



US 20140214830A1

(19) **United States**
(12) **Patent Application Publication**
Fischer

(10) **Pub. No.: US 2014/0214830 A1**
(43) **Pub. Date: Jul. 31, 2014**

(54) **METRICS-BASED ACCESSING OF SOCIAL THREADS**

Publication Classification

(71) Applicant: **salesforce.com, inc.**, San Francisco, CA (US)

(51) **Int. Cl.**
G06F 17/30 (2006.01)

(72) Inventor: **Ronald F. Fischer**, San Francisco, CA (US)

(52) **U.S. Cl.**
CPC **G06F 17/30861** (2013.01)
USPC **707/736**

(73) Assignee: **salesforce.com, inc.**, San Francisco, CA (US)

(57) **ABSTRACT**

(21) Appl. No.: **14/242,367**

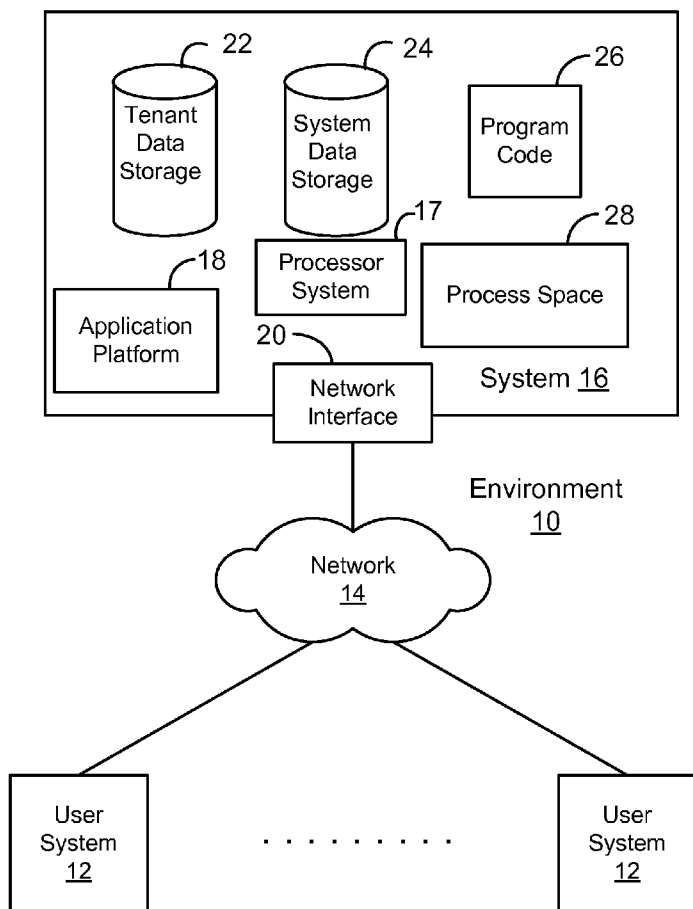
A method of accessing feeds based on metrics is provided. Feeds, each associated with an object stored in a database system, are provided to users of the database system. Inferential user interaction data captures implicit user behavior of users of the database system, wherein the data is generated in relation to a feed. Feed metrics are determined based on the user interaction data, wherein a feed metric is based upon statistics comprising user consumption, user responsiveness, content proliferation, and feed life. Finally, an action is executed in relation to at least one feed based on the feed metrics, wherein the action comprises discontinuing the feed, characterizing a feed, determining that a feed can be monetized, determining that a feed should be cached, or determining that intervention in a feed is advisable.

(22) Filed: **Apr. 1, 2014**

Related U.S. Application Data

(63) Continuation of application No. 13/111,183, filed on May 19, 2011, now Pat. No. 8,738,603.

(60) Provisional application No. 61/346,839, filed on May 20, 2010.



