Performance Comparison of Alternative Web Caching Techniques*

Hossam Hassanein, Zhengang Liang and Patrick Martin
Department of Computing and Information Science
Queen's University
Kingston, Ontario, Canada, K7L 3N6
{hossam, martin}@cs.queensu.ca, leonl@ca.ibm.com

Abstract

Web caching is a popular technique to improve the performance and scalability of the Web by increasing document availability and enabling download sharing. Distributed cache cooperation, a mechanism for sharing documents between caches, can further improve performance by providing a shared cache to a large user population. Layer 5 switching-based transparent Web caching schemes intercept HTTP requests and redirect requests according to their contents. This technique not only makes the deployment and configuration of the caching system easier, but also improves its performance by redirecting non-cacheable HTTP requests to bypass cache servers. In this paper, we compare the performance of a number of cooperative (ICP and Cache Digest) and transparent (L5 transparent Web caching and LB-L5) Web caching techniques. We conduct a number of simulation experiments under different HTTP request intensities, network link delays and populations of cooperating cache servers. The relative merits of the different schemes are reported.

1. Introduction

The World-Wide Web [1,2] (the Web) is an Internet-based globally distributed information system that was originally developed at CERN (Conseil Européen pour la Recherche Nucleaire) for sharing information among collaborating researchers.

Web caching, which temporarily stores Web objects (such as Hypertext documents) for later retrieval, is an effective method of improving the performance and scalability of the Web. Web caching can be performed at Web proxies. A Web proxy consists of application level software that accepts HTTP requests from a set of clients, fetches the requested objects from original Web servers, caches the requested objects and sends these objects back to the clients. Proxy Web caching increases document availability and enables download sharing. It reduces overall access delay and saves network bandwidth by caching frequently requested Web objects.

In addition to local proxy Web caching, distributed cache cooperation, a mechanism for sharing documents between caches, can further improve system performance by providing a shared cache to a large user population [3,4]. In a cooperative Web caching system, if a cache miss occurs at a local cache server, the request can be forwarded to one of a set of cooperating servers. Therefore, if any one of the servers has a cached copy of the requested object then the request results in a cache hit.

In this paper, we compare a number of Web caching techniques. We consider the Internet Cache Protocol (ICP) [6], which is a query-based cooperative approach, Cache Digest [7], which is a directory-based cooperative approach, Layer 5 (L5) switching-based transparent Web caching [10] and LB-L5, which is a transparent Web caching scheme supporting distributed cache cooperation [12].

The remainder of the paper is organized as follows. Section 2 describes the Web caching techniques studied in the paper. Section 3 outlines the simulation model used in our study and presents the results of our experiments. Section 4 concludes the paper and provides our recommendations.

2. Web Caching Techniques

In this section we provide an overview of four Web caching techniques, namely ICP, Cache Digest, L5 switch and LB-L5.

* This research is supported by Communications and Information Technology Ontario (CITO) and the Kingston Software Factory.

2.1. Query-Based Approach - ICP

ICP [6] is the most popular protocol using the query-based technique to coordinate a set of cooperating proxy Web caches. ICP is an application layer protocol running on top of UDP (User Datagram Protocol). Both Harvest [5] and Squid [6] use ICP to coordinate proxy Web caches.

In ICP a client sends a request to its configured proxy cache server. If that cache server cannot find the requested object in its own cache, it broadcasts an ICP query message to all other cooperating cache servers. The configured cache server sends an HTTP request for the object to the first server that responds to the query with an ICP hit message. Upon receiving the object, the configured cache server stores a copy in its cache and then sends the object to the client. If no cooperating cache server responds to the query with an ICP hit message before a time-out period, then the configured cache server fetches the requested object from the original Web server.

2.2. Directory-Based Approach – Cache Digest

Although query-based approaches, such as ICP, work well when cooperating proxy cache servers are located close to each other, the query/response delay becomes significant in a wide area network. Directory-based approaches allow cache servers to make information about their cache content available to peers in order to avoid the query/response delay.

Using an uncompressed directory of cache content can result in huge memory consumption on the cache servers and high directory update traffic on the network. Compressed representations of the cache content directory have therefore been used in several approaches.

