Volume 4, Issue 7, July 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Hybrid Approach for Performance of Web Page Response through Web Usage Mining

Ravinder Singh
CSE & KurukshetraUniversity
Haryana, India

Bhumika Garg
CSE & Kuruksherta University
Haryana, India

Abstract— Web Usage mining aims to automatically discover and analyses the pattern in click stream and associated data is self controlled or generated as a result of user interaction with web browsers, on one or more websites. Our web Usage mining is the combination of the two approaches that is Web Caching and Web Pre-fetching. Web caching is an important technique for which enhance performance of web based applications. Web caching is used to reduce network lattice, server load and user-related delays by replicating popular content on proxy caches that are strategically placed within the network. Web pre-fetching schemes have also been widely discussed where web pages and web objects are pre-fetched into the proxy server cache. This proposed work presents an approach that integrates web caching and web pre-fetching approach to improve the performance of proxy server's cache. Our Hybrid approach Web caching and Web pre-fetching can complement each other since the Web caching technique exploits the temporal locality and compares to Web pre-fetching technique utilizes the spatial locality of Web objects. In this paper approach how the response time of hit taken from the user cache is less as compare to the data taken directly from the log file.

Keywords— Web Usage Mining, Web Caching, Web pre-fetching, Response Time, World Wide Web, Proxy server

I. INTRODUCTION

World Wide Web is a huge repository of data. It has become one of the most important media to store, share and distribute information. The expansion of web is very rapid which has provided a great opportunity to study user and system behaviour by exploring web access [1].Data mining (the analysis step of the "Knowledge Discovery in Databases" process, or KDD) is the process that attempts to discover patterns in large data. WEB MINING has been defined as applying data mining techniques on web data to discover knowledge. Some researchers have applied mining techniques on the web logs maintained byte servers so as to discover user access and traversal path [2].