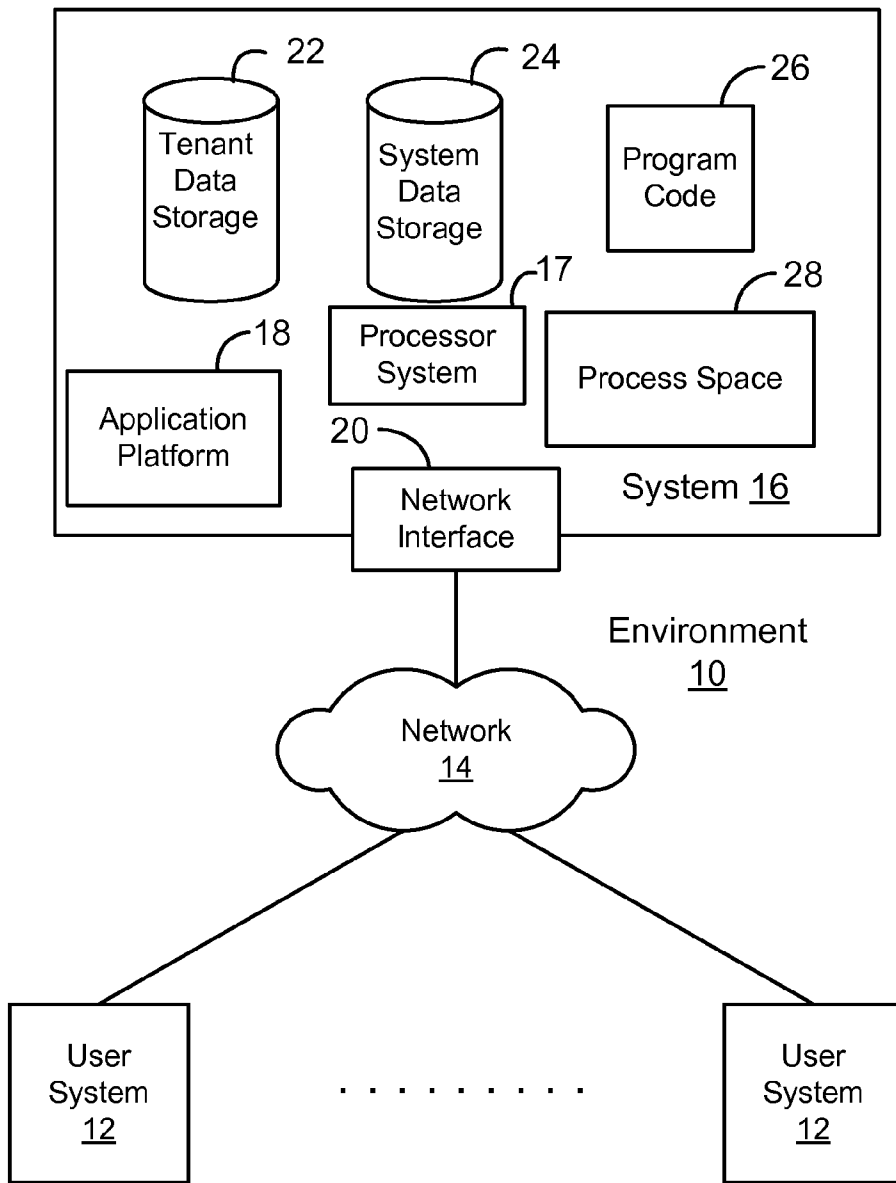


FIG. 1

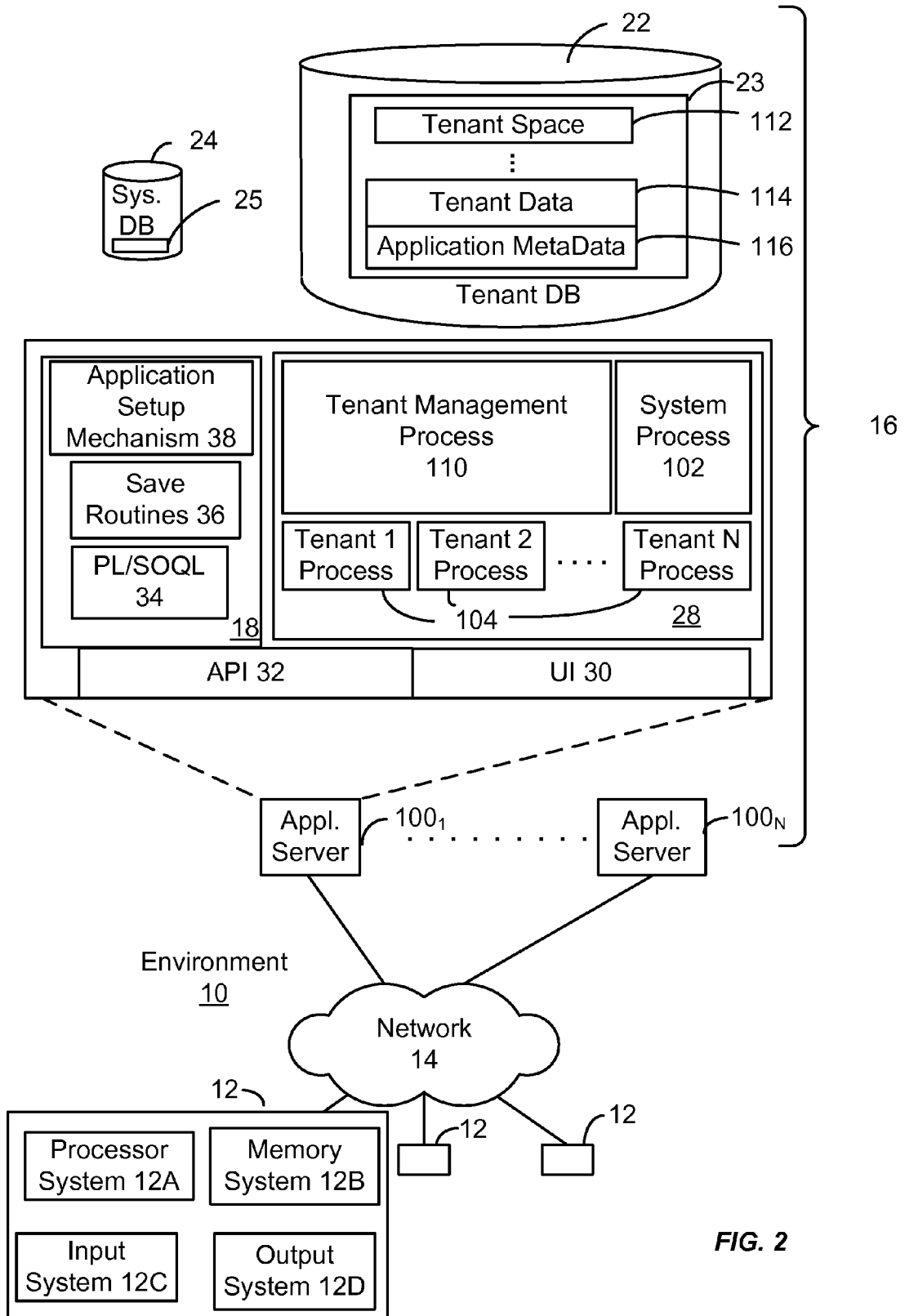


FIG. 2

300

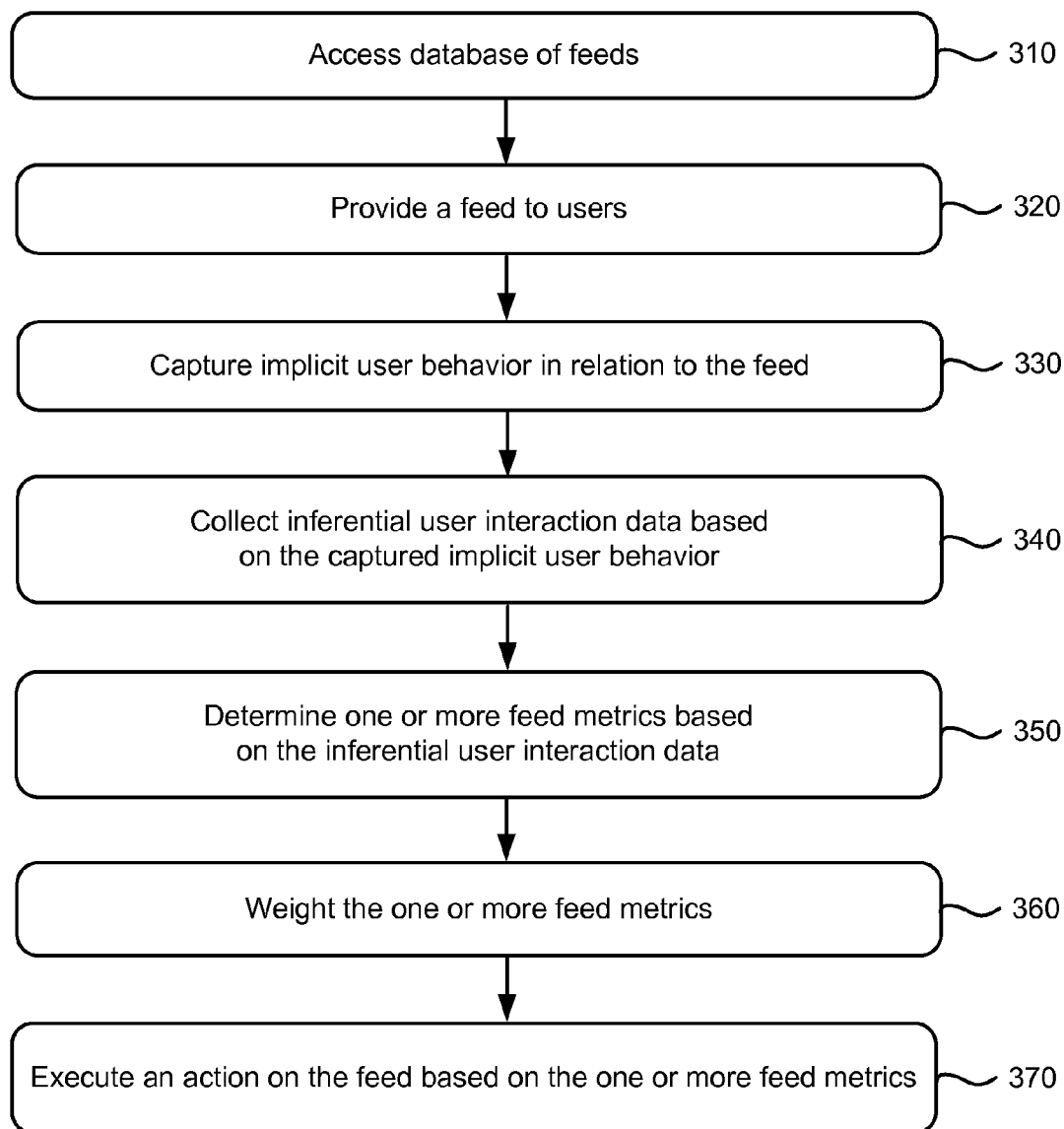


FIG. 3

METRICS-BASED ACCESSING OF SOCIAL THREADS

PRIORITY AND RELATED APPLICATION DATA

[0001] This application is a continuation to co-pending and commonly assigned U.S. patent application Ser. No. 13/111, 183 titled, "METRICS-BASED ACCESSING OF SOCIAL THREADS," by Ronald F. Fischer, filed May 19, 2011, which claims priority to U.S. Provisional Application No. 61/346, 839 titled "METRICS BASED ACCESSING OF SOCIAL THREADS AND RULE BASED PRIORITIZATION OF SOCIAL DATA," by Ronald F. Fischer, filed on May 20, 2010, both of which are hereby incorporated by reference in their entirety and for all purposes.

COPYRIGHT NOTICE

[0002] A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

BACKGROUND

[0003] The present invention relates generally to database systems, and more particularly to determining metrics for social data to allow for filtering and monitoring social data such as threads.

[0004] The subject matter discussed in the background section should not be assumed to be prior art merely as a result of its mention in the background section. Similarly, a problem mentioned in the background section or associated with the subject matter of the background section should not be assumed to have been previously recognized in the prior art. The subject matter in the background section merely represents different approaches, which in and of themselves may also be inventions.

BRIEF SUMMARY

[0005] The present invention provides systems, apparatus, and methods for social data analysis, and more particularly to determining metrics for social data to allow for filtering and monitoring social data such as threads.

[0006] A method of accessing feeds based on metrics is provided. A plurality of feeds are provided to a plurality of users of a database system, wherein each feed in the plurality of feeds is associated with an object stored in a database system. Inferential user interaction data is collected from the plurality of users, wherein the inferential user interaction data is generated in relation to at least one of the plurality of feeds, and wherein the inferential user interaction data captures implicit user behavior of at least one of the plurality of users of the database system. From the user interaction data, one or more feed metrics are determined, using one or more processors associated with the one or more servers. A feed metric may be based upon statistics comprising user consumption, user responsiveness, content proliferation, and feed life. The one or more feed metrics may be weighted or prioritized. Finally, an action is executed in relation to the at least one of the plurality of feeds based on the one or more weighted feed metrics, wherein the action comprises discontinuing the feed, characterizing a feed, determining that a feed can be mon-

itized, determining that a feed should be cached, or determining that intervention in a feed is advisable.

[0007] While the present invention is described with reference to an embodiment in which techniques for performing searches of feeds in an on-demand enterprise services environment are implemented in a system having an application server providing a front end for an on-demand database service capable of supporting multiple tenants, the present invention is not limited to multi-tenant databases nor deployment on application servers. Embodiments may be practiced using other database architectures, i.e., ORACLE®, DB2® by IBM and the like without departing from the scope of the embodiments claimed.

[0008] Any of the above embodiments may be used alone or together with one another in any combination. Inventions encompassed within this specification may also include embodiments that are only partially mentioned or alluded to or are not mentioned or alluded to at all in this brief summary or in the abstract. Although various embodiments of the invention may have been motivated by various deficiencies with the prior art, which may be discussed or alluded to in one or more places in the specification, the embodiments of the invention do not necessarily address any of these deficiencies. In other words, different embodiments of the invention may address different deficiencies that may be discussed in the specification. Some embodiments may only partially address some deficiencies or just one deficiency that may be discussed in the specification, and some embodiments may not address any of these deficiencies.

[0009] Reference to the remaining portions of the specification, including the drawings and claims, will realize other features and advantages of the present invention. Further features and advantages of the present invention, as well as the structure and operation of various embodiments of the present invention, are described in detail below with respect to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] In the following drawings like reference numbers are used to refer to like elements. Although the following figures depict various examples of the invention, the invention is not limited to the examples depicted in the figures.

