ISSN 2278 – 1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 2, April 2012

Review of Algorithms for Web Pre-fetching and Caching

Sandhaya Gawade [1], Hitesh Gupta[2]

PG scholar in Department of Computer Science and Engineering ,PCST, Bhopal

Professor in computer science and engineering, PCST Bhopal

ABSTRACT— ***Increasing popularity of the World Wide Web over the past few years has imposed a significant traffic burden upon the internet. The World Wide Web may be considered to be a large distributed information systems providing access to shared data. A mass research has done to improve the response time of web based system as the information is distributed over a geographical location. Web caching and pre-fetching are two important approaches used to reduce the noticeable response time perceived by users. An ideal pre-fetching caching scheme is a system that able to predict the next (number of next) requests and pre-load those into the cache .The pre-fetched objects are stored in a local cache to reduce the latency time. This is paper presents survey of algorithms for handling a web caching and pre-fetching***

Keywords—Prediction, Pre-fetching, Web Caching

I. Introduction

The www can be considered as a large distributed information system where users can access to shared data objects. Its usage is inexpensive and accessing information is faster using the www than using any other means. The www has documents that keep to a wide range of interests, for example news, education, scientific research, sports, entertainment, stock market growth, travel, shopping, weather and maps [10].

Predictive web prefetching refer to the mechanism of deducing the forthcoming page accesses of a client based on it past accesses. Web prefetching is the process of deducing client's future request for web document and getting that document in to the cache, before an explicit request is made for them. Prefetching capitalizes on the spatial locality present in request streams, that is, correlated reference for different document and exploit the client's idle time, i.e., the time between successive request the main advantage employing prefetching is that it prevents band-width underutilization and hides part of latency . Web prefetching acts complementary to caching; it can significantly improve cache performance and reduce the user perceived latency [7].

The Web caching aims to improve the performance of web-based systems by storing and reusing web objects that are likely to be used in the near future. It has proven to be an effective technique in reducing network traffic, decreasing the access latency and lowering the server load .Web caching has focused on the use of historic information about web objects to aid the cache replacement policies. These policies take into account not only information about the web-document access frequency, but also document sizes and access costs. This past information is used to generate estimates on how often and how expensive it is for the objects saved in the cache to be accessed again in the near future [8].An important advantage of the www is that many web servers keep a server access log of its users. These logs can be used to train a prediction model for future document accesses. Based on these models, it can obtain frequent access patterns in web logs and mine association rules for path prediction. Incorporate our association-based prediction model into proxy caching and prefetching algorithms to improve their performance [9]. Recently, a few researches used mining techniques to explore the browsing behaviors of users in web services [3].

Web prefetching involves two main steps. First, predictions are made based on previous experience about user's accesses and preferences, and the corresponding hints are provided. Second, the prefetching engine decides what objects are going to be prefetched. The prefetching engine can be located at the web browser or at an intermediate web proxy server. The web server can perform the predictions. they can also be done by the web browser or by an intermediate proxy . In this work, it is assumed that the web server provides hints and the web client prefetchs them [13].

II. Basic Principle

Web prefetching is a technique for reducing web latency based on predicting the next future web objects to be accessed by the user and prefetching them during times. If finally the user requests any of these objects, it will be already on the client cache. This technique takes advantage of the spatial locality shown by the web objects [4,14].

The prefetching technique has two main components: The prediction engine and the prefetching engine. The prediction engine runs a prediction algorithm to predict the next user's request. The prefetching engine decide to prefetch them or not depending on some conditions like available bandwidth .Each engine can work at any element of the web architecture [14].

The predictions (PD) are the number of objects. It predicted by the prediction engine. prefetch request (PR) represents the number of objects prefetched. The number of objects prefetched that are requested later by the user is the prefetch hit (PH). The opposite of the prefetch hit is the prefetch miss (PM), which represents the number of prefetched objects that were never demanded by the user (i.e., extra traffic). Finally, user request (UR) refers to the total amount of objects requested by the user (prefetched or not), and the user request not prefetched (URnP) represents the number of objects demanded by the user that were not prefetched [4].

