ADJUSTING WEB CACHING COMPUTERS TO REDUCE COMMUNICATION CHANNEL ALLOCATION

R. Zuck, A. Williams, B. Kair, H. Bui, C. Stringfellow, N. Passos
Department of Computer Science
Midwestern State University
Wichita Falls, TX 76310

Abstract

On the Internet, it is common practice to use a computer to provide web caching services. Such services aim to reduce network utilization and improve access time to a web page. Limited storage space and performance requirements, however, restrict the number of objects a web cache can store. The need to access new objects requires replacement of cached ones. Such replacement decisions may affect users response time and communication channel allocation time. While many research studies focus on the availability of the objects, this paper addresses the problem of reducing the load of communication channels. This study discusses mechanisms used in Web caching and suggests an algorithm based on a time distance metric that should be considered with other cache management algorithms to provide the necessary relief to the communication system. Simulated experiments support the proposed algorithm, showing significant reductions of the average access time to web objects.

1 INTRODUCTION

Advances in computer networking technology have increased the use of the Internet and web based systems. This new communication environment has intensified the use of main communication trunks and has become a major topic of concern with respect to the capacity of currently established network systems. In order to improve network traffic and users response time, computers are installed in network routing nodes with the purpose of improving web access. One of the mechanisms used in this effort involves web caching, in which computer systems can temporarily store information traveling through the network. Most web caching systems focus on reducing the number of bytes being transmitted or improving the user's response time without addressing the network utilization time. This paper focuses on the use of web caching computers to reduce communication time between such machines and Web servers.

Web caching temporarily stores web pages or files, recently accessed by users, presuming that they will be accessed again in the near future. This is a concept similar to the temporal locality principle that resulted in the use of cache memories in computer systems. Due to the limited resources of the web caching system and its need to provide a fast response to the user, stored objects must be removed in order to provide space for the most recently accessed documents. The decision regarding which document to remove is made by cache replacement algorithms, usually centered on the number of bytes being transmitted for a specific document (object size) or by the number of times a document has been accessed, independent of its size [13].

Some cache replacement algorithms are based on traditional and well-known techniques such as Least Recently Used (LRU) and Least Frequently Used (LFU) [5]. While these techniques are applicable to reduce the user's response time and the server load, they do not address the problem of reducing the communication load along the network trunks. This paper proposes a new replacement algorithm that improves the functionality of web caching computers by reducing the time that communication channels are allocated and remain active while a web object is accessed from its server.

Several studies have been conducted in web cache replacement mechanisms [4, 5, 9, 13, 14]. Most of those studies derive from the basic concepts supporting the LRU and LFU techniques. Those techniques are usually modified in order to consider some additional parameter, such as the size of the web page [13]. While such a solution improves what is usually known as the byte hit rate, there is no guarantee that communication channels will be less active. Size, frequency of reference, turnover rate, and how recent the reference was made are also among the common characteristics considered in replacement decisions made by the cache [1, 2, 3, 6, 7]. Taking size, frequency, and recent use into account to calculate the utility of keeping a document in the cache became the basis for the Lowest Relative Value (LRV) replacement technique [11]. This technique uses various factors including the cost/benefit model for Web documents to determine the probability that a document will be accessed [11]. A similar study was conducted by Wooster and Abrams [14]. In their work, the emphasis is on improving the user's response time while reducing network utilization. Two algorithms, Latency (LAT) and Hybrid (HYB), were proposed, based on the estimated download time required from specific servers. However, those algorithms do not consider the possibility of a recently underutilized document being kept in the cache

for extended periods of time. In this paper, a combination of the LRU algorithm and communication load parameters is used to improve the network load and eliminate underutilized documents from the cache.

Network access time is also dependent on the physical characteristics of the network. Bandwidth, speed of the server and user equipment, and propagation time all affect the response time perceived by a user. Assuming that the propagation delay cannot be improved, Padmanabhan and Mogul developed a prediction and prefetching technique that helps reduce this type of latency [10].

Research conducted by Kalbfleisch, et al., focused on mobile devices and the latency perceived by the user, addressing the use of a hybrid hierarchical-distributed caching structure [8]. While most research focuses on the performance of the cache, others propose that the aggregate value received by the user is a more accurate performance metric than hit rate or byte hit rate in measuring the effectiveness of a replacement algorithm [12]. In this case, priority could be provided for those users that are willing to pay more for their network use.