[0011] FIG. 1 illustrates a block diagram of an environment wherein an on-demand database service might be used.

[0012] FIG. 2 illustrates a block diagram of an embodiment of elements of FIG. 1 and various possible interconnections between these elements according to an embodiment of the present invention.

[0013] FIG. 3 is a flowchart of a method 300 for determining metrics for social data.

DEFINITIONS

[0014] As used herein, the term "multi-tenant database system" refers to those systems in which various elements of hardware and software of the database system may be shared by one or more customers. For example, a given application server may simultaneously process requests for a great number of customers, and a given database table may store rows for a potentially much greater number of customers. As used herein, the term "query plan" refers to a set of steps used to access information in a database system.

[0015] As used herein, the term "user's profile" includes data about the user of the database system. The data can

include general information, such as title, phone number, a photo, a biographical summary, and a status (e.g., text describing what the user is currently doing). As mentioned below, the data can include messages created by other users. Where there are multiple tenants, a user is typically associated with a particular tenant. For example, a user could be a salesperson of a company that is a tenant of the database system that provides a database service.

[0016] As used herein, the term “record” refers to an instance of a data object created by a user of the database service, for example, about a particular (actual or potential) business relationship or project. The data object can have a data structure defined by the database service (a standard object) or defined by a subscriber (custom object). For example, a record can be for a business partner or potential business partner (e.g. a client, vendor, distributor, etc.) of the user, and can include an entire company, subsidiaries, or contacts at the company. As another example, a record can be a project that the user is working on, such as an opportunity (e.g. a possible sale) with an existing partner, or a project that the user is trying to get. In one embodiment implementing a multi-tenant database, all of the records for the tenants have an identifier stored in a common table. A record has data fields that are defined by the structure of the object (e.g. fields of certain data types and purposes). A record can also have custom fields defined by a user. A field can be another record or include links thereto, thereby providing a parent-child relationship between the records.

[0017] As used herein, the term “feed” includes a combination (e.g. a list) of feed items. As used herein, the term “feed item” (or feed element) refers to information associated with (posted by or addressed to) a user (“profile feed”) of the database or information about a record (“record feed”) in the database. A user following the user or record can receive the associated feed items. The feed items from all of the followed users and records can be combined into a single feed for the user. In some embodiments, a “feed item” may be (1) a message or (2) a story (also called a feed tracked change). A feed may be a combination of multiple types of feed items.

[0018] A “message” includes text created by a user, and may include other data as well. Examples of messages include posts, status updates, and comments. Messages can be created for a user’s profile or for a record. Posts can be created by various users, potentially any user, although some restrictions can be applied. As an example, posts can be made to a wall section of a user’s profile (which can include a number of recent posts) or a section of a record that includes multiple posts. The posts can be organized in chronological order. In contrast to a post, a status update changes a status of a user and is made by that user. Other similar sections of a user’s profile can also include an “About” section. A record can also have a status, whose update can be restricted to the owner of the record. The owner can be a single user, multiple users, or a group. In one embodiment, there is only one status for a record. In one embodiment, a comment can be made on any feed item. In another embodiment, comments are organized as a list explicitly tied to a particular story, post, or status update. In this embodiment, comments may not be listed in the first layer (in a hierarchal sense) of feed items, but listed as a second layer branching from a particular first layer feed item.

