



- (51) **International Patent Classification:**
G06F 15/16 (2006.01) *H04L 29/06* (2006.01)
- (21) **International Application Number:**
PCT/US2014/049022
- (22) **International Filing Date:**
31 July 2014 (31.07.2014)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (71) **Applicant:** HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P. [US/US]; 11445 Compaq Center Drive West, Houston, Texas 77070 (US).
- (72) **Inventor:** VAQUERO GONZALEZ, Luis Miguel; Long-down Avenue, Stoke Gifford, Bristol South Gloucestershire BS34 8QZ (GB).
- (74) **Agents:** LEE, Rachel Jeong-Eun et al.; Hewlett-Packard Company, Intellectual Property Administration, 3404 E. Harmony Road, Mail Stop 35, Fort Collins, Colorado 80528 (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, 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, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, 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.

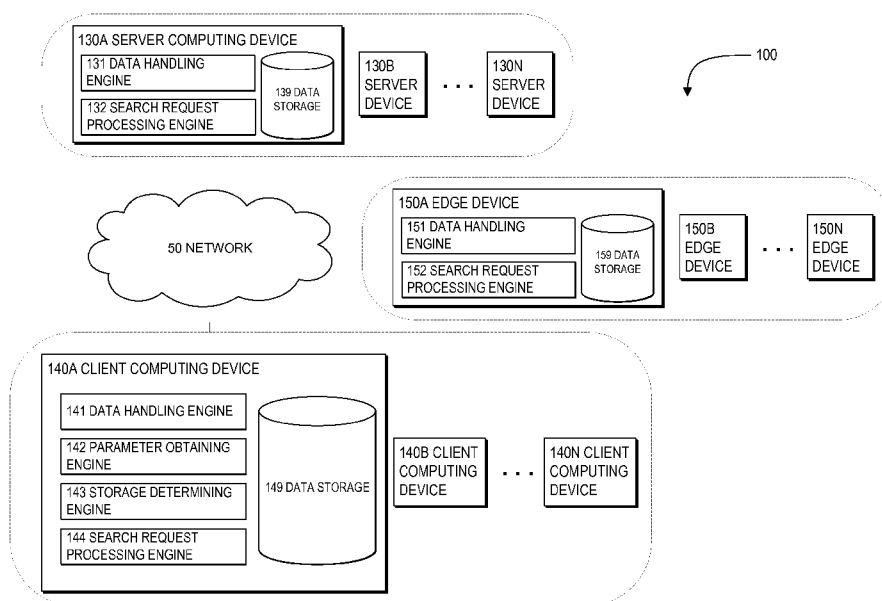
- (84) **Designated States** (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

- as to the identity of the inventor (Rule 4.17(i))
- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))

Published:

- with international search report (Art. 21(3))

(54) **Title:** DATA STORAGE IN FOG COMPUTING**FIG. 1**

(57) **Abstract:** Examples relate to storing data in fog computing. The examples disclosed herein enable storing, by a first client node, the first data set in the first client node. The first client node may communicate with a plurality of client nodes over a first network such as peer-to-peer (P2P) computer network. The first client node may obtain access frequency information indicating a number of accesses to the first data set in the first client node over a first time period. The examples further enable determining, by the first client node, whether to store at least a portion of the first data set in at least one of the plurality of client nodes based on the access frequency information.



DATA STORAGE IN FOG COMPUTING

BACKGROUND

[0001] With the rapid development of the Internet and communication technology, communication-enabled client devices such as Smartphones and the Internet of Things (IoT) have been popularized to a great extent. As an increasing number of such client devices are being used and operated at the edge of the network, their constant data stream over the core network puts an unnecessary burden on the network. By constantly seeking and retrieving data in and out of a mass data storage in the core network, valuable network and storage resources may be wasted.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] The following detailed description references the drawings, wherein:

[0003] FIG. 1 is a block diagram depicting an example system comprising various components including a client computing device in communication with a peer client computing device, an edge device, and a server computing device for determining where to send data based on at least one parameter.

[0004] FIG. 2 is a block diagram depicting an example server computing device for receiving a data set from a client computing device, storing the data set in the server computing device, and processing a search request originated from a client computing device.

[0005] FIG. 3 is a block diagram depicting an example edge device for receiving a data set from a client computing device, storing the data set in the edge device, and processing a search request originated from a client computing device.

[0006] FIG. 4 is a block diagram depicting an example client computing device for storing a data set in the client computing device, determining whether to send the data set to a peer client computing device, an edge device, or a server computing device based on at least one parameter, and obtaining a plurality of search results responsive to a search query.

[0007] FIG. 5 is a flow diagram depicting an example method for determining whether to store a data set in a peer client node based on access frequency information.

[0008] FIG. 6 is a flow diagram depicting an example method for determining whether to store a data set in a peer client node, an edge node, or a server node based on at least one parameter.

[0009] FIG. 7 is a flow diagram depicting an example method for determining whether to store a data set in an edge node or a server node based on at least one parameter.

[0010] FIG. 8 is a flow diagram depicting an example method for storing a set of data sent from a client computing device in a server computing device and processing a search request originated from a client computing device.

DETAILED DESCRIPTION

[0011] The following detailed description refers to the accompanying drawings. Wherever possible, the same reference numbers are used in the drawings and the following description to refer to the same or similar parts. It is to be expressly understood, however, that the drawings are for the purpose of illustration and description only. While several examples are described in this document, modifications, adaptations, and other implementations are possible. Accordingly, the following detailed description does not limit the disclosed examples. Instead, the proper scope of the disclosed examples may be defined by the appended claims.

[0012] With the rapid development of the Internet and communication technology, communication-enabled client devices such as Smartphones and the Internet of Things (IoT) have been popularized to a great extent. As an increasing number of such client devices are being used and operated at the edge of the network, their constant data stream over the core network puts an unnecessary burden on the network. By constantly seeking and retrieving data in and out of a mass data storage like the cloud in the core network, valuable network and storage resources may be wasted. For example, a Facebook chat message to a friend nearby may traverse the metropolitan network, gets into the core network, and in some instances, travel over ocean or several continents only to get delivered to the friend sitting next to you.

[0013] Instead of focusing on storing all of the data in the core network, some short-lived and locally-consumed data may be identified and stored on the client devices themselves or devices that sit between the client devices and the core network. Storing and keeping the data close to the edge of the network rather than in a mass storage like the cloud in the core network may be referred to as the “fog” or “fog computing.”

[0014] Examples disclosed herein relate to determining where to store a data set in the fog based on at least one parameter. In particular, the disclosed examples relate to determining whether to store the data set in a client computing device, a peer computing device of the client computing device, an edge device, and/or a server computing device based on at least one parameter. The at least one parameter may include access frequency information, aggregate access frequency information, access region information, aggregate access region information, user-specified rules, and/or other parameters. These parameters may be used to determine whether to send at least a portion of the data set from the client computing device to a peer client computing device, to an edge device, and/or to a server computing device. For example, based on these parameters, the data set may travel from the client computing device to various peer client computing devices, to an edge device, and/or to a server computing device. As such, as the data set gets more popular among various users from different parts of the world, the data set may be

dispersed across and distributed among various client devices, edge devices, and/or server devices.

[0015] FIG. 1 is a block diagram depicting an example system 100 comprising various components including a client computing device in communication with a peer client computing device, an edge device, and a server computing device for determining where to send data based on at least one parameter.

[0016] The various components may include server computing devices 130 (illustrated as 130A, 130B, ..., 130N), edge devices 150 (illustrated as 150A, 150B, ..., 150N), and client computing devices 140 (illustrated as 140A, 140B, ..., 140N). Each client computing device 140A, 140B, ..., 140N may communicate requests to and/or receive responses from at least one of server computing devices 130. Server computing devices 130 may receive and/or respond to requests from client computing devices 140. Client computing devices 140 may include any type of computing device providing a user interface through which a user can interact with a software application. For example, client computing devices 140 may include a laptop computing device, a desktop computing device, an all-in-one computing device, a tablet computing device, a mobile phone, an electronic book reader, a network-enabled appliance such as a "Smart" television, and/or other electronic device suitable for displaying a user interface and processing user interactions with the displayed interface. While server computing device 130A, 130B, ..., 130N is depicted as a single computing device, server computing device 130A, 130B, ..., 130N may include any number of integrated or distributed computing devices serving at least one software application for consumption by client computing devices 140.

[0017] Each client computing device 140A, 140B, ..., 140N may communicate requests to and/or receive responses from at least one of edge devices 150 (illustrated as 150A, 150B, ..., 150N). Edge devices 150 may receive and/or respond to requests from client computing devices 140. In some implementations, each edge device 150A, 150B, ..., 150N may facilitate and/or enable communication between a set of client computing devices 140 and a core network (also known as a network

core), thus establishing a communication line to server computing device 130. For example, the core network may be the central backbone of a telecommunication network that provides various services to client computing devices 140 that are connected by an access network. The set of client computing devices 140 may communicate with at least one of edge devices 150 over an access network that may include residential access networks, institutional access networks, and/or mobile access networks. Examples of edge devices 150 may include, but not limited to, a base transceiver station (BTS), a router, a routing switch, an integrated access device, a network device (e.g., digital subscriber line access multiplexer (DSLAM)), and/or any other edge device that sits between client computing devices 140 and the core network.

