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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
6 July 2017 (06.07.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/117148 A1

(51) International Patent Classification:
H04W 4/12 (2009.01) *G06F 15/16* (2006.01)
H04W 4/14 (2009.01) *H04M 1/725* (2006.01)
H04W 12/06 (2009.01)

(21) International Application Number:
PCT/US2016/068752

(22) International Filing Date:
27 December 2016 (27.12.2016)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/272,185 29 December 2015 (29.12.2015) US

(71) Applicant: **RADICAL APP LLP** [US/US]; 2233 Barry Avenue, Los Angeles, CA 90064 (US).

(72) Inventors: **KOTIAN, Rohit**; 12411 Fielding Circle, Los Angeles, CA 90094 (US). **OZONIAN, Ryan**; 13365 W. Washington Blvd, Venice, CA 90066 (US). **SHPITALNIK, Igor**; 27906 Agapanthus, Valencia, CA 91354 (US).

(74) Agent: **FITZPATRICK, William**; Fitzpatrick PC, 3419 Via Lido #171, Newport Beach, CA 92663 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, 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, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK,

[Continued on next page]

(54) Title: A SYSTEM, METHOD, AND COMPUTER PROGRAM PRODUCT FOR SECURELY DELIVERING CONTENT BETWEEN STORAGE MEDIUMS

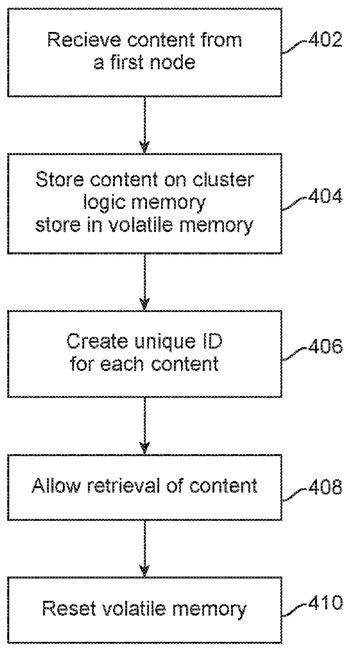
Recieve content from a first node
402
Store content on cluster logic memory store in volatile memory
404
Create unique ID for each content
406
Allow retrieval of content
408
Reset volatile memory
410

FIG. 4

(57) Abstract: A computer program implemented method for securely transferring content sent between clients using a storage media and a computer processor is provided. The method describes receiving the content on a server array from a first node, storing the content on volatile memory on logical memory store, the logical memory store comprising at least two of the servers of the server array, creating a unique identification for the content, allowing retrieval of the content based on pre-determined business logic, the unique identification or both, and resetting the volatile memory on the plurality of servers at a predetermined time after retrieval to permanently erase the content. A system for securely transferring content sent between clients using a storage media and a computer processor is also provided.

WO 2017/117148 A1

SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

NON-PROVISIONAL PATENT APPLICATION

TITLE

A SYSTEM, METHOD, AND COMPUTER PROGRAM PRODUCT FOR SECURELY DELIVERING CONTENT BETWEEN STORAGE MEDIUMS

CROSS- REFERENCE TO RELATED APPLICATION

[0001] The present utility patent application claims the priority benefit of the U.S. provisional application for patent serial number 62/272185 filed on December 29, 2015, A System and Method for Content Delivery.

FIELD OF THE INVENTION

[0002] The present invention relates to content delivery and data security. More particularly, the invention relates to a system, method and computer program product for securely delivering, viewing, or transferring content between storage mediums, and permanently erasing or wiping the content from the storage mediums in an unrecoverable manner.

BACKGROUND OF THE INVENTION

[0003] According to a study by the Poneman Institute, a staggering 43% of companies have experienced a data breach in the past year, costing a mean of $7.6 million dollars per company studied, the highest of which lost upwards of $65 million dollars due to the data breach.

[0004] Most of the companies are experiencing a large percentage of failures in data security, even when they have the most up to date security intelligence systems. This is due to the requirements of physical storage media, such as non-volatile secondary storage primarily used for medium or long-term storage (e.g., hard drives).

[0005] The problems that arise based on these secondary storage types are two-fold. Since the information is stored until erased, criminals can acquire relatively old data that is housed on the drive due to the difficulty in wiping these storage mediums completely clean, and also, in such a manner so that even forensic analysts cannot recover the data. The latter can be particularly troublesome in the context of litigation, when files that were supposed to be deleted are more easily accessible than the user had planned.

[0006] In most organizations that require data security, there are ascending levels of information protection required, depending upon the sensitivity of the information at hand. As an example, instant messages from one low-level employee to another will not require the same level security as CEO emails or the formula to a particular best-selling soda for example. In the data storage universe, there is typically a tiered cost structure based on the sensitivity and level of security required by the client.

[0007] Current devices and methods used for storage utilize volatile storage for less sensitive information (e.g., the instant message chats) including primary storage means such as cache, main memory and flash. However, these primary storage means are costly and the fastest form of storage, not easily or inexpensively scalable, (e.g., usually very small, and managed by the operating system), the contents of main memory are usually lost in a power failure or crash, and in the case of flash memory (EEPROM) writing data is more complicated, and to overwrite what has been written, one has to first erase the entire bank of the memory – which may support only a limited number of erase cycles. As such, current methods of data storage using volatile memory are risky due to the possibility of total data loss, inefficient based on cost and not easily scalable.

[0008] Past ad hoc approaches for wiping non-volatile memory discs clean are tenuous, time consuming, and most importantly, ineffective. For example, in mechanical hard drives, files a user deletes can be recovered by forensic investigators because when the user deletes a file from the drive, the drive simply marks the file's data as deleted. Until it's overwritten in the future, people can scan the drive and recover the marked-as-deleted data. And then, even when it is overwritten, it may not be overwritten in its entirety.

[0009] Other approaches to this problem utilize solid-state drives (SSDs) and a feature called TRIM, in which when a user deletes a file, the operating system informs the drive that the file was deleted. The drive then erases the file's data from its memory cells. However, SSDs cost approximately twice as much as their non-volatile cousins, making scalability for SSDs costly.

[0010] While some critical data must be stored on hard drives, other less critical data can be stored in less expensive means, especially the data that is not required long-term, but only required to reach a recipient once, and then disappear or vanish forever with no ability to be forensically retrieved.

[0011] As an example, in the context of short messaging systems (SMS) and push notifications, it may be desirable for these messages, once received and read, to vanish forever.

[0012] However, with trillions of text messages sent each year, it is exceedingly difficult to forego non-volatile memory, due to the cost and other problems associated with volatile memory, above. However, it is extremely important to some users that their text messages not be recoverable, as they can be used out of context by overzealous attorneys during litigation.

[0013] Accordingly, there is a need for a system and method that stores information for short periods, then fully deletes and wipes this information from the storage medium, while being easily scalable, and inexpensive.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] The present invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

[0015] FIG.1 is a block diagram illustrating an exemplary embodiment of the content delivery platform system according to another embodiment of the present invention;

[0016] FIG.2 is a block diagram illustrating an exemplary embodiment of the content delivery platform system according to another embodiment of the present invention;

[0017] FIG.3 is a block diagram illustrating an exemplary embodiment of the content delivery platform system according to another embodiment of the present invention;

[0018] FIG. 4 is a flowchart illustrating an exemplary process for securely transferring and wiping the content according to an embodiment of the present invention;

[0019] FIG.5 is a flowchart illustrating an exemplary process for securely transferring and wiping the content according to an embodiment of the present invention;

[0020] FIG.6 is a flowchart illustrating an exemplary process for securely transferring and wiping the content from a according to an embodiment of the present invention;

[0021] FIG.7 is a flowchart illustrating an exemplary process for securely transferring and wiping the content according to an embodiment of the present invention;

[0022] Unless otherwise indicated, illustrations in the figures are not necessarily drawn to scale.

SUMMARY OF THE INVENTION

[0023] To achieve the forgoing and other aspects and in accordance with the purpose of the invention, a method, system and computer program product for increasing user value is presented.

[0024] An objective of the invention is to provide a distributed in-memory content delivery platform for sharing a plurality of content between nodes (e.g. clients) over a communication network.

[0025] An objective of the invention is to provide a system and method for content sharing using in-memory design such that shared content cannot be recovered once purged from the system.

[0026] An objective of the invention is to make applications such as messaging or digital file sharing fundamentally safer for their users.

[0027] An objective of the invention is to provide a distributed and secure solution for content sharing between the users.

