Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2919664 C 2017/05/16

(11)(21) **2 919 664**

(12) **BREVET CANADIEN**
CANADIAN PATENT

(13) **C**

(86) **Date de dépôt PCT/PCT Filing Date:** 2014/07/28
(87) **Date publication PCT/PCT Publication Date:** 2015/02/05
(45) **Date de délivrance/Issue Date:** 2017/05/16
(85) **Entrée phase nationale/National Entry:** 2016/01/27
(86) **N° demande PCT/PCT Application No.:** US 2014/048401
(87) **N° publication PCT/PCT Publication No.:** 2015/017320
(30) **Priorité/Priority:** 2013/07/30 (US13/954,719)

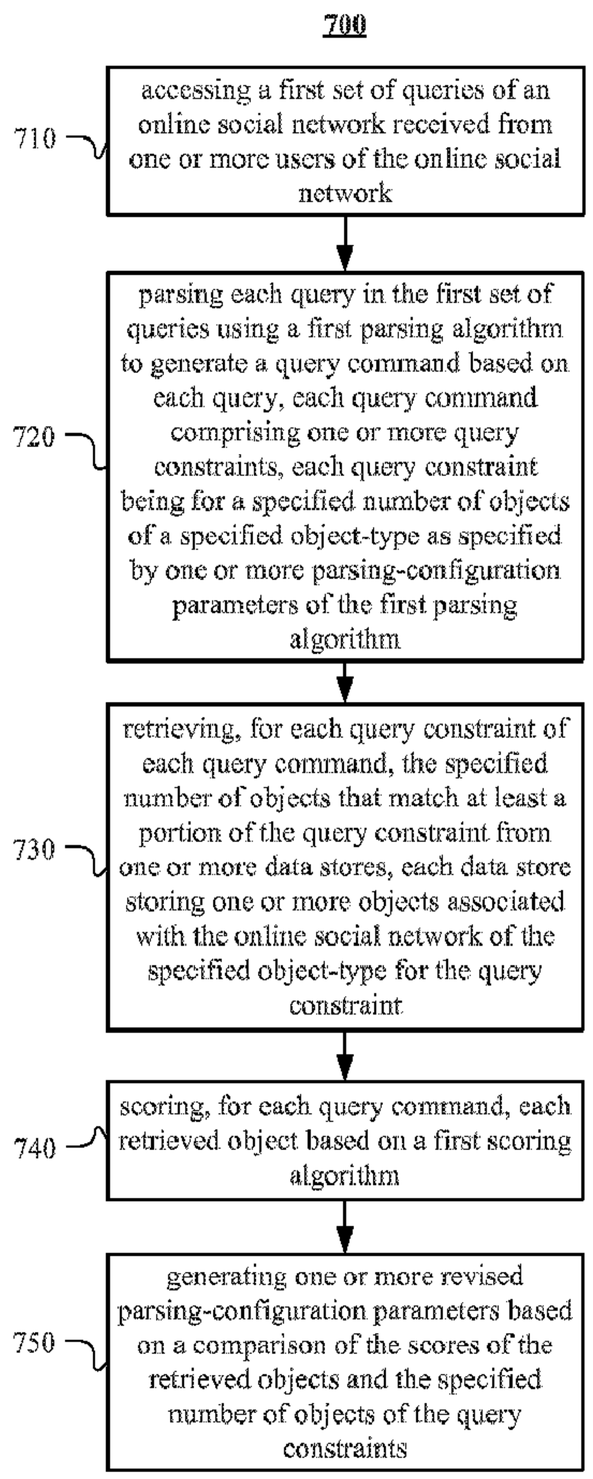
(51) **Cl.Int./Int.Cl.** ***G06F 17/30*** (2006.01),
G06Q 50/30 (2012.01)

(72) **Inventeurs/Inventors:**
RAJAT, RAINA, US;
STEWART, ALLAN, US;
MASHIACH, LI-TAL, US

(73) **Propriétaire/Owner:**
FACEBOOK, INC., US

(74) **Agent:** STIKEMAN ELLIOTT S.E.N.C.R.L.,SRL/LLP

(54) **Titre : REECRITURE D'INTERROGATIONS DE RECHERCHE SUR DES RESEAUX SOCIAUX EN LIGNE**
(54) **Title: REWRITING SEARCH QUERIES ON ONLINE SOCIAL NETWORKS**



(57) **Abrégé/Abstract:**
In one embodiment, a method includes accessing a set of queries of an online social network received from one or more users of the online social network, parsing each query using a parsing algorithm to generate a query command based on each query, each

CA 2919664 C 2017/05/16

(11)(21) **2 919 664**

(13) **C**

(57) **Abrégé(suite)/Abstract(continued):**
query command comprising one or more query constraints for a specified number of objects of a specified object-type as specified by parsing-configuration parameters of the parsing algorithm, retrieving the specified number of objects that match at least a portion of the query constraint from one or more data stores associated with the online social network, scoring each retrieved object based on a scoring algorithm, and generating one or more revised parsing-configuration parameters based on a comparison of the scores of the retrieved objects and the specified number of objects of the query constraints.

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
5 February 2015 (05.02.2015)

WIPO | PCT

(10) International Publication Number
WO 2015/017320 A1

(51) **International Patent Classification**:
G06F 17/30 (2006.01) ***G06Q 50/30*** (2012.01)

(21) **International Application Number**:
PCT/US2014/048401

(22) **International Filing Date**:
28 July 2014 (28.07.2014)

(25) **Filing Language**: English

(26) **Publication Language**: English

(30) **Priority Data**:
13/954,719 30 July 2013 (30.07.2013) US

(71) **Applicant**: **FACEBOOK, INC.** [US/US]; 1601 Willow Road, Menlo Park, CA 94025 (US).

(72) **Inventors**: **RAJAT, Raina**; 1601 Willow Road, Menlo Park, CA 94025 (US). **STEWART, Allan**; 1601 Willow Road, Menlo Park, CA 94025 (US). **MASHIACH, Li-Tal**; 1601 Willow Road, Menlo Park, CA 94025 (US).

(74) **Agent**: **AVERY, Matthew**; Baker Botts L.L.P., 2001 Ross Avenue, Suite 600, Dallas, TX 75201 (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,

[Continued on next page]

(54) **Title**: REWRITING SEARCH QUERIES ON ONLINE SOCIAL NETWORKS

(57) **Abstract**: In one embodiment, a method includes accessing a set of queries of an online social network received from one or more users of the online social network, parsing each query using a parsing algorithm to generate a query command based on each query, each query command comprising one or more query constraints for a specified number of objects of a specified object-type as specified by parsing-configuration parameters of the parsing algorithm, retrieving the specified number of objects that match at least a portion of the query constraint from one or more data stores associated with the online social network, scoring each retrieved object based on a scoring algorithm, and generating one or more revised parsing-configuration parameters based on a comparison of the scores of the retrieved objects and the specified number of objects of the query constraints.

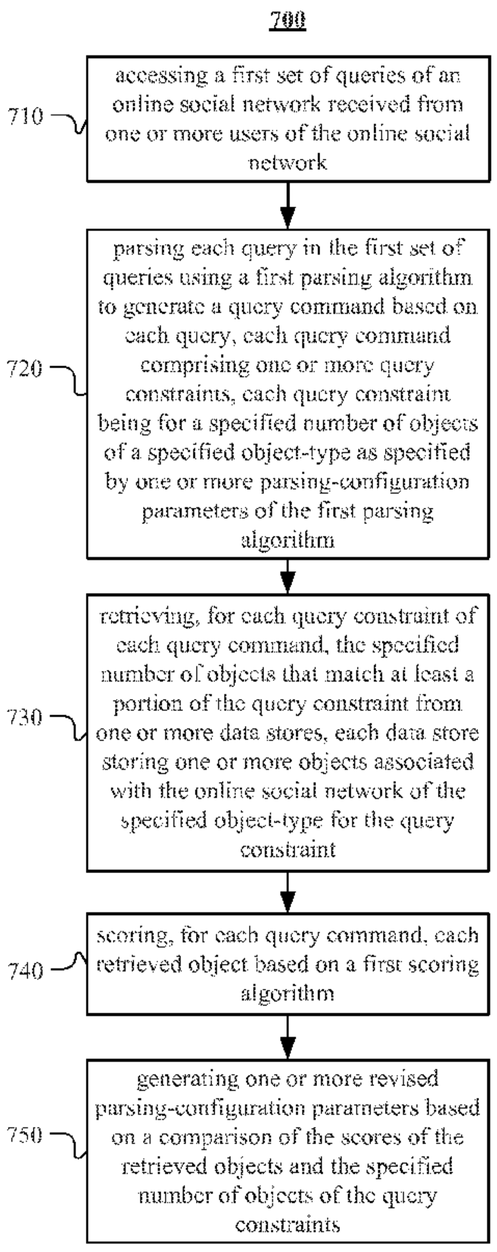


FIG. 7

WO 2015/017320 A1

WO 2015/017320 A1

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 applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

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

Rewriting Search Queries on Online Social Networks

TECHNICAL FIELD

[1] This disclosure generally relates to social graphs and performing searches for objects within a social-networking environment.

BACKGROUND

[2] A social-networking system, which may include a social-networking website, may enable its users (such as persons or organizations) to interact with it and with each other through it. The social-networking system may, with input from a user, create and store in the social-networking system a user profile associated with the user. The user profile may include demographic information, communication-channel information, and information on personal interests of the user. The social-networking system may also, with input from a user, create and store a record of relationships of the user with other users of the social-networking system, as well as provide services (e.g. wall posts, photo-sharing, event organization, messaging, games, or advertisements) to facilitate social interaction between or among users.

[3] The social-networking system may send over one or more networks content or messages related to its services to a mobile or other computing device of a user. A user may also install software applications on a mobile or other computing device of the user for accessing a user profile of the user and other data within the social-networking system. The social-networking system may generate a personalized set of content objects to display to a user, such as a newsfeed of aggregated stories of other users connected to the user.

[4] Social-graph analysis views social relationships in terms of network theory consisting of nodes and edges. Nodes represent the individual actors within the networks, and edges represent the relationships between the actors. The resulting graph-based structures are often very complex. There can be many types of nodes and many types of edges for connecting nodes. In its simplest form, a social graph is a map of all of the relevant edges between all the nodes being studied.

SUMMARY OF PARTICULAR EMBODIMENTS

[5] In particular embodiments, a user of a social-networking system may search for objects associated with the system using a search queries.

[6] In particular embodiments, the social-networking system may improve the processing of search queries by improving the static scores/rankings of objects stored in data stores. When the social-networking system retrieves objects from a data store in response to a query, the objects may be retrieved based on a pre-determined static-score or static-rank associated with the object (which may be based, for example, on how the objects are indexed). The objects retrieved from all data stores may then be aggregated and scored (based on a variety of factors, such as, for example, relevance to the query, social-graph affinity, user history, etc.) by the social-networking system, and these final-scores or final-ranks may then be used to determine which objects are generated as search results that are displayed to the querying user. However, this process may be inefficient if the social-networking system has to retrieve an excess of objects from the data stores in order to generate a sufficient number of search results. This process could be improved if the static-ranks of objects more closely match the final-ranks determined by the social-networking system when generating search results for a user. This may allow the social-networking system to reduce the number of matching objects that need to be retrieved in order to generate a sufficient number of search results in response to a query. In order to improve the static-scores of objects indexed in one or more data stores, the social-networking system may compared the static-scores of objects retrieved from a data store with the final-scores calculated by the social-networking system in order to generate search results for a user, and revise or adjust the static-scores (or the scoring algorithm used to calculate the static-scores) of the indexed objects so they more closely match the final-scores. For example, the social-networking system may access a set of archived search queries and optimize the static-scores of objects retrieved by these queries. This may be done for a variety of queries or query-types, so that the static-scores are optimized to match the final-scores as closely as possible for a variety of queries.

[7] In particular embodiments, the social-networking system may improve the processing of search queries by improving how query commands are generated. When a query is

parsed to generate a query command, the query command may specify a particular number of objects to retrieve of one or more object-types. The number of objects to retrieve of each object-type may be specified by parsing-configuration parameters of the parsing algorithm used to generate the query commands. The retrieved objects may then be scored/ranked and the top-N scoring objects may be sent to the querying user. However, this process may be inefficient if the social-networking system has to retrieve an excess of one or more object-types from particular data stores in order to retrieve the top-N scoring objects, particularly with respect to inefficient use of processing power. This process could be improved if the number of objects retrieved from each data store could be reduced while still retrieving some or all of the objects with the best final-scores, allowing the quality of the generated search results that are sent back to the user to be maintained. In order to reduce the number of retrieved objects, the social-networking system may compare the number of objects retrieved from each data store with the final-scores for those objects as calculated by the social-networking system, and revise the parsing algorithm so query commands request fewer objects while still maintaining substantially the same quality of search results. For example, the social-networking system may access a set of archived search queries and optimize the parsing algorithm based on the final-scores of the objects retrieved by these queries. The archived queries may be submitted to one or more data stores, which may retrieve a first number of results based the number of objects to retrieve specified by the query commands generated for those queries by the parsing algorithm. Each retrieved object may then be scored to determine a final-score/rank, which may then be compared to the number of objects retrieved to determine whether the number of objects retrieved for a particular object-type can be reduced while still retrieving a sufficient number of the top-N scoring results. If so, then the parsing algorithm is may be revised so that query commands generated in response to particular queries specify retrieving fewer objects or object-types.

BRIEF DESCRIPTION OF THE DRAWINGS

[8] FIG. 1 illustrates an example network environment associated with a social-networking system.

[9] FIG. 2 illustrates an example social graph.

[10] FIG. 3 illustrates an example partitioning for storing objects of a social-networking system.

[11] FIG. 4 illustrates an example webpage of an online social network.

[12] FIGs. 5A-5B illustrate example queries of the social network.

[13] FIG. 6 illustrates an example method for improving the static-scoring of objects for search queries.

[14] FIG. 7 illustrates an example method for improving the parsing of search queries.

[15] FIG. 8 illustrates an example computer system.

DESCRIPTION OF EXAMPLE EMBODIMENTS

[16] **FIG. 1** illustrates an example network environment 100 associated with a social-networking system. Network environment 100 includes client system 130, social-networking system 160, and third-party system 170 connected to each other by a network 110. Although FIG. 1 illustrates a particular arrangement of client system 130, social-networking system 160, third-party system 170, and network 110, this disclosure contemplates any suitable arrangement of client system 130, social-networking system 160, third-party system 170, and network 110. As an example and not by way of limitation, two or more of client system 130, social-networking system 160, and third-party system 170 may be connected to each other directly, bypassing network 110. As another example, two or more of client system 130, social-networking system 160, and third-party system 170 may be physically or logically co-located with each other in whole or in part. Moreover, although FIG. 1 illustrates a particular number of client systems 130, social-networking systems 160, third-party systems 170, and networks 110, this disclosure contemplates any suitable number of client systems 130, social-networking systems 160, third-party systems 170, and networks 110. As an example and not by way of limitation, network environment 100 may include multiple client system 130, social-networking systems 160, third-party systems 170, and networks 110.

[17] This disclosure contemplates any suitable network 110. As an example and not by way of limitation, one or more portions of network 110 may include an ad hoc network, an intranet, an extranet, a virtual private network (VPN), a local area network (LAN), a wireless LAN (WLAN), a wide area network (WAN), a wireless WAN (WWAN), a metropolitan area

network (MAN), a portion of the Internet, a portion of the Public Switched Telephone Network (PSTN), a cellular telephone network, or a combination of two or more of these. Network 110 may include one or more networks 110.

[18] Links 150 may connect client system 130, social-networking system 160, and third-party system 170 to communication network 110 or to each other. This disclosure contemplates any suitable links 150. In particular embodiments, one or more links 150 include one or more wireline (such as for example Digital Subscriber Line (DSL) or Data Over Cable Service Interface Specification (DOCSIS)), wireless (such as for example Wi-Fi or Worldwide Interoperability for Microwave Access (WiMAX)), or optical (such as for example Synchronous Optical Network (SONET) or Synchronous Digital Hierarchy (SDH)) links. In particular embodiments, one or more links 150 each include an ad hoc network, an intranet, an extranet, a VPN, a LAN, a WLAN, a WAN, a WWAN, a MAN, a portion of the Internet, a portion of the PSTN, a cellular technology-based network, a satellite communications technology-based network, another link 150, or a combination of two or more such links 150. Links 150 need not necessarily be the same throughout network environment 100. One or more first links 150 may differ in one or more respects from one or more second links 150.

[19] In particular embodiments, client system 130 may be an electronic device including hardware, software, or embedded logic components or a combination of two or more such components and capable of carrying out the appropriate functionalities implemented or supported by client system 130. As an example and not by way of limitation, client system 130 may include a computer system such as a desktop computer, notebook or laptop computer, netbook, a tablet computer, e-book reader, GPS device, camera, personal digital assistant (PDA), handheld electronic device, cellular telephone, smartphone, other suitable electronic device, or any suitable combination thereof. This disclosure contemplates any suitable client systems 130. Client system 130 may enable a network user at client system 130 to access network 110. Client system 130 may enable its user to communicate with other users at other client systems 130.

