# METHOD AND APPARATUS FOR LOCAL IP ADDRESS TRANSLATION

## BACKGROUND OF THE INVENTION

### 1. Technical Field:

The present invention relates generally to an improved data processing system and in particular to a method and apparatus for locating data. Still more particularly, the present invention provides a method and apparatus for locating documents using an improved domain name system.

### 2. Description of Related Art:

The Internet, also referred to as an "internetwork", is a set of computer networks, possibly dissimilar, joined together by means of gateways that handle data transfer and the conversion of messages from a protocol of the sending network to a protocol used by the receiving network. When capitalized, the term "Internet" refers to the collection of networks and gateways that use the TCP/IP suite of protocols.

The Internet has become a cultural fixture as a source of both information and entertainment. Many businesses are creating Internet sites as an integral part of their marketing efforts, informing consumers of the products or services offered by the business or providing other information seeking to engender brand loyalty. Many federal, state, and local government agencies are also employing Internet sites for informational purposes, particularly agencies which must interact with virtually all segments of society such as the Internal Revenue Service and secretaries of state. Providing informational guides and/or searchable databases of online public records may reduce operating costs. Further, the Internet is becoming increasingly popular as a medium for commercial transactions.

Currently, the most commonly employed method of transferring data over the Internet is to employ the World Wide Web environment, also called simply "the Web". Other Internet resources exist for transferring information, such as File Transfer Protocol (FTP) and Gopher, but have not achieved the popularity of the Web. In the Web environment, servers and clients effect data transaction using the Hypertext Transfer Protocol (HTTP), a known protocol for handling the transfer of various data files (e.g., text, still graphic images, audio, motion video, etc.). The information in various data files is formatted for presentation to a user by a standard page description language, the Hypertext Markup Language

(HTML). In addition to basic presentation formatting, HTML allows developers to specify "links" to other Web resources identified by a Uniform Resource Locator (URL). A URL is a special syntax identifier defining a communications path to specific information. Each logical block of information accessible to a client, called a "page" or a "Web page", is identified by a URL. The URL provides a universal, consistent method for finding and accessing this information, not necessarily for the user, but mostly for the user's Web "browser". A browser is a program capable of submitting a request for information identified by an identifier, such as, for example, a URL.

A user may enter a domain name through a graphical user interface (GUI) for the browser to access a source of content. The domain name is automatically converted to the Internet Protocol (IP) address by a domain name system (DNS), which is a service that translates the symbolic name entered by the user into an IP address by looking up the domain name in a database. A significant amount of time used in traversing the Web involves time spent trying to translate domain names into IP addresses. Such a process can be time consuming because DNS tables must be looked up on DNS servers. Such a process is especially time consuming when a dial up connection is used.

Therefore, it is desirable to provide an improved method, apparatus, and computer instructions for providing domain name system translations that reduce the time consuming process of obtaining domain name system translations from DNS servers.

## SUMMARY OF THE INVENTION

Seen from a first aspect, the present invention provides a first method in a data processing system for requesting a page from a server across a network data processing system, the method comprising: requesting, by a client application in the data processing system, the page a first time, wherein the page is requested using a domain name; looking up an Internet Protocol address for the domain name for the page on a remote domain name server; using the Internet Protocol address to request the page; storing the Internet Protocol address locally in the data processing system in a database of Internet Protocol translations, wherein the Internet Protocol address for the page is stored in association with the domain name for the page and wherein the database is accessible only by the client application and used by the client application in a subsequent request for the page; responsive to a request

for a selected page, referencing the database of Internet Protocol translations to determine whether an Internet Protocol address for a selected page is present; requesting the selected page using an Internet Protocol address returned from the database of Internet Protocol translations if the Internet Protocol address is present in the database of Internet Protocol translations; and querying the remote domain name server to determine the Internet Protocol address of the selected page if the Internet Protocol address for the selected page cannot be determined from referencing the database of Internet Protocol translations.

Preferably, there is provided a step of indicating in the database of Internet Protocol translations that certain Internet Protocol addresses of specified domain names are to be excluded from the database.

Preferably, there is provided a step of performing the looking and using steps only if a selected user input is received.

Preferably, there is provided a step of updating the database of Internet Protocol translations, by the client application, if a given Internet Protocol address for a given domain name has changed.

Preferably, there is provided a step of automatically receiving an updated Internet Protocol address for a given domain name if a given Internet Protocol address for the given domain name has changed.

