

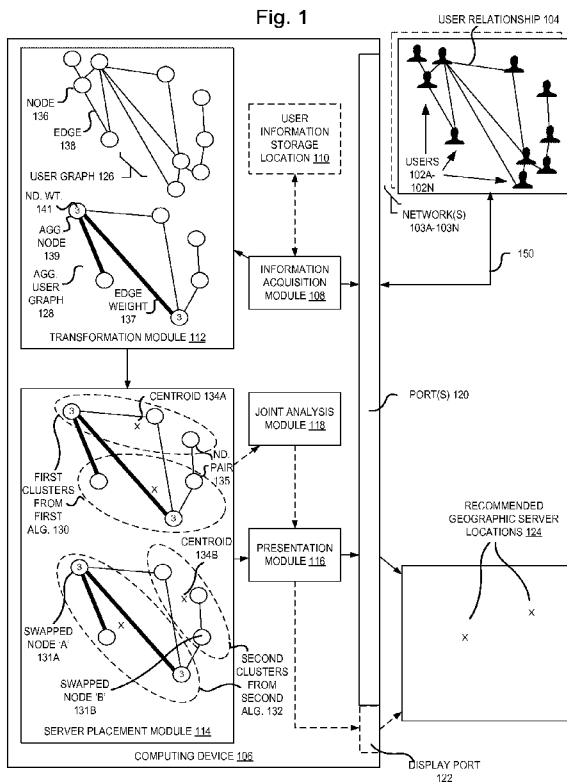


- (51) International Patent Classification:  
H04L 12/24 (2006.01) H04L 29/08 (2006.01)
- (21) International Application Number:  
PCT/IB2013/059633
- (22) International Filing Date:  
24 October 2013 (24.10.2013)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:  
13/684,125 21 November 2012 (21.11.2012) US
- (71) Applicant: TELEFONAKTIEBOLAGET L M ERICSSON (PUBL) [SE/SE]; S-164 83 Stockholm (SE).
- (72) Inventors: ZHANG, Ying; 5474 Midday Common, Fremont, California 94555 (US). LI, Du; 1004 Plover Lane, Palo Alto, California 94303 (US).
- (74) Agents: CAMERON, Michael G. et al.; 6300 Legacy, MS EVR 1-C-11, Plano, Texas 75024 (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).

[Continued on next page]

(54) Title: MULTI-OBJECTIVE SERVER PLACEMENT DETERMINATION



(57) Abstract: Methods and apparatus for determining recommended geographic server locations for online social networks by attempting to minimize user-server latency and inter-user communications latency. In an embodiment, geographic and relationship information for a plurality of users is acquired. The plurality of users may belong to one or more networks. The acquired information is transformed into a graph. A first plurality of clusters is generated with a first clustering algorithm. A second plurality of clusters is generated by iteratively examining pairs of the first plurality clusters, and swapping nodes between the examined clusters if it will reduce a total cut weight of the graph and locate each pair of nodes within a defined maximum distance from the centroid of the target cluster. In an embodiment, a method uses a joint analysis approach based upon characteristics of a plurality of existing networks.

WO 2014/080304 A2

**Published:**

- *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

## MULTI-OBJECTIVE SERVER PLACEMENT DETERMINATION

### FIELD

Embodiments of the invention relate to the field of networking; and more specifically, to the determination of recommended geographic server locations based upon characteristics of existing networks.

### BACKGROUND

A social network is a set of people (or organizations or other social entities) connected by a set of social relationships, such as friendship, co-working or information exchange relationship. There has been a recent unprecedented increase in the use of Online Social Networks (OSNs) to expand our social life, such as finding others of a common interest, discussing and sharing information in forums, and exchanging photos and personal news. The OSNs have become a large-scale distributed system providing services to hundreds of millions of users and delivering messages at very high rate.

Besides handling traditional client-to-server requests, OSNs also need to handle highly interconnected data due to the strong community structure and human relationships among their end users, which often results in complex data sharing among users. Given the tremendous user population and frequent data access by these users, effective resource planning and provisioning strategies are of extreme importance to the performance and revenue of an OSN. In particular, selecting the most suitable locations to deploy server farms is one of the key steps in such resource management.

The development of placement strategies for resources including servers and bandwidth affects the performance of any online web services. An appropriate allocation of resources benefits content providers by reducing latency for their clients and balancing the bandwidth consumption. The goal is to provide content distribution to clients with good Quality of Service (QoS) while retaining efficient and balanced resource consumption of the underlying network infrastructure. Thus, existing server placement proposals mainly focus on minimizing the average latency between the server and the users, given the nature of client/server communication patterns in traditional web services.

Many proposals on the server placement problem rely on extracting clients' requests from history traces collected on the web servers, and then searching for the best placement given the particular client and load distribution. While these proposals might be plausible in improving performance of existing OSN services, these proposals are less helpful to new OSNs that are starting afresh. Thus, a problem arises in that it is difficult for a new born Internet application service to make a decision on where to deploy its servers.

Moreover, much existing work on server placement casts the problem as an integer linear program where a binary decision variable  $b_{ij}$  is used to denote if user  $i$  is assigned to server  $j$ ; and the total number of selected servers should be a predetermined input  $M$ . One of the best known approximation algorithms for this problem (presented by M. Charikar and S. Guha in "Improved combinatorial algorithms for the facility location and  $k$ -median problems" in Proceedings of the 40<sup>th</sup> Annual Symposium on Foundations of Computer Science, 1999) achieves a very large time complexity of  $O((N + P)^3)$ , where  $N$  is the number of servers and  $P$  is the number of users. To make the problem more manageable in reality, a number of approximation and heuristics have been proposed such as the use of a greedy algorithm (by L. Qiu, V. N. Padmanabhan, and G. M. Voelker, in "On the Placement of Web Server Replicas," in Proc. of IEEE INFOCOM 2001, 1587-1596). However, these approaches are mainly based on theoretical analysis and are only validated using simulation in very small graphs. Further, such methods have fundamental issues in scaling to a large value of  $N$ . Accordingly, there is a need for methods to efficiently and flexibly determine, for existing or new OSNs and any value of  $N$ , where to place servers.

#### SUMMARY

According to one embodiment of the invention, a computer implemented method is described to determine a plurality of recommended geographic server locations by attempting to minimize both user-server latency and a latency of inter-user communications. The method includes acquiring geographic information for a plurality of users of a set of one or more networks. The geographic information for each of the plurality of users indicates a geographic location of that user. The method also includes acquiring relationship information for at least some of the plurality of users. The

relationship information indicates those of the plurality of users that are connected on a network of the set of networks. Next, the method includes transforming the geographic information and the relationship information into a graph including a plurality of nodes representing the plurality of users, and a plurality of edges connecting the plurality of nodes according to the relationship information. Each of the plurality of edges includes an edge weight. The method further includes generating a first plurality of clusters by performing a first clustering algorithm on the graph. Each cluster of the first plurality of clusters includes a centroid and a set of one or more nodes of the plurality of nodes. Each of the set of nodes is included in only one of the first plurality of clusters. The method further includes generating a second plurality of clusters by performing a second clustering algorithm. The second clustering algorithm includes iteratively examining pairs of clusters of the first plurality of clusters. Then, for each examined pair of clusters, the second clustering algorithm repeatedly swaps pairs of nodes between the pair of clusters when a swap of a pair of nodes will reduce a total cut weight of the graph, and locate each node of the pair of nodes, when swapped to the other cluster of the examined pair of clusters, within a defined maximum distance from the centroid of the other cluster to thereby bound user-to-server latency. The total cut weight is a sum of edge weights of edges that connect nodes in different clusters. The method further includes causing information describing geographic locations of centroids of the second plurality of clusters to be presented to a user as the plurality of recommended geographic server locations.

According to another embodiment of the invention, a computer implemented method is described to determine a plurality of recommended geographic server locations by attempting to minimize both user-server latency and a latency of inter-user communications using a joint analysis approach based upon characteristics of a plurality of networks. The method includes acquiring geographic information for a plurality of users of the plurality of networks. The geographic information for a user of the plurality of users indicates a geographic location of the user. The method also includes acquiring relationship information for at least some of the plurality of users. The relationship information indicates those of the plurality of users that are connected on at least one of the plurality of networks. The method further includes transforming the geographic information and the relationship information into a plurality of graphs.

Each graph of the plurality of graphs represents one network of the plurality of networks. Each graph of the plurality of graphs includes a plurality of nodes representing those users of the plurality of users that belong to the one network, and a plurality of edges connecting the plurality of nodes according to the relationship information. Each of the plurality of edges includes an edge weight. The method also includes generating, for each graph of the plurality of graphs, a plurality of clusters for that graph by performing a clustering algorithm. Each of the plurality of clusters includes a centroid. The method further includes identifying a first recommended geographic server location by ranking a set of centroids according to the frequency of occurrence of each centroid in the set of centroids (where the set of centroids includes all of the centroids of the plurality of clusters of each of the plurality of graphs), and identifying a centroid of the ranked set of centroids having the highest occurrence as representing the first recommended geographic server location.

In an embodiment of the invention, a server end station is described that is configured to determine a plurality of recommended geographic server locations by attempting to minimize both user-server latency and a latency of inter-user communications. The server end station includes an information acquisition module. The information acquisition module is configured to acquire geographic information for a plurality of users of a set of one or more networks. The geographic information for each user of the plurality of users indicates a geographic location of that user. The information acquisition module is also configured to acquire relationship information for at least some of the plurality of users. The relationship information indicates those of the plurality of users that are connected on at least one network of the set of networks. The server end station also includes a transformation module configured to transform the geographic information and the relationship information into a graph including a plurality of nodes representing the plurality of users and a plurality of edges connecting the plurality of nodes according to the relationship information. Each of the plurality of edges includes an edge weight. The server end station also includes a server placement module. The server placement module is configured to generate a first plurality of clusters by performing a first clustering algorithm on the graph. Each cluster of the first plurality of clusters includes a centroid and a set of one or more nodes of the plurality of nodes. Each node of the set of nodes is included in only one

cluster of the first plurality of clusters. The server placement module is further configured to generate a second plurality of clusters by performing a second clustering algorithm. The second clustering algorithm includes iteratively examining pairs of clusters of the first plurality of clusters. The second clustering algorithm also includes  
5 for each examined pair of clusters, repeatedly swapping pairs of nodes between the pair of clusters when a swap of a pair of nodes will reduce a total cut weight of the graph (wherein the total cut weight is a sum of edge weights of edges that connect nodes in different clusters) and locate each node of the pair of nodes, when swapped to the other cluster of the examined pair of clusters, within a defined maximum distance from the  
10 centroid of the other cluster to thereby bound user-server latency. The server end station further includes a presentation module configured to cause information describing geographic locations of centroids of the second plurality of clusters to be presented to a user as the plurality of recommended geographic server locations.

