ISSN : 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering
VOL. 1, ISSUE 4, JUNE 2012

Improving the Performance of Web Using Semantic Prefetching Methodology

Dr.K.Ramu,
Professor,
Department of CSE,
D.M.I.Collegs of Engineering,
Chennai, Tamilnadu, India
k.ramu147@gmail.com

Abstract:
The growth of the World Wide Web has emphasized the need for improvement in user latency. One of the techniques that are used for improving user latency is Caching and another is Web Prefetching. Web prefetching is one of the techniques proposed to reduce user's perceived latencies in the World Wide Web. The spatial locality shown by user's accesses makes it possible to predict future accesses based on the previous ones. A prefetching engine uses these predictions to prefetch the web objects before the user demands them. This paper proposed a semantic based prefetching methodlogy to improve the prefetch hit ration and minimize the user access latency so that the user requested data can be retrieved as fast as possible.

Keywords: *Web cache, Web prefetch, Latency, Access time*

I. INTRODUCTION

The World Wide Web (WWW) has emerged as the most widely used tool for information access and dissemination throughout the Internet. Current Web is a huge repository, with its explosion in both the range and quantity of web content. Even, when the speed of Internet has improved with the reduced costs, the traffic is getting heavier. The principle technique used to reduce memory access time is the use of some form of Cache hierarchy. Caching is the storage of recently retrieved computer information for future reference. A request for an object found in the cache is called cache hit, otherwise cache miss. Prefetching is used as an attempt to place data close to the processor before it is required, eliminating as many cache misses as possible. Prefetching is the means to anticipate probable future requests and to fetch the most probable documents, before they are actually requested. Web prefetching is a method for reducing Latencies. This area of research gains importance since, an user always expects an interactive response, better satisfaction and quality of output. The prefetching can be defined as "A reference that misses in the object cache is prefetchable for a peak period if the object is cacheable, exists at least since the beginning of the previous off-peak period and it is not modified since the beginning of the previous off-peak period

The basics of web prefetching techniques preprocess the user requests, before they are actually demanded. Therefore, the time that the user must wait for the requested documents can be reduced by hiding the request latencies. Various approaches and algorithms have been proposed for improving the web performance. Web prefetching is an effective technique used to mitigate the user perceived latency by making predictions about the User's future requests and prefetching them before the user actually demands them.

Now the prefetching can be defined according to World Wide Web as "A Web resource is prefetchable if and only if it is cacheable and its retrieval is safe". A number of commercial systems used today implement some form of prefetching. There are also a number of browser extensions for Netscape and Microsoft Internet Explorer as well as some personal proxies that perform prefetching of links of the current page. Many research papers have been published on the use of prefetching as a mechanism to improve the latencies provided by caching systems. The Web prefetching approaches can be characterized according to its short-term and long-term benefits. In this context, the existing prefetching policies as follows:

A. SHORT-TERM PREFETCHING POLICIES

Future requests are predicted to the cache's recent access history. Based on these predictions, clusters of Web objects are prefetched. In this context, the short-term prefetching schemes use Dependency Graph (DG), where the patterns of accesses are held by a graph and Prediction by Partial Match (PPM), where a scheme is used, adopted from the text compression domain. In addition, several short-term prefetching policies are exist namely Predictive Web prefetching, semantic web prefetching and proxy web prefetching. These prefetching methods are based on Markov models, which are used for modeling and predicting user's browsing behavior over the Web.

B.LONG-TERM PREFETCHING POLICIES

Global object access pattern statistics (e.g., objects' popularity, objects' consistency) are used to identify valuable objects for prefetching. In this type of scheme, the

ISSN : 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering
VOL. 1, ISSUE 4, JUNE 2012

objects with higher access frequencies and no longer update time intervals are more likely to be prefetched. There are various types of long term prefetching exist namely frequency based prefetching, greedy-dual-size prefetching and popularity based prefetching. There are various types of Prefetching techniques exist namely, Web Prefetching , Data Prefetching and Other issues, Web Prefetching is further classified into Cache Prefetching , Proxy Prefetching and Semantic Prefetching , Data Prefetching is classified into Content Prefetching and Context Prefetching.