[0018] The various components (e.g., components 130, 140, and 150) depicted in FIG. 1 may be coupled to at least one other component via a network 50. Network 50 may comprise any infrastructure or combination of infrastructures that enable electronic communication between the components. Network 50 may include the core network and/or access networks. For example, network 50 may include at least one of the Internet, an intranet, a PAN (Personal Area Network), a LAN (Local Area Network), a WAN (Wide Area Network), a SAN (Storage Area Network), a MAN (Metropolitan Area Network), a wireless network, a cellular communications network, a Public Switched Telephone Network, and/or other network.

[0019] According to various implementations, system 100 and the various components described herein may be implemented in hardware and/or programming that configures hardware. Furthermore, in FIG. 1 and other Figures described herein, different numbers of components or entities than depicted may be used.

[0020] As detailed below, client computing device 140 may comprise a data handling engine 141, a parameter obtaining engine 142, a storage determining engine 143, a search request processing engine 144, and/or other engines. The term “engine”, as used herein, refers to a combination of hardware and programming that performs a designated function. As is illustrated respect to FIGS. 2-4, the hardware

of each engine, for example, may include one or both of a processor and a machine-readable storage medium, while the programming is instructions or code stored on the machine-readable storage medium and executable by the processor to perform the designated function.

[0021] Data handling engine 141 may receive a set of data and/or store the set of data in a data storage (e.g., a data storage 149) coupled to a first client computing device (also referred herein as “client device,” “client node,” “client,” and the like). The set of data may include, for example, text data, image data, video data, audio data, or any combination thereof.

[0022] In some implementations, a user of the first client computing device may enter the set of data into the first client computing device for storage. For example, the user may take a picture using his/her mobile device and/or save the picture in a local data storage coupled to the mobile device. In some implementations, the set of data may be received from another client computing device (e.g., a second client computing device). The first client computing device may be a peer client device of the second client computing device where the first and second client devices communicate via a peer-to-peer (P2P) computer network. The first client computing device may receive a copy of the data set (or a portion thereof) from the second client computing device. For example, the second client computing device may retain the original copy of the data set while the first client computing device stores a replication of the data set (or a portion thereof). In another example, the second client computing device may remove the data set from its data storage after the data set has been sent to the first client computing device.

[0023] Parameter obtaining engine 142 may obtain at least one parameter comprising access frequency information, aggregate access frequency information, access region information, aggregate access region information, user-specified rules, and/or other parameters. These parameters may be used to determine whether to send at least a portion of the data set from the first client computing device to a peer client computing device (also referred herein as “peer device,” “peer node,” “peer,”

and the like), to an edge device (also referred herein as “edge node,” “edge,” and the like), and/or to a server computing device (also referred herein as “server device,” “server node,” “server,” and the like). For example, based on these parameters, the data set may travel from the first client computing device to various peer client computing devices, to an edge device, and/or to a server computing device. As such, as the data set gets more popular among various users from different parts of the world, the data set may be dispersed across and/or distributed among various client devices, edge devices, and/or server devices.

[0024] The access frequency information may indicate a number of accesses to the data set in the first client computing device (and/or the data storage coupled to the first client computing device) over a first time period. The access frequency information may include a number of times the data set is accessed by other devices such as various client devices, edge devices, and/or server devices. For example, the access frequency information may indicate 100 accesses for the last 5 minutes (e.g., the first time period), which may be converted to 20 accesses per minute.

[0025] The aggregate access frequency information may include information that aggregates the access frequency information across the first client device and a plurality of peer client devices (of the first client device) having at least a portion of the data set. For example, the aggregate access frequency information may comprise a sum of the number of accesses to each copy of the data set (or a portion thereof) being stored in different client devices over the first time period. In another example, the aggregate access frequency information may comprise an average number of accesses to each copy of the data set (or a portion thereof) being stored in different client devices over the first time period. In yet another example, the aggregate access frequency information may comprise the highest (or lowest) access frequency recorded given the aggregate access frequency information.

[0026] In some implementations, when a client node stores only a portion of the data set, the access frequency may be computed proportional to the percentage of the portion in the data set. For example, if the access frequency to a first portion of

the data set being stored in a second client device equals to 200 per minute while the first portion comprises 50% of the data set, the access frequency information for the second client device may be 100 per minute (e.g., $200 * 50\%$). Similarly, if the access frequency to a second portion of the data being stored in a third client device equals to 400 per minute while the second portion comprises 20% of the data set, the access frequency information for the third client device may be 80 per minute (e.g., $400 * 20\%$). In these implementations, the aggregate access frequency information may comprise the sum or average of the frequency of access to the data set in the first client device that has the data set, 100/min for the second client device, and 80/min for the third client device. In other implementations, the aggregate access frequency information may indicate the highest access frequency or, in some instances, the lowest access frequency.

[0027] The access region information may include information indicating an access region from which an access to the data set in the first client device is made. For example, whether a particular access is made from the same region as the first client device and/or which region the access is made from (if not made from the same region) may be tracked and/or logged. For example, let's assume that the first client device is located in a first region (e.g., a region may comprise an area covered by a particular edge node, a county, a city, a zip code, a state, etc.). While the data set in the first client device is accessed by various other devices, an access region from which each access is made may be tracked. To identify the access region of a particular device making the access, an IP address of the particular device, a location of the particular device, a location of an edge node covering the particular device, and/or other location information related to the particular device may be used. In some implementations, a number of accesses made from access regions other than the region associated with the first client device may be determined based on the access region information. In some implementations, a frequency of those outside-the-region accesses over a certain time period may be determined based on the access region information. More accesses (or more frequent accesses) from access

regions other than the region where the first client device is located may mean that the data set is gaining popularity around the world.

[0028] The aggregate access region information may include information that aggregates the access region information across the first client device and the plurality of peer client devices (of the first client device) having at least a portion of the data set. Based on the aggregate access region information, an aggregate number of accesses made from access regions other than the region associated with the first client device may be determined. The aggregate number of accesses may indicate, for example, a sum of the number of outside-the-region accesses or an average of the number of outside-the-region accesses. In some implementations, the aggregate number of accesses may represent the highest (or lowest) number of outside-the-region accesses. In some implementations, an aggregate frequency of the outside-the-region accesses over a certain time period (e.g., the past 5 days) may be determined using the aggregate access region information.

[0029] The access frequency information and/or access region information of the first client device may be tracked or otherwise obtained by the first client device. For example, the access frequency and/or access region information of the first client device may be tracked using generational garbage collection techniques implemented in the first client device. In some implementations, the first client device may be associated at least a delegate client device that may track or otherwise obtain the access frequency information and/or access region information of the first client device on behalf of the first client device. Similarly, the access frequency information and/or access region information may be aggregated by the first client device and/or at least one delegate client device of the first client device. For example, the first client device may be associated at least one delegate client device that may aggregate the access frequency information on behalf of the first client device. The first client device and its various delegate devices may communicate, for example, via the P2P computer network and/or using TCP/IP routing algorithms. When there are multiple delegate client devices associated with the first client device, and there is a

conflict in determining whether to send the data set and/or where to send the data set, various voting methods may be implemented to make the determination. The various voting methods may include at least one of a majority voting method, a weighted voting method, a priority voting method (e.g., first wins), a unanimous voting method, and/or other voting methods.

[0030] In some instances, a user of the first client device may specify at least one user-specified rule that may determine whether to send at least a portion of the data set from the first client computing device to a peer client computing device, to an edge device, and/or to a server computing device. The user-specified rule may define space-temporal policies that dictate where the data set can be stored, when the data set can be sent to and/or stored in devices (e.g., peer devices, edge devices, server devices, etc.) other than the first client device, how long the data set should be stored in a particular device, etc.

[0031] The user-specified rule may comprise a rule related to confidentiality or security, a rule related to convenience, etc. The confidentiality/security rule may designate a particular data set as “confidential” data. In this case, the particular data set may not be sent to peer devices even when the access frequency reaches a predetermined threshold, for example. An example of the convenience rule may include specifying a user-preferred storage location for a particular data set. For instance, a user may decide to have a copy of the data set in a device (e.g., a client device, edge device, server device, etc.) located in Singapore since he/she frequently travels there and wants fast access to the data set.

[0032] Storage determining engine 143 may determine whether to send at least a portion of the data set from the first client device to a peer client computing device, to an edge device, and/or to a server computing device based on at least one of the parameters discussed herein. In some instances, the data set may be divided into multiple “portions” and distributed among various devices. The divided portions may be equal in size or may be of different sizes. In some instances, the entire copy of the data may be sent from the first client device.

[0033] In some implementations, storage determining engine 143 may determine whether to send at least a portion of the data set from the first client device to at least one peer client device, to an edge device, and/or to a server computing device based on the access frequency information that may indicate a number of accesses to the data set in the first client device over a first time period and/or based on the aggregate access frequency information. For example, when a frequency of access to the data set reaches a first frequency threshold, a copy of the data set (or at least a portion thereof) may be sent to at least one peer device of the first client device. In some implementations, a second threshold may be specified such that when the access frequency reaches the second frequency threshold, a copy of the data set (or at least a portion thereof) may be sent to another peer device that is different from the peer device that received the copy when the first threshold was reached.

[0034] Note that the first client device may communicate with multiple peer devices which may be divided into different groups. A single peer device may be linked to more than one peer group. For example, a user of the first client device may group the peer devices into three different peer groups – family, friends, and business. The user may specify which peer group or groups should receive a copy of the data set when the determination to send the data set to a peer device is made by storage determining engine 143. The user may identify individual peer devices to receive a copy of the data set when the determination is made regardless of which peer group they belong to.

