IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 4, Ver. I (July – Aug. 2015), PP 36-47
www.iosrjournals.org

Study of Simulation for Data Webhousing System by Challenging Technology and Performing Tuning Techniques

AtheerHadiIssaalrammahi
College Of Veterinary MedicineUniversity Of Al-Qadisiyah
aulatheer@gmail.com

Abstract: *One of the most widely discussedtechnologiesare theInternet and itsassociated environment theWorldWide Web.Web technologyhasa broad popular supportamong entrepreneursand technicians likewise. The web environment is owned and managed by the corporation. It may be outsourced,but in most cases, the Web is a normal part of computer operations, and is often used as a center for the integration of business systems. An interaction occur when the Web create a transaction to execute a client order, for example. The transaction isformatted and sent to the corporate systems, where it is processed as any other order. In this sense, the Web is not just another source of business transactions that is entered.There is a decision support database, which is kept separated from the organization's operational data, which is Data Warehouse. This is a huge database containing historical data are summarized and strengthened. In fact, data warehouse provide the basis for the functioning of an e-commerce environment based on Web. This document focuses on two subjects – page relevance to a particular sphere and page contents for the explore keywords to advance the quality of URLs to be scheduled thereby avoiding irrelevant or low-quality ones. We need to build a vertical search engine that receives the seed URL and sorts URL-addresses to bypass content-based pages like to go to a specific area, such as medical or financial domains.*

Keywords: *Data webhouse, Data warehouse, Meta-search engine, vertical search, Web Searchengines.*

I. Introduction

Inour contemporary lives, number of clients is increasing on the Web, frequency of use andthe difficulty of sites.The applications ofan information systemare in needto be familiarwith the needsof the user sincetheuserexperienceis increasing. Hence theneed to understand andsatisfyenduser demandis increasing rapidly.

Inthe process of expandingthe Web,in its number of users, innovative confronts forinformation recoveryare formed.The quantity of data on the Webis increasing very fast.Monitoring end-user behavior fan information system(IS) is a precious wayto review its impactand conduct itsdevelopment.Moreover,observinghowend usersactinthe system develops theknowledge and informationabout their desires andlets system version based on theirlong-ago actions.Besides the adaptation of thesystem, there areother benefits and profits connected withmonitoring the useof an IS. [1]

Web Search engines like Google and Alta Vista offer massive quantity of information many of which might not be related to the clients query. We ought to construct a vertical search engine which obtains a seed URL and sorts the URLs crawled based on the page's content as going to a specific domain such as Medical or Finance domains [2].

The filter component of the vertical search engine rates the web pages downloaded by the crawler into appropriate domains. The web pages crawled is checked for relevance depending on the domain selected and indexed. External users query the database with keywords to investigate. The Domain classifiers classify the URLs into appropriate domain and are obtainable in descending order according to the rank amount [3]. This article aims to build simulation requires multiple tests to be optimized in order to implement and take advantage of all technologies. This will be done in a few schemes of construction, the definition of the main dimensions, the construction of the substantive issues and then by placing the hardware and software configuration items that cannot tolerate the implementation of a meta-search engine, data Webhousing and associated with complex technologies suitable methods of performance tuning.

II. Configuration Steps

Construction the steps for configuring the idea of optimizing the search engine were using data Webhouse. Such steps would be distributed between hardware and software.

2.1 Hardware Infrastructure

The first procedure was to prepare and configure the needed servers that support meta-search engine and data Webhousing. We need to make some adjustments to our data warehouse architecture. New architecture wouldl support public Web server and associated data Webhouse.

2.2 Meta Search Engine

While number of search engines on the World Wide Web was being increased, meta-search engine was the solution for providing ease, efficient, and effective access to information from multiple sources. It was built to solve main problems affecting the search process like database selection, document selection, and results merging, which need additional knowledge about the components of search engines as detailed database representatives, underlying similarity functions, term weighting schemes, indexing methods, and so on. No efficient methods existed in order to find such information without the cooperation of the underlying search engines after including good solutions based on different degrees of knowledge about each local search engine which would be applied accordingly. Best solution must scale thousands of databases, with many of them containing millions of documents, and accessed a day. Building a meta-search engine was the launching move for resolving these issues.

2.3 The Public Web Server and Business Transactions

There were two important connection points to the Web. One was the public Web server that everyone got when the URL was entered into the user’s browser. It was a complex system; it generated all the possible static and dynamic page imaged and other information payloads that users requested over the Web. So one of the capabilities of the Web server was to perform meaningful business transactions and in this case a business transaction server was needed; which recorded the business transactions in a legally and financially responsible way and to never lose these transactions. The business transaction server function was very different from the Web server function and these two servers must be logically and physically separated.

2.4 Web Application

Our Website must contain many features where these features would play different roles in optimizing our search engine. Different steps would be applied to make this Website most effectively support data Webhousing depending on a database consisting of the needed dimensions and fact tables.

2.5 Keywords Database

Since we were applying “Keywords Equivalence” issue, we need to use the keywords database which could be considered as outsource to our system and it look as the dictionary found in Microsoft Word application.