[0019] A “story” is data representing an event, and can include text generated by the database system in response to the event. In one embodiment, the data can initially be stored,

and then the database system can later use the data to create text for describing the event. Both the data and/or the text can be a story, as used herein. In various embodiments, an event can be an update of a record and/or can be triggered by a specific action by a user. Which actions trigger an event can be configurable. Which events have stories created and which stories are sent to which users can also be configurable. Messages and stories can be stored as a field or child object of the record. For example, the feed can be stored as a child object of the record.

[0020] As used herein, a “group” is a collection of users. In some aspects, the group may be defined as users with a same or similar attribute, or by membership. In one embodiment, a “group feed” includes any feed item about any user in a group. In another embodiment, a “group feed” includes feed items that are about the group as a whole. In one implementation, the feed items for a group are only posts and comments.

[0021] As used herein, an “entity feed” or “record feed” refers to a feed of feed items about a particular record in the database, such as stories about changes to the record and posts made by users about the record. An entity feed can be composed of any type of feed item. Such a feed can be displayed on a page (e.g. a web page) associated with the record (e.g. a home page of the record). As used herein, a “profile feed” is a feed of feed items about a particular user. In one embodiment, the feed items for a profile feed are posts and comments that other users make about or send to the particular user, and status updates made by the user. Such a profile feed can be displayed on a page associated with the particular user. In another embodiment, feed items in a profile feed could include posts made by the particular user and feed tracked changes (stories) initiated based on actions of the particular user.

DETAILED DESCRIPTION

[0022] The present invention provides systems and methods for social data analysis, and more particularly for determining metrics for social data to allow for filtering and monitoring social data such as threads. The various embodiments are particularly useful in an on-demand multi-tenant database and/or application service.

General Overview

[0023] Systems, apparatus, and methods are provided for implementing enterprise level social and business information networking. Such embodiments can provide more efficient use of a database system. For instance, a user of a database system may not easily know when important information in the database has changed, e.g., about a project or client. Embodiments can provide stories about such changes and other events, thereby keeping users informed.

[0024] By way of example, a user can update a record (e.g. an opportunity such as a possible sale of 1000 computers). Once the update has been made, a story about the update can then automatically be sent (e.g. in a feed) to anyone subscribing to the opportunity or to the user. Thus, the user does not need to contact a manager regarding the change in the opportunity, since the story about the update is sent via a feed right to the manager’s feed page (or other page).

[0025] Next, mechanisms and methods for providing systems and methods for implementing enterprise level social and business information networking will be described with reference to example embodiments.

System Overview

[0026] FIG. 1 illustrates a block diagram of an environment 10 wherein an on-demand database service might be used. Environment 10 may include user systems 12, network 14, system 16, processor system 17, application platform 18, network interface 20, tenant data storage 22, system data storage 24, program code 26, and process space 28. In other embodiments, environment 10 may not have all of the components listed and/or may have other elements instead of, or in addition to, those listed above.

[0027] Environment 10 is an environment in which an on-demand database service exists. User system 12 may be any machine or system that is used by a user to access a database user system. For example, any of user systems 12 can be a handheld computing device, a mobile phone, a laptop computer, a work station, and/or a network of computing devices. As illustrated in FIG. 1 (and in more detail in FIG. 2) user systems 12 might interact via a network 14 with an on-demand database service, which is system 16.

[0028] An on-demand database service, such as system 16, is a database system that is made available to outside users that do not need to necessarily be concerned with building and/or maintaining the database system, but instead may be available for their use when the users need the database system (e.g., on the demand of the users). Some on-demand database services may store information from one or more tenants stored into tables of a common database image to form a multi-tenant database system (MTS). Accordingly, “on-demand database service 16” and “system 16” will be used interchangeably herein. A database image may include one or more database objects. A relational database management system (RDBMS) or the equivalent may execute storage and retrieval of information against the database object(s). Application platform 18 may be a framework that allows the applications of system 16 to run, such as the hardware and/or software, e.g., the operating system. In an embodiment, on-demand database service 16 may include an application platform 18 that enables creation, managing and executing one or more applications developed by the provider of the on-demand database service, users accessing the on-demand database service via user systems 12, or third party application developers accessing the on-demand database service via user systems 12.

[0029] The users of user systems 12 may differ in their respective capacities, and the capacity of a particular user system 12 might be entirely determined by permissions (permission levels) for the current user. For example, where a salesperson is using a particular user system 12 to interact with system 16, that user system has the capacities allotted to that salesperson. However, while an administrator is using that user system to interact with system 16, that user system has the capacities allotted to that administrator. In systems with a hierarchical role model, users at one permission level (profile type) may have access to applications, data, and database information accessible by a lower permission level user, but may not have access to certain applications, database information, and data accessible by a user at a higher permission level. Thus, different users will have different capabilities with regard to accessing and modifying application and database information, depending on a user’s security or permission level, also called authorization.

[0030] Network 14 is any network or combination of networks of devices that communicate with one another. For example, network 14 can be any one or any combination of a

LAN (local area network), WAN (wide area network), telephone network, wireless network, point-to-point network, star network, token ring network, hub network, or other appropriate configuration. As the most common type of computer network in current use is a TCP/IP (Transfer Control Protocol and Internet Protocol) network, such as the global internetwork of networks often referred to as the “Internet” with a capital “I,” that network will be used in many of the examples herein. However, it should be understood that the networks that embodiments might be use are not so limited, although TCP/IP is a frequently implemented protocol.

[0031] User systems 12 might communicate with system 16 using TCP/IP and, at a higher network level, use other common Internet protocols to communicate, such as HTTP, FTP, AFS, WAP, etc. In an example where HTTP is used, user system 12 might include an HTTP client commonly referred to as a “browser” for sending and receiving HTTP messages to and from an HTTP server at system 16. Such an HTTP server might be implemented as the sole network interface between system 16 and network 14, but other techniques might be used as well or instead. In some implementations, the interface between system 16 and network 14 includes load sharing functionality, such as round-robin HTTP request distributors to balance loads and distribute incoming HTTP requests evenly over a plurality of servers. At least as for the users that are accessing that server, each of the plurality of servers has access to the MTS’ data; however, other alternative configurations may be used instead.

[0032] In one embodiment, system 16, shown in FIG. 1, implements a web-based customer relationship management (CRM) system. For example, in one embodiment, system 16 includes application servers configured to implement and execute CRM software applications as well as provide related data, code, forms, webpages and other information to and from user systems 12 and to store to, and retrieve from, a database system related data, objects, and Webpage content. With a multi-tenant system, data for multiple tenants may be stored in the same physical database object, however, tenant data typically is arranged so that data of one tenant is kept logically separate from that of other tenants so that one tenant does not have access to another tenant’s data, unless such data is expressly shared. In certain embodiments, system 16 implements applications other than, or in addition to, a CRM application. For example, system 16 may provide tenant access to multiple hosted (standard and custom) applications, including a CRM application. User (or third party developer) applications, which may or may not include CRM, may be supported by the application platform 18, which manages creation, storage of the applications into one or more database objects and executing of the applications in a virtual machine in the process space of the system 16.

