Review of Web Pre-Fetching and Caching Algorithms

Arun Pasrija¹ and Shikha Sharma²

¹Computer Faculty at Sinesoft Education, Bathinda, India.
²Computer Faculty at Sinesoft Education, Bathinda, India.

Abstract: Due to the fast development of internet services and a huge amount of network traffic, it is becoming an essential issue to reduce World Wide Web user-perceived latency. Web perfecting is a technique focused on web latency reduction based on predicting the next future web object to be accessed by the user and perfecting it in idle times. So, if finally the user requests it, the object will be already at the client’s cache. In this paper, we analyze the perceived latency versus the traffic increase (both in bytes and in objects) to evaluate the benefits from the user’s perspective. In addition, we also analyze the performance results from the prediction point of view, to provide insights on the observed performance.

Keywords: Web Perfecting; Performance Evaluation; User Perceived Latency, URL, Web Caching, Prediction.

I. Introduction

Web perfecting is an effective tool for improving the access to the World Wide Web. Pre-fetching can be initiated either at the client side or at the server side. The benefit of web perfecting is to provide low retrieval latency for users, which can be explained as high hit ratio. Perfecting also increases system resource requirements in order to improve hit ratio. Resources consumed by prefetching include server CPU cycles, server disk I/O’s, and network bandwidth. The prefetching technique has two main components: The prefetching engine. The prediction engine runs a prediction al-growth to predict the next user’s request and provide these predictions as hints to the prefetching engine. The prefetching engine handles the hints and de-codes to prefect them or not depending on some con-dictions like available bandwidth or idle time. Several ways of prefetching user’s requests have been proposed in the literature: the preprocessing of a request by the server the transference of the object requested in advance and the pre-establishment of connections that are predicted to be made. In this paper, we compare different prediction algorithms and evaluate their performance using both old and current traces.

II. Literature Review

Yin-Fu Huang, Jhao-Min Hsu in 2006 presented a methodology to improve the hit ratios of prefetching and caching. In the paper, they developed an access sequence miner to mine popular surfing 2-sequences with their conditional probabilities from the proxy log, and stored them in the rule table. Then, according to buffer contents and the rule table, a prediction based buffer manager also developed here will make appropriate actions such as document caching, document prefetching, and even cache/prefetch buffer size adjusting to achieve better buffer utilization. The method is as follows:

1) System architecture.
2) Mining mechanisms
3) Log file filter and access sequence miner (ASM).
4) Prediction based buffer manager (PBM).

Through the simulation, they found that their approach has much better performance than the other ones, in the quantitative measures such as hit ratios and byte hit ratios of accessed documents.

Johan Domenech in (2007) the knowledge and comprehension of the behavior 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 behavior permitsto infer and predict future accesses. This information can be used, for instance, for improving Web usability developing on-line marketing techniques or reducing user-perceived latency, which is the main goal of prefetching techniques. These techniques use access predictorsto process a user request before the user actually makes it. Several ways of prefetching user’s requests have been proposed in the literature: the preprocessing of a request by the server, the transference of the object requested in advance, and
the pre-establishment of connections that are predicted to be made. Despite the large amount of research works focusing on this topic, comparative and evaluation studies from the user’s point of view are rare. This fact leads to the inability to quantify in a real working environment which proposal is better for the user. On the one hand, the underlying baseline system where prefetching is applied differs widely among the studies. On the other hand, different performance key metrics were used to evaluate their benefits. In addition, the used workloads are in most cases rather old, which significantly affect the prediction performance, making the conclusions not valid for current workloads.

Johann Marquez, Josep Domenech, Ana Pont, Jose A. Gilin 2008 [3] an intelligent technique for controlling web prefetching costs at the server side. Prefetching is an interesting technique for improving Web performance by reducing the user-perceived latency when surfing the web... In this paper they propose an intelligent prefetching mechanism that dynamically adjusts the aggressiveness of the prefetching algorithm at the server side. To this end, they also propose a traffic estimation model that permits to accurately calculate, in the server side, the extra load and traffic generated by the prefetching. With the aim of reducing these negative effects we have developed an adaptive prefetching mechanism at the server side to control the traffic increase and its impact on the system. Applying the proposed adaptive mechanism, the prefetching technique can be improved since the prediction algorithm requires neither a long period to reach a stable state nor extra resources. Since our proposal proves that the negative effects of web prefetching can be controlled even at the server side, the use of prefetching can be safely spread among users and system administrators.

Sandhya Gawade and Hitesh Gupta in (2012) [4] presented a review of algorithms for prefetching and caching. Increasing popularity of the World Wide Web over the past few years has imposed a significant traffic burden upon the internet. 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 can predict the next (number of next) requests and preload those into the cache. The pre-fetched objects are stored in a local cache to reduce the latency time. The paper presents a survey of algorithms for handling a web caching and prefetching. 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 Image mining is an extension of data mining technique. Most of the image processing algorithms include image mining. Therefore, image mining is always an emerging field and has attracted a lot of researchers to investigate its applications in recent years.

III Proposed Method
We know that web data mining architecture have following steps:
• Integration and merging of sample web log files.
• Pre-processing technique executed in which data Cleaning, user identification and session identification occurs.
• After that pattern discovery.
• Then Pattern Analysis.
• And at last analysis reports.
So, this step is the proposed methodology which we used to make our web-log files more useful and small.

