Review on Various Techniques of Web Prefetching and Web Caching Techniques

C. Anuradha, Sundararajan. M, Arulselvi S
Assistant Professor, Department of Computer Science, Bharath University, Chennai, TamilNadu
Director, Research Center for Computing And Communication, Bharath University, Chennai, India
Co-Director, Research Center for Computing And Communication, Bharath University, Chennai, India

ABSTRACT: The continued increase in the growth and popularity of the internet imposes a heavy network traffic in the internet. The users nowadays demand a very high speed in accessing and downloading of web data. Various web prefetching and web caching techniques are there to reduce the time taken to access the web objects. Web prefetching is the method of prefetching the web objects in advance before the user access based on various kinds of prediction. This paper focuses on various techniques available for web prefetching and web caching.

KEYWORDS: Internet, web objects, prediction, prefetching

I. INTRODUCTION

Web prefetching is the method of prefetching the web objects well in advance before a client makes a request for a web page to a web server. Various methods of predictions are available and by using any of these, web objects can be stored in web caches. When a request is made by a client, the request can be satisfied by the web cache itself rather than a web server. The result is reduced latency. The user perceived latency is greatly reduced. The web traffic is also greatly minimized. Web caching focuses on improving the performance of web based systems. It reduces the user perceived latency as well as the server load. Prefetching should be done for web objects which will be requested in the near future. A web prefetching algorithm is considered a successful one when the hit rate is higher. The hit rate is higher when most of the user requests can be satisfied from a web cache. The cache miss results when the user’s request cannot be satisfied from a web cache. The web prefetching algorithms predict user accesses based on various criteria. Most of these algorithms predict based on user’s past access history. This paper reviews various methods available for prefetching the web objects and the various issues involved in using such schemes.

II. VARIOUS WEB PREFETCHING TECHNIQUES

2.1 ADAPTIVE WEB PREFETCHING SCHEME

The semantic prefetching scheme is used for making the predictions by applying decision tree algorithm to compute the probability value of anchor texts present in a web page. The DT algorithm is a classification and regression algorithm provided by Microsoft SQL Server Analysis Services (SSAS) for use in predictive modeling of both discrete and continuous attributes. DTs are one of the most popular methods that are used for Data Mining purposes. In this paper, we focus on extracting anchor links from a displayed web page, convert the anchor text associated with each anchor link into set to tokens (keywords), apply decision tree algorithm over the tokens to compute probability, generate preference list based on the probability to perform prefetching of web objects. When the user access an anchor link, now the system tries to satisfy the user request using web objects stored in the prefetching cache. Performance of the proposed scheme is evaluated.
by observing the browsing patterns of users in different web sessions. The results show that the hit rate improves when a user is involved in longer browsing sessions and access the web links relevant to a particular topic of interest.

2.2 INTERACTIVE PREFETCHING SCHEME

The World Wide Web can be considered as a page-oriented service. A page which is written in HTML consists of texts, in-line images, and references to other pages. While interacting with the world wide web, the client requests for HTML resource and in-line images and displays them as a formatted page. The prefetching system should act in a page-oriented manner. Users click highlighted text on the page. This means that many requests are to retrieve referenced resources, so prefetching of referenced pages makes faster retrieval of web objects at the client site. Therefore, this prefetching scheme is designed to get these referenced pages. The prefetching is not considered as a recursive copying of all resources. The prefetching technique aims to gather references to other pages by parsing the HTML of the client’s retrieval. The referenced pages (HTML and in-line image) are collected at each request of the client. For example, if page A of Figure 2 is retrieved, the prefetching system collects page B and C. The system should prefetch target pages in a short time until the next request, because the WWW is an interactive service and the interval of its requests is short. In addition, fetching of many resources puts heavy load onto the host. So an effective selection method for prefetching targets is required. We have designed a scheme to prefetch several referenced pages at the top of the retrieval page.

An alternative method is to select prefetching target resources by contents or attributes (content length, type, or date). But this method is not practical, because the prefetching system cannot know these contents and attributes without fetching.

Figure 1: The WWW consists of pages
2.2 PPM Models

PPM models were commonly used in Web prefetching for predicting the user’s next request by extracting useful knowledge from historical user requests. Factors such as page access frequency, prediction feedback, context length and conditional probability influence the performance of PPM models in prefetching. An online PPM model based on non-compact suffix tree was implemented that used maximum entropy principle to improve the prefetching performance. A novel PPM model based on stochastic gradient descent was proposed that defined a target function to describe a node’s prediction capability and then selected a node with maximum function value to predict the next most probable page.