[0033] One arrangement for elements of system 16 is shown in FIG. 1, including a network interface 20, application platform 18, tenant data storage 22 for tenant data 23, system data storage 24 for system data 25 accessible to system 16 and possibly multiple tenants, program code 26 for implementing various functions of system 16, and a process space 28 for executing MTS system processes and tenant-specific processes, such as running applications as part of an application hosting service. Additional processes that may execute on system 16 include database indexing processes.

[0034] Several elements in the system shown in FIG. 1 include conventional, well-known elements that are explained only briefly here. For example, each user system 12

could include a desktop personal computer, workstation, laptop, PDA, cell phone, or any wireless access protocol (WAP) enabled device or any other computing device capable of interfacing directly or indirectly to the Internet or other network connection. User system **12** typically runs an HTTP client, e.g., a browsing program, such as Microsoft's Internet Explorer browser, Netscape's Navigator browser, Opera's browser, or a WAP-enabled browser in the case of a cell phone, PDA or other wireless device, or the like, allowing a user (e.g., subscriber of the multi-tenant database system) of user system **12** to access, process and view information, pages and applications available to it from system **16** over network **14**. Each user system **12** also typically includes one or more user interface devices, such as a keyboard, a mouse, trackball, touch pad, touch screen, pen or the like, for interacting with a graphical user interface (GUI) provided by the browser on a display (e.g., a monitor screen, LCD display, etc.) in conjunction with pages, forms, applications and other information provided by system **16** or other systems or servers. For example, the user interface device can be used to access data and applications hosted by system **16**, and to perform searches on stored data, and otherwise allow a user to interact with various GUI pages that may be presented to a user. As discussed above, embodiments are suitable for use with the Internet, which refers to a specific global internetwork of networks. However, it should be understood that other networks can be used instead of the Internet, such as an intranet, an extranet, a virtual private network (VPN), a non-TCP/IP based network, any LAN or WAN or the like.

[0035] According to one embodiment, each user system **12** and all of its components are operator configurable using applications, such as a browser, including computer code run using a central processing unit such as an Intel Pentium® processor or the like. Similarly, system **16** (and additional instances of an MTS, where more than one is present) and all of their components might be operator configurable using application(s) including computer code to run using a central processing unit such as processor system **17**, which may include an Intel Pentium® processor or the like, and/or multiple processor units. A computer program product embodiment includes a machine-readable storage medium (media) having instructions stored thereon/in which can be used to program a computer to perform any of the processes of the embodiments described herein. Computer code for operating and configuring system **16** to intercommunicate and to process webpages, applications and other data and media content as described herein are preferably downloaded and stored on a hard disk, but the entire program code, or portions thereof, may also be stored in any other volatile or non-volatile memory medium or device as is well known, such as a ROM or RAM, or provided on any media capable of storing program code, such as any type of rotating media including floppy disks, optical discs, digital versatile disk (DVD), compact disk (CD), microdrive, and magneto-optical disks, and magnetic or optical cards, nanosystems (including molecular memory ICs), or any type of media or device suitable for storing instructions and/or data. Additionally, the entire program code, or portions thereof, may be transmitted and downloaded from a software source over a transmission medium, e.g., over the Internet, or from another server, as is well known, or transmitted over any other conventional network connection as is well known (e.g., extranet, VPN, LAN, etc.) using any communication medium and protocols (e.g., TCP/IP, HTTP, HTTPS, Ethernet, etc.) as are well known. It will

also be appreciated that computer code for implementing embodiments of the present invention can be implemented in any programming language that can be executed on a client system and/or server or server system such as, for example, C, C++, HTML, any other markup language, Java™, JavaScript, ActiveX, any other scripting language, such as VBScript, and many other programming languages as are well known may be used. (Java™ is a trademark of Sun Microsystems, Inc.).

[0036] According to one embodiment, each system **16** is configured to provide webpages, forms, applications, data and media content to user (client) systems **12** to support the access by user systems **12** as tenants of system **16**. As such, system **16** provides security mechanisms to keep each tenant's data separate unless the data is shared. If more than one MTS is used, they may be located in close proximity to one another (e.g., in a server farm located in a single building or campus), or they may be distributed at locations remote from one another (e.g., one or more servers located in city A and one or more servers located in city B). As used herein, each MTS could include one or more logically and/or physically connected servers distributed locally or across one or more geographic locations. Additionally, the term "server" is meant to include a computer system, including processing hardware and process space(s), and an associated storage system and database application (e.g., OODBMS or RDBMS) as is well known in the art. It should also be understood that "server system" and "server" are often used interchangeably herein. Similarly, the database object described herein can be implemented as single databases, a distributed database, a collection of distributed databases, a database with redundant online or offline backups or other redundancies, etc., and might include a distributed database or storage network and associated processing intelligence.

[0037] FIG. 2 also illustrates environment **10**. However, in FIG. 2 elements of system **16** and various interconnections in an embodiment are further illustrated. FIG. 2 shows that user system **12** may include processor system **12A**, memory system **12B**, input system **12C**, and output system **12D**. FIG. 2 shows network **14** and system **16**. FIG. 2 also shows that system **16** may include tenant data storage **22**, tenant data **23**, system data storage **24**, system data **25**, User Interface (UI) **30**, Application Program Interface (API) **32**, PL/SOQL **34**, save routines **36**, application setup mechanism **38**, applications servers **100₁-100N**, system process space **102**, tenant process spaces **104**, tenant management process space **110**, tenant storage area **112**, user storage **114**, and application metadata **116**. In other embodiments, environment **10** may not have the same elements as those listed above and/or may have other elements instead of, or in addition to, those listed above.

[0038] User system **12**, network **14**, system **16**, tenant data storage **22**, and system data storage were discussed above in FIG. 1. Regarding user system **12**, processor system **12A** may be any combination of one or more processors. Memory system **12B** may be any combination of one or more memory devices, short term, and/or long term memory. Input system **12C** may be any combination of input devices, such as one or more keyboards, mice, trackballs, scanners, cameras, and/or interfaces to networks. Output system **12D** may be any combination of output devices, such as one or more monitors, printers, and/or interfaces to networks. As shown by FIG. 2, system **16** may include a network interface **20** (of FIG. 1) implemented as a set of HTTP application servers **100**, an application platform **18**, tenant data storage **22**, and system

data storage 24. Also shown is system process space 102, including individual tenant process spaces 104 and a tenant management process space 110. Each application server 100 may be configured to tenant data storage 22 and the tenant data 23 therein, and system data storage 24 and the system data 25 therein to serve requests of user systems 12. The tenant data 23 might be divided into individual tenant storage areas 112, which can be either a physical arrangement and/or a logical arrangement of data. Within each tenant storage area 112, user storage 114 and application metadata 116 might be similarly allocated for each user. For example, a copy of a user's most recently used (MRU) items might be stored to user storage 114. Similarly, a copy of MRU items for an entire organization that is a tenant might be stored to tenant storage area 112. A UI 30 provides a user interface and an API 32 provides an application programmer interface to system 16 resident processes to users and/or developers at user systems 12. The tenant data and the system data may be stored in various databases, such as one or more Oracle™ databases.

