

(51) International Patent Classification:
G06F 17/30 (2006.01)(21) International Application Number:
PCT/EP2010/055408(22) International Filing Date:
23 April 2010 (23.04.2010)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
12/432,808 30 April 2009 (30.04.2009) US(71) Applicant (for all designated States except US): **INTERNATIONAL BUSINESS MACHINES CORPORATION** [US/US]; New Orchard Road, Armonk, New York 10504 (US).(71) Applicant (for MG only): **IBM UNITED KINGDOM LIMITED** [GB/GB]; PO Box 41, North Harbour, Portsmouth Hampshire PO6 3AU (GB).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **ROITMAN, Hag-gai** [IL/IL]; IBM Israel, Haifa University, Mount Carmel, 31905 Haifa (IL). **CARMEL, David** [IL/IL]; IBM Israel, Haifa University, Mount Carmel, 31905 Haifa (IL). **SHMUELI-SCHEUER, Michal** [IL/IL]; IBM Israel, 6, Habarzel Street, 69710 Ramat Hahayal (IL). **MASS, Yosi** [IL/IL]; IBM Israel, 6, Habarzel Street, 69710 Ramat Hahayal (IL).(74) Agent: **ROBERTS, Scott**; IBM United Kingdom Limited, Intellectual Property Law, Hursley Park, Winchester Hampshire SO21 2JN (GB).

(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, 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, IS, JP, KE, KG, KM, 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, PE, PG, PH, PL, PT, RO, RS, RU, 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.

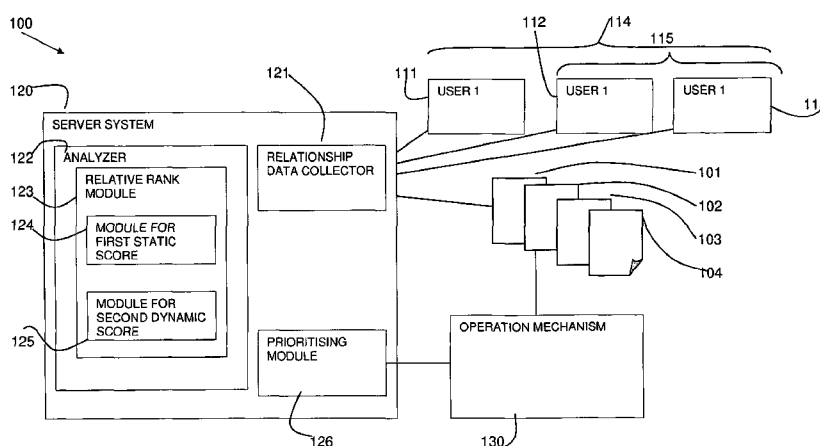
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS,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

Published:

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

(54) Title: METHOD AND SYSTEM FOR PRIORITISING OPERATIONS ON NETWORK OBJECTS

FIG. 1



(57) Abstract: A method and system for prioritising operations on network objects are provided. The method includes gathering available relationship data on the relationships between network entities, wherein network entities are network users and network objects. The relationship data for a network entity is analysed and a first relative score is determined based on the relationship data. For a network object, a second relative score is determined which is a dynamic score based on user interactions with the network object and formed using the first relative scores of network entities interacting with the object. The method then prioritizes an operation on a network object using the second relative score.

METHOD AND SYSTEM FOR PRIORITISING OPERATIONS ON NETWORK OBJECTS

FIELD OF THE INVENTION

This invention relates to the field of prioritising operations on network objects. In particular, the invention relates to prioritising operations on network objects using available data.

BACKGROUND OF THE INVENTION

The term "Web 2.0" refers to a perceived second generation of web development and design, that aims to facilitate communication, secure information sharing, interoperability, and collaboration on the World Wide Web. Web 2.0 concepts have led to the development and evolution of web-based communities, hosted services, and applications; such as social-networking, video-sharing, wikis, blogs, and folksonomies.

The sometimes complex and continually evolving technology infrastructure of Web 2.0 includes server-software, content-syndication, messaging-protocols, standards-oriented browsers with plugins and extensions, and various client-applications.

Web 2.0 websites typically include some of the following features/techniques.

- Search - the ease of finding information through keyword search which makes the platform valuable.
- Links - guides to important pieces of information. The best pages are the most frequently linked to.
- Authoring – the ability to create constantly updating content over a platform that is shifted from being the creation of a few to being the constantly updated, interlinked work. In wikis, the content is iterative in the sense that the people undo and redo each other's work. In blogs, content is cumulative in that posts and comments of individuals are accumulated over time.
- Tags - categorization of content by creating tags that are simple, one-word descriptions to facilitate searching and avoid rigid, pre-made categories.

- Extensions - automation of some of the work and pattern matching by using algorithms.
- Signals - the use of RSS (Really Simple Syndication) technology to notify users with any changes of the content, e.g., by sending e-mails to them.

5

Web 2.0 further introduced the notion of social networks which is a social structure made of nodes (which are generally individuals or organizations) that are tied by one or more specific types of interdependency, such as values, visions, ideas, financial exchange, friendship, kinship, dislike, conflict or trade.

10

In Web 2.0 users become not only consumers of information but also producers of data resulting in data which can be used to improve operations on web data objects.

SUMMARY OF THE INVENTION

15

According to a first aspect of the present invention there is provided a method of prioritising operations on network objects, comprising: gathering available relationship data on the relationships between network entities, wherein network entities are network users and network objects; analysing the relationship data for a network entity; determining for a network entity a first relative score based on the relationship data; determining for a network object a second relative score which is a dynamic score based on user interactions with the network object and formed using the first relative scores of network entities interacting with the object; and prioritising an operation on a network object using the second relative score; wherein any of said steps are implemented in either of computer hardware or computer software and embodied in a computer-readable medium. Preferably the available relationship data is Web 2.0 available relationship data.

20

25

The network entities may be web entities including web users and web objects or other entities available via a network.

30

The step of determining for a network object a second relative score includes may use the history of users' interactions with the object to provide the second relative score.

The step of determining for a network object a second relative score may include predicting users' interactions with the object to provide the second relative score.

5 The Web 2.0 available relationship data includes metadata generated by the network users and may include one or more of: social network data, folksonomy data, object authoring data, and object updating data.

10 In one embodiment, the step of prioritising an operation on a network object using the second relative score prioritises caching of network objects. The second relative score may be a dynamic score predicting the probability of future requests for cached network objects based on past users requests.

15 In another embodiment, the step of prioritising an operation on a network object using the second relative score prioritises crawling of network objects. The second relative score may be a dynamic score predicting where updates are likely to occur in network objects based on past user update patterns.

20 According to a second aspect of the present invention there is provided a computer program product for prioritising operations on network objects, the computer program product comprising: a computer readable medium; computer program instructions operative to: gathering Web 2.0 available relationship data on the relationships between network entities, wherein network entities are network users and network objects; analysing the relationship data for a network entity; determining for a network entity a first relative score based on the relationship data; determining for a network object a second relative score which is a
25 dynamic score based on user interactions with the network object and formed using the first relative scores of network entities interacting with the object; and prioritising an operation on a network object using the second relative score; wherein said program instructions are stored on said computer readable medium.

