RESEARCH ARTICLE OPEN ACCESS

Reducing Web Latency through Web Caching and Web Pre-Fetching

Rupinder Kaur[1], Vidhu Kiran[2]
Research Scholar[1], Assistant Professor[2]
Department of Computer Science and Engineering
Jan Nayak Ch. Davilal College of Engineering, Sirsa
Haryana - India

ABSTRACT

The world wide web has evolved in less than two decades as the major source of data and information access, for all domains, but today's users often experience long access latency due to network congestion. Web mining can alleviate this problem, as web mining aims to discover new, relevant and reliable information by investigating the web structure, its contents and its usage. Our Web mining is the combination of two approaches that is Web Caching and Web Pre-fetching. Caching can reduce load on both the network and servers, and improve access latency. In web pre-fetching scheme, 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 to improve the performance of web based applications.

Keywords:- Web Mining, Caching, Pre-Fetching

I. INTRODUCTION

The World Wide Web has become today not only an accessible and searchable information source but also one of the most important communication channels, almost a virtual society. Web mining process is similar to data mining, the techniques, algorithms and methodologies used to mine the web encompass those specific to data mining [1,4].Web mining is the process of discovering useful information or knowledge from hyperlink structure, pages content and data usage. In this project Web logs are analyzed and the access patterns of the users are identified .Here we mean by access pattern, the information regarding web pages most widely visited by a user in a website. Users are grouped into different categories based on similarity in the access pattern. Hence, whenever a user logs into the site, by analyzing their access patterns, the pages liked by them is determined. These pages can then be pre-fetched and cached for considerably increasing the access speed of the pages to the user.

II. CATEGORIES OF WEB MINING

There are three web mining tasks: web structure mining, web content mining, web usage mining.

A. ***Web Structure Mining*** – It is about discovering knowledge from hyperlinks. Important web pages can be identified; also users that have common interests, that means who are using same clusters of linked pages. Pages are ranked according to their prestige or authority.

B. ***Web Content Mining*** – It aims to extract useful information or knowledge from the content of web pages. Pages can be clustered and classified based on their topics, patterns concerning user's opinion on different projects or forms postings can also be found from unstructured texts that have been generated by the user.

C. ***Web Usage Mining*** – It aims to automatically discover and analyze patterns in click stream and associated data collected or generated, as a result of user interactions with web resources, on one or more web sites. Behavioral patterns and profiles of users interacting with a website are captured, modeled and analyzed in order to improve services.