[0039] Application platform 18 includes an application setup mechanism 38 that supports application developers' creation and management of applications, which may be saved as metadata into tenant data storage 22 by save routines 36 for execution by subscribers as one or more tenant process spaces 104 managed by tenant management process 110 for example. Invocations to such applications may be coded using PL/SQL 34 that provides a programming language style interface extension to API 32. A detailed description of some PL/SQL language embodiments is discussed in commonly owned U.S. Pat. No. 7,730,478 entitled, METHOD AND SYSTEM FOR ALLOWING ACCESS TO DEVELOPED APPLICATIONS VIA A MULTI-TENANT ON-DEMAND DATABASE SERVICE, by Craig Weissman, filed Sep. 21, 2007, which is incorporated in its entirety herein for all purposes. Invocations to applications may be detected by one or more system processes, which manages retrieving application metadata 116 for the subscriber making the invocation and executing the metadata as an application in a virtual machine.

[0040] Each application server 100 may be communicably coupled to database systems, e.g., having access to system data 25 and tenant data 23, via a different network connection. For example, one application server 100₁ might be coupled via the network 14 (e.g., the Internet), another application server 100N₁ might be coupled via a direct network link, and another application server 100N might be coupled by yet a different network connection. Transfer Control Protocol and Internet Protocol (TCP/IP) are typical protocols for communicating between application servers 100 and the database system. However, it will be apparent to one skilled in the art that other transport protocols may be used to optimize the system depending on the network interconnect used.

[0041] In certain embodiments, each application server 100 is configured to handle requests for any user associated with any organization that is a tenant. Because it is desirable to be able to add and remove application servers from the server pool at any time for any reason, there is preferably no server affinity for a user and/or organization to a specific application server 100. In one embodiment, therefore, an interface system implementing a load balancing function (e.g., an F5 Big-IP load balancer) is communicably coupled between the application servers 100 and the user systems 12 to distribute requests to the application servers 100. In one embodiment, the load balancer uses a least connections algorithm to route

user requests to the application servers 100. Other examples of load balancing algorithms, such as round robin and observed response time, also can be used. For example, in certain embodiments, three consecutive requests from the same user could hit three different application servers 100, and three requests from different users could hit the same application server 100. In this manner, system 16 is multi-tenant, wherein system 16 handles storage of, and access to, different objects, data and applications across disparate users and organizations.

[0042] As an example of storage, one tenant might be a company that employs a sales force where each salesperson uses system 16 to manage their sales process. Thus, a user might maintain contact data, leads data, customer follow-up data, performance data, goals and progress data, etc., all applicable to that user's personal sales process (e.g., in tenant data storage 22). In an example of a MTS arrangement, since all of the data and the applications to access, view, modify, report, transmit, calculate, etc., can be maintained and accessed by a user system having nothing more than network access, the user can manage his or her sales efforts and cycles from any of many different user systems. For example, if a salesperson is visiting a customer and the customer has Internet access in their lobby, the salesperson can obtain critical updates as to that customer while waiting for the customer to arrive in the lobby.

[0043] While each user's data might be separate from other users' data regardless of the employers of each user, some data might be organization-wide data shared or accessible by a plurality of users or all of the users for a given organization that is a tenant. Thus, there might be some data structures managed by system 16 that are allocated at the tenant level while other data structures might be managed at the user level. Because an MTS might support multiple tenants including possible competitors, the MTS should have security protocols that keep data, applications, and application use separate. Also, because many tenants may opt for access to an MTS rather than maintain their own system, redundancy, up-time, and backup are additional functions that may be implemented in the MTS. In addition to user-specific data and tenant-specific data, system 16 might also maintain system level data usable by multiple tenants or other data. Such system level data might include industry reports, news, postings, and the like that are sharable among tenants.

[0044] In certain embodiments, user systems 12 (which may be client systems) communicate with application servers 100 to request and update system-level and tenant-level data from system 16 that may require sending one or more queries to tenant data storage 22 and/or system data storage 24. System 16 (e.g., an application server 100 in system 16) automatically generates one or more SQL statements (e.g., one or more SQL queries) that are designed to access the desired information. System data storage 24 may generate query plans to access the requested data from the database.

[0045] Each database can generally be viewed as a collection of objects, such as a set of logical tables, containing data fitted into predefined categories. A "table" is one representation of a data object, and may be used herein to simplify the conceptual description of objects and custom objects according to the present embodiments. It should be understood that "table" and "object type" may be used interchangeably herein. Each table generally contains one or more data categories logically arranged as columns or fields in a viewable schema. Each row or record of a table contains an instance of

data for each category (type) defined by the fields. For example, a CRM database may include a table that describes a customer with fields for basic contact information such as name, address, phone number, fax number, etc. Another table might describe a purchase order, including fields for information such as customer, product, sale price, date, etc. In some multi-tenant database systems, standard entity tables might be provided for use by all tenants. For CRM database applications, such standard entities might include tables for Account, Contact, Lead, Opportunity data, and other object types, each containing pre-defined fields. It should be understood that the word “entity” may also be used interchangeably herein with “object” and “table”, when entity or object is referring to a collection of objects or entities of a particular type.

[0046] In some multi-tenant database systems, tenants may be allowed to create and store custom objects, or they may be allowed to customize standard entities or objects, for example by creating custom fields for standard objects, including custom index fields. U.S. Pat. No. 7,779,039, entitled “Custom Entities and Fields in a Multi-Tenant Database System”, by Craig Weissman, filed Apr. 2, 2004, which is hereby incorporated herein by reference, teaches systems and methods for creating custom objects as well as customizing standard objects in a multi-tenant database system. In certain embodiments, for example, all custom entity data rows are stored in a single multi-tenant physical table, which may contain multiple logical tables per organization. It is transparent to customers that their multiple “tables” are in fact stored in one large table or that their data may be stored in the same table as the data of other customers.

Metrics Based Accessing of Social Feeds

[0047] Many social networking systems offer explicit filtering of social threads (or feeds). Feeds are generally filtered by: user, group, channel, subject, or keyword. These are all quite useful for viewing feeds, but there is also a need to access message feeds based upon feed metrics that track user behavior. With the explosion of social data, feed metrics will become more useful for filtering and monitoring within social applications.

[0048] Conventionally, feeds provided through internet browsing software have provided simple metrics, such as how many subscribers a feed has, or how many total messages a feed currently includes. Feeds also often provide a way to explicitly indicate that the user finds a particular feed item to be important or useful (e.g., FaceBook’s “Like” function). Software, such as the salesforce.com Chatter® Desktop, or a sophisticated Internet browser, that is able to provide a detailed level of feedback regarding implicit user behavior may allow for detection of more subtle user behavior on their computer, such as (1) performing a “mouse-over” of a given location even if no click occurs, (2) the duration of time a user passively views a designated area of content, (3) viewing the comments on a message, (4) “highlighting” (a.k.a. “selecting” text or an image), (5) copying the text of a feed, (6) expanding long messages from a truncated form, or (7) “zooming in” (increasing the view of the page). In addition, other user actions, while not so subtle, may also implicitly indicate user interest, such as commenting on a message, or explicitly forwarding a feed item to someone else. The ability to detect user behavior the implicitly indicates user interest in a feed item enables the collection of detailed user interaction data, which enables assessment of novel feed metrics.

