Distributed Cache Management Architecture

To reduce the Internet Traffic by integrating Browser and Proxy caches

Al Imtiaz
Department of CSE
University of Information Technology and Sciences (UITS)
Dhaka, Bangladesh
al.imtiaz@yahoo.com

MD. Jayed Hossain
Department of IT
University of Information Technology and Sciences (UITS)
Dhaka, Bangladesh
jrupul@gmail.com

***Abstract*—The World Wide Web is one of the most popular Internet applications, and its traffic volume is increasing and evolving due to the popularity of social networking, file hosting, and video streaming sites. Wide ranges of research have been done on this field and numbers of architecture exist for caching those web content. Each of those has their own advantages and limitations. Browser caches handle single user by caching and storing web content on user computer. Where the proxy caches could handles thousands of users by handling, providing, and optimizing those web contents. But the World Wide Web (WWW) suffers from scaling and reliability problems due to overloaded and congested proxy servers. Distributed and Hierarchical architecture could be integrated as hybrid architecture for better performance and efficiency. Based on the secondary information by literature review, this paper is aimed to propose few feasible strategies to improve the cache management architecture by integrating browser with proxy caches server, where the browser cache will act as proxy cache server by sharing its content through hybrid architecture. This paper will also focus on the present architecture and challenges of current system that are needed to be resolved.**

Keywords—Web cache; Browser cahe; Distributed cache; Web Traffic; Cache management

I. INTRODUCTION

In this age of digitalization and globalization the appeal of better internet facility is become one of the major issues. Web traffics have been increasing tremendously and it becomes a fundamental issue to provide better experience to the end user by managing those traffics [1]. These growths of web traffic are expected to continue, not only as the web becomes a de facto front-end for many emerging cloud-based services, but also as many applications getting migrated to the Web [2].

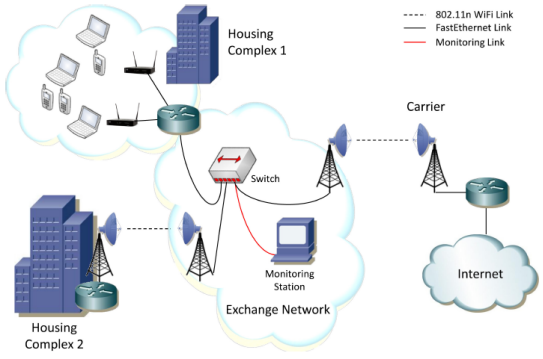
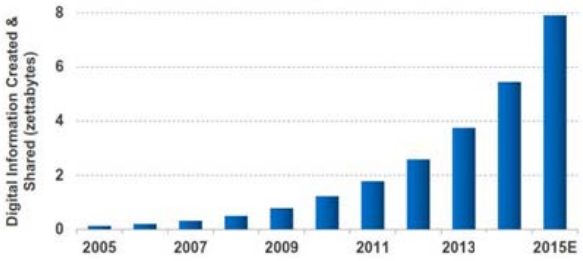


Fig. 1. An approach of monitoring Web Traffic [4]

Web traffic is the amount of data traveled on the internet and those data are sent and received by the visitors of a web site and server. This internet has already become giant networks which have to maintain huge amounts of web traffic in every moment. This huge web traffic may cause the slowness of the entire system according to the growth rate shown in Figure 2. The amount of Global digital information created and shared will be double within next two years according to the current growth rate of web traffic (Figure 1) [3].

Web traffic is been acquired from the access log of web data. It only assumes simple and static Web pages, and once Web pages are identified, other contents can be derived, such as the number of embedded objects, total page size, total page time, and inter-arrival time (Figure 1).



Note: 1 Zeta byte= 1 trillion Gigabyte
data of 2014 and 2015 are estimated

Fig. 2. Global growth rate of web traffic

There are very few popular website those are browsed more frequently. Figure 3 shows the top 10 web sites. Among them 80% of top ten global internet properties are owned by USA [3].

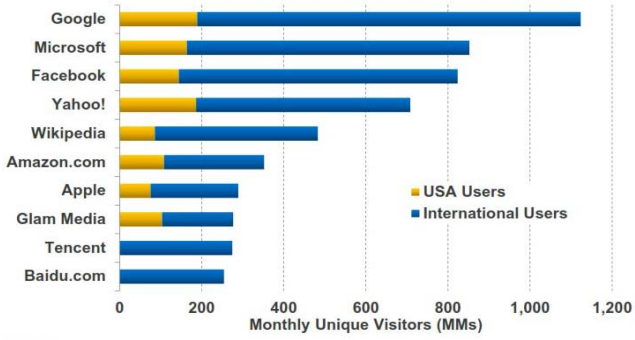


Fig. 3. Top 10 Internet Properties by global monthly unique visitors, Feb, 2013

Web caches is being used to handle this huge amount of web traffic by storing frequently browsed web content on the browser of client's computer or local proxy server. Wide range of researches have been done on this field and numbers of architecture have been developed for cache management. This paper is aimed to propose few feasible strategies to improve the cache management architecture by integrating browser and proxy caches. The main aim of this proposed architecture is:

- To distribute the load of proxy server
- To minimize the response time
- To reduce the Web Traffic

This architecture will also reduce the amount of bandwidth used by a client. This will also save money if the client is paying for the traffic.

