the processing of data items, and is sometimes dependent on the type of data item being processed. For example, a data processor for a digital image may differ from a data processor for an audio signal.

In all of the prior data processing systems the names or identifiers provided to identify data items (the data items



being files, directories, records in the database, objects in object-oriented programming, locations in memory or on a physical device, or the like) are always defined relative to a specific context. For instance, the file identified by a particular file name can only be determined when the directory containing the file (the context) is known. The file identified by a pathname can be determined only when the file system (context) is known. Similarly, the addresses in a process address space, the keys in a database table, or domain names on a global computer network such as the Internet are meaningful only because they are specified relative to a context.

In prior art systems for identifying data items there is no direct relationship between the data names and the data item. The same data name in two different contexts may refer to different data items, and two different data names in the same context may refer to the same data item.

In addition, because there is no correlation between a data name and the data it refers to, there is no a priori way to confirm that a given data item is in fact the one named by a data name. For instance, in a DP system, if one processor requests that another processor deliver a data item with a given data name, the requesting processor cannot, in general, verify that the data delivered is the correct data (given only the name). Therefore it may require further processing, typically on the part of the requester, to verify that the data item it has obtained is, in fact, the item it requested.

A common operation in a DP system is adding a new data item to the system. When a new data item is added to the system, a name can be assigned to it only by updating the context in which names are defined. Thus such systems require a centralized mechanism for the management of names. Such a mechanism is required even in a multiprocessing system when data items are created and identified at separate processors in distinct locations, and in which there is no other need for communication when data items are added.

In many data processing systems or environments, data items are transferred between different locations in the system. These locations may be processors in the data processing system, storage devices, memory, or the like. For example, one processor may obtain a data item from another processor or from an external storage device, such as a floppy disk, and may incorporate that data item into its system (using the name provided with that data item).

However, when a processor (or some location) obtains a data item from another location in the DP system, it is possible that this obtained data item is already present in the system (either at the location of the processor or at some other location accessible by the processor) and therefore a duplicate of the data item is created. This situation is common in a network data processing environment where proprietary software products are installed from floppy disks onto several processors sharing a common file server. In these systems, it is often the case that the same product will be installed on several systems, so that several copies of each file will reside on the common file server.

In some data processing systems in which several processors are connected in a network, one system is designated as a cache server to maintain master copies of data items, and other systems are designated as cache clients to copy local copies of the master data items into a local cache on an as-needed basis. Before using a cached item, a cache client must either reload the cached item, be informed of changes to the cached item, or confirm that the master item corre-

True Names in Relational and Object-Oriented Databases

Although the preferred embodiment of this invention has been presented in the context of a file system, the invention of True Names would be equally valuable in a relational or object-oriented database. A relational or object-oriented database system using True Names would have similar benefits to those of the file system employing the invention. For instance, such a database would permit efficient elimination of duplicate records, support a cache for records, simplify the process of maintaining cache consistency, provide location-independent access to records, maintain archives and histories of records, and synchronize with distant or disconnected systems or databases.

The mechanisms described above can be easily modified to serve in such a database environment. The True Name registry would be used as a repository of database records. All references to records would be via the True Name of the record. (The Local Directory Extensions table is an example of a primary index that uses the True Name as the unique identifier of the desired records.)

In such a database, the operations of inserting, updating, and deleting records would be implemented by first assimilating records into the registry, and then updating a primary key index to map the key of the record to its contents by using the True Name as a pointer to the contents.

The mechanisms described in the preferred embodiment, or similar mechanisms, would be employed in such a system. These mechanisms could include, for example, the mechanisms for calculating true names, assimilating, locating, realizing, deleting, copying, and moving True Files, for mirroring True Files, for maintaining a cache of True Files, for grooming True Files, and other mechanisms based on the use of substantially unique identifiers.

While the invention has been described in connection with what is presently considered to be the most practical and preferred embodiments, it is to be understood that the invention is not to be limited to the disclosed embodiment, but on the contrary, is intended to cover various modifications and equivalent arrangements included within the spirit and scope of the appended claims.