2.6 Initializing data Webhouse

Our data Webhouse should be initialized with the needed data such as the keywords and the documents for the first time. This data could be bought from certain companies that deal with them such companies as Google and others.

2.7 Monitoring Data Webhouse

The main idea of supporting data Webhousing system was to monitor user’s behaviors in order to optimize our engine. So, the Webhouse manager had to monitor the data derived from the clickstream data source to improve the system.

2.8 Applying Models periodically

As we have seen before, every document (web page) had a rank. This rank would be updated and modified from period to period according to our ranking model. So this model should be applied periodically. Moreover, by applying the voting issue we would be able to improve the error margin that was used in the ranking model.

2.9 Refreshing data Webhouse periodically

From time to time, the data found in the data Webhouse need to be updated once from the web at a certain defined period of time. But if the web page was already found, then its rank would not be affected. And in this case the data in the data Webhouse would be always up to date.

III. Implementation Issues

3.1 User Authentication

Every user would be using our Website to retrieve special information from the Web; he need to be a member in this site. So we could imagine the first page which would contain two textboxes for username and password and a login button where these fields would be accessed by users that have already been registered and created their accounts in our Website. This page would contain also a signup button for new users to our engine, and in this case certain information about the user would be gathered.

3.2 Classifying and Categorizing Users

An important feature might play a new role in optimizing our search engine which was to distribute the keywords and the documents to categories. New users who were creating their accounts in our Data Webhousewere required to add some information that was needed in some way to optimize the search engine. Such information was the major category of the user where this category might be the main job or the main interest of that user. When gathering information from the end user who was creating an account at our site, there was a part where that user was defining his major category. Our Website was modified to prompt the user to enter the keyword in the textbox area and to select the category before searching.

Whenever the user loged into our Website and need some data from the Web. He typed the keyword in the textbox and selected the category he need his data about then the search engine would fetch the document dimension in the data Webhouse where that dimension would be containing the keyword which was entered by the user and relating to the selected category. Search engine was optimized in this way by allowing the user to retrieve from the Web the needed data only which deal with certain category because there exit similar keywords in the world and these keywords might be belonging to different categories. So with this issue the search would concentrate on the category specified by the user and the non-useful data or documents would not appear for the user. By applying some features of data Webhouse, we could monitor how users behave on the system so the whole engine would be enriched with the knowledge about their needs and would be adapted on their previous behaviors. Besides system adaptation, there were other benefits associated with monitoring the use of an IS. Under this issue, whenever the user loged into the system the latter knew the most categories searched by that user. Then a certain number of topics or documents would appear on the screen where these documents were related to the categories this user was interested in. In this case, that user might find his information in the list that would be provided by the system without even typing a query.

3.3 Click stream

The Web log contained what were often called click stream data. Each time a user clicked on the Internet to move elsewhere, the record of clicks generated. As a user look at a variety of products, the record that the user look at what the user purchased, and that the user was going to buy considered. No less important was the fact that the Internet was not a client sought and acquired could be determined. By understanding the mentality of the customer to the Internet, a business analyst could understand and recognize very soon how to create promotional products and share of the Company in a much more quantitative and more influential than ever. [4]. So use the resources of the Web and store, we have to create a website to support the storage in order to monitor the behavior of users of an information system (IS), giving you an important way to assess their impact and to show improvement. In distributed Web based status was often made to resolve web server logged or traced of clicks, an extremely low stage approach, with large amounts of information. Dealing with this requires information storage techniques, which summarized the flow click to notions such as session path and navigation and join them with information about the structure and meaning of IS and its members, resulting in what was calledData Webhouse[5].

Two most important steps were:

- Provide an allocation made to each Web page for the data mart clicks you could understand what the visitor was playing.
- Establish a cookie program that identified visitors steadily over time and the servers that belong to the company.

3.4 Web Search Engine

Furthermore, the search engine was a data recovery system programmed to help out information stored in a computer system. Help search engines to minimize the time required to locate the information and the amount of information that should be reviewed, as well as other techniques for implementation of information overload. The most public search engine was the web search engine which searches for information on the World Wide Web. Search engines offered a web interface to explore information about the World Wide Web. The information could be web pages, images and other file types. Some search engines also removed the existing data in newsgroups, databases, directories or release [6]. Search engines collected and organized information automatically, enabling customers to contact with huge amounts of data quickly and successfully. In addition, because the operations were automated in search engines, which seems to elude structural biases and data presentations biased inherent in the media edited by humans. Search engines even promise not to be "wrong." Knowledge of search engine should radically decreased to keep the web of enlargement. Consumers might think (or desire) to the contrary, but the survey results reflected human editors judgments like those made by traditional media. So these editorial decisions systematically bias searched results in a variety of ways. Moreover, the number of new end users without technical expertise in Web research was relatively high [7]. These citizens were probable breaking the Web by means of the main search engines and meta-search engines like Google, Yahoo, AV, etc. These Meta search engines frequently depended on keyword identical and often returned low quality matches. To create matters worse, some advertisers were trying to represent concentration of people taking events to mislead automated search engines. The rapid growth of the Web and the number of users, generated new confrontations and challenges for information retrieval. Crawler-based search engines such as Google used spiders to search the Web to find information. Scanning the Web was carried out by several distributed crawlers. Tools to read the content of the pages of the site, Meta data, and follow the links that the site can connect. This information was analyzed and indexed, for example meta tags could be reactivated containing words from the titles, headings, or special fields, and placed in a directory database for use in subsequent consultations [7].