[20] In particular embodiments, client system 130 may include a web browser 132, such as MICROSOFT INTERNET EXPLORER, GOOGLE CHROME or MOZILLA FIREFOX, and may have one or more add-ons, plug-ins, or other extensions, such as TOOLBAR or YAHOO TOOLBAR. A user at client system 130 may enter a Uniform Resource Locator

(URL) or other address directing the web browser 132 to a particular server (such as server 162, or a server associated with third-party system 170), and the web browser 132 may generate a Hyper Text Transfer Protocol (HTTP) request and communicate the HTTP request to server. The server may accept the HTTP request and communicate to client system 130 one or more Hyper Text Markup Language (HTML) files responsive to the HTTP request. Client system 130 may render a webpage based on the HTML files from the server for presentation to the user. This disclosure contemplates any suitable webpage files. As an example and not by way of limitation, webpages may render from HTML files, Extensible Hyper Text Markup Language (XHTML) files, or Extensible Markup Language (XML) files, according to particular needs. Such pages may also execute scripts such as, for example and without limitation, those written in JAVASCRIPT, JAVA, MICROSOFT SILVERLIGHT, combinations of markup language and scripts such as AJAX (Asynchronous JAVASCRIPT and XML), and the like. Herein, reference to a webpage encompasses one or more corresponding webpage files (which a browser may use to render the webpage) and vice versa, where appropriate.

[21] In particular embodiments, social-networking system 160 may be a network-addressable computing system that can host an online social network. Social-networking system 160 may generate, store, receive, and send social-networking data, such as, for example, user-profile data, concept-profile data, social-graph information, or other suitable data related to the online social network. Social-networking system 160 may be accessed by the other components of network environment 100 either directly or via network 110. In particular embodiments, social-networking system 160 may include one or more servers 162. Each server 162 may be a unitary server or a distributed server spanning multiple computers or multiple datacenters. Servers 162 may be of various types, such as, for example and without limitation, web server, news server, mail server, message server, advertising server, file server, application server, exchange server, database server, proxy server, another server suitable for performing functions or processes described herein, or any combination thereof. In particular embodiments, each server 162 may include hardware, software, or embedded logic components or a combination of two or more such components for carrying out the appropriate functionalities implemented or supported by server 162. In particular embodiments, social-networking system 164 may include one or more data stores 164. Data stores 164 may be used to store various types of information.

In particular embodiments, the information stored in data stores 164 may be organized according to specific data structures. In particular embodiments, each data store 164 may be a relational, columnar, correlation, or other suitable database. Although this disclosure describes or illustrates particular types of databases, this disclosure contemplates any suitable types of databases. Particular embodiments may provide interfaces that enable client system 130, social-networking system 160, or third-party system 170 to manage, retrieve, modify, add, or delete, the information stored in data store 164.

[22] In particular embodiments, social-networking system 160 may store one or more social graphs in one or more data stores 164. In particular embodiments, a social graph may include multiple nodes – which may include multiple user nodes (each corresponding to a particular user) or multiple concept nodes (each corresponding to a particular concept) – and multiple edges connecting the nodes. Social-networking system 160 may provide users of the online social network the ability to communicate and interact with other users. In particular embodiments, users may join the online social network via social-networking system 160 and then add connections (i.e., relationships) to a number of other users of social-networking system 160 whom they want to be connected to. Herein, the term "friend" may refer to any other user of social-networking system 160 with whom a user has formed a connection, association, or relationship via social-networking system 160.

[23] In particular embodiments, social-networking system 160 may provide users with the ability to take actions on various types of items or objects, supported by social-networking system 160. As an example and not by way of limitation, the items and objects may include groups or social networks to which users of social-networking system 160 may belong, events or calendar entries in which a user might be interested, computer-based applications that a user may use, transactions that allow users to buy or sell items via the service, interactions with advertisements that a user may perform, or other suitable items or objects. A user may interact with anything that is capable of being represented in social-networking system 160 or by an external system of third-party system 170, which is separate from social-networking system 160 and coupled to social-networking system 160 via a network 110.

[24] In particular embodiments, social-networking system 160 may be capable of linking a variety of entities. As an example and not by way of limitation, social-networking system 160 may enable users to interact with each other as well as receive content from third-party systems 170 or other entities, or to allow users to interact with these entities through an application programming interfaces (API) or other communication channels.

[25] In particular embodiments, third-party system 170 may include one or more types of servers, one or more data stores, one or more interfaces, including but not limited to APIs, one or more web services, one or more content sources, one or more networks, or any other suitable components, e.g., that servers may communicate with. A third-party system 170 may be operated by a different entity from an entity operating social-networking system 160. In particular embodiments, however, social-networking system 160 and third-party systems 170 may operate in conjunction with each other to provide social-networking services to users of social-networking system 160 or third-party systems 170. In this sense, social-networking system 160 may provide a platform, or backbone, which other systems, such as third-party systems 170, may use to provide social-networking services and functionality to users across the Internet.

[26] In particular embodiments, third-party system 170 may include a third-party content object provider. A third-party content object provider may include one or more sources of content objects, which may be communicated to client system 130. As an example and not by way of limitation, content objects may include information regarding things or activities of interest to the user, such as, for example, movie show times, movie reviews, restaurant reviews, restaurant menus, product information and reviews, or other suitable information. As another example and not by way of limitation, content objects may include incentive content objects, such as coupons, discount tickets, gift certificates, or other suitable incentive objects.

[27] In particular embodiments, social-networking system 160 also includes user-generated content objects, which may enhance a user's interactions with social-networking system 160. User-generated content may include anything a user can add, upload, send, or "post" to social-networking system 160. As an example and not by way of limitation, a user communicates posts to social-networking system 160 from client system 130. Posts may include data such as status updates or other textual data, location information, photos, videos, links,

music or other similar data or media. Content may also be added to social-networking system 160 by a third-party through a "communication channel," such as a newsfeed or stream.

[28] In particular embodiments, social-networking system 160 may include a variety of servers, sub-systems, programs, modules, logs, and data stores. In particular embodiments, social-networking system 160 may include one or more of the following: a web server, action logger, API-request server, relevance-and-ranking engine, content-object classifier, notification controller, action log, third-party-content-object-exposure log, inference module, authorization/privacy server, search module, ad-targeting module, user-interface module, user-profile store, connection store, third-party content store, or location store. Social-networking system 160 may also include suitable components such as network interfaces, security mechanisms, load balancers, failover servers, management-and-network-operations consoles, other suitable components, or any suitable combination thereof. In particular embodiments, social-networking system 160 may include one or more user-profile stores for storing user profiles. A user profile may include, for example, biographic information, demographic information, behavioral information, social information, or other types of descriptive information, such as work experience, educational history, hobbies or preferences, interests, affinities, or location. Interest information may include interests related to one or more categories. Categories may be general or specific. As an example and not by way of limitation, if a user "likes" an article about a brand of shoes the category may be the brand, or the general category of "shoes" or "clothing." A connection store may be used for storing connection information about users. The connection information may indicate users who have similar or common work experience, group memberships, hobbies, educational history, or are in any way related or share common attributes. The connection information may also include user-defined connections between different users and content (both internal and external). A web server may be used for linking social-networking system 160 to one or more client systems 130 or one or more third-party system 170 via network 110. The web server may include a mail server or other messaging functionality for receiving and routing messages between social-networking system 160 and one or more client systems 130. An API-request server may allow third-party system 170 to access information from social-networking system 160 by calling one or more APIs. An action logger may be used to receive communications from a web server about a user's actions

on or off social-networking system 160. In conjunction with the action log, a third-party-content-object log may be maintained of user exposures to third-party-content objects. A notification controller may provide information regarding content objects to client system 130. Information may be pushed to client system 130 as notifications, or information may be pulled from client system 130 responsive to a request received from client system 130. Authorization servers may be used to enforce one or more privacy settings of the users of social-networking system 160. A privacy setting of a user determines how particular information associated with a user can be shared. The authorization server may allow users to opt in or opt out of having their actions logged by social-networking system 160 or shared with other systems (e.g., third-party system 170), such as, for example, by setting appropriate privacy settings. Third-party-content-object stores may be used to store content objects received from third parties, such as third-party system 170. Location stores may be used for storing location information received from client systems 130 associated with users. Advertisement-pricing modules may combine social information, the current time, location information, or other suitable information to provide relevant advertisements, in the form of notifications, to a user.

[29] **FIG. 2** illustrates example social graph 200. In particular embodiments, social-networking system 160 may store one or more social graphs 200 in one or more data stores. In particular embodiments, social graph 200 may include multiple nodes – which may include multiple user nodes 202 or multiple concept nodes 204 – and multiple edges 206 connecting the nodes. Example social graph 200 illustrated in FIG. 2 is shown, for didactic purposes, in a two-dimensional visual map representation. In particular embodiments, social-networking system 160, client system 130, or third-party system 170 may access social graph 200 and related social-graph information for suitable applications. The nodes and edges of social graph 200 may be stored as data objects, for example, in a data store (such as a social-graph database). Such a data store may include one or more searchable or queryable indexes of nodes or edges of social graph 200.

[30] In particular embodiments, a user node 202 may correspond to a user of social-networking system 160. As an example and not by way of limitation, a user may be an individual (human user), an entity (e.g., an enterprise, business, or third-party application), or a group (e.g.,

of individuals or entities) that interacts or communicates with or over social-networking system 160. In particular embodiments, when a user registers for an account with social-networking system 160, social-networking system 160 may create a user node 202 corresponding to the user, and store the user node 202 in one or more data stores. Users and user nodes 202 described herein may, where appropriate, refer to registered users and user nodes 202 associated with registered users. In addition or as an alternative, users and user nodes 202 described herein may, where appropriate, refer to users that have not registered with social-networking system 160. In particular embodiments, a user node 202 may be associated with information provided by a user or information gathered by various systems, including social-networking system 160. As an example and not by way of limitation, a user may provide his or her name, profile picture, contact information, birth date, sex, marital status, family status, employment, education background, preferences, interests, or other demographic information. In particular embodiments, a user node 202 may be associated with one or more data objects corresponding to information associated with a user. In particular embodiments, a user node 202 may correspond to one or more webpages.

[31] In particular embodiments, a concept node 204 may correspond to a concept. As an example and not by way of limitation, a concept may correspond to a place (such as, for example, a movie theater, restaurant, landmark, or city); a website (such as, for example, a website associated with social-network system 160 or a third-party website associated with a web-application server); an entity (such as, for example, a person, business, group, sports team, or celebrity); a resource (such as, for example, an audio file, video file, digital photo, text file, structured document, or application) which may be located within social-networking system 160 or on an external server, such as a web-application server; real or intellectual property (such as, for example, a sculpture, painting, movie, game, song, idea, photograph, or written work); a game; an activity; an idea or theory; another suitable concept; or two or more such concepts. A concept node 204 may be associated with information of a concept provided by a user or information gathered by various systems, including social-networking system 160. As an example and not by way of limitation, information of a concept may include a name or a title; one or more images (e.g., an image of the cover page of a book); a location (e.g., an address or a geographical location); a website (which may be associated with a URL); contact information

(e.g., a phone number or an email address); other suitable concept information; or any suitable combination of such information. In particular embodiments, a concept node 204 may be associated with one or more data objects corresponding to information associated with concept node 204. In particular embodiments, a concept node 204 may correspond to one or more webpages.

[32] In particular embodiments, a node in social graph 200 may represent or be represented by a webpage (which may be referred to as a "profile page"). Profile pages may be hosted by or accessible to social-networking system 160. Profile pages may also be hosted on third-party websites associated with a third-party server 170. As an example and not by way of limitation, a profile page corresponding to a particular external webpage may be the particular external webpage and the profile page may correspond to a particular concept node 204. Profile pages may be viewable by all or a selected subset of other users. As an example and not by way of limitation, a user node 202 may have a corresponding user-profile page in which the corresponding user may add content, make declarations, or otherwise express himself or herself. As another example and not by way of limitation, a concept node 204 may have a corresponding concept-profile page in which one or more users may add content, make declarations, or express themselves, particularly in relation to the concept corresponding to concept node 204.

[33] In particular embodiments, a concept node 204 may represent a third-party webpage or resource hosted by third-party system 170. The third-party webpage or resource may include, among other elements, content, a selectable or other icon, or other inter-actable object (which may be implemented, for example, in JavaScript, AJAX, or PHP codes) representing an action or activity. As an example and not by way of limitation, a third-party webpage may include a selectable icon such as "like," "check in," "eat," "recommend," or another suitable action or activity. A user viewing the third-party webpage may perform an action by selecting one of the icons (e.g., "eat"), causing client system 130 to send to social-networking system 160 a message indicating the user's action. In response to the message, social-networking system 160 may create an edge (e.g., an "eat" edge) between a user node 202 corresponding to the user and a concept node 204 corresponding to the third-party webpage or resource and store edge 206 in one or more data stores.

[34] In particular embodiments, a pair of nodes in social graph 200 may be connected to each other by one or more edges 206. An edge 206 connecting a pair of nodes may represent a relationship between the pair of nodes. In particular embodiments, an edge 206 may include or represent one or more data objects or attributes corresponding to the relationship between a pair of nodes. As an example and not by way of limitation, a first user may indicate that a second user is a "friend" of the first user. In response to this indication, social-networking system 160 may send a "friend request" to the second user. If the second user confirms the "friend request," social-networking system 160 may create an edge 206 connecting the first user's user node 202 to the second user's user node 202 in social graph 200 and store edge 206 as social-graph information in one or more of data stores 24. In the example of FIG. 2, social graph 200 includes an edge 206 indicating a friend relation between user nodes 202 of user "A" and user "B" and an edge indicating a friend relation between user nodes 202 of user "C" and user "B." Although this disclosure describes or illustrates particular edges 206 with particular attributes connecting particular user nodes 202, this disclosure contemplates any suitable edges 206 with any suitable attributes connecting user nodes 202. As an example and not by way of limitation, an edge 206 may represent a friendship, family relationship, business or employment relationship, fan relationship, follower relationship, visitor relationship, subscriber relationship, superior/subordinate relationship, reciprocal relationship, non-reciprocal relationship, another suitable type of relationship, or two or more such relationships. Moreover, although this disclosure generally describes nodes as being connected, this disclosure also describes users or concepts as being connected. Herein, references to users or concepts being connected may, where appropriate, refer to the nodes corresponding to those users or concepts being connected in social graph 200 by one or more edges 206.

[35] In particular embodiments, an edge 206 between a user node 202 and a concept node 204 may represent a particular action or activity performed by a user associated with user node 202 toward a concept associated with a concept node 204. As an example and not by way of limitation, as illustrated in FIG. 2, a user may "like," "attended," "played," "listened," "cooked," "worked at," or "watched" a concept, each of which may correspond to a edge type or subtype. A concept-profile page corresponding to a concept node 204 may include, for example, a selectable "check in" icon (such as, for example, a clickable "check in" icon) or a selectable

"add to favorites" icon. Similarly, after a user clicks these icons, social-networking system 160 may create a "favorite" edge or a "check in" edge in response to a user's action corresponding to a respective action. As another example and not by way of limitation, a user (user "C") may listen to a particular song ("Imagine") using a particular application (SPOTIFY, which is an online music application). In this case, social-networking system 160 may create a "listened" edge 206 and a "used" edge (as illustrated in FIG. 2) between user nodes 202 corresponding to the user and concept nodes 204 corresponding to the song and application to indicate that the user listened to the song and used the application. Moreover, social-networking system 160 may create a "played" edge 206 (as illustrated in FIG. 2) between concept nodes 204 corresponding to the song and the application to indicate that the particular song was played by the particular application. In this case, "played" edge 206 corresponds to an action performed by an external application (SPOTIFY) on an external audio file (the song "Imagine"). Although this disclosure describes particular edges 206 with particular attributes connecting user nodes 202 and concept nodes 204, this disclosure contemplates any suitable edges 206 with any suitable attributes connecting user nodes 202 and concept nodes 204. Moreover, although this disclosure describes edges between a user node 202 and a concept node 204 representing a single relationship, this disclosure contemplates edges between a user node 202 and a concept node 204 representing one or more relationships. As an example and not by way of limitation, an edge 206 may represent both that a user likes and has used at a particular concept. Alternatively, another edge 206 may represent each type of relationship (or multiples of a single relationship) between a user node 202 and a concept node 204 (as illustrated in FIG. 2 between user node 202 for user "E" and concept node 204 for "SPOTIFY").

