# Web Caching

## A Technique to Speedup Access to Web Contents

*Harsha Srinath and Shiva Shankar Ramanna*

Harsha Srinath is currently working at the Department of Computer Science and Automation, Indian Institute of Science, Bangalore. His research interests include operating systems, computer networks, mobile computing and wireless communication.

Shiva Shankar Ramanna currently working at the Department of Computer Science and Automation, Indian Institute of Science, Bangalore. His research interests include operating systems, computer networks, distributed systems, E-commerce and security.

**The World Wide Web has been growing in leaps and bounds. Studies have indicated that this massive distributed system can benefit greatly by making use of appropriate caching methods. Intelligent Web caching can lessen the burden on Web servers, improves its performance and at the same time reduces the network traffic.**

### Introduction

For a long time since its evolution, the internet was primarily used by university researchers and government organizations. The revolutionary application, which brought the Internet into public consciousness, was the World Wide Web, Web or WWW for short. WWW links documents stored on computers all round the world. In essence, WWW allows reference to documents on one computer to refer to textual or non-textual information stored on other computers connected to the Internet anywhere in the world. The world simply becomes one big 'hyper linked' document. This model of client/server consists of portable universal clients that talk to larger servers that stores the documents. The Web achieves a huge reach by using highly portable protocols (mainly hypertext transfer protocol, abbreviated HTTP) on top of the protocol used by the internet, TCP/IP, meaning that the Internet forms its backbone. As such, the exponential growth of the Internet can be attributed to the phenomenal surge of the usage of WWW. In other words, the ever-increasing Internet traffic has been dominated by a high percentage of HTTP traffic, with earlier protocols like FTP, Gopher, etc., now taking a back seat.

The HTTP, which forms the backbone of WWW, views data as Objects (or Web objects) – which may be HTML pages, images

Keywords
World wide web, data caching, internet traffic, webpage access.

and files. Out of billions of Web pages on the Web it has been shown that a few of them are accessed far more frequently than others. The concept of Web cache arose to make use of this unique feature.

A Web cache is a dedicated computer system, which stores frequently accessed Web objects. It will be located within the Internet in order to monitor these Web object requests. On subsequent requests for the same object, the cache delivers the object from its storage rather than passing the request on to the original server. Every Web object changes over time, so each Web object has a lifetime before it becomes 'stale'. Caches determine whether their copy of an object is 'stale' and if so, retrieve a new copy from the origin server. The higher the number of people requesting the same object during its useful life, the more upstream traffic the cache eliminates.

The exponential growth of the WWW can be better understood by looking at the graph of the Internet domain survey host count (Web servers), as shown in *Figure* 1. As the WWW continues its exponential growth (the size of static Web pages increases approximately 15% per month), two of the major problems faced by today's Web users are network congestion and server over-

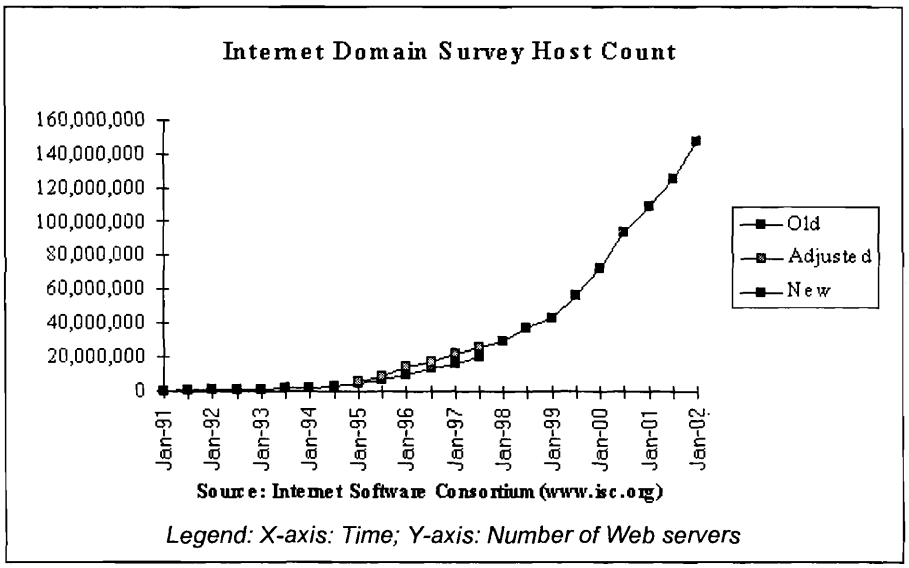


***Figure 1. Internet domain survey host count.***

If some kind of solution is not found for the problems caused by its rapidly increasing growth, the WWW would become too congested and its entire appeal would eventually be lost.

loading. The rapid growth of the WWW could be attributed to the fact that at least till now, its usage is quite inexpensive, and accessing information is faster using the WWW than using any other means. Also, the WWW has documents that appeal to a wide range of interests, e.g. news, education, medicine, scientific research, sports, entertainment, stock market, travel, shopping, weather, maps, etc.