Summary Cache [8] and Cache Digest [7] are two similar approaches. Both use a Bloom Filter to represent the directory of cache content. The major difference between them is that Summary Cache extends ICP to update the directory, while Digest Cache uses HTTP to transfer directory information.

2.3. L5 Transparent Web Caching

Transparent Web caching uses network devices to redirect HTTP traffic to cache servers. The technique is called transparent because Web browsers do not have to be explicitly configured to point to a cache server, that is, the caches are transparent to the browsers [10].

A Layer 4 switching device can be used to redirect TCP/IP packets destined to HTTP ports to cache servers, and to forward all other network traffic directly to the WAN router. While L4 switches are optimized for the transport layer, they are completely unaware of the Application Layer protocols such as HTTP and FTP.

Layer 5 (L5) switches, like L4 switches, provide high speed switching of traffic. L5 switches use nformation in the TCP and HTTP request headers, for example the URL requested, to make routing decisions based on the actual content of the requested object and to manage request/response flows from beginning to end [11].

2.4. LB-L5 Web Caching

Load-Balanced-Layer-5 (LB-L5) Web caching [12] uses information about the proxy cache server workload, network link delay and cache content to redirect HTTP requests, which in turn balances cache server workload and reduces average response times. LB-L5 uses a Bloom Filter to represent cache content. LB-L5 extends ICP, to support communication between cache servers and switches. The information is obtained as follows:

- Cache content and access-frequency information is obtained and represented with a Bloom Filter by each cache server. It is sent to each LB-L5 switch using the extended ICP message **ICP_UPDATE_CONTENT**.
- An LB-L5 switch obtains the workload information from a cache server by sending each cache server an **ICP_QUERY_WORKLOAD** message. A caches server responds to the query with an **ICP_UPDATE_WORKLOAD** message, whose payload field carries the server's workload information.
- There are several approaches to measuring network latency. One approach uses tools such as ping, traceroute, and Zone transfer from a DNS server. Another approach uses network services such as SONAR, IDMAPS and ReMoS [14]. LB-LB uses the message round trip time between an LB-L5 switch and a cache server to measure the network latency. This message round trip time includes the propagation delay, packet transmission delay and network access delay.

Upon receiving an HTTP request, an LB-L5 switch makes a routing decision as follows:

1. If the request is non-cacheable, the switch redirects it to the original Web server.
2. For every cache server, the switch estimates the time needed to fetch the requested object from that server as follows:
   - 2.1 The switch computes hash values for the request's URL and uses the Bloom filters to predict if a server has the requested object in its cache.
   - 2.2 The switch estimates the time (T) needed to fetch the requested object from the server as:
     - 2.2.1 If the requested object is available at the cache server, then T includes the time for connecting to the server, sending the request from the switch to the cache server, searching for the object at the server, moving the object

from disk to memory, and sending the object from the server to the switch.

2.2.2 If the object is not available at the cache server, then the time for the cache server to fetch the object from the Web server is added.

3. The switch pre-selects a set of cache servers with a response time below a specified threshold.
4. The switch chooses the cache server from the pre-selected set that has the highest frequency for the requested objects, and redirects the request to the server.

3. Performance Evaluation

In this section, we present a performance comparison of the different Web caching schemes presented in Section 2, namely ICP, Cache Digest, basic L5 transparent Web caching and the LB-L5 scheme. Section 3.1 explains the simulation model adopted in this study, which includes the network model, proxy traces and the simulation software. The effects of network link delay, HTTP request intensity, and the number of cooperating proxy servers on the performance of the Web caching schemes are reported in Section 3.2.

3.1. Simulation Model

In this study, we simulate a distributed cache cooperation architecture. In the ICP and Cache Digest schemes each proxy cache server accepts HTTP requests from a cluster of clients, and has a link to every other cooperating proxy server. In the basic L5 and LB-L5 Web caching schemes a switch transparently intercepts HTTP requests from a cluster of clients. The switch redirects a cacheable request to a cache server. Non-cacheable requests are routed directly to the Web server. The difference between the basic L5 and LB-L5 is that LB-L5 supports distributed cache cooperation. In the LB-L5 scheme, a switch can make routing decisions and redirect a HTTP request to one of a set of cooperating cache servers.