[36] In particular embodiments, social-networking system 160 may create an edge 206 between a user node 202 and a concept node 204 in social graph 200. As an example and not by way of limitation, a user viewing a concept-profile page (such as, for example, by using a web browser or a special-purpose application hosted by the user's client system 130) may indicate that he or she likes the concept represented by the concept node 204 by clicking or selecting a "Like" icon, which may cause the user's client system 130 to send to social-networking system 160 a message indicating the user's liking of the concept associated with the concept-profile page. In response to the message, social-networking system 160 may create an edge 206 between

user node 202 associated with the user and concept node 204, as illustrated by "like" edge 206 between the user and concept node 204. In particular embodiments, social-networking system 160 may store an edge 206 in one or more data stores. In particular embodiments, an edge 206 may be automatically formed by social-networking system 160 in response to a particular user action. As an example and not by way of limitation, if a first user uploads a picture, watches a movie, or listens to a song, an edge 206 may be formed between user node 202 corresponding to the first user and concept nodes 204 corresponding to those concepts. Although this disclosure describes forming particular edges 206 in particular manners, this disclosure contemplates forming any suitable edges 206 in any suitable manner.

[37] **FIG. 3** illustrates an example partitioning for storing objects of social-networking system 160. A plurality of data stores 164 (which may also be called "verticals") may store objects of social-networking system 160. The amount of data (e.g., data for a social graph 200) stored in the data stores may be very large. As an example and not by way of limitation, a social graph used by Facebook, Inc. of Menlo Park, CA can have a number of nodes in the order of 10^8, and a number of edges in the order of 10^{10}. Typically, a large collection of data such as a large database may be divided into a number of partitions. As the index for each partition of a database is smaller than the index for the overall database, the partitioning may improve performance in accessing the database. As the partitions may be distributed over a large number of servers, the partitioning may also improve performance and reliability in accessing the database. Ordinarily, a database may be partitioned by storing rows (or columns) of the database separately. In particular embodiments, a database maybe partitioned by based on object-types. Data objects may be stored in a plurality of partitions, each partition holding data objects of a single object-type. In particular embodiments, social-networking system 160 may retrieve search results in response to a search query by submitting the search query to a particular partition storing objects of the same object-type as the search query's expected results. Although this disclosure describes storing objects in a particular manner, this disclosure contemplates storing objects in any suitable manner.

[38] In particular embodiments, each object may correspond to a particular node of a social graph 200. An edge 206 connecting the particular node and another node may indicate a relationship between objects corresponding to these nodes. In addition to storing objects, a

particular data store may also store social-graph information relating to the object. Alternatively, social-graph information about particular objects may be stored in a different data store from the objects. Social-networking system 160 may update the search index of the data store based on newly received objects, and relationships associated with the received objects.

[39] In particular embodiments, each data store 164 may be configured to store objects of a particular one of a plurality of object-types in respective data storage devices 340. An object-type may be, for example, a user, a photo, a post, a comment, a message, an event listing, a webpage, an application, a user-profile page, a concept-profile page, a user group, an audio file, a video, an offer/coupon, or another suitable type of object. Although this disclosure describes particular types of objects, this disclosure contemplates any suitable types of objects. As an example and not by way of limitation, a user vertical P1 illustrated in FIG. 3 may store user objects. Each user object stored in the user vertical P1 may comprise an identifier (e.g., a character string), a user name, and a profile picture for a user of the online social network. Social-networking system 160 may also store in the user vertical P1 information associated with a user object such as language, location, education, contact information, interests, relationship status, a list of friends/contacts, a list of family members, privacy settings, and so on. As an example and not by way of limitation, a post vertical P2 illustrated in FIG. 3 may store post objects. Each post object stored in the post vertical P2 may comprise an identifier, a text string for a post posted to social-networking system 160. Social-networking system 160 may also store in the post vertical P2 information associated with a post object such as a time stamp, an author, privacy settings, users who like the post, a count of likes, comments, a count of comments, location, and so on. As an example and not by way of limitation, a photo vertical P3 may store photo objects (or objects of other media types such as video or audio). Each photo object stored in the photo vertical P3 may comprise an identifier and a photo. Social-networking system 160 may also store in the photo vertical P3 information associated with a photo object such as a time stamp, an author, privacy settings, users who are tagged in the photo, users who like the photo, comments, and so on. In particular embodiments, each data store may also be configured to store information associated with each stored object in data storage devices 340.

[40] In particular embodiments, objects stored in each vertical 164 may be indexed by one or more search indices. The search indices may be hosted by respective index server 330

comprising one or more computing devices (e.g., servers). The index server 330 may update the search indices based on data (e.g., a photo and information associated with a photo) submitted to social-networking system 160 by users or other processes of social-networking system 160 (or a third-party system). The index server 330 may also update the search indices periodically (e.g., every 24 hours). The index server 330 may receive a query comprising a search term, and access and retrieve search results from one or more search indices corresponding to the search term. In some embodiments, a vertical corresponding to a particular object-type may comprise a plurality of physical or logical partitions, each comprising respective search indices.

[41] In particular embodiments, social-networking system 160 may receive a search query from a PHP (Hypertext Preprocessor) process 310. The PHP process 310 may comprise one or more computing processes hosted by one or more servers 162 of social-networking system 160. The search query may be a text string or a structured query submitted to the PHP process by a user or another process of social-networking system 160 (or third-party system 170).

[42] More information on indexes and search queries may be found in U.S. Patent No. 9,158,801, filed 27 July 2012, U.S. Patent No. 8,983,991, filed 27 July 2012, U.S. Patent No. 8,935,271 filed 21 December 2012, and U.S. Patent Publication No. US2014/0324963, filed 25 April 2013.

[43] **FIG. 4** illustrates an example webpage of an online social network. In particular embodiments, a user may submit a query to the social-network system 160 by inputting text into query field 450. A user of an online social network may search for particular content objects (hereinafter "objects") or content-object-types (hereinafter "object-types") associated with the online social network (e.g., users, concepts, webpages, external content or resources) by providing a short phrase describing the object or object-type, often referred to as a "search query," to a search engine. The query may be a text query and may comprise one or more character strings (which may include one or more n-grams). In general, a user may input any character string comprising one or more characters into query field 450 to search for objects on social-networking system 160 that match at least a portion of the character string. Social-networking system 160 may then search one or more verticals 164 to identify objects matching the query. The search engine may conduct a search based on the query using various search

algorithms and generate search results that identify objects (e.g., user-profile pages, content-profile pages, or external resources) that are most likely to be related to the search query. To conduct a search, a user may input or send a search query to the search engine. In response, the search engine may identify one or more resources that are likely to be related to the search query, each of which may individually be referred to as a "search result," or collectively be referred to as the "search results" corresponding to the search query. The identified objects may include, for example, social-graph elements (i.e., user nodes 202, concept nodes 204, edges 206), profile pages, external webpages, or any combination thereof. Social-networking system 160 may then generate a search-results webpage with search results corresponding to the identified objects and send the search-results webpage to the user. In particular embodiments, the search engine may limit its search to objects associated with the online social network. However, in particular embodiments, the search engine may also search for objects associated with other sources, such as third-party system 170, the internet or World Wide Web, or other suitable sources. Although this disclosure describes querying social-networking system 160 in a particular manner, this disclosure contemplates querying social-networking system 160 in any suitable manner.

[44] In connection with search queries and search results, particular embodiments may utilize one or more systems, components, elements, functions, methods, operations, or steps disclosed in U.S. Patent No. 8,402,094, filed 11 August 2006, U.S. Patent No. 8,572,129, filed 19 April 2010, U.S. Patent Publication No. US2012/0166433, filed 22 December 2010, U.S. Patent Publication No. US2012/0166532, filed 23 December 2010, U.S. Patent No. 8,782,080, filed 23 July 2012, and U.S. Patent No. 8,868,603, filed 31 December 2012.

[45] In particular embodiments, social-networking system 160 may improve the processing of search queries by improving the static scores/rankings of objects stored in data stores 164. When social-networking system 160 retrieves objects from a data store 164 in response to a query, the objects may be retrieved based on a pre-determined static-score or static-rank associated with the object (which may be based, for example, on how the objects are indexed). The objects retrieved from all data stores 164 may then be aggregated and scored (based on a variety of factors, such as, for example, relevance to the query, social-graph affinity, user history, etc.) by social-networking system 160, and these final-scores or final-ranks may then be used to determine which objects are generated as search results that are displayed to the

querying user. However, this process may be inefficient if social-networking system 160 has to retrieve an excess of objects from the data stores 164 in order to generate a sufficient number of search results. As an example and not by way of limitation, if social-networking system 160 retrieves 100 matching objects from a particular data store 164, where each object has an associated static-rank, and then these 100 objects are scored by social-networking system 160, the top-five ranked objects might be, for example, objects having static-ranks of 4, 12, 20, 78, and 95. This process could be improved if the static-ranks of objects more closely match the final-ranks determined by social-networking system 160 when generating search results for a user. This may allow social-networking system 160 to reduce the number of matching objects that need to be retrieved in order to generate a sufficient number of search results in response to a query. In order to improve the static-scores of objects indexed in one or more data stores 164, social-networking system 160 may compared the static-scores of objects retrieved from a data store 164 with the final-score calculated by social-networking system 160 in order to generate search results for a user, and revise or adjust the static-scores (or the scoring algorithm used to calculate the static-scores) of the indexed objects so they more closely match the final-scores. As an example and not by way of limitation, social-networking system 160 may access a set of archived search queries and optimize the static-scores of objects retrieved by these queries. The archived queries may be submitted to one or more data stores 164, which may retrieve a first number of results based on their static-ranks. Each retrieved objects may then scored to determine a final-rank, which may then be compared to the static-rank, and the static-rank may be modified so it more closely matches the final-rank. This may be done for a variety of queries or query types, so that the static-scores are optimized to match the final-scores as closely as possible for a variety of queries. Although this disclosure describes improving static rankings in a particular manner, this disclosure contemplates improving static rankings in any suitable manner.

[46] In particular embodiments, social-networking system 160 may improve the processing of search queries by improving how query commands are generated. When a query is parsed to generate a query command, the query command may specify a particular number of objects to retrieve of one or more object-types (e.g., the number to score for each vertical 164 accessed). As an example and not by way of limitation, in response to the text query "steph",

social-networking system 160 may generate a query command that requests ten first-degree connections of the querying user from a users vertical 164, fifty second-degree connections, and twenty pages from a pages vertical 164. The number of objects to retrieve of each object-type may be specified by parsing-configuration parameters of the parsing algorithm used to generate the query commands. The retrieved objects may then be scored/ranked and the top-N scoring objects may be sent to the querying user. However, this process may be inefficient if social-networking system 160 has to retrieve an excess of one or more object-types from particular verticals 164 in order to retrieve the top-N scoring objects, particularly with respect to inefficient use of processing (CPU) power. This process could be improved if the number of objects retrieved from each vertical 164 could be reduced while still retrieving some or all of the objects with the best final-scores/ranks, allowing the quality of the generated search results that are sent back to the user to be maintained. As an example and not by way of limitation, continuing with the prior example, it may be possible to generate search results of the same quality (i.e., still retrieve substantially all of the top-N scoring objects) by generating a query command that only requests ten first-degree connections, twenty-five second-degree connections (instead of fifty), and ten pages (instead of twenty). In order to reduce the number of retrieved objects, social-networking system 160 may compare the number of objects retrieved from each vertical 164 with the final-scores for those objects as calculated by social-networking system 160, and revise the parsing algorithm so query commands request fewer objects while still maintaining substantially the same quality of search results. As an example and not by way of limitation, social-networking system 160 may access a set of archived search queries and optimize the parsing algorithm based on the final-scores of the objects retrieved by these queries. The archived queries may be submitted to one or more data stores 164, which may retrieve a first number of results based the number of objects to retrieve specified by the query commands generated for those queries by the parsing algorithm. Each retrieved object may then be scored to determine a final-score/rank, which may then be compared to the number of objects retrieved to determine whether the number of objects retrieved for a particular object-type can be reduced while still retrieving the top-N scoring results (or at least retrieving a sufficient number of the top-N results). If so, then the parsing algorithm is may be revised so that query commands generated in response to particular queries specify retrieving fewer objects or object-types.

Although this disclosure describes improving how query commands are generated in a particular manner, this disclosure contemplates improving how query commands are improved in any suitable manner.

[47] In particular embodiments, social-networking system 160 may access a set of queries of the online social network received from one or more users of the online social network. Search queries submitted by users may be saved by social-networking system 160 and later retrieved in order to run experiments to optimize the processing of search queries. As an example and not by way of limitation, the set of queries may comprise a plurality of archived queries from a plurality of users of the online social network. Experiments using the archived queries may be done, for example, by having social-networking system 160 execute the queries, analyze the parsing of the queries and the objects retrieved by the queries, and then optimizing particular aspects of the querying process. Although this disclosure describes accessing particular sets of queries in a particular manner, this disclosure contemplates accessing any suitable sets of queries in any suitable manner.

[48] **FIGs. 5A-5B** illustrate example queries of the online social network. In particular embodiments, in response to a text query received from a first user (i.e., the querying user), social-networking system 160 may parse the text query and identify portions of the text query that correspond to particular social-graph elements. Social-networking system 160 may then generate a set of structured queries, where each structured query corresponds to one of the possible matching social-graph elements. These structured queries may be based on strings generated by a grammar model, such that they are rendered in a natural-language syntax with references to the relevant social-graph elements. These structured queries may be presented to the querying user, who can then select among the structured queries to indicate that the selected structured query should be run by social-networking system 160. FIGs. 5A-5B illustrate various example text queries in query field 450 and various structured queries generated in response in drop-down menus 400 (although other suitable graphical user interfaces are possible). By providing suggested structured queries in response to a user's text query, social-networking system 160 may provide a powerful way for users of the online social network to search for elements represented in the social graph 200 based on their social-graph attributes and their relation to various social-graph elements. Structured queries may allow a querying user to search

for content that is connected to particular users or concepts in the social graph 200 by particular edge-types. The structured queries may be sent to the first user and displayed in a drop-down menu 400 (via, for example, a client-side typeahead process), where the first user can then select an appropriate query to search for the desired content. Some of the advantages of using the structured queries described herein include finding users of the online social network based upon limited information, bringing together virtual indexes of content from the online social network based on the relation of that content to various social-graph elements, or finding content related to you and/or your friends. Although this disclosure describes and FIGs. 5A-5B illustrate generating particular structured queries in a particular manner, this disclosure contemplates generating any suitable structured queries in any suitable manner.

[49] In particular embodiments, social-networking system 160 may receive from a querying/first user (corresponding to a first user node 202) an unstructured text query. As an example and not by way of limitation, a first user may want to search for other users who: (1) are first-degree friends of the first user; and (2) are associated with Stanford University (i.e., the user nodes 202 are connected by an edge 206 to the concept node 204 corresponding to the school "Stanford"). The first user may then enter a text query "friends stanford" into query field 450, as illustrated in FIGs. 5A-5B. As the querying user enters this text query into query field 450, social-networking system 160 may provide various suggested structured queries, as illustrated in drop-down menus 400. As used herein, an unstructured text query refers to a simple text string inputted by a user. The text query may, of course, be structured with respect to standard language/grammar rules (e.g. English language grammar). However, the text query will ordinarily be unstructured with respect to social-graph elements. In other words, a simple text query will not ordinarily include embedded references to particular social-graph elements. Thus, as used herein, a structured query refers to a query that contains references to particular social-graph elements, allowing the search engine to search based on the identified elements. Furthermore, the text query may be unstructured with respect to formal query syntax. In other words, a simple text query will not necessarily be in the format of a query command that is directly executable by a search engine (e.g., the text query "friends stanford" could be parsed to form the query command "intersect(school(Stanford University), friends(me)", which could be executed as a query in a social-graph database). Although this disclosure describes receiving

particular queries in a particular manner, this disclosure contemplates receiving any suitable queries in any suitable manner.

[50] In particular embodiments, social-networking system 160 may parse the unstructured text query (also simply referred to as a search query) received from the first user (i.e., the querying user) to identify one or more n-grams. In general, an n-gram is a contiguous sequence of *n* items from a given sequence of text or speech. The items may be characters, phonemes, syllables, letters, words, base pairs, prefixes, or other identifiable items from the sequence of text or speech. The n-gram may comprise one or more characters of text (letters, numbers, punctuation, etc.) entered by the querying user. An n-gram of size one can be referred to as a "unigram," of size two can be referred to as a "bigram" or "digram," of size three can be referred to as a "trigram," and so on. Each n-gram may include one or more parts from the text query received from the querying user. In particular embodiments, each n-gram may comprise a character string (e.g., one or more characters of text) entered by the first user. As an example and not by way of limitation, social-networking system 160 may parse the text query "friends stanford" to identify the following n-grams: friends; stanford; friends stanford. As another example and not by way of limitation, social-networking system 160 may parse the text query "friends in palo alto" to identify the following n-grams: friends; in; palo; alto; friends in; in palo; palo alto; friend in palo; in palo also; friends in palo alto. In particular embodiments, each n-gram may comprise a contiguous sequence of *n* items from the text query. Although this disclosure describes parsing particular queries in a particular manner, this disclosure contemplates parsing any suitable queries in any suitable manner. In connection with element detection and parsing search queries, particular embodiments may utilize one or more systems, components, elements, functions, methods, operations, or steps disclosed in U.S. Patent No. 8,782,080, filed 23 July 2012, U.S. Patent No. 8,868,603, filed 31 December 2012.