[0028] An objective of the invention is to delete, permanently and without the chance for recovery, certain data from the distributed in-memory store based on the content deletion request or expiration of the time period, and other business logic.

[0029] An objective of the invention is to securely eliminate unnecessary data to avoid data storage overflow.

[0030] An objective of the invention is to provide greater speed of data flow between a plurality of nodes.

[0031] An objective of the invention is to provide enhanced reliability and redundancy of content data transfer, while at the same time ensuring high data flow speed.

[0032] In embodiments of the present invention, a computer program implemented method for securely transferring content sent between clients using a storage media and a computer processor is provided. The method comprises receiving the content on a server array from a first node, storing the content on volatile memory on cluster logical memory store, the cluster logical memory store comprising at least two of the servers of the server array;, creating a unique identification for the content; allowing retrieval of the content based on predetermined logic, the identification or both, and resetting the volatile memory on the plurality of servers at a predetermined time after retrieval to permanently erase the content.

[0033] In embodiments of the present invention, a computer program implemented system for securing data sent between clients using a storage media and a computer processor is provided. The system comprises a server array capable of receiving data from a node, the server array comprising a storage medium provisioned for a cluster server memory store, memory storing instructions that when executed by at least one processor cause the system to receive the content on a server array from a first node, store the content on volatile memory on cluster logical memory store, the cluster logical memory store comprising at least two of the servers of the server array; create a unique identification for the content, allow retrieval of the content based on predetermined logic, the identification or both, and reset the volatile memory on the plurality of servers at a predetermined time after retrieval to permanently erase the content.

[0034] Other features, advantages, and aspects of the present invention will become more apparent and be more readily understood from the following detailed description, which should be read in conjunction with the accompanying drawings.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0035] The present invention is best understood by reference to the detailed figures and description set forth herein.

[0036] Embodiments of the invention are discussed below with reference to the figures. However, those skilled in the art will readily appreciate that the detailed description given herein with respect to these figures is for explanatory purposes as the invention extends beyond these limited embodiments. For example, it should be appreciated that those skilled in the art will, in light of the teachings of the present invention, recognize a multiplicity of alternate and suitable approaches, depending upon the needs of the particular application, to implement the functionality of any given detail described herein, beyond the particular implementation choices in the following embodiments described and shown. That is, there are numerous modifications and variations of the invention that are too numerous to be listed, but that all fit within the scope of the invention. Also, singular words should be read as plural and vice versa and masculine as feminine and vice versa where appropriate, and alternative embodiments do not necessarily imply that the two are mutually exclusive.

[0037] It is to be further understood that the present invention is not limited to the particular methodology, compounds, materials, manufacturing techniques, uses and applications, described herein, as these may vary. It is also to be understood that the terminology used herein is used for the purpose of describing particular embodiments only, and is not intended to limit the scope of the present invention. It must be noted that as used herein and in the appended claims, the singular forms "a," "an," and "the" include the plural reference unless the context clearly dictates otherwise. Thus, for example, a reference to "an element" is a reference to one or more elements and includes equivalents thereof known to those skilled in the art. Similarly, for another example, a reference to "a step" or "a means" is a reference to one or more steps or means and may include sub-steps and subservient means. All conjunctions used are to be understood in the most inclusive sense possible. Thus, the word "or" should be understood as having the definition of a logical "or" rather than that of a logical "exclusive or" unless the context clearly necessitates otherwise. Structures described herein are to be understood also to refer to functional equivalents of such structures. Language that may be construed to express approximation should be so understood unless the context clearly dictates otherwise.

[0038] Unless defined otherwise, all technical and scientific terms used herein have the same meanings as commonly understood by one of ordinary skill in the art to which this invention belongs. Preferred methods, techniques, devices and materials are described, although any methods, techniques, devices, or materials similar or equivalent to those described herein may be used in the practice or testing of the present invention. Structures described herein are to be understood also to refer to functional equivalents of such structures. The present invention will now be described in detail with reference to embodiments thereof as illustrated in the accompanying drawings.

[0039] Those skilled in the art will readily recognize, in accordance with the teachings of the present invention, that any of the foregoing steps and/or system modules may be suitably replaced, reordered, removed and additional steps and/or system modules may be inserted depending upon the needs of the particular application, and that the systems of the foregoing embodiments may be implemented using any of a wide variety of suitable processes and system modules, and is not limited to any particular computer hardware, software, middleware, firmware, microcode and the like. For any method steps described in the present application that can be carried out on a computing machine, a typical computer system can, when appropriately configured or designed, serve as a computer system in which those aspects of the invention may be embodied.

[0040] While exemplary embodiments of the present invention will be described with reference to certain types of mobile applications, a skilled artisan will realize that embodiments of the invention are applicable to any type of data security or messaging system, or applications in which data storage space is limited.

[0041] As used herein, the terms “delivering” and “transferring” shall comprise the ability for one node to view content on a server, and vice versa. Indeed, the content does not need to be physically transferred to fall under these definitions.

[0042] FIG.1 is a block diagram illustrating an overview of the content delivery system **100** according to the present invention. The content delivery platform **100** comprises node **102**, client **104**, web host **106,** load balancer **108,** server array **110,** memory store **112**, database **114** and network **116**.

[0043] As used herein, "client" and "node" are used interchangeably, with a node representing a client. The node may be client endpoints, endpoint nodes, and the like. In FIG. 1, as referenced, three nodes are shown: Node **102**, client **104**, and web **106** to show a few of the exemplary nodes that may be employed, generally. The nodes **102, 104** and **106** may comprise, but are not limited to, personal computers, workstations, mobile wireless devices, including dedicated wireless devices, smart phones, pagers, PDA, or other similar type of devices capable of processing instructions and receiving and transmitting data to and from other computing devices. The content residing thereon, for example, may include audio, videos, text SMS, messages, multimedia files, images, e-mails, office content, and the like.

[0044] The clients or nodes, as is known, may comprise instructions to execute executable instructions; a virtual machine; a hypervisor; a web browser; a web-based client; a client-server application; a thin-client computing client; an ActiveX control; a Java applet; software related to voice over internet protocol (VoIP) communications like a soft IP telephone; an application for streaming video and/or audio; an application for facilitating real-time-data communications; a HTTP client; a FTP client; an Oscar client; a Telnet client; or any other set of executable instructions.

[0045] Server array **110**, and the exemplary servers **118**, **120**, **122** may be referred to as servers, server farm, or hosts. The servers of the server array **110** are logically grouped together, and can be geographically dispersed, or located proximate to each other, and may be configured to communicate using a WAN, or LAN. In some embodiments, a server array 110 can include servers that execute a substantially similar type of operating system platform (e.g., WINDOWS NT, UNIX, LINUX, or MacOS, or could be a docker container within a VM on an AWS host). In other embodiments, the server array can include a first group of servers that execute a first type of operating system platform, and a second group of servers that execute a second type of operating system platform.

[0046] The server array **110** may comprise servers of different types. Exemplary servers include file servers, an application server, web servers; a proxy servers, and the like. Some embodiments include a first server **118** that receives requests from a client **102**, forwards the request to a second server **120**, and responds to the request generated by the client **102** with a response from the second server **120**.

[0047] In the embodiment shown in FIG. 1, the servers comprise Unix servers in networked communication, as shown by arrows **124**. The servers, in this embodiment, comprise virtual machines **126** (e.g., Java virtual machine (JVM)) and provides for loading, linking, and initialization.

[0048] Each of the servers **118, 120**, and **122** further comprise business logic **128**$^{\text{i-iii}}$. The business logic **128** is configured as a set of predetermined rules for data flow and data deletion requests, determine how data can be created, displayed, stored and changed. Exemplary business logic comprises to which content or data to assign an ID to, for which the servers notify the node/client/web about available content or data, the amount of time content or data is on the system before it is purged in an unrecoverable manner, which content or data entries are to be polled and in which order (e.g., interval polling based on the age of the data), to name a few. Further business logic may comprise which data is sorted and which is permanently deleted, which data receives redundancies on servers, and the like.

