Make Web Page Instant: By Integrating Web-Cache and Web-Prefetching

Mahesh Manchanda,

Research Scholar, Department of Computer Science, Gurukul Kangri Mahavidyalaya Haridwar, UK (India)
manchandamahesh@rediffmail.com

Dr. Neena Gupta

Assistant Professor, Department of Computer Application, Kanya Gurukul Mahavidyalaya, Dehradun, Second campus of GKV, Haridwar,
neena71@hotmail.com

ABSTRACT *As the Internet continues its exponential growth, two of the major problems that today's Web users are suffering from are the network congestion and Web Server overloading. Web caching and pre-fetching are well known strategies for improving the performance of Internet systems. Web caching techniques have been widely used with the objective of caching as many web pages and web objects in the proxy server cache as possible to improve network performance. Web pre-fetching schemes have also been widely used where web pages and web objects are pre-fetched into the nearby proxy server cache. In this paper, we present an application of web log mining to obtain web-document access patterns of closely related pages based on the analysis of the request from the proxy server log files.*

Keywords: *Pattern mining, Sequence mining, Graph Mining, Web log mining*

1. INTRODUCTION

The Internet can be considered as a large distributed information system that provides access to shared data objects. As the Internet continues its exponential growth, two of the major problems that today's Web users are suffering from are the network congestion and Web Server overloading. Researchers have been working on how to improve Web performance since the early 90's. Caching popular objects at locations close to the clients has been recognized as one of the effective solutions to alleviate Web service bottlenecks, reduce traffic over the Internet and improve the scalability of the WWW system.

Web caching is a well-known strategy for improving the performance of Web-based system by keeping Web objects that are likely to be used in the near future in location closer to user. The Web caching mechanisms are implemented at three levels: client level, proxy level and original server level [1]. Significantly, proxy servers play the key roles between users and web sites in lessening of the response time of user requests and saving of network bandwidth. Therefore, for achieving better response time, an efficient caching approach should be built in a proxy server.

Unfortunately, the cache hit ratio is not improved much with caching schemes. Even though with a cache of infinite size, the hit ratio is still limited only at the range from 26% to about 45%, regardless of the caching scheme [2],[3]. This is because most people browse and explore the new web pages trying to find new information. In order to improve the hit ratio of cache, Web pre-fetching technique is integrated with web caching to overcome these limitations.

The Web prefetching is another very effective technique, which is utilized to complement the Web caching mechanism. The web prefetching predicts the web object expected to be requested in the near future, but these objects are not yet requested by users. Then, the predicted objects are fetched from the origin server and stored in a cache. Thus, the web prefetching helps in increasing the cache hits and reducing the user-perceived latency.

2. PROBLEM DEFINITION

Web caching has been used to reduce the network traffic by caching web pages at the proxy server level. The work presented in this paper seeks to explore an analysis based pre-fetching scheme to improve the performance of the proxy server. The prefetching scheme interprets the user's access pattern to form a group of closely related pages based on the analysis of the requests from the proxy server's log files. When the user requests a web page that is part of such a group, other related web pages in the same group can be pre-fetched into the proxy server's cache in the expectation that the next set of web pages requested by the web user would be from the pre-fetched web pages. The approach presented in this paper integrates the pre-fetching approach with the web caching scheme with the objective of improving performance of the proxy server. The integrated scheme would increase the performance of the proxy server in terms of the Hit Ratio and the Byte Hit Ratio as opposed to a plain web caching approach.

3. RELATED WORK

Web caching is an important technique for improving the performance of WWW systems. Lying in the heart of caching algorithms is the so-called "page replacement policy", which specifies conditions under which a new page will replace an existing one. The basic idea behind most of these caching algorithms is to rank objects according to a key value computed by factors such as size, frequency and cost. When a replacement is to be made, lower-ranked objects will be evicted from the cache. The performance and effectiveness of such an approach is entirely dependent
on the cache replacement techniques that are used by the proxy server. The most successful replacement algorithm is GDSF[4]. Unfortunately, the cache hit ratio is not improved much with caching approach. Even though with a cache of infinite size, the hit ratio is still limited only at the range from 25% to about 65%, regardless of the caching scheme [2],[3]. This is because most people browse and explore the new web pages trying to find new information. In order to improve the hit ratio of cache, Web pre-fetching technique is integrated with web caching to overcome these limitations.