30 According to a third aspect of the present invention there is provided a system for prioritising operations on network objects, comprising: a processor; a collector for gathering Web 2.0 available relationship data on the relationships between network entities, wherein

network entities are network users and network objects; an analyzer for analysing the relationship data for a network entity; a module for determining for a network entity a first relative score based on the relationship data; a module for determining for a network object a second relative score which is a dynamic score based on user interactions with the network object and formed using the first relative scores of network entities interacting with the object; and a module for prioritising an operation on a network object using the second relative score; wherein any of said collector, analyzer, and modules are implemented in either of computer hardware or computer software and embodied in a computer readable medium.

The module for determining for a network object a second relative score may include using the history of users' interactions with the object obtained from a log to provide the second relative score.

The module for determining for a network object a second relative score may include predicting users' interactions with the object to provide the second relative score.

In one embodiment, the module for prioritising an operation on a network object using the second relative score prioritises caching of network objects. The second relative score may be a dynamic score predicting the probability of future requests for cached network objects based on past users requests.

In another embodiment, the module for prioritising an operation on a network object using the second relative score prioritises crawling of network objects. The second relative score may be a dynamic score predicting where updates are likely to occur in network objects based on past user update patterns.

According to a fourth aspect of the present invention there is provided a method of providing a service to a customer over a network of prioritising operations on network objects, the service comprising: gathering Web 2.0 available relationship data on the relationships between network entities, wherein network entities are network users and network objects; analysing the relationship data for a network entity; determining for a network entity a first

relative score based on the relationship data; determining for a network object a second relative score based on a combination of the first relative scores of network objects and network users; and prioritising an operation on a network object using the second relative score; wherein any of said steps are implemented in either of computer hardware or computer software and embodied in a computer-readable medium.

BRIEF DESCRIPTION OF THE DRAWINGS

Preferred embodiments of the present invention will now be described, by way of example only, with reference to the following drawings in which:

Figure 1 is a block diagram of a system in accordance with a preferred embodiment of the present invention;

Figure 2 is a block diagram of a computer system in which a preferred embodiment of the present invention may be implemented;

Figure 3 is a flow diagram of a method in accordance with a preferred embodiment of the present invention;

Figure 4 is a flow diagram of a method of caching in accordance with a preferred embodiment of the present invention; and

Figure 5 is a flow diagram of a method of crawling in accordance with a preferred embodiment of the present invention.

It will be appreciated that for simplicity and clarity of illustration, elements shown in the figures have not necessarily been drawn to scale. For example, the dimensions of some of the elements may be exaggerated relative to other elements for clarity. Further, where considered appropriate, reference numbers may be repeated among the figures to indicate corresponding or analogous features.

DETAILED DESCRIPTION OF THE INVENTION

In the following detailed description, numerous specific details are set forth in order to provide a thorough understanding of the invention. However, it will be understood by those skilled in the art that the present invention may be practiced without these specific details.

In other instances, well-known methods, procedures, and components have not been described in detail so as not to obscure the present invention.

5 A method and system are described of prioritising operations on network objects using available data such as Web 2.0 available data. Network objects include objects provided via a network. The network may be the Internet in which case the network objects are web objects, or an intranet, or other form of network in which Web 2.0 data is generated. Web objects may include web documents, web pages, etc. Web 2.0 available data includes, among others, social networks data (i.e., user to user relationships), and folksonomy or
10 social indexing data (i.e., object to object and user to object relationships such as (tag, document), (user, document), (user, tag)).

The term network entities is used to include both network objects and network users who interact with the objects and therefore have relationships with the objects. The users may be
15 individuals, automated computers, groups, organisations, etc.

Using an analysis the data Web 2.0 available data, the relative social rank can be obtained for each object and used to prioritize operations on the object, for example, data caching and gathering tasks.
20

Objects can be prioritised that are both important due to their static social rank and dynamic social rank. The dynamic social rank is obtained by examining the history of users that interact with the object and related objects.

25 Referring to Figure 1, a system 100 is provided for prioritising network objects 101-104 which may be objects available over a network, including the Internet, such as web documents, web pages, etc. Users 111-113 interact with each other and the network objects 101-104.

30 Web 2.0 includes new social network technology and object and user interaction which can be used as metadata and used to rank or score the importance of network objects and users.

The users' 111-113 interaction with each other is defined by social networks 114, 115. A social network is a social structure made of nodes of users (which are generally individuals or organizations) that are tied by one or more specific types of interdependency, such as values, visions, ideas, financial exchange, friendship, kinship, dislike, conflict, trade, familiarity, similarity, or any kind of implicit relationship.

The objects 101-104 have relationships to other objects and to users 111-113 which are defined by folksonomies, authoring, updating, and other means.

The combined data from Web 2.0 sources relating to the relationships between users 111-113, objects 101-104, and users and objects is referred to as relationship data.

A server system 120 is described including a relationship data collector 121 and an analyser 122 of the relationship data for an object. The analyser 122 includes a relative rank module 123 for rating the relative rank of objects.

The relative rank module 123 includes a module 124 for determining for entities (both web objects and web users) a first relative score which is a static score for the entity. The relative rank module 123 includes a module 125 for determining for network objects a second relative score which is a dynamic score based on user interaction with the object. The system 120 also includes a prioritising module 126 for prioritising objects based on the relative rank.

The objects 101-104 are operated on by some form of operation mechanism 130 using the prioritisation of the prioritising module 126. The operation mechanism 130 may be, for example, a web caching mechanism, or a content management mechanism.

Referring to Figure 2, an exemplary system for implementing embodiments of the invention includes a data processing system 200 suitable for storing and/or executing program code including at least one processor 201 coupled directly or indirectly to memory elements through a bus system 203. The memory elements can include local memory employed during actual execution of the program code, bulk storage, and cache memories which

provide temporary storage of at least some program code in order to reduce the number of times code must be retrieved from bulk storage during execution.

5 The memory elements may include system memory 202 in the form of read only memory (ROM) 204 and random access memory (RAM) 205. A basic input/output system (BIOS) 206 may be stored in ROM 204. System software 207 may be stored in RAM 205 including operating system software 208. Software applications 210 may also be stored in RAM 205.

10 The system 200 may also include a primary storage means 211 such as a magnetic hard disk drive and secondary storage means 212 such as a magnetic disc drive and an optical disc drive. The drives and their associated computer-readable media provide non-volatile storage of computer-executable instructions, data structures, program modules and other data for the system 200. Software applications may be stored on the primary and secondary storage means 211, 212 as well as the system memory 202.

15 The computing system 200 may operate in a networked environment using logical connections to one or more remote computers via a network adapter 216.