After crawling and indexing, when an end user performed a search by typing a keyword, the search engine applied an algorithm on its huge database and found pages that were best matching and relevant. From the perspective of a search engine, the search requested from a searcher was considered successful only if one or more of the top search results were related to the seeker’s objective. Therefore, to maximize the perceived search success, search engines generally adjusted their ranking algorithms to support the interests of the majority. Consequently, minority interests often received marginal exposure in search results. To address the interests of the majority, search engines often included the popularity of metrics in its ranking algorithm. Moreover search engines payed attention only to the first blows before attempting a new search, so it was very important to optimize the classification procedure. [4][8]

IV. Simulating Scenario

4.1 Analysis of Clickstream:

To understand what kind of information can be gathered, consider the behavior of a particular person X who decides to buy a book online. He signs to the Internet and uses a search engine to find sites that sell your favorite book. As the results appear on your screen, he clicks on the first link. This takes you to an online book store, and he begins to navigate the site. Person X has the entire category and language he want and add it to your shopping cart. He is ready to checkout and enter your personal information, credit card number and shipping address. The next screen displays the order information and total cost. After seeing how much the company charges for shipping and handling, person X decides to cancel the operation and return to the search engine to find your book in a different online store. Throughout this process, the data of click-stream of the person X was collected by Web retailer, providing a detailed view of how he arrived at the scene, the Web pages he saw, the goods he considered purchase, and point he left the site. In our system, clickstream data for a particular user who is retrieving special information will be collected by the manager Webhouse data, giving a complete picture to the Web pages he sought, information he recovered, and the state he ended his search and left our site. The data specific click-current that can be collected includes information such as the client's IP address, knocked date and time, HTTP status, bytes sent, download time, the HTTP method (get, post), landing page, the user Agent, query strings, server IP address, and cookie data. If the user enters our site to search certain information from the Web, we’ll usually be able to determine the referrer page and the search words that will be entered, and we’ll be able to manage the ranking of the search results which will appear.

For example, the click-stream data typically looks like:

```
dial1-30-45.nbn.net - - [2/Feb/2012:19:54:14 +0000] “GET/html/win7_updates.htm HTTP/1.0” 200 54
http://www.infoseek.com/Titles?qt=%220EM+service+release+2%22&col=New+Search&oq=%22service+rele
ase+2%22&sv=N4&lk=ip-noframes&nh=10 “Mozilla/15.1 (Win7; I)” [3].
```

We can notice that the IP address of the user visiting the site is listed first (dial1-30-45.nbn.net). The next field sometimes shows the login ID of users who have entered a password protected area of your site. The date and time of the page request is listed next in Greenwich Mean Time (GMT). Following this is the name of the page viewed on our site. Then the referrer field is given which tells you what page the user came from. In this case, the visitor found our site by using Infoseek.com. The user agent field (“Mozilla/15.1 [en] (Win7, I)”) is last, which shows us what browser the visitor was using. To find the golden nuggets, we first have to reduce the amount of data to analyze. There are a variety of other techniques that we can follow to optimize the performance of our system and improve the management of our clickstream data.We can reduce the total data to be processed by using Include/Omit processing to quickly identify the exact records that we need from the Web logs. Similar data can be merged together into a single file to analyze. During this process, we can specify how the data is ordered. By joining data, we're matching keys in multiple files and creating one record from two records that have a common key. To optimize the record format, we can perform inner, outer, left and right joins. Another technique is to search for patterns anywhere in specified fields. Once a pattern is found, we can then extract portions of the field. To make it easier to sort through data, we can add a unique record identifier number to the output. This record number can start at any value and can be useful for databases with a uniqueness constraint.To accomplish the fastest large data extracts, remove all GROUP BY, ORDER BY and DISTINCT clauses from the SQL SELECT statement that unloads the data, and then perform those operations with a special-purpose sort tool on the unloaded file. Keeping in mind that SUMMARIZE processing is a much faster equivalent to the GROUP BY and DISTINCT clauses, while the KEY option is a faster equivalent to the ORDER BY. We can use SUMMARIZE processing to generate pre-stored aggregates. This technique will help us optimize query processing and data warehouse response time. To split our data into separate partitioned table ranges, we can use multiple OUTFILE processing. Then we can utilize Pre-Sort to accelerate the database load itself.We can take the advantage of our Web servers to optimize its processing clickstream. We use our website to do business with a growing number of online users. Behind the scenes, data Web logs form click with the flow, including details such as the number of unique visits to the site, most popular pages, most web pages fetched.

4.2 Star Schema

Relational schema that has a central fact table called table that contains one or more units of measurement and a table of one or more dimension that reference the fact table. Each dimension table has attributes that represent the various granularities dimension tables are not standardized because the attributes in a table set different granularities. It is regarded as the central table. It contains a unit of measurement that is Measure1, Measure2 ... They are used to make the calculation using aggregated group by function (sum, average ...). It has a dimension table for each FK reference to join the fact that the dimension tables. Dimension tables (Dimension1, Dimension2, Dimension3, Dimension4) are not normalized because different attribute, define different granularities.