Web pre-fetching schemes, pre-fetch web pages into the cache in the expectation that the user would request these pages in the future requests. Web pre-fetching is performed based on analysis and in a measured way. In (Lee, An, & Kim, 2009) the authors present a brief discussion about the pre-fetching schemes. Pre-fetching schemes can be classified into two types: short-term pre-fetching schemes and long-term prefetching schemes. In short-term pre-fetching scheme pre-fetches web pages in the cache by analyzing the web cache's recent access history. Based on the analysis, the scheme computes the cluster of closely related web pages and pre-fetches group of web pages from the origin web servers [5]. Various approaches have been suggested for the short-term pre-fetching scheme. In [6], the authors disccuss a Partial-Match (P.P.M) model. On the other hand in Long-term pre-fetching scheme, the popular web pages are identified by analyzing the global access pattern for the web pages [2]. In this scheme, objects with higher access frequencies and without longer update time intervals are more likely to be pre-fetched. Thus, Web pre-fetching is a pro-active approach where the web pages are prefetched into the proxy server cache from the origin web servers. If the web pre-fetching results in the pre-fetching of too many web pages, it may result in a performance decrease, rather than a performance improvement [7].

4. OVERVIEW OF NEW GROUP BASED APPROACH

The complete approach is discussed in detail as follows:

Preprocessing of the proxy server Log File
The raw proxy server log files are unsuitable for access pattern analysis. The proxy server log requires effective preprocessing to remove irrelevant data from the proxy server log file for analysis

Segregation of the Proxy Server Log File
Once we have obtained a processed log file by removing all the irrelevant entries in the web proxy log file, the web pages frequently requested have to be identified. This is a two step process in which, we first identify the closely related websites for each client IP address and then identify the frequently requested web pages within the websites. To achieve this, we need to segregate the web proxy log file in two different ways:

i. Create a separate proxy log file for each client IP address, which stores all the web requests originating from a particular client IP address.

ii. We need to identify the frequently requested web pages within a website by each client IP address. To achieve this, the proxy log file obtained in 1) is divided into separate log

files for each website URL, which would list all the web pages visited within a website URL by the client IP address.

Log files for identifying closely related websites URLs

Inter-website cluster refers to closely related website URLs that belong to the different web servers. To achieve the objective of identifying inter-website clusters for each of the client IP addresses, we first need to identify the transactions for the client IP addresses. Thus, we segregate the web proxy log file according to the originating client IP address. The proxy server log file obtained in this step is used to construct a website traversal graph for the requests made by the client IP address.

Proxy log files for identifying intra-website pages

To achieve the objective of identifying frequently requested intra-website pages or frequently requested web pages within a website for each of the website URLs identified in the inter-website cluster, it is necessary to divide the requests made by client IP address according to the website URLs to which the requests were made for the client IP address.

Algorithm for pre-fetching of web pages

1) Identify the client IP address making the request for the web page.
2) Identify the requested website URL information from the request.
3) Obtain the page information from the website URL.
4) Search if the requested website URL occurs in any of the inter-website clusters for the client IP address. If the website URL occurs in the inter-website group, obtain other inter-website URLs within the same cluster.
5. Obtain the frequently requested intra-website pages for the requested website URL. Check if the web page requested by the client IP occurs in the intra-website cluster for the requested website URL. If not, the request is handled as a routine web request.
6) If the requested web page occurs in the intra-website cluster of web pages for the
website URL, the scheme pre-fetches all the intra-website pages for the other inter-website
URLs. Once all the pages are pre-fetched into the cache, the Least Recently Used (LRU) and Least Frequently Used (LFU) algorithms manage these web pages.

5. DESIGN

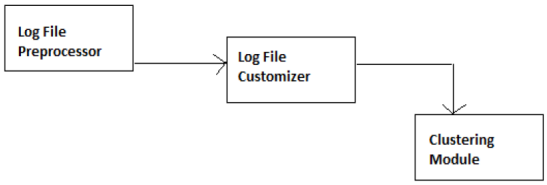

Figure 1: Designing the Grouping Module

The key components in the scheme are the proxy server log file preprocessor, customized log file processor, and the clustering module.

log file processor processes the raw proxy server log file to obtain a processed log file., **Log File Customizer** uses the data obtained from the cleaned proxy log file for sampling. The **Grouping or Clustering module** then processes the customized log files obtained from the customized log file processor to obtain closely related websites using the graph based approach discussed and identifies the frequently requested intra-website pages within the website that are frequently requested[10].

The above process identifies the most popular web pages for each client IP by analyzing the sampling dataset. Once these have been identified, we run the algorithms of Least Recently Used (LRU), pre-fetching based Least Recently Used (LRU) and Least Frequently Used (LFU) and pre-fetching based Least Frequently Used (LFU) algorithms to obtain the results.