20 Input/output devices 213 can be coupled to the system either directly or through intervening I/O controllers. A user may enter commands and information into the system 200 through input devices such as a keyboard, pointing device, or other input devices (for example, microphone, joy stick, game pad, satellite dish, scanner, or the like). Output devices may include speakers, printers, etc. A display device 214 is also connected to system bus 203 via an interface, such as video adapter 215.

25 Referring to Figure 3, a flow diagram 300 shows the described method. Web 2.0 available relationship data is gathered 301 for relationships between network entities. The network entities include network objects and network users and the relationships may be user to user, object to object, or user to object.

30 The gathered relationship data is analyzed 302 for a network entity. A first relative score is determined 303 for a network entity which is a static score based on the importance of the

network entity, which may be a user or an object. A second relative score is then determined
304 for a network object which is a dynamic score based on user interactions with the object
and formed using the first relative scores of entities interacting with the object.

5 An operation on a network object, such as caching or crawling, is prioritized 305 using the
second relative score.

In a first described embodiment, the operation is data caching and a new caching policy is
built on the prioritisation.

10

In a second described embodiment, the operation is data crawling and, there is another added
value which is described which is the usage of users' update patterns as part of the crawling
strategy, where the crawler can use the identity of users and the social networks overlays to
predict where updates may occur (e.g., by finding similar users that share similar update
15 (rate + content based) patterns to a user whose update was discovered by the system).

15

A first embodiment is described in the context of web caching. Web 2.0 requires an
increasing bandwidth consumption to transfer large multimedia objects (for example, video
and high resolution photos). A common solution to this problem is web caching.

20

Improved caching performance of a content management system can be obtained by utilizing
the additional metadata that can be derived using social networks analysis. Using social
networks ranking techniques, a new caching policy is described that is "social sensitive".
Hence, the described policy ranks each object in the cache by its relative importance as
25 derived using social networks analysis.

25

The object's rank combines both the object's own rank and the rank of users that request the
object in the system. Therefore, an object that is more important and is requested by more
important "authoritative" users in the system is predicted to be more important and has
30 higher chance to be requested in the future. The described caching policy can be referred to
as "Least Socially influencing First (LSIF)". The proposed caching policy may also be used
to boost other existing caching policies.

30

A described caching policy predicts the probability of future requests for objects managed by the cache management system based on both dynamic and static statistical data gathered from analysis of social networks that exist within different communities of a content management system or outside of it.

5

The caching policy assigns each object a combination of static and dynamic scores. The static score of an object within the cache is determined by the relative rank of the object as derived from social networks analysis. For example, the object's FolkRank in a Folksonomy analysis such as described in A. Hotho, R. Jaschke, C. Schmitz, G. Stumme, "FolkRank: A Ranking Algorithm for Folksonomies"; its HubRank as described in S. Chakrabati, "Dynamic Personalized PageRank in Entity-Relation Graphs, WWW 2007, or its EntityRank as derived from a Multi-Entity Graph analysis as described in T. Cheng, X. Yan, K.C.C. Chang, "EntityRank: Searching Entities Directly and Holistically".

10

15

The dynamic score of an object predicts the probability of future requests for the object by other users in the system that may be influenced by other more authoritative users that requested the object in the past. This probability is determined using a log analysis of the history of previous requests for the object by different users in the system and estimating each user's relative authority in the system as derived from an analysis of the user's social networks or folksonomy data (For example, determined by user's FolkRank or HubRank in the system).

20

This dynamic score follows the following basic principle:

Given two objects in the cache O1 and O2 that were requested by users U1 and U2

25

respectively, then if user U1 is more authoritative than user U2 (as derived from the analysis of the users' social networks or folksonomy data), then O1 is predicted to be more qualitative than O2 in the system (i.e. voted as a better object by a more important user in the system).

30

Thus, the requesting user's relative authority in the system implies more chance of a recommendation for the object (or voting) by these users to other (less authoritative) users in their social networks (i.e., propagation of recommendations about the object in the social network). Therefore, an object that has more authoritative users requesting it, is predicted to

be more likely required by more users that relate to those more authoritative users that requested the object in the past and may propagate recommendations to other less authoritative users.

Metadata gathered from an analysis of the system's social networks (e.g., Multi-Entity Graph analysis or Folksonomy analysis) is used to derive for each entity in the system (either user or object) its relative rank in the system (herein denoted as *SocialRank*).

For each object in the system, the following metadata is maintained:

- Static metadata: *SocialRank*(O) of the object (social) rank in the system.
- Dynamic metadata: H(O) : a log that contains the request history of each object in the system.
- For each entry in the log, the following data is kept:
 - Requester user id (Ui)
 - *SocialRank*(Ui) of the user in the system.
 - The id of requested object (Oj)
 - The object request time tk

The static part of an object Oj score (S(Oj)) in the cache is calculated as the relative (normalized) rank of the object as derived from the social networks analysis:

$$S(O_j) = \frac{SocialRank(O_j)}{\sum_{O_j \in Cache} SocialRank(O_j)}$$

The dynamic part of an object Oj score (D(Oj)) in the cache is calculated as the relative rank of the object as derived from the history of requests for the object in the system's log by different users and their relative rank as derived from the social networks analysis.

$$D(O_j) = \frac{\sum_{i \in H(O_j)} SocialRank(U_i)}{\sum_{O_j \in Cache} \sum_{i \in H(O_j)} SocialRank(U_i)}$$

The time of requests can also be utilized to further refine the dynamic score to consider also the decay in object's relative importance among different users in the system (for example, as time passes by from the object request time by some user, there is less chance for

propagation of recommendations about the object by that user to other users in his social network that rank this user as more authoritative). Furthermore, the more authoritative the user, the decay is longer.

$$D(O_j) = \frac{\sum_{i \in H(O_j)} e^{\text{SocialRank}(U_i)^{-1}(t_k - t)}}{\sum_{O_j \in \text{Cache}} \sum_{i \in H(O_j)} e^{\text{SocialRank}(U_i)^{-1}(t_k - t)}}, \text{ where } t \text{ is current system}$$

time.

Finally, calculate the object's score in the cache as:

$$\text{Score}(O_j) = S(O_j) \times D(O_j)$$

Referring to Figure 4, a flow diagram 400 shows a method of caching an object. A request for an object is received 401 and a record is made 402 of the request (time, user ID, object ID) in the history log. It is determined 403 if the object is in the cache. If so return 404 the object in the cache. If not, rank 405 objects in the cache according to their social score, and replace 406 the object in the cache with the lowest social score with the current requested object. Then return 404 the object in the cache.

Using the new definition of object score in the cache, a new cache placement/replacement policy is defined, which may be termed: "Least Socially influencing First (LSIF)", where an object with the current lowest social score in the cache is picked for replacement.

The described method measures the importance of an object based on social network analysis (for example, the object folkrank, hubrank, etc.). Such analysis is based on folksonomy analysis which considers users and objects and the relationships among them to calculate the overall (static) score of an object. This score is SocialRank of the object.

The method further uses the same social analysis to determine the importance of the users that submit requests to the content management system. This importance implies the authority of different users in the system. This user importance is the SocialRank of the user that requests objects.