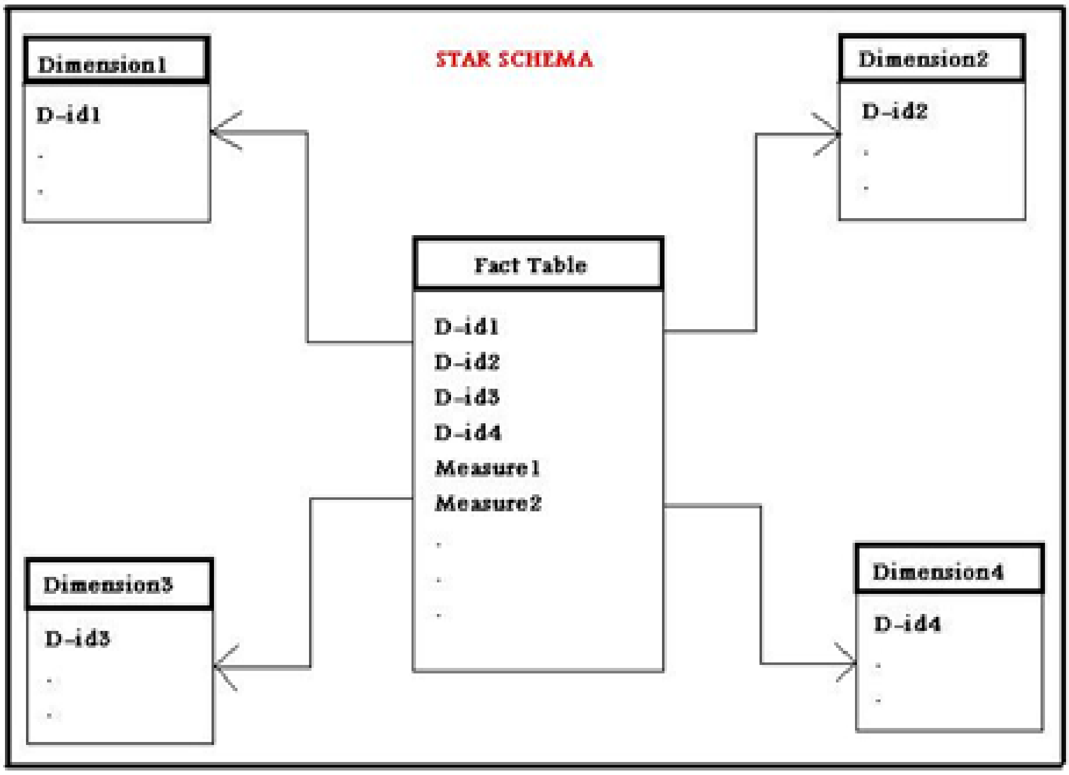

Fig. (1) : Star Schema

4.3 Snow-Flake Schema

Relational schema, where a central table contains the dimension tables facts analyzed references. Dimension table can be described by a sequence of tables that represent the various granularity. Snowflake scheme adapts the same representation in fact a table, but standard size tables when possible. Therefore, each granularity is represented by one or more sequence tables.