[0035] In some implementations, storage determining engine 143 may determine whether to send at least a portion of the data set from the first client device to at least one edge device based on the access frequency information and/or the aggregate access frequency information. For example, when the frequency of access to the data set stored in the first client device reaches a third frequency threshold, a copy of the data set (or at least a portion thereof) may be sent to at least one edge device. The at least one edge device may enable communication between the first client

device and a second network (e.g., core network), wherein the first client device and at least one server device communicate over the second network.

[0036] In some instances, the third frequency threshold may be set higher than the first and/or second frequency thresholds such that as the access frequency increases, the data set (or at least a portion thereof) may be sent from the first client device to at least one peer device and to at least one edge device. Similarly, a fourth frequency threshold may be set even higher than the third frequency threshold such that as the access frequency increases, the data set (or at least a portion thereof) may be sent from the first client device to at least one peer client device, to at least one edge device, and to at least one server device. The at least one server device may communicate with the first client device over the second network (e.g., core network).

[0037] In other instances, the access frequency may be measured over different time periods. For example, a first access frequency may be measured over a first time period whereas a second access frequency may be measured over a second time period that is longer than the first time period. If the first access frequency reaches the first frequency threshold, for example, storage determining engine 143 may be configured to determine that the data set (or at least a portion thereof) should be sent to at least one peer device. On the other hand, if the second access frequency reaches the first frequency threshold (or another threshold value), storage determining engine 143 may be configured to determine that the data set (or at least a portion thereof) should be sent to at least one edge device. A third access frequency may be measured over a third time period that is longer than the second time period. If the third access frequency reaches the first frequency threshold (or another threshold value), storage determining engine 143 may be configured to determine that the data set (or at least a portion thereof) should be sent to at least one server device. As such, when the data set is frequently accessed for a longer time period, it may be more likely that more copies of the data set get distributed among different devices (e.g., peer client devices, edge devices, server devices, etc.).

Note that for various examples related to the access frequency information discussed herein, the access frequency information and/or the number of accesses may also refer to the aggregate access frequency information and/or the aggregate number of accesses as discussed herein with respect to parameter obtaining engine 142.

[0038] In some implementations, storage determining engine 143 may determine whether to send at least a portion of the data set from the first client device to at least one peer client device, to an edge device, and/or to a server computing device based on the access region information and/or the aggregate access region information. The access region information and/or the aggregate access region information may comprise a number of accesses made from access regions other than the region associated with first client device and/or a frequency of those outside-the-region accesses over a certain time period. More accesses (or more frequent accesses) from access regions other than the region where the first client device is located may mean that the data set is gaining popularity around the world. For example, storage determining engine 143 may compare the number (or the frequency) of outside-the-region accesses to a first region threshold to determine whether to send at least a portion of the data set to at least one server device. When the data set is frequently accessed and/or accessed from many different parts of the world, it may make sense to store the data set (or at least a portion thereof) in a data storage (e.g., a data storage 139) coupled to a server device so that the data set may be accessed more easily and quickly by multiple users from different parts of the world. The data storage coupled the server device may comprise, for example, a cloud-based data storage. Note that for various examples related to the access region information discussed herein, the access region information and/or the number or frequency of outside-the-region accesses may also refer to the aggregate access region information and/or the aggregate number or frequency of outside-the-region accesses as discussed herein with respect to parameter obtaining engine 142. Further, although only a limited number of thresholds (e.g., access frequency and/or access

region related thresholds) are discussed herein, additional thresholds (and corresponding actions when reached) may be specified as well.

[0039] In some implementations, storage determining engine 143 may determine whether to send at least a portion of the data set from the first client device to at least one peer client device, to an edge device, and/or to a server computing device based on at least one user-specified rule. For example, a user of the first client device may specify at least one user-specified rule that may determine whether to send at least a portion of the data set from the first client computing device to a peer client computing device, to an edge device, and/or to a server computing device. The user-specified rule may define space-temporal policies that dictate where the data set can be stored, when the data set can be sent to and/or stored in devices (e.g., peer devices, edge devices, server devices, etc.) other than the first client device, how long the data set should be stored in a particular device, etc.

[0040] Search request processing engine 144 may receive a search query that may comprise one or more query terms. The first client device may search various data storages including its peer devices, edge devices, and/or server devices using the search query. A plurality of search results responsive to the search query may be obtained. Each of the plurality search results may be identified with its source location. For example, a search result that is currently stored in a particular peer device may be shown with an identifier associated with the particular peer device. A user who submitted the query to the first client device may be able to select the search result, via a user interface coupled to the first client device, to access the data set corresponding to the search result stored in the particular peer device.

[0041] In performing their respective functions, engines 141-144 may access data storage 149. Data storage 149 may represent any memory accessible to engines 141-144 that can be used to store and retrieve data.

[0042] As detailed below, edge device 150 may comprise a data handling engine 151, a search request processing engine 152, and/or other engines.

[0043] Data handling engine 151 may receive, by an edge device, the data set (or a portion thereof) for storage in a data storage (e.g., a data storage 159) coupled to the edge device and/or store the received data set in the data storage. Data handling engine 151 may receive the data set after the first client computing device determines that the set of data should be sent from the first client device to the edge device based on at least one of the parameters discussed herein. In some implementations, data handling engine 151 may receive the data set after the first client device determines that the data set should be sent from the first client computing device to at least one peer client device (of the first client device) based on at least one of the parameters discussed herein.

[0044] Search request processing engine 152 may receive, from the first client device or another client device, a request to search for the data set. In response to the request, search request processing engine 152 may provide the data set to the client device from which the search request originated.

[0045] In performing their respective functions, engines 151-152 may access data storage 159. Data storage 159 may represent any memory accessible to engines 151-152 that can be used to store and retrieve data.

[0046] As detailed below, server computing device 130 may comprise a data handling engine 131, a search request processing engine 132, and/or other engines.

[0047] Data handling engine 131 may receive, by a server device, the data set (or a portion thereof) for storage in a data storage (e.g., data storage 139) coupled to the server device and/or store the received data set in the data storage. Data handling engine 131 may receive the data set after the first client computing device determines that the set of data should be sent from the first client device to the server device based on at least one of the parameters discussed herein. In some implementations, data handling engine 131 may receive the data set after the first client device determines that the data set should be sent from the first client computing device to at least one edge device based on at least one of the parameters discussed herein. In some implementations, data handling engine 131 may receive the data set after the

first client device determines that the data set should be sent from the first client computing device to at least one peer client device (of the first client device) based on at least one of the parameters discussed herein.

[0048] Search request processing engine 132 may receive, from the first client device or another client device, a request to search for the data set. In response to the request, search request processing engine 132 may provide the data set to the client device from which the search request originated.

[0049] In performing their respective functions, engines 131-132 may access data storage 139. Data storage 139 may represent any memory accessible to engines 131-132 that can be used to store and retrieve data.

[0050] Data storages 139, 149, and 159 may comprise random access memory (RAM), read-only memory (ROM), electrically-erasable programmable read-only memory (EEPROM), cache memory, floppy disks, hard disks, optical disks, tapes, solid state drives, flash drives, portable compact disks, and/or other storage media for storing computer-executable instructions and/or data.

[0051] Data storages 139, 149, and 159 may include a database to organize and store data. Database may be, include, or interface to, for example, an OracleTM relational database sold commercially by Oracle Corporation. Other databases, such as InformixTM, DB2 (Database 2) or other data storage, including file-based (e.g., comma or tab separated files), or query formats, platforms, or resources such as OLAP (On Line Analytical Processing), SQL (Structured Query Language), a SAN (storage area network), Microsoft AccessTM, MySQL, PostgreSQL, HSpace, Apache Cassandra, MongoDB, Apache CouchDBTM, or others may also be used, incorporated, or accessed. The database may reside in a single or multiple physical device(s) and in a single or multiple physical location(s). The database may store a plurality of types of data and/or files and associated data or file description, administrative information, or any other data.

[0052] FIG. 2 is a block diagram depicting an example server computing device 130 for receiving a data set from a client computing device, storing the data set in the server computing device, and processing a search request originated from a client computing device.

[0053] In the foregoing discussion, engines 131-132 were described as combinations of hardware and programming. Engines 131-132 may be implemented in a number of fashions. Referring to FIG. 2, the programming may be processor executable instructions 231-232 stored on a machine-readable storage medium 230 and the hardware may include a processor 238 for executing those instructions. Thus, machine-readable storage medium 230 can be said to store program instructions or code that when executed by processor 238 implements engines 131-132 of FIG. 1.

[0054] Machine-readable storage medium 230 (and other machine-readable storage medium discussed herein such as machine-readable storage medium 240 and 250) may be any electronic, magnetic, optical, or other physical storage device that contains or stores executable instructions. In some implementations, machine-readable storage medium 230 (and other machine-readable storage medium discussed herein such as machine-readable storage medium 240 and 250) may be a non-transitory storage medium, where the term “non-transitory” does not encompass transitory propagating signals.

[0055] Machine-readable storage medium 230 may be implemented in a single device or distributed across devices. Likewise, processor 238 may represent any number of processors capable of executing instructions stored by machine-readable storage medium 230. Processor 238 may be integrated in a single device or distributed across devices. Further, machine-readable storage medium 230 may be fully or partially integrated in the same device as processor 238, or it may be separate but accessible to that device and processor 238.

