EFFICIENT PROXY SERVER CACHING USING WEB USAGE MINING TECHNIQUE ON WEB LOGS
For Improving Hit Rate And Response Time

Dr. A K. Ramani
Department. of Computer Science & Engineering,D.A.V.V. , Indore

Sanjay Silakar, Parul Saxena
Department of Computer Science & Engineering, S.A.T.I . , Vidisha

Keywords: World Wide Web(WWW),Proxy Cache

Abstract: This paper presents a vertical application of web usage mining: efficient web caching for improving the response time , for the internet users ,specially due to increase in number of users of e-commerce on the internet Introducing efficient web caching algorithms that employ predictive models of web requests; the general idea is to extend the cache replacement policies of proxy servers by making it sensible to web access models extracted from web log data using web usage mining. The development of an efficient web caching architecture, capable of adapting its behaviour on the basis of the access patterns of the users/clients. Such usage patterns ,or models are extracted from the historical access data recorded in log files by means of data mining techniques known as web usage mining. The approach that has been studied in the present work is based on sequential patterns technique. In the present work a prototypical system has been designed and developed, which supports web usage mining and simulation of the web caching algorithm. The experimental results of the new algorithm developed for cache replacement technique show substantial improvement over LRU-based cache replacement technique, in terms of Hit Rate and Response Time.

1 INTRODUCTION

The World Wide Web (WWW) continues to grow at an astounding rate in both the sheer volume of traffic and the size and complexity of Web sites as the increase in usage specially due to electronic commerce. The continuous increase in demand for information services in the internet is showing signs of strain. While the internet is highly distributed system, individual data objects most often have only a single source. Host computers and network links can easily become overloaded when a large number of users access very popular data. Proxy-caching is currently a popular way to reduce network band width, server load and to improve response time to the user.

Web Usage Mining implies that data mining techniques are applied to the data from the World Wide Web (WWW) where the data emerges from the web server log files. The Web Usage Mining is aimed at discovering regularities and patterns hidden in web data, and patterns in the structure and content of web resources, as well as in the way web resources are accessed and used. Such usage patterns, or models, are extracted from the historical access data recorded in log files, by means of data mining techniques. A particular Web Usage Mining technique and the development of an efficient web caching architecture, capable of adapting its behaviour on the basis of the access patterns of the clients/users is studied in the present work.

K. Ramani A., Silakari S. and Saxena P. (2004).
EFFICIENT PROXY SERVER CACHING USING WEB USAGE MINING TECHNIQUE ON WEB LOGS - For Improving Hit Rate And Response Time .
In *Proceedings of the First International Conference on E-Business and Telecommunication Networks*, pages 177-182
DOI: 10.5220/0001390601770182

2 EFFICIENT WEB CACHING

2.1 Web Caching

Web caching inherits several techniques and issues from caching in processor memory and file systems. However, the peculiarities of the web give rise to a number of novel issues which call for adequate solutions.(Pitkow,1994)(Brain,2002)(Duane,1995).

Caches can be deployed in different ways, which can be classified depending on where the cache is located within the network. The spectrum of possibilities ranges from caches close to the client (browser caching, proxy server caching) to caches close to the origin server (web server caching). Proxy server caching provides an interface between many clients and many servers. The effectiveness of proxy caching relies on common accesses of many users to a same set of objects. Even closer to the client we find the browser caches, which perform caching on a per user basis using the local file system. However, since they are not shared, they only help the single user activity. Web server caching provides an interface between a single web server and all of its users. It reduces the number of requests the server must handle, and then helps load balancing, scalability and availability.

2.2 Web Cache Replacement strategies

The cache replacement strategy decides which objects will remain in cache and which are evicted to make space for new objects. The choice of this strategy has an effect on the network bandwidth demand and object hit rate of the cache (which is related to page load time).

Caching algorithms are characterized by their replacement strategy, which mainly consists of ordering objects in cache accordingly to some parameters (the arrival order, the request frequency, the object size, and compositions of them): objects are evicted according to such an order.

