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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
15 July 2010 (15.07.2010)

PCT

(10) International Publication Number
WO 2010/080284 A2

(51) **International Patent Classification**:
G06Q 50/00 (2006.01) *G06F 17/30* (2006.01)

(21) **International Application Number**:
PCT/US2009/067033

(22) **International Filing Date**:
7 December 2009 (07.12.2009)

(25) **Filing Language**: English

(26) **Publication Language**: English

(30) **Priority Data**:
12/338,117 18 December 2008 (18.12.2008) US

(71) **Applicant** *(for all designated States except US)*: **YAHOO! INC.** [US/US]; 701 First Avenue, Sunnyvale, California 94089 (US).

(72) **Inventors; and**

(75) **Inventors/Applicants** *(for US only)*: **BAEZA-YATES, Ricardo** [CL/ES]; Ocata 1, E-08003 Barcelona (ES). **GIONIS, Aristides** [GR/ES]; Av. Diagonal 177, E-08018 Barcelona (ES). **JUNQUEIRA, Flavio** [BR/ES]; Av. Diagonal 177, 8th Floor, E-08018 Barcelona (ES). **PLACHOURAS, Vassilis** [GR/ES]; Carrer Sevilla 14 2-1, E-08003 Barcelona (ES). **TELLOLI, Luca** [IT/ES]; Ocata 1, E-08003 Barcelona (ES).

(74) **Agents**: **MIKHAIL, Peter, G.** et al.; Weaver Austin Villeneuve & Sampson LLP, P.O. Box 70250, Oakland, California 94612-0250 (US).

(81) **Designated States** *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, 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, 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, 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,

[Continued on next page]

(54) **Title**: SEARCH ENGINE DESIGN AND COMPUTATIONAL COST ANALYSIS

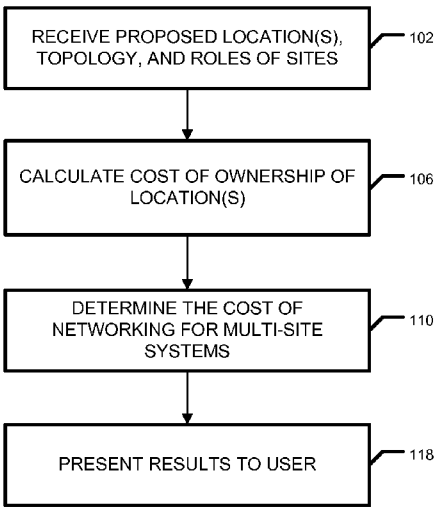

FIG. 1

(57) **Abstract**: A computer implemented system for search engine facility architecting and design. The system estimates the costs of power and networking based on system parameters, such as average CPU utilization, connection time, and bytes transferred over the network. Regional distribution of facilities may be evaluated to take into account the various parameters and optimize the cost and speed of the systems being designed. The parameters used in analyzing and formulating an architecture are independent of a particular indexing or query processing technique.

WO 2010/080284 A2

TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

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

SEARCH ENGINE DESIGN AND COMPUTATIONAL COST ANALYSIS

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims benefit of and priority to USSN: 12/338,117, filed on December 18, 2008, which is incorporated herein by reference in its entirety for all purposes.

BACKGROUND OF THE INVENTION

[0002] This invention relates generally to search engines and queries.

[0003] Search engines use a large number of servers to perform tasks going from crawling, through indexing, and query processing. Centralized solutions are beneficial when the capacity of the system is not required to grow or grows slowly. However, centralized solutions provide limited scalability: the system can only grow to the extent allowed by the initial design of the data center hosting the system.

[0004] A better understanding of the costs associated with centralized and distributed architectures is necessary to efficiently plan and operate search facilities.

SUMMARY OF THE INVENTION

[0005] Embodiments of the invention estimate the costs of power and networking based on system parameters, such as average CPU utilization, connection time, and bytes transferred over the network. Regional distribution of facilities may be evaluated to take into account the various parameters and optimize the cost and speed of the systems being designed. The parameters used in analyzing and formulating a search system architecture are independent of a particular indexing or query processing technique.

[0006] One embodiment relates to a computer system configured to: receive a target query volume; calculate the cost of operation for a proposed distributed search system comprising at least one search repository site geographically distant from a second search repository site; calculate the cost of networking the search repository sites of the distributed search system; calculate the cost of operation for a proposed centralized search system; and determine whether the cost of operation of the proposed distributed system is greater or less than the cost of operation of the proposed centralized system. Similarly, the system can also calculate and compare the costs of different distributed systems and determine the relative costs of the different distributed systems

[0007] Another embodiment relates to a computer program product, comprising a computer usable medium having a computer readable program code embodied therein. The computer readable program code is adapted to be executed to implement a method for designing a search engine system. The method comprises: determining a sum of power costs for at least two designs; determining a sum of bandwidth costs for the at least two designs, and determining an optimal number of nodes for the search engine system. The method may be used to compare the cost of different distributed architectures with a different number of nodes from the other, or the cost of designs with the same number of nodes, but with different networking topologies.