A method is preferred in a data processing system for identifying Internet Protocol addresses, the method comprising: identifying a domain name for a selected page; querying a local data structure of address translations using the domain name for the page to determine whether an Internet Protocol address is present for the domain name for the requested page; responsive to identifying the Internet Protocol address using the local data structure, requesting the selected page;

responsive to an inability to identify the Internet Protocol address using the local data structure, obtaining the Internet Protocol address from a remote domain name server; responsive to obtaining the Internet Protocol address from the remote domain name server, requesting the selected page; and responsive to obtaining the Internet Protocol address from the remote domain name server, updating the local data structure to include the Internet Protocol address in association with the domain name.

Preferably, wherein the querying step is performed only in response to a selected user input.

Preferably, the data structure is a database.

Preferably, there is provided a step of automatically updating the local data structure in response to a change in the Internet Protocol address for the domain name.

From a preferred viewpoint, there is provided a data processing system for requesting a page from a server across a network data processing system, the data processing system comprising: a bus system; a communications unit connected to the bus system; a memory connected to the bus system, wherein the memory includes a set of instructions; and a processing unit connected to the bus system, wherein the processing unit executes the set of instructions to request, by a client application in the data processing system, the page a first time, wherein the page is requested using a domain name; look up an Internet Protocol address for the domain name for the page on a remote domain name server; use the Internet Protocol address to request the page; store the Internet Protocol address locally in the data processing system in a data structure of Internet Protocol translations, wherein the Internet Protocol address for the page is stored in association with the domain name for the page and wherein the data structure is accessible only by the client application and used by the client application in a subsequent request for the page; reference the data structure of Internet Protocol translations to determine whether an Internet Protocol address for a selected page is present in response to a request for a selected page; request the selected page using an Internet Protocol address returned from the data structure of Internet Protocol translations if the Internet Protocol address is present in the data structure of Internet Protocol translations; and query the remote domain name server to determine the Internet Protocol address of the selected page if the Internet Protocol address for the selected page cannot be determined from referencing the data structure of Internet Protocol translations.

Preferred is a data processing system for identifying Internet Protocol addresses, the data processing system comprising: a bus system; a communications unit connected to the bus system; a memory connected to the bus system, wherein the memory includes a set of instructions; and a processing unit connected to the bus system, wherein the processing unit executes the set of instructions to identify a domain name for a selected

page; query a local data structure of address translations using the domain name for the page to determine whether an Internet Protocol address is present for the domain name for the requested page; request the selected page in response to identifying the Internet Protocol address using the local data structure; obtain the Internet Protocol address from a remote domain name server in response to an inability to identify the Internet Protocol address using the local data structure; request the selected page in response to obtaining the Internet Protocol address from the remote domain name server; and update the local data structure to include the Internet Protocol address in association with the domain name in response to obtaining the Internet Protocol address from the remote domain name server.

In a second aspect, there is provided a data processing system for requesting a page from a server across a network data processing system, the data processing system comprising: requesting means for requesting, by a client application in the data processing system, the page a first time, wherein the page is requested using a domain name; looking means for looking up an Internet Protocol address for the domain name for the page on a remote domain name server; using means for using the Internet Protocol address to request the page; storing means for storing the Internet Protocol address locally in the data processing system in a data structure of Internet Protocol translations, wherein the Internet Protocol address for the page is stored in association with the domain name for the page and wherein the data structure is accessible only by the client application and used by the client application in a subsequent request for the page; referencing means, responsive to a request for a selected page, for referencing the data structure of Internet Protocol translations to determine whether an Internet Protocol address for a selected page is present; requesting means for requesting the selected page using an Internet Protocol address returned from the database of Internet Protocol translations if the Internet Protocol address is present in the database of Internet Protocol translations; and querying means for querying the remote domain name server to determine the Internet Protocol address of the selected page if the Internet Protocol address for the selected page cannot be determined from referencing the database of Internet Protocol translations.

Preferably, there is provided an indicating means for indicating in the database of Internet Protocol translations that certain Internet Protocol addresses of specified domain names are to be excluded from the database.

Preferably, there is provided a performing means for performing the looking and using steps only if a selected user input is received.

Preferably, there is provided an updating means for updating the database of Internet Protocol translations, by a client application, if a given Internet Protocol address for a given domain name has changed.