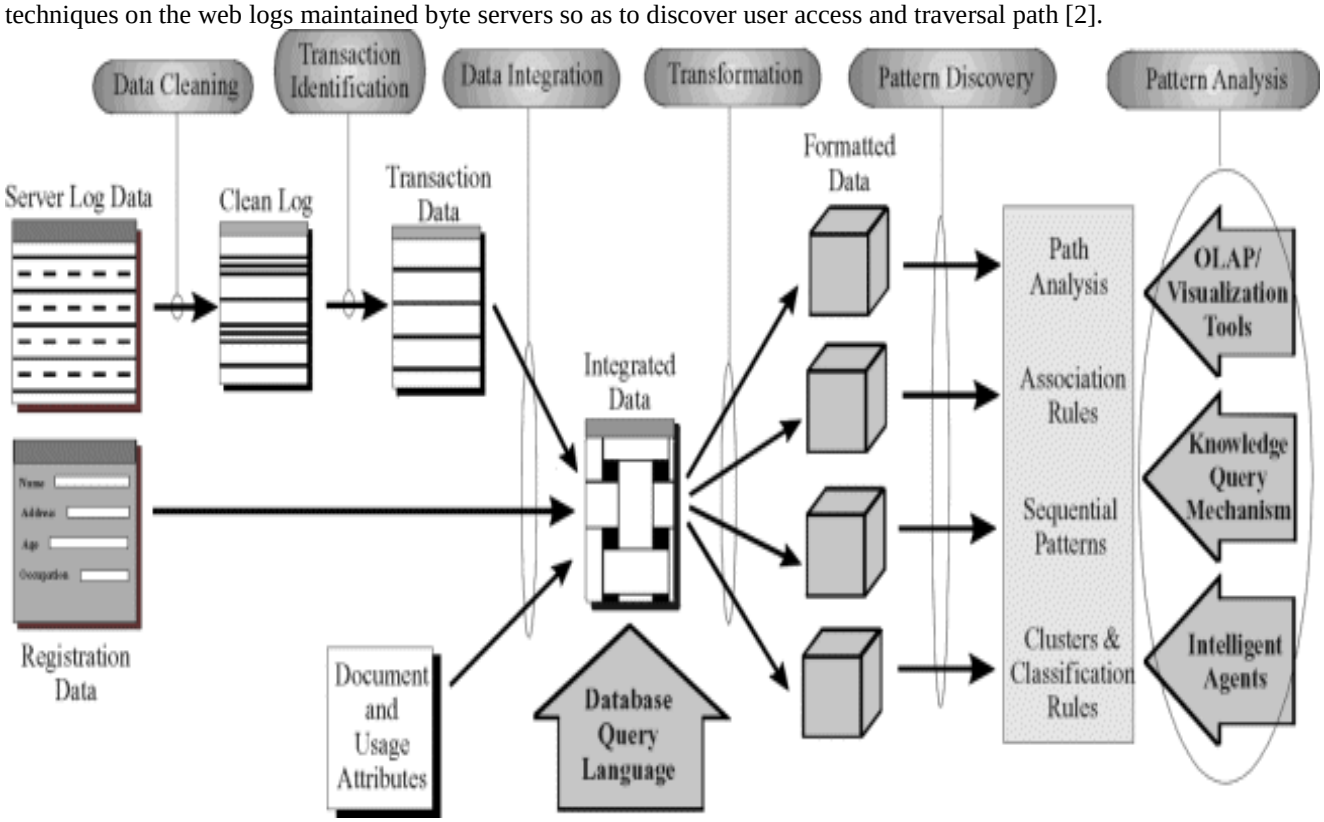

Figure 1: Web Mining Architecture

Web usage mining is the type of data mining process for discovering the usage patterns from web information for the purpose of understanding and better provide the requirements of web-based applications. Web usage mining imitates the actions of humans as they interact with the Internet. Examination of user actions in communication with web site can offer insights causing to customization and personalization of a user's web practice. As a result of this, web usage mining is of extreme attention for e-marketing and e-commerce professionals. Web usage mining is the type of data mining process for discovering the usage patterns from web information for the purpose of understanding and better provide the requirements of web-based applications. Web usage mining imitates the actions of humans as they interact with the Internet. Examination of user actions in communication with web site can offer insights causing to customization and personalization of a user's web practice. As a result of this, web usage mining is of extreme attention for e-marketing and e-commerce professionals. Web usage mining involves of three phases, namely, pre-processing, pattern discovery and pattern analysis. There are different techniques available for web usage mining with its own advantages and disadvantages. Web caching has been used as one of the effective techniques to reduce network traffic, thereby decrease user access latencies. However, the cache storage space is limited. Some pages must be removed when the cache is full. As a result, the Efficiency is dropping from what supposed to be, because the deleted page may be requested again. A lot of studies have been done to improve the Web caching performance ([3], [4]) For example; in the study of Web mining technique is applied to predict the future web access. In the study of classification and association rules techniques are used to provide the behaviour of website utilization. Similar to several prior studies, related work have applied data mining techniques for building the model of cache replacement policy.

The knowledge and comprehension of the behaviour of a web user are important keys in a wide range of fields related to the web architecture, design, and engineering. The information that can be extracted from web user's behaviour permits to infer and predict future accesses. This information can be used, for instance, for improving Web usability, developing on-line marketing techniques [5] or reducing user-perceived latency, which is the main goal of perfecting techniques. These techniques use access predictors to process a user request before the user actually makes it.

II. CATEGORIES OF WEB MINING

The Web data mining should focus on these three issues:

(i) Web Structure Mining,
(ii) Web Content mines
(iii) Web usage mining.

All of the three categories focus on the process discovery for unknown and potentially very useful information from the web. Though each of them focuses on same attribute but each might be different mining objects of the web.

(i) Web Structure Mining

It involves the web documents structure and links. In some insight is given on mining structural information on the web. Web structure mining is very useful in generating information such as visible web documents, luminous web documents and luminous paths, a path common to most of the result returned, use linkage information to improve search engines, hyperlink structure analysis, link analysis, graph, categorization, mining the document structure .[6]

(ii) Web Content Mining

Describes the automatic search of information resources available online .It represents structured, unstructured, semi structured documents and model to interactive retrieval view and DB View. All the above it is a Mining, extraction and integration of useful data, information and knowledge discovery from Web page contents. Web content mining examines the contents of web pages as well as results of web searching. Text mining is directed toward specific information provided by the customer search information in search engines. This allows for the scanning of the entire Web to retrieve the cluster content triggering the scanning of specific Web pages within those clusters. The results are pages relayed to the search engines through the highest level of relevance to the lowest. Though, the search engines have the ability to provide links to Web pages by the thousands in relation to the search content, this type of web mining enables the reduction of irrelevant information.[7]

(iii) Web Usage Mining

Web usage mining is the third category in web mining. This type of web mining allows for the collection of Web access information for Web pages. This usage data provides the paths leading to accessed Web pages. This information is often gathered automatically into access logs via the Web server. CGI scripts offer other useful information such as referrer logs, user subscription information and survey logs. This category is important to the overall use of data mining for companies and their internet/ intranet based applications and information access.