II. INTRODUCTION TO WEB CACHE

Web cache is a mechanism to store the web contents temporary (caching), such as HTML pages and images, supporting style sheet and other scripts this. Cache helps to reduce Network traffic, bandwidth usage, loading time, and lighten the load on a Web server by reducing the number of incoming requests; browsers retrieve portions of data from the cache rather than directly from the server. Web cache stores copies of documents passing through it; then, if there is another request for the same URL, it can use the response that it has stored previously, instead of asking the origin server for it again. There are three types of web cache.

A. Browser Caches

Browser's "Cache" is used for the temporary storage of Web pages on your hard disk. Every web page retrieve from the server for the first time, the browsers cheek the cache first time, if there are the same content then it server form the cache . Then the browser attempt to read the page from its "cache", instead of spending the extra time and network traffic to download it again.

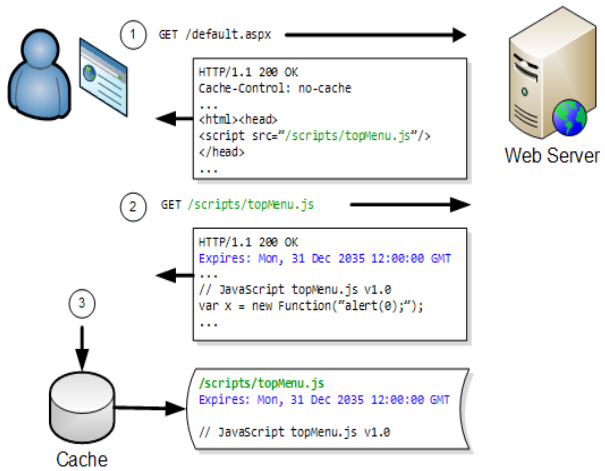


Fig. 4. Browser store cacheable contents

The Figure 4 shows how a JavaScript file is saved into the browser cache on the first visit to a web page and Figure 5 shows how browser uses that cached contents:

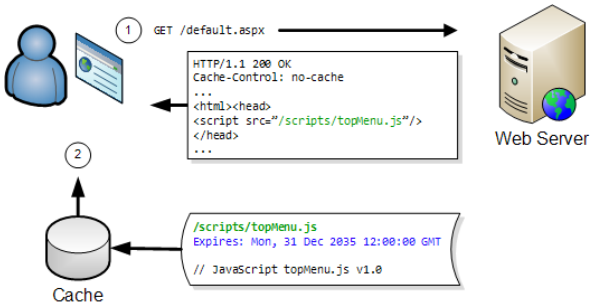


Fig. 5. Browser cache is being used.

This browser cache helps to avoid a network round trip and it is typically 100 to 1000 times faster than downloading the file proxy server.

B. Proxy Caches server

A proxy cache server is a shared network device that can undertake Web transactions on behalf of a client, and, like the browser, the proxy cache server stores the web content. Subsequent requests for this content, by this or any other client of this network of the cache will trigger the cache to deliver the locally stored copy of the content, without downloading the original content from server.

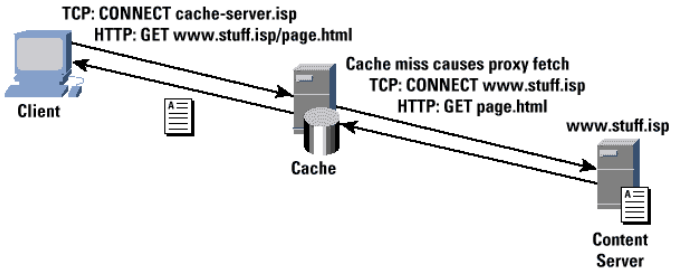


Fig. 6. Proxy Caches Server [5]

Proxy caches aren't part of the client's content or the server content. Proxy caches are one kind of shared cache; rather than just having one person using them, they usually have been used by a large number of user by this they are reducing latency and network traffic. That's because popular representations are reused a number of times.