[0056] In one example, the program instructions may be part of an installation package that when installed can be executed by processor 238 to implement engines

131-132. In this case, machine-readable storage medium 230 may be a portable medium such as a floppy disk, CD, DVD, or flash drive or a memory maintained by a server from which the installation package can be downloaded and installed. In another example, the program instructions may be part of an application or applications already installed. Here, machine-readable storage medium 230 may include a hard disk, optical disk, tapes, solid state drives, RAM, ROM, EEPROM, or the like.

[0057] Processor 238 may be at least one central processing unit (CPU), microprocessor, and/or other hardware device suitable for retrieval and execution of instructions stored in machine-readable storage medium 230. Processor 238 may fetch, decode, and execute program instructions 231-232, and/or other instructions. As an alternative or in addition to retrieving and executing instructions, processor 238 may include at least one electronic circuit comprising a number of electronic components for performing the functionality of at least one of instructions 231-232, and/or other instructions.

[0058] In FIG. 2, the executable program instructions in machine-readable storage medium 230 are depicted as data handling instructions 231 and search request processing instructions 232. Instructions 231-232 represent program instructions that, when executed, cause processor 238 to implement engines 131-132, respectively.

[0059] FIG. 3 is a block diagram depicting an example edge device 150 for receiving a data set from a client computing device, storing the data set in the edge device, and processing a search request originated from a client computing device.

[0060] In the foregoing discussion, engines 151-152 were described as combinations of hardware and programming. Engines 151-152 may be implemented in a number of fashions. Referring to FIG. 3, the programming may be processor executable instructions 251-252 stored on a machine-readable storage medium 250 and the hardware may include a processor 258 for executing those instructions. Thus, machine-readable storage medium 250 can be said to store program

instructions or code that when executed by processor 258 implements engines 151-152 of FIG. 1.

[0061] Machine-readable storage medium 250 may be implemented in a single device or distributed across devices. Likewise, processor 258 may represent any number of processors capable of executing instructions stored by machine-readable storage medium 250. Processor 258 may be integrated in a single device or distributed across devices. Further, machine-readable storage medium 250 may be fully or partially integrated in the same device as processor 258, or it may be separate but accessible to that device and processor 258.

[0062] In one example, the program instructions may be part of an installation package that when installed can be executed by processor 258 to implement engines 151-152. In this case, machine-readable storage medium 250 may be a portable medium such as a floppy disk, CD, DVD, or flash drive or a memory maintained by a server from which the installation package can be downloaded and installed. In another example, the program instructions may be part of an application or applications already installed. Here, machine-readable storage medium 250 may include a hard disk, optical disk, tapes, solid state drives, RAM, ROM, EEPROM, or the like.

[0063] Processor 258 may be at least one central processing unit (CPU), microprocessor, and/or other hardware device suitable for retrieval and execution of instructions stored in machine-readable storage medium 250. Processor 258 may fetch, decode, and execute program instructions 251-252, and/or other instructions. As an alternative or in addition to retrieving and executing instructions, processor 258 may include at least one electronic circuit comprising a number of electronic components for performing the functionality of at least one of instructions 251-252, and/or other instructions.

[0064] In FIG. 3, the executable program instructions in machine-readable storage medium 250 are depicted as data handling instructions 251 and search request processing instructions 252. Instructions 251-252 represent program instructions

that, when executed, cause processor 258 to implement engines 151-152, respectively.

[0065] FIG. 4 is a block diagram depicting an example client computing device 140 for storing a data set in the client computing device, determining whether to send the data set to a peer client computing device, an edge device, or a server computing device based on at least one parameter, and obtaining a plurality of search results responsive to a search query.

[0066] In the foregoing discussion, engines 141-144 were described as combinations of hardware and programming. Engines 141-144 may be implemented in a number of fashions. Referring to FIG. 4, the programming may be processor executable instructions 241-244 stored on a machine-readable storage medium 240 and the hardware may include a processor 248 for executing those instructions. Thus, machine-readable storage medium 240 can be said to store program instructions or code that when executed by processor 248 implements engines 141-144 of FIG. 1.

[0067] Machine-readable storage medium 240 may be implemented in a single device or distributed across devices. Likewise, processor 248 may represent any number of processors capable of executing instructions stored by machine-readable storage medium 240. Processor 248 may be integrated in a single device or distributed across devices. Further, machine-readable storage medium 240 may be fully or partially integrated in the same device as processor 248, or it may be separate but accessible to that device and processor 248.

[0068] In one example, the program instructions may be part of an installation package that when installed can be executed by processor 248 to implement engines 141-144. In this case, machine-readable storage medium 240 may be a portable medium such as a floppy disk, CD, DVD, or flash drive or a memory maintained by a server from which the installation package can be downloaded and installed. In another example, the program instructions may be part of an application or applications already installed. Here, machine-readable storage medium 240 may

include a hard disk, optical disk, tapes, solid state drives, RAM, ROM, EEPROM, or the like.

[0069] Processor 248 may be at least one central processing unit (CPU), microprocessor, and/or other hardware device suitable for retrieval and execution of instructions stored in machine-readable storage medium 240. Processor 248 may fetch, decode, and execute program instructions 241-244, and/or other instructions. As an alternative or in addition to retrieving and executing instructions, processor 248 may include at least one electronic circuit comprising a number of electronic components for performing the functionality of at least one of instructions 241-244, and/or other instructions.

[0070] In FIG. 4, the executable program instructions in machine-readable storage medium 240 are depicted as data handling instructions 241, parameter obtaining instructions 242, storage determining instructions 243, and search request processing instructions 244. Instructions 241-244 represent program instructions that, when executed, cause processor 248 to implement engines 141-144, respectively.

[0071] FIG. 5 is a flow diagram depicting an example method 500 for determining whether to store a data set in a peer client node based on access frequency information. The various processing blocks and/or data flows depicted in FIG. 5 (and in the other drawing figures such as FIGS. 6-8) are described in greater detail herein. The described processing blocks may be accomplished using some or all of the system components described in detail above and, in some implementations, various processing blocks may be performed in different sequences and various processing blocks may be omitted. Additional processing blocks may be performed along with some or all of the processing blocks shown in the depicted flow diagrams. Some processing blocks may be performed simultaneously. Accordingly, method 500 as illustrated (and described in greater detail below) is meant to be an example and, as such, should not be viewed as limiting. Method 500 may be implemented in the form of executable instructions stored on a machine-readable storage medium, such as storage medium 240, and/or in the form of electronic circuitry.

[0072] Method 500 may start in block 521 where a first client node may store a first data set in the first client node. The first client node may communicate with a plurality of client nodes over a first network comprising a P2P computer network. The plurality of client nodes may be referred to as peer client nodes of the first client node. The first data set may include, for example, text data, image data, video data, audio data, or any combination thereof. In some implementations, a user of the first client node may enter the first data set into the first client node for storage. For example, the user may take a picture using his/her mobile device and/or save the picture in a local data storage coupled to the mobile device. In some implementations, the first data set may be received from another client node (e.g., a second client node). The first client node may be a peer node of the second client node where the first and second client nodes communicate over a P2P network.

[0073] In block 522, method 500 may include obtaining, by the first client node, access frequency information that may indicate a number of accesses to the first data set in the first client node over a first time period. For example, the access frequency information may include a number of times the first data set is accessed by other devices such as various client devices, edge devices, and/or server devices.

[0074] In block 523, method 500 may include determining whether to store at least a portion of the first data set in at least one of the plurality of client nodes based on the access frequency information. For example, when a frequency of accesses to the first data set reaches a predetermined frequency threshold, a copy of the data set (or at least a portion thereof) may be sent to and/or stored in at least one peer node of the first client node.

[0075] Referring back to FIG. 1, data handling engine 141 may be responsible for implementing block 521. Parameter obtaining engine 142 may be responsible for implementing block 522. Storage determining engine 143 may be responsible for implementing block 523.

[0076] FIG. 6 is a flow diagram depicting an example method 600 for determining whether to store a data set in a peer client node, an edge node, or a server node

based on at least one parameter. Method 600 as illustrated (and described in greater detail below) is meant to be an example and, as such, should not be viewed as limiting. Method 600 may be implemented in the form of executable instructions stored on a machine-readable storage medium, such as storage medium 240, and/or in the form of electronic circuitry.

[0077] Method 600 may start in block 621 where a first client node may store a first data set in the first client node. The first data set may include, for example, text data, image data, video data, audio data, or any combination thereof. In some implementations, a user of the first client node may enter the first data set into the first client node for storage. For example, the user may take a picture using his/her mobile device and/or save the picture in a local data storage coupled to the mobile device. In some implementations, the first data set may be received from another client node (e.g., a second client node). The first client node may be a peer node of the second client node where the first and second client nodes communicate over a P2P network.

[0078] In block 622, method 600 may include identifying peer nodes, an edge node, and/or a server node of the first client node.

[0079] In block 623, method 600 may include obtaining at least one parameter to be used in determining where to store the first data set. The parameters may include, for example, frequency and/or geographical region of accesses made to the first data set stored in the first client node.

[0080] In block 624, method 600 may determine whether to store the first data set (or at least a portion thereof) in at least one of the peer nodes based on the access frequency information. For example, when a frequency of accesses to the first data set reaches a first predetermined frequency threshold, method 600 may proceed to block 625 where a copy of the data set (or at least a portion thereof) may be provided to at least one peer node of the first client node. If method 600 determines that the access frequency does not meet the first frequency threshold, method 600 may return to block 623 to obtain updated access frequency information.