Using the SocialRank measure both for requested objects and for the users that request different objects two scores are calculated for each object managed in the system. The first is a static score based on the object own SocialRank in the system. The second is a dynamic score which is based on the request history to each object in the system by different users.

5 The premise is that an object that has more requests by more authoritative users is a more important object and that such an object has more chance to be requested in the future by follower users of authoritative users that already requested the object.

Both static and dynamic scores are then combined to a single unified object importance
10 measure.

A second described embodiment is in the context of data crawling. Many applications use crawlers to crawl all or parts of the web. For example, a web search engine needs to continuously crawl the web to maintain an updated index of the web. There are three
15 important characteristics of the web that generate a scenario in which web crawling is very difficult: its large volume, its fast rate of change, and dynamic page generation. The large volume implies that the crawler can only download a fraction of the web pages within a given time, so it needs to prioritize its downloads.

20 Traditional web crawlers try to devise policies to predict which pages should be crawled at a given time. The policies are based on estimating freshness of a page based on its past change rate. For each page p the rate of changes (denoted λ_p) is associated. The rate of changes could be based on some statistics from a web server (such as last modification time) or by comparing copies of the same page at different time stamps. Based on this, the crawler
25 can estimate how “fresh” is the copy of each page, and to advise the best re-visiting order in order to maximize the freshness on the entire web.

In Web 2.0 users become not only consumers of information but also producers of data, thus page updates are no more “black box” or some “system updates”, but rather, updates could
30 be easily associated with users. Bloggers update their blogs regularly at the rate of over 1.6 million posts per day, or over 18 updates a second.

Traditional web crawlers that look only at past page updates, but ignore the authority of users that updated the pages, and ignore the social networking of those users that updated similar pages, can not optimize their policies for such an environment.

5 A method is described for predicting a "virtual" page update rate that can be used by web crawlers, web monitoring or other information gathering applications. The method leverages social networks to predict future update rates for a page by considering importance of users, their past updates to pages and past updates of similar pages by the social networks of those users.

10

The method targets users as the main source of content updates to web pages instead of the common traditional assumption that updates are made by some content management system (i.e., the server that manages the content).

15

Metadata about users' updates to web pages is used which is becoming more and more available (e.g., by extracting taggers' identities from Flickr pages) and metadata that is extracted from social network analysis (e.g., connections among different users that updated pages).

20

At any given point of time, for every page which information is needed to be gathered, two disjoint subsets of users are first identified: users that already have updated the page in the past and those who have not.

25

A page importance is determined using the metadata about the update rates of users that updated the page up to that time point (using the metadata of user update timestamps) to that page. Using these users relative importance in the social network (using the metadata about the users social network) its marginal contribution to the direct update rate of the page relative to the user's update rate and the user's social importance (i.e., authority) is calculated for each such user.

30

For every user that updated the page, the potential contribution to the page importance by indirect users that have connections to users that already updated the page is considered.

This potential contribution is calculated by looking at other pages that were updated by those indirect users and examining both the update rates in which these users update those pages and the level of similarity these pages have with the current page.

- 5 Therefore, if the page has many indirect users that have strong connection to users that already updated the page and are important users (determined from the social networks metadata) and have high update rates for similar pages to this page, this page will be considered as more important.
- 10 That means that the page has a higher predicted chance to be updated by other users that have not updated this page yet in the past but share high similarity with users that have already updated this page and probably since those users update pages that are similar to the current page, they might update also the page in the future.
- 15 The importance of each page is determined using the following four principles:
- A page that is updated more frequently is more important.
 - A page that is updated by more important users is more important.
 - A page has a potential to be updated in the future by users that are friends (or “followers”) of users that already updated this page in the past.
 - 20 • A user that updates other pages that are similar in their content to a page that hasn't been updated by this user in the past, is a potential user for updating this page in the future.

The four principles can be intuitively justified by the following scenarios:

- Pages that are updated more frequently need more crawls to maintain a fresh version (assuming that every single update to a page is important).
- 25 • A page that gets updated by a popular (or authoritative) user, may have a better chance to get other users reading the contribution of that important user and may be willing to contribute also content to that page, e.g., commenting about some expert user's blog posting.
- 30 • A social relationship between two users, having one user that already updated the page and the other user who didn't, may be used to imply how close is the last user to the first one with respect to their common interests. For example, a social network based on

interests on some topic X between a user that already updated the page and one that didn't may imply about the chance that the page that belong to the same topic X may be updated in the future by the last user.

• Users usually have a limited scope of interests (which define their user profile). The history of updates to pages by users that didn't update yet some page and their similarity of those pages to that page can imply how much chance that page has to be update also by those users.

A set of pages P is assumed that are updated over time by a set of users U that further construct a social network (SN).

For every page p in P and every time point t, two disjoint sets of users are identified with regard to page p:

- $U_t(p)$: subset of users in U that have updated pages p at least once up to time t.
- $\bar{U}_t(p) = U \setminus U_t(p)$: the subset of users in U that did not update page p up to time t.
- Note that over time, users shift from $U_t(p)$ to $\bar{U}_t(p)$, where initially $U_t(p)$ is empty for every page p in P.

For each user u in $U_t(p)$, the update rate of this user is measured to page p (denote: $\lambda_p^u(t)$) taken as the total number of updates of user u to page p up to time t divided by t (assuming uniform update rate).

Therefore, if $\lambda_p(t)$ denotes the update rate of page p at time t, then the general update rate of page p is given by: $\lambda_p(t) = \sum_{u \in U_t(p)} \lambda_p^u(t)$.

For every user in U, assume the availability of user u relative importance (or authority) in the social network at time t ($\omega_t(u)$).

Such authority can be calculated by known methods, for example, using methods described in one of: A. Hotho, R. Jaschke, C. Schmitz, G. Stumme, "FolkRank: A Ranking Algorithm for Folksonomies"; S. Chakrabati, "Dynamic Personalized PageRank in Entity-Relation

Graphs, WWW 2007; or T. Cheng, X. Yan, K.C.C. Chang, "EntityRank: Searching Entities Directly and Holistically", VLDB 2007.

Given a subset U' of users in U , further denote $\bar{\omega}_t(u, U') = \frac{\omega_t(u)}{\sum_{u \in U'} \omega_t(u)}$ the normalized
 5 authority (number in $[0,1]$) of user u relative to the subset U' .

Given two users u and u' in U , denote $\psi_t(u, u') = \text{dist}(u, u', SN)$ the degree of relatedness
 between the two users with respect to the social network SN generated by connections
 among the different users in U (in this case this relatedness is taken as the distance in terms
 10 of number of connections between users in the path from user u to user u' in the social
 network SN).