Preferably, there is provided a receiving means for automatically receiving an updated Internet Protocol address for a given domain name if a given Internet Protocol address for the given domain name has changed.

Preferred is a data processing system for identifying Internet Protocol addresses, the data processing system comprising: identifying means for identifying a domain name for a selected page; querying means for querying a local data structure of address translations using the domain name for the page to determine whether an Internet Protocol address is present for the domain name for the requested page; requesting means, responsive to identifying the Internet Protocol address using the local data structure, for requesting the selected page; obtaining means, responsive to an inability to identify the Internet Protocol address using the local data structure, for obtaining the Internet Protocol address from a remote domain name server; requesting means, responsive to obtaining the Internet Protocol address from the remote domain name server, for requesting the selected page; and updating means, responsive to obtaining the Internet Protocol address from the remote domain name server, for updating the local data structure to include the Internet Protocol address in association with the domain name.

Preferably, the querying means is performed only in response to a selected user input.

Preferably the data structure is a database.

Preferably, there is provided an updating means for automatically updating the local data structure in response to a change in the Internet Protocol address for the domain name.

In a third aspect, there is provided a computer program comprising computer program code to, when loaded into a computer system and executed thereon, cause the said computer system to perform all the steps of a method according to the first aspect.

Preferred is that the invention be embodied as a computer program product in a computer readable medium for requesting a page from a server across a network data processing system, the computer program product comprising: first instructions for requesting, by a client application in the data processing system, the page a first time, wherein the page is requested using a domain name; second instructions for looking up an Internet Protocol address for the domain name for the page on a remote domain name server; third instructions for using the Internet Protocol address to request the page; fourth instructions for storing the Internet Protocol address locally in the data processing system in a database of Internet Protocol translations, wherein the Internet Protocol address for the page is stored in association with the domain name for the page and wherein the database is accessible only by the client application and used by the client application in a subsequent request for the page; fifth instructions, responsive to a request for a selected page, for referencing the database of Internet Protocol translations to determine whether an Internet Protocol address for a selected page is present; sixth instructions for requesting the selected page using an Internet Protocol address returned from the database of Internet Protocol translations if the Internet Protocol address is present in the database of Internet Protocol translations; and seventh instructions for querying the remote domain name server to determine the Internet Protocol address of the selected page if the Internet Protocol address for the selected page cannot be determined from referencing the database of Internet Protocol translations.

Preferred is that the invention be embodied as a computer program product in a computer readable medium for identifying Internet Protocol addresses, the computer program product comprising: first instructions for identifying a domain name for a selected page; second instructions for querying a local data structure of address translations using the domain name for the page to determine whether an Internet Protocol address is present for the domain name for the requested page; third instructions, responsive to identifying the Internet Protocol address using the local data structure, for requesting the selected page; fourth instructions, responsive to an inability to identify the Internet Protocol address using the local data structure, for obtaining the Internet Protocol address from

a remote domain name server; fifth instructions, responsive to obtaining the Internet Protocol address from the remote domain name server, for requesting the selected page; and sixth instructions, responsive to obtaining the Internet Protocol address from the remote domain name server, for updating the local data structure to include the Internet Protocol address in association with the domain name.

The aspects and preferred features of the present invention thus provide a method, apparatus, and computer instructions for requesting a page from a server across a network data processing system. The page is requested a first time from a client application in the data processing system in which the page is requested using a domain name. An Internet Protocol address for the domain name is looked up for the page on a remote domain name server. The Internet Protocol address is used to request the page. The Internet Protocol address is stored locally in the data processing system in a database of Internet Protocol translations in association with the domain name for the page and wherein the database is accessible only by the client application. As a result, subsequent requests for the page may be processed using this database. The database of Internet Protocol translations is referenced to determine whether an Internet Protocol address for a selected page is present in response to a request for the selected page. The selected page is requested using an Internet Protocol address returned from the database of Internet Protocol translations if the Internet Protocol address is present in the database of Internet Protocol translations. The remote domain name server is queried to determine the Internet Protocol address of the selected page if the Internet Protocol address for the selected page cannot be determined from referencing the database of Internet Protocol translations.