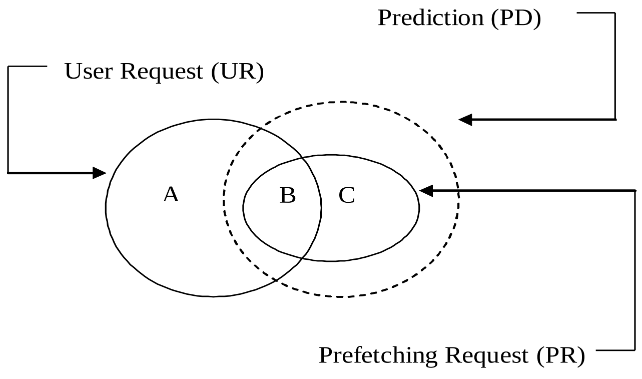

Fig. 1. Web prefetching type of request

As shown in fig. 1, the set of prefetch request (PR) is a subset of the prediction set (PD). The result of the intersection between the user request set (UR) and Prefetch request set is the prefetch hit subset (PH). This subset is the main factor to reduce the perceived latency. In Fig. 1, A represents a user request not prefetched (URnP), which is a user request neither predicted nor prefetched. B is a prefetch request made by the prefetching engine that is requested later by the user, thus becoming a prefetch hit. C is a prefetch miss (PM) resulting from an unsuccessful prediction that was prefetched but never demanded by the user. This request becomes extra traffic and extra server load.

A. Web Prefetching examples

It is easy to visualize the following three prefetching instances in fig. 2 prefetching between web clients and web servers, prefetching between web clients and proxy caches, and prefetching between proxy caches and web servers [16].

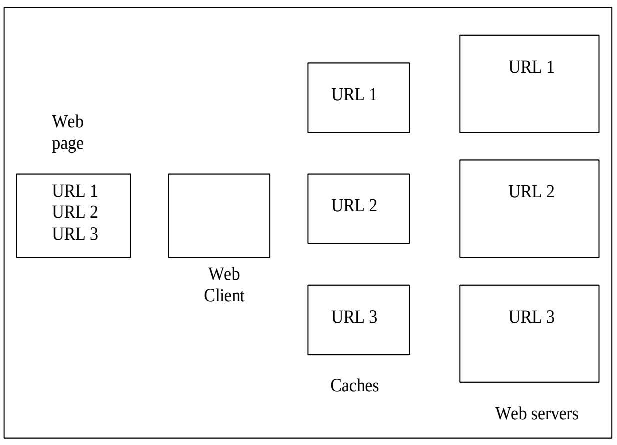

Fig. 2. Prefetching possibilities

Reduce latency: A Proxy server saves the results of all the requests from various clients for a certain amount of time. For instance, consider a case where both users x and y access the www through a proxy server. Let us assume that user x requests for a certain web page say page 1. Sometime later, user y also requests the same page. Instead of forwarding the request to the web server where page 1 actually resides, which can be a time-consuming operation, the proxy server simply returns this page from its cache where all the downloaded pages are retained before being over written by new arrivals. Since proxy server is often on the same network as the user, this is a much faster operation, thereby reducing the perceived latency to some extent [12].

III. Prediction Model Techniques

A. The Prediction Algorithm based on Maximum-Weight-Matrix

The basic idea is to train the machine (caching system) to learn the request pattern from the client side little by little. The learning process [2] is by prediction on next request following the current one. If the prediction is proved to be correct, the corresponding probability is increased. Otherwise, the probability matrix is not changed. In visualization, the training algorithm can be said correct or incorrect. Define the map f as the following:

$$f(n) = c + (s - i + 1)d,$$
$$n = 0, 1, \ldots, N - 1,\ i = 1, 2, \ldots s,$$

here n represents page with index n, i is the index such that page n is in ci and c, d are grater than equal to 0 are constants. We will refer the function f as weight step function, or simply the step-function. Every element is a probability from one page to another. Therefore, all elements are in the range of (0, 1). We modify the probability matrix by eliminating the common denominator for each element. The obtained new matrix is called pre-fetching weight matrix. The initial state of the weight-matrix is (1) N×N. The idea is to simply add a M-W-Matrix (Maximum-weight-matrix) to reduce the search time for the highest weight parameter in a row. At any time, the (1 × N) M-W-matrix contains the biggest parameter at each row and the page index. The M-W Matrix may be updated when a pre-fetching successes and the weight-matrix is updated by applying the step-function. In prefetching cache size 75% and Systems hit rate 70%.