[0008] Another embodiment relates to a computer program product, comprising a computer usable medium having a computer readable program code embodied therein. The computer readable program code is adapted to be executed to implement a method for designing a search engine system. The method comprises: establishing a target latency for queries of a search processing system that services queries from a first geographic area and a second geographic area distant from the first geographic

area; receiving a proposed topology for the search processing system; receiving a proposed location for a first site to service queries of the first and second geographic areas; receiving a proposed location for a second site to service queries of the first and second geographic areas, the first site being geographically distant from the second site; determining a power cost for power consumption of the first site by estimating power consumption of crawling operations of the first site; determining a power cost for power consumption of the first site by estimating power consumption of query processing operations of the first site; determining a power cost for power consumption of the second site by estimating power consumption of crawling operations of the second site; determining a power cost for power consumption of the second site by estimating power consumption of query processing operations of the second site; and calculating an overall operating cost of the search processing system from the power costs given the target latency, geographic areas to be served, proposed topology and locations.

[0009] A further understanding of the nature and advantages of the present invention may be realized by reference to the remaining portions of the specification and the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] FIG. 1 is a flow chart of a method according to an embodiment of the invention.

[0011] FIGS. 2 and 3 are graphs illustrating examples of the cost of processing with a distributed architecture.

[0012] FIG. 4 is a simplified diagram of a computing environment in which embodiments of the invention may be implemented.

[0013] A further understanding of the nature and advantages of the present invention may be realized by reference to the remaining portions of the specification and the drawings.

DETAILED DESCRIPTION OF SPECIFIC EMBODIMENTS

[0014] Reference will now be made in detail to specific embodiments of the invention including the best modes contemplated by the inventors for carrying out the invention. Examples of these specific embodiments are illustrated in the accompanying drawings. While the invention is described in conjunction with these specific embodiments, it will be understood that it is not intended to limit the invention to the described embodiments. On the contrary, it is intended to cover alternatives, modifications, and equivalents as may be included within the spirit and scope of the invention as defined by the appended claims. In the following description, specific details are set forth in order to provide a thorough understanding of the present invention. The present invention may be practiced without some or all of these specific details. In addition, well known features may not have been described in detail to avoid unnecessarily obscuring the invention.

[0015] Distributed architectures for search engines address issues with the scalability problem of centralized Web retrieval. As the data centers that host servers for a search engine have limited capacity, it is beneficial to have a system design that can cope with the growth of the Web, and that is not constrained by the physical limitations of a data center.

[0016] A typical solution to this design problem is to use a single, centralized site, since it is a simple and competitive solution, and to locate such a system in the place that provides the lowest cost of operation and the maximum benefit. Such a

preference for a centralized solution often comes from a lack of understating of the benefits and drawbacks of a distributed solution. In fact, it is intuitively unclear whether the benefits of a distributed architecture compensate for the extra communication costs between the physical locations. An example of an important benefit of a distributed solution is the proximity between the engine machinery to data and users. Being closer to data implies that the system requires fewer machines to perform the same crawling, as the Web connections are shorter and the data transfer are faster. For the same reason fewer front end servers are necessary to handle the same query volume due to the faster service time. Embodiments of the present invention create a physical model and detailed cost analysis, allowing potential architectures to be analyzed and the cost-benefit ratio to be determined.

[0017] In general, as the overall workload is distributed, the cost of handling network bandwidth saturation, redundancy, and fault tolerance may also decrease. A distributed architecture also enables the service to exploit the potential local properties of the workload. First, locality implies lower utilization of the network, and thus, reduces the communication cost. Second, locality of queries may imply better local customization, since teams of developers can use local expertise to tailor services to local preferences, thus improving the user experience and increasing the advertising revenue.

[0018] Distributed solutions designed and evaluated with embodiments of the present invention are able to process a significant fraction of the queries locally. In practice, achieving the goal of processing all queries locally is difficult. More than one site might need to be used to process some of the submitted queries, hereinafter called non-local queries. The additional communication cost increases the total latency of query processing, and hence the latency for non-local queries is higher. On the other

hand, local queries are processed faster. Local queries are those queries that can be processed by the site to which they are submitted. Locality refers to the fraction of the volume of queries that are local. Thus, if a relatively high percentage of queries are processed locally, then the average latency will be reduced.

[0019] In addition to locality, another factor is the volume of queries for which the distributed system retrieves more or fewer clicked documents than a centralized system, assuming that a click by a user on a retrieved document is an indication of relevance.

[0020] An example of a practical distributed architecture is a star topology. Such a topology has a minimal number of connections and requires only two hops between any pair of sites. The main drawback of this architecture is having to provision the center site in such a way that it can handle more traffic compared to other sites. That is, building and maintaining the center site is more costly. A central, more provisioned site, however, turns out to have advantageous aspects including that the central site may handle a significant fraction of the queries that are not processed locally. Moreover, this site may be located in the region with the highest query traffic and therefore benefit from a larger, well-provisioned site. The organization of the sites does not need to be flat, and sites can have special roles. For instance, embodiments of the system can organize them hierarchically with the sites having distinct roles. The optimal network topology to use is also part of the design process/parameters in analyzing distributed system architecture. For a collection of documents D over a set of terms T, the documents D are partitioned into two subsets: local (L) and global (G). Global documents are present in all sites, whereas local documents are further partitioned disjointly among the sites of S.