We use publicly available proxy traces from the National Laboratory for Applied Network Research (NLANR) [13] cache servers to generate HTTP requests in the simulation. In our experiments, proxy trace files are used in a controlled way. This is achieved by modifying the traces to examine the effects of different parameters. We condense or expand the traces with different factors (shorten or enlarge the interval between requests proportionally) and use different network link delays and different numbers of cooperating cache servers to investigate their effects on the performance of Web caching schemes.

The parameters used in the simulation are chosen according to data measured by Rousskov [15,16] and described in detail by Liang [12]. We assume that the request processing time at a proxy cache server, which includes the time to search for a requested object in a cache and the disk access time for moving the object from disk to memory, is proportional to the number of concurrent requests.

The simulator used in this paper conducts discrete event driven simulation. It was developed using the Java programming language. The simulation software consists of the following six major components:

- Client Cluster: responsible for simulating a cluster of clients. It generates HTTP request traffic using the request logs from proxy trace files.
- Basic L5 and LB-L5 Switch: responsible for simulating the basic L5 switch and LB-L5 switch. The basic L5 switch redirects a cacheable request to its associated cache server and a non-cacheable request to a Web server. The LB-L5 switch supports extended ICP messages and communicates with cooperating cache servers. It redirects a cacheable request to one of a set of distributed cache servers based on the cache content and access-frequency information, server workload, and network link delays.
- ICP/CacheDigest/L5/LB-L5 Proxy Servers: responsible for simulating a proxy cache server. They use the Least Recently Used (LRU) replacement algorithm to maintain their caches. The basic L5 proxy server does not support cache cooperation. Therefore, it only performs the LRU cache management function. Other proxy cache servers perform additional functions. The ICP proxy server uses the ICP protocol to support query-based cache cooperation. The Cache Digest uses a Bloom Filter to represent cache content and supports directory-based cache cooperation. The LB-L5 proxy server uses a Bloom Filter to represent cache content. It uses extended ICP messages to communicate with LB-L5 switches to publish workload, cache content and access-frequency information.
- Web Server: responsible for simulating a Web server. It accepts HTTP requests and then sends back HTTP responses and requested objects.
- Network Link: responsible for simulating a network link connecting proxy servers, L5 switches, Web server and client clusters. It passes messages from one end to the other with a specified link delay.
- Event Manager: responsible for simulation event queuing and dispatching. The event manager handles all simulation events.

3.2. Simulation Results

The parameters used in the experiments are network link delay, HTTP request intensity, and the number of cooperating cache servers. These parameters are set as follows:

- Network link delay: the network link delays are varied from 5 to 200 milliseconds, which represent a wide range of link distance and/or network congestion levels.
- HTTP request intensity: the different HTTP request intensities are simulated by condensing or expanding the base proxy traces with a controlled factor (shortening or enlarging the interval between requests proportionally). In the experiments, the HTTP request intensities are set from 100% to 250%. The intensity range is chosen according to the raw-trace data where the peak request intensity (1043 requests per minute) is about 224% of the average request intensity (465 requests per minute) [12]
- Number of cache servers: we consider cases with 4 and 10 cooperating cache servers.

The primary performance measure used in this paper is the average response time of Web requests measured from the initiation of a Web request at a client to the delivery of the first bit of information back to the client. In the simulation experiments, results are sampled every minute in 30-minute durations after warm-up time. Simulations are run long enough to obtain a 90% confidence level with 10% confidence intervals.

Figure 1 shows the performance of the caching schemes under a variety of conditions. The graphs plot response time versus network delay for different request intensities and different numbers of cooperating proxy cache servers. Except for the basic L5 scheme, the response times of the Web caching schemes increase as the network link delay increases. The extent to which the times increase, however, is different in each scheme. The response time of the basic L5 scheme is not affected by link delay since it does not support cache server cooperation.

ICP is greatly affected by the network link delay since it is a query-based scheme. As described in Section 2, if an ICP proxy cache server cannot find a requested object in its own cache, it queries all other cooperating cache servers to find a cached copy of the object. As the network link delay increases, the inter-proxy query/response time increases. As shown in Figure 1 (c.2), the response time of ICP increases up to 370 milliseconds as the link delay increases from 5 to 200 milliseconds.

