(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
18 May 2007 (18.05.2007)

PCT

(10) International Publication Number
WO 2007/056336 A1

(51) International Patent Classification:
G06F 17/30 (2006.01) *G06F 11/14* (2006.01)

(21) International Application Number:
PCT/US2006/043278

(22) International Filing Date:
7 November 2006 (07.11.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/270,188 9 November 2005 (09.11.2005) US

(71) Applicant *(for all designated States except US)*: **COMPUTER ASSOCIATES THINK, INC.** [US/US]; One Computer Associates Plaza, Islandia, NY 11749 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **HARVEY, Richard, Hans** [AU/AU]; 4 Odette Court, Ringwood East, VIC 3135 (AU). **GROENVELD, Damon, Nicholaas** [AU/AU]; 261 O'Grady's Road, Pheasant Creek, VIC 3757 (AU).

(74) Agent: **PEDERSEN, Luke, K.**; Baker Botts L.L.P., Suite 600, 2001 Ross Avenue, Dallas, TX 75201 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

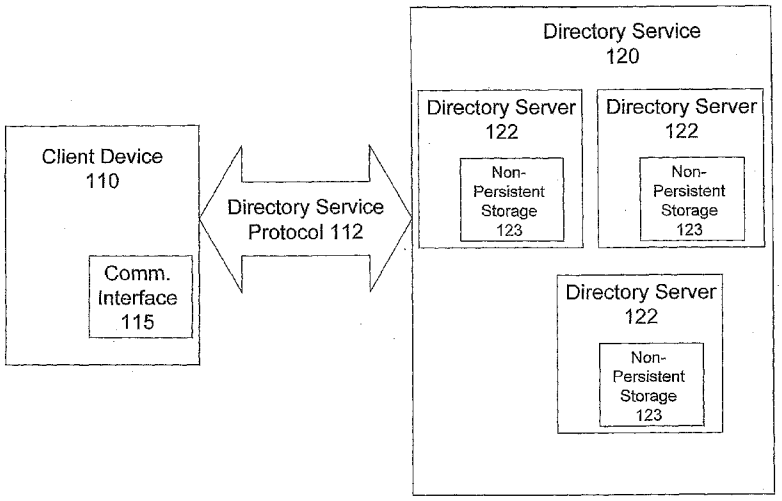
(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:
— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:
— *with international search report*

[Continued on next page]

(54) Title: SYSTEM AND METHOD FOR WRITING DATA TO A DIRECTORY



(57) Abstract: A system and method are provided for processing directory service operations. The system includes a client device communicatively coupled to one or more directory servers. Each directory server includes a communications interface, a storage mechanism, and an evaluation module. According to an exemplary embodiment, the storage mechanism is a non-persistent storage mechanism which increases read and write performance. When a directory server receives a directory service operation, it determines whether it is capable of processing the operation, and if so, processes it with respect to its non-persistent storage mechanism. If the directory service operation is an update operation, the operation is broken up into permanent data and temporary data. The permanent data is written to a persistent storage mechanism whereas the temporary data is written to a non-persistent storage mechanism. In some embodiments, the permanent data is additionally written to a non-persistent storage mechanism.

WO 2007/056336 A1

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

SYSTEM AND METHOD FOR WRITING DATA TO A DIRECTORY

TECHNICAL FIELD

The invention relates to a system and method for processing directory service operations using non-persistent storage.

BACKGROUND

Directory services are commonly used to manage identities, policies, security information, and network resources. A directory service typically organizes electronic content and runs on a directory server. The directory service is the interface to a directory which holds information about objects that are to be managed by the directory service.

Directories differ from databases in several ways. A primary difference is that information in a directory is generally read more often than it is written. Still, information has to at times be written to directories. Additionally, data in a directory is accessed using one or more directory service protocols. A client requesting from a directory need not know which physical device stores the data. There is an ongoing need to increase directory read and write performance without sacrificing consistency and reliability.

Past attempts to increase read and write performance have resulted in increased complexity of operations and increased cost. Previous attempts at increasing read performance have included the use of various caching techniques. However, in a mixed read/write environment, read performance still suffers in order to maintain consistent information. Caching as well as turning off transaction logging has been used in attempts to increase write performance. However, these techniques do not ensure reliability because information may be lost in the event of a power failure or other service interruption.

The inventors have determined that it would be desirable to provide a system and method to increase both read and write performance of existing systems without sacrificing consistency and reliability.

SUMMARY

A system and method are provided for increasing the speed at which data is written in a directory system. The system of the present invention provides a directory server that uses non-persistent storage in order to expedite the process of handling directory service operations. The directory server may include a communications interface, a non-persistent storage mechanism, an evaluation module, and a directory service knowledge module.

The communications interface may be configured to receive one or more directory service operations from one or more client devices and to provide responses to the client device. The evaluation module may be configured to determine whether the directory server is equipped to perform a received directory service operation and to perform the operation.

If the directory is capable of performing the directory service operation, and if the operation is an update operation (e.g., add, modify, or delete), the update operation is broken up into two types, "permanent" and "temporary," based upon predetermined criteria. This determination is based on the practical reality that some types of "temporary" data do not need to survive a machine failure because they can be either regenerated or are not important (e.g., session information, online status, or any other similar transient data). It could also be the case that the directory system needs to ignore applications that are attempting to update particular attribute types (e.g., the "created by" attribute is not necessary if there is only one application updating a particular part of the directory).

The differentiation between "permanent" and "temporary" may be made according to any predefined criteria. Examples of different criteria which may be utilized in this determination include:

1. The directory "attribute type" (e.g., commonName, lastLoginTime, etc.).
2. The value of an "attribute type" (e.g., eventcounter only written each time it is a factor of 10).
3. User selected criteria (e.g., a time interval).
4. Any combination of the above.

All of the "permanent" data is written through to a persistent storage mechanism (e.g., a hard drive, a file system, etc.). Before, during or after the data has

been written to the persistent storage mechanism, an additional determination may also be made as to whether the "permanent" data should be written to a non-persistent storage mechanism.

"Temporary" data is written through to a non-persistent storage mechanism. The entire contents of the non-persistent storage mechanism, or selected portions, may be immediately or periodically written to the persistent storage mechanism.

The non-persistent storage mechanism may be configured to store data that may be accessed by the client devices.

According to some embodiments, the directory server may be part of a directory service network. As such, the directory service knowledge module may be configured have knowledge of the directory service network and the capabilities of each directory server in the directory service network.

In operation, a directory server may receive a directory service operation from a client device. The directory server may then determine whether it is equipped to evaluate and perform the operation. If it is an update operation, and the directory server is capable of handling the operation, the update operation is broken up into "temporary" and "permanent" data. The permanent data is written to a persistent storage mechanism while the temporary data is written to a non-persistent storage mechanism. An additional determination may also be made as to whether the temporary data should also be written to a persistent storage mechanism.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 illustrates client/server directory service system, in accordance with various embodiments of the invention.

FIG. 2 illustrates a directory server, in accordance with various embodiments of the invention.

FIG. 3A illustrates a process for evaluating a directory service operation, in accordance with various embodiments of the invention.

FIG. 3B illustrates a process for breaking up an operation which includes a write, in accordance with various embodiments of the invention.

FIG. 4 illustrates a directory service network, in accordance with various embodiments of the invention.

FIG. 5 illustrates a directory server knowledge module, in accordance with various embodiments of the invention.

FIG. 6 illustrates a process for mutually updating directory server, in accordance with various embodiments of the invention.

FIG. 7 illustrates a directory service network, in accordance with various embodiments of the invention.

FIG. 8 depicts a plurality of directory service operations, in accordance with various embodiments of the invention.

FIG. 9 depicts a process for selecting a server to respond to a directory service operation, in accordance with various embodiments of the invention.

DETAILED DESCRIPTION

In accordance with various disclosed embodiments, a system and method are provided for efficient directory service performance. Persistent writes to disk or other permanent storage need not be a mandatory part of a directory server. For example, transient data such as a session object created when a user logs into a system, need not be maintained once the user logs out. In another example, tasks created as a part of a workflow application need not be maintained once the task is complete.

In both of the above examples, the data is created as required and it is not necessary that the data pre-exist before being used. Furthermore, the data may be removed once its usage is complete, and the data need not survive a machine failure because the data may be regenerated as needed. Leveraging these concepts, the present invention uses non-persistent information storage to expedite the data read and write processes.

Figure 1 depicts a client/server system 100 for providing efficient directory service access, according to some embodiments of the invention. System 100 may include client 110 communicatively coupled over a network to directory service network 120. Client 110 and directory service network 120 may communicate using one or more directory service protocols 112. Directory service network 120 may include one or more directory servers 122.