[0021] FIG. 1 is a flow chart, depicting, at a high level, a method of designing and evaluating search engine systems. In step 102, the system receives proposed location(s), topology, and roles of the sites. Then in step 106, the system calculates the cost of ownership of each of the location(s). In a preferred embodiment, the cost of ownership is primarily based upon the power consumption, although other factors may be taken into account, as discussed below. In determining the power consumption, many factors may be taken into account. For example, the number of operations per second that are needed, the number of servers needed for crawling, the number of servers needed for query processing, the CPU utilization, and target latency.

[0022] The cost of a data center is the sum of its initial cost and the cost of operating it over some period of time. The initial cost varies significantly, depending on factors such as the design choices (raised floor, server density, etc.), location and the value of local labor. This cost is usually amortized over the lifetime of the data center. Operational costs also vary significantly, and depend on factors such as power consumption, amount of network bandwidth, and maintenance costs. The described embodiments focus upon on the operational costs, and more specifically upon power consumption and network utilization. Power consumption and related expenses typically represent more than 60% of the cost in the lifetime of a data center. For more information, please refer to a paper from American Power Conversion entitled "Determining total cost of ownership for data center and network room infrastructure: White Paper #6," available at, http://www.apcmedia.com/salestools/CMRP-5T9PQG_R3_EN.pdf, 2005.

[0023] The cost of a multi-site system is the sum of the individual costs of each site over some period of time. To build a site there is an initial cost (Init), which consists

of setting up all the infrastructure necessary to host servers, network equipment, and to operate the data center. Once the data center is operating, there is the cost of maintaining it, known as cost of ownership. As we mentioned before, the cost of ownership may be represented here by the power consumption, and we use $Own(\Delta t)$ to denote the cost of ownership for the whole system for a period of time Δt. We also use $W(t, i)$ to denote the power consumption of site S_i consumed at time t, and $C_w(\Delta t, i)$ to be the cost of power consumption for site S_i over time Δt.

$$\begin{aligned} Cost(\Delta t) &= Init + Own(\Delta t) \\ Own(\Delta t) &= Own'(\Delta t) + \sum_i C_w(\Delta t, i) \end{aligned}$$

[0024] where $Own'(\Delta t)$ corresponds to all the costs other than power, and the cost of power is given by the amount of power used in watts multiplied by the cost per watt. We compute the cost of power from the power consumption of a site:

$$C_w(\Delta t, i) = \left(\int_{t_1}^{t_2} W(t, i) \cdot dt\right) \cdot u_w, \ \Delta t = t_2 - t_1$$

[0025] To account for different functionality, we further split the power cost into different classes, according to the functionalities of the system: $W(t, i) = \sum_f W_f(t, i)$, where f is a functionality of the system, such as crawling and query processing. To estimate the power consumption of each function, we use the following:

$$W_f(t, i) = TOPS(i) \cdot \frac{\ell_f(i)}{c_f(i)} \cdot e_f(t, i)$$

[0026] where TOPS(i) is the target number of operations per second (e.g., queries processed, Web pages fetched) that site S_i performs at time t; $\ell f(i)$ is the target latency to perform an operation at site S_i; $c_f(i)$ is the capacity in number of simultaneous

operations for a server or a cluster, depending on the functionality f; $e_f(t, i)$ estimates the power consumption per server or cluster at time t. To estimate such a value, CPU utilization is used, as described in detail in a paper by X. Fan, W.-D. Weber, and L. A. Barroso, entitled "Power provisioning for a warehouse-sized computer," *In Proceedings of the 34th International Symposium on Computer Architecture*, pages 13–23, 2007 (which is hereby incorporated by reference in the entirety):

$$e_f(t, i) = m_i \cdot (W_{\text{idle}} + (W_{\text{busy}} - W_{\text{idle}}) \cdot cpu(OPS(t, i)) \quad (1)$$

[0027] where m_i is the size of a group of servers, W_{idle} is the power utilization of a server when the CPU is idle, W_{busy} is the power utilization of a server when the CPU is busy, and $cpu(OPS(t, i))$ evaluates to the CPU utilization of a server at time t in site S_i. Note that the CPU utilization is a function of the workload at time t given by $OPS(t, i)$.

[0028] We use $TOPS(i)$, $\ell_f(i)$, and $c_f(i)$ to estimate the number of servers or clusters necessary for a particular function. We use a server when the processing unit is a server. For example, for crawling, we assume that each server crawls individually. For query processing, however, we assume that the processing unit is a cluster because typically systems use document or term partition to increase parallelism when processing a query. Although both document and term partition can potentially cause load imbalance across the servers of a cluster, we do not address such issues here, and simply assume that $e_f(t, i)$ evaluates to the total amount of power used at time t. In practice, the values of $TOPS(i)$, $\ell_f(i)$, and $c_f(i)$ can be estimated from demand. For example, through experimentation, practitioners can determine that a given cluster of machines is able to process simultaneously $c_f(i)$ operations keeping the average latency at $\ell_f(i)$, and estimate that the total traffic of a site will be on average $TOPS(i)$.

Also note that $e_f(t, i)$ implicitly introduces the current traffic, since the amount of watts depends upon the current traffic.