6. EXPERIMENT RESULT

The scheme explained in the paper was tested with 2 different datasets. These datasets are obtained from a proxy server installation. The filenames of the datasets used for testing of the schemes are geu.sanitized-access.20070109.har (Dataset 2) and uni.sanitized-access.20070110.har (Dataset 1) under the dummy website.

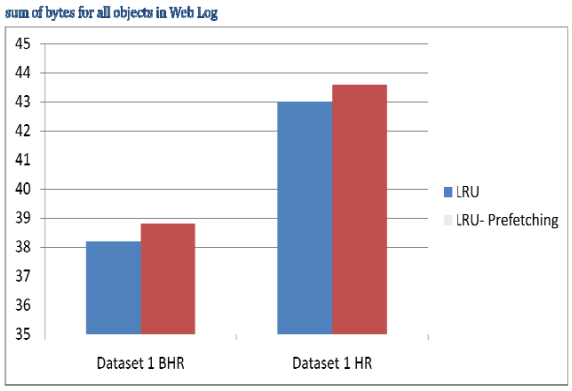

Figure 2: Dataset 1: LRU vs Prefetch LRU

Cache Size 15% of sum of bytes for all object in Web Log. We plotted a graph for Hit Rates and Byte Hit Rates obtained for the LRU and pre-fetching the LRU after testing improves to 38.8% for pre-fetching based LRU when compared to 38.2% for a plain LRU and there is an improvement in Hit Ratio for improve from 43% for LRU to 43.6% for pre-fetching based LRU for the same data set.

Cost of pre-fetching

The total number of items that were pre-fetched into the cache was 3700. Out of these, about 2150 items did not generate cache hits. The total size of the items pre-fetched into the cache was about 1.25% o the total size of the items that the cache was tested against.

7. CONCLUSION

The work presented in the paper discusses a comprehensive approach to analyze web access pattern for users by using the information present in the proxy server log files. Using the approach, we identify frequently requested web pages by the users and integrate the pre-fetching scheme with the web caching to achieve performance improvement for the proxy server cache. In above discussed experiment, we could observe that there was performance improvement in terms of the Hit Ratio and the Byte Hit Ratio.

We also learn that the information obtained from the web proxy log file can contain large number requests of for a particular client IP address. This causes greater pre-fetching for a particular client IP address, which can result in decrease in the performance of the proxy server cache.

REFERENCES

[1] H.T. Chen, Pre-fetching and Re-fetching in Web caching systems: Algorithms and Simulation, Master Thesis, TRENT UNIVESITY, Peterborough, Ontario, Canada(2008).

[2]H.k. Lee, B.S. An, and E.J. Kim, "Adaptive Prefetching Scheme Using Web Log Mining in Cluster-Based Web Systems", 2009 IEEE International Conference on Web Services (ICWS), (2009), pp.903-910.

[3]L. Jianhui, X. Tianshu, Y. Chao. "Research on WEB Cache Prediction Recommend Mechanism Based on Usage Pattern", First International Workshop on Knowledge Discovery and Data Mining(WKDD), (2008), pp.473-476.

[4]J. Domenech, J. A. Gil, J. Sahuquillo, and A. Pont, "DDG: An efficient prefetching algorithm for current web generation," in Proc. of the 1st IEEE Workshop on Hot Topics in Web Systems and Technologies, 2006.

[5] Chen, Y, Qiu, L, Chen, W, Nguyen, L, & Katz, RH. (2003). Efficient and adaptive web replication using content clustering. Selected Areas in Communications, IEEE Journal on 21(6), 979-994

[6] T. Palpanas, Web Prefetching Using Partial Match Prediction, master's thesis, Dept. Computer Science, Univ. Toronto, 1998; available as tech. report CSRG- 76; www.cs.toronto.edu/~themis/publicationsp. html.

[7] X. Chen, X.Zhang "Popularity based prediction model for web prefetching" Published by the IEEE Computer Society, March 2003.

[8] Lou, W, Liu, G, Lu, H, & Yang, Q. (2002). Cut-and-pick transactions for proxy log mining. In: Proceedings of the 8th international conference on extending database technology (EDBT 2002), 88–105.

[9] Pallis, G, Angelis, L, & Vakali, A. (2007). Validation and interpretation of web users' sessions clusters. Science Direct, Information Processing & Management, 43(5), 1348-1367.

[10] Jyoti, Sharma, A, & Goel, A. (2009). A novel approach for clustering web user sessions using RST. Advances in Computing, Control, & Telecommunication Technologies, 2009. ACT, 2(1), 656-661.

[11] J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan, "Web usage mining: Discovery and applications of usage patterns from web data," SIGKDD Explorations, Vol. 1, No. 2, pp. 12-23, 2000