[0049] FIG. 3 is a flowchart of a method **300** for accessing feeds based on feed metrics. The system accesses a database of feeds (**310**) and provides a feed to a network of users of the database (**320**).

[0050] In one embodiment, client software installed on a user’s computer (e.g., Chatter® Desktop) is able to use APIs built into the operating system to detect the use of peripheral input devices, such as the motion and button clicks of a mouse and keystrokes from a keyboard. In this manner, the client software is able to detect implicit user behavior, such as the duration of time that a user kept their mouse over a popup transient notification to hold it open (and prevent it from disappearing after just few seconds), or, on a mobile device, the duration of time after a user enlarged some text or an image using a pinch-zoom motion. Another example of detecting implicit user behavior may include measuring the speed with which a user reacts to a notification (whether visual or auditory, etc.)—for example, upon receiving a notification, how quickly did the user react to the notification, e.g., by rapidly sliding their mouse over to a control interface (or, on a mobile device, pressing a button or tilting the device) to mute an alert sound. The users view feed items and interact with the feed-implicit user behavior is thereby captured (**330**). From such implicit user behavior, inferential user interaction data can be gathered. Such inferential user interaction data may provide feedback on user behavior that does not involve explicit actions, such as clicking on a link. In one embodiment, sophisticated browser plug-ins (e.g., Flash) may provide the ability to detect implicit user behavior of the type described above.

[0051] Such implicit user behavior is captured in inferential user interaction data (e.g., statistics generated by client software on a user’s computing device), which is collected (**340**) and analyzed to determine various feed metrics (**350**). Analysis of the inferential user interaction data may comprise different aspects of implicit user behavior, such as calculating the average time that a typical user held open a popup notification for a particular feed, or calculating the average number of users who held open the popup notification, or calculating the viral nature of the particular feed (i.e., the acceleration over time in the number of users who affirmatively acted to hold open the popup notification for the particular feed. For any of these actions, another interesting analysis might look at the occurrence of subsequent actions (or lack thereof), such as determining that a high percentage of users take an affirmative action to open and read a particular message but fail to affirmatively indicate interest or perhaps even affirmatively take an action to characterize the message as being of low interest or little use or even offensive.

[0052] The various feed metrics may include: metrics that track the user consumption of content from the feed (e.g., a most- and/or least-viewed feed metric), user responsiveness within the context of the feed (e.g., a most- and/or least-commented feed metric, or other user actions to mark or characterize the feed or feed elements), proliferation of content from the feed by users (e.g., a most- and/or least-forwarded feed metric, or a most- and/or least-viral feed metric), and feed life.

[0053] In one embodiment, a Most- and/or Least-Commented Feed Metric is used. This metric is determined by counting comments added to a feed, for example, feeds which have the most (or least) comments, or feeds that meet a minimum (maximum) criteria for comments.

[0054] In one embodiment, a Most- and/or Least-Viewed Feed Metric is used. Four possible ways of generating this metric include:

[0055] 1) The time during which a feed is 'on screen' or viewable by the user.

[0056] 2) Whether or not a user viewed comments associated with the feed, regardless of whether they added a new comment.

[0057] 3) The time during which a feed is on screen or viewable, but only when the user has been determined to actively be using the application, e.g., as measured by user input activity such as mouse or keyboard activity.

[0058] 4) Whether or not a user chose to hide or delete or otherwise de-prioritize the feed. In one embodiment, this metric identifies feeds with respect to user views regardless of whether the user comments upon the feed or not (or feeds that meet a minimum criteria for views).

[0059] In one embodiment, a Most- and/or Least-Forwarded Feed Metric is used. Forwarding of a feed can be measured, including by:

[0060] 1) How many times the text of the feed is copied (the actual copy operation can be noted), with the assumption that it will be pasted into a document or some other communication.

[0061] 2) How many times an operation to get a link or forward a feed is accessed. Again, the operation used to link or forward the feed can be intercepted/noted to create this metric. A simple way to do this is to provide a button to forward a feed. In one embodiment, this metric identifies feeds that are sent to other users in one form or another, or, in the 'least' case, feeds that no one is bothering to forward.

[0062] In one embodiment, a Most- and/or Least-Viral Feed Metric is used. The viral nature of a feed can be measured by determining how quickly the rate at which a feed is viewed, forwarded, commented upon, or subscribed to by users increases over a designated period of time.

[0063] In one embodiment, a Feed Life Metric is used. This metric identifies feeds that continue to be viewed or receive updates over a long period of time (or feeds that are no longer accessed). In some embodiments, this metric combines the Most-/Least-Viewed and/or the Most-/Least-Forwarded Feed Metrics with the initial and most recent dates on which the feed is accessed to determine the longevity of a feed.

[0064] In some embodiments, one or more of the metrics may be weighted (360) according to any conventional methodology. Such weighting of the metrics may affect how combinations of metrics are analyzed.

[0065] In some embodiments, a feed metric may be calculated with respect to a subgroup of users, wherein users in the subgroup share some common characteristic. Common characteristics may include demographic similarities, association with a particular organization or object (e.g., Account, Lead, Opportunity), location in a particular geographic region, use of particular types of technology (e.g., accessing a feed from a mobile device using a particular application or version of the operating system), etc.

[0066] Finally, the system executes an action with respect to the feed, based on the feed metrics (370). In some embodiments, the system uses the feed metrics to select actions to take. In some embodiments, the system may pass the feed metrics as input to the action and modify the action based on one or more of the feed metrics. In some embodiments, feed metrics may be used to, for example: (1) determine which feeds are important to watch for marketing or support rea-

sons; (2) indicate which feeds are critical to viewers; (3) filter multiple feeds down to a "most important" list for daily viewing; (4) cache commonly accessed feeds; (5) determine whether to discontinue a feed that is not drawing interest; (6) determine whether and/or how to monetize a feed, and (7) automatically assign badges or other distinguishing marks to a feed. Some embodiments using feed metrics are described in further detail in U.S. patent application Ser. No. 13/112,805, (attorney docket no. 88262-795689; 374US2), which is hereby incorporated by reference for all purposes, and which claims priority to U.S. Patent Application No. 61/354,638.

[0067] The specific details of particular embodiments may be combined in any suitable manner without departing from the spirit and scope of embodiments of the invention. However, other embodiments of the invention may be directed to specific embodiments relating to each individual aspect, or specific combinations of these individual aspects.

[0068] It should be understood that any of the embodiments of the present invention can be implemented in the form of control logic using hardware and/or using computer software in a modular or integrated manner. Based on the disclosure and teachings provided herein, a person of ordinary skill in the art will know and appreciate other ways and/or methods to implement embodiments of the present invention using hardware and a combination of hardware and software.

[0069] Any of the software components or functions described in this application may be implemented as software code to be executed by a processor using any suitable computer language such as, for example, Java, C++ or Perl using, for example, conventional or object-oriented techniques. The software code may be stored as a series of instructions or commands on a computer readable medium for storage and/or transmission, suitable media include random access memory (RAM), a read only memory (ROM), a magnetic medium such as a hard-drive or a floppy disk, or an optical medium such as a compact disk (CD) or DVD (digital versatile disk), flash memory, and the like. The computer readable medium may be any combination of such storage or transmission devices.