[51] In particular embodiments, social-networking system 160 may generate one or more structured queries. The structured queries may be based on the natural-language strings generated by one or more grammars of a grammar model. Each structured query may include references to one or more of the identified nodes or one or more of the identified edges 206. This type of structured query may allow social-networking system 160 to more efficiently search for resources and content related to the online social network (such as, for example, profile pages)

by searching for content connected to or otherwise related to the identified user nodes 202 and the identified edges 206. As an example and not by way of limitation, in response to the text query, "show me friends of my girlfriend," social-networking system 160 may generate a structured query "Friends of Stephanie," where "Friends" and "Stephanie" in the structured query are references corresponding to particular social-graph elements. The reference to "Stephanie" would correspond to a particular user node 202 (where social-networking system 160 has parsed the n-gram "my girlfriend" to correspond with a user node 202 for the user "Stephanie"), while the reference to "Friends" would correspond to friend-type edges 206 connecting that user node 202 to other user nodes 202 (i.e., edges 206 connecting to "Stephanie's" first-degree friends). When executing this structured query, social-networking system 160 may identify one or more user nodes 202 connected by friend-type edges 206 to the user node 202 corresponding to "Stephanie". As another example and not by way of limitation, in response to the text query, "friends who like facebook," social-networking system 160 may generate a structured query "Friends who like Facebook," where "Friends," "like," and "Facebook" in the structured query are references corresponding to particular social-graph elements as described previously (i.e., a friend-type edge 206, a like-type edge 206, and concept node 204 corresponding to the company "Facebook"). Although this disclosure describes generating particular structured queries in a particular manner, this disclosure contemplates generating any suitable structured queries in any suitable manner.

[52] In particular embodiments, social-networking system 160 may receive from the querying user a selection of one of the structured queries. The nodes and edges referenced in the received structured query may be referred to as the selected nodes and selected edges, respectively. As an example and not by way of limitation, the web browser 132 on the querying user's client system 130 may display the sent structured queries in a drop-down menu 300, as illustrated in FIGs. 5A-5B, which the user may then click on or otherwise select (e.g., by simply keying "enter" on his keyboard) to indicate the particular structured query the user wants social-networking system 160 to execute. Upon selecting a particular structured query, the user's client system 130 may call or otherwise instruct to social-networking system 160 to execute the selected structured query. Although this disclosure describes receiving selections of particular

structured queries in a particular manner, this disclosure contemplates receiving selections of any suitable structured queries in any suitable manner.

[53] More information on generating structured queries and grammar models may be found in U.S. Patent No. 8,782,080, filed 23 July 2012, U.S. Patent No. 9,105,068, filed 12 November 2012, and U.S. Patent No. 9,367,607, filed 31 December 2012.

[54] In particular embodiments, social-networking system 160 may generate a query command based on a query (e.g., a text query or a structured query) received from a querying user. The query command may then be used in a search against objects in a data store 164 of the social-networking system 160. In particular embodiments, the query command may be provided for a search using search indices for one or more data stores or verticals of social-networking system 160. The query command may comprise one or more query constraints. Each query constraint may be identified by social-networking system 160 based on a parsing of the query by a parsing algorithm. Each query constraint may be a request for a particular object-type. In particular embodiments, the query command may comprise query constraints in symbolic expression or s-expression. Social-networking system 160 may parse the structured query "Photos I like" to a query command (photos_liked_by:<me>). The query command (photos_liked_by: <me>) denotes a query for photos liked by a user (i.e., <me>, which corresponding to the querying user), with a single result-type of photo. The query constraint may include, for example, social-graph constraints (e.g., requests for particular nodes or nodes-types, or requests for nodes connected to particular edges or edge-types), object constraints (e.g., request for particular objects or object-types), location constraints (e.g., requests for objects or social-graph entities associates with particular geographic locations), other suitable constraints, or any combination thereof. In particular embodiments, a query command may comprise prefix and an object. The object may correspond to a particular node in the social graph 200, while the prefix may correspond to a particular edge 206 or edge-type (indicating a particular type of relationship) connecting to the particular node in the social graph 200. As an example and not by way of limitation, the query command (pages_liked_by:<user>) comprises a prefix pages_liked_by, and an object <user>. Although this disclosure describes generating particular query commands in a particular manner, this disclosure contemplates generating any suitable query commands in any suitable manner. In particular embodiments, social-networking system

160 may generate a query command comprising a "weak and" (WAND) or "strong or" operator (SOR). More information on WAND and SOR operators may be found in U.S. Patent No. 8,983,991, filed 27 July 2012, and U.S. Patent No. 9,367,625, filed 03 May 2013.

[55] In particular embodiments, the parsing algorithm used to generate query commands may comprise one or more parsing-configuration parameters. The parsing-configuration parameters may specify how to generate a query command for a particular type of query received from a user. The parsing-configuration parameters may specify, for example, instructions for generating a query commands having a specified number of query constraints for a specified number of objects of a specified object-type to be retrieved from a specified number of data stores 164. In other words, the parsing-configuration parameters may specify the types of objects that should be searched and the types/number of verticals 164 that should be accessed. For each vertical 164 accessed, the parsing-configuration parameters may specify the number of objects to retrieve from each vertical 164. As an example and not by way of limitation, in response to a search query input "kais", social-networking system 160 may generate the following query command:

(AND (name: "kais")
(OR friends_of: (friends_of: <me>) : num_to_score: 50)
(OR pages: <> : num_to_score: 25)).

This query command contains a first query constraint (OR friends_of: (friends_of: <me>) : num_to_score: 50), which instructs social-networking system 160 to access a users vertical 164 to search for users that are friends-of-friends of the querying user that match the character string "kais", and to retrieve the top fifty results. The second query constraint, (OR pages: <> : num_to_score: 25), instructs social-networking system 160 to access a webpages vertical 164 to search for pages that match the character string "kais", and to retrieve the top twenty-five results. However, this process may be inefficient if social-networking system 160 has to retrieve an excess of objects of particular object-types in order to generate a sufficient number of search results. In order to improved the amount of processing (CPU) power consumed when processing queries, social-networking system 160 may use parsing-configuration parameters that minimize the number of object-types and the number of objects retrieved from each vertical 164, while still retrieving a sufficient number of object to retrieve the top-N scoring objects. As an example and

not by way of limitation, continuing with the prior example, in order to generate the top-10 search results, social-networking system 160 may only need to retrieve the top twenty-five friends-of-friends and the top fifteen pages. This may be because, for example, the friends-of-friends ranked twenty-six to fifty all have final-scores that put them outside of the top-10 search results. Thus, fewer users need to be pulled in order to maintain the same quality of search results. This allows the processing power consumed by each search query. The parsing-configuration parameters may be revised so that more or less object-types (and possibly additional verticals 164) are searched, or that more or less objects of each object-type are retrieved. Although this disclosure describes generating particular query commands in a particular manner, this disclosure contemplates generating any suitable query commands in any suitable manner.

[56] In particular embodiments, social-networking system 160 may retrieve objects from one or more verticals 164 that match at least a portion of the query constraints of a query command. Social-networking system 160 may access one or more verticals 164 in response to a search query received from a user, as specified by the query command. Each vertical 164 may store one or more objects associated with the online social network. The number and type of verticals 164 accessed in response to the search query may be based on the query constraints of the query command. Each vertical 164 may store objects associated with the online social network of the object-type specified by the query constraint. As an example and not by way of limitation, one of the query constraints of a query command for users, social-networking system 160 may access a users vertical 164 to identify one or more users who match the query. Social-networking system 160 may identify matching objects in any suitable manner, such as, for example, by using one or more string matching algorithms to match the character string with a string of characters associated with each of one or more of the objects. As an example and not by way of limitation, in response to a search query input "kais", social-networking system 160 may access one or more users verticals 164 and one or more pages verticals 164 and search the accessed verticals to identify objects (e.g., user-profile pages or concept-profile pages) stored in those verticals. Social-networking system 160 may submit the following query command to each accessed vertical:

(AND (name: "kais")

(OR friends_of: (friends_of: <me>) : num_to_score: 50)

(OR pages: <> : num_to_score: 25)).

Social-networking system 160 may access the index servers 330 of each vertical 164, causing index server 330 to return results that match the query command. As an example and not by way of limitation, social-networking system 160 may access index server 330 of a users vertical 164, causing index server 330 to identify users <Kaisen L>, <Nathen Kaiser>, <Catie Kaiser>, and <Alex Kaiser> (each represented by an user identifier). That is, users <Kaisen L>, <Nathen Kaiser>, <Catie Kaiser>, and <Alex Kaiser> may have a name matching "kais". Furthermore, each of these identified users matches the query constraint (friends_of: (friends_of: <me>)), which request objects corresponding to user that are friend-of-friends of the querying user. Social-networking system 160 may also access index server 330 of a pages vertical 164, causing index server 330 to identify the page for the band <Kaiser Chiefs>. That is, the band <Kaiser Chiefs> has a name matching "kais". Furthermore, the identify page matches the query constraint (pages: <>), which request objects corresponding to pages. In particular embodiments, social-networking system 160 may identify objects matching a query command by traversing the social graph 200 from the particular node along the particular connecting edges 206 (or edge-types) to nodes corresponding to objects specified by query command in order to identify one or more search results. As an example and not by way of limitation, the query command (pages_liked_by:<user>) may be executed by social-networking system 160 by traversing the social graph 200 from a user node 202 corresponding to <user> along like-type edges 206 to concept nodes 204 corresponding to pages liked by <user>. Although this disclosure describes searching for objects in a particular manner, this disclosure contemplates searching for objects in any suitable manner.

[57] In particular embodiments, when searching verticals 164 to identify matching objects, social-networking system 160 may only identify and score up to a threshold number of matching nodes in a particular vertical 164. When social-networking system 160 retrieves objects from a vertical 164 in response to a query (or a particular query constraint), the objects may be retrieved based on a static-score or static-rank of the indexed objects. As an example and not by way of limitation, the objects with the static-ranks, up to the threshold number, may be retrieved and further processed, for example, by a scoring algorithm that may calculate a final-

score for the retrieved objects based on a variety of factors in order to determine search results to send back to the querying user. Each object stored in a vertical 164 may be associated with a pre-determined static-score based on a static-scoring algorithm. In particular embodiments, the pre-determined static-score of each object may a pre-determined ranking of the object for a particular type of query. As an example and not by way of limitation, when a structured query comprises "friends of Alex" (which may be a portion of a larger query, such as, "photos of friends of Alex", or "friends of friends of Alex"), user nodes 202 corresponding to friends of the user "Alex" may have pre-determined static-scores with respect to this structured query. Alex's top-three friends may be, for example, "Larry", "Moe," and "Joe", ranked in that order. Thus, when searching a users vertical 164 in response to the query "friends of Alex" (or the query command friends_of:<Alex>), the users "Larry", "Moe," and "Joe" may be retrieved as the top-three objects. When searching a vertical 164, social-networking system 160 may retrieve objects based on the static-scores of the objects, where the objects with the highest/best static-scores may be retrieved. The threshold number of matching objects may then be scored and ranked by the social-networking system 160. The threshold number may be chosen to enhance search quality or to optimize the processing of search results. As an example and not by way of limitation, social-networking system 160 may only identify the top-N matching objects (i.e., the number to score, or "num_to_score" for an s-expression in the examples used herein) in a users vertical 164 in response to a query command requesting users. The top-N objects may be determined by their static-scores (e.g., ranking based on the current social-graph affinity of the user with respect to the querying user) of the objects in a search index corresponding to the users vertical 164. The static-scores may be pre-determined by social-networking system 160 using a static-scoring algorithm. However, this process may be inefficient if social-networking system 160 has to retrieve an excess number of objects from a vertical 164 in order to find the top-N scoring objects according the scoring algorithm that determines which objects to send back to a user as search results. As an example and not by way of limitation, social-networking system 160 may access a particular vertical 164 in response to a query and retrieves one-hundred matching objects, where each object has an associated static-rank. A final-score may then be calculated for these one-hundred objects (e.g., based on social-graph affinity) by a scoring algorithm. The top-5 scoring objects according to the scoring-algorithm may be, for example, objects having a static-

rank of 4, 12, 20, 78, and 95. This process could be improved, for example, if the top-N objects static-rank were the same as the top-N objects by final-rank. By more closely aligning the static-rank of object with the final-ranks calculated by the search engine, social-networking system may be able to reduce the number of matching objects it needs to retrieve and score in order to generate a sufficient number of search results. In particular embodiments, the static-score of an object may be based on the search query itself. In other words, depending on the particular query or query-type, an object may have a different static-score with respect to that query or query-type. As an example and not by way of limitation, if the number to score is 500, the top 500 objects may be identified. These 500 objects may then be scores based on one or more factors (e.g., match to the search query or other query constraints, social-graph affinity, search history, etc.), and the top M results may then be generated as search results to display to the querying user. In particular embodiments, the top results after one or more rounds of rankings may be sent to an aggregator 320 for a final round of ranking, where identified objects may be reordered, redundant results may be dropped, or any other type of results-processing may occur before presentation to the querying user. Although this disclosure describes identifying particular numbers of objects, this disclosure contemplates identifying any suitable numbers of objects. Furthermore, although this disclosure describes ranking objects in a particular manner, this disclosure contemplates ranking objects in any suitable manner.

[58] In particular embodiments, social-networking system 160 may score one or more objects identified as matching a query constraint. The score (also referred to as a final-score) for each retrieved/identified object may be calculated in any suitable manner, such as, for example, by using a particular scoring algorithm. Each identified object may correspond to a particular user node 202 or concept node 204 of social graph 200. When a query command includes a plurality of query constraints, social-networking system 160 may score the nodes matching each query constraint independently or jointly. Social-networking system 160 may score the first set of identified nodes by accessing a data store 164 corresponding to the object-type of the identified nodes. As an example and not by way of limitation, when generating identified nodes matching the query constraint (extract authors: (term posts_liked_by: <Mark>)), social-networking system 160 may identify the set of users (<Tom>, <Dick>, <Harry>) in the user vertical 164. Social-networking system 160 may then score the users <Tom>, <Dick>, and

<Harry> based on their respective social-affinity with respect to the user <Mark>. For example, social-networking system 160 of the post vertical 164 may then score the identified nodes of users <Tom>, <Dick>, and <Harry> based on a number of posts in the list of posts liked by the user <Mark>. The users <Tom>, <Dick>, and <Harry> may have authored the following posts liked by the user <Mark>: <post 1>, <post 2>, <post 3>, <post 4>, <post 5>, <post 6>. If user <Dick> authored posts <post 1>, <post 2>, <post 3>, user <Tom> authored posts <post 5> and <post 6>, and user <Harry> authored post <post 4>, social-networking system 160 may score user <Dick> as highest since his authored most of the posts in the list of posts liked by the user <Mark>, with <Tom> and <Harry> having consecutively lower scores. As another example and not by way of limitation, using the prior example, social-networking system 160 may access a forward index that maps a post to a count of likes of the post. The index server may access the forward index and retrieve counts of likes for each post of the list of posts liked by the user <Mark>. The index server may score the posts in the list of posts (i.e., <post 1>, <post 2>, <post 3>, <post 4>, <post 5>, <post 6>) based on respective counts of likes, and return to social-networking system 160 authors of top scored posts (e.g., top 3 scored or most liked posts) as the first identified node. After each appropriate scoring factor is considered for a particular identified node, an overall score for the identified node may be determined. Based on the scoring of the nodes, social-networking system 160 may then generate one or more sets of identified nodes. As an example and not by way of limitation, social-networking system 160 may only generate a set of identified nodes corresponding to nodes having a score greater than a threshold score. As another example and not by way of limitation, social-networking system 160 may rank the scored nodes and then only generate a set of identified nodes corresponding to nodes having a rank greater than a threshold rank (e.g., top ten, top twenty, etc.). Although this disclosure describes scoring matching nodes in a particular manner, this disclosure contemplates scoring matching nodes in any suitable manner.

[59] In particular embodiments, social-networking system 160 may score the search results based on a social-graph affinity associated with the querying user (or the user node 202 of the querying user). The scoring algorithm used to score retrieved object may use social-graph affinity as a factor. Social-networking system 160 may determine the social-graph affinity (which may be referred to herein as "affinity") of various social-graph entities for each other.

