



An Analysis of Web Caching Strategies to Improve Web Performance

G.N.K.SURESH BABU¹ and Dr.S.K.SRIVATSA²

¹Associate Professor, Department of Computer Applications, GKM College of Engineering and Technology, Chennai, INDIA.

²Sr.Professor – St.Joseph's College of Engineering, Chennai, INDIA.

Abstract : The World Wide Web (WWW) has grown exponentially in the past few years. Consequently, there is an ever-increasing demand for network bandwidth. One way to optimize network bandwidth usage is to implement a caching system which stores previously fetched files at a location close to the Web user, thereby allowing the user to experience reduced response time, and also potentially allowing network bandwidth to be conserved. This research work has proposed an enhanced server side Web cache algorithm which has improved the hit ratio and has reduced the cache miss. Web caching and prefetching are the most popular techniques that play a key role in improving the Web performance by keeping Web objects that are likely to be visited in the near future closer to the client. Web caching can work independently or integrated with the Web prefetching. Web caching systems can lead to significant bandwidth savings, higher content availability, reduction of client latency and increase in server scalability and availability.

Keywords: Web Cache, Web Prefetching, Bandwidth, Latency, Server Load

I. Introduction

A Web cache is a mechanism for the temporary storage of Web documents, such as HTML pages and images, to reduce bandwidth usage, server load, and perceived lag. A Web cache stores copies of documents passing through it; subsequent requests may be satisfied from the cache if certain conditions are met. Web caches have an important role in the reduction of server loads, client request latencies, and network traffic. Web caching is a well-known technique for reducing access latencies and bandwidth consumption. As in caching systems, a replacement policy is necessary for determining when and what to evict from the cache and many proxy caching algorithms have been proposed and evaluated. This chapter analyzes the distribution of current Web contents and re-evaluates various proxy cache replacement algorithms including LFU, LRU and MRU. Web caching is different from traditional caching in different ways. An important difference is that the size of Web objects is not uniform, and the transfer time costs are not identical. Due to the increased Web usage and its development, a huge amount of network traffic prevails in present Web services. It is very much essential to reduce this traffic for utilizing the Internet facilities in an efficient manner. Web caching improves the overall performance of Web and its related services.

II. Cache consistency mechanism

In general, cache coherence (also cache coherency) refers to the consistency of data stored in local caches of a shared resource. When clients in a system maintain caches of a common memory resource, problems may arise with inconsistent data. This is particularly true of CPUs in a multiprocessing system. Referring to the "Multiple Caches of Shared Resource" shown in Figure 4.1, if the top client has a copy of a memory block from a previous read and the bottom client changes that memory block, the top client could be left with an invalid cache of memory without any notification of the change. Cache coherence is intended to manage such conflicts and maintain consistency between cache and memory.

Figure 2.1 shows the representation of multiple caches of shared memory. A coherency protocol is a protocol which maintains the consistency between all the caches in a system of distributed shared memory. The protocol maintains memory coherence according to a specific consistency model. Older multiprocessors support the sequential consistency model, while modern shared memory systems typically support the release consistency or weak consistency models.

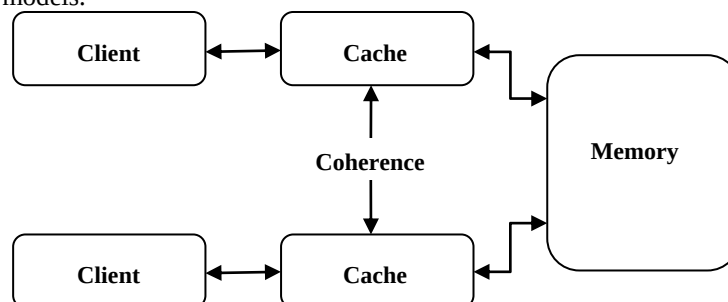


Figure 2.1 Multiple caches of shared memory

III. Design of Server Side Proxy Caching

A Server-Side Proxy Caching (SSPC) is one that acts as an intermediary for requests from clients seeking resources from other servers. A client connects to the proxy server, requesting some service, such as a file, connection, Web page, or other resource available from a different server. The proxy server evaluates the request according to its filtering rules. For example, it may filter traffic by IP address or protocol.