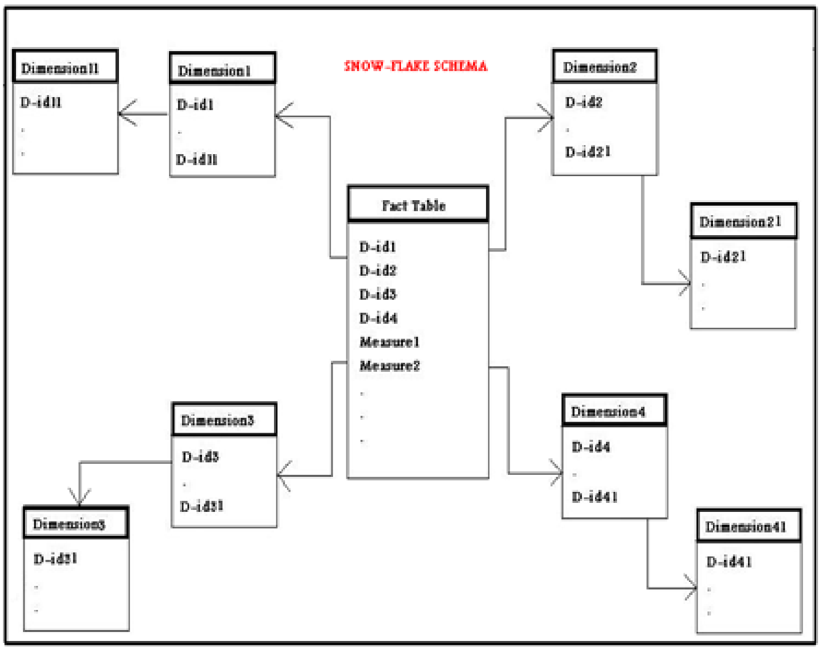

Fig. (2) : Snow-flake Schema

4.4 Surrogate key

This key is an anonymous integer. The first record in this dimension may as well have key value = 1. The second record will have key value = 2, and so on. The point of a surrogate key is that there is absolutely no semantics in the value of the key. It is just a vehicle for joining the dimension table to a fact table. We make surrogate keys so that we can deal with some important administrative situations that are unique to the data warehouse. These situations don't usually arise in an OLTP environment, and OLTP database designers often don't see the need for data warehouse surrogate keys. The first situation that arises in the data warehouse is if the object being referred to is unknown, unmeasured, corrupted, inapplicable, or hasn't happened yet. In all these cases, a semantically meaningful production key cannot be applied, yet the data warehouse has to provide a key. A second reason for surrogate keys is that this insulates the data warehouse from changes in key administration in the production data. This may not be a big issue with dates, but it is a big issue with most of the other dimensions.A final reason for surrogate keys is the need to handle slowly changing dimensions. Suppose that we want to revise our description of a user. A corollary to this surrogate key discussion is a kind of discipline in our application. The following constraints will be applied:

- Never build in any dependence on the surrogate keys directly in our queries.
- Do not constrain directly on a key value.
- Do not assume the surrogate keys are assigned in some meaningful sequential order.
- We have plenty of other fields in every dimension that can be used for constraining, navigating, and ordering.
- Leave the keys themselves out of our application logic.

4.5 Dimensions

4.5.1 Page Dimension

The page dimension describes the page context for a Web page event. The grain of this dimension is the individual page type. Our definition of "page" must be flexible enough to handle the evolution of Web pages from the current mostly static page delivery to highly dynamic page delivery in which the exact page the user sees is unique at that instant in time. We will assume even in the case of the dynamic page that there is a well-defined function that characterizes the page, and we will use that to describe the page. We will not create a page record for every instance of a dynamic page because that would yield a dimension with an astronomical number of records. These records also would not differ in interesting ways. What we want is a record in this dimension for each interesting distinguishable type of page. Static pages probably get their own record, but dynamic pages would be grouped by similar function and type.When the definition of a static page changes, because it is altered by the Webmaster, the record in the page dimension can either be overwritten or can be duplicated. This decision is a master of policy for the data Webhouse, and it depends on whether the old and new descriptions of the page differ materially, and whether the old definition should be kept for historical analysis purposes.

4.5.2 Event Dimension

The event dimension describes what happened on a particular page at a particular point in time. The main interesting events are Open Page, Refresh Page, Click Link, and Enter Data. As dynamic pages based on XML become more common, the Event dimension will get much more interesting because the semantics of the page will be much more obvious to the Web server. Each field in an XML document can be labeled with a user-defined tag.

4.5.3 Session Dimension

The session dimension provides one or more levels of diagnosis for the user's session. For example, the overall session context might be retrieving information from the Web. The success status would diagnose whether the mission was completed. The user status attribute is a convenient place to label the user for periods of time, with labels that are not immediately clear either from the page of the immediate session. This dimension is extremely important because it provides a way to group sessions for insightful analysis.

4.5.4 Document Dimension

The Document dimension describes certain information about the Web pages that might be retrieved from the Web and would appear on the screen as a result for a specific keyword. This dimension consists of many attributes that will define each document and affect its position in the list of results that should appear on the screen.

4.5.5 Keyword Dimension

The keyword dimension describes one of the two main services that our engine will be working on, where this dimension will contain words and information about these words. It will be attached permanently to the clickstream and takes on meaningful values in many contexts. Keyword dimension will be very well understood by the data warehouse design community. It will contain the needed attributes to describe each keyword. Some of these attributes may create a merchandise hierarchy that allows groups of keywords to be rolled up into ever larger clumps. Other attributes have nothing to do with the merchandise hierarchy but are simple useful descriptors. In other words, each group of keywords will be belonging to a certain category. So we may have an attribute in this dimension which looks as "Category". Moreover, the category "PROGRAMMING" for example may be belonging to a higher category which is "COMPUTER". In this case, we may see another attribute in this dimension also which looks as "Category". This dimension will be containing the two attributes "Sub-Category" and "Category".
Other dimensions can be tracked as **User Dimension**, **Time of Day Dimension**, and **Calendar Date Dimension**.

4.6 Personal Data Repository

Eventually a user's trust level or business requirements will evolve to the point where he is willing, or required, to identify himself to one of an enterprise's Websites. There are a number of very strong arguments for consolidating personal data, including demographics and credit information, into one and only one location within the enterprise. Regardless of which of an enterprise's subsidiary Websites first collects user information, we recommend that it be forwarded and maintained in a central repository. There are many reasons for this:

- Information such as addresses and telephone numbers stays current for all internet users.
- Access can be much better controlled and audited if the information to be protected is in one location and not distributed throughout an enterprise.
- The user is asked only once for personal information.
- Websites can quickly respond to changing legal requirements for privacy control.

The user's access privileges to controlled or premium content can be monitored and granted on an enterprise-wide basis when needed.