Affinity may represent the strength of a relationship or level of interest between particular objects associated with the online social network, such as users, concepts, content, actions, advertisements, other objects associated with the online social network, or any suitable combination thereof. In particular embodiments, social-networking system 160 may measure or quantify social-graph affinity using an affinity coefficient (which may be referred to herein as "coefficient"). The coefficient may represent or quantify the strength of a relationship between particular objects associated with the online social network. The coefficient may also represent a probability or function that measures a predicted probability that a user will perform a particular action based on the user's interest in the action. In particular embodiments, social-graph affinity may be used as a factor when scoring search results. As an example and not by way of limitation, in response to the structured query "Photos of my friends", social-networking system 160 may generate the query command (photos_of(users:<friends>)), and may determine that the search intent of this query is to view group photos showing the user's friends. When scoring identified concept nodes 204 corresponding to photos with the user's friends tagged in the photo, social-networking system 160 may score photos better based on the querying user's respective social-graph affinity (e.g., as measured by a affinity coefficient) of the user's tagged in the photo with respect to the querying user. Furthermore, photos showing more of the querying user's friends may be tagged higher than photos showing fewer of the user's friends, since having more friends tagged in the photo may increase the querying user's affinity with respect to that particular photo. Although this disclosure describes scoring search results based on affinity in a particular manner, this disclosure contemplates scoring search results based on affinity in any suitable manner. Furthermore, in connection with social-graph affinity and affinity coefficients, particular embodiments may utilize one or more systems, components, elements, functions, methods, operations, or steps disclosed in U.S. Patent No. 8,402,094, filed 11 August 2006, U.S. Patent Publication No. US2012/0166433, filed 22 December 2010, U.S. Patent Publication No. US2012/0166532, filed 23 December 2010, and U.S. Patent Publication No. US2014/0095606, filed 01 October 2012.

[60] In particular embodiments, social-networking system 160 may determine one or more revised static-scores for one or more of the retrieved objects based on a comparison of the final-scores and the static-scores of the retrieved objects. The static-scores associated with

indexed object may be improved by revising the static-scores based on experiments run using archived search queries. The archived queries can be parsed to generate query commands, which can be submitted to a vertical 164 in order to retrieve a first number of objects based on their static-scores. The retrieved objects can have their final-scores calculated. The final-scores can then be compared to the static-scores, and the static-scores can be modified so they more closely match the final-scores. This can be done for a variety of queries, so that the static-scores are optimized to match the final-scores as closely as possible for a variety of queries. In particular embodiments, social-networking system 160 may revise the static-scoring algorithm based on the revised static-scores. The static-scoring algorithm may be revised to calculate pre-determined static-scores for objects based on one or more of the revised static-scores of one or more of the retrieved objects, respectively. In particular embodiments, social-networking system 160 may revise static-scores by determining a difference between the pre-determined static-score for each object and the calculated final-score for each object. Social-networking system 160 may then revise one or more of the static-scores of one or more of the objects based on the determined differences. As an example and not by way of limitation, continuing with a previous example, if the top-5 objects by final-score according to the scoring-algorithm may be, for example, objects having a static-rank of 4, 12, 20, 78, and 95. The static-ranks of all the objects may be revised upward so that these objects have static-ranks closer to 1-to-5. Note that, theoretically the ideal static-ranks would be 1, 2, 3, 4, and 5. However, because the final-scores may be based on a variety of factors, such as social-graph affinity and user history, the ideal static-ranks with respect to a first querying user or a first query-type may be different than the ideal static-ranks with respect to a second querying user or a second query-type. Thus, the static-ranks of objects may be revised to they more closely match the final-ranks of objects with respect to a variety of users and query-types. Although this disclosure describes revising static-scoring algorithms in a particular manner, this disclosure contemplates revising static-scoring algorithms in any suitable manner.

[61] In particular embodiments, social-networking system 160 may generate one or more revised parsing-configuration parameters based on a comparison of the final-scores of the retrieved objects and the specified number of objects of the query constraints. The parsing algorithm may be improved by revising the way query constraints are generated based on

experiments run using archived search queries. The archived queries can be parsed to generate query commands, which can be submitted to one or more verticals 164 in order to retrieve a first number of objects. Social-networking system 160 may then calculate final-scores for the retrieved objects, and the final-scores may then be analyzed to determine whether the number of objects retrieved for any specified object-type can be reduced while still retrieving some or all of the top-N scoring results. In particular embodiments, social-networking system 160 may revise the parsing algorithm based on the parsing-configuration parameters such that one or more of the specified number of objects of a specified object-type is reduced based on the revised parsing-configuration parameters. As an example and not by way of limitation, for a particular query command s-expression generated by the parsing algorithm in response to a particular query, social-networking system 160 may revise the parsing-configuration parameters used to generate that query command so the specified number of objects specified by "num_to_score" is reduced while still retrieving some or all of the top-N scoring results (e.g., retrieving a sufficient number of the top-N scoring results to maintain a threshold quality of search results). If the num_to_score can be reduced, then the parsing algorithm (or particular parsing-configuration parameters) may be revised to retrieve fewer objects or object-types. The amount that num_to_score is reduced may correlated directly to processing power consumed by social-networking system 160. As these experiments are run using archived queries, social-networking system 160 may generate data of score-quality versus CPU power (or simply num_to_score), and use that data to find a point where, for particular queries or query-types, social-networking system 160 is still retrieving sufficient high-quality results (i.e., high-scoring results) while significantly reducing the power consumed. In other words, it may be worthwhile to sacrifice some search result quality if there is enough savings in processing power. In particular embodiments, social-networking system 160 may revise the parsing algorithm based on the parsing-configuration parameters such that one or more of the query constraints is removed from the query commands generated by the parsing algorithm based on the revised parsing-configuration parameters. In particular embodiments, social-networking system 160 may revise the parsing algorithm based on the parsing-configuration parameters such that one or more of the specified number of data stores 164 to access is reduced based on the revised parsing-configuration parameters. As an example and not by way of limitation, continuing with a prior

example, in response to a search query input “kais”, social-networking system 160 may generate the following query command:

(AND (name: “kais”)

(OR friends_of: (friends_of: <me>) : num_to_score: 50)

(OR pages: <> : num_to_score: 25)).

If an analysis of the final-scores of the retrieved pages from the pages vertical 164 shows that none of the retrieved pages are within the top-N results, then that entire query constraint may be removed. In other words, the parsing algorithm may be revised so that pages verticals 164 are not searched in response to this query-type. In particular embodiments, social-networking system 160 may revise the parsing algorithm based on the number of objects that need to be retrieved from the data store in order to retrieve all objects having a final-score greater than or equal to a threshold score. As an example and not by way of limitation, social-networking system may identify each retrieved object having a score (or rank) greater than or equal to a threshold score. Social-networking system 160 may then determine, for each query constraint of each query command, a number of objects that need to be retrieved from the data store to retrieve each identified object having a score greater than or equal to the threshold score. Based on the determined number of objects that need to be retrieved from the data store, social-networking system 160 may then revise one or more of the parsing-configuration parameters. Although this disclosure describes revising parsing algorithms in a particular manner, this disclosure contemplates revising parsing algorithms in any suitable manner.

[62] FIG. 6 illustrates an example method 600 for improving the static-scoring of objects for search queries. The method may begin at step 610, where social-networking system 160 may access a first set of queries of an online social network received from one or more users of the online social network. At step 620, social-networking system 160 may retrieve, for each query of the first set of queries, a first number of objects that match at least a portion of the query from one or more data stores 164. Each data store 164 may store one or more objects associated with the online social network. Furthermore, each object may be associated with a pre-determined static-score based on a static-scoring algorithm. Social-networking system 160 may retrieve the first number of objects based on the static-scores of the objects. At step 630, social-networking system 160 may calculate, for each query, a final-score for each retrieved object

based on a final-scoring algorithm. At step 640, social-networking system 160 may determine one or more revised static-scores for one or more of the retrieved objects based on a comparison of the final-scores and the static-scores of the retrieved objects. Particular embodiments may repeat one or more steps of the method of FIG. 6, where appropriate. Although this disclosure describes and illustrates particular steps of the method of FIG. 6 as occurring in a particular order, this disclosure contemplates any suitable steps of the method of FIG. 6 occurring in any suitable order. Moreover, although this disclosure describes and illustrates an example method for improving the static-scoring of objects for search queries including the particular steps of the method of FIG. 6, this disclosure contemplates any suitable method for improving the static-scoring of objects for search queries including any suitable steps, which may include all, some, or none of the steps of the method of FIG. 6, where appropriate. Furthermore, although this disclosure describes and illustrates particular components, devices, or systems carrying out particular steps of the method of FIG. 6, this disclosure contemplates any suitable combination of any suitable components, devices, or systems carrying out any suitable steps of the method of FIG. 6.

[63] **FIG. 7** illustrates an example method 700 for improving the parsing of search queries. The method may begin at step 710, where social-networking system 160 may access a first set of queries of an online social network received from one or more users of the online social network. At step 720, social-networking system 160 may parse each query in the first set of queries using a first parsing algorithm to generate a query command based on each query. Each query command may comprise one or more query constraints. Furthermore, each query constraint may be for a specified number of objects of a specified object-type as specified by one or more parsing-configuration parameters of the first parsing algorithm. At step 730, social-networking system 160 may retrieve, for each query constraint of each query command, the specified number of objects that match at least a portion of the query constraint from one or more data stores 164. Each data store 164 may store one or more objects associated with the online social network of the specified object-type for the query constraint. At step 740, social-networking system 160 may score, for each query command, each retrieved object based on a first scoring algorithm. At step 750, social-networking system 160 may generate one or more revised parsing-configuration parameters based on a comparison of the scores of the retrieved

objects and the specified number of objects of the query constraints. Particular embodiments may repeat one or more steps of the method of FIG. 7, where appropriate. Although this disclosure describes and illustrates particular steps of the method of FIG. 7 as occurring in a particular order, this disclosure contemplates any suitable steps of the method of FIG. 7 occurring in any suitable order. Moreover, although this disclosure describes and illustrates an example method for improving the parsing of search queries including the particular steps of the method of FIG. 7, this disclosure contemplates any suitable method for improving the parsing of search queries including any suitable steps, which may include all, some, or none of the steps of the method of FIG. 7, where appropriate. Furthermore, although this disclosure describes and illustrates particular components, devices, or systems carrying out particular steps of the method of FIG. 7, this disclosure contemplates any suitable combination of any suitable components, devices, or systems carrying out any suitable steps of the method of FIG. 7.

[64] In particular embodiments, in response to a query received from a querying user, social-networking system 160 may generate one or more search results, where the search results correspond to the query. Social-networking system 160 may identify objects (e.g., users, photos, profile pages (or content of profile pages), etc.) that satisfy or otherwise match the query. Each search result may correspond to a node of social graph 200. A search result corresponding to each identified object may then be generated. As an example and not by way of limitation, in response to the query "Photos of Matt and Stephanie", social-networking system 160 may identify a photo where the user's "Matt" and "Stephanie" are both tagged in the photo. A search result corresponding to this photo may then be generated and sent to the user. In particular embodiments, each search result may be associated with one or more objects, where each query constraint of the query command corresponding to the query is satisfied by one or more of the objects associated with that particular search result. As an example and not by way of limitation, continuing with the prior example, in response to the structured query "Photos of Matt and Stephanie", social-networking system 160 may parse the query to generate the query command (intersect(photos_of:<Matt>), (photos_of:<Stephanie>)), which could be executed to generate a search result corresponding to a photo where the user's "Matt" and "Stephanie" (who were both referenced in the structured query) are both tagged in the photo (i.e., their user nodes 202 are connected by tagged-in-type edges 206 to the concept node 204 corresponding to the photo). In

other words, the constraints for (photos_of:<Matt>) and (photos_of:<Stephanie>) are both satisfied by the photo because it is connected to the user nodes 202 for the user's "Matt" and "Stephanie". The nodes identified as matching the query may be scored (and possibly ranked), and then one or more (e.g., a threshold number) may be generated as search result to display to the user. Although this disclosure describes generating search results in a particular manner, this disclosure contemplates generating search results in any suitable manner.

[65] In particular embodiments, social-networking system 160 may send one or more search results to the querying user. The search results may be sent to the user, for example, in the form of a list of links on the search-results webpage, each link being associated with a different webpage that contains some of the identified resources or content. In particular embodiments, each link in the search results may be in the form of a Uniform Resource Locator (URL) that specifies where the corresponding webpage is located and the mechanism for retrieving it. Social-networking system 160 may then send the search-results webpage to the web browser 132 on the user's client system 130. The user may then click on the URL links or otherwise select the content from the search-results webpage to access the content from social-networking system 160 or from an external system (such as, for example, third-party system 170), as appropriate. In particular embodiments, each search result may include link to a profile page and a description or summary of the profile page (or the node corresponding to that page). The search results may be presented and sent to the querying user as a search-results page. When generating the search results, social-networking system 160 may generate one or more snippets for each search result, where the snippets are contextual information about the target of the search result (i.e., contextual information about the social-graph entity, profile page, or other content corresponding to the particular search result). In particular embodiments, social-networking system 160 may only send search results having a score/rank over a particular threshold score/rank. As an example and not by way of limitation, social-networking system 160 may only send the top ten results back to the querying user in response to a particular search query. Although this disclosure describes sending particular search results in a particular manner, this disclosure contemplates sending any suitable search results in any suitable manner.

[66] More information on generating search results may be found in U.S. Patent Application No. 13/731939, filed 31 December 2012.

[67] **FIG. 8** illustrates an example computer system 800. In particular embodiments, one or more computer systems 800 perform one or more steps of one or more methods described or illustrated herein. In particular embodiments, one or more computer systems 800 provide functionality described or illustrated herein. In particular embodiments, software running on one or more computer systems 800 performs one or more steps of one or more methods described or illustrated herein or provides functionality described or illustrated herein. Particular embodiments include one or more portions of one or more computer systems 800. Herein, reference to a computer system may encompass a computing device, and vice versa, where appropriate. Moreover, reference to a computer system may encompass one or more computer systems, where appropriate.

[68] This disclosure contemplates any suitable number of computer systems 800. This disclosure contemplates computer system 800 taking any suitable physical form. As example and not by way of limitation, computer system 800 may be an embedded computer system, a system-on-chip (SOC), a single-board computer system (SBC) (such as, for example, a computer-on-module (COM) or system-on-module (SOM)), a desktop computer system, a laptop or notebook computer system, an interactive kiosk, a mainframe, a mesh of computer systems, a mobile telephone, a personal digital assistant (PDA), a server, a tablet computer system, or a combination of two or more of these. Where appropriate, computer system 800 may include one or more computer systems 800; be unitary or distributed; span multiple locations; span multiple machines; span multiple data centers; or reside in a cloud, which may include one or more cloud components in one or more networks. Where appropriate, one or more computer systems 800 may perform without substantial spatial or temporal limitation one or more steps of one or more methods described or illustrated herein. As an example and not by way of limitation, one or more computer systems 800 may perform in real time or in batch mode one or more steps of one or more methods described or illustrated herein. One or more computer systems 800 may perform at different times or at different locations one or more steps of one or more methods described or illustrated herein, where appropriate.

[69] In particular embodiments, computer system 800 includes a processor 802, memory 804, storage 806, an input/output (I/O) interface 808, a communication interface 810, and a bus 812. Although this disclosure describes and illustrates a particular computer system

having a particular number of particular components in a particular arrangement, this disclosure contemplates any suitable computer system having any suitable number of any suitable components in any suitable arrangement.

[70] In particular embodiments, processor 802 includes hardware for executing instructions, such as those making up a computer program. As an example and not by way of limitation, to execute instructions, processor 802 may retrieve (or fetch) the instructions from an internal register, an internal cache, memory 804, or storage 806; decode and execute them; and then write one or more results to an internal register, an internal cache, memory 804, or storage 806. In particular embodiments, processor 802 may include one or more internal caches for data, instructions, or addresses. This disclosure contemplates processor 802 including any suitable number of any suitable internal caches, where appropriate. As an example and not by way of limitation, processor 802 may include one or more instruction caches, one or more data caches, and one or more translation lookaside buffers (TLBs). Instructions in the instruction caches may be copies of instructions in memory 804 or storage 806, and the instruction caches may speed up retrieval of those instructions by processor 802. Data in the data caches may be copies of data in memory 804 or storage 806 for instructions executing at processor 802 to operate on; the results of previous instructions executed at processor 802 for access by subsequent instructions executing at processor 802 or for writing to memory 804 or storage 806; or other suitable data. The data caches may speed up read or write operations by processor 802. The TLBs may speed up virtual-address translation for processor 802. In particular embodiments, processor 802 may include one or more internal registers for data, instructions, or addresses. This disclosure contemplates processor 802 including any suitable number of any suitable internal registers, where appropriate. Where appropriate, processor 802 may include one or more arithmetic logic units (ALUs); be a multi-core processor; or include one or more processors 802. Although this disclosure describes and illustrates a particular processor, this disclosure contemplates any suitable processor.

[71] In particular embodiments, memory 804 includes main memory for storing instructions for processor 802 to execute or data for processor 802 to operate on. As an example and not by way of limitation, computer system 800 may load instructions from storage 806 or another source (such as, for example, another computer system 800) to memory 804. Processor

802 may then load the instructions from memory 804 to an internal register or internal cache. To execute the instructions, processor 802 may retrieve the instructions from the internal register or internal cache and decode them. During or after execution of the instructions, processor 802 may write one or more results (which may be intermediate or final results) to the internal register or internal cache. Processor 802 may then write one or more of those results to memory 804. In particular embodiments, processor 802 executes only instructions in one or more internal registers or internal caches or in memory 804 (as opposed to storage 806 or elsewhere) and operates only on data in one or more internal registers or internal caches or in memory 804 (as opposed to storage 806 or elsewhere). One or more memory buses (which may each include an address bus and a data bus) may couple processor 802 to memory 804. Bus 812 may include one or more memory buses, as described below. In particular embodiments, one or more memory management units (MMUs) reside between processor 802 and memory 804 and facilitate accesses to memory 804 requested by processor 802. In particular embodiments, memory 804 includes random access memory (RAM). This RAM may be volatile memory, where appropriate Where appropriate, this RAM may be dynamic RAM (DRAM) or static RAM (SRAM). Moreover, where appropriate, this RAM may be single-ported or multi-ported RAM. This disclosure contemplates any suitable RAM. Memory 804 may include one or more memories 804, where appropriate. Although this disclosure describes and illustrates particular memory, this disclosure contemplates any suitable memory.