B. Dynamic web pre-fetching

In dynamic web pre-fetching technique, subsequent links are pre-fetched only if bandwidth usage of existing network is less than a predefined threshold. For each web page request, the retrieved page is parsed to identify the subsequent links and URL's corresponding to these links are searched in the hash table to get its weight information. Intelligent agents monitor the bandwidth usage, user's preferences and hash table weights to identify the number of URLs to be pre-fetched.. The main features incorporated into the dynamic pre-fetching model are listed below [5].

In dynamic prefetching cache hit ratio 40% –75% and latency is reduced 20% – 63%.

C. The Matrix Pre-fetching Algorithm

Our idea of predicting is based on the requesting probability to decide which page should be pre-fetched following the current page. The page with the maximum probability is selected as the predictive page that together with the requested page are transferred to the cache. Both pages reside in cache when the request and pre-fetching process are completed. Once the predictive page is proved to be correct in the next request, the pre-fetching is said to be a success, otherwise, it is said a failure. Page request probabilities are recorded as elements of a matrix and are assumed to be uniformly distributed at the beginning (the machine does not know any of them). These probabilities are updated when a new request arrives. The request possibilities are not changed until a success happens. In the case of a prediction success, some of elements in the matrix are updated based on some calculation. As the probabilities in the matrix are modified dynamically (the process of learning), the pre-fetching always picks up the web page corresponding to the maximum probability [6].

In prefetching cache size 10% of the server and Hit rate system without pre-fetching 35.33% system pre-fetching one page, 55.61%.system pre-fetched two pages, 63.58%.

D. Semantic prefetching

"Semantics", hidden in web documents. From certain point of view, the semantics of web document is already considered in history-based prediction. In that case, this semantics is derived from user interest assuming that users passing the same URL-graph are interested in the same thing semantically. They do not consider real semantics of document, however. As semantic prefetching we understand prefetching based on preferences of past retrieved documents in semantics, rather than on the chronological relationships between URL accesses. Semantically based prefetching tries to extract a semantic description of a document and asks server to provide pages with similar semantics, with the same so called "semantic locality". Based on the document semantics, this approach is capable of prefetching documents whose URLs have never been accessed [15].

E. Prediction-based Web Caching

The prediction-based web cache model [11] consists of 3 modules. The ARS log analyzer (LA) is automating processes of identifying usage pattern of client requests. For example, with web cache log information, web usage pattern can be analyzed to identify potential attribute of web objects which has been accessed for classifying usage request pattern. The analyzer also report average response latency and hit/missed statistics to be put in decision maker module name web access miner. The mining module (MM) is an importance engine to classify user requests to explore weight value for rule table which support predict and prefetch future request web objects into web cache server. Prefetching Module (PM) has prefetching engine to check resource usage status of web cache server. The research scope also proposed prefetch concept to enhance with web cache management policy. The detail of such technique will be described in next section. Web cache manager holds the category of replacement policies by enhancing with prefetching. In prefetching hit rate 40- 80% of the system.

F. Adaptive Pre-fetching Scheme Cluster-based System

Our adaptive scheme consists of three components; Double Prediction-by-Partial-Match Scheme (DPS), Adaptive Rate Controller (ARC) and Memory Aware Request Distribution (MARD).The DPS scheme to obtain the relation information of objects and increase the hit rate of pre-fetched data. Also, it proposes the ARC scheme to perform an efficient management of pre-fetch memory in cluster environments. Finally, it suggests the MARD to distribute web workload to improve the efficiency in web pre-fetch [1].