Client 110 may be a person, an application, a directory, and/or any network accessible electronic device. For example, client device may be a desktop computer,

a laptop computer, a portable digital assistant, a wireless telephone, and/or other electronic devices. Client 110 may include a directory services communication interface 115 enabling communication with directory service network 120.

Directory service protocols 112 may include any known or proprietary directory service protocols such as, for example, lightweight directory access protocol (LDAP), X.500, directory service markup language (DSML), and/or other directory service protocols. Directory service protocols 112 enable a user at client device 110 to access one or more directory entries stored on one or more of directory servers 122.

Directory services 112 enable clients to locate directory entries and attributes associated with the entries. Each directory server 122 may store directory entries in a hierarchical, tree-like structure. Each directory entry may comprise a collection of attributes having a distinguished name. Additionally, each entry's attributes may have a type and one or more values. The distinguished name may be constructed by taking the name of the entry, known as the relative distinguished name, and concatenating the names of its ancestors in the hierarchy.

Directory service protocols 112 define a plurality of operations for accessing data stored with the directory service network. For example, the protocol may define operations for searching the directory, adding an entry, updating an entry, deleting an entry, and/or other operations. When client device 110 issues a directory service operation, the client need not be aware of which directory server 122 is responsible for evaluating the operation.

According to an exemplary embodiment of the invention, one or more directory servers 122 may include non-persistent storage mechanism 123. In a typical client/server directory service arrangement, a persistent storage mechanism, such as a database, file system, and/or other permanent storage device may be used for evaluating directory service operations. As used herein, persistent storage refers to permanent, static, long-term storage mechanisms. Persistent data is used herein to refer to data that is read from and/or written to a persistent storage mechanism. Each time a read or write directory operation is performed, the responding directory server consults the persistent storage device to either retrieve an object or write an object to the storage device. However, as described above, many directory service operation do not require permanent storage.

Referring now to Figure 2, a directory server 200 is depicted, in accordance with various embodiments of the invention. Directory server 200 may include a communications interface 210, a non-persistent storage mechanism 220, and evaluation module 230. Additionally, directory server 200 may include directory server knowledge module 240.

Communications interface 210 may be configured to receive directory service operations from one or more client devices. Additionally, communications interface 210 may be configured to forward directory service operations to other directory servers or to receive forwarded directory service operations when connected as part of a network.

Directory server 200 may include non-persistent storage mechanism 220. Non-persistent storage mechanism 220 may be a portion of memory associated with server 200, according to some embodiments of the invention. While depicted in Figure 2 as internal to directory server 200, non-persistent storage mechanism 220 may be an external storage mechanism. Non-persistent storage mechanism 220 may initially be empty, according to some embodiments of the invention. In other embodiments, non-persistent storage mechanism 220 may be initialized with information stored in other persistent or non-persistent storage mechanisms. Updates to the data, such as modifications, additions, and/or deletions, may be stored in non-persistent storage mechanism 220. Additionally, read operations may be evaluated with respect to the data stored in non-persistent storage mechanism 220.

According to some embodiments of the invention, non-persistent storage mechanism 220 may be an alternate evaluator. The alternate evaluator may be configured to cache an entire directory store. The directory store may be a persistent storage mechanism. Both read and write operations may be evaluated in relation to the alternate evaluator. Data may be read from the alternate evaluator and operations that add or modify data may be written to the alternate evaluator.

Evaluation module 230 may be configured to determine whether directory server 200 is equipped to perform a received directory service operation. For example, directory server 200 may not be equipped to perform a read operation if its accessible storage mechanism(s) is empty. According to some embodiments of the invention, evaluation module 230 may consider only the contents of non-persistent

storage mechanism 220 in determining whether directory server 200 may perform a received directory service operation. In other embodiments, evaluation module 230 may consider non-persistent storage mechanism 220 in addition to other storage mechanisms. Additionally, evaluation module 230 may be configured to perform the requested operation.

If directory server 200 is capable of performing the directory service operation, and if the operation is an update operation (e.g., add, modify, or delete), evaluation module 230 breaks up the update operation into two parts, permanent data and temporary data. If the data type is permanent, evaluation module 230 will write the data to a permanent storage mechanism. In some instances, the permanent data may additionally be written to a persistent storage mechanism. If the data type is temporary, evaluation module will write the data to non-persistent storage-mechanism 220.

Directory server knowledge module 240 may be configured to query other directory servers within a directory service network to resolve an operation that evaluation module 230 has determined may not be performed by directory server 200.

Figure 3A depicts a process 300 for receiving and evaluating a directory service operation, according to various embodiments of the invention. As depicted at 302, a directory server may receive a directory service operation. The directory server may determine what type of operation it has received. For example, the operation may be an add, modify, search, remove, or other directory service operation.

The directory server may then determine whether the received operation depends on existing information, as depicted at 304. For example, to perform a search operation, the server must have access to an information storage mechanism that already includes data. For an add operation, the server need not have access to a storage mechanism that contains data.

If the operation does depend on existing information, the directory server may then determine whether it is able to process the operation, as depicted at 306. As described above, a directory server may be configured to evaluate a directory service operation solely with respect to its associated non-persistent storage mechanism or to additionally evaluate an operation with respect to other storage mechanisms. Thus,

the directory server may determine whether it has access to the stored data. According to some embodiments of the invention, a non-persistent storage mechanism may be loaded or pre-configured with information from persistent or non-persistent storage mechanism, or other directory servers after which the directory server may commence evaluation of operations. In other embodiments, data may be loaded into the non-persistent storage mechanism via add operations, after which more complex operations such as search, compare, read, list, modify, rename, delete, and/or other operations, may be performed.

If the directory server is unable to process the directory service operation, it may, according to some embodiments, query another server to process the request, as depicted at 308. Alternatively, the server may respond with a message indicating it is unable to complete the request, as depicted at 310.

As depicted at 312, the directory server may perform the directory service operation. According to some embodiments of the invention, the server may store the result of the directory service operation to its non-persistent storage mechanism. A directory server, according to some embodiments of the invention, may include or interface to a persistent storage mechanism in addition to a non-persistent storage mechanism. As such, certain predefined categories of data may be written to the persistent storage mechanism. For example, configuration data may be stored in the persistent storage mechanism and may be retrieved and/or updated as required. Additionally, certain types of data may be stored and/or pre-loaded in the persistent storage mechanism and may be used during evaluation of a particular operation.

In some embodiments of the invention, the directory server may additionally evaluate an operation 322 using process 320 depicted in Figure 3B. The steps of Figure 3B may be performed by process 320, or by another suitable device. Operation 322 is a directory update operation, such as add-entry, remove-entry, modify-entry, modify-DN, or remove-entry. It may also be an inquiry operation that results in an update, such as a compare of a password attribute or a bind operation, because this can affect operational attributes.

A test 324 checks for any attributes in operation 322 which have not yet been processed. If there are attributes to be tested, path 326 is followed and the attribute

may be tested repeatedly, for example at test 328 and test 330. The number of order of tests may vary as required, dependent on the particular implementation.

In this embodiment, test 328 checks to determine whether the attribute is temporary. If the attribute is temporary, it is forwarded along path 332 to be added to a non-persistent store buffer in step 334. It is then forwarded along path 336 back to test 324.

If the attribute is permanent, it is forwarded along path 338 to be added to a persistent store buffer in step 340. It is then forwarded along path 342 to test 330 which checks to determine if the attribute needs to also be stored in a non-persistent storage mechanism. If the attribute does need to be stored in non-persistent storage, path 344 is followed and the attribute is added to a non-persistent store buffer in step 334. Otherwise, path 346 is followed back to test 324.

If there are no more attributes associated with operation 322 to be processed, then path 348 is followed and the contents of the respective buffers are applied as necessary and in any order. The attributes from the non-persistent store buffer (from step 334) are applied to a non-persistent storage mechanism in step 350. Attributes in the persistent store buffer (from step 340) are applied to a persistent storage mechanism in step 352.

The application of the attributes is consistent with the type of operation. For example, an add-entry would add attributes, a remove-entry would delete attributes, etc. Furthermore, the application of the attributes can be applied at any time, not necessarily waiting full completion of the various tests noted above. Additionally, the update steps 350 and 352 can occur in any order or in parallel. The update concludes at 354 after all update operations have been applied.

Figure 4 depicts another example of a directory service network, in accordance with various embodiments of the invention. Directory service network 400 may include a plurality of directory servers 402-408. Directory servers 402-408 may be any type of directory server and may interface to a non-persistent and/or persistent storage mechanism. Directory servers 402-408 may be replicated servers, each providing the same type of information, according to some embodiments of the invention. In other embodiments of the invention, directory servers 402-408 may

form a distributed network, wherein one or more of directory servers 402-408 store certain types or categories of data.