2.3 TOP 10 APPROACH

This approach was proposed by Evangelos P. Markatos. In this, the most popular documents are prefetched and placed in the web cache. It does not depend on the user’s access patterns. The calculation is done by the web server. Every server keeps in track of the accesses made over the web objects. It pushes the most popular top 10 objects to caches whenever they are updated. Since the users tend to access the most popular objects than others, most of the users accesses will be cache hit. It results in a high hit ratio.

2.4 MARKOV MODELS

Markov models were effectively used in Web prefetching by utilizing the information gathered from Web logs. Different techniques were presented for intelligently selecting the parts of different order Markov models to create a new model with
reduced state complexity and improved prediction accuracy. Three schemes of pruning (support, confidence and error) were presented to prune the states of All-K th order markov model. A Markov–Knapsack approach was proposed that combined Multi-Markov Web-application centric prefetch model with a Knapsack Web object selector for enhancing the Web page rendering performance. An integration model was designed to combine clustering, association rules and Markov models to achieve better prediction accuracy with minimal state space complexity. Markov tree was used for effective page predictions and cache prefetching, which used the training data set to construct the tree structure for representing the Web page access patterns of users. Domain ontology provides useful semantic information to be used in next page prediction systems. Two methods were discussed to integrate the semantic information into Markov models for prediction. The methods allowed low order Markov models to make intelligent accurate predictions with less complexity than the higher order models. An approach for Web page prediction through linear regression was proposed that depended on the transition probability and ranking of links in the current Web page for the prediction accuracy.

2.5 SEMANTIC PREFETCHING SCHEME
The proposed scheme uses the decision tree algorithm is used for classification. This scheme 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.

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

2.7 WEB PREFETCHING BASED ON LIFETIME
Prefetching leads to additional network traffic i.e., extra bandwidth consumption. It is always preferable to load the cache with web objects that tend to change infrequently. When the objects are less frequently updated, they need not be downloaded from the server frequently. This may result in less bandwidth consumption.

2.8 GOOD FETCH METHOD
In this prefetching technique, a balance is maintained between the update frequency and the access frequency of the web objects. The objects that are less updated may not be the objects that are frequently accessed. When such objects are prefetched and placed in the web caches, the users requests will not be satisfied from the web caches. But if frequently accessed objects are the most updated ones, they have to be downloaded from the servers very often. This will also lead to high bandwidth consumption. So there should always a proper balance maintained between access frequency and the update frequency.

2.9 WEB COMPANION METHOD
Every request for a Web resource from the browser is routed through a WebCompanion component called Fetcher . WebCompanion inspects its document cache and returns the requested resource if it is stored in the cache[6]. If not, WebCompanion accesses the addressed origin server, fetches the resource, stores it in its cache, and forwards it to the browser. If the resource is an HTML document, an HTML document parser in WebCompanion compiles a list of all embedded hyperlinks. When the process of loading this HTML document and all embedded resources is completed,
WebCompanion activates its prefetching engine and prefetches a subset of the documents addressed by the embedded hyperlinks. All prefetched documents are transferred to WebCompanion's cache. The advantage of using this technique is that it reduces network overhead. But it is limited to prefetching of static web pages.

2.10 PREFETCHING MECHANISM USING APL ALGORITHM

Considering the overall access rate as $a$, the frequency of access to object as $p$ and the average lifetime of this object as $l$, this method uses apl for prefetching decision. The APL for whatever objects that exceed a threshold will be selected for prefetching. The apl value of an object represents the possible number of accesses to this object during its lifetime. The higher this value, the more the chances (and possibly the more the times) this object is accessed during its lifetime. Thus, prefetching such objects seems to have a better effect on improving the overall hit rate. This algorithm also intends to balance the hit rate and bandwidth consumption.

2.11 WEB PRE-FETCHING USING DISPLAY BASED PREDICTION

This scheme constructs link graph, which contains both nodes and links by gathering usage information of a client inorder to analyze the reference patterns and predict the next request based on overall displayed documents in the web browser. The node indicates a HTML document with unique URL and the link represents hyperlink or embed link on it. The access counter of a node is increased by one whenever a user fetches or follows it.

III. CONCLUSION

Lot of research is going on in web prefetching in various directions. In this paper various web prefetching techniques and other directions of web prefetching are analyzed and discussed. The web prefetching scheme focus on the property spatial locality of web objects. These techniques are applied to reduce the network traffic and improve the user satisfaction. Web prefetching and caching can also be integrated to get better performance. At present, web prefetching and caching in mobile environment opens a vast scope in the research area.

REFERENCES

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2015