In Cache Digest, when a proxy cache server cannot find the requested object from its own cache, it searches the digests of all other cooperating servers, and if it finds the object then it fetches the object from other cache server. The Cache Digest scheme does not involve inter-proxy query/response. Fetching an object from a remote server, however, makes the HTTP request response time still susceptible to network link delay. As shown in Figure 1 (c.2), the response time of Cache Digest increases up to 84 milliseconds as the link delay increases from 5 to 200 milliseconds.

The basic L5 scheme is not affected by network link delays because it does not support distributed cache cooperation. On the other hand, the response time of LB-L5 is affected by link delay. When the link delay is small, LB-L5's routing decision is based mainly on the cache content and workload information of cache servers. HTTP requests can be redirected to any one of the cooperating cache servers to increase cache hit rate or to balance server workload. As the link delay increases, the response time improvement achieved by redirecting requests to remote cache servers decreases. Therefore, when the link delay is very large, LB-L5's performance closely resembles the basic L5 scheme. As shown in Figure 1 (c.2), the response time of LB-L5 increases up to 750 milliseconds as the link delay increases from 5 to 200 milliseconds. However, LB-L5 always outperforms the other schemes. We observe that, regardless of the network link delay, the HTTP request response time for all Web caching schemes increases as we increase the request intensity. Again, LB-L5 performs as least as well the other three schemes under all combinations of request intensity and link delay.

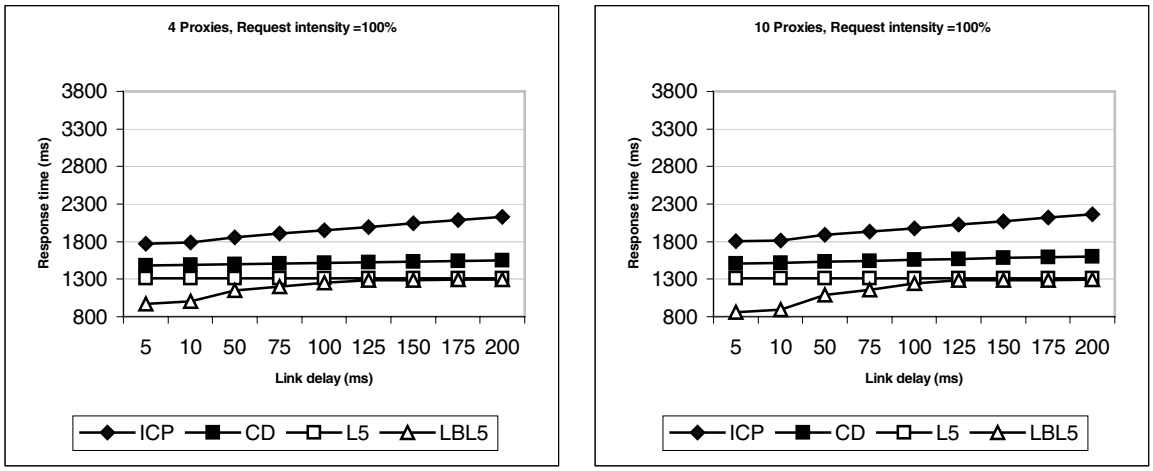
Simulation results show that LB-L5 adapts better to high request intensities than the other schemes. As request intensity increases, the increase in the LB-L5 response times is less than the increases experienced by the other schemes.

The basic L5 scheme does not support cache server cooperation. Therefore, the number of cache servers does not affect it. Increasing the number of cooperating cache servers does not guarantee performance improvements in ICP and Cache Digest. This is because ICP and Cache Digest try to achieve the best hit rate but do not consider cache server workload and network link delay. Simulation results show that the response times of ICP and Cache Digest increase as more requests are redirected to remote or busy cache servers to achieve higher hit rates.

On the other hand, in LB-L5, as the number of cooperating cache servers increases, switches are better able to redirect requests to balance server workloads. Thus the performance gain increases. When the link delay is small, the performance improvement is significant. However, under very large link delays, requests are not likely to be redirected to remote servers. LB-L5 cannot gain the benefits of server workload balancing and cache sharing in this case.