Although the Internet backbone capacity increases as much as 60% per year, the demand for bandwidth is likely to outstrip supply in the foreseeable future as more and more information services are moved onto the Web. If some kind of solution is not found for the problems caused by its rapidly increasing growth, the WWW would become too congested and its entire appeal would eventually be lost. The unparalleled growth of the Internet in terms of total bytes transferred among hosts, coupled with the sudden dominance of the HTTP protocol, suggest much can be leveraged through Web caching technology. Specifically, Web caching becomes an attractive solution because it represents an effective means for reducing bandwidth demands, improving Web server availability, and reducing network latencies.

## Functions of a Web Cache

The main function of a Web cache is to store the frequently accessed Web pages locally thereby making Web access faster. It accumulates the requests and sends a single individual request in their place to the destination server. On acquiring the requested data, it forwards it to the requester, making copies of that data within itself. Browsers retrieve portions of data from the cache rather than directly from the server. Thus, Web caches can help lighten the load on a Web server by reducing the number of incoming requests. However, most Web content providers neither have access nor control on which users or how many users arrive at their site. The cache server needs to sit nearer to the user end than to the Web server end. (Web load-balancing schemes distribute the incoming load across multiple servers at the Web content provider end.)

In addition to reducing outgoing traffic by bundling duplicate requests from browsers, Web caches act like Web routers, assisting in sending the Web traffic efficiently over a network. While Internet Protocol routing directs low-level traffic (individual IP packets) irrespective of the data contents, Web routing directs application-specific HTTP traffic across the network. Because Web traffic constitutes most of the Internet traffic, improving Web routing can improve the overall performance of the Internet.

In addition to reducing outgoing traffic by bundling duplicate requests from browsers, Web caches act like Web routers, assisting in sending the Web traffic efficiently over a network.

Web caching is similar to memory system caching – a Web cache stores Web resources in anticipation of future requests. However, significant differences between memory caching system and Web caching result from the non-uniformity of Web object sizes, retrieval costs, and cacheability. Once the cache server receives a Web request, it checks its database to see if it has the contents of the requested page stored somewhere. A successful retrieval from the local cache is called a *cache hit*, and an unsuccessful one is called a *cache miss*. In the case of a cache miss, the request is forwarded to the Web server or the next caching point. This server begins its own access to the requested URL. Such a first-time access to a page forces the cache server to contact the origin Web server that hosts the page. The cache server checks to see if the page can be cached, retrieves the data to cache locally, and, at the same time, passes the contents to the client. The user may never realize that the cache is between the client and server except in special circumstances

It is important to distinguish between Web cache and a proxy server as their functions are often misunderstood. Proxy servers serve as an intermediary to place a firewall between network users and the outside world. A proxy server makes the outgoing network connection more secure, but does little to reduce network traffic. Web caches can help lighten the load on a Web server by reducing the number of incoming requests, by storing frequently accessed pages. But if the required Web page is found on the proxy server, then this proxy server in now acting as the Web server. But it still has other functions to perform, like maintaining the firewall. The caching technique or architecture

Statistically speaking, a Web cache could eliminate at least 30% of the Web traffic that would normally be going out over a wide area network.

usually used on a proxy server is proxy caching. But this is not a must, meaning a proxy server can be using a transparent caching mechanism, as will be explained later.

The most obvious beneficiary of Web caching is the user, who avoids some traffic snarls when browsing. The network administrator and the remote Web site also benefit. According to the National Laboratory for Applied Network Research (NLANR), large caches with lots of clients may field as many as 50% of the hits that would otherwise travel through a network individually to the origin site. A typical cache could easily field about 30% of the intended hits, says the NLANR's 1996 research. Thus, statistically speaking, a Web cache could eliminate at least 30% of the Web traffic that would normally be going out over a wide area network (WAN).