#### BRIEF DESCRIPTION OF THE DRAWINGS

15 The invention may best be understood by referring to the following description and accompanying drawings that are used to illustrate embodiments of the invention. In the drawings:

Figure 1 illustrates a system for determining a plurality of recommended geographic server locations using multiple objectives according to one embodiment of  
20 the invention;

Figure 2 illustrates a “joint before” approach for determining a plurality of recommended geographic server locations using aggregated user graphs from a plurality of networks according to one embodiment of the invention;

Figure 3 illustrates portions of a “joint after” approach for determining a  
25 plurality of recommended geographic server locations using centroids from within clusters from different network graphs according to one embodiment of the invention;

Figure 4 illustrates additional portions of the “joint after” approach presented in Figure 3 according to one embodiment of the invention;

Figure 5 illustrates a high-level view of an approach for determining a plurality  
30 of recommended server locations according to one embodiment of the invention;

Figure 6 illustrates two possible cost functions useful in particular clustering algorithms according to one embodiment of the invention;

Figure 7 illustrates three server placement algorithms according to one embodiment of the invention;

5           Figure 8 illustrates a flow for determining a plurality of recommended geographic server locations by attempting to minimize both user-server latency and a latency of inter-user communications according to one embodiment of the invention; and

10           Figure 9 illustrates determine a plurality of recommended geographic server locations by attempting to minimize both user-server latency and a latency of inter-user communications using a joint analysis approach based upon characteristics of a plurality of networks according to one embodiment of the invention.

#### DESCRIPTION OF EMBODIMENTS

15           In the following description, numerous specific details are set forth. However, it is understood that embodiments of the invention may be practiced without these specific details. In other instances, well-known circuits, structures and techniques have not been shown in detail in order not to obscure the understanding of this description.

20           References in the specification to “one embodiment,” “an embodiment,” “an example embodiment,” etc., indicate that the embodiment described may include a particular feature, structure, or characteristic, but every embodiment may not necessarily include the particular feature, structure, or characteristic. Moreover, such phrases are not necessarily referring to the same embodiment. Further, when a particular feature, structure, or characteristic is described in connection with an embodiment, it is submitted that it is within the knowledge of one skilled in the art to effect such feature, structure, or characteristic in connection with other embodiments  
25           whether or not explicitly described.

30           In the following description and claims, the terms “coupled” and “connected,” along with their derivatives, may be used. It should be understood that these terms are not intended as synonyms for each other. “Coupled” is used to indicate that two or more elements, which may or may not be in direct physical or electrical contact with each other, co-operate or interact with each other. “Connected” is used to indicate the

establishment of communication between two or more elements that are coupled with each other.

An electronic device (e.g., an end station, a network element, a computer / computing system / computing device) stores and transmits (internally and/or with  
5 other electronic devices over a network) code (composed of software instructions) and data using computer-readable media, such as non-transitory tangible computer-readable media (e.g., computer-readable storage media such as magnetic disks; optical disks; read only memory; flash memory devices) and transitory computer-readable  
10 transmission media (e.g., electrical, optical, acoustical or other form of propagated signals – such as carrier waves, infrared signals). In addition, such electronic devices typically include a set of one or more processors coupled to one or more other components, such as one or more non-transitory machine-readable media (to store code and/or data), user input/output devices (e.g., a keyboard, a mouse, a touchscreen, and/or a display), and network connections (to transmit code and/or data using propagating  
15 signals). The coupling of the set of processors and other components is typically through one or more busses and bridges (also termed as bus controllers). Thus, a non-transitory computer-readable medium of a given electronic device typically stores instructions for execution on one or more processors of that electronic device. One or more parts of an embodiment of the invention may be implemented using different  
20 combinations of software, firmware, and/or hardware.

As used herein, a network element (e.g., a router, switch, bridge) is a piece of networking equipment, including hardware and software, which communicatively interconnects other equipment on the network (e.g., other network elements, end stations). Subscriber end stations are computing devices (e.g., servers, workstations,  
25 laptops, netbooks, palm tops, mobile phones, smartphones, multimedia phones, Voice Over Internet Protocol (VOIP) phones, user equipment, terminals, portable media players, GPS units, gaming systems, set-top boxes) that access content/services provided over the Internet and/or content/services provided on virtual private networks (VPNs) overlaid on (e.g., tunneled through) the Internet. The content and/or services  
30 are typically provided by one or more end stations (e.g., server end stations) belonging to a service or content provider or end stations participating in a peer to peer service, and may include, for example, public webpages (e.g., free content, store fronts, search

services), private webpages (e.g., username/password accessed webpages providing email services), and/or corporate networks over VPNs. Typically, subscriber end stations are coupled (e.g., through customer premise equipment coupled to an access network (wired or wirelessly)) to edge network elements, which are coupled (e.g.,  
5 through one or more core network elements) to other edge network elements, which are coupled to other end stations (e.g., server end stations).

An online social network (OSN) is an online service, platform, or website that facilitates the building of social networks or social relations among its users. Some OSNs are general purpose and provide social networking services for a wide variety of  
10 users, while others provide such services for users having particular interests, such as users with common hobbies, activities, backgrounds, or real-life connections. OSNs often include a representation of each user (often a profile, including biographic data, photographs, and other information) and a set of his/her social links (e.g. connections to other users, organizations, entities, etc.). Many OSNs are web-based and provide means  
15 for users to interact through the use of computing devices and the Internet. Certain embodiments of the invention described herein involve server placement for OSNs; however, other embodiments of the invention apply to server placement for other scenarios such as networks of users, including but not limited to users of a cellular network, wired network, wireless network, etc.

20 Embodiments of the invention utilize a scalable server placement algorithm based upon graph partitioning. Some embodiments employ clustering techniques that partition the whole client space into non-overlapping groups (or clusters) according to a dual purpose, wherein user locations in a cluster are both topologically close to the centroid of the cluster (i.e. which indicate the best suitable locations to minimize user-  
25 to-server latency) and wherein each centroid is topologically close to other centroids (i.e. which indicate the best suitable locations to minimize server-to-server latency). Embodiments of the invention, in implementing the scalable server placement algorithm, take inter-user data sharing / communication into consideration. Some embodiments also utilize publicly available social network data to make better resource  
30 provisioning decisions for social networks that do not currently exist.

Embodiments of the invention help improve the performance and reduce operational costs of future (i.e. not yet existing) as well we current online social

network services. Use of the invention may also reduce energy consumption and carbon emission of such services, especially at their hosting data centers and the communication networks, and help make better usage of computing resources and preserve natural resources.

5           Although there has been some research into server placement algorithms, the following three challenges have been unsolved by any existing work. First, existing work has not been computationally scalable to accommodate the selection of any number of server locations. Most existing work formulates the server placement as a facility location problem: choosing  $M$  replicas among  $N$  potential sites ( $N > M$ ) such  
10 that some objective function is optimized under a given traffic pattern. However, the assumption of having a candidate pool, i.e.,  $N$  potential sites, may not hold in reality. Especially with the growth of cloud computing, platform and data centers, the choices of hosting resources are becoming more and more pervasive. Further, telecommunications companies have started building their own cloud computing  
15 platforms. Under the emerging trend of available computing resources in cloud platforms, the cost for web providers has been significantly reduced while the server selection range has an order of magnitude increase. Moreover, most of the solutions do not scale well with the number of server candidates or the number of clients. Existing work that relies upon optimization software to find a feasible solution can only handle  
20 thousands of nodes at most. Thus, the effectiveness and the accuracy of such solutions are inherently limited by the candidate sets.

          Second, existing work has not considered inter-user data sharing. In OSNs, communications are more complex than in traditional flat client-server scenarios, as they include reads and writes across particular users, such as the writing/reading of a  
25 friend's user profile page, which is dubbed "inter-user access to shared data." The performance of such communication is as important as the client/server access, and is not accounted for in most server placement algorithms. The user experience is degraded if updates to shared data are not quickly reflected to other clients. Moreover, the inter-user communications include users scattered worldwide, contributing another factor to  
30 consider in server selection. Besides the concerns on the user perceived performance, data sharing may also have impact on the OSN operational cost. Data sharing between users in different locations can introduce significant network bandwidth cost to the

OSN providers. For example, updating a user profile for one user may trigger notifications to be sent to multiple other users. In this case, the subscriptions between connected users triggers the data transfer among storage entities within the OSN. When these servers reside in different sites, the traffic will have to traverse the Internet, causing additional inter-site network consumption and/or congestion, delay, and additional bandwidth cost. When selecting servers to minimize cost, such a factor should be taken into consideration. Therefore, these connections fundamentally transform the problem's mathematics: in addition to connections between clients and the servers to store their data, there are connections in the communication graph between users' data.

Finally, existing approaches to server selection do not consider the fact that, for a new network service (such as an OSN), there is not any available server data to harvest when making decisions. However, new OSN operators need to make the decision of where to place the servers even before the system is in production. No existing work has specified how to solve this forecasting problem.

Certainly there are many different factors causing diverse user distribution, request patterns and inter-user data sharing. However, OSNs are believed to reflect the real-life social relationships of people. Therefore, there are some factors that have been common among most OSNs. Users may be connected (i.e. have a "relationship") based on one relation only, e.g., as members of the same organization. Alternatively, users may maintain a multiplex tie based on many relations, such as sharing information, giving financial support and attending conferences together. These ties reflect the real-life social relationships of people and person-to-person interaction. Moreover, a person's social personality is likely to stay consistent across communities and environments. For instance, a user who is active in one social network is likely to be active in another.

Embodiments of the invention utilize public information from existing, established Online Social Networks (OSNs), which provides sufficient information to provide intelligent server placement suggestions to new born social network applications.

Embodiments of the invention utilize a solution that is formulated as an optimization problem of minimizing a number of required replicas. For this purpose,

we use the following notation. Let  $G = (V, E)$  denote the social graph of an OSN, with node set  $V$  representing users, and edge set  $E$  representing friendship relationships among users. In an embodiment, social graph  $G$  is an undirected symmetric graph with weighted edges, i.e., if  $(u, v) \in E$ , then  $(v, u) \in E$ . Each edge is associated with a weight  $w_{ij}$  representing the number of relationship between user  $u$  and  $v$ . Thus,  $w_{ij} = 1$  if the two nodes stand for a single user. If each node in  $G$  represents a group of users,  $w_{ij}$  is the number of pairwise friendships between this group of users. The objective is to find a subset  $M$  of the locations from all candidate nodes  $N = |V|$ , and connect each user to the server so that the total cost is minimized.

To support the scalability with a large  $N$  value, embodiments of the invention solve the problem as a graph partitioning or clustering problem. A clustering of a graph is a partition  $P$  of  $V$ , which includes a set of clusters  $C_0, C_1, \dots, C_k$  --- i.e.,  $\forall i, j : C_i \cap C_j = \emptyset$ , and  $C_0 \cup C_1 \cup \dots \cup C_k = V$ . A cost function  $f$  assigns a real number to any given clustering of  $G$ . The goal is to find a clustering that minimizes a given cost function. For instance, the cost function could be the sum of the distance between each node and its centroid, or it could be the negative sum of edge weights between clusters. Two common ways to partition a graph include using an agglomerative approach, which initializes each element to belong to its own cluster and proceeds to merge clusters until a certain terminating condition is met, and using a partitive clustering approach that starts with a single cluster containing all elements and proceeds by splitting clusters.

