~~PCT/US98/05304~~

US6421733

Title:

System For Dynamically Transcoding Data Transmitted Between Computers

Abstract:

A system for dynamically transcoding data transmitted between computers is implemented in an apparatus for use in transmitting data between a network server ~~(10)~~ and a network client ~~(12)~~ over a communications link ~~(14).~~. The apparatus includes a parser ~~(22)~~ coupled to a transcode service provider ~~(24).~~. The parser ~~(22)~~ is configured to selectively invoke the transcode service provider ~~(24)~~ in response to a predetermined selection criterion.

Specification:

~~Background of the Invention~~

This application claims the benefit of U.S. Provisional Application No. 60/041,366, filed ~~March~~Mar. 25, 1997.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates generally to the field of data communications for personal computers (PCs), and in particular to a system for dynamically transcoding data transmitted between two computers over a communications link.

2. Related Art

The Internet is quickly becoming the preferred data communications medium for a broad class of computer users ranging from private individuals to large multi-national corporations. Such users now routinely employ the Internet to access information, distribute information, correspond electronically, and even conduct personal conferencing. An ever-growing number of individuals, organizations and businesses have established a presence on the Internet through "web pages" on the World-Wide Web (WWW).

For a wide variety of reasons, it may be desirable to manipulate data transmitted between a local client computer and a network server computer. For example, in certain instances it may be advantageous to dynamically add, modify or delete content retrieved from an Internet server

computer before that content is provided to a client computer. Conversely, it may be advantageous to modify a content request from a client computer prior to transmitting the request to an Internet server computer. While such dynamic manipulation of requests and responses is desirable, it is impractical to expect the expansive Internet infrastructure to quickly change to accommodate such a new capability. For this reason, it is desirable to implement such new capabilities in a way that does not require changes to either existing client computers or Internet server computers.

It is known to deploy a proxy server, or network proxy, as an intermediary between one or more client computers and an external network such as the Internet. Network proxies are described generally in Ian S. Graham, HTML Source Book: A Complete Guide to HTML 3.0 403 (2d ed. 1996). One common application for a proxy server is as a so-called "firewall," wherein the proxy server is responsible for all communications with the outside world. In other words, local devices are not permitted to communicate directly with external network computers, such as Internet servers. Instead, each local device directs requests for ~~network-resident~~networkresident data to the proxy server. When the proxy server receives such a request, it forwards the request to the appropriate external computer, receives the response from the external computer, and then forwards the response to the local device. The external computer thus has no knowledge of the local devices. In this way, the local devices are protected from potential dangers such as unauthorized access.

Existing proxy servers do not manipulate the data passing through them. In essence, proxy servers are merely blind conduits for requests and responses. This limitation of existing proxy servers restricts these devices from being used to full advantage when facilitating communications between local devices and network devices. There is therefore a need for a so-called "smart" proxy capable of examining the data passing through it, whether it be a request intended for an external network device or network content being returned to a local device, and dynamically acting upon that data. Such a device can be used to transparently provide a wide range of services that were heretofore impossible without modifying existing Internet infrastructure.

SUMMARY OF THE INVENTION

Embodiments of the present invention relate to devices, systems and methods for transcoding information transmitted between computers, such as a network server computer and a network client computer.

According to one embodiment, an apparatus for use in transmitting data between a network server and a network client over a communications link includes a parser coupled to a transcode service provider. The parser is configured to selectively invoke the transcode service provider in response to a predetermined selection criterion.

~~Brief~~ Description ~~of the Drawings~~

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a schematic diagram illustrating an environment in which embodiments of the present invention may be applied.

FIG. 2 is a schematic diagram illustrating a transcoder module according to an embodiment of the present invention.

FIG. 3 is a schematic diagram illustrating an embodiment of the present invention for a non-enabled network client.

FIG. 4 is a schematic diagram illustrating an example of a user interface for providing a non-enabled network client with control over transcoding functionality.

FIG. 5 is a schematic diagram illustrating an embodiment of the present invention for an enabled network client.

FIG. 6 is a schematic diagram illustrating a network client with transcoding functionality integrated in a browser according to an embodiment of the present invention.

FIGS. 7-9 are flow charts illustrating logic for presenting a requested URL object to a network client according to an embodiment of the present invention.

DETAILED DESCRIPTION

Embodiments of the present invention provide the ability to dynamically transcode information transmitted between, for example, a network server computer and a network client computer. As used herein, the term "transcode" applies to virtually any manipulation of data including, but not limited to, adding, modifying or deleting data.

Referring now to FIG. 1, which illustrates an environment in which embodiments of the present invention may be advantageously applied, a network server 10 manages the transfer of data from the Internet 18 to a network client 12. Network client 12 may be any computer having suitable data communications capability.

Network client 12 communicates requests for information to, and receives information from, network server 10 over a client/server communications link 14. Client/server communications link 14 may comprise, for example, a so-called "slow network" using, for example, POTS (Plain Old Telephone System) dial-up technology or wireless connections. Alternatively, client/server communications link 14 may comprise a so-called "fast network," such as a LAN or WAN (Wide Area Network), which is capable of operating at much higher speeds than are possible with slow networks. Combinations of these access methods are also possible. For example, network client 12 may use a POTS or wireless dial-up connection to a modem bank maintained by an ISP (Internet Service Provider), which is in turn connected to network server 10 over a LAN. Network server 10 communicates with computers resident on Internet 18 through server/network communications link 16, which may comprise any suitable communications medium known in the art.