Proxy servers may also perform additional tasks, such as compressing data and optimizing documents. Systems that assist in such way are referred to as web accelerators. Sometimes caching proxy servers are most useful in distributed environments, where the proxy server can act as both a load balancer and optimizer.

C. Server Side Caches

Also known as "reverse proxy caches" or "gateway caches," Server Side Caches are also intermediaries, but instead of being deployed by network administrators to save bandwidth, they're typically deployed by Webmasters themselves, to make their sites more scalable, reliable and better performing.

III. WEB CACHING ARCHITECTURES

Multiple Web-caching servers can be used together to provide for more efficient caching. There are two basic

caching architectures that use multiple caching servers working together:

- Distributed Caching
- Hierarchical Caching

As the name implies, distributed caching distributes, or spreads, the cached Web objects across two or more caching servers. These servers are all on the same level on the network. Figure 7 illustrates how distributed caching work [6].

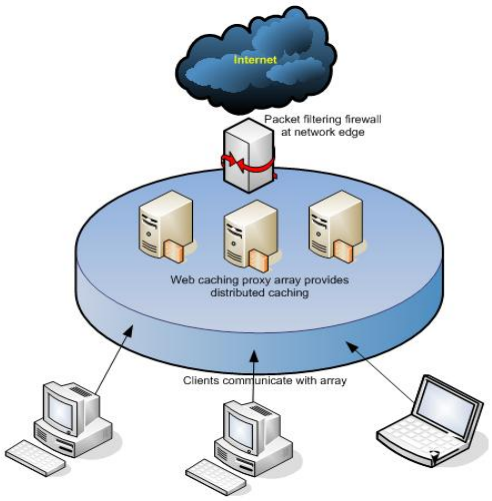


Fig. 7. Distributed architecture of web cache.

Hierarchical caching works a little differently. In this setup, caching servers are placed at different levels on the network. Upstream caching servers communicate with downstream proxies.

Hierarchical caching is more efficient in terms of bandwidth usage, but distributed caching is more efficient in terms of disk space usage.

Finally, you can combine the two methods to create a hybrid caching architecture. The combination gives you the "best of both worlds," improving performance and efficiency.

A. Web Cache control

HTTP 1.1 introduced a new kind of validator called the ETag. ETags are unique identifiers that are generated by the server and changed every time the representation does.

Almost all caches use Last-Modified times as validators. This is the time that the document last changed. When a cache has a representation stored that includes a Last-Modified header, it can use it to ask the server if the representation has changed since the last time it was browsed.

Most modern Web servers will generate both ETag and Last-Modified headers to use as validators for static content (i.e., files) automatically. ETag validation is also becoming prevalent.

Validators are very important; if one isn't present, and there isn't any freshness information (Expires or Cache-Control) available, caches will not store a representation at all.

Together, freshness and validation are the most important ways that a cache works with content. A fresh representation will be available instantly from the cache, while a validated representation will avoid sending the entire representation over again if it hasn't changed. Figure 8 shows a generic view of cache control.

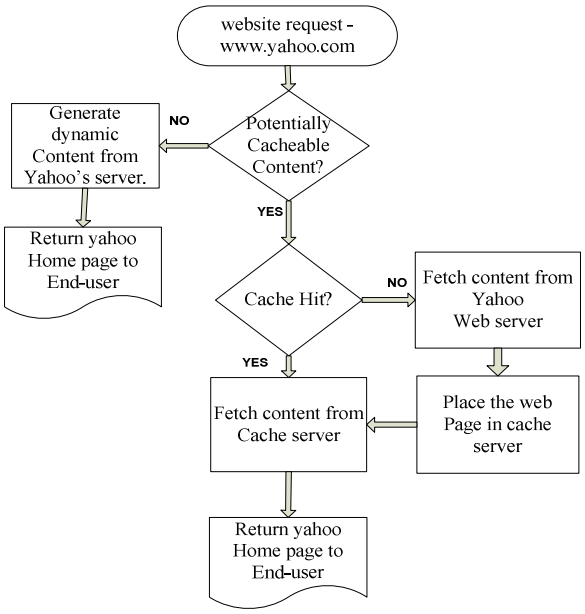


Fig. 8. How Web cache works

B. Web Caching Protocols

When multiple Web caching servers work together, they need a way to communicate with each other. There are a number of different protocols that can be used for communicate with Caches.

- Cache Array Routing Protocol (CARP),
- Internet Cache Protocol (ICP),
- Hypertext Caching Protocol (HTCP),
- Web Cache Coordination Protocol (WCCP),
- Cache Digests.