What is claimed is:

**1**. In a system in which a plurality of files are distributed across a plurality of computers, a method comprising:

obtaining a name for a data file, the name being based at least in part on a given function of the data, wherein the data used by the given function to determine the name comprises the contents of the data file; and

in response to a request for the a data file, the request including at least the name of the particular file, causing a copy of the file to be provided from a given one of the plurality of computers, wherein a copy of the requested file is only provided to licensed parties.

**2**. A method as in claim **1** further comprising:

determining, using at least the name, whether a copy of the data file is present on a particular one of said computers.

**3**. A method as in claim **1** further comprising:

determining, using at least the name, whether an unauthorized or unlicensed copy of the data file is present on a particular one of said computers.

**4**. A method as in claim **1**, further comprising:

maintaining accounting information relating to the data files.

**5**. A method as in claim **4**, wherein the maintaining of accounting information includes at least some of activities selected from:

(a) tracking which files have been stored on a computer; and

(b) tracking which files have been transmitted from a computer.

**6**. A method, in a system in which a plurality of files are distributed across a plurality of computers, wherein data in a file in the system may represent a digital message, a digital image, a video signal or an audio signal, the method comprising:

obtaining a name for a data file, the name having been determined using an MD5 function of the data, wherein the data used by the MD5 function comprises the contents of the data file; and

in response to a request for the data file, the request including at least the name of the data file, providing a copy of the data file from a given one of the plurality of computers, said providing being based at least in part on the obtained name, and wherein a copy of the requested file is only provided to licensed parties.

**7**. A method, in a system in which a plurality of files are distributed across a plurality of computers, wherein some of the computers communicate with each other using a TCP/IP communication protocol, the method comprising:

obtaining a name for a data file, the contents of said data file representing a digital image, the name having been determined using at least a given function of the data in the data file, wherein the data used by the given function to determine the name comprises the contents of the data file; and

in response to a request for the data file, the request including at least the name of the data file, providing a copy of the file from a given one of the plurality of computers, wherein a copy of the requested file is not provided to unlicensed parties or to unauthorized parties.

**8**. A method, in a network comprising a plurality of computers, some of the computers functioning as servers and some of the computers functioning as clients, wherein some computers in the network communicate with each other using a TCP/IP communication protocol, wherein a key is required to identify a file on the network, the method comprising:

storing some files on a first computer in the network and storing copies of some of the files from the first computer on a set of computers distinct from the first computer;

for a particular file, determining a different cache key from an ordinarily used key for the file, the different key being determined at least in part using a message function MD5 of the data, wherein the data used by the function to determine the name comprises the contents of the particular file; and

responsive to a request for the particular file, the request including the different key for the file, causing a copy of the particular file to be provided to the requester,

wherein the requested file is not provided to unlicensed parties, and

wherein the contents of the file may represent: a page in memory, a digital message, a digital image, a video signal or an audio signal.

**9**. A content delivery method, comprising:

distributing files across a network of servers;

for a particular file having a contextual name specifying locations in the network at which the file may be located, determining another name for the particular

file, the other name including at least a data identifier determined using a given function of the data, where said data used by the given function to determine the other name comprises the contents of the particular file;

obtaining a request for the particular file, the request including the contextual name and the other name of the particular file; and

responsive to the request, providing a copy of the particular file from one of the servers of the network of servers, said providing being based, at least in part, on the other name of the particular item, wherein the requested file is not provided to unlicensed parties.

**10**. A method, in a system in which a plurality of files are distributed across a plurality of computers, the method comprising:

obtaining a name for a data file, the name being based at least in part on a given function of the data, wherein the data used by the function comprises the contents of the particular file;

determining, using at least the name, whether a copy of the data file is present on at least one of said computers; and

determining whether a copy of the data file that is present on a at least one of said computers is an unauthorized copy or an unlicensed copy of the data file.

**11**. A method as in claim **10** further comprising:

allowing the file to be provided from one of the computers having an authorized or licensed copy of the file.