Various cache replacement strategies have been described and analyzed since processor memory caching was first invented .Such as **FIFO** – First In First Out Strategy (order by arrival time).**LFU** – Least Frequently Used Strategy (order inversely to number of requests). **LRU** - Least Recently Used Strategy (order by last request time). One of the most popular replacement strategies is the Least Recently Used (LRU) strategy, which evicts the object that has not been accessed for the longest time. This strategy works well when there is a high temporal locality of reference in the workload -that is, when most recently referenced objects are most likely to be referenced again in the near future .**SLRU** - (order inversely to ΔT . Size , ΔT being the number of requests since the last request to the object). **LRU-K-**The LRU-K replacement strategy considers both frequency and recency of reference when selecting an object for replacement. **LRU-MIN-** (when the cache has to make room for an object of size S, first it tries to do it by evicting objects of size S or greater in LRU order; if that fails, it tries with objects of size S/2 or more, then S/4 or more and so on).

To summarize, an ideal cache replacement policy should be able to accurately determine future popularity of documents and choose how to use its limited space in the most advantageous way. In the real world, we develop heuristics to approximate this ideal behaviour.

2.3 Measures of Efficiency

The most popular measure of cache efficiency is hit rate. This is the number of times that objects in the cache are re-referenced. A hit rate of 70% indicates that seven of every 10 requests to the cache found the object being requested. Another important measure of web cache efficiency is byte hit rate.

This is the number of bytes returned directly from the cache as a fraction of the total bytes accessed. This measure is not often used in classical cache studies because the objects (cache lines) are of constant size.

However, web objects vary greatly in size, from a few bytes to millions. Byte hit rate is of particular interest because the external network bandwidth is a limited resource (sometimes scarce, often expensive). A byte hit rate of 30% indicates that 3 of 10 bytes requested by clients were returned from the cache; and conversely 70% of all bytes had to be retrieved across the external network link.

Other measures of cache efficiency include the cache server CPU or I/O system utilization, which are driven by the cache server's implementation. Average object retrieval latency (or page load time) is a measure of interest to end users and others.

3 WEB USAGE MINING

3.1 Data Preparation For Web Usage Mining

Web Usage mining is the application of data mining techniques to discover usage patterns, or models, extracted from the web log data

(Robert,1999)(Karuna,1999). There are several preprocessing tasks that must be performed prior to applying data mining algorithms to the data collected from server logs (Robert,1999). Various data preparation techniques are used to identify unique users and user sessions.

3.2 Web Usage Mining Techniques

3.2.1 Association Rules

In the web usage mining ,association rules are used to find out which pages are frequently visited together (J.Han,2000)(Rakesh,1994). An association rule is usually presented as an expression of the form: A⇨ B[S,C], where A and B are set of items, S is the support of the rules ,defined as the rate of transactions containing all items in A and all items in B ;C is the confidence of the rule, defined as the ratio of S with the rate of transactions containing A. A rule such as res1⇨ res2, informally means that if res1 appears in a user session, res2 too is expected to appear in the same session, though possibly in reverse order and not consecutively.

3.2.2 Sequential Patterns

The temporal correlation among items in user transactions are discovered (J.Han,2000). Unlike Association rules ,sequential patterns take into account the time component. Consider a user session as a time-ordered sequence of requested resources. Given a database of user sessions, a sequential pattern is an expression of the form $res_1 \triangleleft res_2$.$\triangleleft$.….$\triangleleft$ res_n [S], where res_i (I ∈(1.., n}) is a resource; S is the support of the sequential pattern, defined as the percentage of user sessions containing the subsequence $res_{1,} \triangleleft$ $res_{2,\ldots} \triangleleft res_{n..}$ In frequent sequential patterns, i.e. we only consider patterns with a support greater than a given minimum threshold.

3.2.3 Clustering

Clustering is a technique to group together a set of items having similar characteristics (J.Han,2000).In the Web Usage domain, there are two kinds of interesting clusters to be discovered : usage clusters and page clusters.