Embodiments of the invention utilize graph partitioning processes that include optimizing for multiple objectives. Multi-objective optimization problems are minimization problems that work with a set of cost functions  $F = (f_1, f_2, \dots, f_n)$ . The goal is to find the set  $S$  of Pareto-optimal solutions. Embodiments of the invention define two objectives to capture both latency and inter-site traffic cost, although in other embodiments these formations can be easily modified to account for other types of performance metrics and costs.

In embodiments, the center of any cluster "C" is defined as the following, where  $v$  is used to enumerate all nodes in the cluster:

$$\text{Centroid}(C) = \frac{1}{|C|} * \sum_{v \in C} v$$

To satisfy the requirements for OSN server placements, embodiments of the invention utilize one or more of the two cost functions, which are detailed in Figure 6. First is an intra-cluster latency cost function 602:

$$f = \sum_{C \in P} \sum_{u \in C} \text{dist}(u, \text{centroid}(C))$$

5 Second is an inter-cluster traffic cost function 604:

$$g = - \frac{1}{|V|} \sum_{u \in V} \frac{1}{h} \sum_{j=1} \delta(u, \text{nei}_{uj})$$

In the above functions, P is a given partition solution, and  $\text{dist}(u, v)$  is the weight of edge  $(u, v)$ , which is defined to be the user-to-centroid latency. In the inter-cluster traffic cost function 604,  $\text{nei}_{uj}$  is defined as node u's most connected  $j^{\text{th}}$  neighbor according to the definition of connections among users. The value j is used to enumerate all of u's neighbors. The value  $\delta(u, v)$  is a binary variable to denote if u and v are in the same partition. Note that the inter-cluster traffic cost function 604 "g" measures inter-cluster connectivity to be minimized, while intra-cluster latency cost function 602 "f" measures intra-cluster difference, which should also be minimized.

10 Taken together, they do not favor trivial clusterings where all nodes are assigned to the same cluster, or each node is assigned to its own cluster. While this description focuses upon the above two objective functions 602, 604, in other embodiments the functions are modified to describe other cost functions such as energy bills, cooling spending, and infrastructure leasing costs, thereby allowing for such costs to also be considered.

20 Figure 1 illustrates a system for determining a plurality of recommended geographic server locations using multiple objectives according to one embodiment of the invention. This system represents one embodiment of the invention that implements the general process depicted in Figure 5, which illustrates a high-level view of an approach for determining a plurality of recommended server locations. From this high-level view of one embodiment, data from one or more OSNs (502A-502N) is acquired by an information acquisition module 108 using a set of ports 120 (e.g. physical network interfaces) and transformed into one or more graphs by a transformation module 112. Next, a server placement module 114, which utilizes one or more clustering algorithms (highlighted above) to cluster the graphs, which may generate one

25 level view of one embodiment, data from one or more OSNs (502A-502N) is acquired by an information acquisition module 108 using a set of ports 120 (e.g. physical network interfaces) and transformed into one or more graphs by a transformation module 112. Next, a server placement module 114, which utilizes one or more clustering algorithms (highlighted above) to cluster the graphs, which may generate one

30 or more recommended geographic server locations based upon the clustering. Following 514, a presentation module 116 may display the results to a user or transmit

the results so that a user is so informed. In embodiments, this transmission is one or more of transmitting the results to a display, transmitting the results to a separate device for viewing or further processing, or transmitting the results to another device (e.g., a printer) to create a physical document or element. In other embodiments, the presentation module 116 is enabled to automatically or semi-automatically reserve or allocate server resources at the recommended geographic server locations, such as by enabling one or more virtual machine and/or network instances at or near the recommended geographic server locations. Instead of following line 514 to the presentation module 116, the process may also continue to a joint analysis module 118 configured to utilize data from multiple OSNs when determining recommended geographic server locations. In some embodiments, the joint analysis module 118 may include functionalities of transformation module 112 or server placement module 114, which will allow the joint analysis module 118 to generate its own results, which are shifted to the presentation module 116. However, following 516, the joint analysis module 118 may cause the server placement module 114 to be used again. In some embodiments, the joint analysis module 118 or server placement module 114 may immediately (see line 512) be invoked with OSN data (502A-102N). This may occur, in some embodiments, using OSN data (502A-102N) stored by a user information storage location 110 of a computing device 106, as depicted in Figure 1.

Turning back to Figure 1, an information acquisition module 108 of the computing device 106 acquires 150 information about a plurality of users 102A-102N and user relationships 104 of one or more networks 103A-103N. In an embodiment, the networks 103A-103N are OSNs. In an embodiment, this acquired information includes a set of user profiles from the OSNs, as well as a list of other users with whom each user has relationships with on the network (i.e. "friends" of each user, or some other designation of connected users). In embodiments of the invention, the acquired information is maintained in a user information storage location 110 of the computing device 106.

In embodiments of the invention, the transformation module 112 attempts to deduce a geographic location from the user profile for each user (e.g. 102A). Many users explicitly register their current location with the network, while many users enter their birth location, and yet others enter affiliations that reveal a user geographic

location. Interestingly, a non-trivial fraction of users' locations are automatically entered by the smart phone applications in the form of longitude and latitude. In an embodiment, the transformation module 112 first retrieves all possible hints on location from the crawled data. The transformation module 112 may then pre-process the  
5 crawled data to correct typographical errors, eliminate ambiguity, and combine any same location having multiple representations into one (e.g., "California" and "CA" and "Cal."). In an embodiment, the transformation module 112 then translates the user location strings into latitude-longitude coordinates using a geocoding database (internal or external to the computing device 106) or a geocoding API, including but not limited  
10 to APIs such as the Google Geocoding API or the Yahoo! Maps Web Services Geocoding API.

In an embodiment, the acquired information and the deduced geographic coordinates are transformed by the transformation module 112 into a user graph 126 having a plurality of nodes 136 and a plurality of edges 138 connecting some of those  
15 nodes. In an embodiment, the location of each node 136 is based upon the deduced geographic coordinates of the users, and the edges 138 exist between any two nodes 136 that represent users that have a connection on the OSN or OSNs (based upon the acquired user relationship 104 data).

In an embodiment, a latency map is constructed between any pair of users.  
20 While in one embodiment, each latency value is a round-trip delays collected for all users. However, in many embodiments due to limited access to end hosts at a large scale, a hypothetical direct link latency is calculated to approximate the latency between two users using the deduced geographic locations. Thus, in an embodiment, the latency between two users is the transmission latency of a hypothetical direct link  
25 (e.g. optical, metallic) between those users. In an embodiment, locations are treated as points and are clustered into regions using the well-known Vincenty's formula, which is used to calculate geodesic distances between points. (Except for networks that use circuit-switching, geographic distance correlates well with the minimum network delay. This method is more robust than extracting link latencies from trace-route data; such  
30 latencies can be contaminated by network queuing and unknown asymmetric reverse path delay.)

In an embodiment, representing each individual user as one node 136 in the user graph 126 would result in tremendously huge graph. Moreover, users physically near each other will share similar network topological properties. Therefore, in an embodiment of the invention users are aggregated to groups to form an aggregated user graph 128. In an embodiment, users located within a range of latitude and longitude from each other (e.g., within 2 degrees of latitude and longitude) are grouped together as one aggregated node 139. Such aggregation has a much finer granularity than performing grouping based on cities or even zip codes. Accordingly, in an embodiment a node weight 141 is assigned to each node (e.g. 139) based upon the number of users in the group represented by the node 139. In the depicted example of Fig. 1, the top left aggregated node 139 contains a node weight 141 of '3' because there are three nodes from the original user graph 126 that were located within a particular distance of each other and thus were aggregated together. In an embodiment, an edge weight 137 is assigned to an edge that represents the total number of relationships (e.g. friendship connections) between users in the two connected groups. These weights are represented in Figure 1 using thick lines (for heavier comparative weights) or thin lines (for lighter/smaller comparative weights), but in embodiments of the invention the weights are integers, real numbers, binary values, or the like.

After the aggregated user graph 128 is constructed by the transformation module 112, the server placement module 114 utilizes the aggregated user graph 128 to determine recommended geographic server locations 124 using a variety of placement algorithms. While in one embodiment an ultimate goal is to minimize user-to-server latency and a latency of inter-user communications simultaneously, the following description presents three separate algorithms for determining the recommended geographic server locations 124, which may be more suited for certain applications based upon the constraints and preferences involved: one algorithm for minimizing user-to-server latency 702, one algorithm for minimizing a latency of inter-user communications 704, and one algorithm to optimize for both 706.

Figure 7 illustrates these three server placement algorithms according to one embodiment of the invention. In an embodiment, the server placement module 114 is configured to utilize an algorithm for primarily minimizing user-to-server latency 702.

In an embodiment, the algorithm 702 is based upon a k-means or k-medoids clustering algorithm, but with several enhancements. The algorithm 702 is reproduced below:

**procedure** *OSN-KMEANS* ( $G=(V,E),k$ )

- 1: Initialize cluster centroid  $C = c_1, c_2, \dots, c_k$
- 2: Assign points to closest centroid  $c_i$  according to  $P_i := \operatorname{argmin} \|x_j - c_i\|^2$
- 3: Find new centroid  $C' = c'_1, \dots, c'_k$  to find  $c'_i = \frac{1}{|P_i|} \sum_{x_j \in P_i} x_j$
- 4: IF  $C' \neq C$
- 5:  $C = C'$ , GOTO 2
- 6: Compute the client to server latency between each point and its assigned centroid
- 7: Compute the friendship links across servers

To summarize, the server placement module 114 begins with  $k$  elements as the centroid. In an embodiment, the value of  $k$  is selected by the user to signify the number of recommended geographic server locations 124 that are sought by the user. With the centroids selected, the server placement module 114 iteratively selects a new centroid to minimize the distance between samples in the same group to the centroid of the group. Given a centroid of cluster  $C_i$ , all the points  $U_j$  belonging to this cluster are computed with center  $c_i$  in step 2. At step 3, the server placement module 114 re-computes the new center  $C'$  for each cluster  $P_i$  according to the new grouping of  $x_j$ . The complexity of this algorithm is  $O(nk\lambda)$ , where  $k$  is the final number of clusters,  $n$  is the total data points, and  $\lambda$  is the number of iterations.

In an embodiment, the value  $\lambda$  has been significantly reduced through a careful selection of  $k$  and the initial set based on realistic conditions. Note that the server placement module 114 doesn't simply apply the classic k-means or k-medoids style clustering solutions blindly, but carefully selects the initial sets according to our problem context. In an embodiment, the server placement module 114 starts the selection of known data center locations of popular OSNs and uses them as the initial candidates (i.e. centroids) for the algorithm, which efficiently reduces the number of iterations of clustering. Additionally, in an embodiment, for each  $k$ , the server placement module 114 always includes the set of servers selected in the  $k - 1$  experiment in the initial set. The remaining is selected based on ranking of user

population. Moreover, in an embodiment the algorithm takes the sample density information (e.g., the weights) into consideration, instead of treating each data point as equivalent.