This paper is organized as follow: Section I gives the introduction about the Web prefetching concepts. Section II is helpful to understand the background of related work. Section III explains the Semantic web prefetching. Section IV show the design of proposed methodperformance of proposed technique. The section V describes about the experimental method. Section VI explains the other research diorections in web prefetching and at last section VII concludes the paper and followed by the references.

II. RELATED WORK

The web prefetching has been applied to a variety of distributed and parallel systems to reduce communication latency.

Davison [1] conducted a detailed analysis that focused on examining the descriptive quality of web pages and the presence of textual overlap in web pages. A keyword-based semantic prefetching approach was designed by Cheng and Ibrahim [2] that could predict future requests based on semantic preferences of past retrieved Web documents. The scheme was evaluated by considering the Internet news services. Neural network was applied over the keyword set to predict future requests. R.Cooley et al [3] developed a quantitative model based on support logic that used information such as usage, content and structure to automatically identify interesting knowledge from web access patterns.

The effectiveness of link-based and content-based ranking method in finding the web sites was analyzed by Craswell et al [4]. The results indicated that anchor texts are highly useful in site finding. Davison [5] proposed a text analysis method that examined web page content to predict user's next request. The algorithm used text in and around the hypertext anchors of selected web pages to determine user's interest in accessing web pages. Chen et al [6] proposed a framework that used link analysis algorithm for exploiting both explicit (hyperlinks embedded in web page) and implicit (imagined by end-users) link structures. The framework had the ability to analyze interactions between users and the web.

III. SEMANTIC WEB PREFETCHING

The Semantic Web Prefetching is responsible for making efficient predictions of web objects to be prefetched for satisfying the user's future requests with low latency. It is based on the concept of client-side prefetching, where the client directly prefetches web documents from the server and stores it in local cache to service user requests. It significantly reduces the latency when servicing user requests, since there is no network latency for retrieving locally cached documents. The prefetching scheme consists of the following components: Tokenizer, Prediction unit, Prefetching unit and the Prefetching cache.

A. *Tokenizer*

When user is viewing a web page, Tokenizer parses the web page to extract anchor links (URL) and its associated anchor text. It then identifies the tokens (keywords) from each anchor text of a link. A token is considered as the meaningful word within anchor text of a URL. When a user clicks anchor link in a web page, then tokenizer moves the tokens of that particular anchor text into user token repository. The repository has collection of tokens with their frequencies, where token frequency indicates the number of times a particular token is seen in the anchor text of links selected by the user. When a token occurs for the first time, new entry is created in the repository with initial count value as 1. For the existing tokens its count value gets incremented. The user token repository is used by the prediction unit to compute probability values of anchor links that are not accessed by the user in a web page.

B.Prediction Unit

Prediction Unit is responsible for computing the probability value of each anchor link using Naïve Bayes Classifier. The advantages of using Naïve Bayes Classifier for computing the probability values are:

a) Simple mechanism to compute values for the specified data

b) Requires minimal storage, since it stores only token counts and

c) Incremental update whenever new data is processed. Anchor text associated with each link is taken and the tokens from it are compared with tokens stored in the user token repository to compute the probability value. The anchor links are then arranged based on the probability value to be given as input to the prefetching unit.

C.Prefetching Unit

The web objects that are required to satisfy the user requests with low latency are retrieved by the prefetching unit from the web server and stored in the local cache. The selection of web objects for prefetching is based on the preference list generated as output by the prediction unit. Prefetching is normally performed when the client is viewing a web page (i.e. 'idle' time). Prefetch requests are given

ISSN : 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering
VOL. 1, ISSUE 4, JUNE 2012

low priority than the regular user requests, so whenever user makes a request the prefetching unit suspends any ongoing prefetch activity. It is possible to prefetch only limited number of links at any time because of small time period to perform prefetch before user makes a new selection.

D.Prefetching Cache

The web documents that are prefetched from the server are stored in the prefetching cache to satisfy the user's future requests. To eliminate the caching impact due to temporal locality exhibited in the user access patterns, prefetching cache is managed separately from the browser's in-built cache. When new web documents need to be stored in the prefetching cache, it selects documents that are not accessed for a long time and purge it whenever there is insufficient space in the cache.