The bandwidth between institutional networks and regional ISP (Internet Service Provider) and that between transoceanic and regional networks plays a vital part in the Web traffic. Web caching reduces this bandwidth consumption, thereby decreases network traffic and lessens network congestion. Overloaded Web servers and congested exchange points are the main reason for Web latency. Caching reduces access latency due to two reasons:

a. Frequently accessed documents are fetched from nearby cache servers instead of remote data servers, minimizing the transmission delay.

b. Because of the reduction in network traffic, those documents not cached can also be retrieved relatively faster than without caching due to less congestion along the path and less workload at the server.

If the remote server is not available due to the remote server's crash or network partitioning, the client can obtain a cached copy at the cache server. Thus, the robustness of the Web service is enhanced.

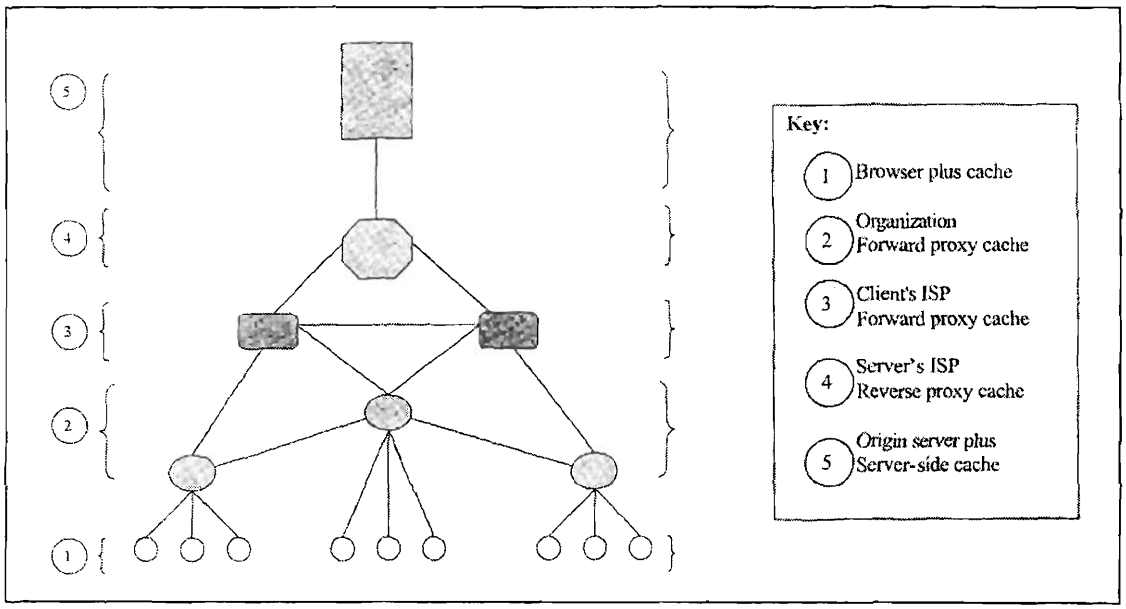


***Figure 2. Web caching architecture – a possible chain of caches in the WWW through which a request and responses might flow. Starting in a browser, Web request can travel through multiple caching systems on its way to the origin server. At any point in the sequence a response can be sent if the request matches a valid copy of the requested data in the cache.***

Web caching reduces the workload of the remote Web server by disseminating data among the cache servers over the wide area network. This conserves the Web server resources. Furthermore, a group of caches co-operating with each other in terms of serving each other's requests and making storage decisions result in a powerful paradigm to improve cache effectiveness. An added bonus of making use of Web caching is that network administrators and network analysts get a chance to analyze an organization's usage patterns.

The main disadvantage is that a client might be looking at stale data due to the lack of proper cache server updating. The access latency may increase in the case of a cache miss due to the extra cache server processing. Hence, cache hit rate should be maximized and the cost of a cache miss should be minimized when designing a caching system.

A single cache server is always a bottleneck. A limit has to be set for the number of clients a cache server can serve. An efficiency lower bound (i.e. the cache system ought to be at least as efficient as using direct contact with the remote servers) should also be enforced. The reliability of the system can go down since single

The performance of a Web cache system depends on the size of its client community; the bigger the user community, the higher is the probability that a cached document (previously requested) will soon be requested again.

cache server is a single point of failure. Using a cache server will reduce the hits on the original remote server, which might disappoint a lot of information providers, since they cannot maintain a true log of the hits to their pages. Hence, they might decide not to allow their documents to be cacheable. Finally, the caching of dynamic objects may sometimes increase the overheads as it may require user authentications.