[72] In particular embodiments, storage 806 includes mass storage for data or instructions. As an example and not by way of limitation, storage 806 may include a hard disk drive (HDD), a floppy disk drive, flash memory, an optical disc, a magneto-optical disc, magnetic tape, or a Universal Serial Bus (USB) drive or a combination of two or more of these. Storage 806 may include removable or non-removable (or fixed) media, where appropriate. Storage 806 may be internal or external to computer system 800, where appropriate. In particular embodiments, storage 806 is non-volatile, solid-state memory. In particular embodiments, storage 806 includes read-only memory (ROM). Where appropriate, this ROM may be mask-programmed ROM, programmable ROM (PROM), erasable PROM (EPROM), electrically erasable PROM (EEPROM), electrically alterable ROM (EAROM), or flash memory or a combination of two or more of these. This disclosure contemplates mass storage 806 taking any

suitable physical form. Storage 806 may include one or more storage control units facilitating communication between processor 802 and storage 806, where appropriate. Where appropriate, storage 806 may include one or more storages 806. Although this disclosure describes and illustrates particular storage, this disclosure contemplates any suitable storage.

[73] In particular embodiments, I/O interface 808 includes hardware, software, or both, providing one or more interfaces for communication between computer system 800 and one or more I/O devices. Computer system 800 may include one or more of these I/O devices, where appropriate. One or more of these I/O devices may enable communication between a person and computer system 800. As an example and not by way of limitation, an I/O device may include a keyboard, keypad, microphone, monitor, mouse, printer, scanner, speaker, still camera, stylus, tablet, touch screen, trackball, video camera, another suitable I/O device or a combination of two or more of these. An I/O device may include one or more sensors. This disclosure contemplates any suitable I/O devices and any suitable I/O interfaces 808 for them. Where appropriate, I/O interface 808 may include one or more device or software drivers enabling processor 802 to drive one or more of these I/O devices. I/O interface 808 may include one or more I/O interfaces 808, where appropriate. Although this disclosure describes and illustrates a particular I/O interface, this disclosure contemplates any suitable I/O interface.

[74] In particular embodiments, communication interface 810 includes hardware, software, or both providing one or more interfaces for communication (such as, for example, packet-based communication) between computer system 800 and one or more other computer systems 800 or one or more networks. As an example and not by way of limitation, communication interface 810 may include a network interface controller (NIC) or network adapter for communicating with an Ethernet or other wire-based network or a wireless NIC (WNIC) or wireless adapter for communicating with a wireless network, such as a WI-FI network. This disclosure contemplates any suitable network and any suitable communication interface 810 for it. As an example and not by way of limitation, computer system 800 may communicate with an ad hoc network, a personal area network (PAN), a local area network (LAN), a wide area network (WAN), a metropolitan area network (MAN), or one or more portions of the Internet or a combination of two or more of these. One or more portions of one or more of these networks may be wired or wireless. As an example, computer system 800 may

communicate with a wireless PAN (WPAN) (such as, for example, a BLUETOOTH WPAN), a WI-FI network, a WI-MAX network, a cellular telephone network (such as, for example, a Global System for Mobile Communications (GSM) network), or other suitable wireless network or a combination of two or more of these. Computer system 800 may include any suitable communication interface 810 for any of these networks, where appropriate. Communication interface 810 may include one or more communication interfaces 810, where appropriate. Although this disclosure describes and illustrates a particular communication interface, this disclosure contemplates any suitable communication interface.

[75] In particular embodiments, bus 812 includes hardware, software, or both coupling components of computer system 800 to each other. As an example and not by way of limitation, bus 812 may include an Accelerated Graphics Port (AGP) or other graphics bus, an Enhanced Industry Standard Architecture (EISA) bus, a front-side bus (FSB), a HYPERTRANSPORT (HT) interconnect, an Industry Standard Architecture (ISA) bus, an INFINIBAND interconnect, a low-pin-count (LPC) bus, a memory bus, a Micro Channel Architecture (MCA) bus, a Peripheral Component Interconnect (PCI) bus, a PCI-Express (PCIe) bus, a serial advanced technology attachment (SATA) bus, a Video Electronics Standards Association local (VLB) bus, or another suitable bus or a combination of two or more of these. Bus 812 may include one or more buses 812, where appropriate. Although this disclosure describes and illustrates a particular bus, this disclosure contemplates any suitable bus or interconnect.

[76] Herein, a computer-readable non-transitory storage medium or media may include one or more semiconductor-based or other integrated circuits (ICs) (such, as for example, field-programmable gate arrays (FPGAs) or application-specific ICs (ASICs)), hard disk drives (HDDs), hybrid hard drives (HHDs), optical discs, optical disc drives (ODDs), magneto-optical discs, magneto-optical drives, floppy diskettes, floppy disk drives (FDDs), magnetic tapes, solid-state drives (SSDs), RAM-drives, SECURE DIGITAL cards or drives, any other suitable computer-readable non-transitory storage media, or any suitable combination of two or more of these, where appropriate. A computer-readable non-transitory storage medium may be volatile, non-volatile, or a combination of volatile and non-volatile, where appropriate.

[77] Herein, "or" is inclusive and not exclusive, unless expressly indicated otherwise or indicated otherwise by context. Therefore, herein, "A or B" means "A, B, or both," unless

expressly indicated otherwise or indicated otherwise by context. Moreover, "and" is both joint and several, unless expressly indicated otherwise or indicated otherwise by context. Therefore, herein, "A and B" means "A and B, jointly or severally," unless expressly indicated otherwise or indicated otherwise by context.

[78] The scope of this disclosure encompasses all changes, substitutions, variations, alterations, and modifications to the example embodiments described or illustrated herein that a person having ordinary skill in the art would comprehend. The scope of this disclosure is not limited to the example embodiments described or illustrated herein. Moreover, although this disclosure describes and illustrates respective embodiments herein as including particular components, elements, functions, operations, or steps, any of these embodiments may include any combination or permutation of any of the components, elements, functions, operations, or steps described or illustrated anywhere herein that a person having ordinary skill in the art would comprehend. Furthermore, reference in the appended claims to an apparatus or system or a component of an apparatus or system being adapted to, arranged to, capable of, configured to, enabled to, operable to, or operative to perform a particular function encompasses that apparatus, system, component, whether or not it or that particular function is activated, turned on, or unlocked, as long as that apparatus, system, or component is so adapted, arranged, capable, configured, enabled, operable, or operative.

CLAIMS:

1. A method comprising, by one or more computing devices: accessing a first set of archived queries of an online social network received from a plurality of users of the online social network, each query being of a particular type of query; parsing each query in the first set of queries using a first parsing algorithm to generate a query command based on each query, each query command comprising one or more query constraints, each query constraint being for a specified number of objects of a specified object-type as specified by one or more parsing-configuration parameters of the first parsing algorithm, wherein the parsing-configuration parameters specify, for each particular type of query, instructions for generating a query command based on the type of query comprising a specified number of query constraints for a specified number of objects of a specified object-type to be retrieved from a specified number of data stores for that type of query; retrieving, for each query constraint of each query command, the specified number of objects that match at least a portion of the query constraint from one or more data stores, each data store storing one or more objects associated with the online social network of the specified object-type for the query constraint; scoring, for each query command, each retrieved object based on a first scoring algorithm; generating one or more revised parsing-configuration parameters based on a comparison of the scores of the retrieved objects and the specified number of objects of the query constraints; and revising the first parsing algorithm based on the parsing-configuration parameters, wherein one or more of the specified number of objects of a specified object-type is reduced based on the revised parsing-configuration parameters.

2. The method of claim 1, further comprising: revising the first parsing algorithm based on the parsing-configuration parameters, wherein one or more of the query constraints is removed from the query commands generated by the first parsing algorithm based on the revised parsing-configuration parameters.

3. The method of claim 1, wherein the first parsing algorithm further specifies a number of data stores to access, the method further comprising: revising the first parsing algorithm

based on the parsing-configuration parameters, wherein one or more of the specified number of data stores to access is reduced based on the revised parsing-configuration parameters.

4. The method of claim 1, wherein scoring each retrieved object based on the first scoring algorithm comprises ranking all of the retrieved objects.

5. The method of claim 1, wherein generating the one or more revised parsing-configuration parameters comprises: identifying each retrieved object having a score greater than or equal to a threshold score; determining, for each query constraint of each query command, a number of objects that need to be retrieved from the data store to retrieve each identified object having a score greater than or equal to the threshold score; and revising one or more of the parsing-configuration parameters based on the determined number of objects that need to be retrieved from the data store.

6. The method of claim 1, further comprising: accessing a social graph comprising a plurality of nodes and a plurality of edges connecting the nodes, each of the edges between two of the nodes representing a single degree of separation between them, the nodes comprising: a plurality of user nodes corresponding to a plurality of users of the online social network, respectively; and a plurality of concept nodes corresponding to a plurality of concepts associated with the online social network, respectively; wherein each query in the first set of queries corresponds to a particular user node, and each retrieved object corresponds to a user node or concept node of the plurality of nodes.

7. The method of claim 6, wherein each query of the first set of queries is a structured query comprising references to one or more selected nodes from the plurality of nodes and one or more selected edges from the plurality of edges.

8. The method of claim 6, wherein each query constraint is for one or more nodes of the plurality of nodes connected to a first of the selected nodes by a first of the selected edges.

9. The method of claim 1, wherein each query of the first set of queries is an unstructured text query comprising one or more n-grams.

10. The method of claim 1, wherein retrieving the specified number of objects that match at least a portion of the query constraint from one or more data stores comprises, for each query constraint: accessing one or more data stores storing objects of the specified object-type of the query constraint; and identifying one or more objects from the accessed data stores that match at least a portion of the query constraint.

11. The method of claim 1, wherein the specified object-type is selected from a group consisting of: a user, a photo, a post, a webpage, an application, a location, or a user group.

12. The method of claim 1, wherein each data store is selected from a group consisting of: a users data store, a photos data store, a posts data store, a webpages data store, an applications data store, a locations data store, or a user-groups data store.

13. One or more computer-readable non-transitory storage media embodying software that is operable when executed to: access a first set of archived queries of an online social network received from a plurality of users of the online social network, each query being of a particular type of query; parse each query in the first set of queries using a first parsing algorithm to generate a query command based on each query, each query command comprising one or more query constraints, each query constraint being for a specified number of objects of a specified object-type as specified by one or more parsing-configuration parameters of the first parsing algorithm, wherein the parsing-configuration parameters specify, for each particular type of query, instructions for generating a query command based on the type of query comprising a specified number of query constraints for a specified number of objects of a specified object-type to be retrieved from a specified number of data stores for that type of query; retrieve, for each query constraint of each query command, the specified number of objects that match at least a portion of the query constraint from one or more data stores, each data store

storing one or more objects associated with the online social network of the specified object-type for the query constraint; score, for each query command, each retrieved object based on a first scoring algorithm; generate one or more revised parsing-configuration parameters based on a comparison of the scores of the retrieved objects and the specified number of objects of the query constraints; and revising the first parsing algorithm based on the parsing-configuration parameters, wherein one or more of the specified number of objects of a specified object-type is reduced based on the revised parsing-configuration parameters.

14. A system comprising: one or more processors; and a memory coupled to the processors comprising instructions executable by the processors, the processors operable when executing the instructions to: access a first set of archived queries of an online social network received from a plurality of users of the online social network, each query being of a particular type of query; parse each query in the first set of queries using a first parsing algorithm to generate a query command based on each query, each query command comprising one or more query constraints, each query constraint being for a specified number of objects of a specified object-type as specified by one or more parsing-configuration parameters of the first parsing algorithm, wherein the parsing-configuration parameters specify, for each particular type of query, instructions for generating a query command based on the type of query comprising a specified number of query constraints for a specified number of objects of a specified object-type to be retrieved from a specified number of data stores for that type of query; retrieve, for each query constraint of each query command, the specified number of objects that match at least a portion of the query constraint from one or more data stores, each data store storing one or more objects associated with the online social network of the specified object-type for the query constraint; score, for each query command, each retrieved object based on a first scoring algorithm; generate one or more revised parsing-configuration parameters based on a comparison of the scores of the retrieved objects and the specified number of objects of the query constraints; and revising the first parsing algorithm based on the parsing-configuration parameters, wherein one or more of the

specified number of objects of a specified object-type is reduced based on the revised parsing-configuration parameters.

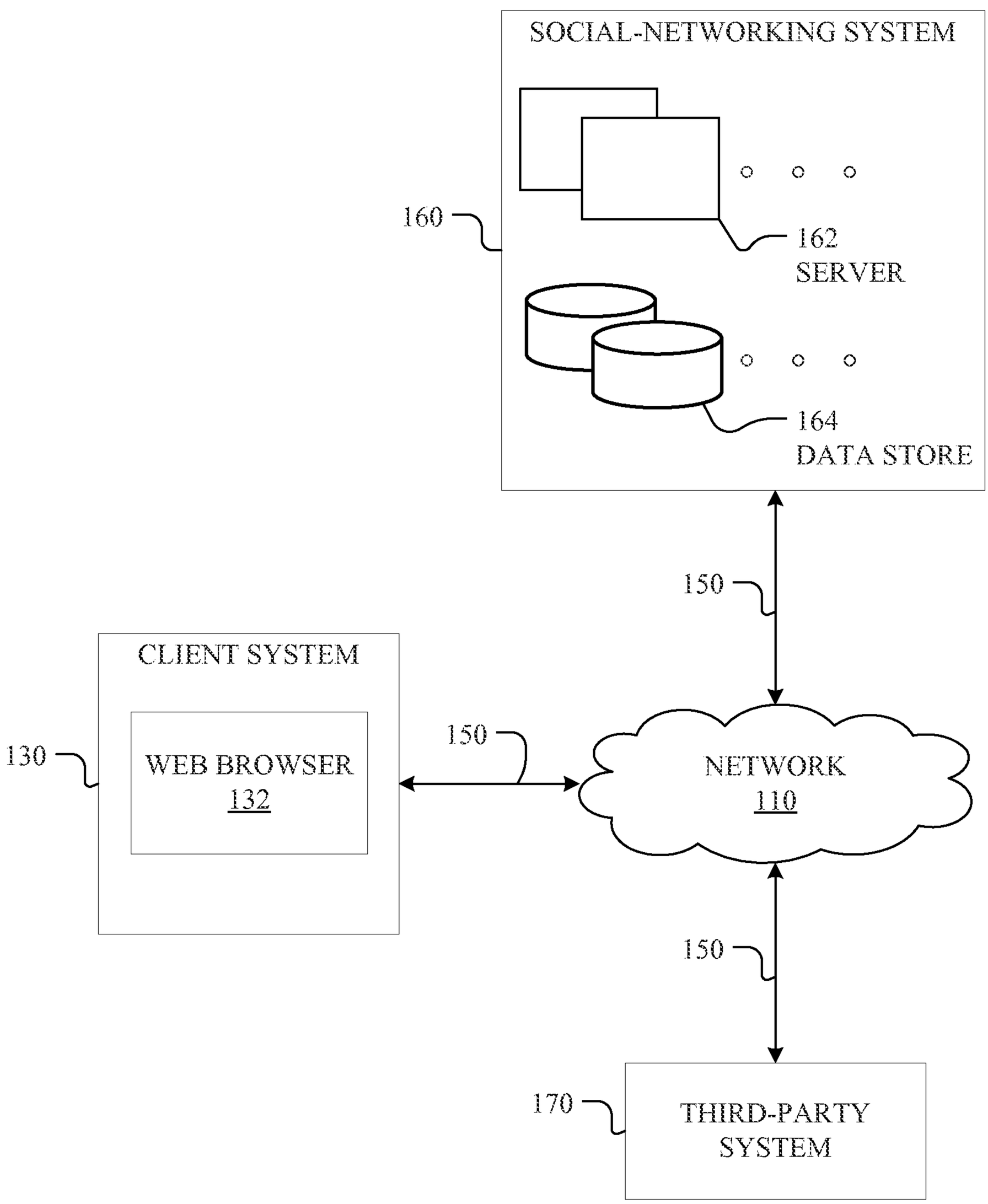

100
SOCIAL-NETWORKING SYSTEM
160
162
SERVER
164
DATA STORE
150
CLIENT SYSTEM
130
WEB BROWSER
132
150
NETWORK
110
150
170
THIRD-PARTY
SYSTEM