As depicted in Figure 4 by way of example only, directory server 402 includes an internal non-persistent storage mechanism, directory server 404 interfaces to an external non-persistent storage mechanism, directory server 406 includes a non-persistent storage mechanism and additionally interfaces to external persistent storage mechanism 407, and directory server 408 interfaces to external persistent storage mechanism 409. Other server configurations may be used. For example, each server may interface only to non-persistent storage mechanisms.

As described above, each directory server may include an evaluation module and a directory server knowledge module. Figure 5 depicts an exemplary directory server knowledge module, in accordance with various embodiments of the invention. Directory server knowledge module 500 may include one or more sub-modules such, for example, server knowledge maintenance module 510, initialization module 520, and mutual updating module 530.

Server knowledge maintenance module 510 may be used to maintain a list of all servers in directory service network 400. The list may provide information such as, for example, a name associated with the directory server, a network address associated with the directory server, a location of the server, the server status, the type of data maintained by the server, and/or other information. Information maintenance module 510 may be configured to update information a directory server in the directory service network and to add new servers to the server list. According to some embodiments of the invention, each server maintains a server list. In other embodiments of the invention, a dedicated server may be configured to maintain the server list.

Mutual updating module 530 may be configured to enable one or more servers in a directory service network to share received directory service updates with one or more other servers in the network.

Figure 6 depicts a process 600 for mutually updating a directory server in a directory server network. As depicted at 602, a directory server may receive a directory service update operation. The directory server may then determine whether the update operation should be applied locally to its directory store, as depicted at

603. For example, the receiving directory server may determine whether the update operation refers to the type of data stored in its directory store. If it is determined that the operation should applied locally, the operation is performed as depicted at 606.

The receiving server may determine whether there are other directory servers within the directory service network that should receive the update whether or not the operation was performed locally, as depicted at 604. As described above, a directory server may include a directory server knowledge module which may maintain a knowledge list. The knowledge list may include information about other servers in the network. Determining whether another directory server should receive the update may include, for example, determining whether the other servers store the same type of information, whether the other servers interface to a persistent or non-persistent storage device, and/or other criteria.

If there are no additional servers in the receiving server's knowledge list which should receive the update, the receiving server may simply wait to receive the next operation, as depicted at 610. If there are additional servers in the network directory server list, the receiving server may forward the update to the additional server, as depicted at 608. According to some embodiments of the invention, replication to other servers in the directory server network occurs after confirming the operation with the requesting client device. In other embodiments, replication may occur before confirming the update with the requesting client device. For example, replication to servers which interface to a persistent storage mechanism may be performed after confirming the operation with the client device while replication to servers which interface to a non-persistent storage mechanism may be performed before confirming the operation.

Mutually updating the additional servers may be performed using various synchronization or replication techniques. Table 1 illustrates an exemplary list of techniques that may be used to perform the mutual updating process. Other techniques may be used.

Table 1

Technique	***Description***
Write-Behind (Asynchronous) Replication	Update to another server occurs outside the update transaction
Write-Through (Synchronous) Replication	Update to another server occurs inside the update transaction
Replay-Based Replication	Every Update to one server is applied to another server
State-Based Replication	Final state of one or more changes in conveyed to another server
Meshed Replication	Server is capable of updating another server directly
Cascading Replication	Mutual update propagates through the network from one server to another
Tiered Replication	Mutual update propagates through the network in a hierarchy
Multi-Master Replication	Any server may receive client update operations and replicate them to another server
Preferred-Master Replication	System has an order of preference when it needs to replicate to another server
Master-Slave Replication	Slave servers do not receive client update operation and are only updated via a master server
Full Replication	Each replicated server receives a complete copy of information from another server
Selective Replication	Subset of information is replicated from one server to another
On-Demand Replication	Replication to another server occurs at the time of the update
Batch Replication	Replication to another server occurs at predetermined time or event

Initialization sub-module 520 may be configured to enable a directory server to receive information stored by a second directory server in the network, prior to bringing the server online. In a replicated directory server network, a new directory server may be initialized by any other directory server in the directory service network. In a distributed directory service network, a new directory server may serve as a replica for one or more other servers in the network. As such, the newly added

server may be initialized by the one or more server in the network for which it is a replica.

According to various embodiments of the invention, a routing function may be used to increase the performance of a directory service network. Figure 7 depicts a system 700 for improving the performance of a directory service network. System 700 may include router 710 communicatively coupled to one or more directory servers 720. Additional routers may be included.

Router 710 may include a server knowledge list 712. Server knowledge list 712 may provide a list of servers in the network, and an indication of each server's status and/or other information. Status information may include, for example, an indication of the server's availability, a time associated with the last operation performed by the server, and/or other status information. A server may be unavailable, for example, if the server is offline, recovering from another operation, busy, or otherwise marked unavailable.

Server knowledge list 712 may also include information identifying the type or category of information managed by each server. In a replicated environment, multiple servers may serve the same information, whereas in a distributed environment, each server may serve a specific subset of information. In either environment, a client making a request need not be aware of the server allocation.

According to some embodiments of the invention, router 710 may be a directory server adapted to perform one or more routing functions. For example the router may be adapted to perform functions such as load sharing, fail-over, fail-back, and/or other routing functions. If router 710 is configured for fail-over and directs an operation to a server that fails, the operation can be redirected to another server without interrupting the client application. If router 710 is configured for fail-back and a new server is brought online, the new server is configured and used transparently without the need for client input.

As depicted in Figure 7, each directory server 720 may include a non-persistent information store. One or more persistent storage devices 730 may also be provided. Persistent storage device 730 may interface to any one or more of directory servers 720.

Load-sharing between the one or more directory servers 720 may be based on availability and other pre-determined criteria. For example, router 710 may be configured to route directory service operations to one of directory servers 720 in a round robin fashion wherein each server in turn receives a request. In some embodiments, the directory server having the least number of queued requests may be chosen by router 710. Other methods of choosing a directory server may be used such as, for example, choosing randomly, defining directory server precedence, operation complexity, operation type, security classification, and/or other methods.

Router 710 may be configured to respond to a plurality of events. As depicted in Figure 8, router 710 may be configured to respond to the receipt of a client directory operation request, as depicted at 802, receipt of a disconnect indication, as depicted at 804, receipt of a connection indication, as depicted at 806, and/or other events.

A router receiving a directory operation request may be adapted to forward the request to a directory service that can handle the request. This process, is known as navigation or chaining, and may be based on a unique attribute provided in the directory operation request. For example, deciding where to forward the request may be based on the distinguished name in the request. As described above, the distinguished name is a collection of attributes that make up a directory entry. Using the distinguished name, the router may locate a directory service that manages a namespace that includes the directory name of the target entry.

Figure 9 depicts a process 900 for determining a server which is capable of responding to a directory service operation and forwarding a directory operation request to the determined server. As depicted at 902, a router in a directory service network may receive a directory operation request. Upon receipt of the request, the router may consult its server knowledge list to determine one or more directory servers that may service the request, as depicted at 904. If no servers are available, the router may return an error response to the client.

The directory operation request may include, among other things, a distinguished name associated with the operation. As such, the router may determine which one or more servers manage the provided distinguished name. Other criteria may be used, as would be apparent.

As described above, one or more servers may be capable of responding to a directory operation request. As depicted at 906, a determination is made as to whether more than one server may service the request. The router may consult its knowledge list to determine whether more than one server manages the information identified in the directory operation request. If only one server is identified in the knowledge list, the request is forwarded to the identified server, as depicted at 908. If more than one server may respond to the request, the router may choose a server, as depicted at 910.

The router may determine which of the two or more directory servers to route a directory operation request to based on availability. A server may be unavailable if it is offline, recovering from another operation, busy, or otherwise marked unavailable. In addition to availability, a variety of methods to determine whether a directory operation request should be routed to a server. As described above, this evaluation may be based on one or more of round robin, least busy, multicast, random, preferred server rankings, operation type, operation complexity, security, or any other suitable determination method. For example, if available servers are evaluated using least busy and preferred, the router first determines which server is least busy and then of those which is highest in the preference list. Once a server have been chosen, the directory operation request may be forwarded to the chosen server.

A router in a directory service network may additionally be configured to receive server disconnect indications. According to some embodiments of the invention, a router may receive a disconnect indication due to a server or network failure. Upon receipt of a disconnect indication, the router may mark the server as unavailable in its server availability list. The router may additionally store the time the disconnect indication was received. As such, the router may then send a connection request to the failed server after a pre-defined time-out period has elapsed if the server does not automatically reconnect.

The router may also receive connection indications from one or more directory servers joining the directory service network. When a new server comes online, it may be added to the router's available server list. Furthermore, the router may

receive a connection indication when a previously failed server comes back online. The router may then unmark its failure in the availability list.