In an embodiment, the server placement module 114 is configured to utilize an algorithm for primarily minimizing a latency (or, an average latency) of inter-user communications 704. Intuitively, if graph edges represent relationships between vertices, then a goal is to achieve many edges within clusters and few edges between clusters. However, if the cost function is set to be the number of inter-cluster edges, then the problem of minimizing it is trivial, i.e., picking the clustering that contains a single cluster. Thus, in addition to the minimum cut requirement, the algorithm 704 requires that the partition needs to be as balanced as possible. The balance property can also help providing balanced load and best resource utilization. For example, if one best location is used to serve all the users, it may easily create bandwidth bottlenecks. In the following description, the terms “cuts”, “inter-cluster edges” and “inter-site traffic” are utilized interchangeably.

In an embodiment, the algorithm 704 is based upon a minimum-cut clustering algorithm with modifications. The algorithm 704 is reproduced below:

```

procedure OSN-MINCUT ( $G=(V,E),k$ )
1: Initialize bisection partition  $P = \{C_1, C_2\}$ 
2: repeat
3: for any pair of unmarked ( $v_1 \in C_1; v_2 \in C_2$ ) do
4:   compute  $h(v_1, v_2) = h(v_1) + h(v_2) - 2w(v_1, v_2)$ 
5: end for
6: Find  $(v_1, v_2) = \operatorname{argmax} h(v_1, v_2)$ : swap and mark  $v_1, v_2$ 
7: until  $\forall v_1, v_2, h(v_1, v_2) < 0$  — iter=ITERmax — all nodes marked
8: if  $k > 2$  then
9:   OSN-MINCUT( $G_1=(C_1, E_1), k - 1$ )
10:  OSN-MINCUT( $G_2=(C_2, E_2), k - 2$ )
11: end if

```

Given an initial bisection, the server placement module 114 searches to find a sequence of node pair exchanges that leads to an improvement of the cut size. For example, Let  $C_1, C_2$  be the bisection of graph  $G=(V,E)$  --- i.e.,  $C_1 \cup C_2 = V$  and  $C_1 \cap C_2 = \emptyset$ . For each  $v \in N$ , the server placement module 114 defines  $\text{int}(v) = \sum_{(v,u) \in E \& P(v)=P(u)} \omega(v, u)$  and  $\text{ext}(v) = \sum_{(v,u) \in E \& P(v) \neq P(u)} \omega(v, u)$ . The server placement module 114 defines the gain of moving edge  $v$  to a different partition to be  $h(v) = \text{ext}(v) - \text{int}(v)$ . During each iteration, the server placement module 114 finds the best pair of nodes  $v_1 \in C_1$  and  $v_2 \in C_2$  to exchange to maximize the gain. In an embodiment, the server placement module 114 executes this procedure recursively until no further gain can be obtained by changing any pairs of nodes. Note that the target of the algorithm 704 is still a partitioning of  $k$  clusters. Lines 9-10 further partition the two initial two clusters  $C_1$  and  $C_2$ , respectively. However, their targets are reduced to  $k-1$  and  $k-2$  accordingly.

In an embodiment, the server placement module 114 is configured to utilize an algorithm for minimizing both user-to-server latency and a latency of inter-user communications 706, which is an extension of both the algorithm for minimizing user-to-server latency 702 and the algorithm for minimizing a latency of inter-user communications 704. The algorithm 706 is reproduced below:

```

procedure OSN-MULTI( $G=(V,E),k$ )
1:  $P \leftarrow \text{OSN-KMEANS}(G,k)$ 
2: for every pair of clusters  $C_1, C_2 \in P$  with centroids  $c_1, c_2$  do
3:   repeat
4:     for any pair of unmarked ( $v_1 \in C_1; v_2 \in C_2$ ) do
5:       compute  $h(v_1, v_2) = h(v_1) + h(v_2) - 2w(v_1, v_2)$ 
6:     end for
7:     Find  $(v_1, v_2) = \text{argmax } h(v_1, v_2)$  and  $\text{dist}(v_1, c_2) < D_{\max}$  and  $\text{dist}(v_2, c_1) < D_{\max}$ : swap and mark  $v_1, v_2$ 
8:   until  $\forall v_1, v_2, h(v_1, v_2) \leq 0$  ———  $\text{iter} = \text{ITER}_{\max}$  ——— all nodes marked
9: end for

```

In an embodiment, the server placement module 114 initializes the partition using the algorithm for minimizing user-to-server latency 702 and then iteratively switches nodes to reduce the cuts similar to that of the algorithm for minimizing a latency of inter-user communications 704. In an embodiment, to prevent resulting in large user-to-server latency during the swapping process, the server placement module 114 imposes maximum latency constraints between the centroid and the new node if the move is to be performed. While in some embodiments the algorithm 706 may finish with a local optimal solution instead of a global optimal one, given the NP hard property of the problem, the algorithm is much more efficient and scalable than other integer programming based solutions.

Turning back to Figure 1, the server placement module 114 is depicted as configured to utilize the algorithm for minimizing both user-to-server latency and a latency of inter-user communications 706, which, in an embodiment, includes two sub-algorithms, each performing operations similar to algorithms 702 and 704. Accordingly, the server placement module 114 includes two first clusters obtained from performing the first sub-algorithm 130, which depicts two potential clusters (and centroids 134A) generated by the algorithm that attempted to minimize user-to-server latency. Then, the server placement module 114 performs the second sub-algorithm and generates another two clusters as a result 132. As described above, in an embodiment this process entails examining pairs of nodes 135 from different clusters and swapping these nodes if and only if the cut size is reduced and the distance of each swapped node is within a maximum latency constraint distance from the centroid (e.g. 134B) of the new cluster. Accordingly, this second set of clusters 132 is shown with a first swapped node 'A' 131A and a second swapped node 'B' 131B, which reveals that the swap occurred because the cut size was reduced and each node was within the maximum latency constraint distance from the centroid (e.g. 134B). In an embodiment, each centroid (e.g. 134B) of these second set of clusters from the second sub-algorithm 132 represents the recommended geographic server locations 124. In some embodiments, upon completion of the algorithm 706 by the server placement module 114, the presentation module 116 utilizes port(s) 120 to cause the recommended geographic server locations 124 to be presented to a user, but in other embodiments, the presentation module 116 utilizes a display port 122 to display the recommended

geographic server locations 124 to a user on a display device. In some embodiments, the display port 122 is a physical interface of the computing device 106 used to transmit visual information to a user, either through a wired or wireless interface.

In some embodiments of the invention, the computing device 106 includes a joint analysis module 118. As depicted, the joint analysis module 118 interacts with the server placement module 114 and the presentation module 116; however, in other embodiments (not depicted) the joint analysis module 118 is used with other modules and used in a different order than that depicted.

The joint analysis module 118 allows the computing device 106 to utilize information from multiple existing OSNs in additional ways when determining recommended geographic server locations 124. One use enabled by the joint analysis module 118 is depicted in Figure 2 and provides “joint before” analysis of OSN data, and another use enabled by the joint analysis module 118 is depicted in Figure 3 and Figure 4 and provides “joint after” analysis of OSN data.

Figure 2 illustrates a “joint before” approach for determining a plurality of recommended geographic server locations using aggregated user graphs from a plurality of networks according to one embodiment of the invention. In an embodiment, the joint analysis module 118 first combines all the input nodes and edges to create a super graph:

$$\hat{G} = (\hat{V}, \hat{E}) = \cup G_i$$

where node weights:

$$w_{\hat{u}} = (\sum_u w_u) / |\cup u|$$

and edge weights:

$$w_{\hat{u},\hat{v}} = (\sum w_{u,v}) / |\cup (u,v)|$$

In Figure 2, this is depicted as the first, second, and third aggregated user graphs (202, 204, and 206) as being combined into “joint before” super graph 208. In other embodiments, the aggregated user graphs (202, 204, and 206) are not first created; rather, individual nodes and edges from the different OSNs are added one by one to create the “joint before” super graph 208.

Then, the server placement module 114 takes  $\hat{G}$  (the “joint before” super graph 208) as the input for partitioning from the joint analysis module 118.

Figure 3 and Figure 4 illustrate portions of a “joint after” approach for determining a plurality of recommended geographic server locations using centroids from within clusters from different network graphs according to one embodiment of the invention.

5           According to an embodiment of the invention, the “joint after” approach works as such: given the solution sets  $CP = \{ C_1, C_2, \dots, C_p \}$  for  $p$  OSNs, the joint analysis module 118 is configured to rank all centroids  $i$  based on their occurrence  $\tau_i$  in  $CP$ ,  $\tau < p$ . Each time, the joint analysis module 118 selects the centroid with largest  $\tau$ . Together, the joint analysis module 118 removes all nodes covered by that centroid,  
10 and updates the edge weights of the rest of each graph accordingly. Then, the server placement module 114 generates graph partitions again on the updated graph  $G' = G - C_i$  for each OSN, and selects the next centroid based on the new solution sets. Iteratively, the computing device 106 selects  $k$  centroids as the placement for new OSNs.

15           For example, in an embodiment, the server placement module 114 is configured to generate graph partitions for multiple networks 350. Figure 3 depicts this as a first, second, and third network clustered graph (302, 304, and 306). Each of these network clustered graphs includes clusters (e.g. 301) and a centroid (e.g. 311) for each cluster. Next, the server placement module 114 identifies the centroids from each networked  
20 clustered graph as 303, 305, and 307.

The joint analysis module 118 is then tasked with selecting a centroid having the largest occurrence 352. In an embodiment, the joint analysis module 118 consolidates the centroids 308 and then ranks the centroids 310 according to their occurrence. Next, the centroid with the largest occurrence 312 is selected - here, a  
25 centroid in both P1 302 and P2 304 is the same - at (8,3) - and thus is the centroid having the largest occurrence 312. This largest occurrence centroid 312 is identified as the first recommended server location 314. The coordinate or location of the centroid, depending upon the embodiment, may already represent latitude and/or longitude or another geographic indication, or it may need to be transformed from some other  
30 representation into a geographic location.

At this point, if the value of “ $k$ ” was set to ‘1’, the process would end. However, in an embodiment the process will repeat until  $k$  recommended server locations are

calculated. Thus, optionally (assuming the value of “k” is larger than ‘1’), the process proceeds to 454 to remove nodes covered by the selected centroid. At this point, in an embodiment the joint analysis module 118 removes all nodes and edges covered by the selected centroid from each graph 302, 304, and 306 to generate updated network  
5 graphs 402, 404, 406. In an embodiment, the joint analysis module 118 is configured to remove the nodes and edges covered by the selected centroid by removing the cluster (and all nodes within) associated with the centroid from the graphs (here, removing a cluster from each updated network graph 302 and 304). For those graphs not including that centroid (e.g. 306), the cluster located closest to the selected centroid (or the cluster  
10 having a centroid located closest to the selected centroid) is removed. Thus, the process continues with the server placement module 114 utilizing the updated network graphs 402, 404, and 406 to again generate graph partitions for the multiple networks 350. This results in new clusters (e.g., 420) and centroids (e.g. 421) being generated within the updated network graphs 412, 414, and 416. At this point, the process continues to  
15 352 to again select the centroid with the largest occurrence to find another recommended server location. Of course, if the number of recommended server locations does not yet equal the value of k, the process will continue until they are equal.