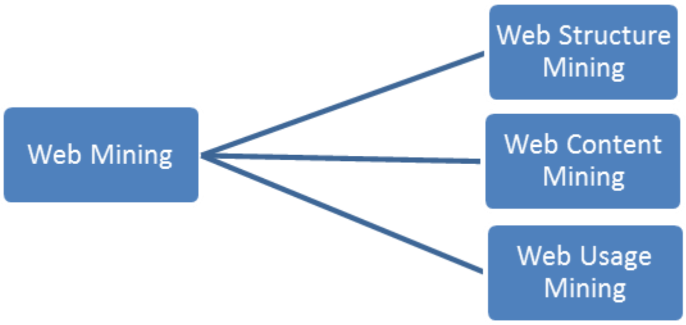

Figure 1 Categories of Web Mining

III. RELATED WORK

Clustering users based on their Web access patterns is an active area of research in Web usage mining. R. Cooley et al. [1997] propose taxonomy of Web Mining and present various research issues, techniques and future directions in this field. Phoha et al. use competitive neural networks and data mining techniques to develop schemes for fast allocation of Web pages [2002]. M. N. Garofalakis et al.[1999] review popular data mining techniques and algorithms for discovering Web, hypertext, and hyperlink structure. Y. Fu et al. [1999] present a generalization based clustering approach, which combines attribute oriented induction, and BIRCH [1996] to generate hierarchical clustering of Web users based on their access patterns. I. Cadez et al. [2002] uses first-order Markov models to cluster users according to the order in which they request Web pages. The Expectation Maximization algorithm is then used to learn the mixture of first-order Markov models that represent each cluster. Although the algorithms and techniques discussed in this section succeed in grouping the users' according to their diverse interests, they lack the ability to adapt to the changes in users' Web interests over time. Pre-fetching means fetching the URL objects before the users request them. For a pre-fetching scheme to be effective there should be an efficient method to predict user's requests. An efficient pre-fetching scheme effectively reduces the user perceived Web latency. However, an inefficient pre-fetching technique causes wastage of network resources by increasing the Web traffic over the network, which in turn increases Web latency. Loon and Bhargavan [1997] present an approach for pre-fetching URLs based on users' profiles. Each user's profile is represented by a weighted directed graph in which the nodes represent URLs and the edges represent the access paths. The weight of anode represents the frequency of access of URLs and the weight of an edge represents the frequency of access of one URL after another. This weighted directed graph is used to predict the user's request. Ibrahim and Xu [2000] present a context specific pre-fetching technique, which uses an artificial neural network for predicting users' requests. Li Fan et al. [1999] investigate an approach to reduce Web latency, by pre-fetching between caching proxies and browsers. Their technique uses the Prediction by Partial Matching (PPM) algorithm for pre-fetching. The prediction accuracy of PPM ranges from 40% to 73%, and generates an extra traffic ranging from 1-15%. Evangelos and Chronaki [2000] present a simple and effective Top-10 approach for pre-fetching. In their approach, the ten most popular Web pages are pre-fetched. The authors show that the Top-10 approach accurately predicts 60% of the future requests. Padmanabhan and Mogul [1996] present a pre-fetching scheme in which the server computes the likelihood that a particular Web page will be accessed and conveys this information to the client. The client program then decides whether to pre-fetch the Web page. The prediction is based on a dependency graph similar to the one used in [1997]. The authors conclude that their methodology results in substantial reduction in Web latency, but increases the traffic on the network. Tian, Choi, and Phoha [2002] present an intelligent and adaptive neural network predictor, which uses the back propagation learning rule to learn the changing access patterns of pages in a Web site. Most of the research discussed in pre-fetching concentrates on pre-fetching individual users' requests according to their previous access patterns. Although these methods are efficient for pre-fetching, they may considerably overload the network with unnecessary traffic when pre-fetching for a large number of users. To reduce such an effect of pre-fetching, we present a pre-fetching scheme that uses ART1 clustering technique to pre-fetch requests for a large community of users instead of pre-fetching individual users' requests. Caching and pre-fetching have often been studied as separate tools for reducing the latency observed by the users in accessing the Web. Less work has been done on integration of caching and pre-fetching techniques. Kroeger et al [1997] study the combined effect of caching and pre-fetching on end user latency. Yang and Zhang have proposed an Integrated Pre-fetching and Caching Algorithm using a Correlation-Based Prediction Model [2000, 2001]. Lan et al. [2000] have proposed a Rule- Assisted Pre-fetching in Web-Server Caching. Yang et al. [2001, 2001] have proposed a method for Mining Web Logs to obtain a Prediction Model and using the model to extend the well known GDSF caching policy. Curcio, Leonardi, and Vitaletti [2001] have proposed an integrated Pre fetching and caching for the World Wide Web via User Cooperation. There has been extensive theoretical and empirical work done on exploring web caching policies that perform best under different performance metrics. Many algorithms have been proposed and found effective for web proxy caching. These algorithms range from simple traditional schemes such as Least-Recently Used (LRU), Least-Frequently Used (LFU), First-In First-Out (FIFO), and various size-based algorithms, to complex hybrid algorithms such as LRU-Threshold, which resembles LRU with a size limit on single cache elements, Lowest-Relative Value (LRV), which uses cost, size and last reference time to calculate its utility, and Greedy Dual [1991], which combines locality, size and cost considerations into a single online algorithm. Several studies have proposed and empirically evaluated variants on Greedy Dual [1997, 2000, and 2001] that incorporate a combination of long term popularity and frequency of access. In addition to these factors, some other studies have focused on the role of aging in web cache replacement policies [18]. Finally, theoretical work has employed randomization to derive new optimal replacement policies [2003].

IV. PROBLEM DEFINITION

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 are given as under:

a. Develop an optimized technique for optimizing the web caching and web pre-fetching processes.

b. To 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 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 nearly 60% and the term that appears less than 60% will be removed and the more combination is applied to take the proper frequent set of the given data.

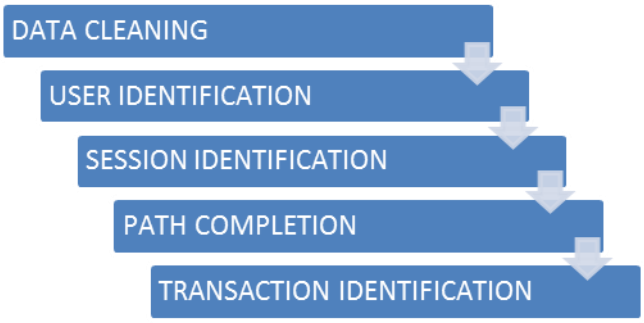

Figure 2 Steps for data Pre-Fetching

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://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 nearly 65,536 according 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/13f8e31710a8 0b22 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 since 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 is 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 into the database now the user cache will contain the data from the user cookies and the data from the top 10 domain that we have fetched here we are taking the top entries 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 entirely by three users, the user cache contain the following data.

a. Top 10 Domains.
b. Top user cookies.
c. 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 Approaches

a. Without Priority

b. With Priority

A. Without Priority:-In this module we are having three users i.e. user1, user2 and 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 most frequent pages will be displayed in the buffer for user.