[0049] The array of servers **110** also comprise a cluster logical memory (CLM) store **112**. As shown, the CLM store **112**, more broadly, is memory virtualization, or decoupled volatile random access memory (RAM) resources from individual servers **118**, **120**, and **122** which aggregates those resources into a virtualized memory pool (CLM **112**) available to any server in the cluster. In embodiments of the present invention, the memory pool is accessed by the operating system or applications running on top of the operating system **126** and the business logic **128**. The distributed memory pool can then be utilized as a high-speed cache, a messaging layer, or a large, shared memory resource for a CPU or a GPU application. Importantly, in embodiments of the present invention, the business logic instructs for copies of content on a plurality of servers **118**, **120**, and **122** of the server array **110** to provide redundancies and latency in case of a unintended power down or fault so that important data is not lost when the volatile memory is wiped. Consequently, n some embodiments of the present invention, different bits of content and data will be stored on memory store **130**$^{i-iii}$ until they are reassembled for use. These memory stores **130** are also fault tolerant so as to provide redundancy (e.g., dual module redundancy) in case there is a failure of any one or more of the servers in the array of servers, which may happen for a myriad of reasons.

[0050] In this way, the server array **110** and system is further configured to persist the content and data onto memory stores **130**, which may be any number of stores.

[0051] Referring still to **FIG. 1**, network **116** is configured to transmit data and content from nodes **102**, **104** and **106** to server array **110**. The network **104** may comprise one or more sub-networks, and may comprise a local-area network (LAN), or a wide area network (WAN) and the like. The network topology may differ within different embodiments, possible network topologies include: a bus network topology; a star network topology; a ring network topology; a repeater-based network topology; or a tiered-star network topology.

[0052] Load balancer **110** may be, in some embodiments, a smart load balancer, or in other embodiments, a dumb load balancer. The load balancer **110** is configured to distribute workloads from the nodes across multiple computing resources, such as used in large Internet Relay Chat networks, high-bandwidth File Transfer Protocol sites, Network News Transfer Protocol (NNTP) servers, Domain Name System (DNS) servers, and databases. Known load balancing techniques such as round-robin and DNS delegation may be employed herein.

[0053] Database **114** is non-volatile memory (e.g. hard disk memory) configured to store user profiles and other information that is considered non-sensitive, such as, but not limited to, name, age, sex, location, areas of interest, dislikes, email address, hobbies, and occupation. The database **114** is in communication with but partitioned from the server array **110**, and is configured such that the data can be received on the server array **110**, and associated with the content or data that resides on the memory store **112** from each of the nodes **102**, **104** and **106**.

[0054] The term “in-memory database,” or “IMDB,” is used herein to refer broadly and inclusively to any database management system that primarily relies on main memory, rather than a disk-based mechanism, to store and manage data, such as memory store **112**. The memory can be, for example and without limitation, semiconductor memory such as random-access memory (RAM), high-speed volatile memory, or other similar type of memory or storage device. The use of a distributed in-memory database as described herein is intended to provide improved performance and scalability, and security.

[0055] In operation, content or data is sent from a first node, in this example, node **102**, which may be a smartphone. The content is distributed to the server array **110** via load balancer **108**, which optimizes resource use, maximizes throughput, minimize response time, and avoids overload of any single resource. When the content is received at the server array **110**, it is persisted and distributed to the pool of memory stores **130**$^{i\text{-}iii}$. As an example, content may reside on servers **118** and **120,** and also **122**. Further, portions of the content may reside on one or multiple servers. Due to the unique IDs generated, the servers can locate the content in whole or in part when it is ready to be consumed, viewed, or otherwise processed once pulled from memory store **112.** The business logic is configured to run rules, coordinate the data, process commands, make logical decisions and evaluations, and in some embodiments, perform calculations based on received data.

[0056] The servers **112** and business logic **128** are further configured to create a unique identification that is used to identify the message and ensure it is associated with the correct and intended user.

[0057] The servers are further configured to ncrypt the content or data using an encryption system, for example, a key that services to encrypt the content and ensure the content is only consumed by the intended node. Exemplary embodiments of encryption systems comprise public-key cryptography that utilizes asymmetric key algorithms in addition to or in place of traditional symmetric key algorithms, but other keys and encryption system may be used as well.

[0058] In operation, and still with reference to FIG. 1, the server array **112** is configured to generate a notification and send the notification to client **106** that there is content that is ready to be viewed or consumed. Importantly, this notification relies on information received from database **114**, such that the server associates the content with the proper node (e.g., web client **106**). Optionally, the content may be requested by web client **106** for utilization. In other embodiments, it is automatically sent once the node is confirmed as the appropriate receipting. Should the request be authorized by the ID/key and business logic **128**, the content is viewable on the client **106**. Once the content is utilized and consumed by client **106**, the business logic **128** may execute a resetting step. The resetting step is configured to permanently delete the content or data from the volatile memory across the cluster, such that even forensics are unable to retrieve bits or portions of it. In embodiments of the present invention, the resetting step may comprise a very brief power down, or an executed operation. Powering down the servers comprises setting the logic circuit gates to power off and powering up comprises setting the gates back to power on.

[0059] Referring now to **FIG. 2**, a block diagram of a secured messaging system is shown generally at **200** together with step-wise diagram inculcated therein. The content delivery system **200** comprises smartphones **202** and **204**, server cloud **206** and the servers that reside therein **208**, **210** and **212n**.

[0060] In operation, a message is sent from the first phone **202**, and resides on server **208**. The server **208** creates a message ID, step **214**, and stores the messages on server array **212n** in a fault tolerant manner. The message is then persisted, in part or in whole, and distributed to the pool servers **212n**, step **216**, each of which are in communication with the other. The message resides on volatile memory in each of the servers **212n**.

[0061] Business logic, which resides on servers, reviews, evaluates and coordinates the message, then ensures it is stored for retrieval at some predetermined interval. Should the message reside on the servers for over a predetermined time, a reset function may be employed to permanently erase the message as described with reference to **FIG. 1**.

[0062] The servers then generate a notification step **218**, in this case a message letting smartphone **204** know that a message is available for viewing. Importantly, this notification relies on information received from database, the ID, and/or the encryption keys, such that the server associates the content with the proper smartphone.

[0063] The smartphone **204** can then request to view the message, and should it be authorized by the ID/key and business logic **128**, the message is viewable on the smartphone **204**, step **220**. Once the message is viewed, after a predetermined time has passed, the servers execute a resetting step to permanently delete the message from the servers **208**, **210**, and **212**.

[0064] In an optional embodiment, referring now to **FIG. 3**, a block diagram depicting an exemplary client/server system for securely transferring data is shown generally at **300** in which management tasks (including responding to client requests and enforcing content expiration) distributed over multiple servers. Such distribution enables the invention to be scaled to a massive volume of management tasks with higher fault tolerance.

[0065] The server array **302** comprises of one or more management servers **304**, **306**, **308**. The servers may be running on heterogeneous systems, for example, varying in processing power. Each server can in principle handle any functionality previously described, though in practice, a system administrator may improve computational cost efficiency by specializing certain servers to handle various subsets of full functionality. In preferred embodiments, all management servers synchronize their clock times with authoritative time sources.

[0066] The distribution layer **310** may be any suitable load-distribution system, including many commercially available solutions. Accordingly, the distribution layer **310** may itself comprise multiple devices cooperating to perform the functions described, so as to also support massive scale with fault tolerance.

[0067] In embodiments of the present invention, one function of the distribution layer **310,** is to ensure that at least one of the servers within the server array **302** is enforcing expiration of content. In one embodiment, a plurality of management servers may contend to delete the same expired content, which has no ill effect other than redundant computation. In an alternative embodiment, the servers may cooperate. For example, each server may briefly lock the metadata of expired items that it is actively processing for deletion. Alternatively, the servers can select exclusive time slices in which they manage expiry or prioritize which expired items they act upon first, utilizing an ordinal number that is determined and received from the Distribution Layer **310** or some other well-known arbitration mechanism.

[0068] In embodiments of the present invention, the distribution layer **310** is further configured to proxy requests from clients **302**. Requests originating from a Client **312** (e.g. to store, retrieve, and delete content) are sent to the distribution layer **310**, which in turn selects a server from server array **302** to process the request. The selected server will in turn fulfill the request as previously described, by contacting the metadata partition **314** and object partition **316**. The servers **304**, **306**, **308** response is then returned to the distribution layer **310** for passing back on to the client **312** that originated the request.

[0069] In selecting a management server to handle a client request, the distribution layer **310** may use a variety of methods, as is well-known to those skilled in the art, such as round-robin or a dynamic hash table. The distribution layer **310** may have a provision to exclude selecting servers that are still processing previous requests but have been marked by a system administrator as unavailable for further incoming requests. When one or more servers are configured to handle some subset of the full management functionality, the distribution layer **310** must be configured to select only from management servers that are eligible to fulfill the type of request being proxied.