Other researchers focus on solutions that attribute different priority values to different servers [4, 9]. Such values are usually translated into numbers charged to the users. Servers with higher values will have priority over lower value ones. The result of such mechanisms is to improve a new metric, known as value hit rate. However, the byte hit rate may be significantly decreased, and the transmission time increased, which could result from the lack of information on the communication channels utilization.

This paper introduces a new web caching replacement algorithm, which takes into consideration the time distance between the web caching computer system and the servers. Section 2 provides the basic background on the computer application to the web caching function, followed by Section 3 describing the proposed algorithm. Section 4 shows results of a simulated environment where the new technique was applied and observations on the improvements of the communication channels utilization. A summary follows, wrapping up the concepts presented.

2 BACKGROUND

The concept of web caching is very similar to the traditional definition of cache memory. In a computer system, caches are placed close to the central processing unit so that data can be available to the processor without accessing main memory. In a network system, in order to speed up access to a web document, an intermediary computer system stores a copy of such objects close to its final destination. Web caches are therefore placed close to the user's computer. Figure 1 shows a representation of possible locations for a web caching computer, according to a hierarchical structure. In such a structure, the web browser maintains a copy of the most recently visited web pages in the local (user) system. When a new web page request is initiated, the browser attempts to fulfill the request from the documents located within its cache. When the requested page is not found in the local cache, a request is sent to the Web server. Since a web caching computer can also be placed at the point where an organization's computer network connects to the Internet, that cache will try to answer the request before sending it to the target server. This process propagates along the network until all caches have registered a miss and the initial request has reached the server. This process reduces the user's response time and the data traffic on the external network.

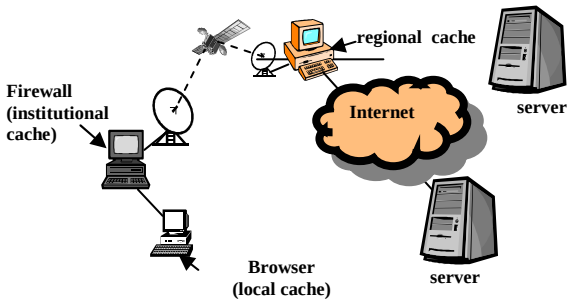

Figure 1. Possible placement of Web caches.

Depending on the protocol being utilized by the network and the security policies adopted, a request to the server may result in the allocation of a communication channel for the transmission of the requested document. Such a situation is usually necessary on wireless communication and protected virtual private networks. If one assumes that this allocation lasts the necessary time interval to transmit the entire object, a reduction in the average access time of those objects is expected to improve the availability of the network system.

3 SERVER TIME DISTANCE AND LRU

One of the most common cache replacement mechanisms is the LRU algorithm. It is a traditional technique in replacing lines in a processor cache and pages in a virtual memory system. This technique is based on the concept that information recently used (or accessed) will be used again in the near future. However, there are a number of other methods that can be applied to the caching problem with similar efficiency. In this paper, the LRU technique is adopted in order to conduct the replacement decision process when the information being replaced is so recent that any other assumption cannot be considered reliable.

As mentioned earlier, in the case of allocation of communication channels in network trunks, the transmission time becomes the essential metric used to evaluate the trunk's utilization. A significant reduction in the average transmission time is expected to reduce the channel allocation and improve the availability of network resources. By assuming that all contacts with the Internet will be initiated by user systems connected to some network routing node, which also works as a web caching system, one may also assume that the amount of time for information to be exchanged between the web caching system and the target servers will basically depend on the time distance between those two machines.

The time distance between two systems, connected in a computer network, is dependent on several factors, which may include the geographic distance between those machines, the number of intermediary machines to be traversed, the communication speed of the channels utilized in the connection. etc. In this paper, the time distance is associated with the two connected machines and the document being accessed, adding the amount of data or the size of the document to the behavior of the network to produce an index, which will be used in the replacement mechanism of the cache system. The time distance index function is defined below.

Definition: the time distance index function of a web object, *tdif (object)*, is the time in seconds measured between the request and the arrival of the object in the caching system, when that object was last accessed from its server.

Based on this definition, the web caching computer administrator may establish a time threshold that will allow the system to qualify web objects as residing in servers NEAR or FAR from the cache. Notice that the concept of distance is associated with access time and not geographic location.