Usage mining allows companies to produce productive information pertaining to the future of their business function ability. Some of this information can be derived from the collective information of lifetime user value, product cross marketing strategies and promotional campaign effectiveness. The usage data that is gathered provides the companies with the ability to produce results more effective to their businesses and increasing of sales. Usage data can also be useful for developing marketing skills that will out-sell the competitors and promote the company's services or product on a higher level. ([7],[8])

III. PROBLEM DEFINITION

Caching is an important technique for improving the performance of web based applications with help of web caching techniques. Web caching provides great features like traffic reduction, less load on servers, user-end retrieval delays by replicating popular content on proxy caches that are strategically placed within the network. Web pre-fetching schemes have also been widely discussed where web pages and web objects are pre-fetched into the proxy server cache. In our research we will work on integration of web caching and web pre-fetching approach to improve the performance of proxy server's cache. In Domain Top approach for web pre-fetching, combination of knowledge of most popular domains and most popular documents is done by proxy server. In this approach proxy is responsible for calculating the most popular domains and most popular documents in those domains, and then prepares a rank list for pre-fetching. In Dynamic web pre-fetching technique, each user can keep a list of sites to access immediately called user's preference list. The preference list is stored in proxy server's database. In our research we will bring concept of preference list from Dynamic technique into Domain Top approach. Optimized top domain approach will consist of preference list along with the rank list. Advantage of this is that the pre fetching would have wider scope (will be more fast).

The main focus of the research is to improve accuracy in the web mining process. Our research is started with information fetching of pre-fetching and caching techniques. The major targets and objectives for our research is given as below:

- Develop an optimized technique for optimizing the web caching and web pre-fetching processes.
- For find loopholes and issues in new approach and to highlight the benefits for new approach.

This research has focused on providing solution for said problem by enhancing web pre-fetching process. For experimentation we have used database with various web entries and have done cleaning process on the database. Data cleaning is the first step that is applied to the web mining and any other web searching technique. In data cleaning all the images, spaces and the user shown data is just terminated. Suppose we are we are having String http://www.facebook.com/friends/ajax/lists.php. In this string, we will be removing the php link, and user detail that is friend. So the string that will be left to us after cleaning will be www.facebook.com.After the cleaning of the data whatever the data is required by the user will be displayed on the basis of the apriori algorithm. In this algorithm we will assume the confidence level near about 60% and the term that appears less than 60% will be remove and the more combination is applied to take the proper frequent set of the given data. The data cleaning process is shown in below figure.

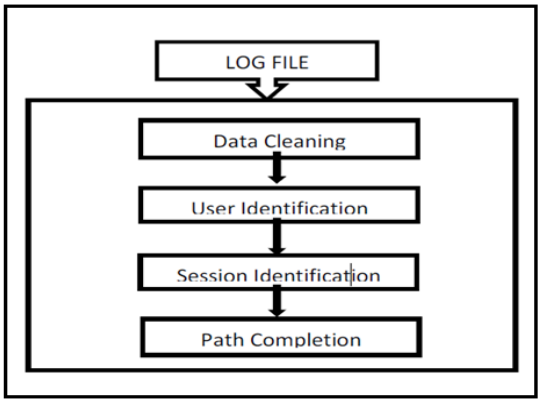

Figure2: Steps for Data Pre-processing

So the list of our weblog file is, the number with each link just show us the number of times that website appear in the link.