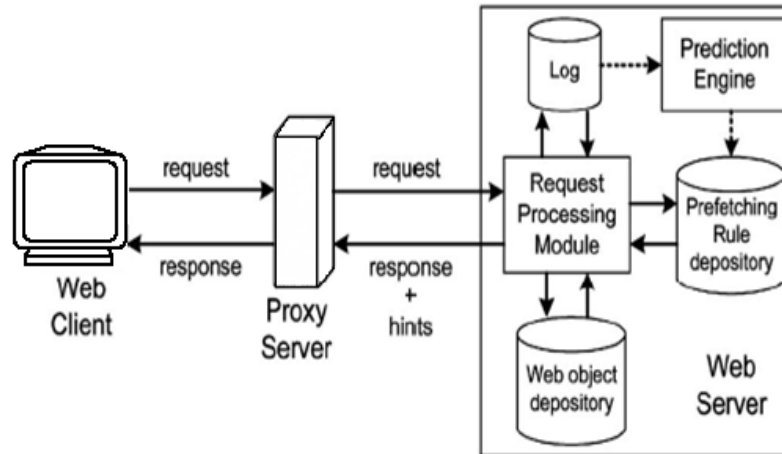


Figure 3.2 Architecture of server side proxy caching

If the request is validated by the filter, the proxy provides the resource by connecting to the relevant server and requesting the service on behalf of the client. A proxy server may on its own alter the client's request or the server's response, and sometimes it may serve the request without contacting the specified server. In this case, it 'caches' responses from the remote server, and returns subsequent requests for the same content directly.

Figure 3.2 shows a server side proxy caching which operates with many features such as reduced network traffic, reduced latency time, reduced load on Web server and saving of processing power. This architecture also inherently helps faster browsing of Web pages with the use of least number of grade page replacement algorithms. This server is successfully implemented with a few numbers of clients but it could be implemented for more of them. As mentioned before, it is more reliable, more advantageous than the existing one which uses the old data structures concept. It can work in a larger network and also maintains load balancing. This Server Side Proxy Caching system is executable under any platform and with any number of clients.

The proposed proxy server has been properly designed. The proxy server has provided rich information and Web server may deliberately send all possible prefetching hints with various levels of confidences to the proxy. Without any control, a proxy will prefetch every implied object into its cache, and despite is the confidences of some prefetching rules may be low.

IV. Server side Web cache algorithm

The keys of proposed Web caching have three aspects:

- Algorithm of routing requests
- Algorithm of replacing documents
- Algorithm of updating documents

Table 4.1 Server Side Web cache algorithm

1	WHILE there is a page p Cache in the current window w
2	Serve first such p and mark the page
3	Shift w if possible
4	IF all pages in the cache are marked
5	Unmark all the pages
6	Evict randomly an unmarked page in the cache
7	Fetch and mark the first page in the current window and Serve
8	Shift the current window w

This algorithm, focused on the second aspect, is algorithm of replacing documents. With the study of the characteristics of Web caching going further, algorithms of replacing documents based on the statistics of collected Web data are proposed. Each of this considers one or more of the following factors is its scheme:

- Document reference frequency
- Document reference recency
- Document size
- Strong or loose consistence of documents
- Replica documents in different proxies
- Non-replica documents in different proxies.

Efficient schemes combine more than one of factors in their implementation of Web cache. Some algorithms consider different cache architecture to improve caching performance, such as a fully-connected network of N cache. The Web cache algorithm's hit ratio for different values of the fetch unit is shown in Table 4.2. It is clear that the hit ratio of the proposed algorithm has improved gradually for different number of units and fetches.