3.2.4 Classification

Classification is the task of mapping a data item into one of several predefined classes (J.Han,2000). In the Web domain, one is interested in developing a profile of users belonging to a particular class or category.

4 PROPOSED STRATEGY

The idea is to extend the LRU-Least Recently Used cache replacement strategy adopted by proxy servers by making it sensible to web access models, extracted from web log data. The present study introduces a way to construct efficient web caching algorithm that employ predictive model of web requests. The goal of the algorithm is to maximise the so-called efficiency (i.e. Hit Rate), namely the percentage of requested web entities that are retrieved directly in cache, without requesting them back to the origin servers as well as minimising the Response Time.

The general idea is motivated by the following observation. The LRU-Least Recently Used cache replacement strategy (evicts from cache the least recently used entities) is based on the assumption that the requests that occurred in the recent past will likely occur in the near future too. LRU is effective in particular, when requests are characterized by temporal locality (like the case of web requests). Now, the more information we can extract from the history of recent requests, the more informed cache replacement strategies can be devised. This is a clear indication to mine the web log data for access models which may be employed to the above purpose.

4.1 Evaluation Criteria’s Used In The Model

Our study focused primarily on (object) Hit Rate, Response Time and Cache size.

HIT RATE: It is the ratio of request fulfilled by the cache and then not handled by the origin server. The hit rate is the percentage of all requests that can be satisfied by searching the cache for a copy of the requested object.

RESPONSE TIME: The time that an end user wait ‘s for retrieving a resource. It is the time taken to fulfil the request.

CACHE SIZE: It is the size of cache i.e. the number of URL’s(requests) that can be stored in a cache.

4.2 Web Usage Mining Technique Used

4.2.1 Web Access Sequential Pattern

A web access pattern is a sequential pattern in a large set of pieces of web logs which is pursued frequently by users. Web access sequential patterns are mined using sequential pattern web usage mining technique.

Sequential pattern mining discovers frequent patterns in a sequence database. Given a sequence database where each sequence is a list of transaction ordered by transaction time and each transaction consists of a set of items, all sequential patterns are found out with a user specified minimum support, where support is the number of data sequences that contain the pattern.

A concise and highly compressed tree structure is used that handles the sequences and an efficient mining algorithm is used for mining the complete (non redundant) web sequential access patterns from large set of pieces of web logs.

4.3 Cache Replacement Strategy Used

LRU - Least Recently Used Strategy (order by last request time). One of the most popular replacement strategies is the Least Recently Used (LRU) strategy, which evicts the object that has not been accessed for the longest time. This strategy works well when there is a high temporal locality of reference in the workload -that is, when most recently referenced objects are most likely to be referenced again in the near future. The LRU strategy is implemented using a doubly linked list ordered by last access time. Addition and removal of objects from the list is done in O(1) (constant) time by accessing only the head (or tail) of the list. It evicts the object that has not been accessed for the longest time. Objects are added to the head of the list and removed from the tail of the list in constant time. Updates can also be accomplished in constant time by moving objects to the head of the list when they are referenced.

4.4 Model For Efficient Web Caching

LRU - Least Recently Used Strategy orders by last request time in which, with each reference to an object, the cache is updated in constant time by moving objects to the head of the list when they are referenced and removed from the tail of the list.

First of all, a set of web access sequential patterns is extracted from the past log data obtained from web log files at the proxy server. Now considering the new request. If resource A is requested and A,B is in the set of previously extracted web access sequential patterns then it is predicted that the resource B will be requested soon. Thus if B is already in cache, then its eviction should be delayed which is accomplished by assigning to B the priority it would have if it was requested immediately after A.

4.4.1 Reference Model

1. Web access sequential patterns are generated, using history log file data of fixed time interval, as input.
2. Cache is implemented using LRU- Least Recently Used Strategy.
3. For a requested entity, if the cache contains the entity then the entity is returned to the client this case is considered as a Hit.
4. If requested entity is not in the cache it is considered as a Miss and the entity is retrieved from the server and pushed into the cache and the request is fulfilled.
5. If the inclusion of the requested entity in the cache, makes the cache exceed its maximum size then entities are evicted from the cache according to the LRU- Least Recently Used Strategy.
6. Now considering the Web Access Sequential Patterns generated, next request is predicted, depending upon the present request from the Web Access Sequential Patterns and cache is updated according to the predicted request using the LRU- Least Recently Used Strategy.