While the invention has been described with reference to the certain illustrated embodiments, the words that have been used herein are words of description, rather than words of limitation. Changes may be made, within the purview of the associated claims, without departing from the scope and spirit of the invention in its aspects. Although the invention has been described herein with reference to particular structures, acts, and materials, the invention is not to be limited to the particulars disclosed, but rather can be embodied in a wide variety of forms, some of which may be quite different from those of the disclosed embodiments, and extends to all equivalent structures, acts, and, materials, such as are within the scope of the associated claims.

WHAT IS CLAIMED IS:

1. In a directory system, a method of executing a directory operation which includes an update, the method comprising the steps of:

breaking up said operation which includes an update into permanent data and temporary data; and

performing an operation which includes an update based on whether the data type is permanent or temporary.

2. A method according to Claim 1, wherein said permanent data is written to a persistent storage mechanism.

3. A method according to Claim 2, wherein an additional determination is made as to whether to write said permanent data to a non-persistent storage mechanism.

4. A method according to Claim 1, wherein said temporary data is written to a non-persistent storage mechanism.

5. A method according to Claim 2, wherein said persistent storage mechanism is at least one from the group consisting of a hard drive, a disk, a file system, and a network drive.

6. A method according to Claim 1, wherein said operation which includes an update is broken up according to at least one of the criteria from the group consisting of the directory attribute type, the value of an attribute type, and the attribute is user selected.

7. A method according to Claim 1, wherein said operation which includes an update is at least one from the group consisting of an add-entry operation, a remove-entry operation, a modify-entry operation, a modify-dn operation, a bind operation, a compare operation, and a read operation.

8. A method according to Claim 4, wherein said non-persistent storage mechanism is at least one from the group consisting of an alternate evaluator and a memory-resident table.

9. A method according to Claim 1, wherein said operation which includes an update comprises operations as defined in a standard from the group consisting of X.500, LDAP, and DSML.

10. A directory server for processing directory service operations comprising:

a communications interface configured to receive one or more directory service operations from one or more client devices;

at least one non-persistent storage mechanism;

at least one persistent storage mechanism; and

an evaluation module to breakup an operation which includes an update into temporary data and permanent data.

11. A directory server according to Claim 10, wherein said evaluation module writes said permanent data to said persistent storage mechanism.

12. A directory server according to Claim 11, wherein said evaluation module makes an additional determination as to whether to write said permanent data to said non-persistent storage mechanism.

13. A directory server according to Claim 10, wherein said evaluation module writes said temporary data to said non-persistent storage mechanism.

14. A directory server according to Claim 10, wherein said non-persistent storage mechanism is a internal memory resident table.

15. A directory server according to Claim 10, wherein said non-persistent storage mechanism is an alternate evaluator.

16. A directory server according to Claim 10, wherein said directory operations comprise operations as defined in a standard from the group consisting of X.500, LDAP, and DSML.

17. In a directory system, a method of executing an operation which includes an update, the method comprising the steps of:

breaking up said operation which includes an update into permanent data and temporary data;

writing said permanent data to a persistent storage mechanism; and

writing said temporary data to a non-persistent storage mechanism.

18. A method according to Claim 17, further consisting of the step of:

determining whether said permanent data should be written to said non-persistent storage mechanism; and

writing said permanent data to said non-persistent storage mechanism if said determination is positive.

19. A method according to Claim 17 or 18, wherein the step of breaking up said operation comprises checking attributes in the operation one-by-one.

20. A method according to Claim 19, wherein checking each attribute comprises checking whether the attribute is of a permanent type or a temporary type.

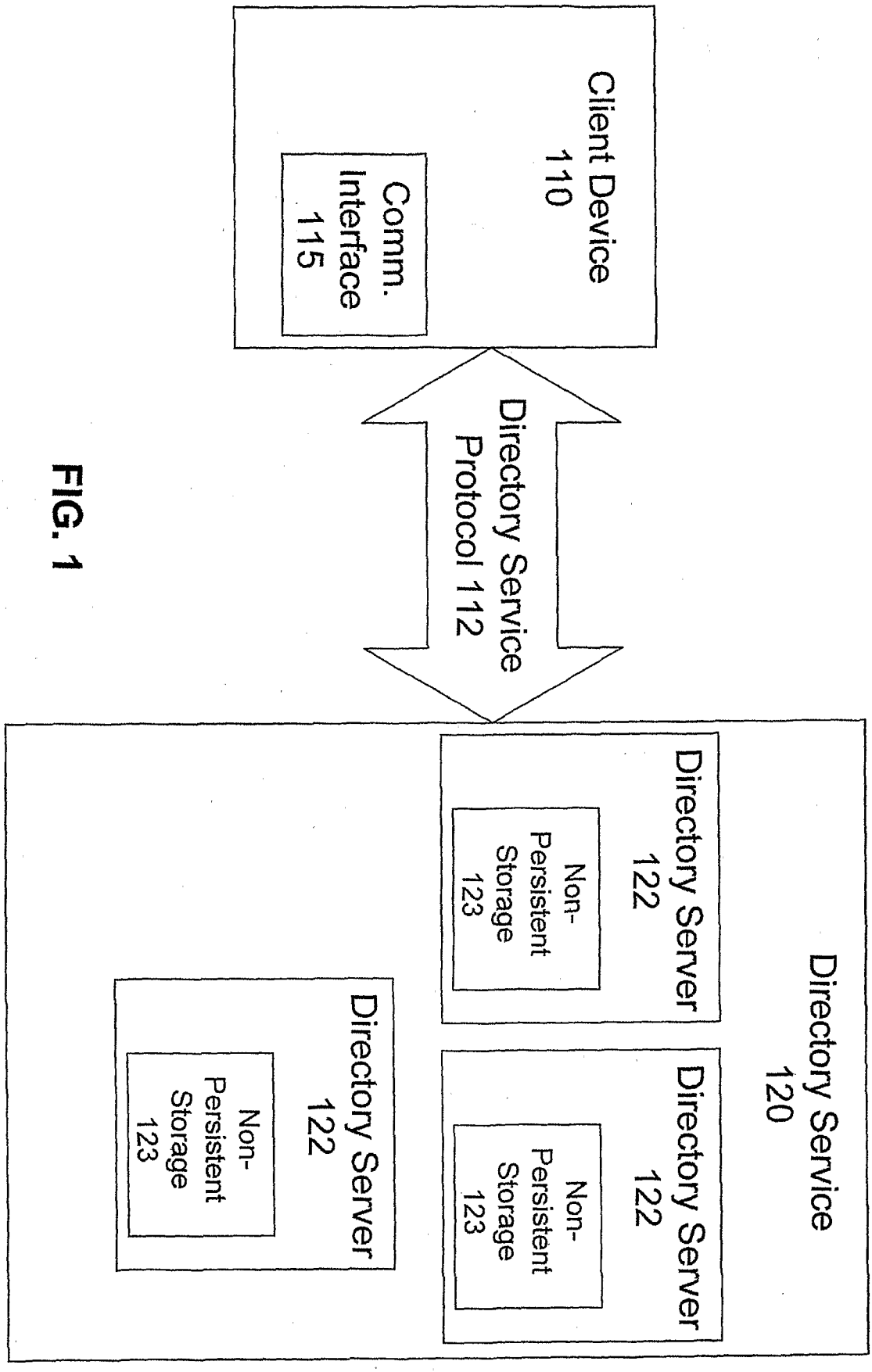

Client Device
110
Comm.
Interface
115
Directory Service
Protocol 112
Directory Service
120
Directory Server
122
Non-
Persistent
Storage
123
Directory Server
122
Non-
Persistent
Storage
123
Directory Server
122
Non-
Persistent
Storage
123