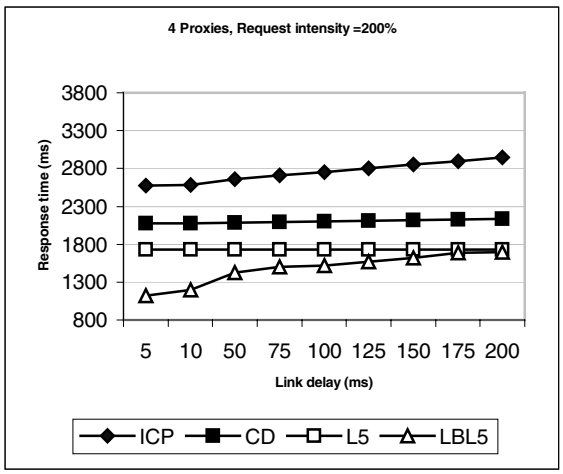
4. Summary

Web caching is considered one of the most effective approaches to improving the performance and scalability of the Web. Emerging Layer 5 switching-based transparent Web caching techniques not only makes the deployment and configuration of the caching system easier, but also improves its performance by redirecting non-cacheable HTTP requests to bypass cache servers.

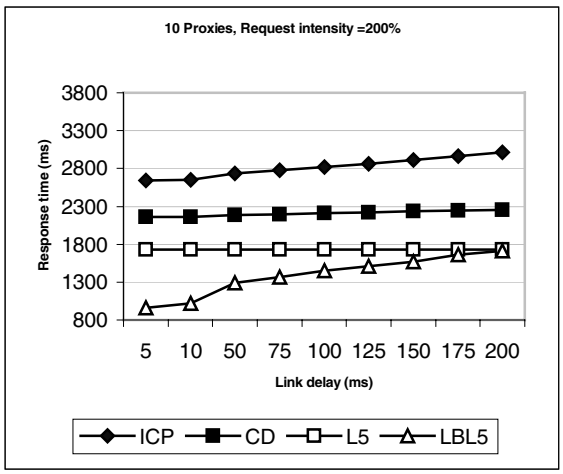


(a.1)

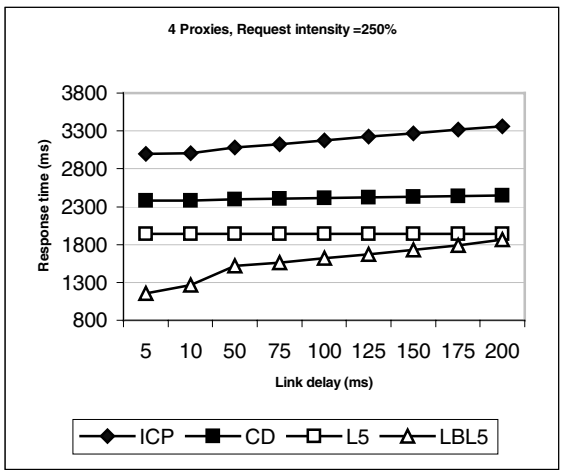
(a.2)



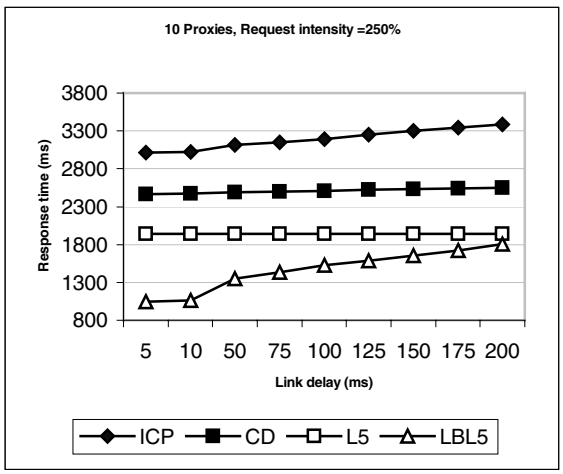
(b.1)



(b.2)



(c.1)



(c.2)

Figure 1. Performance of Web Caching Schemes

In this paper we have compared the performance of two cache cooperation techniques and two transparent Web caching techniques using an extensive simulation model. The adopted network model, simulation experiment settings, and simulation software implementation were described. Controlled-parameter simulation experiments were conducted to investigate the effects of network link delay, HTTP request intensity, and the number of cooperating proxy servers on the performance of the Web caching schemes.