Table 4.2 Hit ratios for different values of the fetch unit

Fetch Unit	Number of fetches	Hit ratio
1	4496503	0.372
2	3473175	0.515
3	3099268	0.567
4	2964553	0.586
5	2861390	0.600
10	2723683	0.620
20	2658159	0.629
40	2527111	0.640

V. SIMULATION RESULTS

The proposed system has performed all experiments on a PC with Pentium IV and 1 GB RAM, under the Windows XP using DOTNET. Comparison of hit ratio for client browser caching, client side proxy caching and server side proxy caching is shown in Table 4.3. It is clear that the hit ratio of the proposed server side proxy caching has improved compared to the existing methods.

Table 5.1 Comparison of Cache Misses for CBC, CSPC and SSPC

Number of fetches	Number of cache misses for CBC	Number of cache misses for CSPC	Number of cache misses for SSPC	% of Cache misses reduced
4496503	5198	5084	4134	18.3
3473175	4644	4351	3876	20.5
3099268	4148	3922	3529	21.7
2964553	4014	3776	3130	23.2
2861390	3891	3613	2868	24.5
2723683	3707	3482	2689	26.3
2658159	3563	3314	2435	27.7
2527111	3459	3198	2217	29.1

Table 5.1 shows the comparative analysis of cache misses for CBC, CSPC and SSPC with different number of fetches. For the proposed SSPC, cache misses has reduced gradually for different number of fetches made.

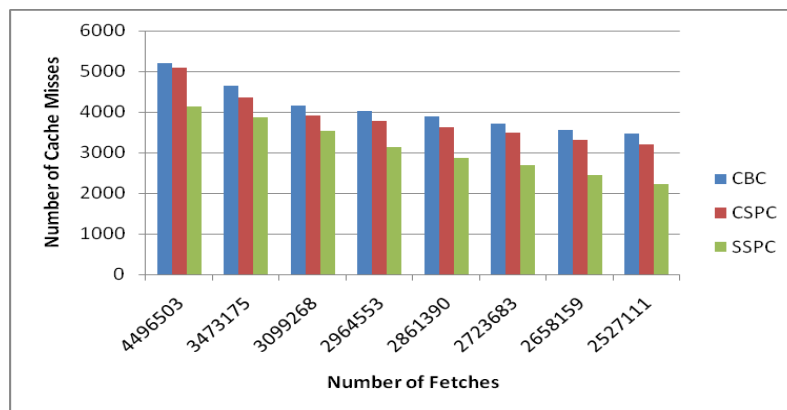


Figure 5.1 Performance comparisons of CBC, CSPC and SSPC with number of fetches

Figure 5.1 shows the performance comparisons of CBC, CSPC and SSPC with different number of fetches made. It is clear that the number of Cache Misses for the proposed SSPC algorithm has been reduced by 18.3%, 20.5, 21.7, 23.2, 24.5, 26.3, 27.7 and 29.1 respectively for different number of fetches made. As it is shown in Table 4.3, the number of cache misses for SSPC method is smallest compared with the existing methods.

The SSPC method is different from the previous strategies, a nondeterministic approach. This strategy uses randomized decisions to remove and replace an object from the cache. It does not need special data structure for inserting or deleting object and is simple to implement. It's a great advantage is that it can be evaluated and different simulation runs will give different results.

VI. PERFORMANCE OF WEB CACHING

There is a desire to improve upon Web caching systems as they can affect the performance of the Web today. This includes:

- **Reducing the cost of connecting to the Internet:** Traffic on the Web consumes more bandwidth than any other Internet service, and so, any method that can reduce the bandwidth requirements is particularly welcome, especially in the parts of the world in which telecommunication services are much more expensive than in the U.S., and in which service is often provided on a cost per bit basis. Even when telecommunication costs are reasonable, a large percentage of traffic is destined to or from the U.S. and so must use expensive trans-oceanic network links.
- **Reducing the latency of today's WWW:** One of the most common end-user desires is for increased speed. Web caching can help reduce the World Wide Wait, and so research (such as in Web prefetching) that improves upon user-perceived response times is quite welcome. Latency improvements are most noticeable, again, in areas of the world in which data must travel long distances, accumulating significant latency as a result of speed-of-light constraints or by accumulating processing time by each system over many network hops, and likewise increasing the chance of experiencing congestion as more networks are traversed to cover such distances. High latency as a result of speed-of-light constraints is particularly taxing in satellite communications.