[0070] Such programs may also be encoded and transmitted using carrier signals adapted for transmission via wired, optical, and/or wireless networks conforming to a variety of protocols, including the Internet. As such, a computer readable medium according to an embodiment of the present invention may be created using a data signal encoded with such programs. Computer readable media encoded with the program code may be packaged with a compatible device or provided separately from other devices (e.g., via Internet download). Any such computer readable medium may reside on or within a single computer program product (e.g. a hard drive or an entire computer system), and may be present on or within different computer program products within a system or network. A computer system may include a monitor, printer, or other suitable display for providing any of the results mentioned herein to a user.

[0071] Any of the methods described herein may be totally or partially performed with a computer system including a processor, which can be configured to perform the steps. Thus, embodiments can be directed to computer systems configured to perform the steps of any of the methods described herein, potentially with different components performing a respective steps or a respective group of steps. Although presented as numbered steps, steps of methods herein can be performed at a same time or in a different order.

Additionally, portions of these steps may be used with portions of other steps from other methods. Also, all or portions of a step may be optional. Additionally, any of the steps of any of the methods can be performed with modules, circuits, or other means for performing these steps.

[0072] The above description of exemplary embodiments of the invention has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form described, and many modifications and variations are possible in light of the teaching above. The embodiments were chosen and described in order to best explain the principles of the invention and its practical applications to thereby enable others skilled in the art to best utilize the invention in various embodiments and with various modifications as are suited to the particular use contemplated.

What is claimed is:

1. A method of accessing feeds based on one or more feed metrics, the method comprising:

receiving inferential user interaction data generated in response to explicit behavior of a plurality of users in relation to a plurality of feeds, the inferential user interaction data indicating implicit user behavior of the plurality of users, the inferential user interaction data captured in response to one of client software operating on a computing device and a browser plug-in;

determining, using one or more processors of a computing device, one or more feed metrics based on statistics of the inferential user interaction data for the plurality of users;

processing the one or more feed metrics with respect to the plurality of feeds; and

altering a characteristic associated with at least one feed based on the one or more processed feed metrics.

2. The method of claim 1, wherein the statistics of the inferential user interaction data comprise at least one of: user consumption, user responsiveness, content proliferation, and feed life.

3. The method of claim 1, wherein altering a characteristic associated with the feed comprises one or more of: discontinuing the feed, characterizing an importance of the feed, determining that the feed can be monetized, determining that the feed is to be cached, and determining that intervention in the feed is advisable.

4. The method of claim 1, wherein altering a characteristic associated with the feed comprises selecting an action to be executed based on the one or more feed metrics.

5. The method of claim 1, wherein altering a characteristic associated with the feed comprises providing the one or more feed metrics as input to an action, and wherein the action is modified based on the one or more feed metrics.

6. The method of claim 1, wherein the implicit user behavior comprises one or more of: hovering a cursor over a location of a page, viewing a designated area of content for a duration of time, viewing a comment on a message, selecting a text or image on a page, copying a text on a page, expanding a message from a truncated form, and increasing a view of a page.

7. The method of claim 1, wherein determining the one or more feed metrics based on statistics of the inferential user interaction data comprises determining a change in a feed metric over time.

8. The method of claim 1, wherein determining the one or more feed metrics based on statistics of the inferential user

interaction data comprises determining a feed metric with respect to a subgroup of users sharing a common characteristic.

9. The method of claim 1, wherein determining the one or more feed metrics based on statistics of the inferential user interaction data comprises determining a first feed metric with respect to a second feed metric.

10. The method of claim 9, wherein the first feed metric is average user responsiveness or average user-initiated content proliferation, and the second feed metric is average user consumption.

11. The method of claim 9, wherein the first feed metric is average user consumption, average user responsiveness, or average user-initiated content proliferation, and the second feed metric is feed life.

12. The method of claim 1, wherein altering a characteristic associated with the feed comprises determining the feed as being important.

13. The method of claim 12, wherein determining the feed as being important is based on a user consumption feed metric, the user consumption feed metric indicating that the feed has a high average user consumption.

14. A non-transitory computer-readable storage medium storing instructions executable by a processor to cause a method to be performed for accessing feeds based on one or more feed metrics, the method comprising:

receiving inferential user interaction data generated in response to explicit behavior of a plurality of users in relation to a plurality of feeds, the inferential user interaction data indicating implicit user behavior of the plurality of users, the inferential user interaction data captured in response to one of client software operating on a client device and a browser plug-in;

determining, using one or more processors of a computing device, one or more feed metrics based on statistics of the inferential user interaction data for the plurality of users;

processing the one or more feed metrics with respect to the plurality of feeds; and

altering a characteristic associated with at least one feed based on the one or more processed feed metrics.

15. The non-transitory computer-readable storage medium of claim 14, wherein the statistics of the inferential user interaction data comprise at least one of user consumption, user responsiveness, content proliferation, and feed life.

16. The non-transitory computer-readable storage medium of claim 14, wherein altering a characteristic associated with the feed comprises one or more of: discontinuing the feed, characterizing an importance of the feed, determining that the feed can be monetized, determining that the feed is to be cached, or determining that intervention in the feed is advisable.

17. The non-transitory computer-readable storage medium of claim 14, wherein the implicit user behavior comprises one or more of: hovering a cursor over a location of a page, viewing a designated area of content for a duration of time, viewing a comment on a message, selecting a text or image on a page, copying a text on a page, expanding a message from a truncated form, and increasing a view of a page.

18. The non-transitory computer-readable storage medium of claim 14, wherein determining the one or more feed metrics based on statistics of the inferential user interaction data comprises determining the change in a feed metric over time.

19. A computing device for accessing feeds based on one or more feed metrics, the computing device comprising:

one or more processors capable of executing one or more instructions configured to cause:

receiving inferential user interaction data generated in response to explicit behavior of a plurality of users in relation to a plurality of feeds, the inferential user interaction data indicating implicit user behavior of the plurality of users, the inferential user interaction data captured in response to one of client software operating on a client device and a browser plug-in;

determining one or more feed metrics based on statistics of the inferential user interaction data for the plurality of users;

processing the one or more feed metrics with respect to the plurality of feeds; and

altering a characteristic associated with at least one feed based on the one or more processed feed metrics.

20. The computing device of claim **19**, wherein the statistics of the inferential user interaction data comprise at least one of: user consumption, user responsiveness, content proliferation, and feed life.

21. A system for accessing feeds based on one or more feed metrics, the system comprising:

a database storing a plurality of objects, wherein the database is running on one or more computing devices;

one or more processors associated with the one or more computing devices, the one or more processors capable of executing one or more instructions configured to cause:

receiving inferential user interaction data generated in response to explicit behavior of a plurality of users in relation to a plurality of feeds, the inferential user interaction data indicating implicit user behavior of the plurality of users, the inferential user interaction data captured in response to one of client software operating on a client device and a browser plug-in;

determining one or more feed metrics based on statistics of the inferential user interaction data for the plurality of users;

processing the one or more feed metrics with respect to the plurality of feeds; and

altering a characteristic associated with at least one feed based on the one or more processed feed metrics.

22. The system of claim **21**, wherein the statistics of the inferential user interaction data comprise at least one of: user consumption, user responsiveness, content proliferation, and feed life.

* * * * *