[0070] The distribution layer **310** may use real-time indications of capacity available for each management server in order to avoid overloading any given server. If the distribution layer **310** observes limited capacity in the face of an increasing volume of requests, it may be able to cause deployment additional servers in the server array **302**, or conversely spin down capacity when no longer needed.

[0071] The distribution layer **310** may perform any generic request and response processing functionality that could also be performed by individual management servers. For example, the distribution layer **310** may be similarly capable of logging, buffering, and enforcing timeouts on requests and responses. For client requests, the distribution layer C02 may also perform URI rewrites and other transformations, use client IP address range to restrict or limit the rate of requests, or judiciously retry server selection and proxying the client request when the initial server selected fails to fulfill the request.

[0072] Referring now to FIG. 4, a flowchart illustrating a simplified method for securely wiping data from a server, which is shown generally at **400**.

[0073] At step **402**, the server array receives content from a node.

[0074] At step **404**, the content is stored on volatile memory on cluster logical memory store residing on the server, the cluster logical memory store comprising at least two of the servers of the server array.

[0075] At step **406**, the server creates a unique identification for the content.

[0076] At step **408**, the server allows for retrieval of the content based on predetermined logic, the identification or both.

[0077] At step **410**, the server resets the volatile memory on the plurality of servers at a predetermined time after retrieval of the content to permanently erase the content.

[0078] Referring now to **FIG. 5** a flowchart illustrating a method for viewing and securely wiping data from a server is shown generally at **500**.

[0079] Initially, at step **502**, the operator, who is the manager of the platform, sets business logic, step **502** (i.e., "set bus logic"). The business logic is configured as a set of predetermined rules for data flow and data deletion requests, determine how data can be created, displayed, stored and changed. Exemplary business logic comprises to which content or data to assign an ID to, to which the server notifies the client about available content or data, the amount of time content or data is on the system before it is purged in an unrecoverable manner, which content or data entries are to be polled and in which order (e.g., interval polling based on the age of the data), to name a few. Further business logic may comprise which data is sorted and which is permanently deleted, which data receives redundancies on servers, and the like.

[0080] At step **504**, the server array is configured to extract data from a non-volatile database. The data may comprise user data, for example, data that is not sensitive in nature and data that is required to be stored on a long-term basis. The server is configured to associate this data with nodes based on unique keys and IDs.

[0081] At step **506**, the server array receives content from a first node. The date may comprise content, or data, the latter being small subsets of content.

[0082] At step **508**, the load is distributed to the servers in the array by a load balancer.

[0083] At step **510**, the content is stored in RAM, in a cluster logical memory store, and is persisted throughout the array.

[0084] At step **512**, the server array creates a unique ID or key for the content, which is associated with the node it is to be retrieved by, the node where it came from, or both.

[0085] At steps **514/516**, the server array generates a notification or receives a request to view the content from a second node, and the server array receives a content request, or generates a content notification allowing retrieval of the content.

[0086] At step **518** the second node receives the content.

[0087] At step **520**, a reset is applied to permanently erase the content from the RAM.

[0088] Referring now to **FIG. 6** a flowchart illustrating a method for sending content from a client to another client and securely wiping data from a server is shown generally at **600**. In this optional embodiment, the “distributed cache” referred to may be viewed as analogous to the memory store of **FIG. 1**, for purposes of these upcoming FIGS. In the example implementation shown in **Fig. 6**, which begins at Step **601**, Client A sends content to the server which is to be delivered to client B.

[0089] In step **602,** a unique ID is assigned to the content by the server. In step **603,** the server notifies the Client B about the content by sending the unique ID to the Client B and awaits response from the Client B.

[0090] At decision point **604**, if Client B receives the notification about the content, the method continues to step **605** in which the message is received by the Client B. At step **606**, the Client B requests the server using the unique ID that the message should be deleted from the distributed cache. The server receives the request and deletes the data from distributed cache using the unique ID in an untraceable and unrecoverable manner in step **611**.

[0091] At decision point **604**, if Client B does not receive the notification about the content, the method continues to step **607** in which the server awaits response from the Client B. If the server does not get any response from the Client B, the method proceeds to step **608.**

[0092] At step **608**, the message is examined at preferably regular intervals to determine if the message should be deleted based on the expiration time associated with the message which is defined by the server or one of the clients. If the expiration time has passed, the message is deleted in an untraceable and unrecoverable way from the distributed cache as in step **611**.

[0093] At step **609,** Client A can also request the server for the deletion of the message. At block **610**, the message is examined preferably at regular intervals to determine if the expiration time has passed or not defined by the server or one of the clients. If the expiration time has passed, the method proceeds to step **611** and the message with ID is deleted from distributed cache in an untraceable and unrecoverable manner.

[0094] The content or message once consumed can be deleted from the distributed cache in variety of ways. For example, if distributed cache is implemented in Random Access Memory (RAM), the stored content may not be refreshed and allowed to discharge on its own or else power can be cut off from particular transistors or capacitors storing the said content. Other ways for deleting the content from cache or RAMs are also within the scope of the invention that allows for deletion of content in an unrecoverable manner.

[0095] **Fig.**7 is a flowchart, illustrating a detailed view in accordance with an example of process steps associated with the present invention, and in accordance with communication between clients, server and distributed cache.

[0096] In the present embodiment of this invention as shown in **Fig.** 7, which begins at step **701**, Client A sends content to the server which is to be delivered to client B. In step **702** server receives the message sent by Client A and in step **703** a unique ID is assigned to the message by the server sent by Client A. The message with ID which was assigned by the server is saved in distributed cache in step **704**. The term distributed cache or “in-memory database,” or “IMDB,” is used herein to refer broadly and inclusively to any database management system that primarily relies on main memory, rather than a disk-based mechanism, to store and manage data. The memory can be, for example and without limitation, a semiconductor memory such as random-access memory (RAM), high-speed volatile memory, or other similar type of memory or storage device. The use of a distributed In-memory database as described herein is intended to provide improved performance and scalability.

[0097] From steps **701** to **704,** the severs are also utilized for receiving content from Client A and assigning unique ID to the content. In step **705,** the server sends the notification to the Client B about the content sent by the Client A, by sending unique ID to the Client B and awaits response from the recipient client as in step **706.** Here the recipient client refers to Client B. From steps **705** to **706** the server notifies the Client B about the message and waiting for the response from the Client B.

[0098] At block **707**, the message is examined preferably at regular intervals to determine if there is any response from the Client B. If the Client B has not responded to the notification of the unique ID, then the message is deleted based on the expiration time associated with the message which is defined by the server or one of the clients. If the expiration time has passed, the message with ID is deleted in an untraceable and unrecoverable way from the distributed cache as in step **720**.

[0099] In step **708**, Client B receives the notification sent by the server or notification module and places a request to the server to send him the message along with ID as in step **709.** The server receives the message request sent to him by the Client B at step **710** and processes the request placed by the Client B by examining the expiration time that is associated with the message as in step **711**. If the expiration time has not passed, the message with ID is retrieved from the distributed cache as in step **712** and the method continues to step **713.**

[00100] In step **713**, the server sends the message to Client B and awaits response from the recipient client as in step **714.** At block **715**, the message is examined preferably at regular intervals to determine if there is any response from the Client B. If the expiration time has passed, the method proceeds to step **720** and the message with ID is deleted from distributed cache in an untraceable and unrecoverable manner.

[00101] From steps **709** to **714,** the processing module as explained in FIG. 1a can be utilized by the server for receiving the content request from the Client B and can retrieve the content from the distributed cache based on the unique ID and sends the content to Client B with unique ID.

[00102] At step **716**, the Client B receives the message from the server sent by the Client A. After receiving the message, Client B may request for the deletion of the message from the server sent to him by Client A in step **717.** The server receives the message deletion request at step **718** and examines the message that it has expired or not based on the expiration time associated with the message as in step **719**. If the expiration time has not passed, the method proceeds to step **720** and the message with ID is deleted from distributed cache in an untraceable and unrecoverable manner.

[00103] From steps **717** to **720,** the content deletion module as explained in Fig. 1a can be utilized by the server for receiving the request from either of the clients or calculating the remaining expiration time associated with the stored message and deleting the message with unique ID from the distributed cache. While the various operations of the deletion method are described in reference to the database system, other devices or systems that can delete the data from RAM in an unrecoverable and untraceable manner may also be employed to perform the deletion.