IV. DESIGN OF PROPOSED METHOD

The proposed semantic web prefetching method has the various steps to be achieved in order to improve the performance in terms of hit rate and delay saving ratio. We assume each group contains a fixed number of links and refer to the group size as prefetching breadth. Evaluation of the embedded links by groups is patterned after client browsing behaviors because a client tends to look at a few hyperlinks before he/she decides which to follow. Recall that prefetching needs to be carried out during the time when the client is viewing a page. A recent analysis of client access patterns showed that the average rate that http requests were made was about two requests per minute. The table I shows the different steps involved in implementing Semantic web prefetching.

Each user request is associated with a category. Based on the URL given, the requested data is retrived and shown to the user for verification. Then this method evaluates to identify the fisrt set of links to evaluate. This process is repeated until all the the requests are being processed.

The proposed system was developed entirely using ASP .Net and C# tetchnology and integrated with a The lightweight browser with its open source code gave us a great flexibility in the evaluation of our algorithm. Using such a lightweight browser simplified the process of setting experiment parameters because the browser just did what it was programmed for and there were no hidden functions or Add-Ins to worry about as those with full-blown browsers like Google chrome, Mozilla and Explorer. Such browsers have their own pre-fetching and caching schemes that may conflict with our algorithm. Since the web is essentially a dynamic environment, its varying network traffic and server-load make the design of a repeatable experiment challenge. A trace-based simulation works like an instruction execution trace based machine simulator. It assumes a parameterized test environment with varying network traffic and workload. While there are many web traffic benchmarking and characterization studies, few of the trace data sets available to public provide semantic information. Since the semantics-based prefetching techniques are domain specific, they should be enabled or disabled when clients enter or exist from domain services.

Table I :Steps involved in Semantic Methodology

1. User requests a URL associated with a category.
2. The URL document is retrieved and displayed to the user.
3. The contents of the document are examined to identify the first set of links to evaluate.
4. Each link from the set is processed to extract the keywords.
5. Keywords are matched to the keyword list. They are added to the list if they occur for the first time.
6. The corresponding preferences are computed using the weights associated with the keywords.
7. Links with preferences higher than the neuron threshold are sorted and placed in the prefetching queue.
8. Links are accessed one by one and the objects are placed in the cache.
9. If the prefetching queue is empty, process the next set of links [Step 2].
10. When the user requests a link, the cache list is examined to see if it's there.
11. If it was there, then it is displayed.
12.If web objects are not available in the prefetch cache, then it is retrieved from the web server and displayed to the user.

VI. OTHER RESEARCH DIRECTIONS

Web Prefetching is an elegant technique to handle information provisioning for users under these circumstances. As this is mainly a proposition, the future work involves the implementation of this proposal and determination of constants designed in the proposal. Following the structure of the proposal the implementation will be executed in under mentioned steps.

A. Semantic distance

Experiments must be performed to determine constants for semantic distance. The aim of experiment is to determine

ISSN : 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering
VOL. 1, ISSUE 4, JUNE 2012

which type of semantic measure describes the distance between set of concepts describing web page better for our purpose.

B. Base for keywords selection

The authors in [1] approve that the use of keywords from hyperlink anchor texts is sufficient for document description. Based on experiments with this module the algorithm may be enriched with other sources of keywords used for semantic description. Some pages are already annotated with semantic annotation and also the titles or headlines of Web pages can provide usable keywords. Currently we take into account just the hypernym/hyponym relationship. The experiments may show that more relationships may be used to get better accuracy. The prediction module is the main aim of the whole thesis. The basic proposed method will be refined to achieve the best possible performance. The experiments in this module concerns two fields: estimation of the order of Markov model and determination of thresholds used there.

C.Estimation of the order of Markov model

The main purpose of the whole proposal is to lower and prune basic Markov model to simplify its complexity. The lower the order of Markov model the worse accuracy.

Using the semantic information I want to lower the order as much as possible. Experiments should establish the best proportion of order and efficacy using semantic description.

D.Determination of thresholds

In the proposed method the thresholds are mainly used to determine the boundary where it is profitable to pre-fetch suggested Web page. Again, the experiments should establish the best proportion.