4.6.1 Building Trust

In order to get a visitor to share personal information with a Website and especially to get him to state this information accurately requires a level of trust to be built between the user and the Website. Although a Website is an inanimate object, this is exactly where the user will focus his trust, or lack thereof. And if one Website in an enterprise collection is deemed untrustworthy, other sites in the same enterprise will become untrustworthy by association. Once this happens, it becomes very difficult to entice the user back into a trusting position.

4.6.2 Consistent Cookies

When we cookie visitor, we can track their return visits to our Website. Over time it will be in the enterprise's best interest to maintain one cookie on a visitor's computer that can serve all organizations within the enterprise. This is the role of a cookie server in which its purpose is to maintain an identity for each visitor regardless of his entry point into the enterprise Website complex. With an enterprise-level cookie server we will be able to tie together the user's visits regardless of his entry point. We suggest the use of a single cookie with the ID encoded into the cookie value and containing checksums or perhaps even error correcting codes so that we can detect and discard cookies that have been altered by the user or by cookie protection software. To avoid later privacy conflicts, don't put any human-readable information into cookies. There are many different ways of using the cookie server. One method of obtaining a cookie-encapsulated user ID is to place on every potential departmental or divisional entry page a tag that calls the cookie server to read the corporate-level cookie. If the distributed server needs this value for site personalization the user ID can be returned to a hidden field in the page's HTML. If the user ID for post-analytical purposes, the cookie server can generate a server log entry to be merged with other log entries for post-session clickstream analysis.

4.6.3 Null Logging Server

There are two proactive steps that we should take to make our Website most effectively support data Webhousing. The two most important steps are:

[1] Provide an elaborate attribution of each Web page so that the clickstream data mart can understand what the visitor was touching.

[2] Set up a cookie program that consistently identifies visitors over time and across the servers that belong to your enterprise.

4.7Page Rank Issue

With the rapid growth of the Web, providing relevant pages of the highest quality to the users based on their queries becomes increasingly difficult. The reasons are that some web pages are not self-descriptive and that some links exist purely for navigational purposes. Therefore, finding appropriate pages through a search engine that relies on web contents or makes use of hyperlink information is very difficult. When the user logs into our system and retrieves some information from our data Webhouse, the list of results will appear on the screen according to specific PageRank for each web page.

We assume there is a "random surfer" who is given a web page at random and keeps clicking on links, never hitting "back" but eventually gets bored and starts on another random page. The probability that the random surfer visits a page is its PageRank.

Another intuitive justification is that a page can have a high PageRank if there are many pages that point to it, or if there are some pages that point to it and have a high PageRank. Intuitively, pages that are well cited from many places around the web are worth looking at. Also, pages that have perhaps only one citation from something like the Yahoo! homepage are also generally worth looking at. If a page was not high quality, or was a broken link, it is quite likely that Yahoo's homepage would not link to it.

PageRank handles both these cases and everything in between by recursively propagating weights through the link structure of the web. PageRank of a web page might be updated when it is clicked by the end user. So this attribute will be updated when its document is clicked from the list of results. But the modification of this rank will be according to certain formula where the latter will depend on different factors such as number of clicks and time.

Unfortunately, the number of new end-users inexperienced in the art of Web research is relatively high. These people are likely to surf the Web using leading meta-search engines such as Google, Yahoo, AV, etc. These meta-search engines usually rely on keyword matching and often return low quality matches. To make matters worse, some advertisers attempt to gain people's attention by taking measures to mislead automated search engines.

Any user may not be able to know all the meanings of a single word so we will create a way to cover this problem automatically. By applying this way, most of the topics or web pages that are related to certain idea may be fetched from the data Webhouse. Web users are being distributed over the world and these users may not be available to the exact keywords that belong to different cultures but at the same time they are available to their meanings, so our engine will try to be popular to all habits, traditions and societies.We will take the advantages of different features that could be applied using Data Webhouse and especially to perform this keyword equivalence issue. Such features are monitoring previous users' behaviors on the web and the characteristic that the Webhouse will store historical data, these two technical features will play an important role in this domain.

An important feature may play a new role in optimizing our search engine which is to distribute the keywords and the documents to different categories. We have seen before the description of the “keyword” dimension where that dimension contains “Sub-Category” and “Category” attributes. When gathering information from the end user who is creating an account at our site, there is a part where that user is defining his major category. Our Website is modified to prompt the user to enter the keyword in the entry field area and to select the category before searching, and then the search engine will fetch the document dimension in the data Webhouse where that dimension will be containing the keyword which is entered by the user and relating to the selected category.Search engine is optimized in this way by allowing the user to retrieve from the Web the needed data only which deals with certain category. So with this issue the search will concentrate on the category specified by the user and then non useful data or documents will not appear.

4.7.1Web Usage Mining

Applying data mining techniques on access logs unveils interesting access patterns that can be used to restructure sites in more efficient groupings, pinpoint effective advertising locations, and target specific users for specific search.

4.7.2 Main Idea of ranking feature

So by using clickstream data source we can know number of clicks made by every user on every document. Since our warehouse will be updated periodically then the new model for ranking the documents will depend on number of clicks done on every web page and on the number of periods have passed. By this way, the rank of every document will be as an overall rank for all periods.