The simulation experiments showed that LB-L5 outperforms ICP, Cache Digest, and the basic L5 scheme, with respect to HTTP request response time under various network link delays. Regardless of the HTTP request intensity, the response time of ICP, Cache Digest, and LB-L5, increases as the network link delay increases, whereas the basic L5 scheme is not affected by link delay. Under large link delays, the response time of LB-L5 is close to that of the basic L5 scheme, since LB-L5 avoids redirecting requests to remote servers when the cost is too high.

The experiments also show that LB-L5 adapts better to high request intensities than the other three schemes. Under high request intensity, LB-L5's server workload balancing produces significant performance improvement. Likewise, LB-L5 demonstrated a better capability of supporting cache cooperation than the other three schemes. As the number of cooperating cache servers increases, LB-L5 has more opportunities to redirect requests, which helps to balance server workloads and increase cache sharing. These two factors mean that the performance improves. We, therefore, conclude that combining cache cooperation and transparent Web caching techniques is a promising direction and should be investigated further.

5. References

[1] T. Berners-Lee, R. Cailliau, J. Groff, and B. Pollermann, "World-Wide Web: The Information Universe", *Electronic Networking: Research, Applications, and Policy*, Spring 1992.

[2] CERN – European Laboratory for Particle Physics, "An Overview of the World-Wide Web". Available at: http://www.cern.ch/Public/ACHIEVEMENTS/WEB/

[3] R. Tewari, M. Dahlin, H. Vin and J. Kay, "Beyond Hierarchies: Design Considerations for Distributed Caching on the Internet" Technical Report TR98-04, Department of Computer Sciences, University of Texas at Austin, February 1998.

[4] A. Wolman, G. Voelker, N. Sharma, N. Cardwell, A. Karlin, and H. Levy, "On the scale and performance of cooperative web proxy caching", *Proceedings of the 17th Symposium on Operating Systems Principles*, December 1999.

[5] C. Bowman, P. Danzig, D. Hardy, U. Manber, and M. Schwartz, "The Harvest information discovery and access system", *Proceedings of the Second International World Wide Web Conference*, October 1994.

[6] D. Wessels and K. Claffy, "ICP and the Squid Web Cache", *IEEE Journal on Selected Areas in Communication*, Vol 16, #3, pages 345-357, April 1998.

[7] A. Rousskov and D. Wessels, "Cache digests", *Proceedings of the Third International WWW Caching Workshop*, Manchester, England, June 1998.

[8] L. Fan, P. Cao, J. Almeida, and A. Broder, "Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol". *Proceedings of ACM SIGCOMM*, September 1998.

[9] E. Johnson, "Increasing the Performance of Transparent Caching with Content-aware Cache Bypass", *Proceeding of the 4th International WWW Caching Workshop*, 1999.

[10] B. Williams, "Transparent Web Caching Solutions", *Proceedings of the 3rd International WWW Caching Workshop*, 1998.

[11] ArrowPoint Communications, "Content Smart™ Cache Switching", White paper, available at http://www.arrowpoint.com/solutions/white_papers/printer/Content_Smart_Cache_Switching.pdf

[12] Z. Liang, H. Hassanein and Patrick Martin, "Transparent Distributed Web Caching", *Proceedings of the IEEE Conference on Local Computer Networks*, Nov. 2001, pp. 225-233.

[13] National Laboratory for Applied Network Research (NLANR), *Ircache project*, available at: http://ircache.nlanr.net

[14] R. Siamwalla, R. Sharma, and S. Keshav, "Discovering Internet Topology", July 1998, available at http://www.cs.cornell.edu/skeshav/

[15] A. Rousskov and V. Soloviev, "On performance of caching proxies", *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems* (SIGMETRICS '98/PERFORMANCE '98), pages 272-273, Madison, WI, June 1998.

[16] A. Rousskov and V. Soloviev, "A performance study of the squid proxy on http/1.0", *World Wide Web*, 2(1-2):47-67, January 1999.