VI. CONCLUSIONS

In this paper we have presented a semantic web prefetching for making the predictions of web objects to be prefetched. The scheme provides efficient predictions when users are visiting web pages that contain information related to a specific topic of interest. Since predictions are done based on the computed probability value of anchor texts, it is essential that user's use this scheme only when browsing the web pages for similar content. In case the user visits web pages in random without looking for specific content, then the scheme will provide weak predictions resulting in unnecessary prefetching of web objects. The prefetching scheme helps to minimize the user access latency when satisfying the user requests by achieving good hit rates across different session durations.

The prefetching option could be enabled or disabled by the user based on his interests. In our work we considered only the anchor texts of URL for making the predictions. In some cases the URL contain anchor texts with either a single token or the anchor texts are missing, which may negatively impact the predictions. This limitation could be handled by considering the text in and around the anchor link for making the predictions. The proposed scheme is evaluated by browsing only the news and university

REFERENCES

[1] B.D. Davison, "Topical Locality in the Web", Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval, 2000.

[2] T.I.Ibrahim, C.Xu, "Neural net based predictive pre-fetching to tolerate WWW latency", Proceedings of the 20th International Conference on Distributed Computing Systems, 2000.

[3] R.Cooley, P.Tan, J.Srivastava, "Discovery of interesting usage patterns from Web data", Lecture Notes in Artificial Intelligence, 1836, pp.163-182, Springer, Berlin, 2000.

[4] N. Craswell, D. Hawking, S.E. Robertson, "Effective Site Finding Using Link Anchor Information", Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval, 2001.

[5] B.D.Davison, "Predicting web actions from HTML content", Proceedings of 13th ACM Conference on Hypertext and Hypermedia, 2002.

[6] Z.Chen, L.Tao, J.Wang, L.Wenyin, W.Ma, "A unified framework for web link analysis"; Proceedings of 3rd International Conference on Web Information Systems Engineering, pp. 63-72, Singapore , 2002.

[7] S. Schechter, M. Krishnan, and M. Smith. Using path profiles to predict HTTP requests. In *Proccedings of the 7th International World Wide Web Conference. Also appeared in ComputerNetworks and ISDN Systems*, 20(1998), pages 457—467.

[8] Jason J. Jung, "Semantic Preprocessing of Web Request Streams for Web Usage Mining", Journal of Universal Computer Science, vol. 11, no. 8 pp. 1383-1396, 2005.

[9] J.J.Jung, "Collaborative web browsing based on semantic extraction of user interests with bookmarks", Journal of Universal Computer Sciences, vol.11, no. 2, 2005

[10] Alexander P. Pons, "Semantic prefetching objects of slower web site pages", The Journal of Systems and Software, Vol.79, pp.1715–1724, 2009.

[11] Payal Gulati, A.K. Sharma, Amit Goel, Jyoti Pandey, "A Novel Approach for Determining Next Page Access," icetet, pp.1109-1113, 2008 First International Conference on Emerging Trends in Engineering and Technology, 2008.

[12] 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., May 1999.

[13] Venkata N. Padmanbhan. "Improving World Wide Web Latency", Technical Report, Computer Science Division, University of California, Berkeley, CA, May, 1995.

[14] Markatos E. and Chironaki C. (2006), "A Top 10 Approach for Prefetching the Web", Proceedings of INET Conference, Geneva.

ISSN : 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering
VOL. 1, ISSUE 4, JUNE 2012

[15] Jin S. and Bestavros A., " Popularity-aware greedydual-size web proxy caching algorithms", Proceedings of the 20th International Conference on Distributed Computing Systems, pp.357-378, 2009.

[16] Zhijie B., Zhimin G. and Yu J. (2009), "A Survey of Web Prefetching", Journal of computer research and development, Vol. 46(2), pp. 202-210.

[17] Jiang Y., Wu M.Y, and Shu W. (2010), "Web prefetching : Costs , benefits and performance", Proceedings of the 11th International World Wide Web Conference, New York,pp.472-483.

Biography

K.Ramu has received his Undergraduate Degree in Computer Science and Engineering from Madras University, in 2002 and the Post Graduate degree in Computer Science and Engineering from Sathyabama University, Chennai in 2005. He has received his PhD in Faculty of Computer Science Engineering from Bharath University, Chennai during October 2011. He has more than 5 publications in National Conferences and international journal proceedings. He has more than 10 years of teaching experience. His areas of interest include Data Mining, Data Structures, Database Management Systems, Distributed systems and Operating systems.