VII. CONCLUSIONS

This paper has analyzed various types of Web caching methods and implemented Server Side Proxy Caching (SSPC) Web cache algorithm in order to reduce the cache misses and increase the hit ratio. The various design issues of Web caching algorithms such as load balancing, cache miss, transparency, scalability and cache coherence are analyzed. The performance of the proposed algorithm is compared with the existing methods, namely, Client Browser Caching and Client Side Proxy Caching in terms of cache misses and the hit ratio.

References

- [1]. Abdullah Balamash and Marwan Krunz (2008), "A Client side WWW prefetching model", Proceedings of IEEE Global Telecommunications Conference, pp.287-298.
- [2]. Abhari A., Dandamudi S. P., and Majumdar S. (2006), "Web Object-Based Storage Management in Proxy Caches", Future Generation Computer Systems Journal, Vol.22, No.1, pp. 16-33.
- [3]. Abrams M., Standridge C. R., Abdulla G., S. Williams, and E. A. Fox, "Caching Proxies: Limitations and Potentials", Proceedings of the Fourth International World Wide Web Conference, pp. 119-133.
- [4]. Bahat O. and Makowski A. (2009), "Optimal replacement policies for non-uniform cache objects with optimal eviction", Proceedings of the IEEE INFOCOM Conference on Computer Communications, pp.20-43.
- [5]. Bahn H., Noh S., Min S. L. and K. Koh (2007), "Using Full Reference History for Efficient Document Replacement in Web Caches", Proceedings of the 2nd USENIX Symposium on Internet Technologies and Systems, pp.407-434.
- [6]. Cao P. and S. Irani (2007), "Cost-Aware WWW Proxy Caching Algorithms", Proceedings of the USENIX Symposium on Internet Technology and Systems, pp. 193-206.
- [7]. Cao P., Zhang J., and K. Beach (2008), "Active cache: caching dynamic contents on the Web," Proceedings of IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing (Middleware), pp.373-388.
- [8]. Davison, B. D. (2008), "A Web caching primer", IEEE Internet Computing, Vol.5, Issue 4, pp.38-45.
- [9]. De La Ossa, A. Pont, J. Sahuquillo and J. A. Gil (2010), "Referrer Graph: a low-cost Web prediction algorithm", Proceedings of the 2010 ACM Symposium on Applied Computing, pp.209-223.
- [10]. Jin S. and Bestavros A. (2007), "Popularity-Aware Greedy dual-Size Web Caching Algorithms", Proceedings of 20th International Conference on Distributed Computing Systems, pp.254-261.
- [11]. Jin B, Sihua T, Lin C, Ren X., and Huang Yu. (2007), "An Integrated Prefetching and Caching Scheme for Mobile Web Caching System". Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, pp.289-305.
- [12]. Kahol, A. Khurana, S. Gupta, S.K.S. and Srimani, P.K. (2008), "A Strategy to Manage Cache Consistency in a Disconnected Distributed Environment", IEEE Transactions on Parallel and Distributed Systems. Vol. 12. No.7. pp. 686-700.
- [13]. Keqiu Li, Wenyu Qu, Hong Shen and Takashi Nanya (2005), "Two Cache Replacement Algorithms Based on Association Rules and Markov Models", Proceedings of the First International Conference on Semantics, Knowledge, and Grid, pp.876-892.
- [14]. Liu M., Wang F., and D. Zeng (2008), "Web caching: a way to improve Web qos", Journal of Computing Science and Technology, Vol. 19, No. 2, pp. 113-127.
- [15]. Markatos E. and Chironaki C. (2006), "A Top 10 Approach for Prefetching the Web", Proceedings of INET Conference, Geneva, Switzerland, pp.237-249.