## BRIEF DESCRIPTION OF THE DRAWINGS

A preferred embodiment of the present invention will now be described, by way of example only, with reference to the accompanying drawings, wherein:

**Figure 1** depicts a pictorial representation of a network of data processing systems in which the present invention may be implemented;

**Figure 2** is a block diagram illustrating a data processing system in which the present invention may be implemented;

**Figure 3** is a diagram illustrating components used in providing local IP address translations in accordance with a preferred embodiment of the present invention;

**Figure 4** is a flowchart of a process used for translating domain names in accordance with a preferred embodiment of the present invention;

**Figure 5** is a flowchart of a process used for updating a domain name translation database in accordance with a preferred embodiment of the present invention; and

**Figure 6** is a flowchart of a process for selectively creating entries in a DNS translations database in accordance with a preferred embodiment of the present invention.

## DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

With reference now to the figures, **Figure 1** depicts a pictorial representation of an exemplary network of data processing systems in which the preferred embodiment of the present invention may be implemented. Network data processing system **100** is a network of computers in which the preferred embodiment of the present invention may be implemented. Network data processing system **100** contains a network **102**, which is the medium used to provide communications links between various devices and computers connected together within network data processing system **100**. Network **102** may include connections, such as wire, wireless communication links, or fiber optic cables.

In the depicted example, server **104** is connected to network **102** along with storage unit **106**. In addition, clients **108**, **110**, and **112** are connected to network **102**. These clients **108**, **110**, and **112** may be, for example, personal computers or network computers. In the depicted example, server **104** provides data, such as boot files, operating system images, and applications to clients **108-112**. In these examples, server **104** may be a DNS server and provide translations of domain names into IP addresses for clients **108**, **110**, and **112**. When acting as a DNS server, server **104** maintains a database of domain names (host names) and their corresponding IP addresses. For example, if www.company.com were presented to a DNS server, the IP address 214.26.68.51 would be returned to the client. Network data processing system **100** may include additional servers, clients, and other devices not shown.

The present invention provides an improved method, apparatus, and computer instructions for translating domain names into IP addresses by providing a local translation mechanism to the client. This mechanism provides an advantage in reducing the amount of time spent obtaining IP address translations from a DNS server, such as server **104**.

In the depicted example, network data processing system **100** is the Internet with network **102** representing a worldwide collection of networks and gateways that use the Transmission Control Protocol/Internet Protocol (TCP/IP) suite of protocols to communicate with one another. At the heart of the Internet is a backbone of high-speed data communication lines between major nodes or host computers, consisting of thousands of commercial, government, educational and other computer systems that route data and messages. Of course, network data processing system **100** also may be implemented as a number of different types of networks, such as for example, an intranet, a local area network (LAN), or a wide area network (WAN). **Figure 1** is intended as an example, and not as an architectural limitation for the embodiments of the present invention.

With reference now to **Figure 2**, a block diagram illustrating a data processing system is depicted in which the preferred embodiment of present invention may be implemented. Data processing system **200** is an example of a client computer, such as client **108** in **Figure 1.**

Data processing system **200** employs a peripheral component interconnect (PCI) local bus architecture. Although the depicted example employs a PCI bus, other bus architectures such as Accelerated Graphics Port (AGP) and Industry Standard Architecture (ISA) may be used. Processor **202** and main memory **204** are connected to PCI local bus **206** through PCI bridge **208**. PCI bridge **208** also may include an integrated memory controller and cache memory for processor **202**. Additional connections to PCI local bus **206** may be made through direct component interconnection or through add-in boards. In the depicted example, local area network (LAN) adapter **210**, small computer system interface (SCSI) host bus adapter **212**, and expansion bus interface **214** are connected to PCI local bus **206** by direct component connection. In contrast, audio adapter **216**, graphics adapter **218**, and audio/video adapter **219** are connected to PCI local bus **206** by add-in boards inserted into expansion slots. Expansion bus interface **214** provides a connection for a keyboard and mouse adapter **220**, modem **222**, and additional memory **224**. Small computer system interface (SCSI) host bus

adapter **212** provides a connection for hard disk drive **226**, tape drive **228**, and CD-ROM drive **230**.

An operating system runs on processor **202** and is used to coordinate and provide control of various components within data processing system **200** in **Figure 2**. The operating system may be a commercially available operating system, such as Windows XP, which is available from Microsoft Corporation. An object oriented programming system such as Java may run in conjunction with the operating system and provide calls to the operating system from Java programs or applications executing on data processing system **200**. "Java" is a trademark of Sun Microsystems, Inc. Instructions for the operating system, the object-oriented operating system, and applications or programs are located on storage devices, such as hard disk drive **226**, and may be loaded into main memory **204** for execution by processor **202**.