Given $\bar{U}_t(p)$, further denote $\bar{\psi}_{p,t}(u, u') = \frac{\psi_t(u, u')^{-1}}{\sum_{u' \in \bar{U}_t(p)} \psi_t(u, u')^{-1}}$ the relative relatedness of user u'
 to user u in the social network SN , normalized over all users in $\bar{U}_t(p)$. Further, take the
 15 power of -1 in $\psi_t(u, u')$ to denote that as the two users are more distant in the social network,
 the less is the relatedness degree between them (and therefore, the less this relatedness
 should be considered).

For every user u in U we keep the set of pages that user u has updated up to time t : $P_t(u)$.

For every two pages p and p' in P we denote $\text{sim}_t(p, p')$ the similarity between page p and
 page p' at time t .

Such page similarity can be taken, for example, in terms of content similarity between the
 25 two pages (e.g., using vector space similarity).

Given two pages p and p' in $P_t(u)$ of some user u in U , denote

$\overline{sim}_{u,t}(p, p') = \frac{sim_t(p, p')}{\sum_{p'' \in P_t(u)} sim_t(p, p'')}$ the normalized similarity between the two pages taken over all the pages in $P_t(u)$.

5 Determining a page importance for information gathering

Let $\phi_p(t)$ be the estimated importance of page p at time t .

At each time point t , calculate the importance of every page p in P , $\phi_p(t)$ using the following formula:

$$10 \quad \phi_p(t) = \sum_{u \in U_t(p)} \left(\overline{\omega}_t(u, U_t(p)) \times \lambda_p^u(t) + \sum_{u' \in \overline{U}_t(p)} \left(\overline{\psi}_{p,t}(u, u') \times \overline{\omega}_t(u', \overline{U}_t(p)) \times \sum_{p' \in P_t(u')} \lambda_{p'}^{u'}(t) \times \overline{sim}_{u',t}(p, p') \right) \right)$$

Therefore:

At time t , first consider the set of users in $U_t(p)$ that already updated page p up to time t .

Consider each user's u in $U_t(p)$ marginal contribution to the importance of page t by looking

15 at the combination update rate of user u to page p $\lambda_p^u(t)$ (which represent user's u marginal contribution to updates of p) and user's u relative importance (authority) $\overline{\omega}_t(u, U_t(p))$ over the set of users that updated page p . Therefore, an update of page p by a user that is more authoritative and/or contributes more frequently to the update rate of page p is considered more important than an update of a user that is less authoritative and/or contributes less

20 frequently to the update rate of page p .

For every direct user u that updated page p up to time t we further predict a potential increase in page p importance by indirect users u' in $\overline{U}_t(p)$ that did not update page p up to time t but have some degree of relatedness to user u (who did update page p).

25 For those users u' in $\overline{U}_t(p)$, further consider the combination of the relative degree of similarity between page p and other pages p' in $P_t(u')$ that were updated up to time t by u' , $\overline{sim}_{u',t}(p, p')$, and the update rate of u' to these pages $\lambda_{p'}^{u'}(t)$.

Therefore, $\sum_{p' \in P_t(u')} \lambda_{p'}^{u'}(t) \times \overline{sim}_{u',t}(p, p')$ presents a virtual predicted update rate of user u' to page p at some time after time t by considering similar pages to page p (up to some degree) and the rate of update of user u' to these pages.

- 5 Further multiply each such user u' future predicted contribution to page p importance in user u' relative importance among the users in $\overline{U}_t(p)$ and user u' relative degree of relatedness to user u $\overline{\psi}_{p,t}(u, u') \times \overline{\omega}_t(u', \overline{U}_t(p))$.

10 Therefore, a page that has users u that updated it and have stronger relationships (with respect to their social network) with other authoritative users u' that did not update page p yet but update similar pages very often, is predicted as a page that has a higher chance of getting update in the future by those indirect users u' , and therefore, is more important.

Policy for information gathering

15 Using the suggested method for determining a page importance for information gathering, several different policies can be constructed.

20 As an example, one can consider a greedy policy, where the order on which information is gathered from the different pages is determined by the importance of each page as given above. Therefore, given the set of pages in P and the current time is t , the next page revisit policy would be determined by sorting the pages in P relative to $\phi_p(t)$ and scheduling page crawls according to the resulting order.

25 As another example, one can consider to allocate page crawls according to the relative importance of page p $\phi_p(t)$ compared to other pages. Therefore, given that there are n pages in P and $m \ll n$ allocations for page crawl tasks in some time period T , each page p is

allocated with $m \times \frac{\phi_p(t)}{\sum_{p \in P} \phi_p(t)}$ page crawls.

Referring to Figure 5, a flow diagram 500 shows a method of crawling. The method starts by getting 501 the current available crawl budget. Next, determine 502 the order of crawls according to predicted object score and update potentials. Then perform crawls 503 on selected objects and extract 504 new content contribution and user identities for each updated object. Update 505 the importance of each object in the system and return to the start.

The described web information gathering method (i.e., to be used by web monitors or web crawlers) uses social network overlays in order to discover potential of new updates to different resources in the system according to similarity between different users in the system ("resource updaters") and the content that they contribute in the system. The identity of update sources are not hidden and for many systems those identities are of users that contribute content in the system. Therefore, such metadata is leveraged to improve existing web information gathering methods.

A prioritising system for network objects may be provided as a service to a customer over a network.

Embodiments of the invention can take the form of an entirely hardware embodiment, an entirely software embodiment or an embodiment containing both hardware and software elements. A preferred embodiment is implemented in software, which includes but is not limited to firmware, resident software, microcode, etc.

Embodiments of the invention can take the form of a computer program product accessible from a computer-usable or computer-readable medium providing program code for use by or in connection with a computer or any instruction execution system. For the purposes of this description, a computer usable or computer readable medium can be any apparatus that can contain, store, communicate, propagate, or transport the program for use by or in connection with the instruction execution system, apparatus or device.

The medium can be an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system (or apparatus or device) or a propagation medium. Examples of a

computer-readable medium include a semiconductor or solid state memory, magnetic tape, a removable computer diskette, a random access memory (RAM), a read only memory (ROM), a rigid magnetic disk and an optical disk. Current examples of optical disks include compact disk read only memory (CD-ROM), compact disk read/write (CD-R/W), and DVD.

5

Improvements and modifications can be made to the foregoing without departing from the scope of the present invention.

CLAIMS

1. A method for prioritising operations on network objects, comprising:
gathering available relationship data on the relationships between network entities,
5 wherein network entities are network users and network objects;
analysing the relationship data for a network entity;
determining for a network entity a first relative score based on the relationship data;
determining for a network object a second relative score which is a dynamic score
based on user interactions with the network object and formed using the first relative scores
10 of network entities interacting with the object; and
prioritising an operation on a network object using the second relative score;
wherein any of said steps are implemented in either of computer hardware or
computer software and embodied in a computer-readable medium.
- 15 2. The method of claim 1, wherein the available relationship data is Web 2.0 available
relationship data.
3. The method of claim 1 or 2, wherein the network entities are web entities including
web users and web objects.
- 20 4. The method of claim 1 or 2, wherein determining for a network object a second
relative score includes using the history of users' interactions with the object to provide the
second relative score.
- 25 5. The method of claim 1 or 2, wherein determining for a network object a second
relative score includes predicting users' interactions with the object to provide the second
relative score.
- 30 6. The method of claim 2, wherein the Web 2.0 available relationship data includes
metadata generated by the network users.