FIG. 1

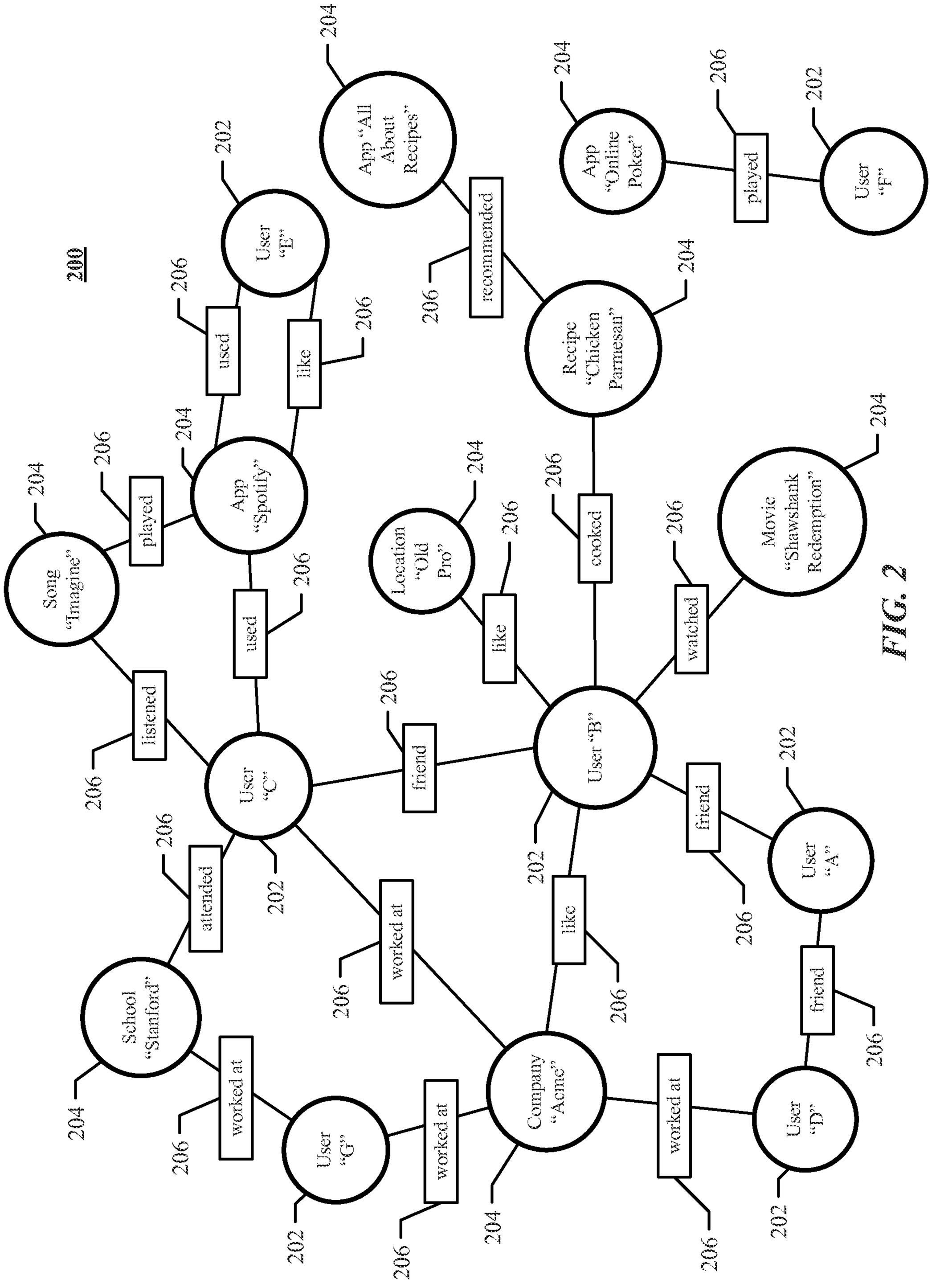

200
Song "Imagine"
204
played
206
App "Spotify"
204
used
206
User "E"
202
used
206
like
206
listened
206
User "C"
202
attended
206
School "Stanford"
204
worked at
206
worked at
206
friend
206
Location "Old Pro"
204
like
206
User "B"
202
like
206
worked at
206
Company "Acme"
204
worked at
206
User "G"
202
cooked
206
Recipe "Chicken Parmesan"
204
recommended
206
App "All About Recipes"
204
played
206
App "Online Poker"
204
played
206
User "F"
202
watched
206
Movie "Shawshank Redemption"
204
friend
206
User "A"
202
friend
206
User "D"
202

FIG. 2

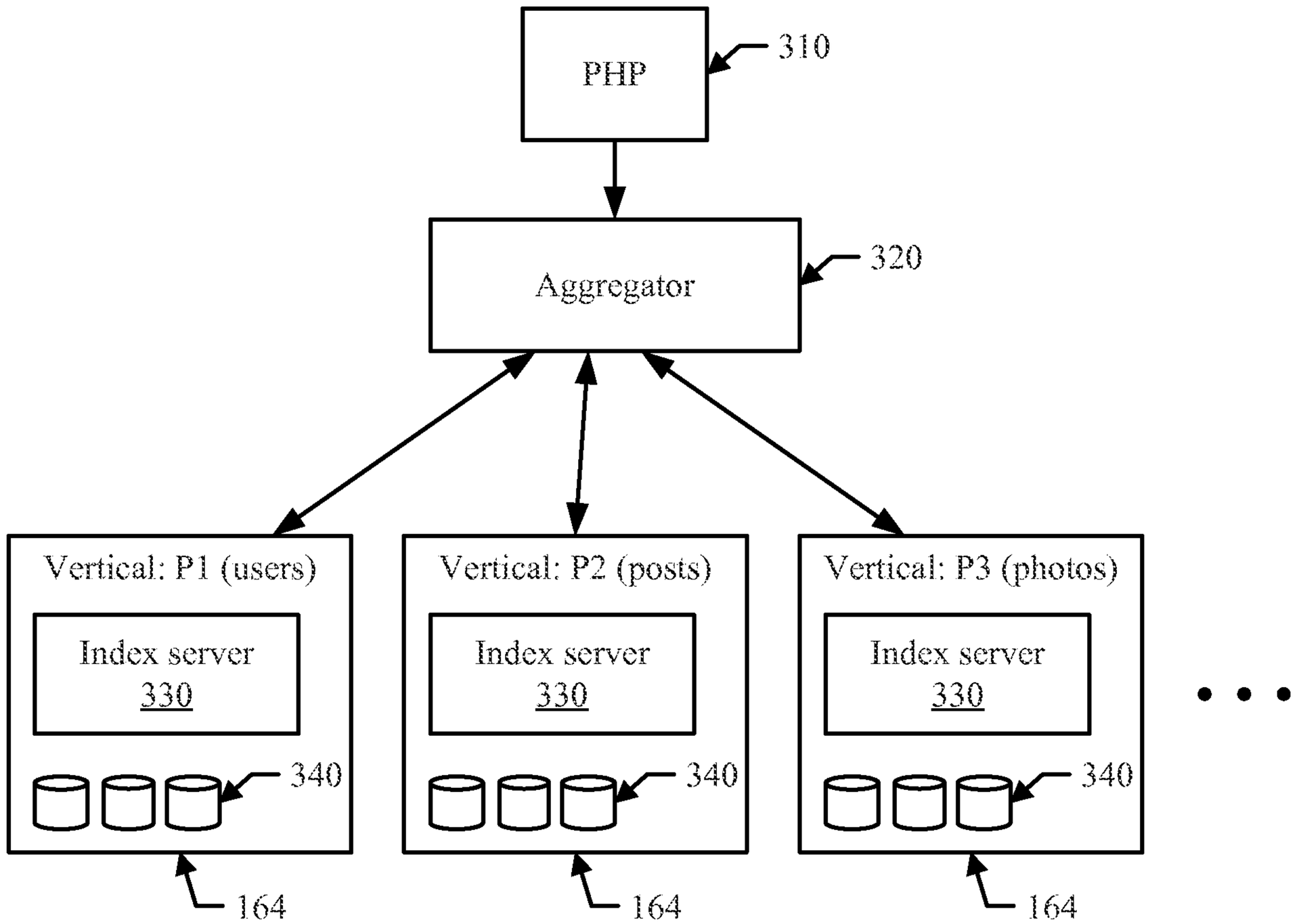

PHP
310
Aggregator
320
Vertical: P1 (users)
Index server
330
340
164
Vertical: P2 (posts)
Index server
330
340
164
Vertical: P3 (photos)
Index server
330
340
164

FIG. 3

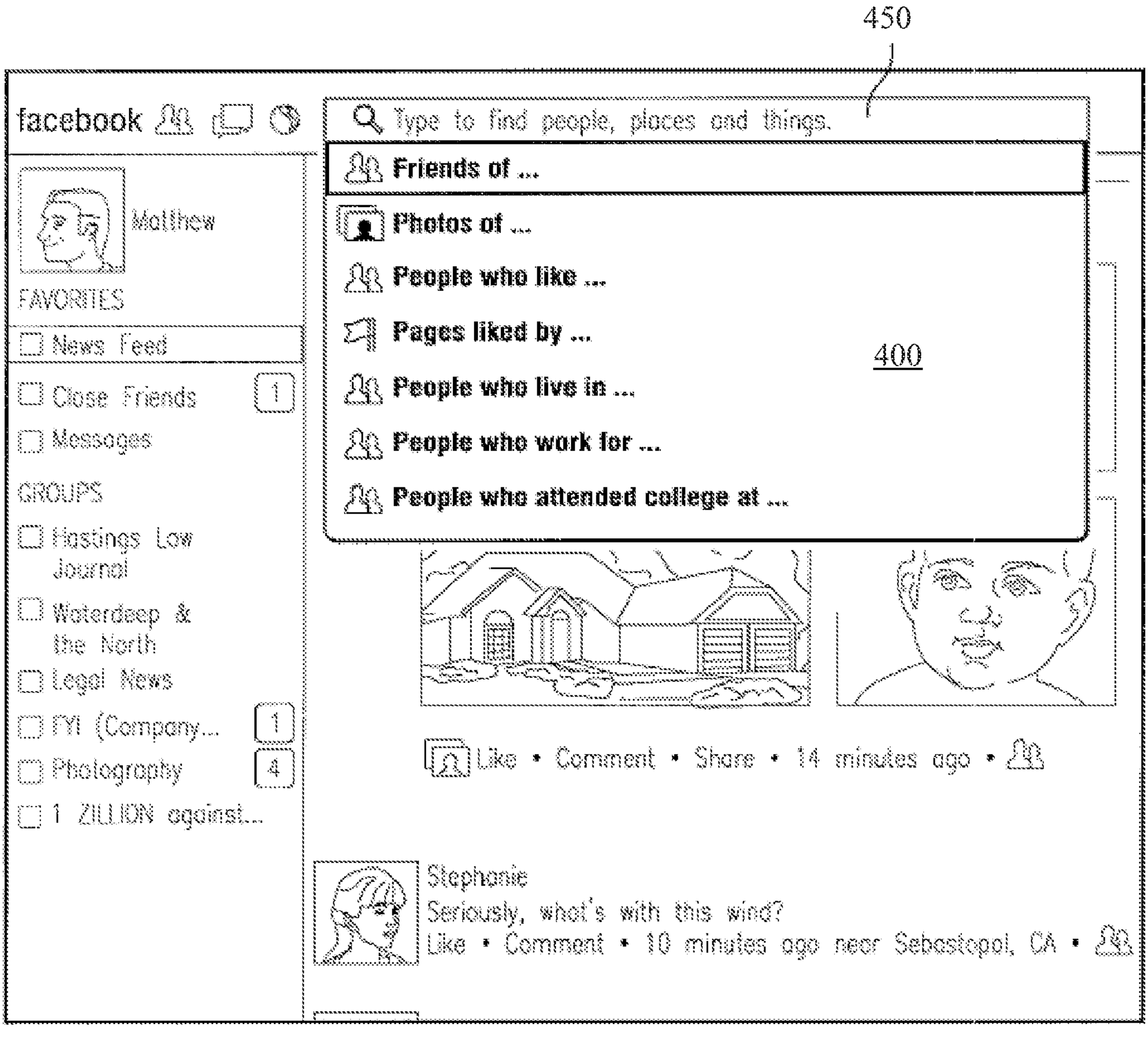

450
facebook
Type to find people, places and things.
Friends of ...
Photos of ...
People who like ...
Pages liked by ...
People who lives in ...
People who work for ...
People who attended college at ...
400
Matthew
FAVORITES
News Feed
Close Friends
Messages
GROUPS
Hastings Law Journal
Waterdeep & the North
Legal News
FYI (Company...
Photography
1 ZILLION against...
Like • Comment • Share • 14 minutes ago •
Stephanie
Seriously, what's with this wind?
Like • Comment • 10 minutes ago near Sebastopol, CA •

FIG. 4

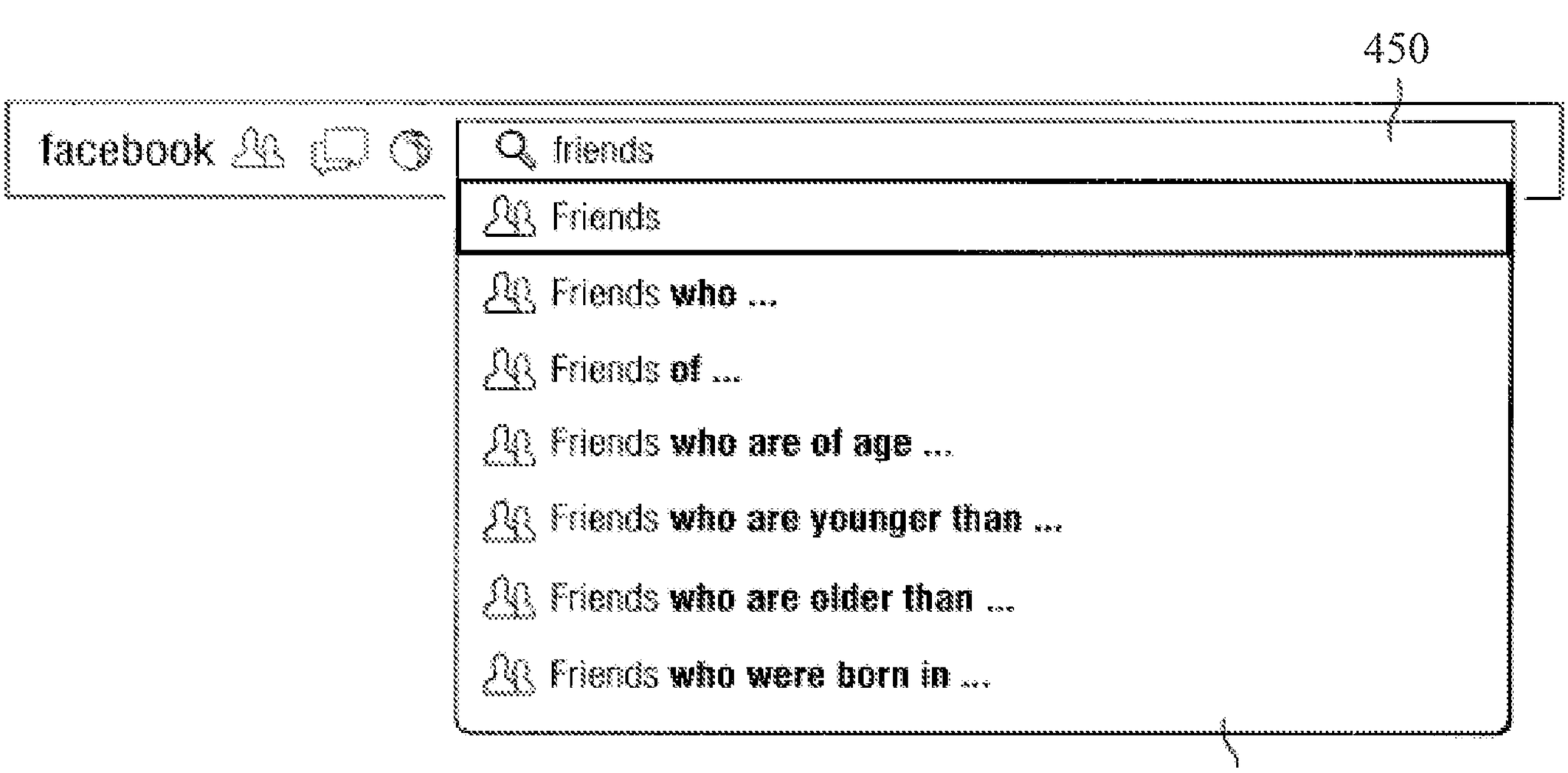

450
facebook
friends
Friends
Friends who ...
Friends of ...
Friends who are of age ...
Friends who are younger than ...
Friends who are older than ...
Friends who were born in ...
400