[00104] In an embodiment of the present invention the message with ID which was assigned by the server is saved in distributed cache that primarily relies on random-access memory (RAM). There exists multiple ways to delete the content stored on RAM in unrecoverable manner. For example, the distributed cache or "in-memory database" is not refreshed and starts discharging on its own for deleting the content. The other operations of deletion can include power cut off from the particular transistors or capacitors that are embedded on the integrated circuit (IC) or RAM. Other ways for deleting the content from cache or RAMs are also within the scope of the invention that allows for deletion of content in unrecoverable manner.

[00105] Without limitation, the system and methods described herein can be utilized on platforms for large data sets are stored, but do not or should not be repentantly saved.

[00106] As an example, sensor data from various intelligent platforms, whether it be large manufacturing and assembly lines or oil pipelines, are capable of taking many millions of measurements each day. Many times, only a small percentage of these measurements are valuable, and thus storing them all on non-volatile memory is expensive and inefficient. The present system is configured to sort measurements in its persisted data stores and associate the relevant measurements using tis business logic to the cull the data set and send only relevant data to the non-volatile memory, providing for increased efficiency of storage.

[00107] While the present invention has been described in connection with what are presently considered to be the most practical and preferred embodiments, it is to be understood that the present invention is not limited to these herein disclosed embodiments. Rather, the present invention is intended to cover all of the various modifications and equivalent arrangements included within the spirit and scope of the appended claims.

[00108] Although specific features of various embodiments of the invention may be shown in some drawings and not in others, this is for convenience only. In accordance with the principles of the invention, the feature(s) of one drawing may be combined with any or all of the features in any of the other drawings. The words "including", "comprising", "having", and "with" as used herein are to be interpreted broadly and comprehensively and are not limited to any physical interconnection. Moreover, any embodiments disclosed herein are not to be interpreted as the only possible embodiments. Rather, modifications and other embodiments are intended to be included within the scope of the appended claims.

Claims

We claim:

1. A computer program implemented method for securely transferring content sent between clients using a storage media and a computer processor, the method comprising:

receiving the content on a server array from a first node;

storing the content on volatile memory on a logical memory store, the logical memory store comprising at least two of the servers of the server array;

creating a unique identification for the content;

allowing retrieval of the content based on predetermined business logic, the unique identification, or both; and

resetting the volatile memory on the plurality of servers at a predetermined time after retrieval of the content to permanently erase the content.

2. The method of claim 1, further comprising:

generating a notification that the content is available for retrieval by a second node based on the predetermined business logic, the unique identification or both; and

receiving a content request from the second node.

3. The method of claim 1, wherein the resetting step is in response to the retrieval of the content by the second node, and is defined by the predetermined business logic residing on at least one of the plurality of servers.

4. The method of claim 1, further comprising receiving data from a non-volatile memory in communication with, but partitioned from, the server array, and associating the content with the data received from the non-volatile memory.

5. The method of claim 4, wherein the content stored in the volatile memory comprises messaging data from a user, and the data stored on the non-volatile data comprises information about the user.

6. The method of claim 1, wherein resetting the volatile memory comprises setting the logic circuit gates for at least one of the servers of the server array to off, and then setting the logic circuit gates back to on.

7. The method of claim 1, wherein the volatile memory comprises a virtualization environment configured for:

storing, in a cluster logic shared memory store, the content residing on a plurality of servers;

storing, on each of the plurality of servers, the predetermined business logic; and

proxying a server of the server array in response to a request from the first node, second node, or both;

selecting a server from the server array in the virtualization environment to a second location of the virtualization environment; and

completing the request by contacting a metadata partition, an object partition or both.

8. The method of claim 7, further comprising distributing workloads over the virtualized environment, wherein the workloads comprise a plurality of contents.

9. The method of claim 1, wherein the array of servers are configured to prioritize content and reset of the server array based on the predetermined business logic.

10. A computer program implemented system for securely transferring content sent between clients using a storage media and a computer processor, the system comprising:

a server array capable of receiving content from a node, the server array comprising a storage medium provisioned for a logical server memory store;

a memory for storing instructions that when executed by the at least one processor cause the system to:

receive the content on a server array from a first node;

store the content on volatile memory on a logical memory store, the logical memory store comprising at least two of the servers of the server array;

create a unique identification for the content;

allow retrieval of the content based on predetermined logic, the identification or both; and

reset the volatile memory on the plurality of servers at a predetermined time after retrieval of the content to permanently erase the content.

11. The system of claim 10, further comprising:

generate a notification that the content is available for retrieval by a second node based on the predetermined logic, the identification or both; and

receive a content request from the second node.

12. The system of claim 10, wherein the reset step is in response to the retrieval of the content by the second node, and is defined by predetermined business logic residing on at least one of the plurality of servers.

13. The system of claim 10, wherein the server array is further configured to receive data from a non-volatile memory in communication with, but partitioned from, the server array, and associate the content with the data received from the non-volatile memory.

14. The system of claim 13, wherein the content stored in the volatile memory comprises messaging data from a user, and the non-volatile data comprises information about the user.

15. The system of claim 10, further wherein resetting the volatile memory comprises setting the logic circuit gates for at least one of the servers of the server array to off, and then setting the logic circuit gates back to on.

16. The system of claim 10, wherein the servers of the array of servers comprise a virtualization environment configured to:

store, in a cluster logic shared memory store, the content residing on a plurality of servers;

store, on each of the plurality of servers, the predetermined business logic; and

proxy a server of the server array in response to a request from the node;

select a server from the server array in the virtualization environment to a second location of the virtualization environment; and

complete the request by contacting a metadata partition, an object partition or both.

17. The system of claim 16, further comprising a load balancer in communication with the server array and configured to distribute workloads over the virtualized environment, wherein the workloads comprise a plurality of contents.

18. The system of claim 10, wherein the servers of the server array is in networked communication such the volatile memory on each of the servers comprises and forms the cluster logical memory store.

19. The system of claim 10, wherein a copy of the content is stored on another server of the server array to provide redundancy.

20. The system of claim 10, wherein the array of servers are configured to prioritize content resetting of the server array based on the business logic.

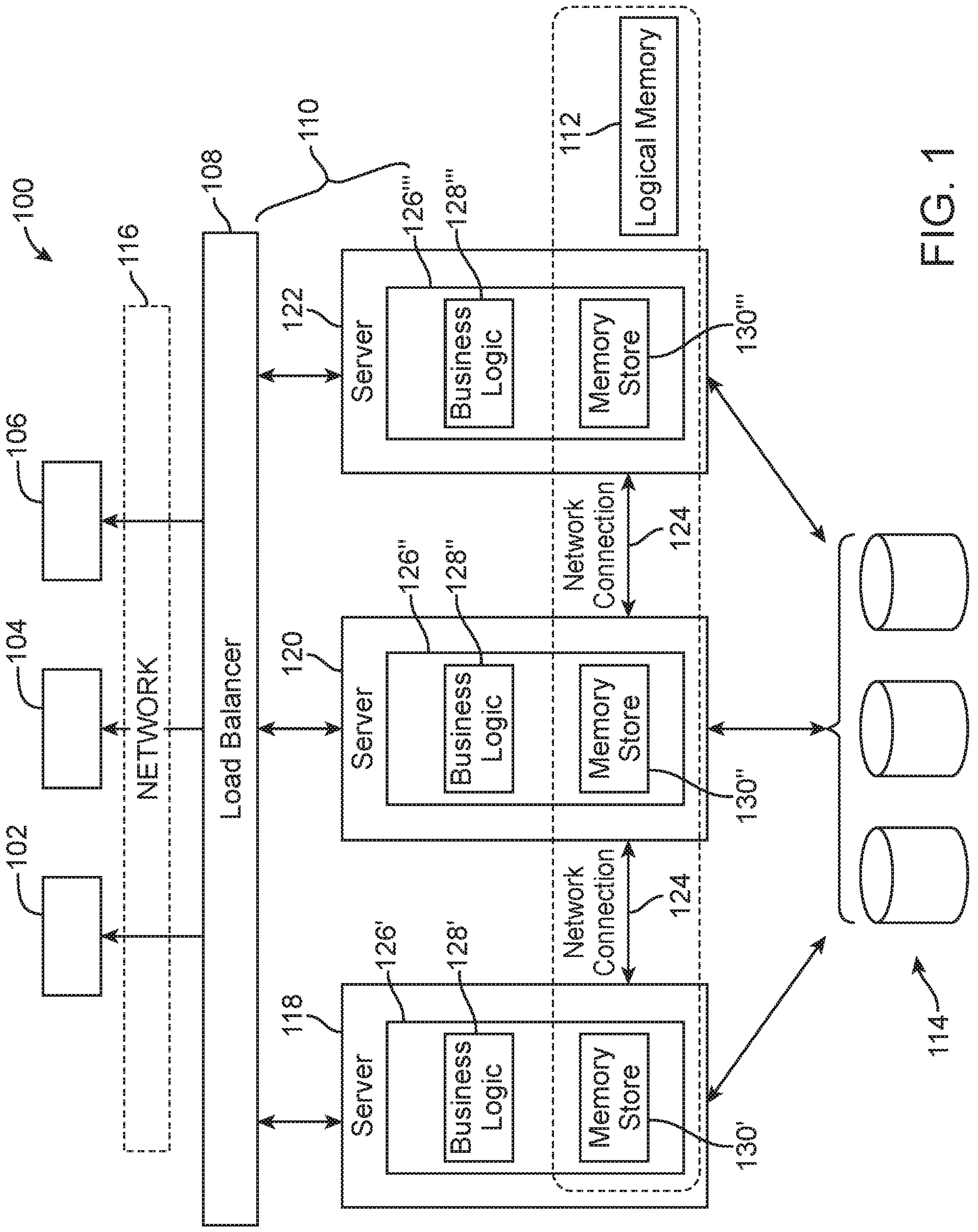
100
102
104
106
116
NETWORK
108
Load Balancer
110
118
120
122
Server
Server
Server
126'
126''
126'''
128'
128''
128'''
Business Logic
Business Logic
Business Logic
Memory Store
Memory Store
Memory Store
130'
130''
130'''
Network Connection
124
Network Connection
124
112
Logical Memory
114