FIG. 1

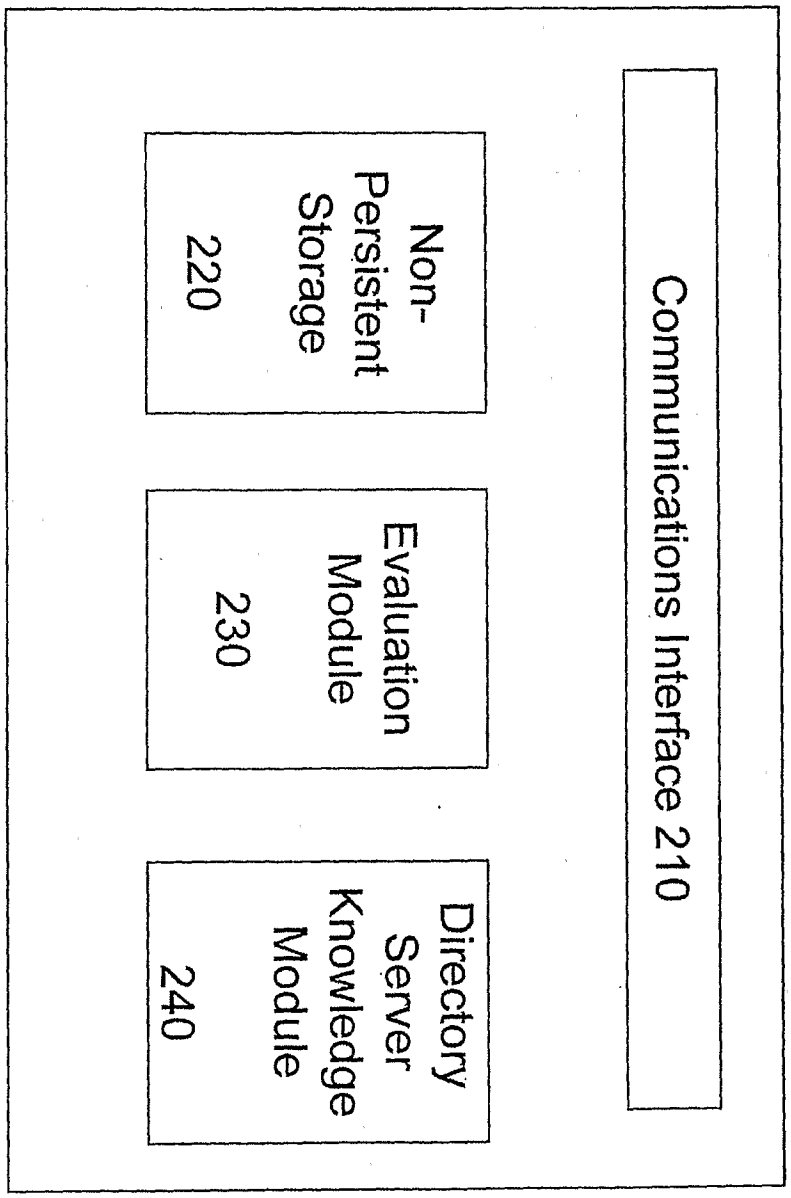

Communications Interface 210
Non-Persistent Storage 220
Evaluation Module 230
Directory Server Knowledge Module 240

FIG. 2

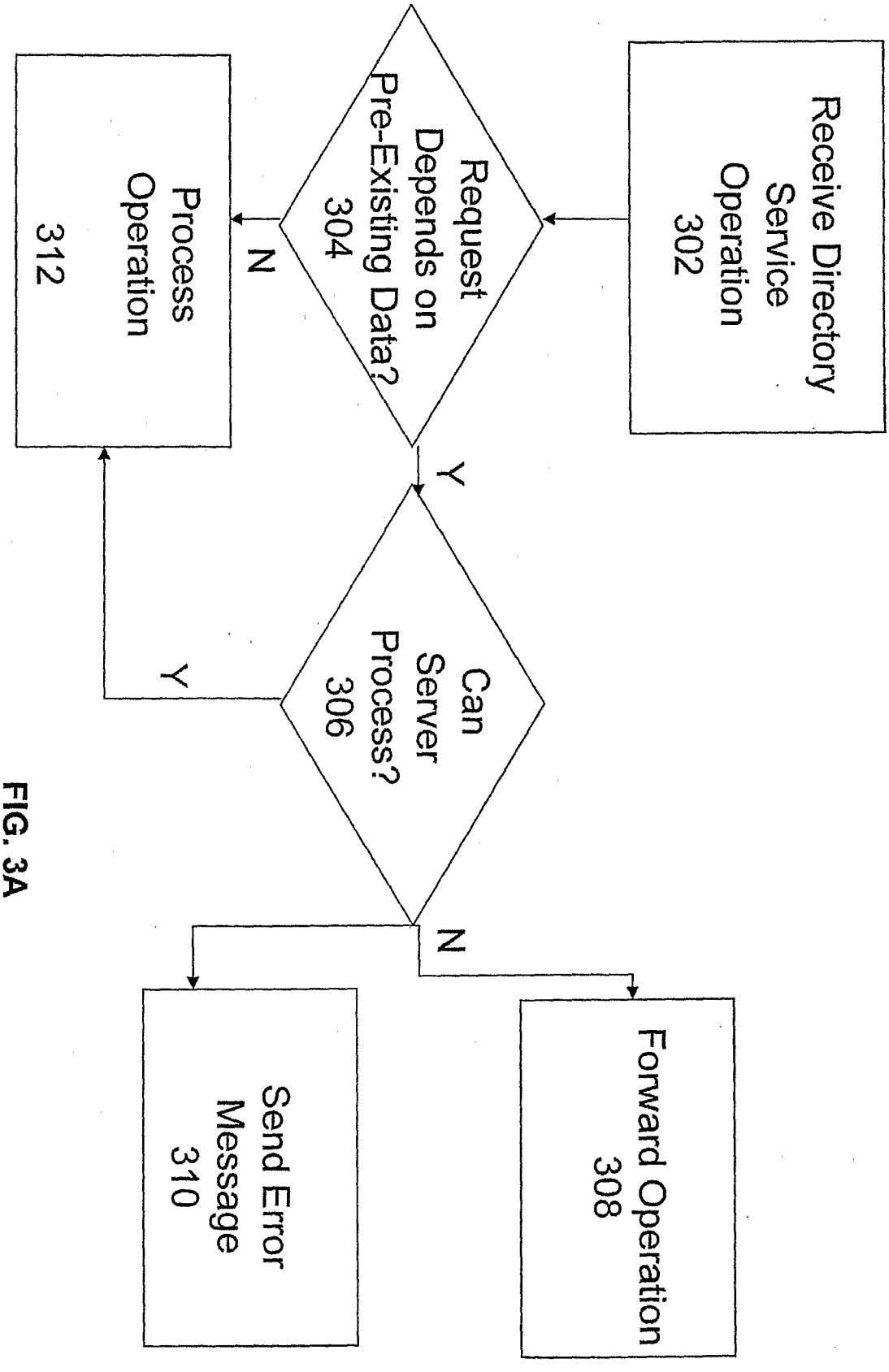

Receive Directory Service Operation 302
Request Depends on Pre-Existing Data? 304
N
Process Operation 312
Y
Can Server Process? 306
Y
N
Send Error Message 310
Forward Operation 308