Objects residing in near servers are expected to consume less communication time than objects located in far servers. Based on this concept, one may assume that it is beneficial to the caching system to replace near pages instead of far pages when the replacement mechanism is activated. There is, however, a contentious point. If there are only a few near pages in the cache system and they were all recently accessed, then by removing them, the system would allow the eventual permanence of old, maybe already obsolete, far pages in the cache. In order to avoid such an anomaly, a reasonable replacement algorithm needs to consider the age of the cached pages in its decision process. In this paper, the cached pages are split into two other groups based on how long they have been in the cache. The group boundary is defined as a percentage of the time the oldest page has been in the cache.

The newly proposed algorithm will select replacement pages from those older than the boundary established above, selecting the near pages first. If no near pages are old enough, then far pages considered old enough, will be removed. If none of the pages fall in the old category (which may not seem possible, but due to the percentage ratio adopted must be considered feasible), then a simple LRU technique will be triggered to make the decision.

In a more formal context, if T_i is the access time required to access the object O_i, where t_i is its date/time of access, and the system administrator establishes a percentage *p* for identification of old and new pages and a threshold *h* for the definition of near and far tuples server/object, then the replacement algorithm will be based on the following criteria:

1. If $T_i < h$ and $t_i \leq t_{boundary}$, then O_i is removed.
2. If $T_k \geq h$ and $t_k \leq t_{boundary}$ and no other object satisfies condition 1, then O_k is removed.
3. If conditions 1 and 2 are not satisfied by any object, then an LRU algorithm is applied, i.e., given $t_j \leq t_i$ for all i such that O_i is cached, then O_j is removed.

where $t_{boundary} = t_{old} + (t_{new} - t_{old})*p$, $t_{old} = \min \{t_i$ such that O_i is cached$\}$ and $t_{new} = t_k$, such that O_k is being cached and requires the replacement technique to be activated.

The application of this algorithm was simulated using a trace log of web accesses. The results are presented in the next section.

4 SIMULATIONS

Experimental simulations were conducted utilizing a two-day web trace file, logging 36,712 successful accesses. These results were replies to requests issued by 2,533 users browsing through 6,657 different web pages. During the simulation, one of three communication channel transmission rates (1, 2, and 4 Kbytes/s) was randomly assigned to a server and used to establish the access time of each object. Figures 2 and 3 show the graphical visualization of the simulation results. In figure 2, the total miss time represents the access time required to retrieve an object not stored in the cache from its originating server. As one can observe, the use of the new TDIF criteria results in a reduction of the access time for caches with size up to 2000 entries.

These results are confirmed by figure 3, where the total access time is graphed as a function of the cache size. In this graph, one can observe that for larger cache sizes there is no significant distinction between the performances of the LRU method and the TDIF mechanism. The explanation is based on the test data used, which represents a limited number of web pages.

This finite number of pages is stored in the larger caches in such a way to eliminate the need of any external access.

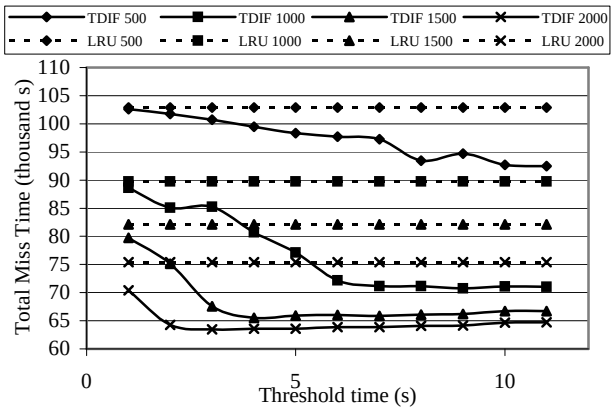

Figure 2. TDIF lines are based on the optimal total miss time for the given threshold time.

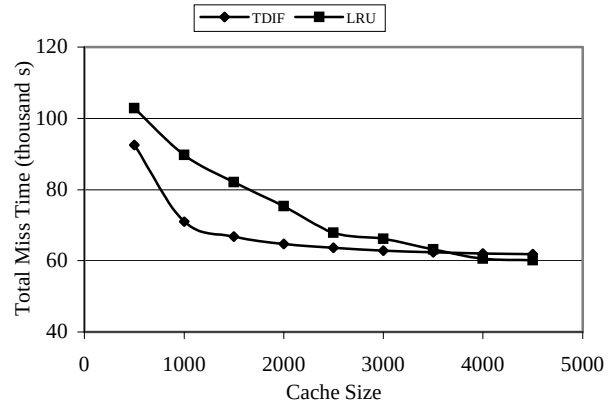

Figure 3. Comparison TDIF and LRU techniques.