[0029] Specializing equation $W_f(t, i)$ to crawling and query processing, we have the following:

$$W_c(t,i) = TPPS(i) \cdot \frac{\ell_c(i)}{c_c(i)} \cdot e_c(t,i)$$

$$W_q(t,i) = TQPS(i) \cdot \frac{\ell_q(i)}{c_q(i)} \cdot e_q(t,i)$$

[0030] The rationale for the above equations is the following. For crawling, a server at site S_i can only have a given number of connections open at a time given by $c_c(i)$. Given the number of pages $TPPS(i)$ crawled and the average amount of time to fetch a page $\ell_c(i)$, we determine the total number of servers necessary to crawl. By multiplying by the average amount of power a server uses, we determine the total amount of power necessary for crawling at site S_i. For query processing, we have a similar derivation. To estimate the total amount of power, we multiply the total number of servers in a query processing cluster and the average amount of power a server uses according to Equation 1. To determine the total number of clusters, we estimate the target arrival rate of queries ($TQPS(i)$) and divide by the number of queries per second a cluster can process ($c_q(i)/\ell_q(i)$). There are different ways to determine the number of servers per cluster. For example, we fix a fraction of the index, and each server holds such a fraction. Note that while equation $W_f(t, i)$ may also be specialized to cover indexing operations, although the general equation already includes the cost of indexing functions.

Adding the cost of networking

[0031] In a multi-site system, the cost of networking between the sites is determined in step 114. As the rates of network circuits and services vary considerably, the system estimates the cost using the total number of bytes that we need to transfer over a period of time, using a function that converts such a requirement for bandwidth into currency. Typically, the cost of bits per sec (bps) decreases as the total amount of aggregated bandwidth increases. That is, the price of bandwidth often increases sublinearly with the bandwidth contracted. We then assume that the cost of bandwidth $C_{bw}(t, i)$ is a function of the total number of bytes that site S_i transfers at time t. The total cost then becomes:

$$\begin{aligned} Cost(\Delta t) &= Init + Own(\Delta t) + C_{bw}(\Delta t) \\ C_{bw}(\Delta t) &= \sum_i C_{bw}(\Delta t, i) \end{aligned}$$

[0032] Latency increases linearly with round-trip time. Longer connections reduce the throughput of crawlers, as their capacity is often given by the total number of simultaneous connections. Having longer connections thus implies fewer requests per second for each server. Front-end servers, which host Web servers that interact with users, also have a similar issue: longer connections imply fewer user requests for each server. Thus, one of the benefits of having sites closer to users is reducing the impact of round trip travel on the cost of search.

[0033] In step 118, the system finally presents the results of the above analysis to the user.

[0034] Embodiments assess the feasibility of distributed Web search engines comprising sites that correspond to different geographical locations. A computer system is utilized to develop cost models and evaluate operational costs.

Embodiments may include a general purpose computer or a special purpose computer. In one embodiment a special purpose computer system typically used to perform searches may be used to develop the architectural and cost models described herein. This is beneficial in that certain search parameters utilized can also be evaluated by the system, in some cases in an iterative fashion. Such a computer system is illustrated in FIG. 4. This is represented in FIG. 4 by server 408 and data store 410 which, as will be understood, may correspond to multiple distributed devices and data stores. The invention may also be practiced in a wide variety of network environments including, for example, TCP/IP-based networks, telecommunications networks, wireless networks, public networks, private networks, various combinations of these, etc. Such networks, as well as the potentially distributed nature of some implementations, are represented by network 412, and devices 401, 402, 403, 404 and 406.

[0035] In addition, the computer program instructions with which embodiments of the invention are implemented may be stored in any type of tangible computer-readable media, and may be executed according to a variety of computing models including a client/server model, a peer-to-peer model, on a stand-alone computing device, or according to a distributed computing model in which various of the functionalities described herein may be effected or employed at different locations.

Examples

[0036] To illustrate how embodiments enable the assessment of distributed architectures, we use two simple examples to demonstrate the potential savings with crawling and query processing in a multi-site engine. Note that while the examples demonstrate the potential savings in crawling and query processing, such savings are

equally applicable for indexing operations, and that embodiments of the invention also factor in indexing operations.

Crawling

[0037] Suppose we have two systems:

[0038] System 1: System 1 has one site S11, and its Web collection comprises P pages;

[0039] System 2: System 2 has five sites $\{S_{j2} : j \in \{1,2,3,4,5\}\}$. The Web collection of site S_{12} comprises αP pages, $1 > \alpha > 0.2$, and the other sites maintain $P(1-\alpha)/4$ pages each. Site S_{12} has the role of a central site, with more computing power than the others.

[0040] We use $Wc_i(t,j)$ to denote $W_c(t,j)$ for system i, and $\ell c_i(j)$ to denote $\ell_c(j)$ for system i. We then have that the power consumption to crawl all P pages with System 1 at a rate $p_r = P/\Delta t$, Δt being an interval of choice, is:

$$W_1(t) = W_{c_1}(t,1) = p_r \cdot X \cdot \ell_{c_1}(1)$$

[0041] where X represents the computation of all other variables. For simplicity, we assume that the power utilization is the same for all servers across all sites.