FIG. 5A

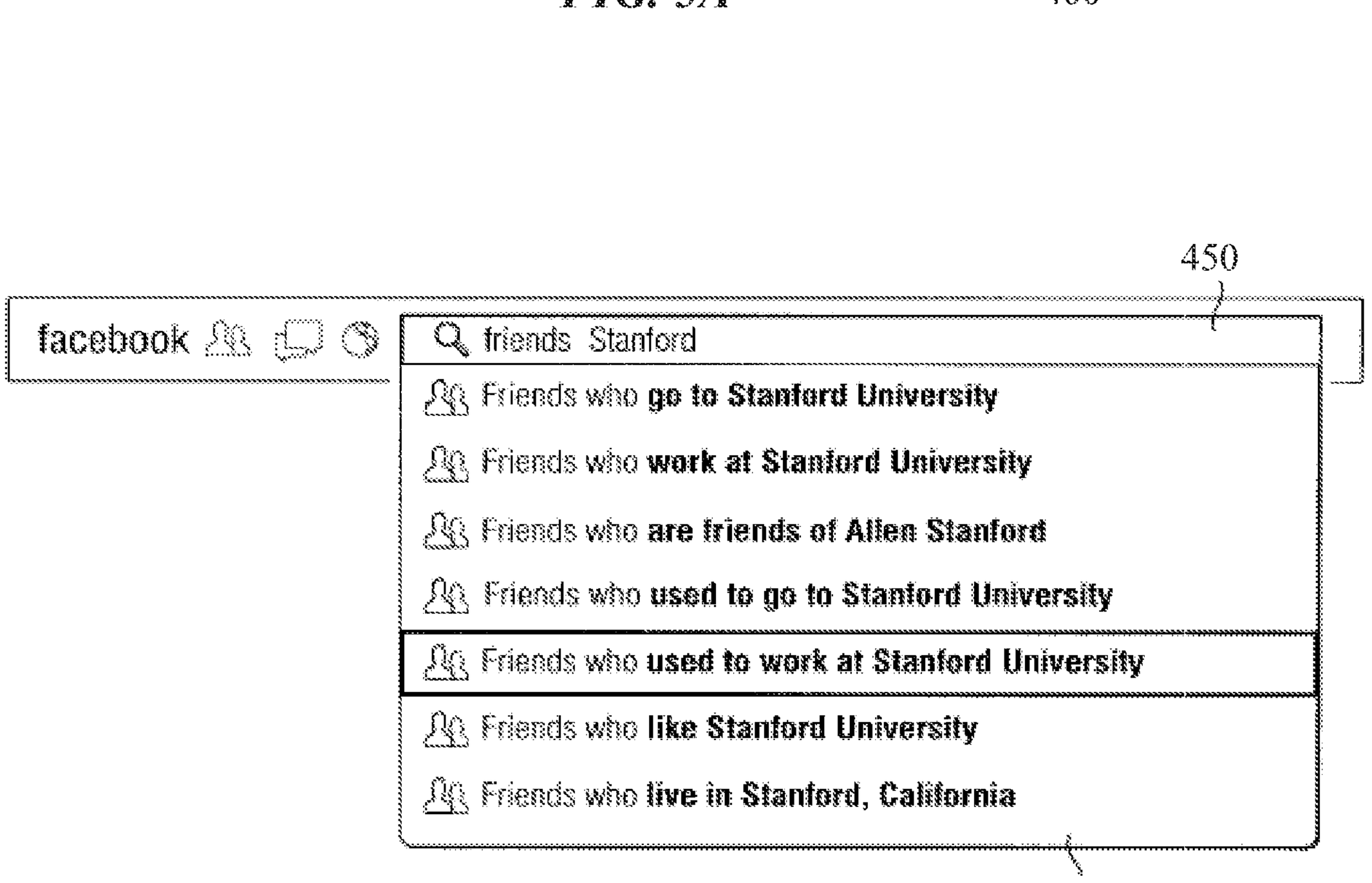

450
facebook
friends Stanford
Friends who go to Stanford University
Friends who work at Stanford University
Friends who are friends of Allen Stanford
Friends who used to go to Stanford University
Friends who used to work at Stanford University
Friends who like Stanford University
Friends who live in Stanford, California
400

FIG. 5B

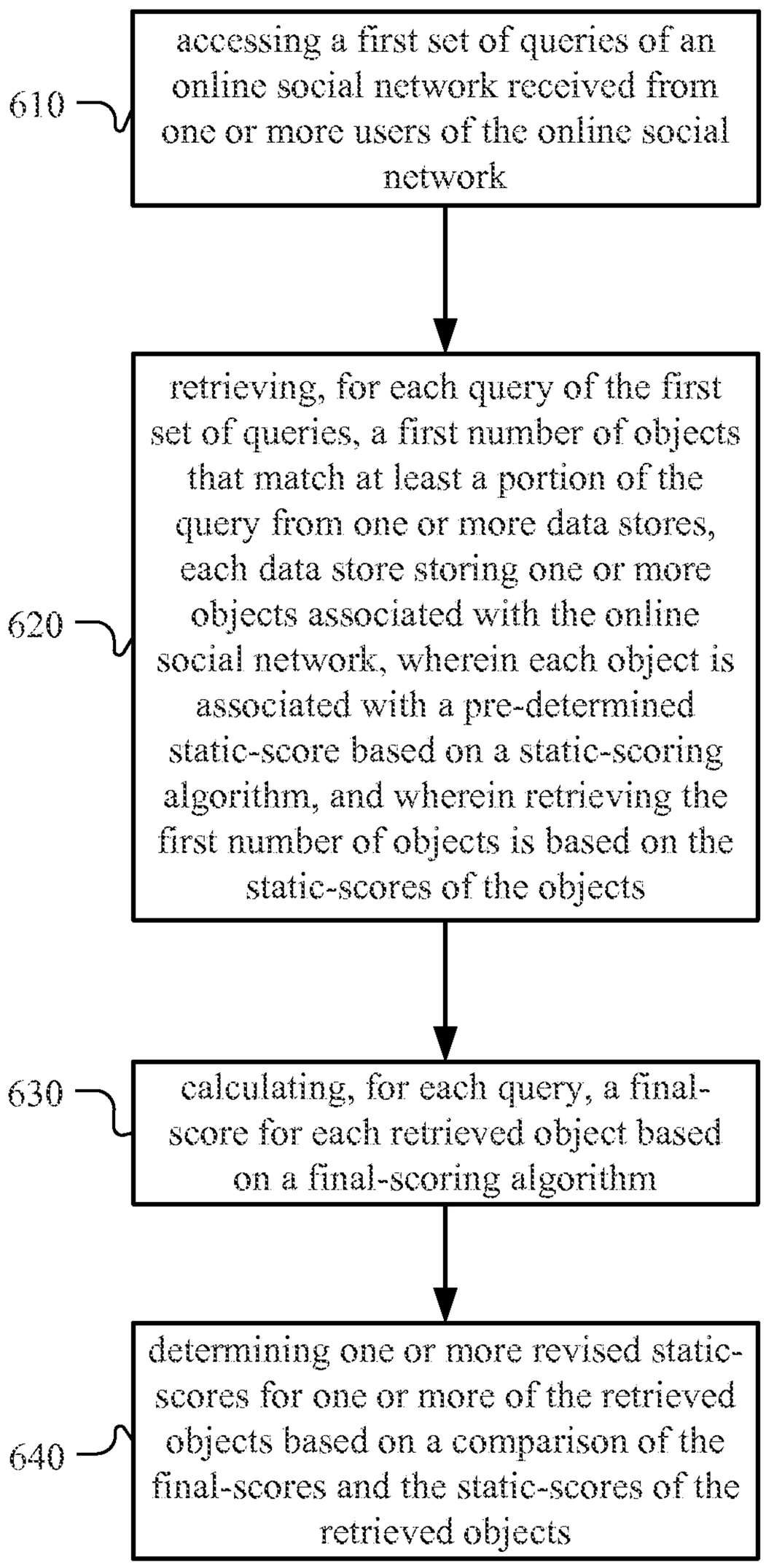

600
610
accessing a first set of queries of an online social network received from one or more users of the online social network
620
retrieving, for each query of the first set of queries, a first number of objects that match at least a portion of the query from one or more data stores, each data store storing one or more objects associated with the online social network, wherein each object is associated with a pre-determined static-score based on a static-scoring algorithm, and wherein retrieving the first number of objects is based on the static-scores of the objects
630
calculating, for each query, a final-score for each retrieved object based on a final-scoring algorithm
640
determining one or more revised static-scores for one or more of the retrieved objects based on a comparison of the final-scores and the static-scores of the retrieved objects

FIG. 6

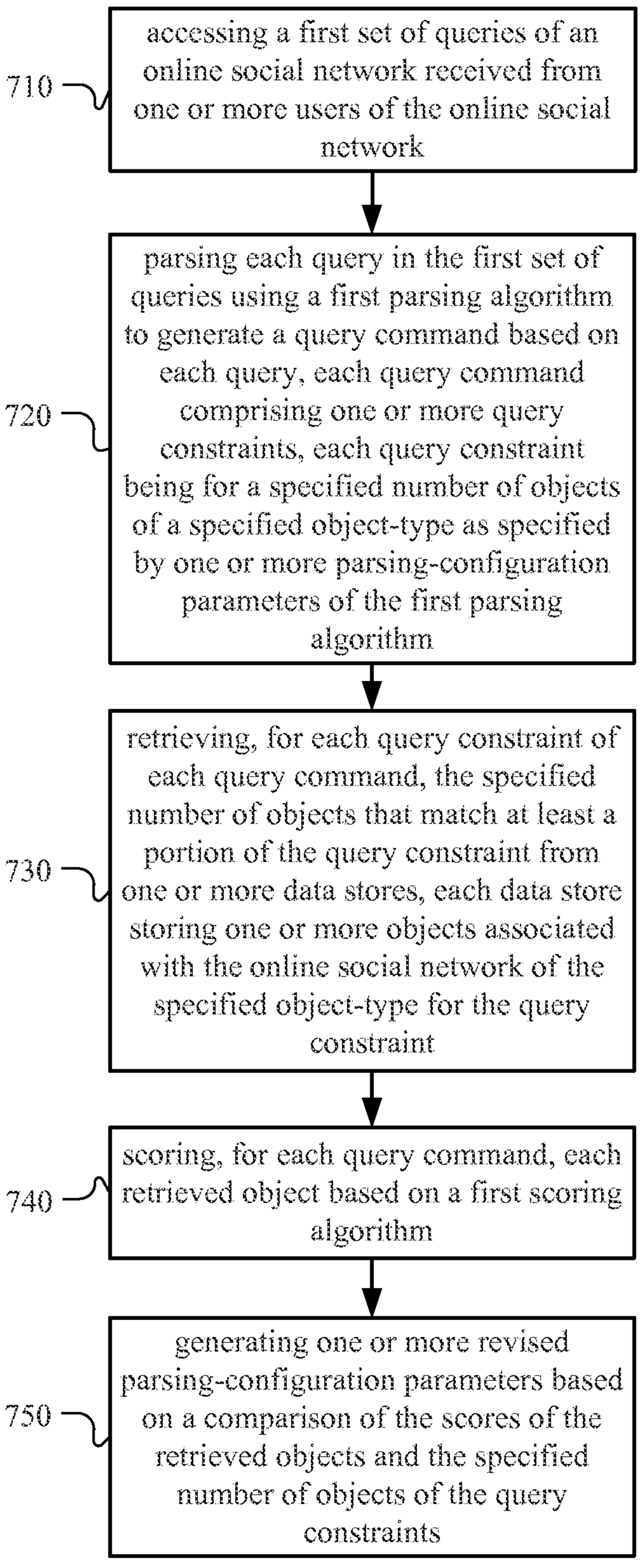

700
710
accessing a first set of queries of an online social network received from one or more users of the online social network
720
parsing each query in the first set of queries using a first parsing algorithm to generate a query command based on each query, each query command comprising one or more query constraints, each query constraint being for a specified number of objects of a specified object-type as specified by one or more parsing-configuration parameters of the first parsing algorithm
730
retrieving, for each query constraint of each query command, the specified number of objects that match at least a portion of the query constraint from one or more data stores, each data store storing one or more objects associated with the online social network of the specified object-type for the query constraint
740
scoring, for each query command, each retrieved object based on a first scoring algorithm
750
generating one or more revised parsing-configuration parameters based on a comparison of the scores of the retrieved objects and the specified number of objects of the query constraints

FIG. 7

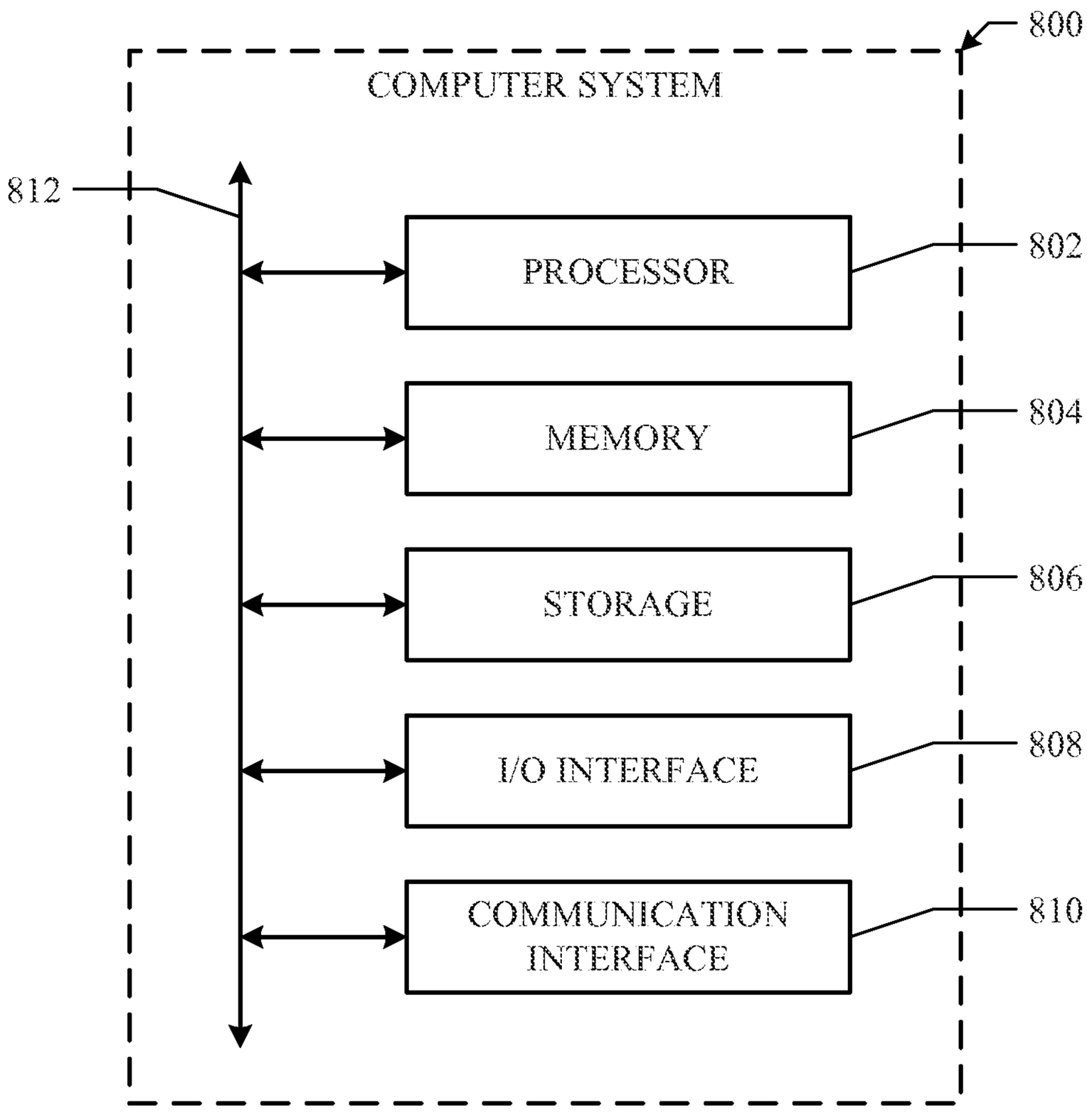

800
COMPUTER SYSTEM
812
PROCESSOR
802
MEMORY
804
STORAGE
806
I/O INTERFACE
808
COMMUNICATION INTERFACE
810

FIG. 8

700

710 accessing a first set of queries of an online social network received from one or more users of the online social network

720 parsing each query in the first set of queries using a first parsing algorithm to generate a query command based on each query, each query command comprising one or more query constraints, each query constraint being for a specified number of objects of a specified object-type as specified by one or more parsing-configuration parameters of the first parsing algorithm

730 retrieving, for each query constraint of each query command, the specified number of objects that match at least a portion of the query constraint from one or more data stores, each data store storing one or more objects associated with the online social network of the specified object-type for the query constraint

740 scoring, for each query command, each retrieved object based on a first scoring algorithm

750 generating one or more revised parsing-configuration parameters based on a comparison of the scores of the retrieved objects and the specified number of objects of the query constraints