First, we propose a dynamic web prediction scheme called DPS. Web access patterns are dynamic depending on the location of a client. When web objects are stored in an intermediate node, requests to those cached objects do not reach the web server. The DPS scheme solves the problem by providing the addictiveness that handles the client's random access pattern. Second, ARC that provides an adaptive pre-fetch rate at run time. There is a trade-off between consuming memory space and the performance of a web cluster system in modern web frameworks. In multiprocessing environments, web processes allocate memory by their needs. However, we cannot provide the system with unlimited memory, so aggressive pre-fetch schemes can interfere with demand requests from the same client or other clients. For improving the performance of prefetch schemes, the ARC scheme prefetches web objects depending on the memory status.

Our last MARD which distributes incoming requests to the prefetch-enabled backend servers efficiently. Locality based distribution is commonly accepted to improve the performance using the locality of incoming requests. However, non-uniform distribution can use up the memory at the selected backend server and an aggressive prefetch scheme also consumes the memory for prefetching useless objects at the selected server. It can cause the delay in the overall web cluster system. MARD avoids the skewed distribution of requests at the web cluster system [1].

In prefetching scheme improves the performance of web cluster system up to 40% in various web workloads.

IV. CONCLUSION

Web caching and prefetching are well known strategies for improving the performance of internet systems. In this paper, a comprehensive survey of web prefetching and caching is presented. It describes various techniques that reducing successful latency time with the aim of reducing these negative effects at the server side to control the traffic and its impact on the system.

.

REFERENCES

[1] Heung Ki Lee, Baik Song an, and Eun Jung Kim, "Adaptive Prefetching Scheme Using Web Log Mining in Cluster-based Web", International Conference On Web Services, IEEE, 2009, pp. 903-910.
[2] Wenying Feng and Karan Vij, "Machine Learning Prediction and Web Access Modeling", IEEE, 2007, pp. 607-612.
[3] Yin-Fu Huang , Jhao-Min Hsu,"Mining web logs to improve hit ratios of prefetching and caching", Elsevier, 2006.
[4] Johann M´arquez, Josep Dom`enech, Jos´e A. Gil and Ana Pont, "An intelligent technique for controlling web prefetching costs at the server side", International Conference IEEE/WIC/ACM, 2008, pp. 669-676.
[5] Achuthsankar S. Nair, Jayasudha J.S., "Dynamic Web Pre-fetching Technique for Latency Reduction", IEEE, 2007 pp. 202-206.
[6] Wenying Feng and Hua Chen ,"A Matrix Algorithm for Web Cache Pre-fetching" , International Conference IEEE/ACIS, 2007, pp. 788-794.
[7] C. Umapathi and J. Raja,"A Prefetching Algorithm for Improving a web cache Performance", Journal of application science , Asian Network scientific information,2006, pp. 3122-3127.
[8] Qiang Yang and Haining Henry Zhang, "Web-Log Mining for Predictive Web Caching", IEEE 2003, pp. 1050-1053.
[9] Q. Yang, H. H. Zhang and T. Li, "Mining web logs for prediction models in www caching and prefetching", International Conference on ACM, 2001.
[10] Sarina Sulaiman, Siti , Ajith Abraham, Shahida Sulaiman, "Web Caching and Prefetching: What, Why, and How?" IEEE, 2008, pp. 1-8.
[11] Areerat Songwattana,"Mining Web logs for Prediction in Prefetching and Caching", IEEE 2008, pp. 1006-1011.
[12] Payal Gulati, A. K. Sharma, Amit Goel, Jyoti Pandey, "A Novel Approach for Determining Next Page Access", IEEE, 2008, pp. 1109-1113.
[13] B. de la Ossa, J. A. Gil, J. Sahuquillo and A. Pont,"Improving Web Prefetching by Making Predictions at Prefetch", IEEE, 2007, pp. 21-27.
[14] Johann M´arquez, Josep Dom`enech, Jos´e A. Gil and Ana Pont, "A Web Caching and Prefetching Simulator", IEEE, 2008, pp. 346-350.
[15] Lenka Hapalova, Ivan Jelinek, "Semantic web access prediction", International Conference on Computer Systems and Technologies-CompSysTech'07,ACM,2007.
[16] S. V. Nagaraj, "Web Caching and Its Applications", Kluwer Academic Publishers,2004.