[0081] In block 626, method 600 may determine whether to store the first data set (or at least a portion thereof) in the edge node based on the access frequency information. For example, when the frequency of accesses to the first data set stored in the first client node reaches a second frequency threshold, the first data set (or at least a portion thereof) may be provided to and/or stored in the edge node (block 627). In some instances, the second frequency threshold may be set higher than the first frequency threshold such that as the access frequency increases, the first data set (or at least a portion thereof) may be sent from the first client node to at least one peer node and to the edge device. If method 600 determines that the access frequency does not meet the second frequency threshold, method 600 may return to block 623 to obtain updated access frequency information.

[0082] In block 628, method 600 may determine whether to store the first data set (or at least a portion thereof) in the server node based on the access frequency information and/or access region information. Going back to the above example in block 626, when the frequency of accesses to the first data set stored in the first client node reaches a third frequency threshold, the first data set (or at least a portion thereof) may be provided to and/or stored in the server node (block 629). In some instances, the third frequency threshold may be set even higher than the second frequency threshold such that as the access frequency increases, the first data set (or at least a portion thereof) may be sent from the first client device to at least one peer node, to at least one edge node, and to at least one server node.

[0083] The access region information may comprise a number (or a frequency) of accesses made from access regions other than the region associated with first client node. More accesses (or more frequency accesses) from access regions other than the region where the first client node is located may mean that the first data set is gaining popularity around the world. For example, method 600 may compare the access region information to a first region threshold to determine whether to send at least a portion of the data set to the server node. When the data set is frequently accessed and/or accessed from many different parts of the world, it may make sense

to store the first data set (or at least a portion thereof) in a data storage coupled to the server node so that the first data set may be accessed more easily and quickly by multiple users from different parts of the world. When the number (or the frequency) of outside-the-region accesses reaches the first region threshold, for example, the first data set (or at least a portion thereof) may be provided to and/or stored in the server node (block 629).

[0084] If method 600 determines that the access frequency does not meet the third frequency threshold and/or the access region information does not meet the first region threshold, method 600 may return to block 623 to obtain updated access frequency information and/or access region information.

[0085] Referring back to FIG. 1, data handling engine 141 may be responsible for implementing blocks 621, 625, 627, and 629. Parameter obtaining engine 142 may be responsible for implementing block 623. Storage determining engine 143 may be responsible for implementing block 622, 624, 626, and 628.

[0086] FIG. 7 is a flow diagram depicting an example method 700 for determining whether to store a data set in an edge node or a server node based on at least one parameter. Method 700 as illustrated (and described in greater detail below) is meant to be an example and, as such, should not be viewed as limiting. Method 700 may be implemented in the form of executable instructions stored on a machine-readable storage medium, such as storage medium 240, and/or in the form of electronic circuitry.

[0087] Method 700 may start in block 721 where a first data set (or a portion thereof) may be stored in a first client node and a plurality of peer client nodes of the first client node.

[0088] In block 722, method 700 may include identifying an edge node and/or a server node of the first client node.

[0089] In block 723, method 700 may obtain, by the first client node or at least one delegate node of the first client node, access frequency information and/or access

region information with regard to the first data set stored in each of the first client node and the plurality of peer client nodes. In block 724, method 700 may include aggregating, by the first client node or the at least one delegate node of the first client node, the access frequency information and/or access region information across the first client node and the plurality of peer client nodes.

[0090] In block 725, method 700 may determine whether to store the first data set (or at least a portion thereof) in the edge node based on the aggregate access frequency. For example, when the aggregate frequency of accesses to the first data set stored in the first client node reaches a first frequency threshold, the first data set (or at least a portion thereof) may be provided to and/or stored in the edge node (block 726). If method 700 determines that the aggregate access frequency does not meet the first frequency threshold, method 700 may return to block 723 to obtain updated access frequency information.

[0091] In block 727, method 700 may determine whether to store the first data set (or at least a portion thereof) in the server node based on the aggregate access frequency information and/or aggregate access region information. Going back to the above example in block 725, when the aggregate frequency of accesses to the first data set stored in the first client node reaches a second frequency threshold, the first data set (or at least a portion thereof) may be provided to and/or stored in the server node (block 728). In some instances, the second frequency threshold may be set higher than the first frequency threshold such that as the aggregate access frequency increases, the first data set (or at least a portion thereof) may be sent from the first client node to the edge node, and to the server node.

[0092] The aggregate access region information may comprise an aggregate number (or aggregate frequency) of accesses made from access regions other than the region associated with first client node. More accesses (or more frequent accesses) from access regions other than the region where the first client node is located may mean that the first data set is gaining popularity around the world. For example, method 700 may compare the number (or the frequency) of outside-the-

region accesses to a first region threshold to determine whether to send at least a portion of the first data set to the server node. When the data set is frequently accessed and/or accessed from many different parts of the world, it may make sense to store the first data set (or at least a portion thereof) in a data storage coupled to the server node so that the first data set may be accessed more easily and quickly by multiple users from different parts of the world. When the aggregate number (or aggregate frequency) of outside-the-region accesses reaches the first region threshold, for example, the first data set (or at least a portion thereof) may be provided to and/or stored in the server node (block 728).

[0093] If method 700 determines that the aggregate access frequency does not meet the second frequency threshold and/or the aggregate number (or frequency) of outside-the-region accesses does not meet the first region threshold, method 700 may return to block 723 to obtain updated access frequency information and/or access region information.

[0094] Referring back to FIG. 1, data handling engine 141 may be responsible for implementing blocks 721, 726, and 728. Parameter obtaining engine 142 may be responsible for implementing block 723 and 724. Storage determining engine 143 may be responsible for implementing block 722, 725, and 727.

[0095] FIG. 8 is a flow diagram depicting an example method 800 for storing a set of data sent from a client computing device in a server computing device and processing a search request originated from a client computing device. Method 800 as illustrated (and described in greater detail below) is meant to be an example and, as such, should not be viewed as limiting. Method 800 may be implemented in the form of executable instructions stored on a machine-readable storage medium, such as storage medium 230, and/or in the form of electronic circuitry.

[0096] Method 800 may start in block 821 where the server computing device receives the data set (or a portion thereof) for storage in a data storage coupled to the server device from a first client computing device. In block 822, the received data set may be stored in the data storage.

[0097] In block 823, method 800 may receive, from the first client device or another client device, a request to search for the data set. In response to the request, method 800 may provide the data set to the client device from which the search request originated (block 824).

[0098] Referring back to FIG. 1, data handling engine 131 may be responsible for implementing blocks 821 and 822. Search request processing engine 132 may be responsible for implementing blocks 823 and 824.

[0099] The foregoing disclosure describes a number of example implementations for data storage in fog computing. The disclosed examples may include systems, devices, computer-readable storage media, and methods for data storage in fog computing. For purposes of explanation, certain examples are described with reference to the components illustrated in FIGS. 1-4. The functionality of the illustrated components may overlap, however, and may be present in a fewer or greater number of elements and components.

[00100] Further, all or part of the functionality of illustrated elements may co-exist or be distributed among several geographically dispersed locations. Moreover, the disclosed examples may be implemented in various environments and are not limited to the illustrated examples. Further, the sequence of operations described in connection with FIGS. 5-8 are examples and are not intended to be limiting. Additional or fewer operations or combinations of operations may be used or may vary without departing from the scope of the disclosed examples. Furthermore, implementations consistent with the disclosed examples need not perform the sequence of operations in any particular order. Thus, the present disclosure merely sets forth possible examples of implementations, and many variations and modifications may be made to the described examples. All such modifications and variations are intended to be included within the scope of this disclosure and protected by the following claims.

CLAIMS

1. A method for execution by a client computing device for storing data in fog computing, the method comprising:

storing, by a first client node, a first data set in the first client node, wherein the first client node communicates with a plurality of client nodes over a first network, the first network comprising a peer-to-peer (P2P) computer network;

obtaining, by the first client node, access frequency information indicating a number of accesses to the first data set in the first client node over a first time period; and

determining, by the first client node, whether to store at least a portion of the first data set in at least one of the plurality of client nodes based on the access frequency information.

2. The method of claim 1, further comprising:

obtaining, by the first client node or at least one delegate node of the first client node, aggregate access frequency information that aggregates the access frequency information across the first client node and the plurality of client nodes having at least a portion of the first data set; and

determining, by the first client node or the at least one delegate node, whether to store at least a portion of the first data set in an edge node based on the aggregate access frequency information, the edge node enabling communication between the first client node and a second network, wherein the first client node and a server node communicate over the second network.

3. The method of claim 2, further comprising:

obtaining, by the first client node or the at least one delegate node, access region information indicating an access region from which an access to the first data set in the first client node is made;

determining, by the first client node or the at least one delegate node, a first number of accesses made from access regions other than the region associated with the first client node based on the access region information; and

determining, by the first client node or the at least one delegate node, whether to store at least a portion of the first data set in the server node based on the first number of accesses.

4. The method of claim 3, further comprising:

obtaining, by the first client node or the at least one delegate node, aggregate access region information that aggregates the access region information across the first client node and the plurality of client nodes having at least a portion of the first data set;

determining, by the first client node or the at least one delegate node, a second number of accesses made from access regions other than the region associated with the first client node based on the aggregate access region information; and

determining, by the first client node or the at least one delegate node, whether to store at least a portion of the first data set in the server node based on the second number of accesses.