7. The method of claim 2, wherein Web 2.0 available relationship data includes: one or more of: social network data, folksonomy data, object authoring data, and object updating data.

5 8. The method of claim 1 or 2, wherein prioritising an operation on a network object using the second relative score prioritises caching of network objects.

9. The method of claim 7, wherein the second relative score is a dynamic score predicting the probability of future requests for cached network objects based on past users requests.
10

10. The method of claim 1 or 2, wherein prioritising an operation on a network object using the second relative score prioritises crawling of network objects.

11. The method of claim 10, wherein the second relative score is a dynamic score predicting where updates are likely to occur in network objects based on past user update patterns.
15

12. A system for prioritising operations on network objects, comprising:
20 a processor;
a collector for gathering available relationship data on the relationships between network entities, wherein network entities are network users and network objects;

an analyzer for analysing the relationship data for a network entity;
a module for determining for a network entity a first relative score based on the relationship data;
25

a module for determining for a network object a second relative score which is a dynamic score based on user interactions with the network object and formed using the first relative scores of network entities interacting with the object; and

a module for prioritising an operation on a network object using the second relative score;
30

wherein any of said collector, analyzer, and modules are implemented in either of computer hardware or computer software and embodied in a computer readable medium.

13. The system of claim 11, wherein the available relationship data is Web 2.0 available relationship data.

14. The system of claim 12 or 13, wherein the network entities are web entities including web users and web objects.

15. A computer program element comprising computer program code to, when loaded into a computer system and executed thereon, cause the computer to perform the steps of a method as claimed in any of claims 1 to 11.

1/5

FIG. 1

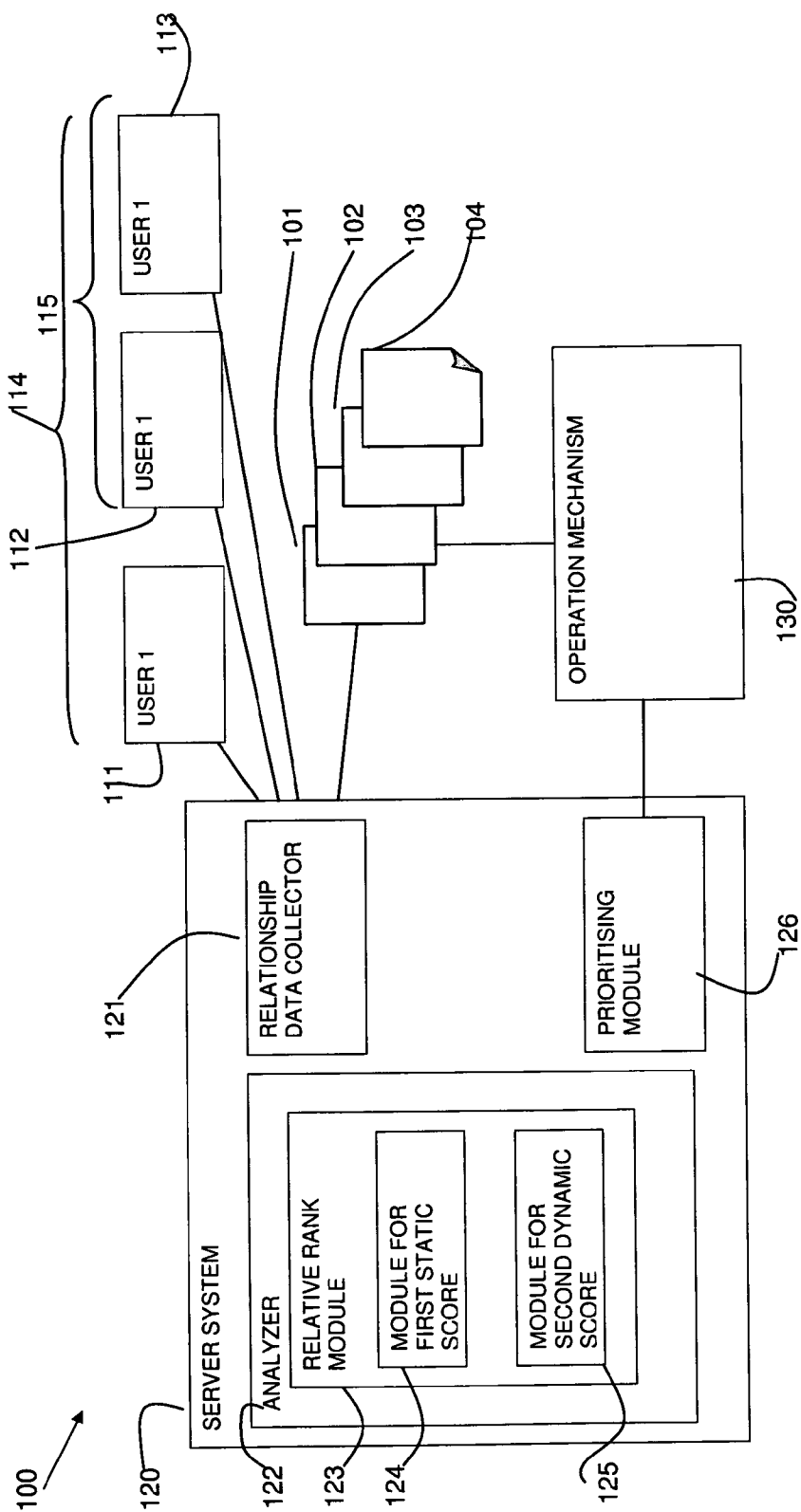
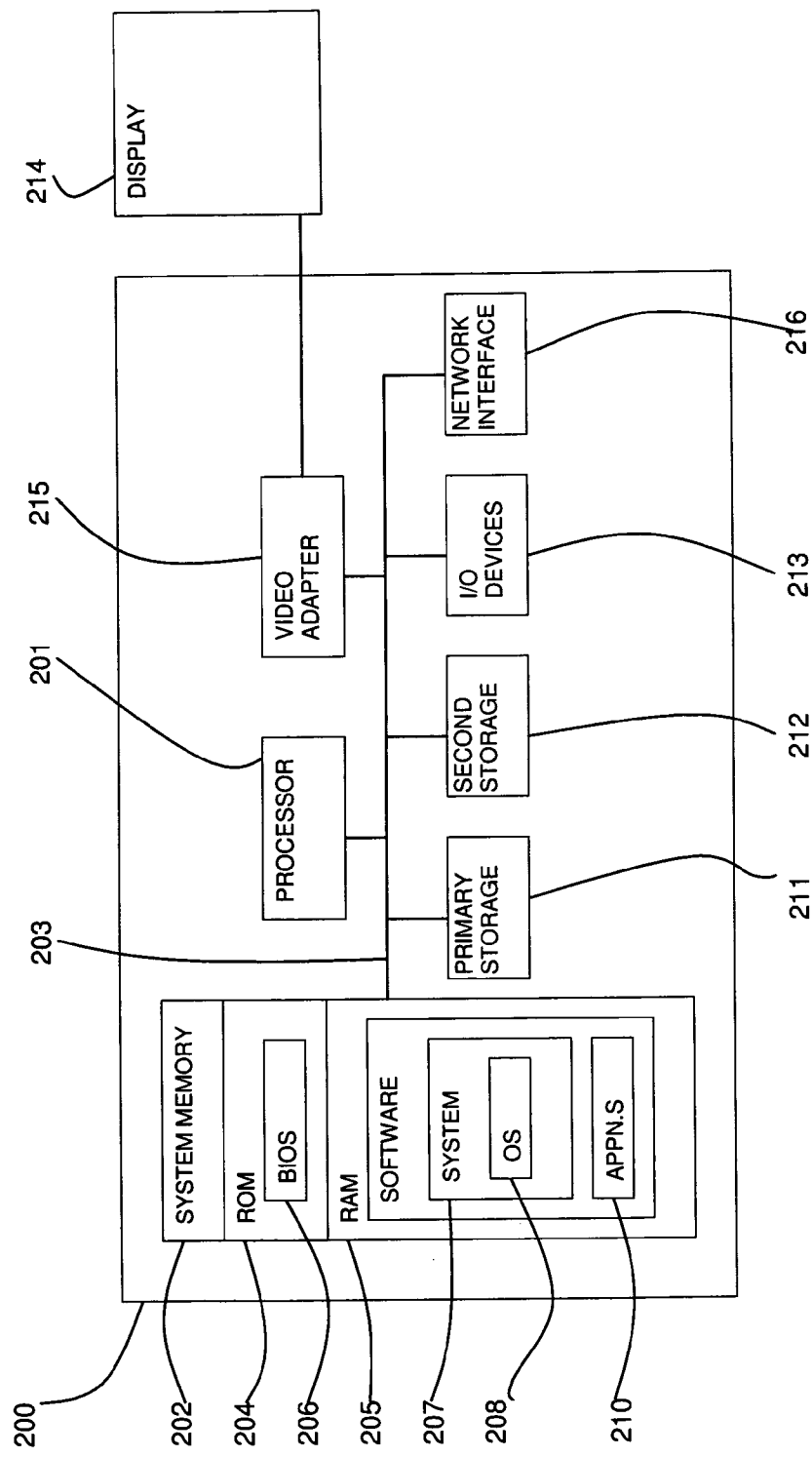
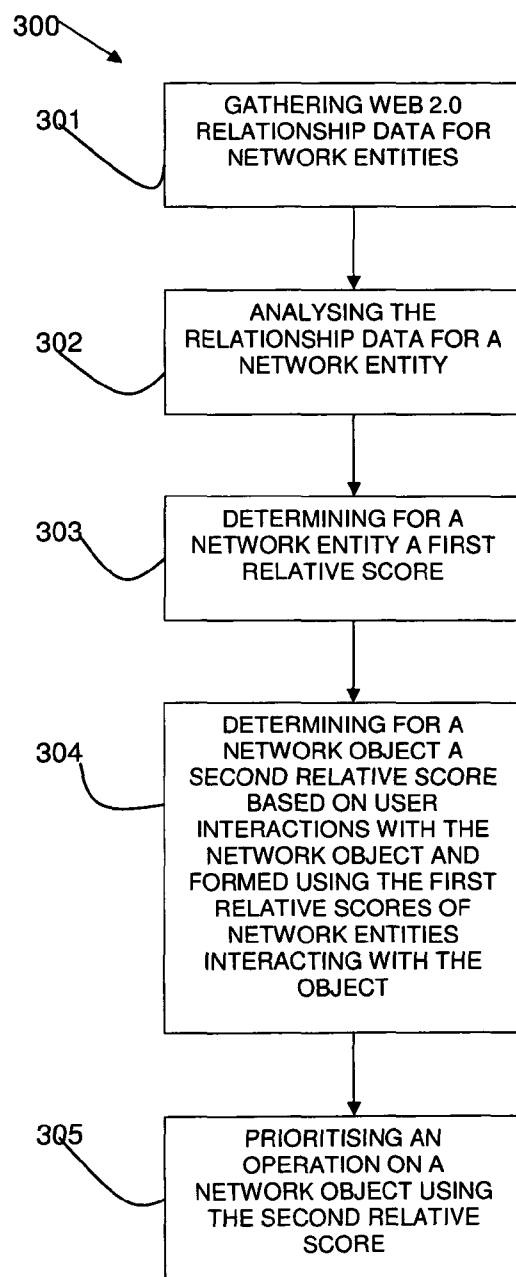