[0042] With System 2, we have the following:

$$W_2(t) = p_r \cdot X \cdot \alpha \cdot \ell_{c_2}(1) + \sum_{i=2,3,4,5} p_r \cdot X \cdot \frac{1-\alpha}{4} \cdot \ell_{c_2}(i)$$

[0043] For the sake of simplicity, we assume that System 2 has been designed in such a way that $\ell c_2(i)$ is the same for all $i \in \{2,3,4,5\}$ and equal to $l_o, l_o < \ell_{c_1}(1)$. We

have that the difference is

$$W_1(t) - W_2(t) = p_e \cdot X \cdot (\ell_{e_1}(1) - \alpha \cdot \ell_{e_2}(1) - (1 - \alpha) \cdot \ell_e),$$

[0044] and $\ell_{e_1}(1) > \ell_{e_2}(i)$, for $i \in \{1, 2, 3, 4, 5\}$ and $\alpha > 0$, we have that

$W_1(t) - W_2(t) > 0$.

[0045] As the latency of fetching pages is reduced, the power consumption of servers used for crawling is also reduced. Note that this simple computation does not include potential costs that might arise from having to communicate crawlers in different sites. It does show, though, that a crawler distributed across a number of sites, and that requires negligible communication among crawlers in different sites, is cheaper compared to a centralized one.

Query processing

[0046] This example illustrates how embodiments determine the cost changes with the number of sites. This example refers to a fully connected topology where every site is connected to every other site, just one example topology that embodiments of may assess. We assume a fully-distributed system in which there are n sites. Users submit queries to the closest site, and the site either processes them locally, or it sends them all other sites. A user request is therefore classified as either local or global, depending on the sites that process the query. Site S_i is able to resolve a query it receives from a user with probability x_i. In this example, we assume that x_i is the same across all sites, and we use x to denote the fraction of the total query volume resolved locally.

[0047] Following the earlier described cost model, we have that the cost is the sum of power costs and bandwidth costs, ignoring initial costs and remaining costs of

ownership. As each site processes a fraction x of the query traffic received locally, and the remainder is processed by all other sites, we have:

$$\begin{aligned} W_q(t) &= \sum_i W_q(t,i) \\ &= \left(\sum_i (q_i + \sum_{j:j\neq i} q_j \cdot T_{ji})\right) \cdot \frac{\ell(n)}{c} \cdot e_q \\ &= \left(QPS \cdot (x + (1-x)\cdot n) \cdot \frac{\ell(n)}{c} \cdot e_q\right) \end{aligned}$$

where:

- $q_i + \sum_{j:j\neq i} q_j \cdot T_{ji} = TQPS(i)$, for all i;
- q_i is the number of queries per second that users submit directly to site S_i, and $QPS = \sum_i q_i$;
- T_{ij} is the fraction of queries that site S_i sends to site S_j for processing;
- $\ell(n)$ is the latency to process a query. We assume that it decreases with the number of sites such that $\ell(n) = k/n$, where k is a constant representing the time to process a query in a single-site system (DQ principle

- c is the capacity of a query cluster. We assume that it is constant across sites and independent of the number of sites;
- e_q is the number of watts that query processors consume. For simplicity, we assume that $e_q(t,i) = e_q$ for all t and i;
- U_w is the cost of energy given in dollars per watt-hour (Wh).

[0048] Note that $W_q(t)$ is a value independent of t in this case, and therefore W_q is used instead. The cost of power considering only the cost of query processing is:

$$\begin{aligned} C_w(\Delta t) &= \left(\int_{t_1}^{t_2} W_q \cdot dt\right) \cdot U_w, \Delta t = t_2 - t_1 \\ &= W_q \cdot \Delta t \cdot U_w \end{aligned}$$

[0049] and to make the units compatible, we have to convert $W_q \cdot \Delta t$ from joules to watt-hour by dividing it by 3600, and we finally have:

$$\begin{aligned} C_w(\Delta t) &= \frac{W_q \cdot \Delta t}{3600} \cdot U_w \\ &= W_q \cdot 720 \cdot U_w \end{aligned}$$

given in dollars and assuming that $\Delta t = 30 \cdot 24 \cdot 3600$ (one month in seconds). The amount of traffic increases linearly with the number of global queries, and with the number of sites. The cost of network bandwidth is thus represented as follows:

$$C_{bw}(\Delta t) = \sum_i C_{bw}(\Delta t, i) = \left(\sum_{i,j:j \neq i} q_j \cdot T_{ji} \cdot b \right) \cdot \Delta t \cdot U_{bw}$$

where b is the average number of bits for each request; U_{bw} is the cost of bandwidth in dollars per Mbps per month ; and Δt is time in number of months. For this particular example, we have that $T_{ji} = (1 - x)$, $q_j = \frac{QPS}{n}$, and $\Delta t = 1$ month:

$$\begin{aligned} C_{bw}(\Delta t) &= \left(\sum_{i,j:j \neq i} \frac{QPS}{n} \cdot (1 - x) \cdot b \right) \cdot U_{bw} \\ &= (QPS \cdot (1 - x) \cdot (n - 1) \cdot b) \cdot U_{bw} \end{aligned}$$

Adding the terms, we have that the total cost is given by the following:

$$\begin{aligned} Cost(1 \text{ month}) &= C_w(1 \text{ month}) + C_{bw}(1 \text{ month}) \\ &= QPS \cdot (U_w \cdot 720 \cdot (x + (1 - x) \cdot n) \cdot \frac{\mathcal{E}(n)}{c} \cdot \\ &\quad e_q + U_{bw} \cdot (1 - x) \cdot (n - 1) \cdot b) \end{aligned}$$