5. The method of claim 1, further comprising:

receiving, by the first client node, a search query; and

obtaining, by the first client node, a plurality of search results responsive to the search query, wherein the plurality of search results comprises a second data set stored in a second client node that communicates with the first client node over the first network, an edge node that enables communication between the first client node and a second network, and/or a server node that communicates with the first client node over the second network.

6. The method of claim 3, further comprising:

obtaining, by the first client node, at least one user-defined rule that specifies whether to store at least a portion of the first data set in at least one of the plurality of client nodes, the edge node, or the server node.

7. A non-transitory machine-readable storage medium comprising instructions executable by a processor of a client computing device for storing data in fog computing, the machine-readable storage medium comprising:

instructions to store a set of data in a first data storage coupled to the client computing device;

instructions to identify a peer client computing device to the client computing device;

instructions to determine whether at least a portion of the set of data should be stored in a second data storage coupled to the peer client computing device based on at least one parameter, wherein the at least one parameter comprises a first access frequency indicating a number of accesses to the set of data stored in the first data storage over a first time period; and

in response to determining that the at least portion of the set of data should be stored in the second data storage based on the at least one parameter, instructions for providing the at least a portion of the set of data to the peer client computing device for storage in the second data storage.

8. The machine-readable storage medium of claim 7, wherein the instructions to determine whether the at least a portion of the set of data should be stored in the second data storage based on the at least one parameter further comprises:

instructions to determine when the first access frequency reaches a first predetermined threshold; and

in response to determining that the first access frequency reaches the first predetermined threshold, instructions to provide the at least a portion of the set of data to the peer client computing device for storage in the second data storage.

9. The machine-readable storage medium of claim 7, further comprising:
instructions to identify an edge device that facilitates communication between the client computing device and a network;
instructions to determine when the first access frequency reaches a first predetermined threshold; and
in response to determining that the first access frequency reaches the first predetermined threshold, instructions to provide at least a portion of the set of data to the edge device for storage in a third data storage coupled to the edge device.
10. The machine-readable storage medium of claim 7, further comprising:
instructions to identify a server computing device that communicates with the client computing device via a network;
instructions to obtain a second access frequency, the second access frequency indicating a number of accesses to the set of data stored in the first data storage over a second time period that is longer than the first time period;
instructions to determine when the second access frequency reaches a first predetermined threshold; and
in response to determining that the second access frequency reaches the first predetermined threshold, instructions to provide at least a portion of the set of data to the server computing device for storage in a third data storage coupled to the server computing device.
11. The machine-readable storage medium of claim 7, wherein the at least one parameter comprises an access region, the method further comprising:
instructions to identify, for an access to the set of data stored in the first data storage, the access region from which the access is made;
instructions to identify a server computing device that communicates with the client computing device via the network;

instructions to determine when a number of accesses made from access regions outside of a geographical region associated with the client computing device reaches a first predetermined threshold; and

in response to determining that the number of accesses reaches the first predetermined threshold, instructions to provide at least a portion of the set of data to the server computing device for storage in a third data storage coupled to the server computing device.

12. A system for storing data in fog computing comprising:

a server computing device comprising at least one processor to:

receive a set of data for storage in a data storage coupled to the server computing device from a first client computing device via a network after the first client computing device determines that the set of data is to be sent from the first client computing device to the server computing device based on at least one parameter, wherein the at least one parameter comprises a frequency of accesses to the set of data in the first client computing device and an access region from which the set of data in the first client computing device is accessed; and

cause storing of the set of data in the data storage coupled to the server computing device.

13. The system of claim 12, wherein the processor is further to:

receive the set of data for storage in the data storage coupled to the server computing device from the first client computing device via the network after the first client computing device determines that the set of data should be sent from the first client computing device to an edge device based on the at least one parameter, wherein the edge device enables communication between the first client computing device and the network.

14. The system of claim 12, wherein the processor is further to:

receive the set of data for storage in the data storage coupled to the server computing device from the first client computing device via the network after the first client computing device determines that the set of data should be sent from the first client computing device to a peer client computing device based on the at least one parameter, wherein the peer client computing device communicates with the first client computing device over a peer-to-peer (P2P) network.

15. The system of claim 12, wherein the processor is further to:

receive, from a second client computing device, a request to search for the set of data; and

in response to the request, provide the set of data to the second client computing device.