**12**. A method as in claim **10** wherein at least some of the plurality of computers comprise a peer-to-peer network.

**13**. A method, in a system in which a plurality of files are distributed across a plurality of computers which form a peer-to-peer network, the method comprising:

obtaining a True Name for a data file, the True Name being based at least in part on a given function of the data, wherein the data used by the given function comprises the contents of the particular file; and

determining, using at least the name, whether an unlicensed or unauthorized copy of the data file is present on a particular computer.

**14**. A method comprising:

obtaining a name for a data file, the name being based at least in part on a function of the data, wherein the data used by the function comprise at least the contents of the file; and

in response to a request for the data file, the request including at least the obtained name of the data file, causing the contents of the data file to be provided from a computer having a licensed copy of the data file.

**15**. A method as in claim **14** wherein the function is a message digest function or a hash function.

**16**. A method as in claim **14** wherein the function is selected from the functions: MD4, MD5, and SHA.

**17**. A method as in claim **14** wherein the given function randomly distributes its outputs.

**18**. A method as in claim **14** wherein the function produces a substantially unique value based on the data comprising the data file.

**19**. A method as in claim **14** wherein a data file may comprise a file, a portion of a file, a page in memory, a digital message, a digital image, a video signal or an audio signal.

**20**. A method as in claim **14** wherein certain processors in the network communicate with each other using a TCP/IP communication protocol.

**21**. A method as in claim **14** wherein said name for said data file, as determined using said function, will change when the data file is modified.

**22**. A method, in a system in which a plurality of files are distributed across a plurality of computers, the method comprising:

obtaining a name for a data file, the name being based at least in part on an MD5 function of the data which comprises the contents of the particular file; and

determining, using at least the obtained name, whether an unauthorized or unlicensed copy of the data file is present on a at least one of said computers.

**23**. A method comprising:

obtaining a list of file names, at least one file name for each of a plurality of files, each of said file names having been determined, at least in part, by applying a function to the contents of the corresponding file; and

using at least said list to determine whether unauthorized or unlicensed copies of some of the plurality of data files are present on a particular computer.

**24**. A method as in claim **23** further comprising:

in response to a request for a particular data file, allowing the contents of the data file to be provided from a computer determined to have a licensed or authorized copy of the data file.

**25**. A method as in claim **23** wherein the particular computer is part of a peer-to-peer network of computers.

**26**. A method as in claim **23** further comprising:

if the computer is found to have a file that it is not authorized or licensed to have, recording information about the computer and about the file.

**27**. A method as in claim **23** wherein the function is a message digest function or a hash function.

**28**. A method as in claim **23** wherein the function is selected from the functions: MD4, MD5, and SHA.

**29**. A method as in claim **23** wherein the given function randomly distributes its outputs.

**30**. A method as in claim **23** wherein the function produces a substantially unique value based on the data comprising the data file.

**31**. A method comprising:

obtaining a list of True Names, one for each of a plurality of files, wherein, for each of the files, the True Name for that file is determined using a function of the contents of the file;

for at least some computers that make up part of a peer-to-peer network of computers, comparing at least some of the contents of the computers to the list of True Names to determine whether unauthorized or unlicensed copies of some of the plurality of data files are present on those computers; and

based at least in part on said comparing, if a computer is found to have content that it is not authorized or licensed to have, recording information about the computer and about the unauthorized or unlicensed content.

**32**. A method as in claim **31** wherein the True Names are determined using a message digest function or a hash function.

**33**. A method as in claim **31** wherein the function is selected from the functions: MD4, MD5, and SHA.

**34**. A method as in claim **31**, further comprising:

in response to a request for the data file, allowing a copy of the file to be provided from a given one of the plurality of computers having an authorized or licensed copy of the file.

**35**. A method comprising:

obtaining a list of True Names, one for each of a plurality of files, wherein, for each of the files, the True Name for

DOCKET ALARM

# Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

## Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

## Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

## Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

## API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

**LAW FIRMS**
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

**FINANCIAL INSTITUTIONS**
Litigation and bankruptcy checks for companies and debtors.

**E-DISCOVERY AND LEGAL VENDORS**
Sync your system to PACER to automate legal marketing.