FIG. 3A

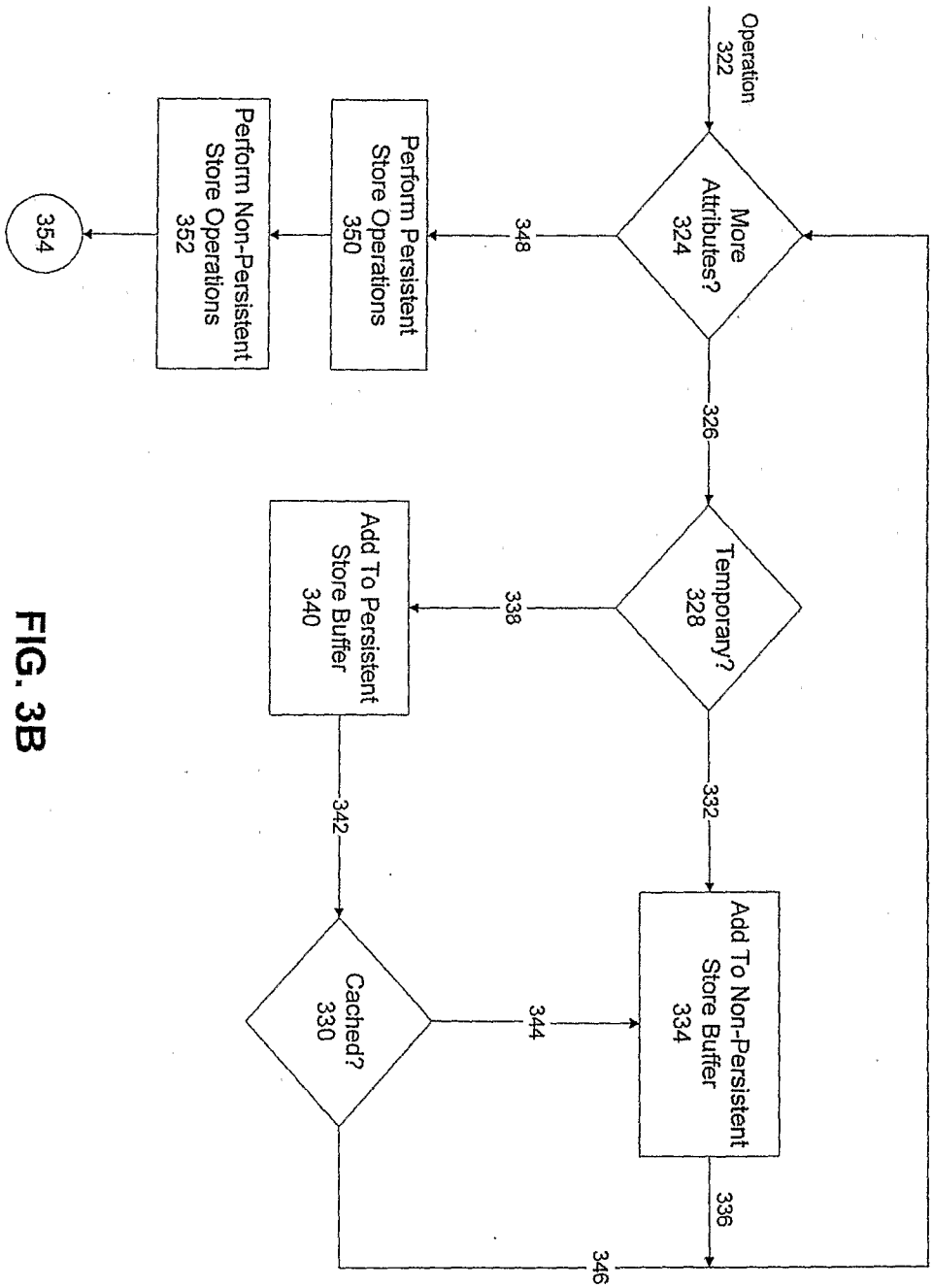

Operation
322
More Attributes?
324
348
Perform Persistent Store Operations
350
Perform Non-Persistent Store Operations
352
354
326
Temporary?
328
338
Add To Persistent Store Buffer
340
342
Cached?
330
344
Add To Non-Persistent Store Buffer
334
336
346

FIG. 3B

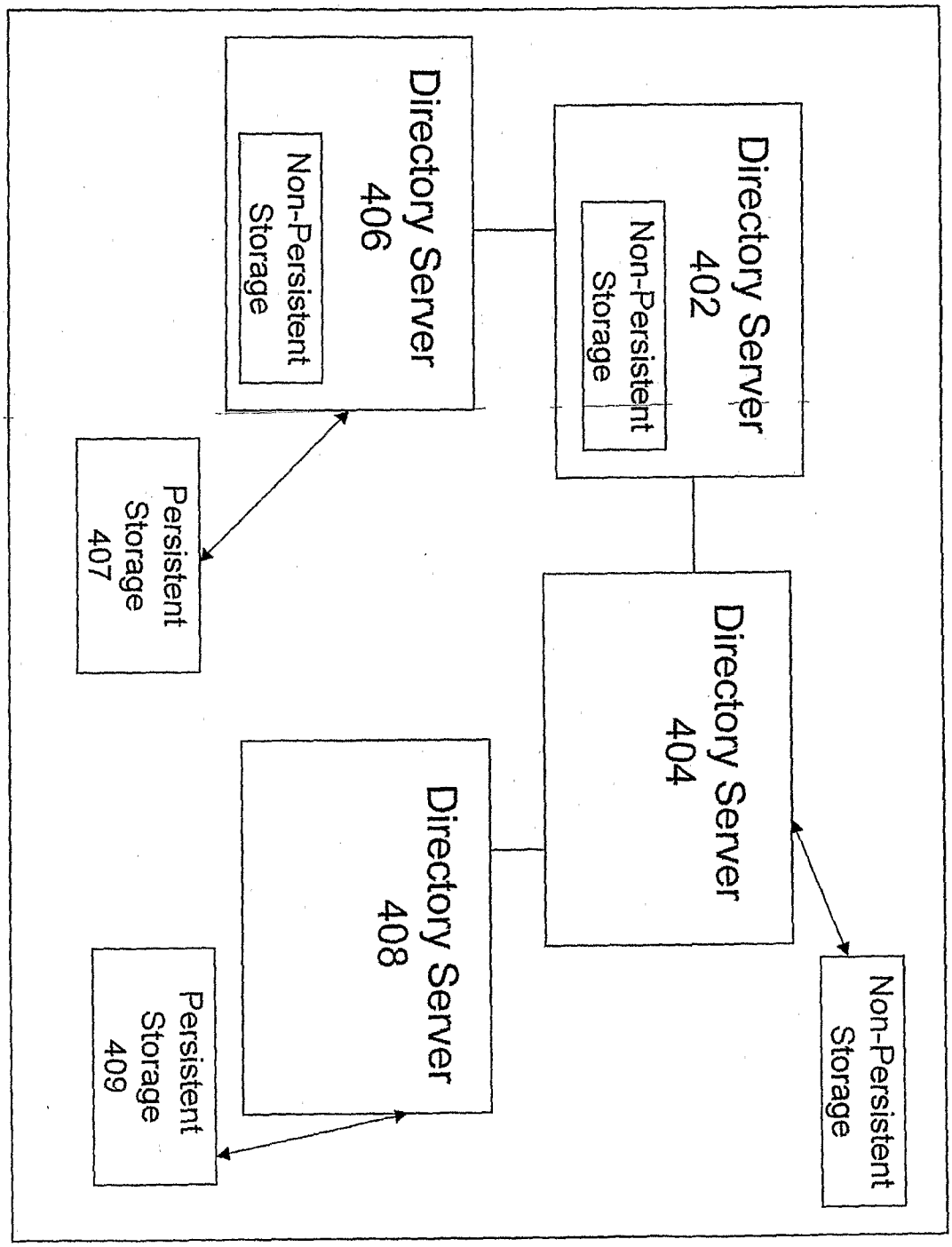

Directory Server
402
Non-Persistent
Storage
Directory Server
406
Non-Persistent
Storage
Persistent
Storage
407
Directory Server
404
Non-Persistent
Storage
Directory Server
408
Persistent
Storage
409

FIG. 4

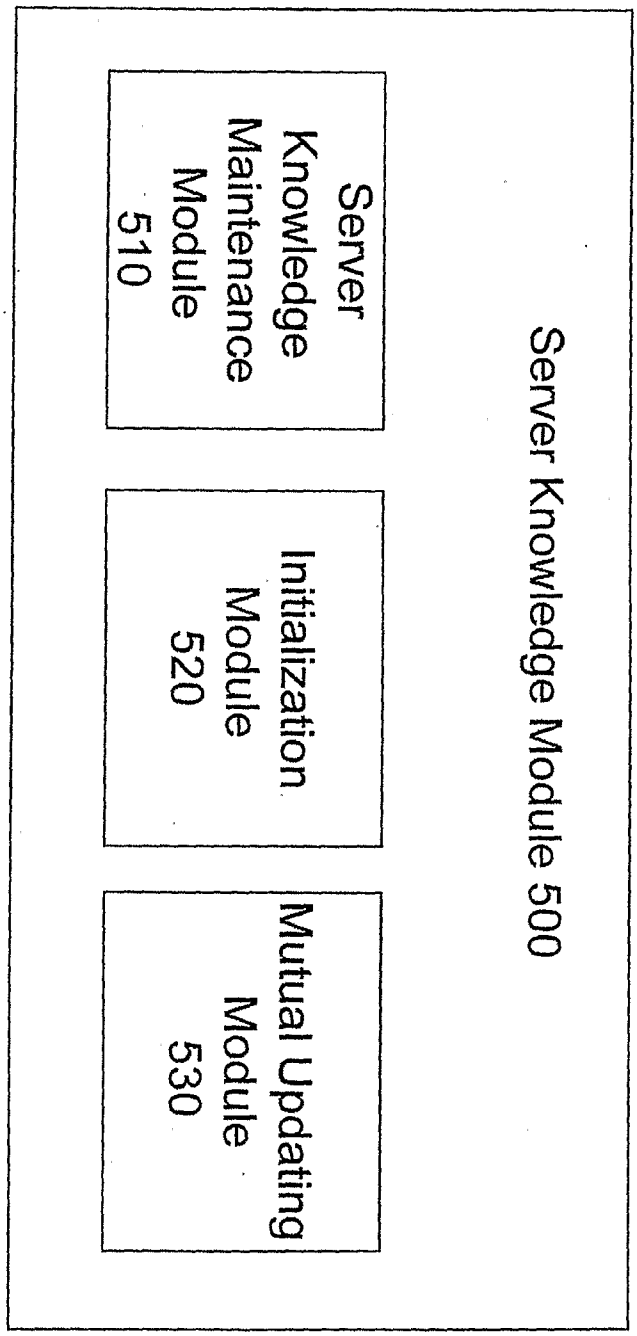

Server Knowledge Module 500
Server Knowledge Maintenance Module 510
Initialization Module 520
Mutual Updating Module 530