Figure 8 illustrates a flow 800 for determining a plurality of recommended  
20 geographic server locations by attempting to minimize both user-server latency and a latency of inter-user communications according to one embodiment of the invention. The operations of this and other flow diagrams will be described with reference to the exemplary embodiments of the other diagrams. However, it should be understood that the operations of the flow diagrams can be performed by embodiments of the invention  
25 other than those discussed with reference to these other diagrams, and the embodiments of the invention discussed with reference these other diagrams can perform operations different than those discussed with reference to the flow diagrams.

Moreover, while the flow diagrams in the figures show a particular order of operations performed by certain embodiments of the invention, it should be understood  
30 that such order is exemplary (e.g., alternative embodiments may perform the operations in a different order, combine certain operations, overlap certain operations, etc.).

At 802, the flow 800 includes acquiring geographic information for a set of one or more users of a set of one or more networks. The geographic information for a user of the plurality of users indicates a geographic location of the user. In an embodiment of the invention, each network of the set of networks is an online social network. At 5 804, relationship information is acquired for at least some of the plurality of users. The relationship information indicates those of the plurality of users that are connected on at least one network of the set of networks.

The geographic information and the relationship information are transformed into a plurality of graphs 806. Each graph of the plurality of graphs represents one 10 network of the plurality of networks. Each of the graphs includes a plurality of nodes representing those users of the plurality of users that belong to the one network, and a plurality of edges connecting the plurality of nodes according to the relationship information. Each of the plurality of edges includes an edge weight.

Next, a first plurality of clusters is generated by performing a first clustering 15 algorithm on the graph 808. Each cluster of the first plurality of clusters includes a centroid and a set of one or more nodes of the plurality of nodes. Each node of the set of nodes is included in only one of the first plurality of clusters.

A second plurality of clusters is generated by performing a second clustering 20 algorithm 810. The second clustering algorithm includes iteratively examining pairs of clusters of the first plurality of clusters. The second clustering algorithm also includes, for each examined pair of clusters, repeatedly swapping pairs of nodes between the pair of clusters when a swap of a pair of nodes satisfies two conditions. The first condition is that the swap would reduce a cut weight of the graph, wherein the cut weight is a sum of edge weights of edges that connect nodes in different clusters. The second 25 condition is that the swap would locate each node of the pair of nodes, when swapped to the other cluster of the examined pair of clusters, within a defined maximum distance from the centroid of the other cluster.

Finally, at 812, information describing geographic locations of centroids of the 30 second plurality of clusters is caused to be presented to a user as the plurality of recommended geographic server locations.

Figure 9 illustrates a flow 900 to determine a plurality of recommended geographic server locations by attempting to minimize both user-server latency and a latency of inter-user communications using a joint analysis approach based upon characteristics of a plurality of networks according to one embodiment of the invention.

5 At 902, the flow 900 includes acquiring geographic information for a plurality of users of a plurality of networks. The geographic information for a user of the plurality of users indicates a geographic location of the user. In an embodiment of the invention, the plurality of networks is a plurality of online social networks. At 904, relationship information is acquired for at least some of the plurality of users. The  
10 relationship information indicates those of the plurality of users that are connected on at least one of the plurality of networks.

The geographic information and the relationship information are transformed into a plurality of graphs 906. Each graph of the plurality of graphs represents one network of the plurality of networks. Each of the graphs includes a plurality of nodes  
15 representing those users of the plurality of users that belong to the one network, and a plurality of edges connecting the plurality of nodes according to the relationship information. Each of the plurality of edges includes an edge weight.

At 908, the flow 900 includes generating, for each graph of the plurality of graphs, a plurality of clusters for that graph by performing a clustering algorithm. Each  
20 cluster of the plurality of clusters includes a centroid. Finally, at 910 a first recommended geographic server location is identified. First, a set of centroids is ranked according to the frequency of occurrence of each centroid in the set of centroids. This set of centroids includes all of the centroids of the plurality of clusters of each of the plurality of graphs. Next, a centroid of the ranked set of centroids that has the highest  
25 occurrence is identified as representing the first recommended geographic server location.

#### Alternative Embodiments

While embodiments of the invention have been described in relation to OSNs, the techniques disclosed herein can easily be extended by one of skill in the art to be  
30 useful in a wide variety of further applications.

For example, in the near future there will likely be billions of new devices connected to the Internet. Going widely beyond traditional computing devices such as the computers and mobile phones of today, future devices may include everyday objects such as desks, chairs, glasses, bikes, microwaves, refrigerators, shopping carts, cars, and traffic lights. While many of these devices are considered geographically stationary (as they typically are used in one location or a few geographically similar locations), many others are mobile, much like human users in OSNs. Such future devices will likely have embedded communication and data processing capabilities, and with the help of software and services, these devices will conceptually be able to communicate with each other as human friends do on OSNs today. Thus, when designing new services or networks that support such a large magnitude of networked devices, communications must be optimized and thus the techniques described herein will be applicable.

Similarly, consider the field of messaging-oriented middleware (MoM) or messaging services, such as the existing Enterprise Service Bus systems (ESB). In general, these messaging services have a large set of endpoints (“publishers” and “subscribers”) that communicate via a conceptually centralized server. The server dispatches messages according to the matching between subscribers’ interests and published contents. In actual implementations, however, the system should also be optimized for scalability reasons by using a set of distributed servers.

In all of the above cases, just like the OSNs discussed herein, these networks also must solve the server placement problem by considering messaging latencies, communication patterns (or social strengths) of nodes, and costs of data centers; accordingly, the systems, methods, and apparatuses disclosed are equally applicable and beneficial in these environments.

While the invention has been described in terms of several embodiments, those skilled in the art will recognize that the invention is not limited to the embodiments described, can be practiced with modification and alteration within the spirit and scope of the appended claims. The description is thus to be regarded as illustrative instead of limiting.

CLAIMS

What is claimed is:

1. A method in a computing device to determine a plurality of recommended geographic server locations 124 by attempting to minimize both user-server  
5 latency and a latency of inter-user communications, the method comprising:  
acquiring geographic information for a plurality of users 102A-102N of a set of  
one or more networks 103A-103N, wherein the geographic information  
for each of the plurality of users 102A-102N indicates a geographic  
location of that user 102A;  
10 acquiring relationship information 104 for at least some of the plurality of users  
102A-102N, wherein the relationship information indicates those of the  
plurality of users 102A-102N that are connected on a network 103A of  
the set of networks 103A-103N;  
transforming the geographic information and the relationship information into a  
15 graph 128 including a plurality of nodes 136 representing the plurality of  
users 102A-102N and a plurality of edges 138 connecting the plurality  
of nodes 136 according to the relationship information, wherein each of  
the plurality of edges 138 includes an edge weight 137;  
generating a first plurality of clusters 130 by performing a first clustering  
20 algorithm on the graph 128, wherein each cluster of the first plurality of  
clusters 130 includes a centroid 134A and a set of one or more nodes of  
the plurality of nodes 136, wherein each of the set of nodes is included  
in only one of the first plurality of clusters 130;  
generating a second plurality of clusters 132 by performing a second clustering  
25 algorithm comprising,  
iteratively examining pairs of clusters of the first plurality of clusters  
130, and  
for each examined pair of clusters, repeatedly swapping pairs of nodes  
135 between the pair of clusters when a swap of a pair of nodes  
30 135 will:

- reduce a total cut weight of the graph, wherein the total cut weight is a sum of edge weights 137 of edges that connect nodes 136 in different clusters, and
- locate each node of the pair of nodes 135, when swapped to the other cluster of the examined pair of clusters, within a defined maximum distance from the centroid of the other cluster to thereby bound user-to-server latency; and
- causing information describing geographic locations of centroids 134B of the second plurality of clusters 132 to be presented to a user as the plurality of recommended geographic server locations 124.
2. The method of claim 1, wherein each of the edge weights 137 indicates an estimated network latency time between those users of the plurality of users 102A-102N represented by the nodes 136 at each end of the edge 138.
  3. The method of claim 2, wherein the estimated network latency time is calculated by the server end station based upon Vincenty's formula.
  4. The method of claim 1, wherein:  
the first clustering algorithm is based on either a k-means clustering algorithm or a k-medoids clustering algorithm; and  
the first clustering algorithm uses known geographic locations of servers of at least one of the set of networks 103A-103N as initial centroid candidates.
  5. The method of claim 1, wherein:  
the count of the plurality of nodes is smaller than the count of the plurality of users 102A-102N; and  
at least one of the plurality of nodes represents a group of two or more of the plurality of users 102A-102N.
  6. The method of claim 5, wherein the group of users includes those of the plurality of users 102A-102N having the geographic location that is within a defined range of latitude and longitude values.

7. The method of claim 1, wherein:  
a first plurality of the plurality of users 102A-102N are users of a first network 103A of the set of networks 103A-103N; and  
a second plurality of the plurality of users 102A-102N are users of a second  
5 network 103B of the set of networks 103A-103N.
8. The method of claim 7, wherein the first network 103A and the second network 103B are online social networks.
9. The method of claim 1, wherein said transforming of the geographic information and the relationship information into the graph 126 comprises:  
10 transforming the geographic location of each user 102A of the plurality of users 102A-102N into a latitude-longitude coordinate;  
repeatedly aggregating those users of the plurality of users 102A-102N having latitude-longitude coordinates located within a defined distance of each other into one node 139 to thereby create a plurality of nodes 136,  
15 wherein a weight 141 of each node of the plurality of nodes 136 indicates how many of the plurality of users 102A-102N are represented by that node;  
connecting those of the plurality of nodes 136 using a plurality of edges 138 based upon whether any users aggregated by those nodes are connected  
20 as indicated by the relationship information; and  
assigning a weight 137 to each edge of the plurality of edges 138 indicating the number of connections between those users of the plurality of users 102A-102N represented by the nodes of the plurality of nodes 136 at each end of that edge.
- 25 10. A computer implemented method to determine a plurality of recommended geographic server locations 124 by attempting to minimize both user-server latency and a latency of inter-user communications using a joint analysis approach based upon characteristics of a plurality of networks 103A-103N, the method comprising:

- acquiring geographic information for a plurality of users 102A-102N of the plurality of networks 103A-103N, wherein the geographic information for a user 102A of the plurality of users 102A-102N indicates a geographic location of the user 102A;
- 5 acquiring relationship information for at least some of the plurality of users 102A-102N, wherein the relationship information indicates those of the plurality of users 102A-102N that are connected on at least one of the plurality of networks 103A-103N;
- transforming the geographic information and the relationship information into a plurality of graphs (302, 304, 306), wherein each graph of the plurality of graphs (302, 304, 306) represents one network of the plurality of networks 103A-103N and includes,
- 10 a plurality of nodes 136 representing those users of the plurality of users 102A-102N that belong to the one network, and
- 15 a plurality of edges 138 connecting the plurality of nodes 136 according to the relationship information, wherein each of the plurality of edges includes an edge weight 137;
- generating, for each graph of the plurality of graphs (302, 304, 306), a plurality of clusters 301 for that graph by performing a clustering algorithm,
- 20 wherein each of the plurality of clusters includes a centroid 311; and
- identifying a first recommended geographic server location 314 by:
- ranking 310 a set of centroids 308 according to the frequency of occurrence of each centroid in the set of centroids 308, wherein the set of centroids includes all of the centroids of the plurality of
- 25 clusters 301 of each of the plurality of graphs (302, 304, 306), and
- identifying a centroid 312 of the ranked set of centroids 310 having the highest occurrence as representing the first recommended geographic server location 314.
- 30 11. The method of claim 10, further comprising:
- identifying a second recommended geographic server location by:

- removing, from each of the plurality of graphs (302, 304, 306), all nodes from a cluster in that graph having a centroid located closest to the identified centroid 312 to create a modified plurality of graphs (402, 404, 406),
- 5 for each of the modified plurality of graphs (402, 404, 406), generating a second plurality of clusters 420 by again performing the clustering algorithm, wherein each of the second plurality of clusters 420 includes a centroid 421,
- 10 ranking 310 a second set of centroids 308 according to the frequency of occurrence of each centroid in the second set of centroids 308, wherein the second set of centroids 308 includes all of the centroids of the second plurality of clusters 420 of each of the modified plurality of graphs (402, 404, 406), and
- 15 identifying a centroid 312 of the ranked second set of centroids 310 having the highest occurrence as the second recommended geographic server location 314.
12. The method of claim 10, wherein each of the edge weights 137 indicates an estimated network latency time between those users of the plurality of users 102A-102N represented by the nodes 136 at each end of the edge 138.
- 20 13. The method of claim 10, wherein the clustering algorithm is based upon a k-means clustering algorithm or a k-medoids clustering algorithm.
14. The method of claim 10, where the plurality of networks 103A-103N are online social networks.
- 25 15. A server end station to determine a plurality of recommended geographic server locations 124 by attempting to minimize both user-server latency and a latency of inter-user communications, the server end station comprising:  
an information acquisition module 108 configured to,  
acquire geographic information for a plurality of users 102A-102N of a set of one or more networks 103A-103N, wherein the geographic

information for each user 102A of the plurality of users 102A-102N indicates a geographic location of that user 102A, and acquire relationship information for at least some of the plurality of users 102A-102N, wherein the relationship information indicates those of the plurality of users 102A-102N that are connected on at least one network of the set of networks 103A-103N;

5 a transformation module 112 configured to transform the geographic information and the relationship information into a graph 128 including a plurality of nodes 136 representing the plurality of users 102A-102N and a plurality of edges 138 connecting the plurality of nodes 136 according to the relationship information, wherein each of the plurality of edges 138 includes an edge weight 137;

10 a server placement module 114 configured to,

generate a first plurality of clusters 130 by performing a first clustering algorithm on the graph 128, wherein each cluster of the first plurality of clusters 130 includes a centroid 134A and a set of one or more nodes of the plurality of nodes 136, wherein each node of the set of nodes 136 is included in only one cluster of the first plurality of clusters 130, and

15 generate a second plurality of clusters 132 by performing a second clustering algorithm comprising,

iteratively examining pairs of clusters of the first plurality of clusters 130, and

for each examined pair of clusters, repeatedly swapping pairs of nodes 131A-131B between the pair of clusters when a swap of a pair of nodes 135 will:

20 reduce a total cut weight of the graph 128, wherein the total cut weight is a sum of edge weights 137 of edges 138 that connect nodes 136 in different clusters, and

25 locate each node of the pair of nodes 135, when swapped to the other cluster of the examined pair of

30

- clusters, within a defined maximum distance from the centroid 134B of the other cluster to thereby bound user-server latency; and
- 5 a presentation module 116 configured to cause information describing geographic locations of centroids of the second plurality of clusters to be presented to a user as the plurality of recommended geographic server locations 124.
16. The server end station of claim 15, wherein the transformation module 112 is configured to compute the edge weights 137 by estimating a network latency
- 10 time between those users of the plurality of users 102A-102N represented by the nodes 136 at each end of the edge 138.
17. The server end station of claim 15, wherein:
- the first clustering algorithm is based on either a k-means clustering algorithm or a k-medoids clustering algorithm; and
- 15 the first clustering algorithm uses known geographic locations of servers of at least one network of the set of networks 103A-103N as initial centroid candidates.
18. The server end station of claim 15, wherein the information acquisition module 108 is configured to acquire said geographic information and said relationship
- 20 information using a set of one or more network interface ports 120.
19. The server end station of claim 15, wherein:
- a first plurality of the plurality of users 102A-102N are users of a first network 103A of the set of networks 103A-103N; and
- a second plurality of the plurality of users 102A-102N are users of a second
- 25 network 103N of the set of networks 103A-103N.
20. The server end station of claim 19, wherein the first network 103A and the second network 103N are online social networks.