FIG. 2

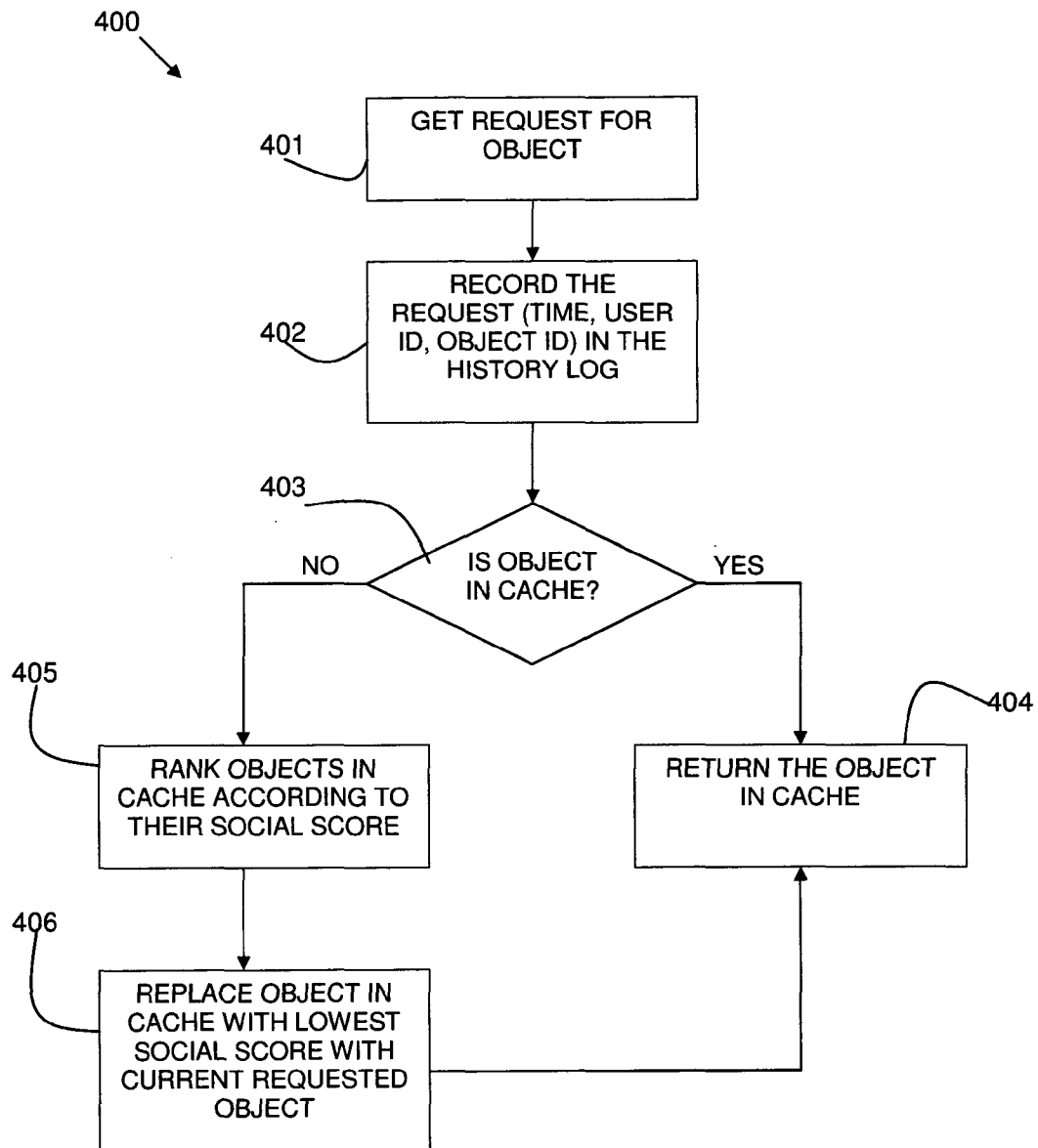


3/5

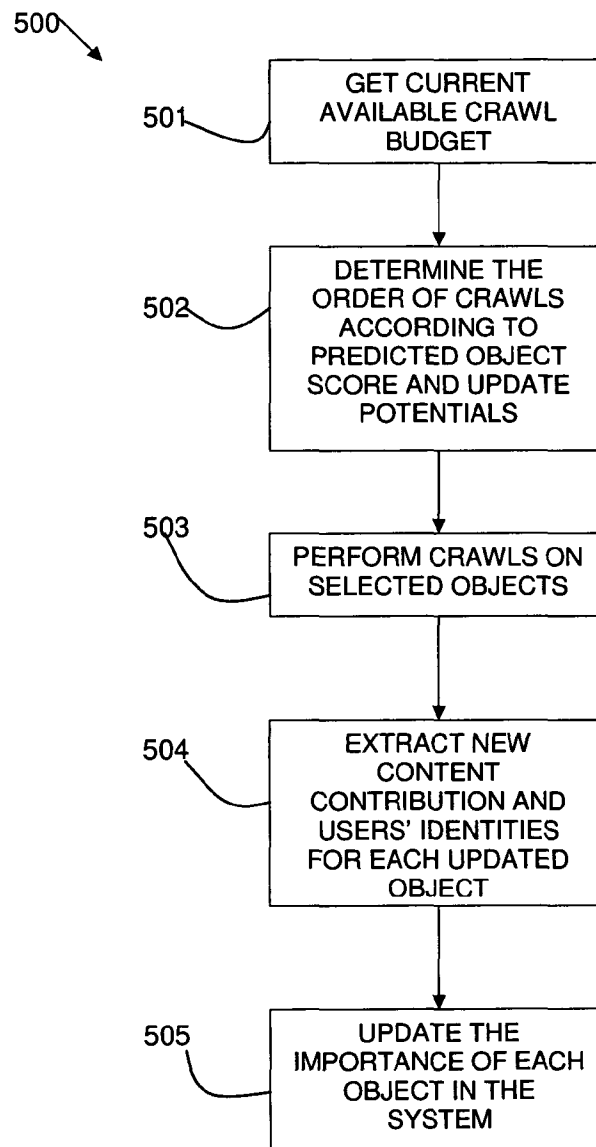
FIG. 3

4/5

FIG. 4



5/5

FIG. 5

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2010/055408

A. CLASSIFICATION OF SUBJECT MATTER

INV. G06F17/30
ADD.

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

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

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	<p>J. Cho, U. Schonfeld: "RankMass crawler: a crawler with high personalized pagerank coverage guarantee"</p> <p>28 September 2007 (2007-09-28), pages 375-386, XP002585597 Retrieved from the Internet: URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.111.1769&rep=rep1&type=pdf [retrieved on 2010-06-03] page 375 - page 377</p> <p style="text-align: center;">----- -/--</p>	1-15

☒ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier document but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

- "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- "&" document member of the same patent family

Date of the actual completion of the international search

4 June 2010

Date of mailing of the international search report

01/07/2010

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Dewyn, Torkild

INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2010/055408

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	<p>M. Richardson, A. Prakash, E. Brill: "Beyond PageRank: Machine Learning for Static Ranking"</p> <p>26 May 2006 (2006-05-26), pages 707-715, XP002585598 Retrieved from the Internet: URL:http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.104.3672&rep=rep1&type=pdf> [retrieved on 2010-06-03] page 707 - page 709</p>	1-15
X	<p>S. Pandey, C. Olston: "Crawl ordering by search impact"</p> <p>12 February 2008 (2008-02-12), XP002585599 Retrieved from the Internet: URL:http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.114.4342&rep=rep1&type=pdf> [retrieved on 2010-06-03] page 1 - page 3</p>	1-15
X	<p>S. Pandey, C. Olston: "User-centric web crawling"</p> <p>14 May 2005 (2005-05-14), pages 401-411, XP002585600 Retrieved from the Internet: URL:http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.59.6287&rep=rep1&type=pdf> [retrieved on 2010-06-03] page 401 - page 405</p>	1-15
A	<p>XIANGPING CHEN ET AL: "Lifetime behavior and its impact on Web caching" PROCEEDINGS 1999 IEEE WORKSHOP ON INTERNET APPLICATIONS (CAT. NO.PR00197) IEEE PISCATAWAY, NJ, USA LNKD- DOI:10.1109/WIAPP.1999.788017, 1999, pages 54-61, XP002585601 ISBN: 0-7695-0197-4 the whole document</p>	1-15

-/--

INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2010/055408

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>Terence P. Kelly, Sugih Jamin, Jeffrey K. MacKie-Mason: "Variable QoS from shared Web caches: User-centered design and value-sensitive replacement"</p> <p>3 December 1999 (1999-12-03), pages 1-14, XP002585602</p> <p>Retrieved from the Internet: URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.32.9386&rep=rep1&type=pdf [retrieved on 2010-06-03] the whole document</p> <p>-----</p>	1-15
A	<p>US 2008/010281 A1 (BERKHIN PAVEL [US] ET AL) 10 January 2008 (2008-01-10) the whole document</p> <p>-----</p>	1-15

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2010/055408

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2008010281 A1	10-01-2008	US 2010023513 A1	28-01-2010