FIG. 5

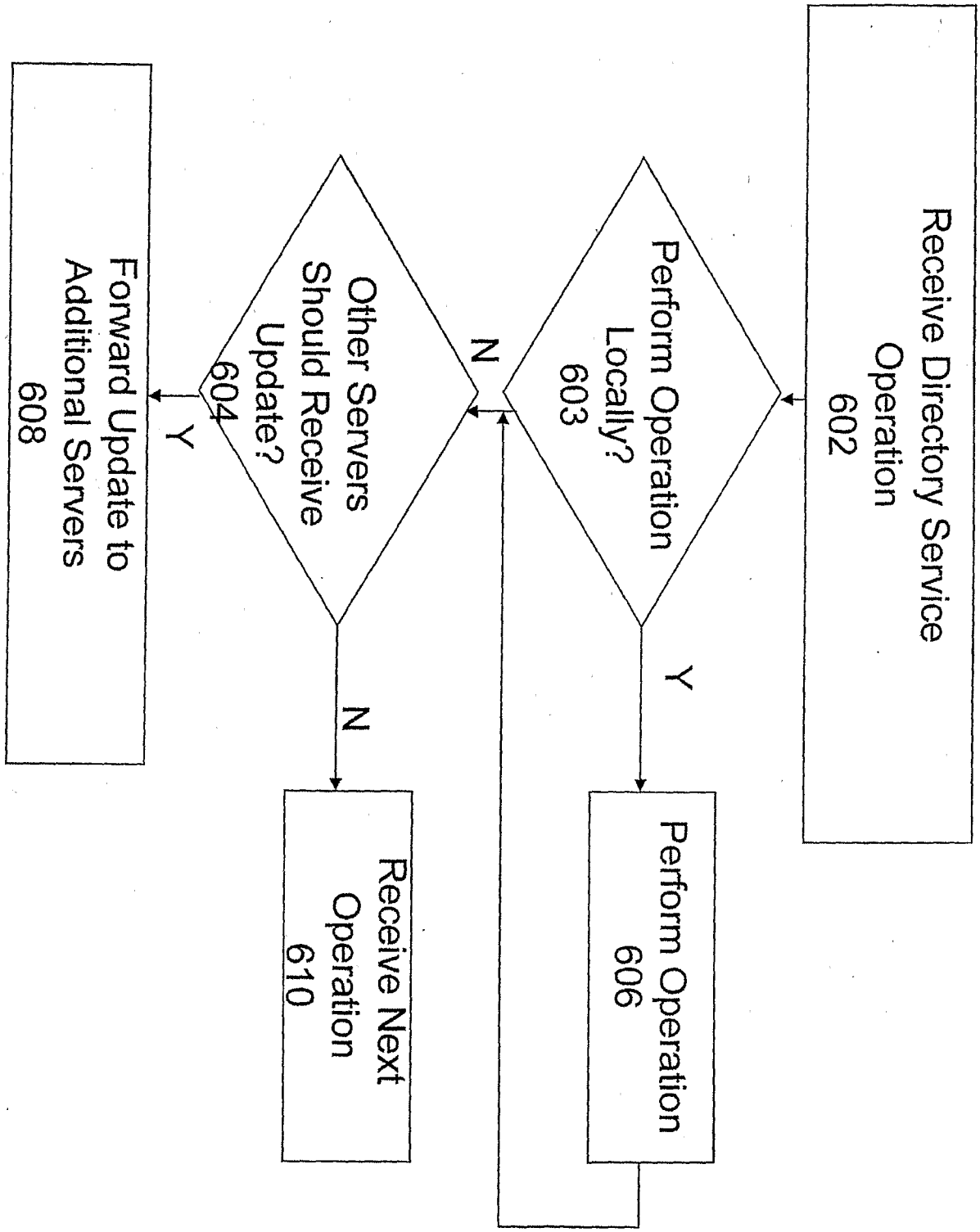

Receive Directory Service Operation 602
Perform Operation Locally? 603
Y
Perform Operation 606
N
Other Servers Should Receive Update? 604
N
Receive Next Operation 610
Y
Forward Update to Additional Servers 608

FIG. 6

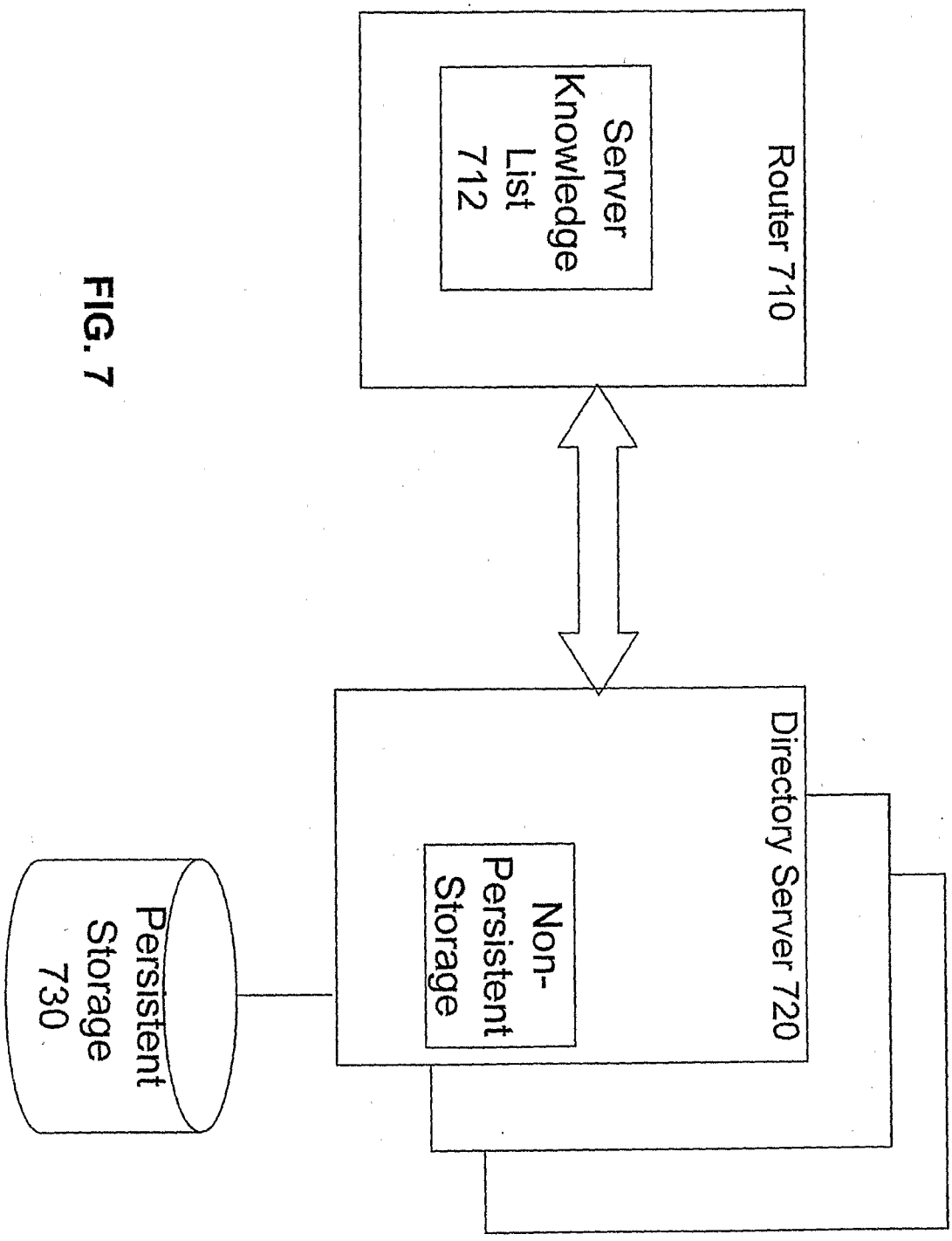

Router 710
Server Knowledge List 712
Directory Server 720
Non-Persistent Storage
Persistent Storage 730