Those of ordinary skill in the art will appreciate that the hardware in **Figure 2** may vary depending on the implementation. Other internal hardware or peripheral devices, such as flash read-only memory (ROM), equivalent nonvolatile memory, or optical disk drives and the like, may be used in addition to or in place of the hardware depicted in **Figure 2**. Also, the processes of the preferred embodiment of present invention may be applied to a multiprocessor data processing system.

As another example, data processing system **200** may be a stand-alone system configured to be bootable without relying on some type of network communication interfaces As a further example, data processing system **200** may be a personal digital assistant (PDA) device, which is configured with ROM and/or flash ROM in order to provide non-volatile memory for storing operating system files and/or user-generated data.

The depicted example in **Figure 2** and above-described examples are not meant to imply architectural limitations. For example, data processing system **200** also may be a notebook computer or hand held computer in addition to taking the form of a PDA. Data processing system **200** also may be a kiosk or a Web appliance.

Turning next to **Figure 3**, a diagram illustrating components used in providing local IP address translations is depicted in accordance with a preferred embodiment of the present invention. Client application **300** executes on a client, such as data processing system **200** in **Figure 2**, and may request a page from a Web server, such as Web server **302**. Web server

**302** is located on another data processing system, such as a server. Typically, a domain name is entered. Therefore, the domain name must be translated into an IP address using domain name server (DNS) **304**, which is located on another data processing system.

In this example, client application **300** is a browser, but may be any application that requests data across a network using domain names and IP addresses. The domain name is sent to DNS server **304** across a network, such as network **102** in **Figure 1**, in request **306**. DNS server **304** contains tables for translating the domain name in request **306** into an IP address. As a result, IP address **308** is returned to client application **300**. Request **310** is generated and sent to Web server **302** using IP address **308**. As a result, Web server **302** returns Web page **312** to client application **300**.

Each time a new domain is translated into an IP address by DNS server **304**, client application **300** locally saves the IP address in association with the domain name in DNS translations database **314**. This database is located locally in the data processing system in which client application **300** is located. In these examples, this database is only accessible by client applications. Of course, any data structure, such as a table, may be used to store these translations for use. In this manner, the next time the same page or domain name is requested, client application **300** may query DNS translations database **314** to determine whether the IP address is present. By performing this IP address translation locally rather than querying DNS server **304**, time may be saved with respect to the time needed to send the request and receive a result from DNS server **304**. Such an advantage is especially evident with respect to dial up connections.

Further, depending on the particular implementation, a user may specify that selected domain names are never stored or looked for in DNS translations database **314**. For example, if the user performs a shift reload command, the local IP address translation may be skipped and the normal process of loading a page using remote DNS server **304** may be employed. Certain domain names or pages may be skipped for various reasons. For example, an IP address for a domain name may be dynamic, changing from time to time. In this case, using a local database to identify the IP address is undesirable. Further, client application **300** may check to see if the IP address for the domain name has been updated. If such an update has occurred, the appropriate translation in DNS translations database **314** may be updated.

Further, Web server **302** could automatically send a new IP address to client application **300** when such a change occurs. This new IP address would be automatically updated in DNS translations database **314**. In these examples, the updates to DNS translations database **314** are made by client applications, such as client application **300**. Applications other than client applications are not allowed to access or update this database in the depicted examples.

Additionally, updates to DNS translations database **314** may be performed automatically at off-peak or off-use times. In such an update, client application **300** would query DNS server **304** for IP addresses of all of the domain names stored in DNS translations database **314**. Changes in IP addresses for domain names stored in DNS translations database **314** may be automatically updated when a comparison is made between the current IP address stored in DNS translations database **314** and the IP address returned by DNS server **304**.

With reference now to **Figure 4**, a flowchart of a process used for translating domain names is depicted in accordance with a preferred embodiment of the present invention. The process illustrated in **Figure 4** may be implemented in a client process, such as client application **300** in **Figure 3**.