FIG. 1

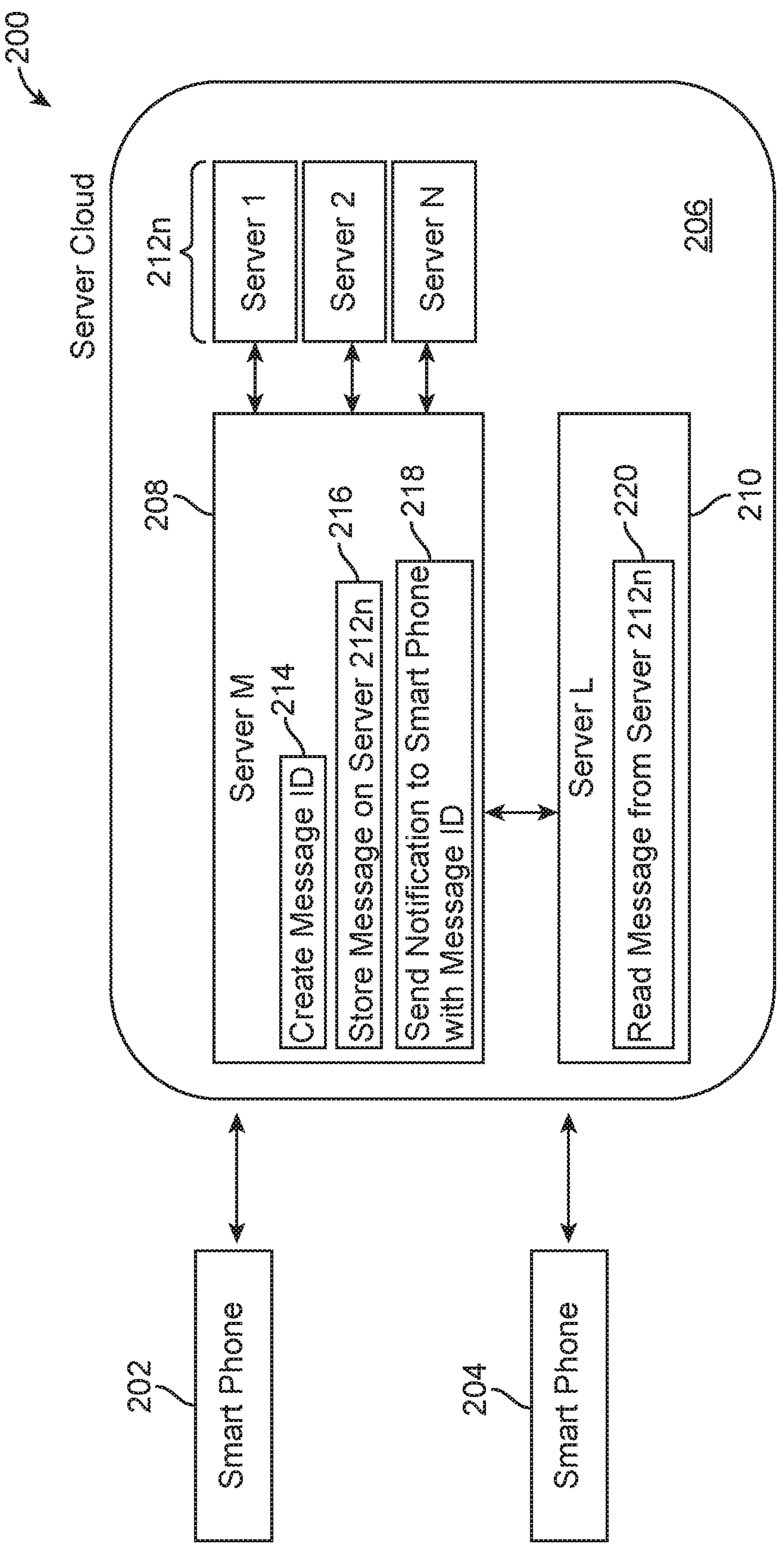
200
Server Cloud
212n
Server 1
Server 2
Server N
206
208
Server M
214
Create Message ID
216
Store Message on Server 212n
218
Send Notification to Smart Phone with Message ID
Server L
220
Read Message from Server 212n
210
202
Smart Phone
204
Smart Phone

FIG. 2

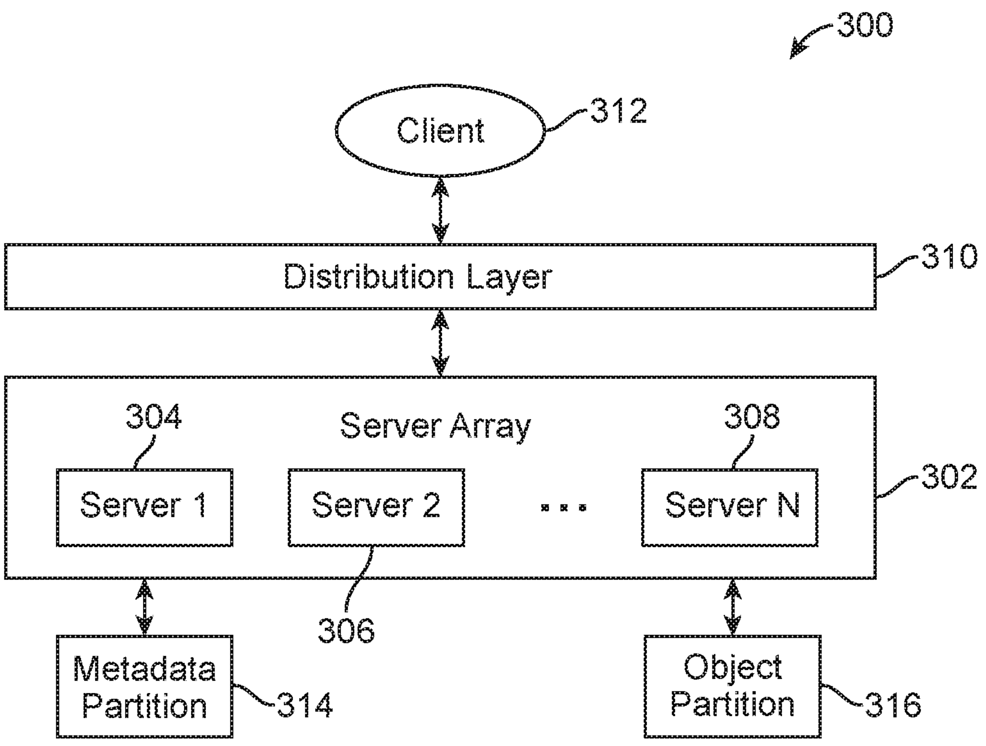
300
Client
312
Distribution Layer
310
Server Array
304
Server 1
Server 2
306
308
Server N
302
Metadata Partition
314
Object Partition
316