[0050] FIGS. 2 and 3 illustrate *Cost*(*t*), assuming that *QPS* = 1 (cost of one query per second). They show how the cost varies for different fractions of locality *x*, assuming that U_w/U_{bw} is 0.1 *Mbps·month/KWh*, and 0.01 *Mbps·month/KWh*, respectively. A centralized architecture corresponds to the point with value $n = 1$. From the figures, if the cost of bandwidth is low enough, then making the engine distributed has a lower overall cost. As we increase the cost of bandwidth, we observe that the cost of a distributed architecture becomes higher, and at some point for no value of the locality parameter a distributed engine has lower costs. In fact, the optimal number of nodes is $C_n \cdot \left(\sqrt{\frac{U_w}{U_{bw}} \frac{1}{1-x}} \right)$, where C_n is a normalization constant that cancels out the unit of $\frac{U_w}{U_{bw}}$ and can be computed from the formula above. Hence, the optimal number grows when locality increases and when the fraction U_w/U_{bw} increases. That is, for small

relative values of the bandwidth cost, such as $U_w/U_{bw} = 0.1$ *Mbps·month/KWh*, it is observed that for all values of the locality parameter there is a number of sites for which the cost is lower. For larger differences in the cost per unit of power and bandwidth, such as $U_w/U_{bw} = 0.01$ *Mbps·month/KWh*, we have that for some values of the locality parameter the cost of a distributed architecture is never lower compared to a centralized architecture. This is because the cost of networking dominates the total cost of the system for such values.

[0051] While the invention has been particularly shown and described with reference to specific embodiments thereof, it will be understood by those skilled in the art that changes in the form and details of the disclosed embodiments may be made without departing from the spirit or scope of the invention.

[0052] In addition, although various advantages, aspects, and objects of the present invention have been discussed herein with reference to various embodiments, it will be understood that the scope of the invention should not be limited by reference to such advantages, aspects, and objects. Rather, the scope of the invention should be determined with reference to the appended claims.

What is claimed is:

1. A computer program product, comprising a computer usable medium having a computer readable program code embodied therein, said computer readable program code adapted to be executed to implement a method for designing a search engine system, said method comprising:

establishing a target latency for queries of a search processing system that services queries from a first geographic area and a second geographic area distant from the first geographic area;

receiving a proposed topology for the search processing system;

receiving a proposed location for a first site to service queries of the first and second geographic areas;

receiving a proposed location for a second site to service queries of the first and second geographic areas, the first site being geographically distant from the second site;

determining a power cost for power consumption of the first site by estimating power consumption of crawling operations of the first site;

determining a power cost for power consumption of the first site by estimating power consumption of query processing operations of the first site;

determining a power cost for power consumption of the second site by estimating power consumption of crawling operations of the second site;

determining a power cost for power consumption of the second site by estimating power consumption of query processing operations of the second site; and

calculating an overall operating cost of the search processing system from the power costs given the target latency, geographic areas to be served, proposed topology and locations.

2. The computer program product of claim 1, wherein determining the power cost for operations of the first and second site comprises:

computing the target number of operations per second that each site performs;

determining a ratio of the target latency to the number of simultaneous operations for a server or cluster; and

determining the power consumption per server or cluster.

3. A computer system configured to:

receive a target query volume;

calculate the cost of operation for a proposed distributed search system comprising at least one search repository site geographically distant from a second search repository site;

calculate the cost of networking the search repository sites of the distributed search system;

calculate the cost of operation for a proposed centralized search system; and

determine whether the cost of operation of the proposed distributed system is greater or less than the cost of operation of the proposed centralized system.

4. The system of claim 3, wherein in order to calculate the cost of operation the system is configured to:

determine the functionality of each site of the distributed system; and

compute the cost of power for each site based upon the functionality of the site and the power consumption of the site.

5. The system of claim 4, wherein in order to compute the cost of power for each site the system is configured to:

(a) Compute the target number of operations per second that each site performs;

(b) Determine a ratio of the target latency to the number of simultaneous operations for a server or cluster;

(c) determine the power consumption per server or cluster; and

(d) multiply (a) (b) and (c).

6. The system of claim 3, wherein in order to calculate the cost of operation the system is configured to factor in the latency requirements of the distributed search system and the centralized search system.

7. The system of claim 6, wherein in order to factor in the latency requirements and calculate the cost of operation the system is configured to determine a redundancy of servers necessary for the distributed search system.

8. The system of claim 7, wherein in order to factor in the latency requirements and calculate the cost of operation the system is configured to determine a redundancy of servers necessary for the centralized search system.

9. The system of claim 6, wherein in order to factor in the latency requirements and calculate the cost of operation the system is configured to determine a redundancy of bandwidth necessary for the distributed search system.

10. The system of claim 9, wherein in order to factor in the latency requirements and calculate the cost of operation the system is configured to determine a redundancy of bandwidth necessary for the centralized search system.

11. The system of claim 3, wherein in order to determine the power consumption of the server or cluster the system is further configured to determine CPU utilization for a CPU of the server or cluster.