According to a first general embodiment of the present invention, illustrated schematically in FIG. 2, a transcoder 20 includes a parser 22 and a plurality of transcode service providers 24. Parser 22 is configured to act upon data received by transcoder 20, such as a request for a network object generated by a client device or a reply to such a request provided by a content server device. In this particular embodiment, parser 22 is responsible for selectively

invoking one or more of transcode service providers 24 based upon a predetermined selection criterion.

Transcoder 20 may be implemented, for example, as a software module installed in a network proxy, in a client device, in a network server device, or in a content server device. In one particular implementation, illustrated in FIG. 3, transcoder 20 is installed in a remote transcoding server 34 arranged between network client 12 and Internet 18. Transcoding server 34 may comprise, or be a part of, a network server, a stand-alone computer in communication with a network server, or a distributed system of computers. Remote transcoding server 34 may be coupled, for example, to an ISP's network, a corporate network, or anywhere on Internet 18, and may provide multiple users (i.e., clients) with a means to obtain content on Internet 18.

In the particular embodiment illustrated in FIG. 3, transcoding server 34 includes an HTTP (HyperText Transfer Protocol) remote proxy 36, capable of accessing Internet 18 over server/network communications link 16. HTTP remote proxy 36 differs from known network proxies, which generally are little more than a conduit for requests to, and replies from, external Internet resources, in that it is capable not only of examining such requests and replies, but also of acting upon commands in the requests by, for example, determining whether or not to transcode content. Moreover, using transcoder 20, HTTP remote proxy 36 is capable of changing content received from Internet 18 prior to returning it to a requesting network client 12, as is explained further below.

Looking more closely at the embodiment in FIG. 3, transcoder 20 is coupled to HTTP remote proxy 36. Parser 22 manages the transcoding of data to be transmitted from transcoding server 34 to network client 12. To this end, parser 22 controls transcode service providers 24 to selectively transcode content based on a predetermined selection criterion. For example, one or more transcode service providers 24 may provide the capability to compress and/or scale different types of data content, such as image, video, or HTML (HyperText Markup Language). Such uses are described further in co-pending U.S. patent applications ~~Serial~~Ser. No. 08/772,164 entitled "System for Enhancing Data Access Over a Communications Link," filed on ~~December~~Dec. 20, 1996, and ~~Serial~~Ser. No. 08/799,654 entitled "Method and Apparatus for Scaling Image Data," filed on ~~February~~Feb. 11, 1997, both of which are assigned to Intel Corporation. For purposes of illustrating certain features of the present invention, a number of embodiments are described below in terms of content scaling/compression; however, as is explained, transcode service providers 24 may provide a wide variety of transcoding functions.

As shown in FIG. 3, transcoding server 34 may also include a server-side cache memory 30 managed by a server-side cache interface 28. Server-side cache memory 30 may be used to store both original and transcoded versions of content for later transmission to network client 12 without the need to re-retrieve the content from Internet 18 or to re-transcode the content.

Transcoding server 34 is coupled to network client 12 by client/server communications link 14. Network client 12 includes a browser 32, such as the Netscape Navigator v.3.0 browser (although the invention is not limited in this respect), which manages the presentation of data to a user. In this embodiment, network client 12 is "non-enabled," meaning no specialized transcoding software is preloaded on network client 12.

Parser 22 may comprise a relatively simple, uniform interface to HTTP remote proxy 36, and may provide an API (Application Programming Interface) for transcoding data received by HTTP remote proxy 36. Parser 22 manages one or more transcode service providers 24 that are accessed through a common SPI (Service Provider Interface). In this particular embodiment, parser 22 is designed in compliance with the Windows Open Systems Architecture (WOSA), and may be implemented as a Win32 DLL (Dynamic Link Library). The WOSA architecture, described in Readings on Microsoft Windows and WOSA (Microsoft Corp. 1995), enables additional transcode service providers 24 to be dynamically added to the system to provide new features and/or better transcoding algorithms, while at the same time not requiring changing or retesting other software components in the system. This feature is especially beneficial where transcoding server 34 also interacts with "enabled" network clients equipped with specialized transcoding software. It should be noted that some of the features of parser 22 described below may be inapplicable to the non-enabled client embodiment of FIG. 3; however, transcoding server 34 may advantageously be configured flexibly enough to process requests from both non-enabled and enabled network clients.

Like parser 22, server-side cache interface 28 may be modeled after a standard Get/Set interface. Server-side cache memory 30 essentially "owns" all cached objects, in that it manages the properties and storage of the objects and may invalidate any non-locked object at any time; however, the actual format of any given cached object is known only by parser 22 and its associated transcode service providers 24. Thus, for data integrity and transcoding efficiency purposes, all access to server-side cache memory 30 in this embodiment is through parser 22.

Server-side cache interface 28 may include the following calls: CreateEntry(URL, &Entry, . . .); GetEntry(URL, &Entry); CreateStream(Entry, &StreamEntry, . . .); GetStream(Entry, &StreamEntry, . . .); CloseEntry(Entry); CloseStreamEntry(StreamEntry); GetProperties(Entry, &Properties, . . .); SetProperties(Entry, &Properties, . . .); Read(StrearnEntry, &OutStream, . . .); Write(StreamEntry, &InStream, . . .).

~~CreateEntry(URL, &Entry, ...);~~

~~GetEntry(URL, &Entry);~~

~~CreateStream(Entry, &StreamEntry, ...);~~

~~GetStream(Entry, &StreamEntry, ...);~~

~~CloseEntry(Entry);~~

~~CloseStreamEntry(StreamEntry);~~

~~GetProperties(Entry, &Properties, ...);~~

~~SetProperties(Entry, &Properties, ...);~~

~~Read(StreamEntry, &OutStream, ...);~~

~~Write(StreamEntry, &InStream, ...).~~

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.