FIG. 3

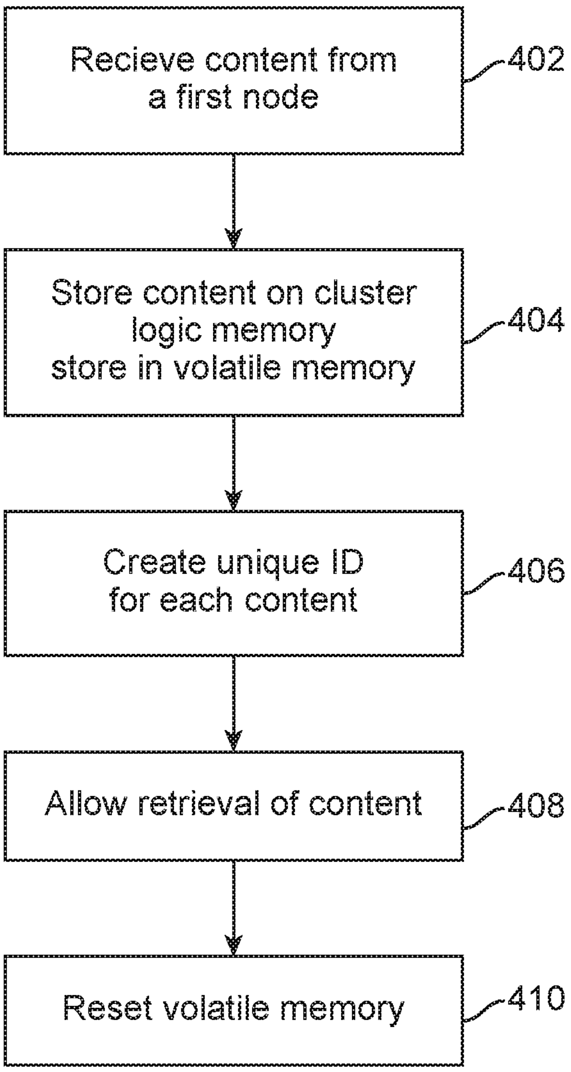
Recieve content from a first node
402
Store content on cluster logic memory store in volatile memory
404
Create unique ID for each content
406
Allow retrieval of content
408
Reset volatile memory
410

FIG. 4

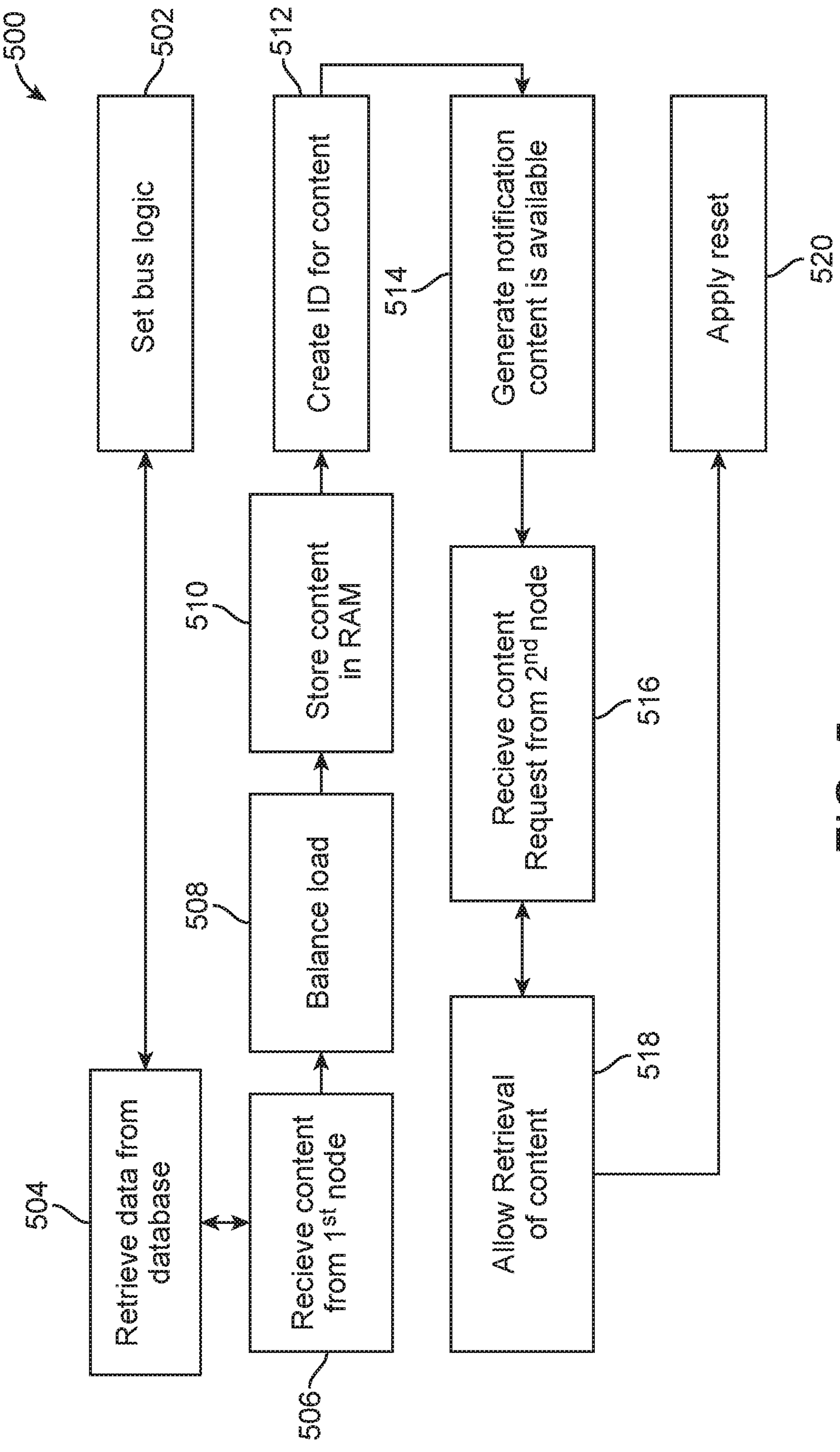
500
502
Set bus logic
504
Retrieve data from database
506
Recieve content from 1st node
508
Balance load
510
Store content in RAM
512
Create ID for content
514
Generate notification content is available
516
Recieve content Request from 2nd node
518
Allow Retrieval of content
520
Apply reset