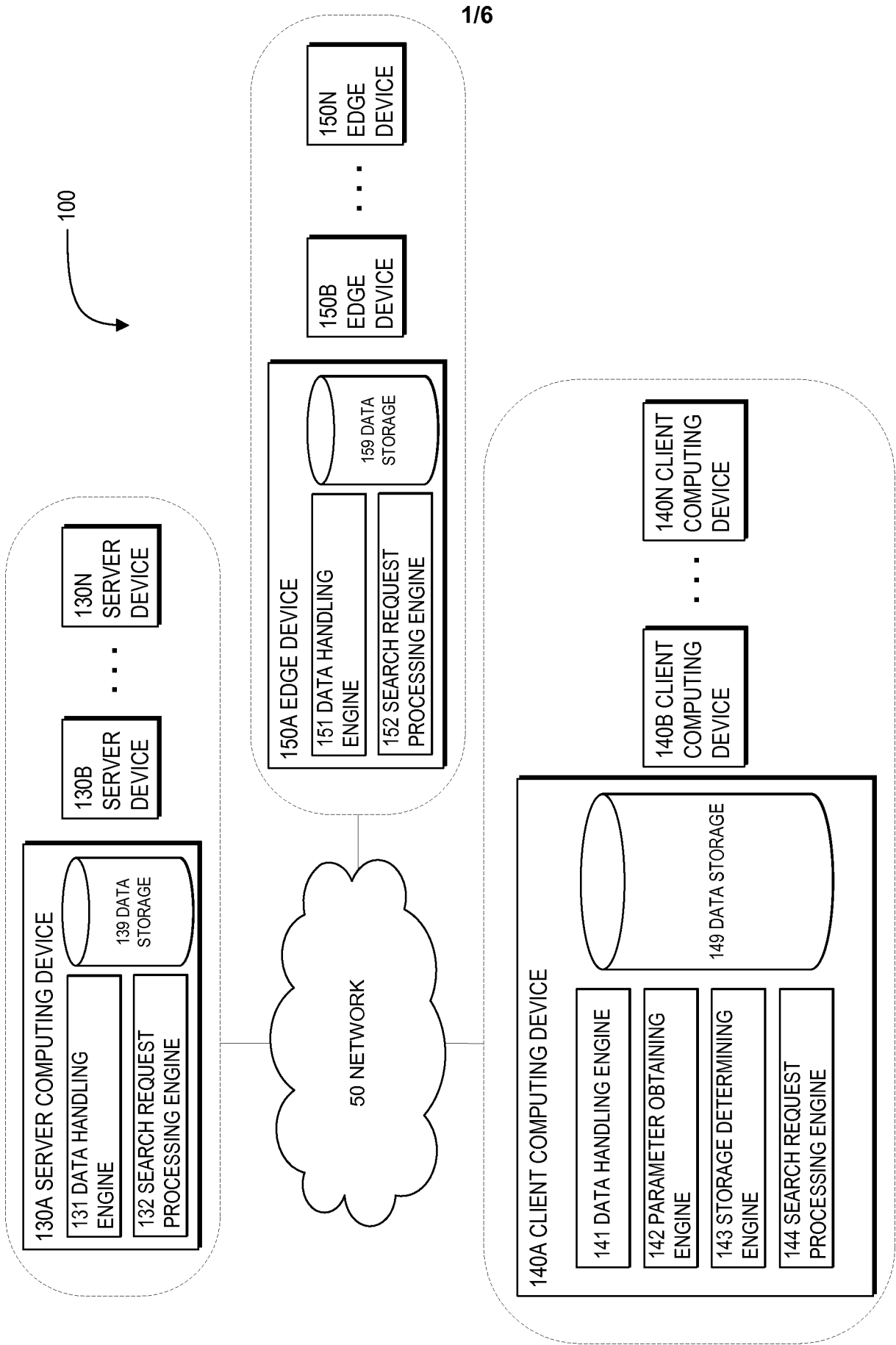
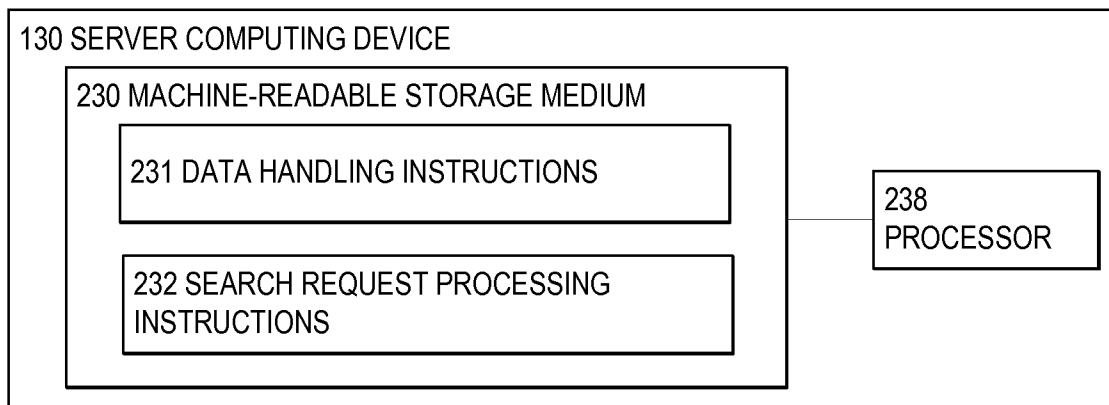
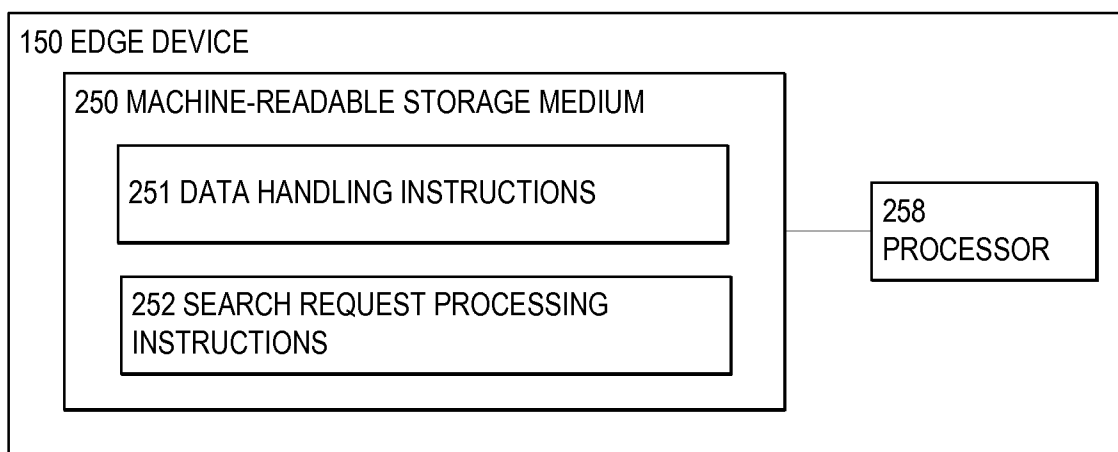
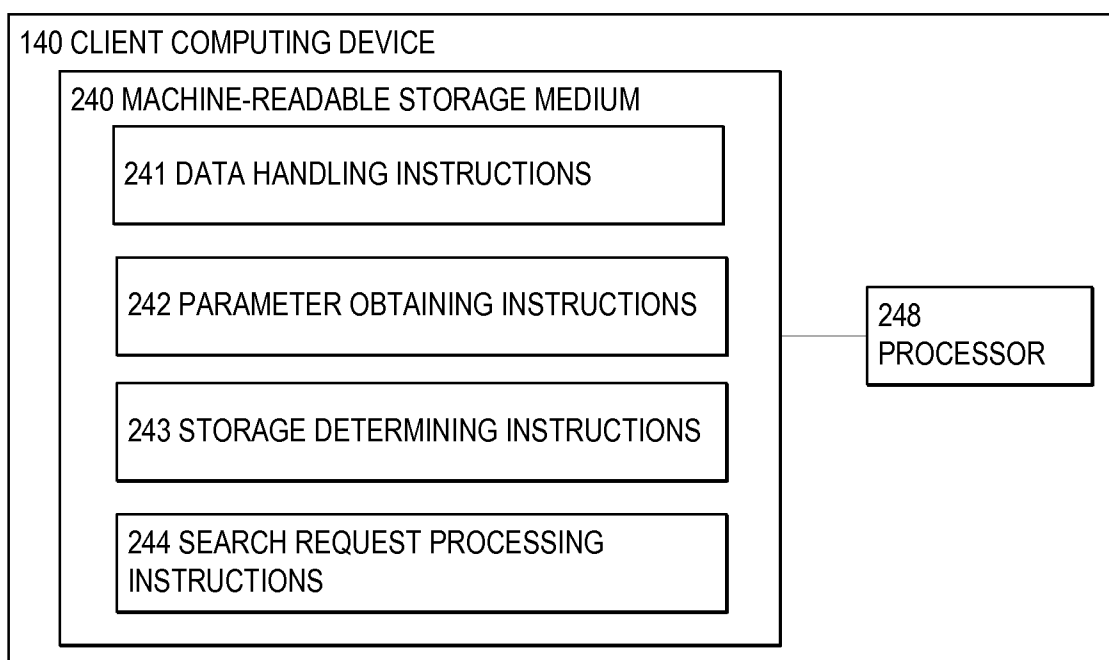
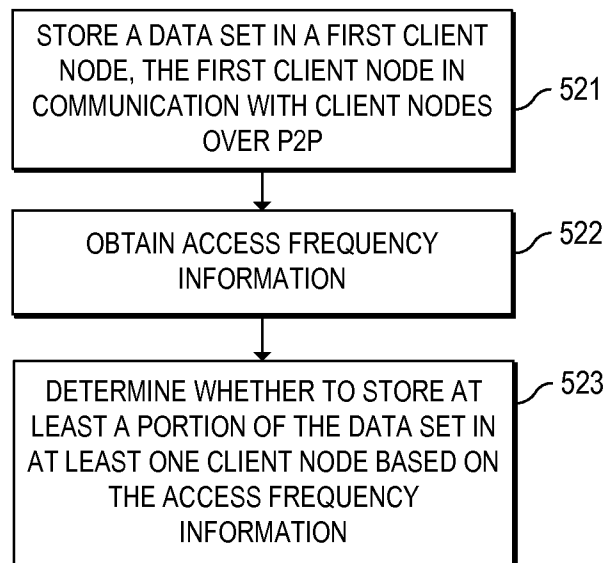