Fig. 1

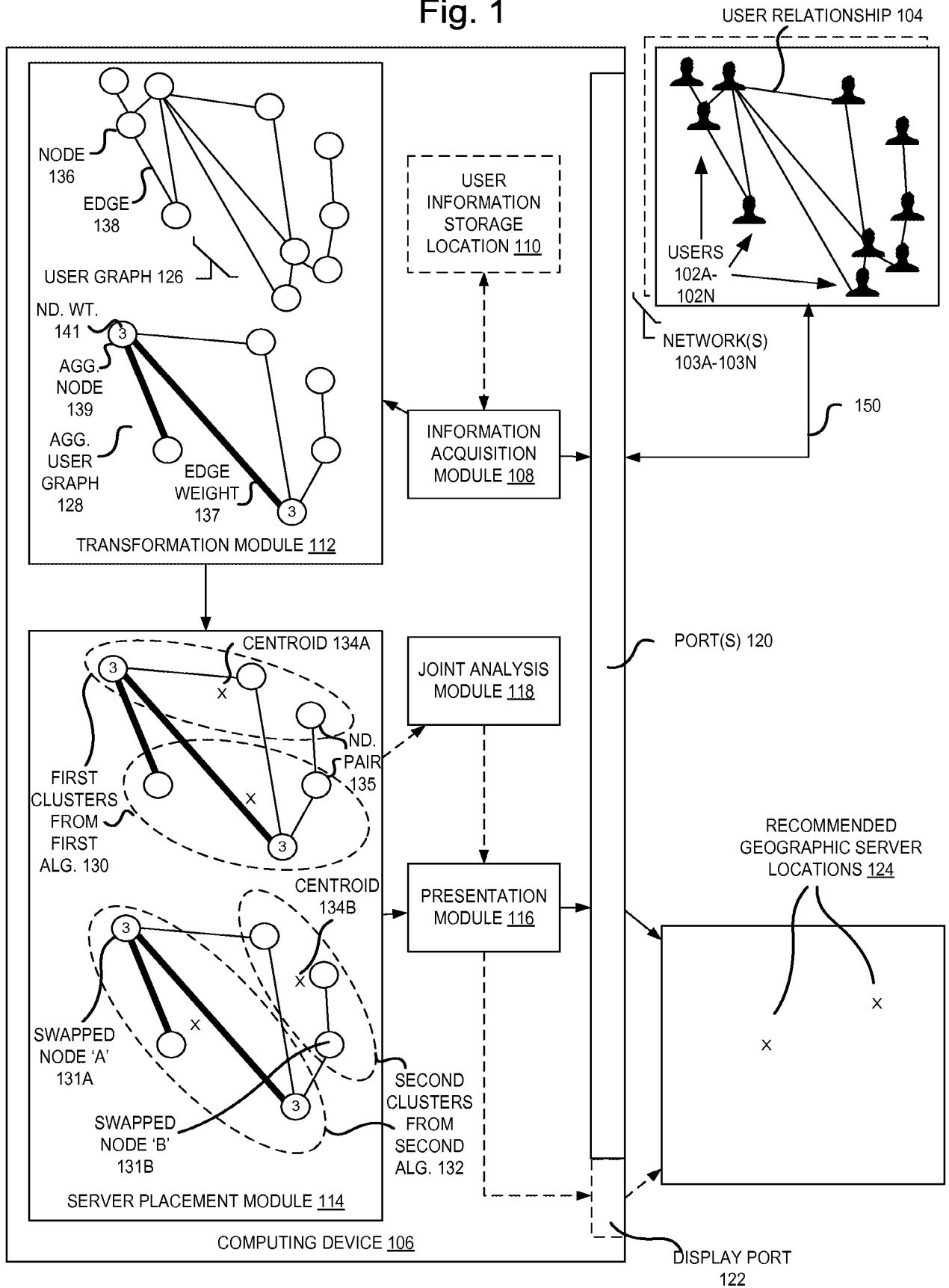


Fig. 2

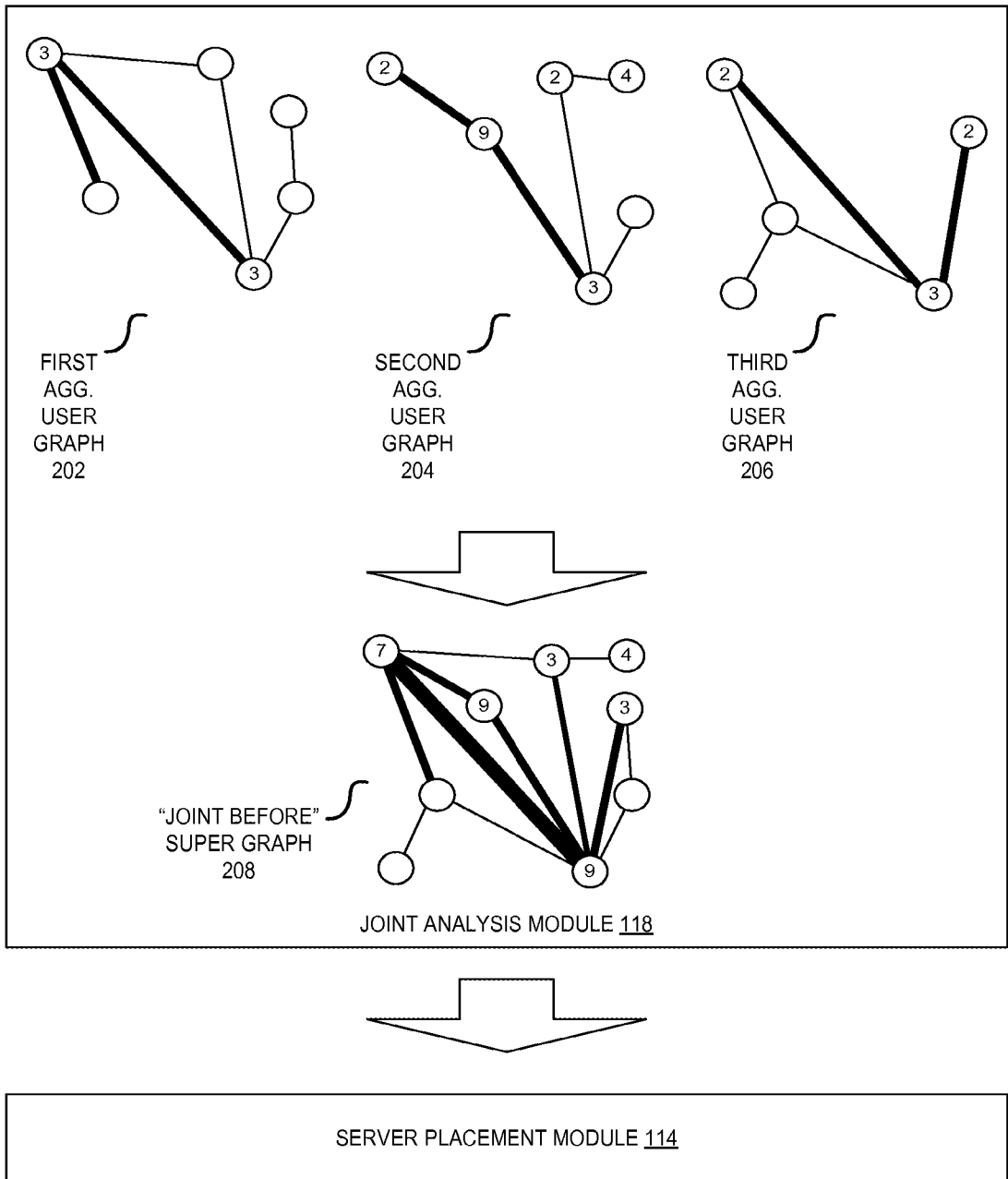
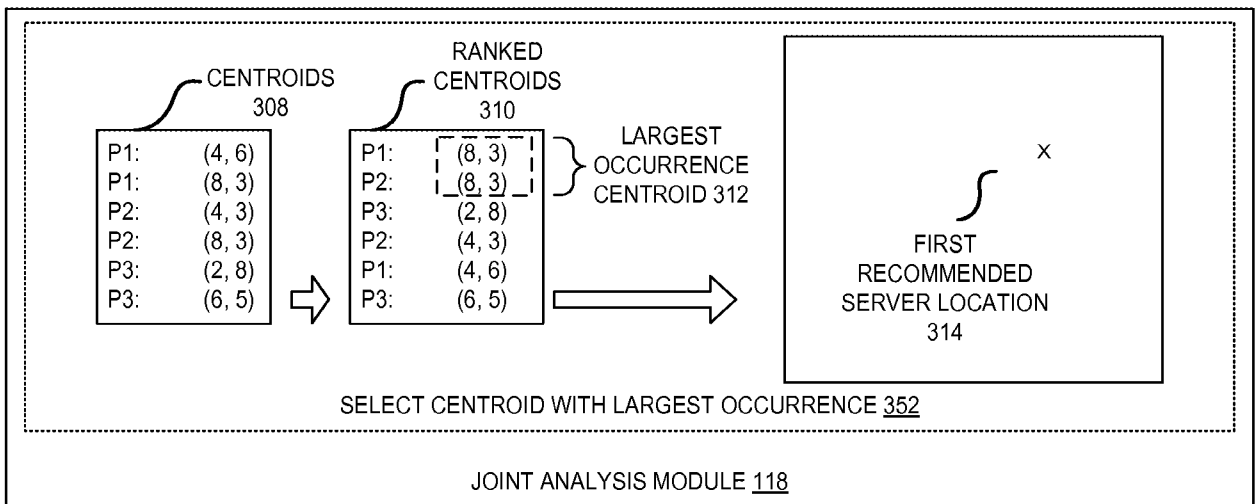
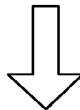
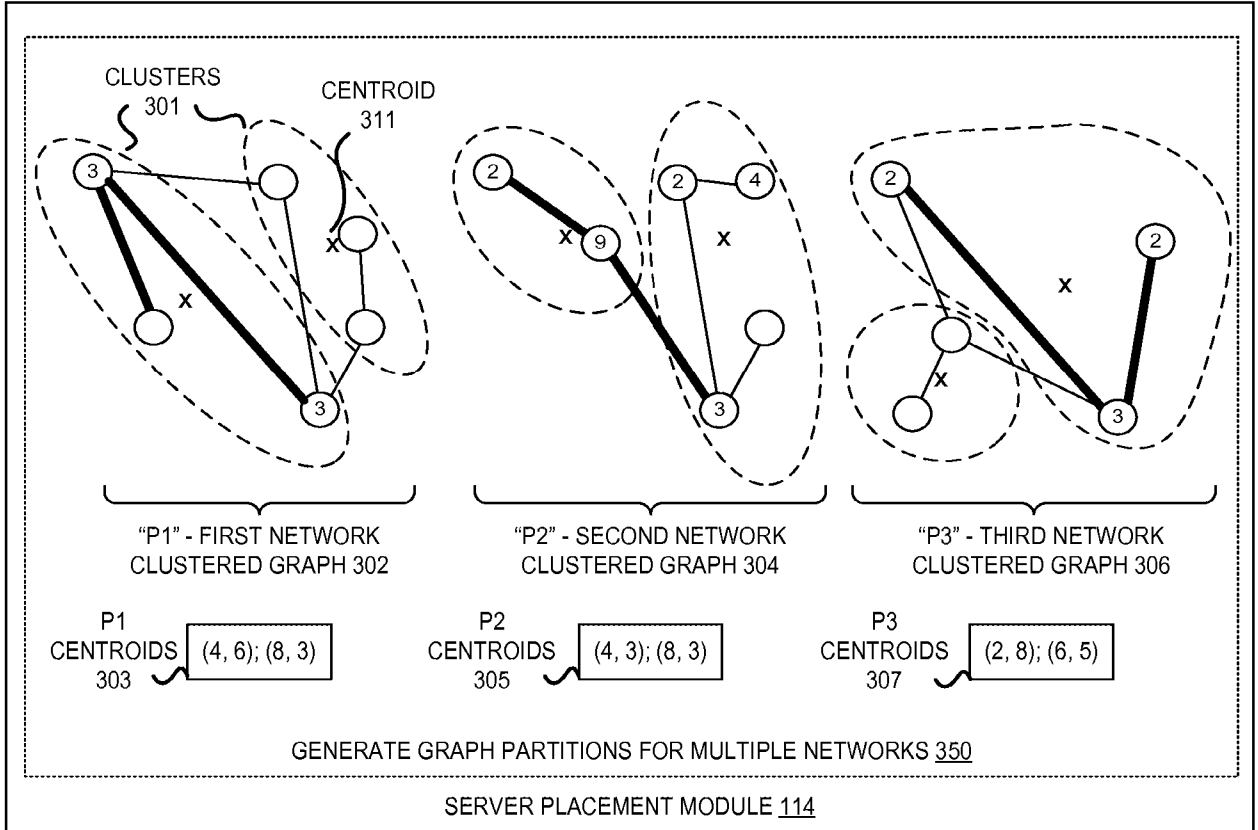


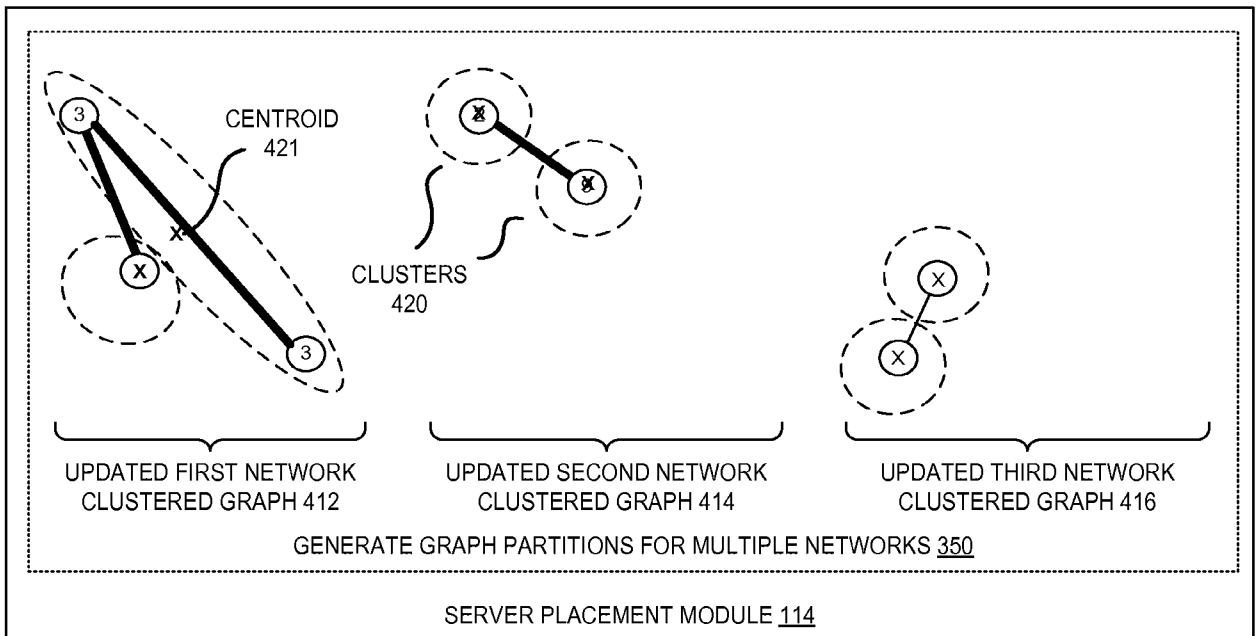
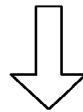
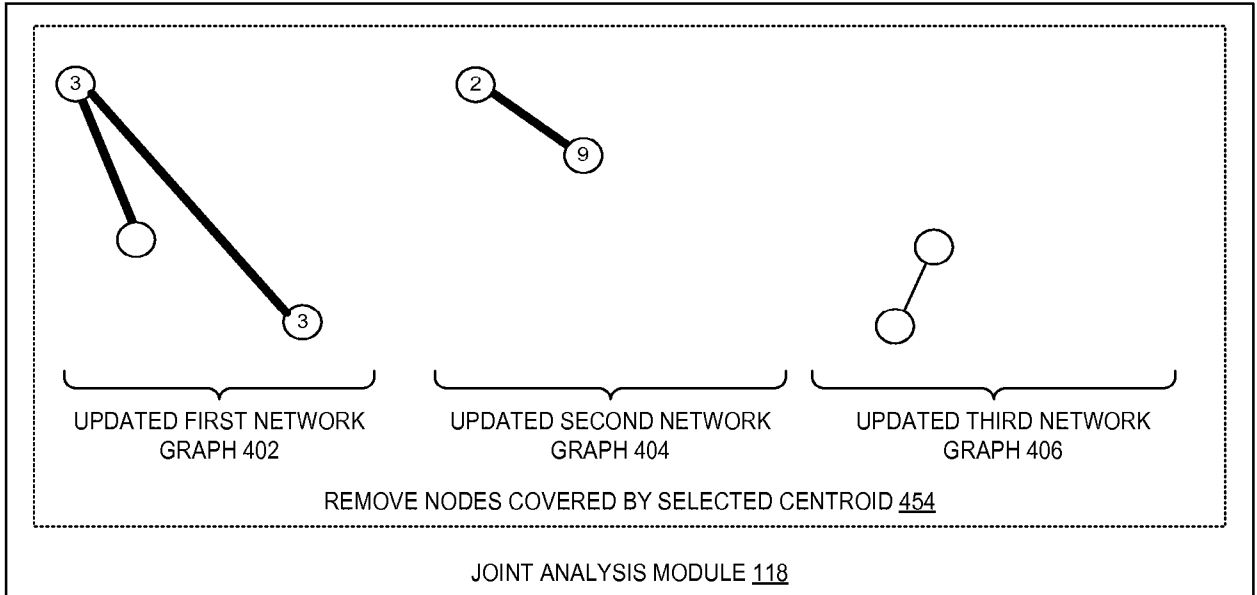
Fig. 3