12. A computer program product, comprising a computer usable medium having a computer readable program code embodied therein, said computer readable program code adapted to be executed to implement a method for designing a search engine system, said method comprising:

determining a sum of power costs for at least two designs, each design having a different number of nodes from the other designs;

determining a sum of bandwidth costs for the at least two designs, each design having a different number of nodes from the other designs; and

determining an optimal number of nodes for the search engine system.

13. The computer program product of claim 12, wherein determining the optimal number of nodes is calculated as $C_n \cdot \left(\sqrt{\frac{U_w}{U_{bw}} \frac{1}{1-x}}\right)$, where U_w is the cost of power per month, and U_{bw} is the cost of bandwidth per month, and C_n is a normalization constant and that cancels out the unit of U_w/U_{bw}.

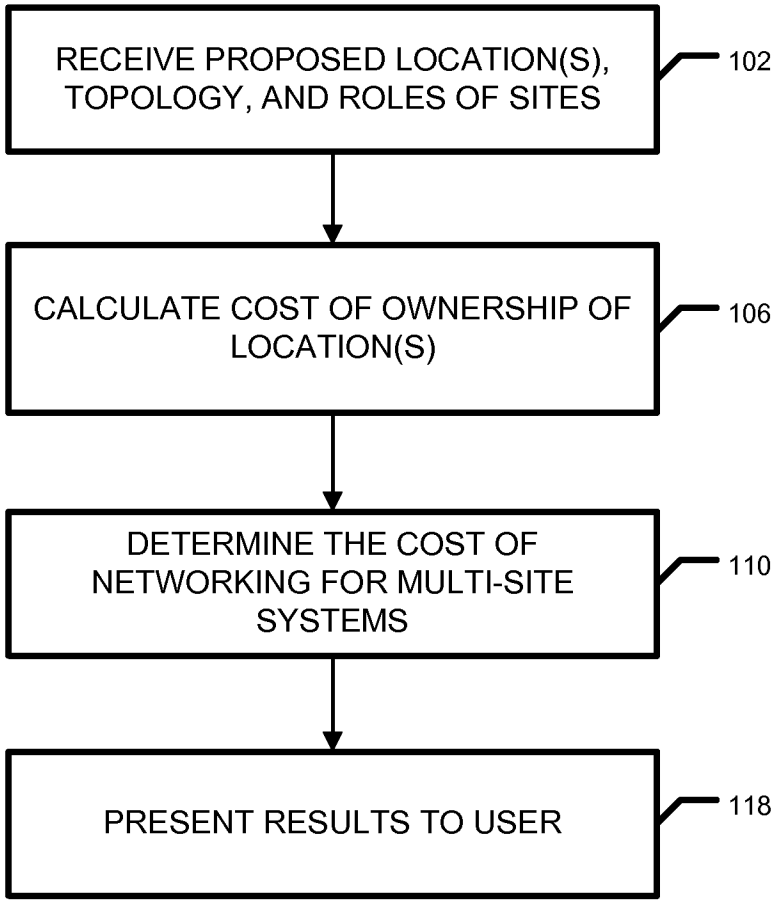
RECEIVE PROPOSED LOCATION(S), TOPOLOGY, AND ROLES OF SITES
102
CALCULATE COST OF OWNERSHIP OF LOCATION(S)
106
DETERMINE THE COST OF NETWORKING FOR MULTI-SITE SYSTEMS
110
PRESENT RESULTS TO USER
118

FIG. 1

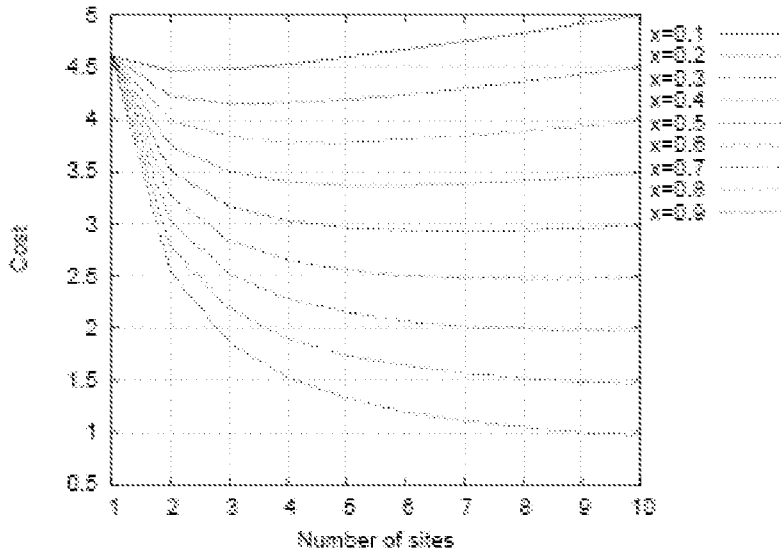

Figure 2: Cost of processing with a fully-distributed architecture, $U_w/U_{bw} = 0.1$ *Mbps-month/KWh*.