5 SUMMARY

The popularization of the web and the ever increasing number of users and web servers have the potential of adding to network congestion, which tend to significantly increase the allocation time of communication channels. One of the most effective ways to reduce network utilization is the use of web caching. While most of the research in this field focuses on users response time and network load, few studies have addressed the need to control allocation time of communication channels. The process of ensuring fast access to Web documents while reducing such allocation requirements demand modifications in web caching mechanisms. This paper introduced a new cache management system that takes into consideration the time necessary to access web objects and represents this time as a time distance index. Such an index allows servers to be classified as far or near caching systems, and associates the cache replacement decisions with the time required to access the objects. Simulation results showed that the utilization of the communication channels, measured by the average access time to web objects, significantly improved over results obtained from the more traditional least-recently used systems.

6 ACKNOWLEDGEMENTS

This work was partially supported by the Texas Advanced Research Program under Grant No. 003656-0108b-2001.

7 REFERENCES

[1] M. F. Arlitt, R. Friedrich, and T. Jin, “Performance Evaluation of Web Proxy Cache Replacement Policies”, Proceedings of the Conference on Computer Performance Evaluation, Modeling Techniques and Tools, Sep. 1998, pp. 193-206.

[2] M. Arlitt, L. Cherkasova, J. Dilley, R. Friedrich and T. Jin, “Evaluating Content Management Techniques for Web Proxy Caches,” ACM SIGMETRICS Performance Evaluation Review, Vol. 27, number 4, Mar. 2000, pp. 3-11.

[3] G. Barish and K. Obraczka, “World Wide Web Caching: Trends and Techniques,” IEEE Communications, May 2000, pp. 178-185.

[4] Y. M. Chan, J. P. Womer, S. Jamin and J. K. MacKie-Mason, “One Size Doesn't Fit All: Improving Network QoS Through Preference-driven Web Caching”, Second Berlin Internet Economics Workshop, May 28-29, 1999.

[5] P. Cao and S. Irani. “Cost-Aware WWW Proxy Caching Algorithms”, Proceedings of the Usenix Symposium on Internet Technologies and Systems, 1997, pp. 178-185.

[6] J. Dilley, M. F. Arlitt, “Improving Proxy Cache Performance: Analysis of Three Replacement Policies,” IEEE Internet Computing, Vol. 3, n. 6, 1999, pp. 44-50.

[7] A. Grilo, P. Estrela, and M Nunes, “Terminal Independent Mobility for IP (TIMIP),” IEEE Communications, Vol. 39, No. 12, Dec. 2001, pp. 34-41.

[8] G. Kalbfleisch, W. Deckert, R. Halverson, N. Passos, “A Brief Study on Web Caching Applied to Mobile Web Applications,” Proceeding of the 18th International Conf. on Computers and their Applications, Honolulu HI, Mar. 2003, pp. 442-445.

[9] T. P. Kelly, Y. M.Chan, S. Jamin, and J. K. MacKie-Mason, “Biased Replacement Policies for Web Caches: Differential Quality-of-Service and Aggregate User Value”, Proceedings of the 4th International Web caching Workshop, Mar. 1999.

[10] V. N. Padmanabhan, J. C. Mogul, “Using Predictive Prefetching to improve World Wide Web Latency”, ACM Computer Communication Review, Vol. 26, n. 3, pp. 22-36, Jul. 1996.

[11] L. Rizzo and L Vicisano, “Replacement Policies for a Proxy Cache” IEEE/ACM Transactions on Networking, pp. 158-170, Apr. 2000.

[12] W. Stallings, Data and Computer Communications, 5th Ed. Prentice Hall, Upper Saddler River, NJ, 1997.

[13] S. Williams, M. Abrams, C.R. Standridge, G. Abdulla and E.A. Fox, “Removal Policies in Network Caches for World-Wide Web Documents”, Proceedings of the ACM SIGCOMM '96 Conference, Aug. 1997, pp. 293-305.

[14] R. Wooster and V. Abrams, “Proxy Caching that Estimates Page Load Delays”, WWW6, April 1997, pp. 325-334.