The process begins by identifying a request with a domain name (step **400**). A determination is made as to whether the local translation process is to be skipped (step **402**). The local translation process may be skipped if some selected user input, such as a shift reload command, is received. If the local translation process is not to be skipped, the local database is queried for an IP address (step **404**). Local translation processes may be skipped in some situations, such as for sites known to have dynamic IP addresses. The local database is a database, such as DNS translations database **314** in **Figure 3**. Of course, other types of data structures may be used and queried. These data structures may include, for example, a table or a linked list. The result is received from the local database (step **406**). A determination is made as to whether the IP address is present in the result. If the IP address is present in the result, the page is requested using the returned IP address (step **410**) and the process terminates thereafter.

Referring again to step **408**, if the IP address is not present in the result, the request is sent to the DNS server (step **412**). A result

containing the IP address is received from the DNS server (step **414**). Thereafter, an entry is created in the local database (step **416**) and the process proceeds to step **410** as described above. With reference again to step **402**, if the local translation process is to be skipped, a request is sent to the DNS server (step **418**). A response is received (step **420**) and the process terminates thereafter. With respect to step **418**, if the domain name is already present in the local database, the IP address received in step **418** may be compared to the IP address for the domain name in the local database. If a new IP address is present, an update may be made in the local database.

Turning now to **Figure 5**, a flowchart of a process used for updating a domain name translation database is depicted in accordance with a preferred embodiment of the present invention. The process illustrated in **Figure 5** may be implemented in a client process, such as client application **300** in **Figure 3**. In these illustrations, the client application performs the update to the domain name translation database.

The process begins by selecting a domain name from the local database (step **500**). The DNS server is queried for an IP address using the selected domain name (step **502**). A result is returned by the DNS server (step **504**). The IP address in the result from the DNS server is compared with the current IP address in the local IP database (step **506**). A determination is made as to whether a match between the IP addresses is present (step **508**). If a match is present, a determination is made as to whether additional unprocessed domain names remain in the local database (step **510**). If additional unprocessed domain names are present in the local database, the process returns to step **500**. Otherwise, the process terminates.

Turning back to step **508**, if a match is absent between the IP address returned from the DNS server and the current IP address in the local database, the IP address in the local database is updated with the new IP address (step **512**) with the process then proceeding to step **510** as described above. The process illustrated in **Figure 5** may be performed at times in which traffic is low on the Internet. For example, this process may be performed at 1:00 a.m. in the morning. Additionally, this process is typically not performed while the user is actively using the data processing system.

Turning next to **Figure 6**, a flowchart of a process for selectively creating entries in a DNS translations database is depicted in accordance with a preferred embodiment of the present invention. The process illustrated in **Figure 6** is a more detailed description of an alternative step for creating an entry in a local database used to perform DNS translations. More specifically, the steps in **Figure 6** may be implemented in step **416** of **Figure 4**.

The process begins by comparing the domain name sent in the request to the DNS server to a queue of visited sites (step **600**). In this example, a queue of visited sites is maintained similar to a history that is maintained in a browser program. A number of visited sites, such as 500 entries, may be stored in the queue.

A determination is made as to whether the domain name of the site for which a DNS translation was requested has been visited more than some threshold level (step **602**). The threshold may take various forms depending on the particular implementation. For example, the threshold may be set to be exceeded if a site is visited more than five times within some period of time, such as two days. Alternatively, the threshold may be exceeded if the site is visited ten times within any period of time. If this threshold is exceeded, a prompt is displayed to the user (step **604**). This prompt may be a pop-up window stating: "You have visited this site five times in the last two days. Do you wish to add this site to a list of local translations?" This pop-up window also may include appropriate graphical controls to receive user inputs. A determination is made as to whether the user desires to add the site to the DNS translations database (step **606**). If the user desires to add the site, the domain name for the site is added to the DNS translations database as a new entry for use in performing local domain name translations (step **608**) with the process terminating thereafter.

Turning back to step **606**, if the user does not wish to add the site to the DNS translations database, the process terminates. Further, the process also terminates if the threshold is not exceeded.

Thus, the preferred and alternative embodiments of present invention provide an improved method, apparatus, and computer instructions for locally translating domain names into IP addresses. When a Web page is requested using a domain name, a local domain name translation database is queried to see whether the IP address is present. If the IP address for the domain name is present, the Web page is then requested without having

to contact or query a remote DNS server. If an IP address is not present for the domain name, the normal process of querying the DNS server is performed. The returned IP address and domain name are added as an entry in the database. In this manner, time may be saved by users traversing the Web by avoiding the time needed to query and receive an IP address for a domain name from a DNS server.