FIG. 2

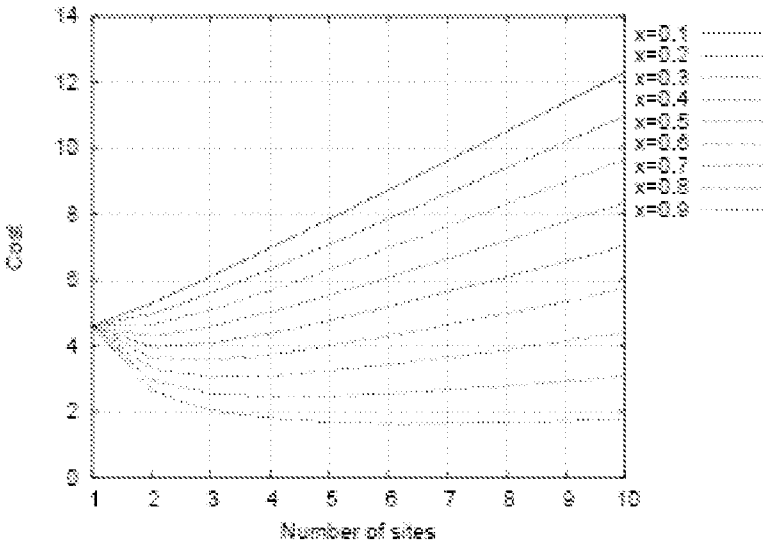

Figure 3: Cost of processing with a fully-distributed architecture, $U_w/U_{bw} = 0.01$ *Mbps-month/KWh*.

FIG. 3

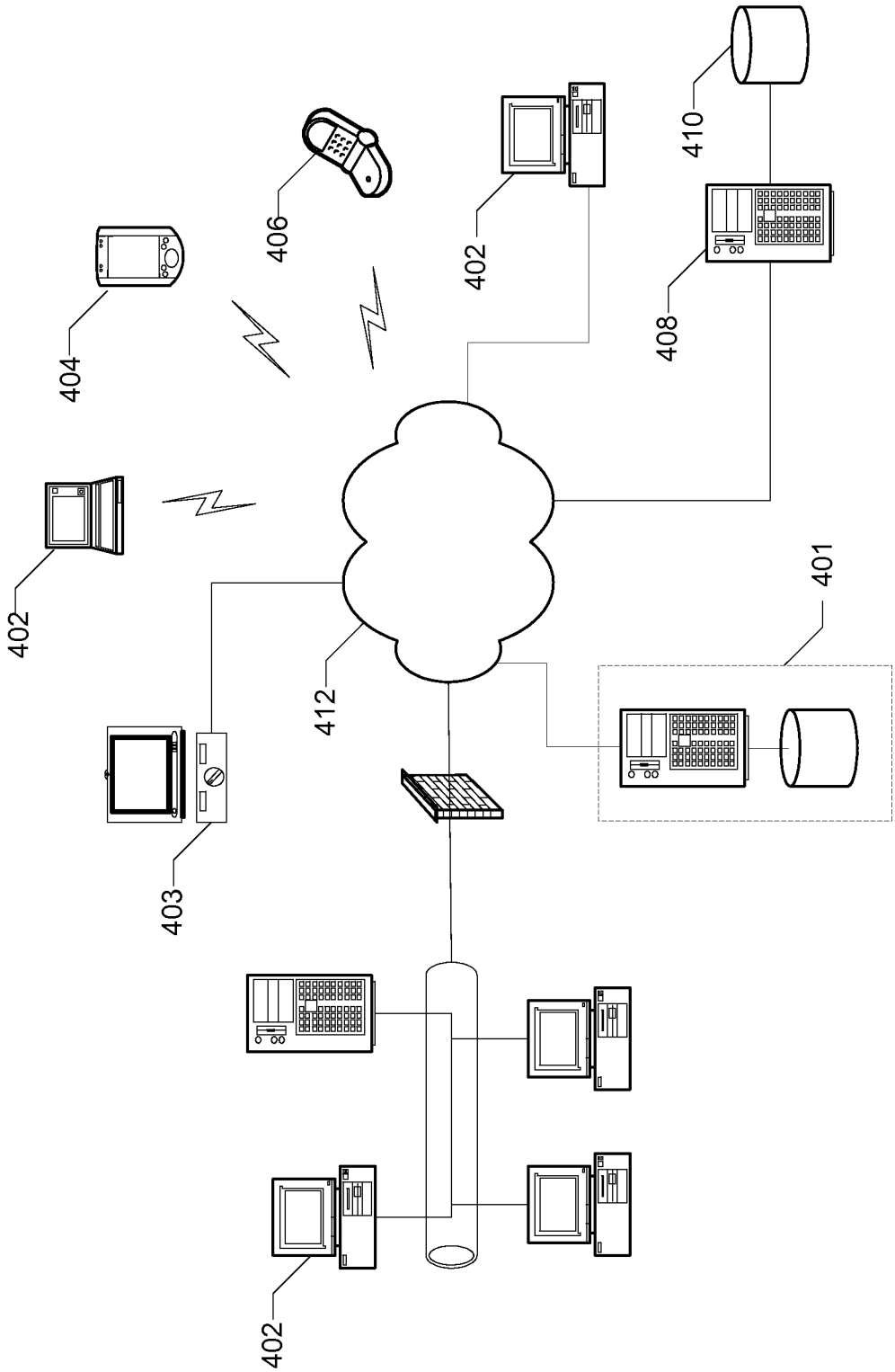

FIG. 4