IV. Web Pre-Fetching Techniques
A. Domain Top Approach
Seung Won Shin et al. proposes a domain top approach for web prefetching, which combines the proxy’s active knowledge of most popular domains and documents [11], [13]. In this approach proxy is responsible for calculating the most popular domains and most popular documents in those domains, then prepares a rank list for prefetching.
B. Dynamic web prefetching
In dynamic web pre-fetching technique [12], 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. Intelligent agents are used for parsing the web page, monitoring the bandwidth usage and maintaining hash table, preference list and cache consistency. It controls the web traffic by reducing pre-fetching at heavy traffic and increasing pre-fetching at light traffic. Thus it reduces the idle time of the existing network and makes the traffic almost constant. A hash table is maintained for storing the list of accessed URLs and its weight information [12], [13]. Depending upon the bandwidth usage and weights in the hash table, the prediction engine decides the number of URLs to be pre-fetched and gives the list to pre-fetch engine for pre-fetching the predicted web pages. After pre-fetching, the proxy server keeps the pre-fetched web pages in a separate area called pre-fetch area.

C. Link Pre-fetching
A web page provides a set of pre-fetching hints to the browser and after the browser finishes loading the page, it starts pre-fetching specified documents and stores them in its cache. When the user visits one of the pre-fetched documents, it can be served up quickly out of the browser’s cache. Fisher et al. proposed a server driven approach for ink prefetching [15]. In this approach browser follows special directives from the web server or proxy server that instructs it to pre-fetch specific documents. This mechanism allows servers to control the contents to be perfectly by the browser. The browser looks for either HTML <link> tag or an HTTP Link: header Tag to pre-fetch the subsequent links. The Link: header can also be specified within the HTML document itself by using a HTML <meta> tag [16]. When the browser is idle, it observes these hints and queues up each unique request to be pre-fetched.

D. Top 10 Approach
Evangelos P. Markatos et al. proposes a top 10 approach to prefetching on the web, in which the server calculates the list of most popular documents [2]. This approach is easy to implement in client server architecture. It considers frequency of access for predicting the web object, not the client characteristics on the web.

E. Model based Predictive Pre-fetching
Yang et al. proposed a model based predictive pre-fetching, in which an integrated web-caching and web-pre-fetching model is used [22]. The prediction model used in this is based on the statistical correlation between web objects. The prediction model is time based, prediction window represents some specific time period than number. The algorithm constructs a logical graph called correlation graph, which shows the correlation between web objects and pre-fetch web objects that are highly correlated to a currently requested object. They developed an integrated caching and prefetching algorithm, Pre-GDF. This algorithm is based on the algorithms GD-Size [23] and its enhancement GDSF [24]. The key components in the algorithm are replacement manager, pre-fetching agent, prediction queue and cache.

F. Adaptive pre-fetching Scheme
Adaptive pre-fetch scheme are developed to adapt user’s browsing history and habits [25]. Jiang and al. proposed an adaptive pre-fetch scheme, in which the number of files to be pre-fetched depends on user access history and network conditions [26]. This scheme consists of two modules: prediction module and threshold module. The prediction module updates the history and computes the access probability of each file. Files whose access probabilities greater than or equal to there-prefetch threshold are only pre-fetched. Chen and et. al [27] proposed an adaptive pre-fetch scheme, in which dynamically adjust the pre-fetch aggressiveness in web servers and uses a threshold to adjust the aggressiveness of pre-fetching. Fagni and et. al [28] proposed uses...
G. 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].

V Conclusion
In these papers, evaluation of web prefetching algorithms has been studied. Each algorithm has its own advantages and disadvantages and each algorithm has its own application area. Applying the proposed adaptive mechanism, the prefetching technique can be also improved since the prediction algorithm requires neither a long period to reach a stable state nor extra resources. Since our proposal proves that the negative effects of web prefetching can be controlled even at the server side, the use of prefetching can be safely spread among users and system administration.

Table 1: Comparison of web prefetching & caching techniques.

<table>
<thead>
<tr>
<th>Author’s Name &amp; Paper Name</th>
<th>Techniques</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yin-Fu Huang et al 2006</td>
<td>Web access prediction, caching, prefetching</td>
<td>Much better performance for hit ratio and byte ratio of access documents</td>
</tr>
<tr>
<td>Josep Domenech et al. 2007</td>
<td>Cost-benefit analysis, latency reduction and</td>
<td></td>
</tr>
</tbody>
</table>

A user-focused evaluation of web prefetching algorithms | Prefetching algorithm | reduce prediction of pages |
Johann Marquez 2008 | An intelligent technique for controlling web prefetching costs at the server side | Prefetching algorithm, traffic estimation model | Better traffic estimation and adaptation mechanism |
Sandhya Gwade et al. 2010 | Review of algorithms for Prefetching and caching | Web access prediction, caching, prefetching | Reducing latency time to control the traffic and its impact on the system |

VI. References
[10] Johann Marquez,Josep Domenech, Ana Pont, Julio ,Jose A.Gil ,“ An intelligent technique for controlling web prefetching costs at the server side” 2008 IEEE.

Arun Pasrija received his B.Tech degree in Computer Science & Engineering from Guru Gobind Singh College Of Engineering & Technology, Talwandi Sabo, Punjab in 2011 and M.Tech. (Regular) degree in computer engineering from Yadawindra College of Engineering Punjabi University Guru Kashi Campus Talwandi Sabo (Bathinda), in 2013. His research interests include web prefetching and caching techniques.

Shikha Sharma received his B.C.A degree from SSD WIT, Punjabi University in 2010 and M.C.A (Regular) degree from Yadawindra College of Engineering Punjabi University Guru Kashi Campus Talwandi Sabo (Bathinda), in 2013. Her research interests include web prefetching and caching techniques.