It is important to note that while the preferred embodiment of the present invention has been described in the context of a fully functioning data processing system, those of ordinary skill in the art will appreciate that the processes of the preferred embodiment of the present invention are capable of being distributed in the form of a computer readable medium of instructions and a variety of forms and that the preferred embodiment of the present invention applies equally regardless of the particular type of signal bearing media actually used to carry out the distribution. Examples of computer readable media include recordable-type media, such as a floppy disk, a hard disk drive, a RAM, CD-ROMs, DVD-ROMs, and transmission-type media, such as digital and analog communications links, wired or wireless communications links using transmission forms, such as, for example, radio frequency and light wave transmissions. The computer readable media may take the form of coded formats that are decoded for actual use in a particular data processing system.

The description of the preferred embodiment of the present invention has been presented for purposes of illustration and description, and is not intended to be exhaustive or limited to the invention in the form disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art. The embodiment was chosen and described in order to best explain the principles of the invention, the practical application, and to enable others of ordinary skill in the art to understand the various embodiments the invention with such modifications as are suited to the particular use contemplated.

**CLAIMS:**

1. A method in a data processing system for requesting a page from a server across a network data processing system, the method comprising:

requesting, by a client application in the data processing system, the page a first time, wherein the page is requested using a domain name;

looking up an Internet Protocol address for the domain name for the page on a remote domain name server;

using the Internet Protocol address to request the page;

storing the Internet Protocol address locally in the data processing system in a data structure of Internet Protocol translations, wherein the Internet Protocol address for the page is stored in association with the domain name for the page and wherein the data structure is accessible only by the client application and used by the client application in a subsequent request for the page;

responsive to a request for a selected page, referencing the data structure of Internet Protocol translations to determine whether an Internet Protocol address for a selected page is present;

requesting the selected page using an Internet Protocol address returned from the data structure of Internet Protocol translations if the Internet Protocol address is present in the data structure of Internet Protocol translations; and

querying the remote domain name server to determine the Internet Protocol address of the selected page if the Internet Protocol address for the selected page cannot be determined from referencing the data structure of Internet Protocol translations.

2. The method of claim 1 further comprising:

indicating in the data structure of Internet Protocol translations that certain Internet Protocol addresses of specified domain names are to be excluded from the database.

3. The method of claim 1 or claim 2, further comprising:

performing the looking and using steps only if a selected user input is received.

4. The method of any of claims 1 to 3, further comprising:

updating the data structure of Internet Protocol translations, by the client application, if a given Internet Protocol address for a given domain name has changed.

5. The method of any of claims 1 to 4, further comprising:

automatically receiving an updated Internet Protocol address for a given domain name if a given Internet Protocol address for the given domain name has changed.

6. A data processing system for requesting a page from a server across a network data processing system, the data processing system comprising:

requesting means for requesting, by a client application in the data processing system, the page a first time, wherein the page is requested using a domain name;

looking means for looking up an Internet Protocol address for the domain name for the page on a remote domain name server;

using means for using the Internet Protocol address to request the page;

storing means for storing the Internet Protocol address locally in the data processing system in a data structure of Internet Protocol translations, wherein the Internet Protocol address for the page is stored in association with the domain name for the page and wherein the data structure is accessible only by the client application and used by the client application in a subsequent request for the page;

referencing means, responsive to a request for a selected page, for referencing the data structure of Internet Protocol translations to determine whether an Internet Protocol address for a selected page is present;

requesting means for requesting the selected page using an Internet Protocol address returned from the data structure of Internet Protocol

translations if the Internet Protocol address is present in the data structure of Internet Protocol translations; and

querying means for querying the remote domain name server to determine the Internet Protocol address of the selected page if the Internet Protocol address for the selected page cannot be determined from referencing the data structure of Internet Protocol translations.

7. The data processing system of claim 6, further comprising:

indicating means for indicating in the data structure of Internet Protocol translations that certain Internet Protocol addresses of specified domain names are to be excluded from the database.

8. The data processing system of claim 6 or claim 7, further comprising:

performing means for performing the looking and using steps only if a selected user input is received.

9. The data processing system of any of claims 6 to 8, further comprising:

updating means for updating the data structure of Internet Protocol translations, by a client application, if a given Internet Protocol address for a given domain name has changed.

10. A computer program comprising computer program code to, when loaded into a computer system and executed thereon, cause the said computer system to perform all the steps of a method according to any of claims 1 to 5.