http://imke2012.chitkara.edu :::::: 40
http://its.edu :::::: 40
http://imke2012.chitkara.edu :::::: 40
http://its.edu :::::: 40
http://images.sbbs.edu :::::: 30
http://www.isi.edu :::::: 15
http://www.grad.buffalo.edu :::::: 12
http://www.chem.uwec.edu :::::: 9
http://www.chitkara.edu :::::: 8
http://phobos.ramapo.edu :::::: 4
http://www.cs.brown.edu :::::: 4
http://phobos.ramapo.edu :::::: 4
http://www.cs.brown.edu :::::: 4
http://www.facebook.com :::::: 3384
http://ad.yieldmanager.com :::::: 1015
http://l.yimg.com :::::: 936
http://www.google-analytics.com :::::: 910
http://www.bhaskar.com :::::: 724

http://armdl.adobe.com :::::: 722
http://ijeir.org :::::: 63
http://www.mozilla.org :::::: 55
http://www.java-forums.org :::::: 55
http://www.mozilla.org :::::: 55
http://www.java-forums.org :::::: 55

The methodology we use for the project is based on the web cleaning and web fetching in web cleaning we are provided with the web log file that contain the entries near about 65,536according to the domain based we have to clean the url which means that the url contains .jpeg, .php and other extension we have to clean all the content so that we are just left with the required url that we are looking:-
Example: https://mail.google.com/mail/?shva=1#inbox/13f8e31710a80b22
We have to clean this url in the form so that all the other content are removed so we will have http://mail.google.com
We are going to clean the URL according to the domain which means all the domain com, in, ac.in, edu, are going to be considered. We have fetched all the top 10 entries of the all the domain so that our approach is domain based. After fetching all the Top10 entries of the entire domain we are going to collect the data that is of user interest because we are going to use the dynamic approach so that the user requirement will also be considered that why we are also going to collect the data from the user cookies. We are taking the user cookies data from three user and putting that data in to the database now the user cache will contain the data from the user cookies and the data from the top 10 domain that we fetched here we are taking the top entries of from the user cookies considering also the data that is used by all the three user and appear less in the cookies. We are going to fetch that type of the data by using apriori algorithm so that we will also get the data that is used by the entire three users, the user cache contain the following data.
1) Top 10 Domains.
2) Top user cookies.
3) Data from user cookies by applying Apriori.

The size of the user Cache is minimum 25 for the entire three users.
Here we are going to use Two Approach
1) **Without Priority.**
2) **With Priority.**

Without Priority:-In this module we are having three user .user1, user2, user3.the cache for the three user is 25 in size which means the if the size of the user requirement will increase the 25 in that case the pages for the user the most frequent pages will be displayed in the buffer.

Example: If the user1 is having 19 pages those pages will be displayed in the cache. Now if the user 2 will request for the page and he is having 14 pages his only 6 pages will brought in the cache. If now again user1 request for the pages in this case the pages will be repaved with the most frequent pages from the database that is stored with the timestamp in the data base..The graph for the user1, user2, user3 will be based on the hit ratio that is the no of time the user entered in the text box to the no of time the match occur for the user request.

With Priority: This module is working according to the Priority which means that the one of the user is having highest priority other will have less priority.

Example: Suppose we are having three users that are HOD, Staff, and Student. So every time when three users are at the same time login the pages for the HOD will be fetched always as compare to the staff and the student so the priority from that user will be highest.
So every time whenever the requests for the entire three users are entered it will check that there is anything entered from user3 if yes than first pages for the user3 are fetched. than the page for the user1 and then the page from the user2.if the pages for user1 and user2 are already fetched and then the user3 come in for the pages in that case the pages are replaced either the recently used pages for the user three and the hit ratio for the user three is maximum because we are considering all the data for user three, the data is matched from the cache in case of priority and non priority. Every time the data is brought in the cache that is stored in the buffer according to the time stamp, whenever the user enter any page if that page is already exists than its timestamp is changes else that page is added to the database.