Updating cache using the above mentioned strategy is being termed as **Extended LRU- Least Recently Used Strategy**.

5 CACHE PERFORMANCE ANALYSIS

The effect of measures of efficiency are analysed. The three efficiency measures which are used for analysis of cache performance are Hit Rate, Response time and Cache size. The cache performance is measured and compared by generating graphs.

5.1 Analysis Of Effect On Efficiency (Relative Hit Rate Of The Normal And Extended Lru- Least Recently Used Strategy)

The Figure 1 shows the object Hit Rate for both, Normal LRU- Least Recently Used Strategy and Extended LRU- Least Recently Used Strategy against the Total number of Requests. As predicted by the reference model the Extended LRU- Least

Recently Used Strategy showed significant increase in cache performance (Hit Rate) over , Normal LRU- Least Recently Used Strategy.

5.2 Analysis Of Effect On Response Time

The graphical representation in Figure 2 exhibits the variation in Response Time of Normal LRU- Least Recently Used Strategy and Extended LRU- Least Recently Used Strategy, against the Total number of Requests. The curves show that there is significant decrease in Response Time in case of Extended LRU- Least Recently Used Strategy.

5.3 Analysis Of Effect On Efficiency (Relative Hit Rate Of The Normal And Extended Lru (Least Recently Used Strategy)) depending on Cache size

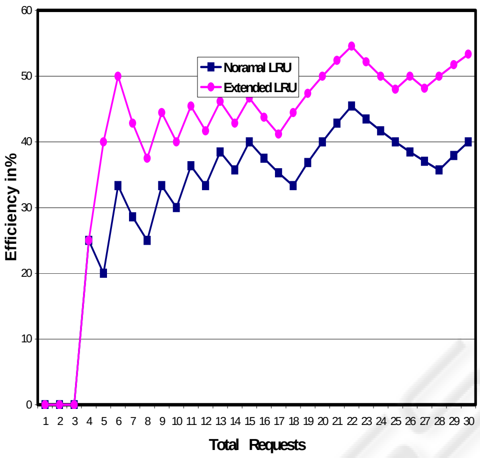
The Figure 3 & 4 presents the effect of Cache size on Efficiency. The graph compares the effect on efficiency with various cache sizes against the Total number of Requests for both Normal LRU- Least Recently Used Strategy and Extended LRU- Least Recently Used Strategy

6 CONCLUSIONS

The performance figures of the developed method, compared with LRU strategy indicate substantial increase in the hit rate and response time. The cache replacement strategy choice can have a marked effect on the object Hit Rate and Response Time of a proxy cache.

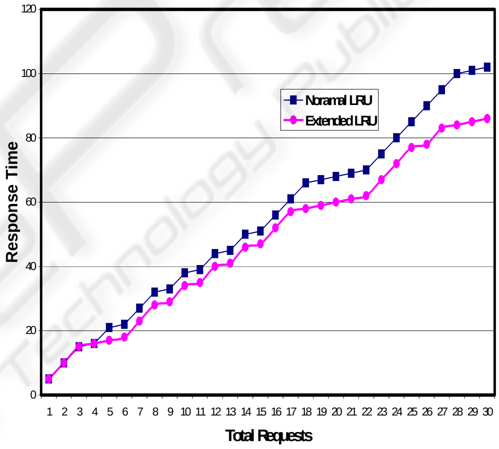
All in all, efficiency of the Web caching strategies lies in the capability of tuning the cache replacement strategy according to the recent history of request extracted from the Web Log data. Any strategy that may be efficient may suffer due to changes in Web Usage Patterns.