FIG. 7

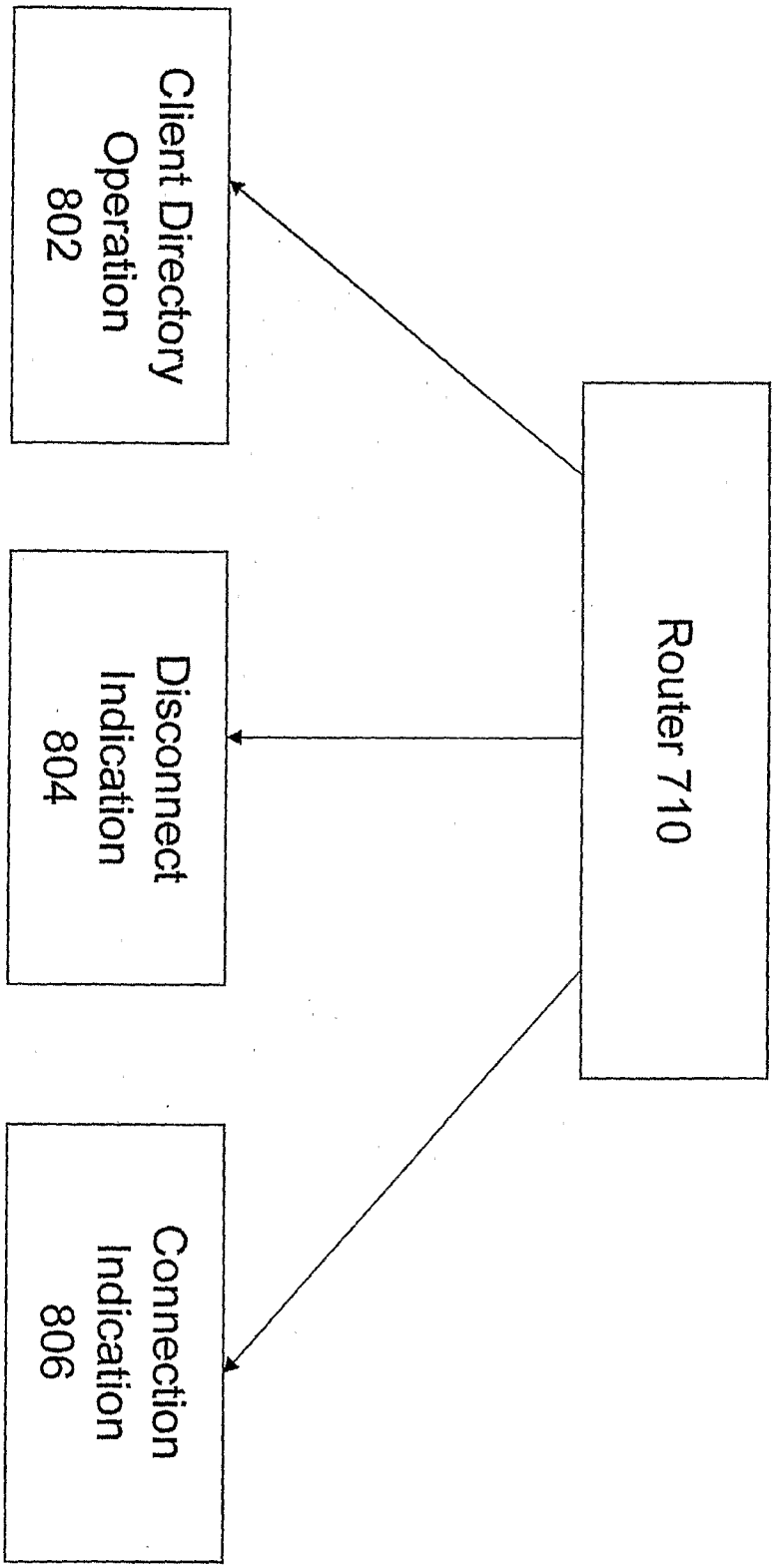

Client Directory Operation 802
Disconnect Indication 804
Connection Indication 806
Router 710

FIG. 8

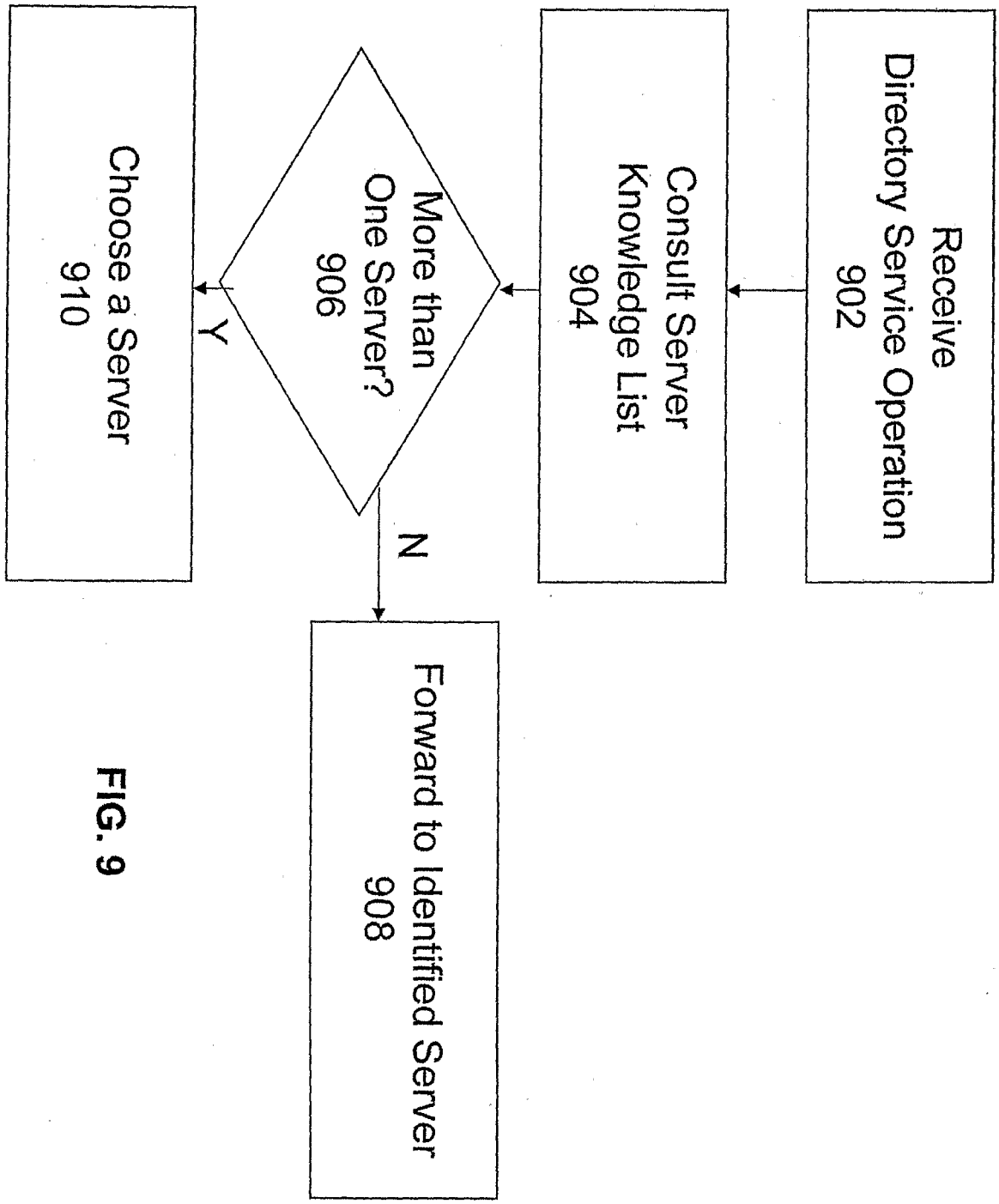

Receive Directory Service Operation 902
Consult Server Knowledge List 904
More than One Server? 906
Y
N
Choose a Server 910
Forward to Identified Server 908

FIG. 9

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2006/043278

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F17/30 G06F11/14

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)
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)
EPO-Internal, WPI Data, INSPEC, IBM-TDB, COMPENDEX

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2004/078623 A1 (TOTOLOS GEORGE [US] TOTOLOS JR GEORGE [US]) 22 April 2004 (2004-04-22) paragraphs [0028], [0032], [0064]; figure 5	1-20
X	JP 63 056891 A (FANUC LTD) 11 March 1988 (1988-03-11) abstract; figure	1-20
X	JP 03 048322 A (MITSUBISHI ELECTRIC CORP) 1 March 1991 (1991-03-01) abstract; figure	1-20
A	JP 04 117518 A (FUJI XEROX CO LTD) 17 April 1992 (1992-04-17) abstract; figure	1,10,17

☐ Further documents are listed in the continuation of Box C.

☒ See patent family 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

Date of the actual completion of the international search	Date of mailing of the international search report
15 March 2007	22/03/2007

Name and mailing address of the ISA/	Authorized officer
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	Deane, Inigo

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2006/043278

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2004078623 A1	22-04-2004	AU 2003282561 A1	04-05-2004
		EP 1552393 A2	13-07-2005
		JP 2006503370 T	26-01-2006
		WO 2004036353 A2	29-04-2004
		US 2006031422 A1	09-02-2006
JP 63056891 A	11-03-1988	NONE	
JP 3048322 A	01-03-1991	NONE	
JP 4117518 A	17-04-1992	JP 3018407 B2	13-03-2000