Web caching originally began as a single-server system that contained all the data of the cache. In this *Single Level Architecture*, a single server runs out of disk space to store the requested pages or cannot process the incoming requests fast enough. With reference to *Figure 2*,we can see three client browsers connected to one organization forward cache. An improvement over this method would be to have more than one Cache servers running and cooperating at a particular level called the *Parallel and Load balancing Architecture*. This is illustrated in *Figure 2*, wherein we see the Client's ISP Forward proxy caches cooperating among themselves at level three. *Multi Level Architecture* caching works almost like the single-cache servers. Here, if there is a cache miss at one server level, the request is propagated up to the next higher level to see if that cache contains the data. Only when the request hits the top level and still encounters a cache miss will the cache server go directly to the origin Web site to retrieve the data. This is illustrated in *Figure 2*, wherein we have shown a multilevel hierarchy consisting of five levels.

The performance of a Web cache system depends on the size of its client community; the bigger the user community, the higher is the probability that a cached document (previously requested) will soon be requested again. A caching architecture should provide the paradigm for proxies to cooperate efficiently with each other.

## Cache Placement or Deployment

Caching could be done at various locations and network points: near the content consumer (consumer-oriented), near the con-

tent provider (provider-oriented), and at strategic points in the network, based on user access patterns and network topology.

Positioning caches near the client, as in proxy caching has the advantage of leveraging one or more caches to a user community. If those users tend to access the same kind of content, this placement strategy improves response time by being able to serve requests locally.

Caches positioned near or maintained by the content provider, on the other hand, improve access to a logical set of content. This type of cache deployment can be critical to delay-sensitive content such as audio or video. Positioning caches near or on behalf of the content provider allows that provider to improve the scalability and availability of content.

Caches positioned near or maintained by the content provider improve access to a logical set of content. This type of cache deployment can be critical to delay-sensitive content such as audio or video.

The use of both consumer-oriented and provider-oriented caching techniques is perhaps the most powerful and effective approach, since it combines the advantages of both while lowering the disadvantages of each.

The last approach is the dynamic deployment of caches at network choke points. Although it would seem to provide the most flexible type of cache coverage, it is still a work in progress and, to the best of the authors' knowledge, there have not been any performance studies demonstrating its benefits. The dynamic deployment technique also raises important questions about the administrative control of these caches.

Finally, a discussion about cache deployment would not be complete without noting the capabilities of browsers to do caching on a per-user basis using the local file system. Obviously, while browser caching is useful for a given user, it does not aid in the global reduction of bandwidth or decline in average network latency for common Web projects.

## Conclusion

As it is with any evolving technology, Web caching techniques are changing rapidly, especially being of interest to both the

research universities and the industry. As mentioned earlier, we have made an attempt to capture the state-of-art schemes and methodologies involved. Hence we would like to modestly remind the readers that issues which are still being evaluated by the research community might have not found a place in this discussion.

## Acknowledgement

We would like to express our sincere gratitude and thanks to Dr.Y Narahari, Professor, Department of Computer Science and Automation, Indian Institute of Science, Bangalore, for his help, encouragement and intellectual influence, which made this paper possible. His invaluable guidance, reviews and suggestions have been instrumental in the successful completion of this work.

*Address for Correspondence*
Harsha Srinath
No.44, Dwaraka
39th Cross
Jayanagar 8th Block
Bangalore 560 070, India.
Email:
harshasrinath@hotmail.com

Shiva Shankar Ramanna
No.113/A, 9th Main
12th Cross, 3rd Phase,
Girinagar
Bangalore 560 085, India.
Email:
get_shiva@rediffmail.com

## Suggested Reading

[1] **WWW: http://www.web-caching.com/, http://www.caching.com.**

[2] **Greg Barish and Katia Obraczka, World Wide Web Caching: Trends and Techniques, USC Information Sciences Institute, *IEEE Communications Magazine*, May 2000.**

[3] **Rawn Shah, *Reduce network traffic with Web caching, Server Web caches speed access to Web pages and ease network traffic*, IBM Developer Works, Library paper, September 1999**

[4] **Jia Wang, *A Survey of Web Caching Schemes for the Internet*, Cornell Network Research Group (C/NRG), Department of Computer Science, Cornell University, 2000.**

Perhaps the most famous example of futurology gone wrong is the predictions made by John von Neumann, the father of the modern electronic computer and one of the great mathematicians of the century. After the war, he made two predictions: first, that in the future computers would become so monstrous and costly that only governments would be able to afford them, and second, the computers would be able to predict the weather accurately.

*Michio Kaku*
*Hyperspace*
Anchor Books, Doubleday, New York, USA, 1994, p.277