However there is definitely room for improvement, and the result of the implementation is encouraging. First and foremost direction for future work is to implement the proposed model on live proxy server, as well as a combination of different web usage mining techniques may be used for mining the Web Log file data.



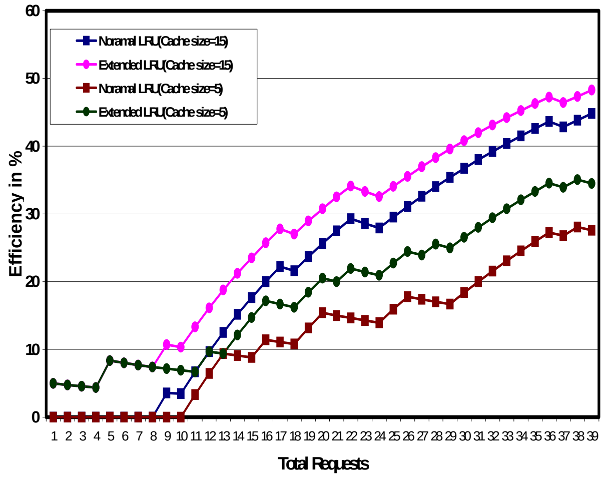
Relative Efficiency of the LRU & Extended LRU Strategy

Figure1



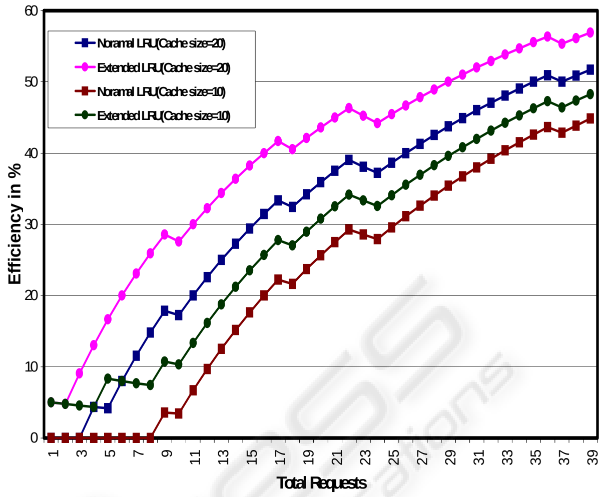
Relative Respone Time Analysis of the Noramal LRU & Extended LRU Starategy

Figure 2



Relative Efficiency of the LRU & Extended LRU strategy for different cache sizes

Figure 3



Relative Efficiency of the LRU & Extended LRU strategy for different cache sizes

Figure 4

REFERENCES

J. Han and M. Kamber.,2000 .Data Mining: Concepts andTechniques. Morgan Kaufmann, San Mateo, CA.

Rakesh Agrawal and Ramakrishnan Srikant 1994. Fast Algorithms for Mining Association Rules. In Proc. of the 20th Int'l Conference on Very Large Databases, Santiago, Chile, September 1994.

Robert Cooley, Bamshad Mobasher, Jaideep Srivastava 1999.Data preparation for mining World Wide Web browsing patterns.

Pitkow and M. Recker, 1994. “A Simple Yet Robust Caching Algorithm Based on Dynamic Access Patterns,” GVU Technical Report No. VU-GIT-94-39; also Proc. Second Int’l World Wide Conf., Chicago.

Jan Kerkhofs,Prof Dr. Koen Vanhoof,Danny Pannemans,2001.Web Usage Mining on Proxy Servers:A Case Study, Limburg University Center .

Brian D. Davison 2002.The Design and Evaluation of Web Prefetching and Caching Techniques.

Duane Wessels 1995. Intelligent Caching For World Wide Web Objects.

Robert Cooley, Bamshad Mobasher, Jaideep Srivastava 1997. Grouping Web page references into transactions for mining World Wide Web browsing patterns.

Andrei Z . Broder, Marc Najork, Janet L. Wiener. Efficient URL Caching for World Wide Web Crawling.

Karuna P Joshi , Anupam Joshi, Yelena Yesha , Raghu Krishnapuram 1999.Warehousing and Mining Web Logs.