New Rank for document A:

TotalClicks = PrevClicks + CurrClicks

⇨ TrueClicks = TotalClicks – (TotalClicks * α)

TotalPeriod = PrevPeriod + CurrPeriod

∴New Rank for document A = TrueClicks / TotalPeriod

The above model will be applied periodically in order to update the rank of each document or web page at the end of every period. As we are seeing in this algorithm, two factors are used; the first factor is the time factor and number of clicks is the second factor. These factors are inserted to apply tangible facts in computing the rank of each document. Each document will be ranked according to the total number of clicks made by the users in all periods.Suppose document A is found in our data Webhouse and the same document have been retrieved several times by different users in different periods. The period is defined by the Webhouse manager which is an interval of time. Suppose that three previous periods passed which means that the Webhouse was updated for three times, and suppose also the total number of clicks made by the users in those previous periods was 50,000 clicks.

Consider that 10,000 clicks were made by different users in the current period.

Previous Clicks = 50,000 clicks
Previous Periods = 3 periods
Current Clicks = 10,000 clicks
Current Period = 1 period

Total Clicks =

$$\begin{array}{r} 50{,}000 \\ +\ \underline{10{,}000} \\ 60{,}000\ \text{clicks} \end{array}$$

Consider that the Error Margin α is 5 %

True Clicks = 60,000 – (60,000 * 0.05) = 57,000 clicks
Total Period = 3 + 1 = 4 periods

➔ New Rank for document A = True Clicks / Total Periods
= 57,000 / 4
= 14,250

The rank of every document or web page will be in one of three states: Increasing, stable or decreasing state in each period. Webhouse manager will be able to know at which interval of time a document is preferred by most end users and he will be able to know at which interval of time a document is becoming less important than before with respect to the most end users. Using the historical data found in the data Webhouse, different analysis could be done by monitoring web pages ranking.

4.8 Keywords General Analysis

A user needs special information to be retrieved from the Web; he logs into our website and navigates to the search page. The user is prompted to take three actions; one action is to type his keyword in the text area, the second action should be taken is to select the category to search in, and the third action should be taken is to select the type of search where this action deals with the "Keyword Equivalence" issue.

In the third action, the type of search, the user is prompted to select between "Search" and "Advanced Search". If that user clicks on the first button (Search), then the engine will fetch directly the keyword typed from the keyword dimension found in the data Webhouse. List of URLs will be retrieved containing only the needed keyword.

If that user clicks on the second button (Advanced Search), then the engine will make one more operation which is Database lookup from the "Keywords Equivalence" repository before accessing the data Webhouse. Then the retrieving from the Webhouse will consider the keyword typed by the user and its equivalents fetched from equivalence database.

Keywords Equivalence Lookup:

- Given the keyword typed by the user.
- Select all keywords that are equivalent to the given one from this database.
- Result is a list of more keywords.

Data Webhouse accessing:

- Given the list of equivalent keywords resulted from the above lookup.
- Select all documents (URLs) from the data Webhouse and containing the list of equivalent keywords.

Suppose we have in our data Webhouse keyword "Data" which is found in "document1, document2, document6", keyword "Information" which is found in "document2, document3, document4", and keyword "Knowledge" which is found in "document2, document5". The keyword "Data" is equivalent to "Information" and to "Knowledge", and then whenever the user types the keyword "Data", the result will be: URLs of Document1, Document2, Document3, Document4, Document5, and Document6.

4.9 Voting Issue

Using Data Webhousing system provides us different ideas and different options. These ideas could be applied in a way that our search engine is optimized. One of the main tasks that should be done is the feedback of the user. Such thing drives us to construct new idea which is allowing the whole community to vote for every document or web page they have seen or they have fetched. Monitoring users' behaviors on our system allows us to create new technique which results in extracting users' interests and preferences. This result can be used in many ways to optimize the search engine.User logs into our website to retrieve some information from the Webhouse which means from the Web. He types the keyword in the text area and selects a category. A list of URLs (web pages) will appear as a result of the keyword typed by the user. There is a small statement next to each URL document that prompts the user to vote for this URL which is visited by this user. After the user clicks on this page and retrieves his needed information from it, he can vote for this document by checking YES or NO. At the end of every search done, we can result in something like user interests and preferences.

Voting issue could be used to help both end users who are using this engine and us who are implementing the different issue of our Webhousing system. When the user retrieves some information from our Webhousing system, then the URLs list will appear on the screen and there will be found a percentage next to each URL page. This percentage represents the number of voters who like this web page and the number of voters who don't, and this percentage will be computed for the previous periods and not for the current period. This percentage will also use a model similar to ranking model, so it will depend on a factor such as number of YES clicks and number of NO clicks that will be made by different users during the current period of time.