TO 454 - "REMOVE NODES COVERED BY SELECTED CENTROID"

Fig. 4

FROM 352 – “SELECT CENTROID WITH LARGEST OCCURRENCE”



TO 352 – “SELECT CENTROID WITH LARGEST OCCURRENCE”

Fig. 5

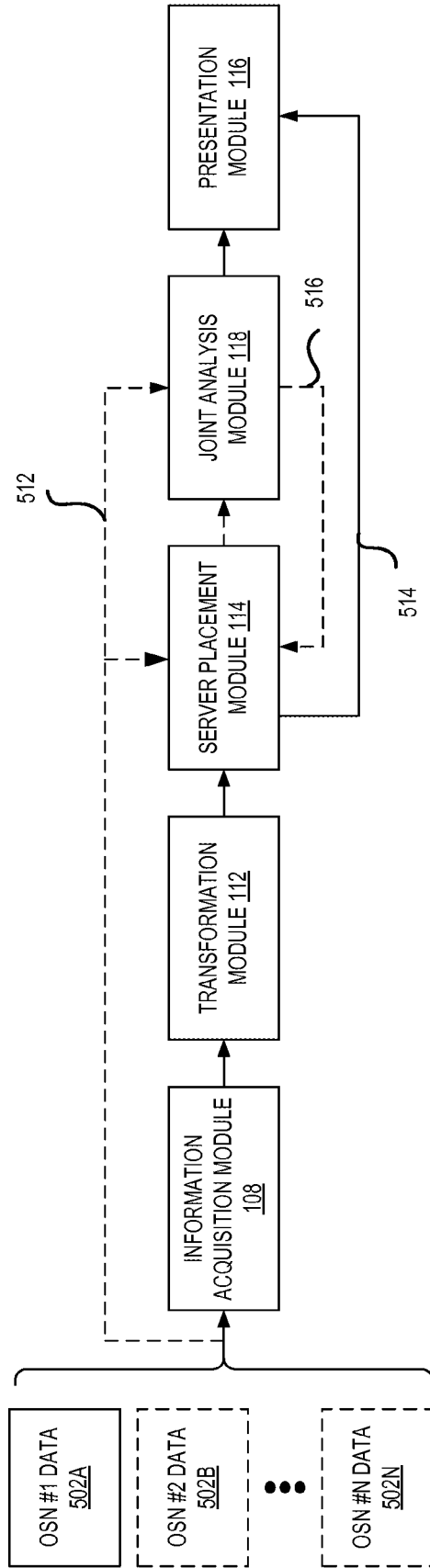


Fig. 6

INTRA-CLUSTER  
LATENCY COST  
FUNCTION  
602

$$f = \sum_{C \in P} \sum_{u \in C} dist(u, centroid(C))$$

INTER-CLUSTER  
TRAFFIC COST  
FUNCTION  
604

$$g = \frac{1}{|W|} \sum_{u \in V} \frac{1}{h} \sum_{j=1}^h \delta(u, ne^{i_{o_j}})$$

Fig. 7

702

**Algorithm 1 K-means based Server Placement Algorithm**

procedure *OSN-KMEANS*( $G=(V,E),k$ )

- 1: Initialize cluster centroid  $C = c_1, c_2, \dots, c_k$
- 2: Assign points to closest centroid  $c_i$  according to  $P_i := \text{argmin} \|x_j - c_i\|^2$
- 3: Find new centroid  $C' = c'_1, \dots, c'_k$  to find  $c'_i = \frac{1}{|P_i|} \sum_{x_j \in P_i} x_j$
- 4: IF  $C' \neq C$
- 5:  $C = C'$ , GOTO 2
- 6: Compute the client to server latency between each point and its assigned centroid
- 7: Compute the friendship links across servers

704

**Algorithm 2 Mincut-based Server Placement Algorithm**

procedure *OSN-MINCUT*( $G=(V,E),k$ )

- 1: Initialize bisection partition  $\mathcal{P} = \{C_1, C_2\}$
- 2: repeat
- 3: for any pair of unmarked  $(v_1 \in C_1, v_2 \in C_2)$  do
- 4: compute  $h(v_1, v_2) = h(v_1) + h(v_2) - 2w(v_1, v_2)$
- 5: end for
- 6: Find  $(v_1, v_2) = \text{argmax} h(v_1, v_2)$ : swap and mark  $v_1, v_2$
- 7: until  $\forall v_1, v_2, h(v_1, v_2) < 0$  ———  $iter = ITER_{max}$  ——— all nodes marked
- 8: if  $k > 2$  then
- 9: *OSN-MINCUT*( $G_1=(C_1, E_1), k-1$ )
- 10: *OSN-MINCUT*( $G_2=(C_2, E_2), k-2$ )
- 11: end if

706

**Algorithm 3 Multi-objective Server Placement Algorithm**

procedure *OSN-MULTI*( $G=(V,E),k$ )

- 1:  $\mathcal{P} \leftarrow \text{OSN-KMEANS}(G, k)$
- 2: for every pair of clusters  $C_1, C_2 \in \mathcal{P}$  with centroids  $c_1, c_2$  do
- 3: repeat
- 4: for any pair of unmarked  $(v_1 \in C_1, v_2 \in C_2)$  do
- 5: compute  $h(v_1, v_2) = h(v_1) + h(v_2) - 2w(v_1, v_2)$
- 6: end for
- 7: Find  $(v_1, v_2) = \text{argmax} h(v_1, v_2)$  and  $\text{dist}(v_1, c_2) < D_{max}$  and  $\text{dist}(v_2, c_1) < D_{max}$ : swap and mark  $v_1, v_2$
- 8: until  $\forall v_1, v_2, h(v_1, v_2) \leq 0$  ———  $iter = ITER_{max}$  ——— all nodes marked
- 9: end for

Fig. 8

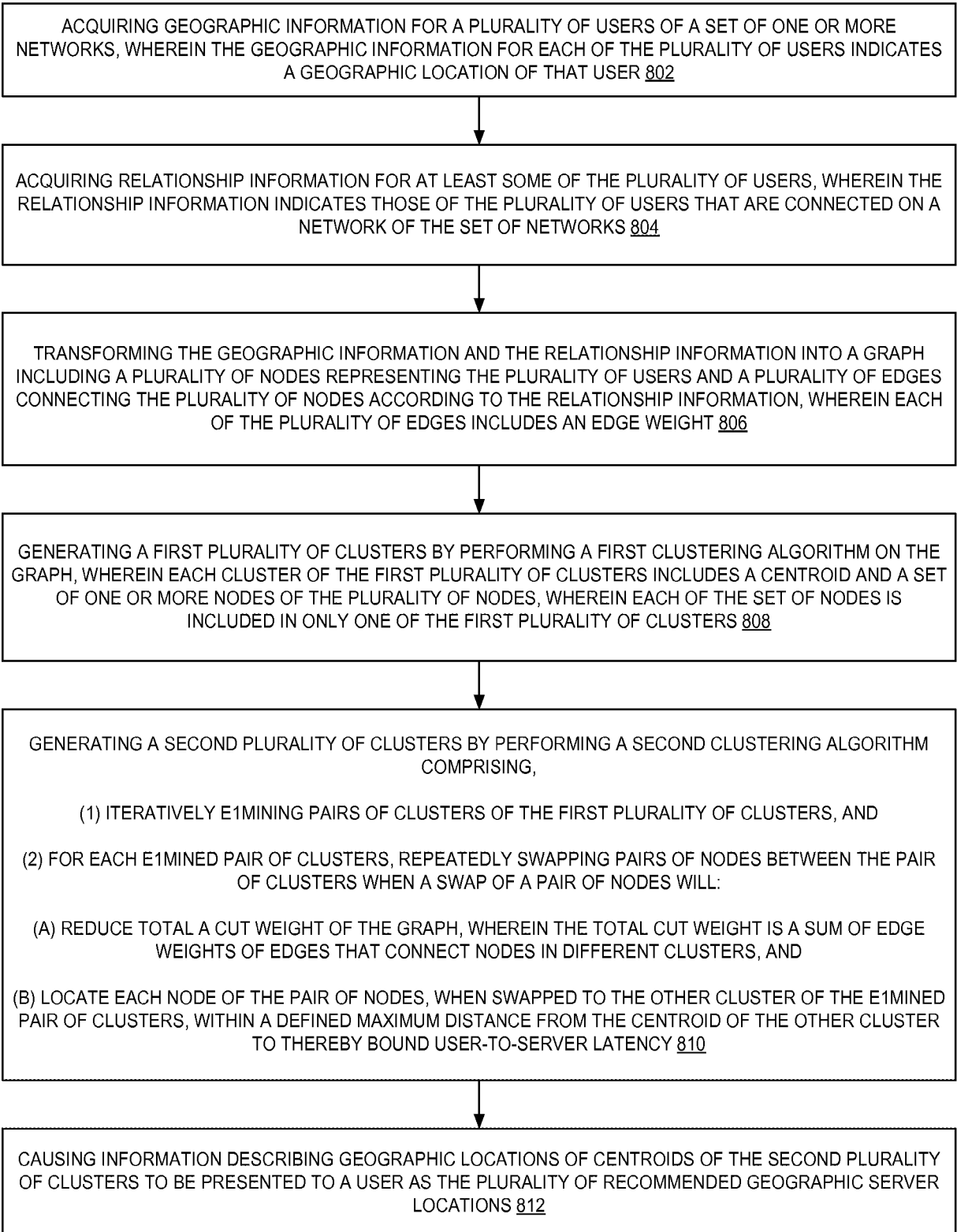
800  
↓

Fig. 9

900