IV. APRIORI ALGORITHM

The naive method of finding large itemsets would be to generate all the subsets of the universe of m items, count their support by scanning the database, and output those meeting minimum support criterion. It is not hard to see that the naive method exhibits complexity exponential in m, and is quite impractical. Apriori follows the basic iterative structure discussed earlier. However the key observation used is that any subset of a large itemset must also be large. During each iteration of the algorithm only candidates found to be large in the previous iteration are used to generate a new candidate set to be counted during the current iteration. A pruning step eliminates any candidate which has a small subset. The algorithm terminates at step t, if there are no large t-itemsets. The general structure of the algorithm is given in figure 3,

$L_1 = \{\text{large 1-itemsets}\};$
for $(k = 2; L_{k-1} \neq \emptyset; k++)$
  /*Candidate Itemsets generation*/
  $C_k = \text{Set of New Candidates};$
  /*Support Counting*/
  for all transactions $t \in \mathcal{D}$
    for all k-subsets s **of** t
      if $(s \in C_k)$ $s.count++;$
  /*Large Itemsets generation*/
  $L_k = \{c \in C_k | c.count \geq \text{minimum support}\};$
Set of all large itemsets $= \cup_k L_k;$

Figure 3**:** The Apriori Algorithm

The size of the hash table, also called the *fan-out*, is denoted as . All the itemsets are stored in the leaves. To insert an itemset in, we start at the root, and at depth *d* we hash on the *d*-th item in the itemset until we reach a leaf. If the number of itemsets in that leaf exceeds a *threshold* value, that node is converted into an internal node. We would generally like the fan-out to be large, and the threshold to be small, to facilitate fast support counting. The maximum depth of the tree in iteration *k* is *k*.To count the support of candidate *k*-itemsets, for each transaction *T* in the database; we form all *k*-subsets of *T* in lexicographical order. This is done by starting at the root and hashing on items *0* through *(n-k+1)* of the transaction. If we reach depth *d* by hashing on item *i* then we hash on items *i* through *(n-k+1) +d*. This is done recursively, until we reach a leaf. [9]

V. CONCLUSION

In this paper, the performance of the web logs fetching has been discussed and developed according to the user of the different domains. The main focus was to show the performance of pre-fetching process with priority. Database of university process has been taken for experimentation and data cleaning process has been done on the database so that the useful data can be fetched and unwanted and repeated data can be removed. We have done with cleaning of the URL according to the domain which means all the domain com, in, ac.in, edu, has been considered. We have fetched all the top 10 entries of the all the domain so that our approach is domain based. After fetching all the top 10 entries of the entire domain . We have collected the data that is of user interest because we has used the dynamic approach so that the user requirement has been considered that why we have collected the data from the user cookies. We have fetched that type of the data by using apriori algorithm so that we also get the data that is used by the entire three users, the user cache contain the following data.

REFERENCES

[1] P. Somrutai, “Improving the Performance of a Proxy Server using Web log mining,” M.S. thesis, San Jose State University, 2011.
[2] Alexandros Nanopoulos, Dimitrios Katsaros, and Yannis Manolopoulos, “Exploiting Web Log Miningfor Web Cache Enhancement,” LNAI 2356, Springer-Verlag Berlin Heidelberg, pp. 68–87, 2002.
[3] Dzitac, “Advanced AI techniques for web mining,” Proceedings of the 10th WSEAS international conference on Mathematical methods, computational techniques and intelligent systems. Cor fu Greece. 2008.
[4] J. B. Patil and B. V. Pawar, “Improving Performance on WWW using Intelligent Predictive Caching for Web Proxy Servers,” IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 1, January 2011.
[5] V. Sathiyamoorthi, V. Murali Bhaskaran,“Improving the Performance of Web Page Retrieval through Pre-Fetching and Caching using Web Log Mining”, European Journal of Scientific Research, Vol.66, No.2, pp. 207-218, 2011.
[6] TIAN Meirong and CHEN Xuedong, “Application of Agent Based Web Mining in E-Business”, 2nd international Conference on Intelligent Human-Machine Systems and Cybernetics, In IEEE, 2010.[6]
[7] S.K.Madria, S.SBhowmick, W.K, Ng, and E.P.Lim.”Research issues in web data mining”, In Proceedings of data ware housing and knowledge Discovery, first International conference, DaWak’99, pages 303-312, 1999.
[8] Cooley, R. Mobasher, B. and Srivastave, J. (1997) “Web Mining: Information and Pattern Discovery on the World Wide Web” In Proceedings of the 9th IEEE International Conference on Tool with Artificial Intelligence
[9] Rakesh Agrawal and Ramakrishnan Srikant Fast algorithms for mining association rules in large databases. Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, pages 487-499, Santiago, Chile, September 1994.