The following model was formulated in order to update this attribute:

- Let **PrevClicksYES** be the number of the YES clicks on document A made by different users in the overall previous periods and not in the current period.
- Let **PrevClicksNO** be the number of the NO clicks on document A made by different users in the overall previous periods and not in the current period.
- Let **CurrClicksYES** be the number of YES clicks on document A made by different users in the current period only.
- Let **CurrClicksNO** be the number of NO clicks on document A made by different users in the current period only.
- New voting percentage for document A:

TotalClicksYES = PrevClicksYES + CurrClicksYES
TotalClicksNO = PrevClicksNO + CurrClicksNO
TotalClicksYesNo = TotalClicksYES + TotalClicksNO

∴ New Percentage for YES voters on document A
= (TotalClicksYES / TotalClicksYesNo) * 100
And New Percentage for NO voters on document A
= (TotalClicksNO / TotalClicksYesNo) * 100

The above model will be applied periodically in order to update the percentage of YES and NO voters on each document or web page at the end of every period. In this way, each document will be voted by the users and a percentage will be computed according to the total number of YES and NO votes made by the users in all periods.By monitoring the user who is using this website, we can know from the clickstream data source if the user votes on a document or not, and if the user votes on a document then we can know if he likes a web page or not. This clickstream will be recorded in our Webhousing system which will be analyzed on different stages to be used. By applying the above model we can know the number of users who really fetch a web page since the result will be derived from tracking the user's action. Then we can end with a factor which affects every web page, such factor is the number of voters who like this web page. Every web page has a certain rank where this rank is calculated according to certain model.

V. Results And Discussion

One result can be obtained is to monitor the number of users searching every category per one period.

Another result can be obtained also is to monitor the total number of categories clicked by one user in a certain period. This result leads to optimize the search engine in a way that whenever that user logs into our Website to use this search engine then the system will know the most categories that user is interested and then a URL list of best topics might appear on the screen, where this result will be related to those categories. In this way, a user may find his information in the above URL list even without typing a keyword in the text area and performing a search.[9]

Suppose we are tracing web page A. This web page is taking nine clicks in this instant; four of these clicks are made by one user, and the other five clicks are made by different users. Then we can know that the first user likes this web page because he visits it many times. So this web page is more important to that user than to others. In this case and after monitoring user's behavior on our Webhousing system, we can know the most important pages visited by this user and we can know the pages that are visited only once by that user. Then we can build an analysis to deduce the important web pages for one user and this result could be used when that user logs into our Webhousing system to retrieve certain information. We can say that in this way the search engine is optimized for every user, where he can access and fetch his important web pages directly when logging into this system. Moreover, we can construct our analysis in a graph to monitor a web page's importance according to number of clicks made by one user and according to time. We can see that the rate of a web page's requests will be swinging between two different states, either increasing state or decreasing state for one user.[10][11]

VI. Conclusion

- Integrating a warehouse to the external system and closing the loop by captivating action based on investigation is the most current technology challenge. Most existing technologies permit communication with a data warehouse with a pull model. In such a model the request or user periodically polls the database for investigation or state changes.
- if we are an ISP extending Web access to the directly associated customers, we have an exclusive perspective because we observe each click of the user that can permit much extra commanding and invasive analysis of the end user's sessions than can be offered by a particular target Website.
- the Website may create a persistent cookie in the user's machine that is not canceled by the browser when the session ends. Of course it's possible that customers will include their browsers set to decline cookies or may by hand clean out their cookie file, so there is no unlimited warranty that even a persistent cookie will endure. Although any agreed cookie can simply be examined by the Website that caused it to be produced, confident groups of Websites can have the same opinion to store a familiar ID tag that would allow these sites merge their disconnect notions of a user session into "supersession".

Reference

[1]. Lawrence Steve; Lee Giles C., "Accessibility of information on the web"New York, NY, USA, ACM, Volume 11, Issue 1, (2000), pp: 32-39

[2]. Joachims T., "Optimizing search engines using click through data"Ithaca, NY USA, Cornell University, Department of Computer Science, (2002).

[3]. Yuan Wang, David J. Devitt, "Computing PageRank in a Distributed Internet Search System" Toronto, Canada, VLDB Endowment, (2004).

[4]. Inmon, William H. "An Architecture for Managing Click Stream Data," BILLINMON.COM, (March, 2001), 5-7, 10-11

[5]. Jesper Andersen, Anders Giversen, Allan H. Jensen, Rune S. Larsen, Torben Bach Pederson, JanneSkyt. "Analyzing clickstreams using subsessions", Proceedings of the 3rd ACM international workshop on Data warehousing and OLAP, pp. 25-32, ACM Press, (2000).

[6]. http://searchengineland.com/search-30-the-blended-vertical-search-revolution-12775Nov 27, 2007 at 9:11am ET by Danny Sullivan

[7]. 7- Castillo, C. ,"Effective Web Crawling", PhD thesis, University of Chile, (2004).

[8]. Michael Chau and Hsinchun Chen, "Comparison of Three Vertical Search Spiders", Journal of Computer ,Vol. 36, No. 5, 2003, ISSN 0018-9162, pp. 56-62, publisher IEEE Computer Society.

[9]. W.H. Inmon, "Building the Data Warehouse"Chichester, John Wiley & Sons, third edition,(2002). pp: 297-305

[10]. Craig Abramson and Kenny Kistler, "Managing Clickstream Data"(2001).

[11]. Kimball Ralph, Merz Richard , "The Data Webhouse: Building the Web-enabled Data Warehouse", New York, John Wiley & Sons, (2000).