FIG. 5

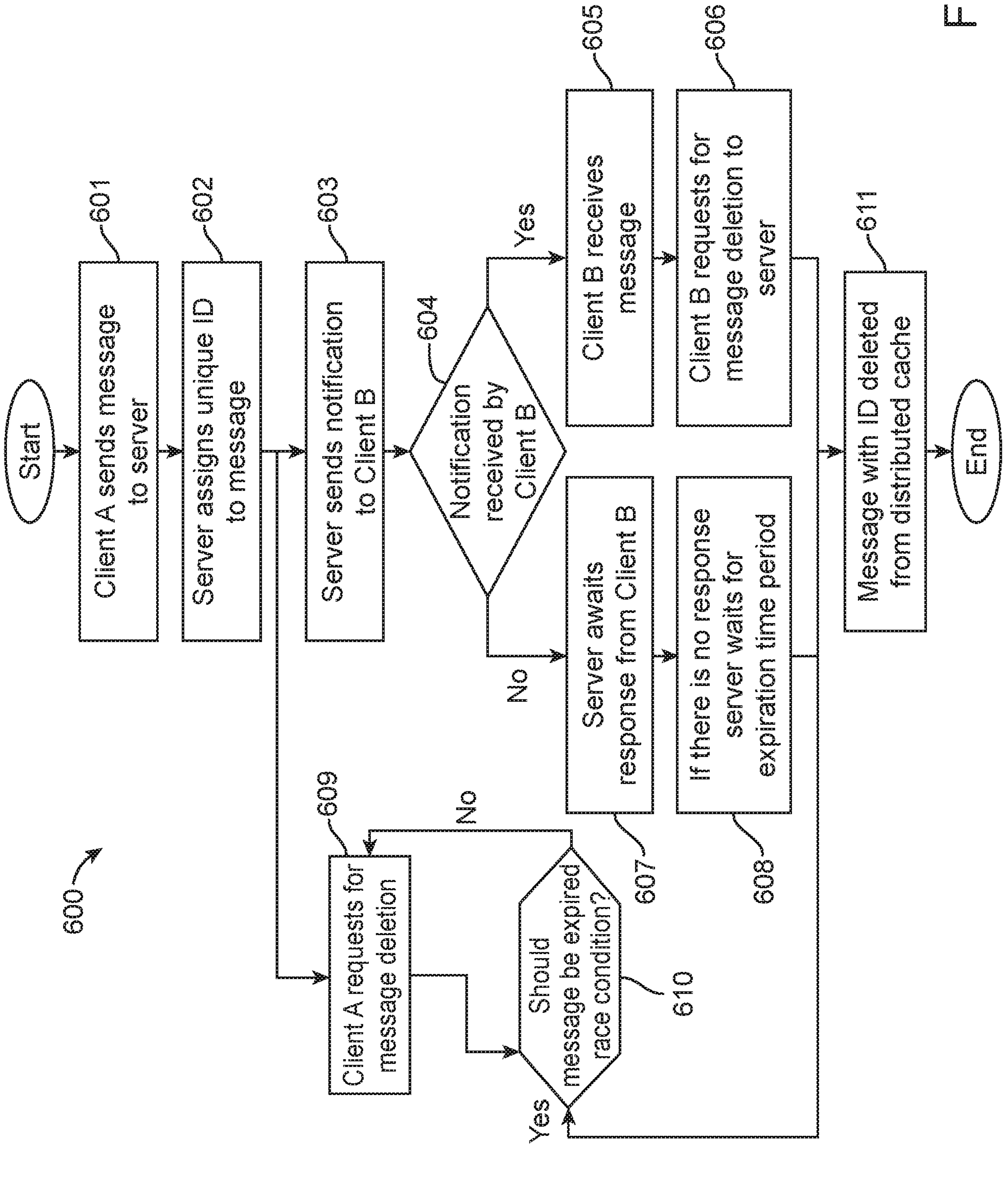
600
Start
Client A sends message to server
601
Server assigns unique ID to message
602
Server sends notification to Client B
603
Notification received by Client B
604
Yes
No
Client B receives message
605
Client B requests for message deletion to server
606
Server awaits response from Client B
607
If there is no response server waits for expiration time period
608
Client A requests for message deletion
609
Should message be expired race condition?
610
No
Yes
Message with ID deleted from distributed cache
611
End

FIG. 6

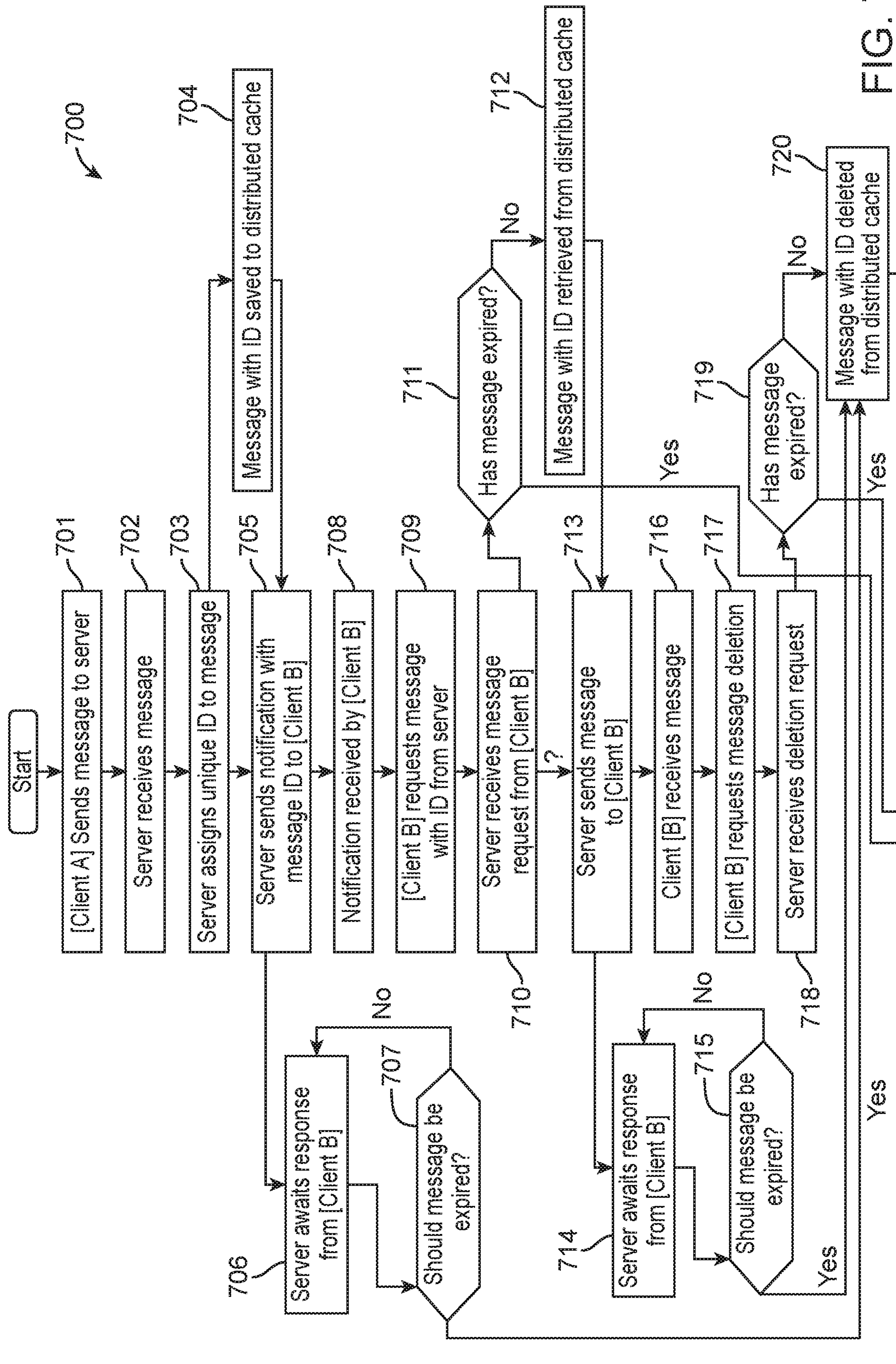
700
Start
701
[Client A] Sends message to server
702
Server receives message
703
Server assigns unique ID to message
704
Message with ID saved to distributed cache
705
Server sends notification with message ID to [Client B]
706
Server awaits response from [Client B]
707
Should message be expired?
No
Yes
708
Notification received by [Client B]
709
[Client B] requests message with ID from server
710
Server receives message request from [Client B]
711
Has message expired?
No
Yes
712
Message with ID retrieved from distributed cache
?
713
Server sends message to [Client B]
714
Server awaits response from [Client B]
715
Should message be expired?
No
Yes
716
Client [B] receives message
717
[Client B] requests message deletion
718
Server receives deletion request
719
Has message expired?
No
Yes
720
Message with ID deleted from distributed cache
End

FIG. 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US16/68752

A. CLASSIFICATION OF SUBJECT MATTER
IPC - H04W 4/12, 4/14, 12/06; G06F 15/16; H04M 1/725 (2017.01)
CPC - H04L 12/581, 12/5895, 51/04, 51/12, 51/38; H04W 4/12, 4/14, 12/06; G06Q 10/107; H04W 4/02

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)
See Search History document

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
See Search History document

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
See Search History document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X --- Y	WO 2016/179235 A1 (SNAPCHAT, INC.) 10 November 2016; abstract; paragraphs [0016]-[0019], [0021]-[0022], [0024], [0028], [0037]-[0038], [0050]-[0051], [0059], [0073]-[0074], [0078], [0083]-[0086], [0090]-[0091], [0096], [0101], [0108], [0115], [0118], [0120]-[0123]; figures 1, 2a-2C, 7-9, 11.	1-5, 7-14, 16-20 -------------------- 6, 15
Y	US 2015/0256675 A1 (24/7 CUSTOMER, INC.) 10 September 2015; abstract; figure 1; paragraphs [0037], [0040], [0062], [0088].	6, 15
A	WO 2015/156629 A1 (TUMTREE CO., LTD.) 15 October 2015; entire document.	1-20
A	US 2002/0026487 A1 (OGILVIE, J et al.) 28 February 2002; entire document.	1-20
A	US 6,137,864 A (YAKER, R) 24 October 2000; entire document.	1-20

☐ 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 application or patent 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 19 February 2017 (19.02.2017)	Date of mailing of the international search report 08 MAR 2017
Name and mailing address of the ISA/ Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300	Authorized officer Shane Thomas PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774