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

V. RESULT

The result of the concept shows the priority process of the web caching and pre-fetching of the information as per user priority. The main focus was to show the pre-fetching with priority. Most of the online users want faster processing of web services which requires lighting fast processing of web logs. Pre-Fetching based on priority and higher ranking can provide a good solution to the need of internet. Our work includes providing the priority to the number of various users and also done fetching according to the different communities of users. Moreover, in our research we have provided a concept of priority based pre-fetching of web logs.

VI. 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's 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.

ACKNOWLEDGEMENT

I wish to express my heartily gratitude towards those who provided contribution towards this research paper and I profusely thank *Mrs. Vidhu Kiran, Assistant Professor* at JCDCOE, Sirsa for her valuable inputs, comments, sharing her knowledge, expert guidance and for her keen interest, valuable sugges*tions and persistent encouragement.*

REFERENCES

[1] Cooley R., Mobasher B., and Srivatsava J., "Web Mining: Information and Pattern Discovery on the World Wide Web.*" ICTAI'97, 1997.*

[2] Phoha V. V., Iyengar S.S., and Kannan R., "Faster Web Page Allocation with Neural Networks," IEEE Internet Computing, Vol. 6, *No. 6, pp. 18-26, December 2002.*

[3] Garofalakis M. N., Rastogi R., Sheshadri S., and Shim K., "Data mining and the Web: past, present and future." In Proceedings of the second international workshopon Web information and data management, *ACM, 1999.*

[4] Fu Y., Sandhu K., and Shih M., “Clustering of Web Users Based on Access Patterns.” International Workshop on Web Usage Analysis and User Profiling *(WEBKDD’99), San Diego, CA, 1999*.

[5] Zhang T., Ramakrishnan R., and Livny M., “Birch: An Efficient Data Clustering Method for Very Large Databases.” In Proceedings of the ACM SIGMOD Conference on Management of Data, *pages 103-114, Montreal, Canada, June 1996.*

[6] Cadez I., Heckerman D., Meek C., Smyth P., and Whire S., “Visualization of Navigation Patterns on a Website Using Model Based Clustering.” Technical Report MSR-TR-00-18, Microsoft Research, *March 2002*.

[7] Loon T. S., and Bharghavan V., “Alleviating the Latency and Bandwidth problems in WWW Browsing.” In Proceedings of the USENIX Symposium on Internet Technologies and Systems (USITS '97), *December 1997*.

[8] Ibrahim T. I., and Xu C. Z., “Neural Nets based predictive Pre-fetching to tolerate WWW Latency”. In Proceedings of the 20th International Conference on Distributed Computing Systems, IEEE, Taipei, Taiwan, Republic of China, *April 2000.*

[9] Fan L., Cao P., and Jacobson Q., “Web Prefetching between Low-Bandwidth Clients and Proxies: Potential and Performance.” In Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS’99) *Atlanta, GA, May 1999.*

[10] Markatos E. P., and Chronaki C. E., “A Top-10 Approach to Prefetching on the Web.” In Proceedings of the Eighth Annual Conference of the Internet Society (INET'98), Geneva Switzerland, *July 1998.*

[11] Padmanabhan V. N., and Mogul J. C., “Using Predictive Prefetching to Improve World Wide Web Latency.” ACM Computer Communication *Review, Vol. 26, No.3,page 2336, July 1996*.

[12] Tian W., Choi B., and Phoha V. V, “An Adaptive Web Cache Access Predictor Using Neural Network.” In Proceedings of the15th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, *pages 450-459,IEA/AIE, Cairns, Australia, June 2002.*

[13] O. Bahat and A. Makowski. Optimal replacement policies for non-uniform cache objects with optimal eviction. In IEEE INFOCOM Conference on Computer Communications, *April 2003.*

[14] P. Cao and S. Irani. Cost-aware WWW proxy caching algorithms. In USENIX Symposium on Internet Technologies and Systems, *December 1997*.

[15] S. Jin and A. Bestavros. Popularity-aware greedy dual-size web proxy caching algorithms. In 20th Intl. Conf. on Distributed Computing Systems, *April 2000.*

[16] S. Jin and A. Bestavros. Greedydual* web caching algorithm: Exploiting the two sources of temporal locality in web request streams. International Journal on Computer Communications,*24(2):174–183, February 2001*.

[17] N. Young. On-line caching as cache size varies. In 2nd ACMSIAM symposium on Discrete algorithms, pages *241–250, 1991*.

[18] E. Cohen and H. Kaplan. The age penalty and its effect on cache performance. In 3rd Usenix Symposium on Internet Technologies and Systems, *pages 73–84, 2001*.