FIG. 1

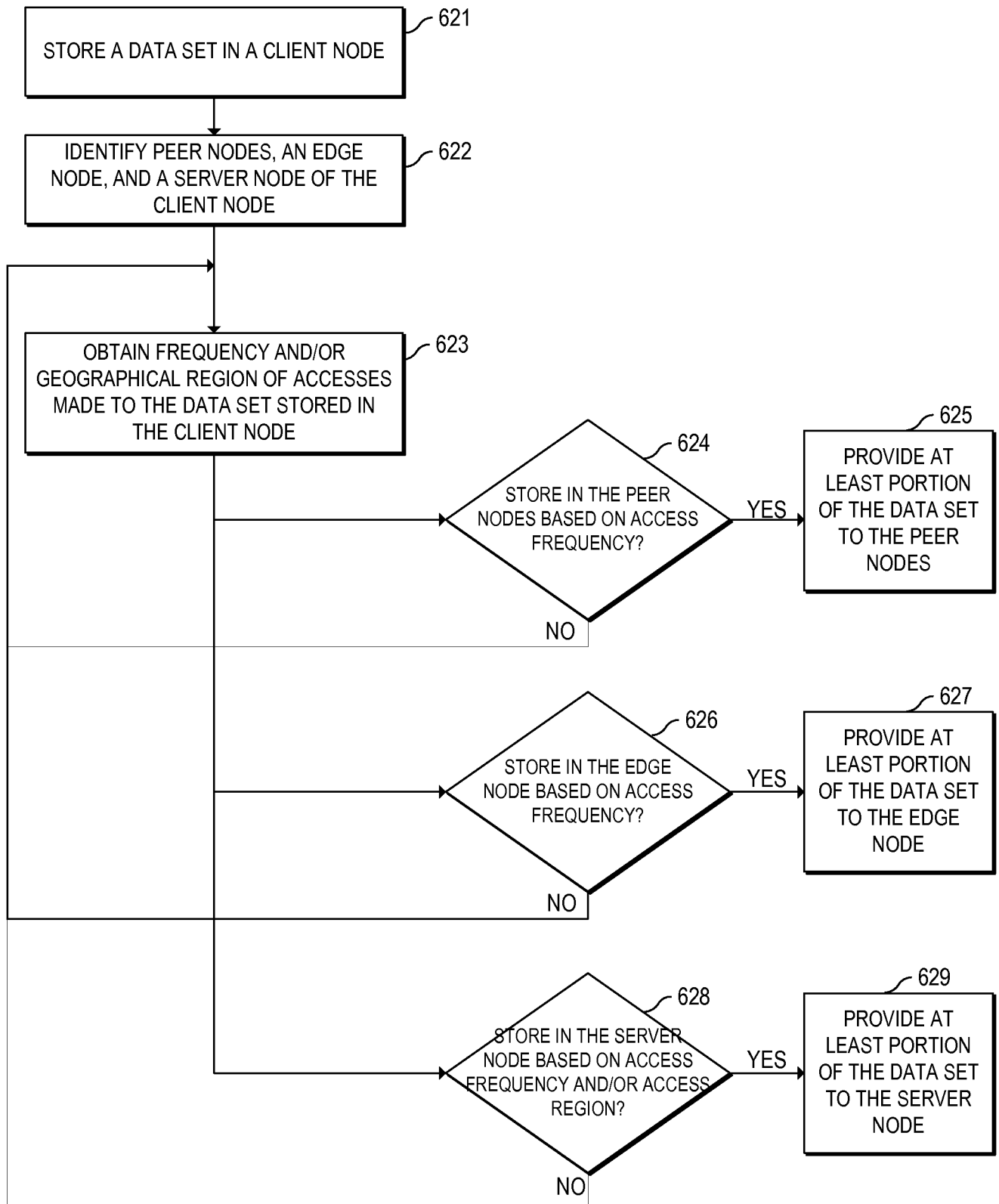
2/6

**FIG. 2****FIG. 3****FIG. 4**

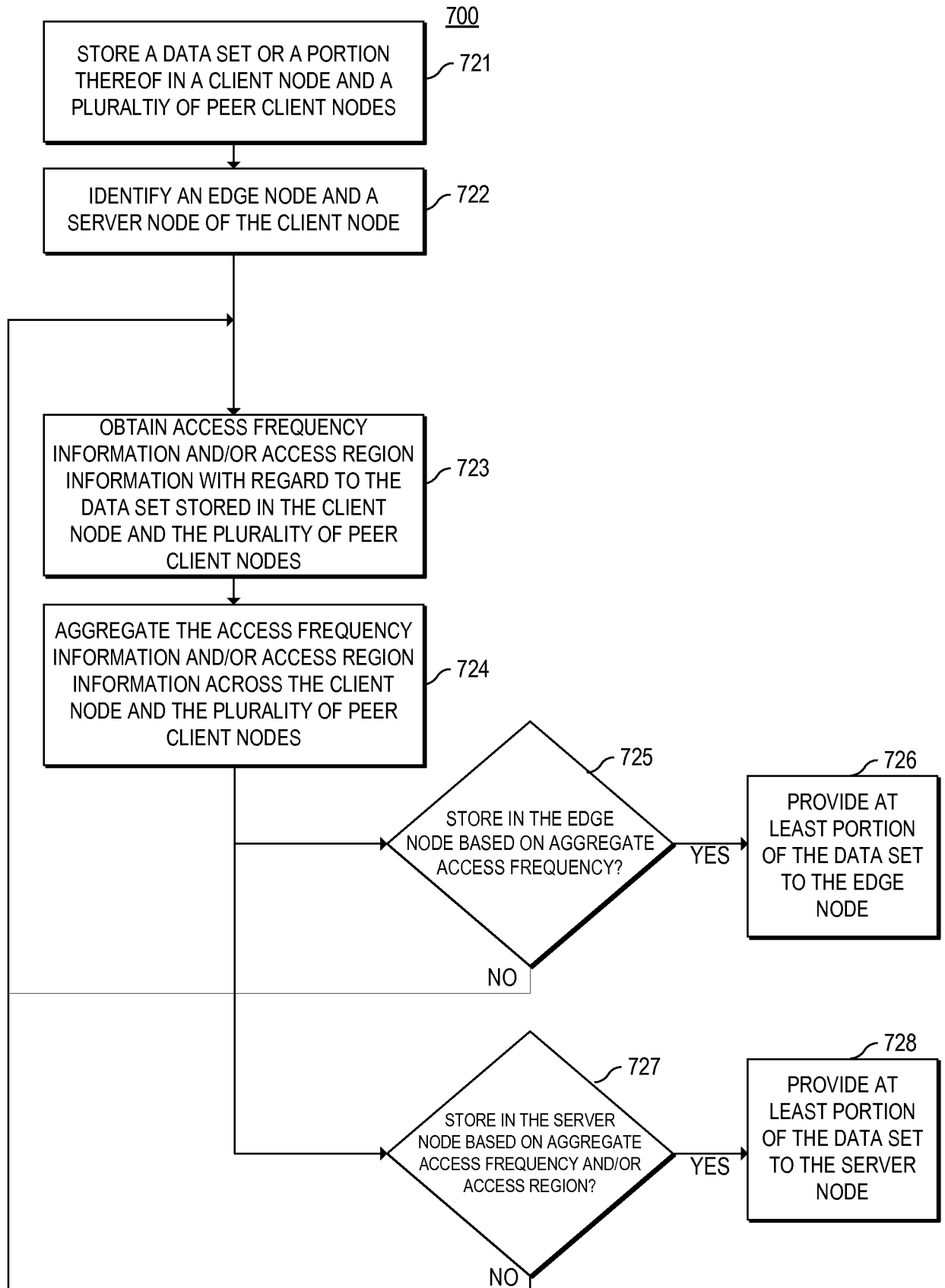
3/6

500**FIG. 5**

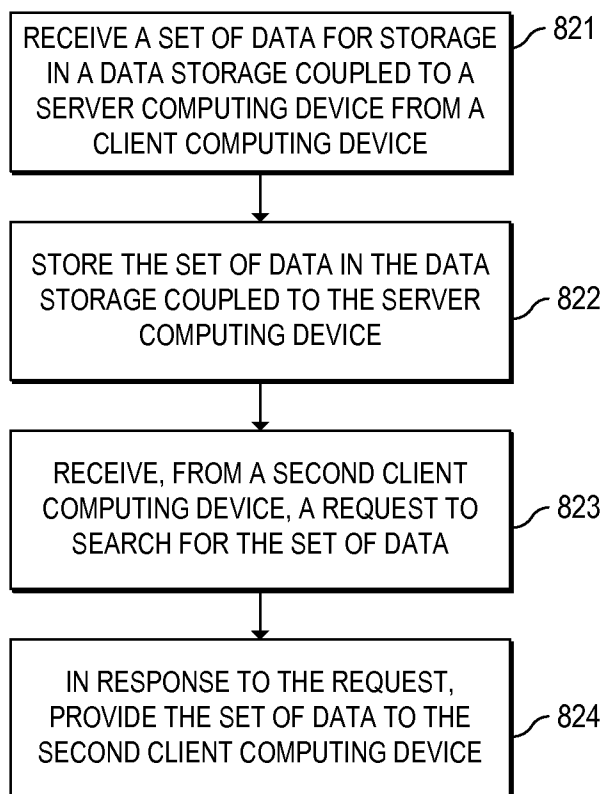
4/6

600**FIG. 6**

5/6



6/6

800**FIG. 8**

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2014/049022**A. CLASSIFICATION OF SUBJECT MATTER****G06F 15/16(2006.01)i, H04L 29/06(2006.01)i**

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHEDMinimum documentation searched (classification system followed by classification symbols)
G06F 15/16; G06F 15/173; G06F 17/00; G06F 7/00; G06F 17/30; H04L 12/56; H04L 29/06Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility modelsElectronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: location, frequency information, edge node, search result**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2011-0270839 A1 (JENSEN CRAIG et al.) 03 November 2011	1,5,7-9
A	See abstract; paragraphs [0026]-[0027], [0039]-[0043]; figure 1B, 3.	2-4,6,10-15
Y	EP 1443716 B1 (P. DIMITRI et al.) 02 May 2007	1,5,7-9
A	See abstract; paragraphs [0022]-[0029]; claims 1,2; figure 1.	2-4,6,10-15
A	US 8589356 B2 (UTZ BACHER et al.) 19 November 2013	1-15
	See abstract; Col.3 line 29 - Col.6 line 11; figure 1.	
A	US 8539098 B2 (ROBERT W. TWITCHELL, JR.) 17 September 2013	1-15
	See abstract; Col.18 line 6 - line 53; figure 5B.	



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

23 March 2015 (23.03.2015)

Date of mailing of the international search report

23 March 2015 (23.03.2015)

Name and mailing address of the ISA/KR

International Application Division
Korean Intellectual Property Office
189 Cheongsu-ro, Seo-gu, Daejeon Metropolitan City, 302-701,
Republic of Korea

Facsimile No. ++82 42 472 7140

Authorized officer

KYUNG, Youn Jeong

Telephone No. +82-42-481-3452



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2014/049022

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2011-0270839 A1	03/11/2011	US 2008-027905 A1	31/01/2008
		US 2011-087657 A1	14/04/2011
		US 2011-258186 A1	20/10/2011
		US 2014-108392 A1	17/04/2014
		US 7870128 B2	11/01/2011
		US 8645347 B2	04/02/2014
		US 8732148 B2	20/05/2014
		US 8892611 B2	18/11/2014
EP 1443716 B1	02/05/2007	AT 361611 T	15/05/2007
		DE 60313572 D1	14/06/2007
		DE 60313572 T2	27/12/2007
		EP 1443716 A1	04/08/2004
		US 2005-0021873 A1	27/01/2005
US 8589356 B2	19/11/2013	US 2011-153687 A1	23/06/2011
		US 2014-032618 A1	30/01/2014
US 8539098 B2	17/09/2013	US 2009-0106439 A1	23/04/2009
		US 2010-0009758 A1	14/01/2010
		US 2012-014389 A1	19/01/2012
		US 2012-016955 A1	19/01/2012
		US 2012-016956 A1	19/01/2012
		US 2012-016984 A1	19/01/2012
		US 2012-017005 A1	19/01/2012
		US 2012-017006 A1	19/01/2012
		US 2012-017008 A1	19/01/2012
		US 2012-017212 A1	19/01/2012
		US 2012-017265 A1	19/01/2012
		US 2012-020352 A1	26/01/2012
		US 2012-020353 A1	26/01/2012
		US 2012-023202 A1	26/01/2012
		US 2012-023244 A1	26/01/2012
		US 2012-023258 A1	26/01/2012
		US 2012-026889 A1	02/02/2012
		US 2012-030362 A1	02/02/2012
		US 2012-166653 A1	28/06/2012
		US 2012-272315 A1	25/10/2012
		US 7895348 B2	22/02/2011
		US 8341291 B2	25/12/2012
		US 8341292 B2	25/12/2012
		US 8352636 B2	08/01/2013
		US 8423664 B2	16/04/2013
		US 8429226 B2	23/04/2013
		US 8429293 B2	23/04/2013
		US 8433818 B2	30/04/2013
		US 8433819 B2	30/04/2013
		US 8447882 B2	21/05/2013
		US 8848704 B2	30/09/2014

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2014/049022Patent document
cited in search reportPublication
datePatent family
member(s)Publication
date

WO 2009-052452 A2

23/04/2009

WO 2009-052452 A3

02/07/2009