C. Cache-Control HTTP Headers

HTTP 1.1 introduced a new class of headers, Cache-Control response headers, to give Web publishers more control over their content, and to address the limitations of Expires. Cache-Control response headers includes; max-age=[seconds] , s-maxage=[seconds] ,public, private , no-cache, no-store ,must-revalidate ,proxy-revalidate and so on.

Example: Cache-Control: max-age=3600, must-revalidate

When both Cache-Control and Expires are present, Cache-Control takes precedence.

IV. PROPOSED ARCHITECTURE

The goal of this project could be achieved by introducing an advance integrated cache management architecture. Decentralization of proxy server by shareable browser cache will reduce the load of proxy server, network traffic and web pages will be more faster. Normally Browser caches are stored in user computer. The aim of this project is to make shareable those caches with others.

The load of proxy server will be distributed among user's computer. Proxy server/ local cache server have to maintain a

log of cache contents with their local sources (user computer). A Browser plug-in "Cache Manager" could be introduced to handle intra-networked cache requests. Figure 9 shows how the proposed architecture will work.

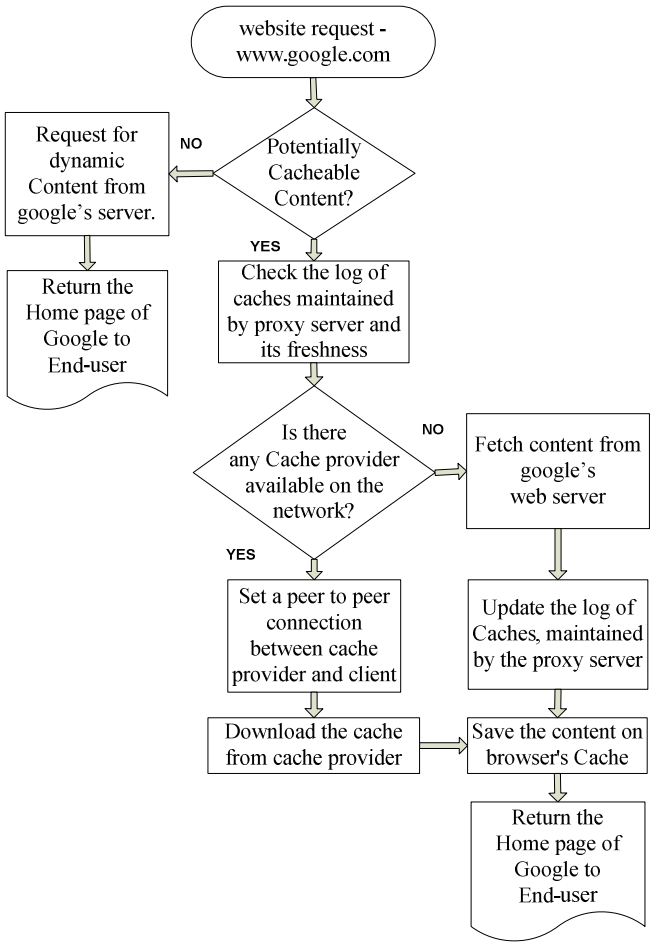


Fig. 9. Proposed Distributed Cache Management Architecture

Here the proxy server is playing a vital role in this architecture. Proxy server will check the availability of cache provider, which have the requested cache and alive on the local network. If there are any, then it will set a peer to peer connection with client and cache provider. The proxy is responsible to check the freshness of cache by using the E-Tag or/and modification date of that contents. If the cache failed to pass the validation test, then the cache will be fetch from main server.

V. CHALLENGES

There are three sensitive issues of this architecture. Those issues are prime challenges of this proposed solution.

A. Security

The security of caches is one of the major issues of this proposed solution. The caches will be locally stored and shared among each other. Any one may inject malware or virus with those caches. Encryption could be a handy solution for this issue.

B. Multiple Browsers

Most of the user may use more than one browser on their computer. And the user may not install this plug-in on all browser, if they do then synchronizing and maintaining those caches of all browser is another issue.

C. Dynamic Contents

Dynamic contents are not cacheable, because the contents have been served from the database of main server.

VI. RECOMMENDATION

There are few more options to make a web site more cacheable. Besides using freshness information and validation, there are a number of other facts those are also related with cache validation.

Use URLs consistently — this is the easiest and most effective way to make a site cache-friendly.

Make cacheable images and pages that don't change often by using a Cache-Control: max-age header with a large value. Just change the file name to whenever needed.

Minimize use of SSL — because encrypted pages are not stored by shared caches, use them only when you have to, and use images on SSL pages sparingly.

VII. CONCLUSION

The performance of internet browsing depends on the bandwidth and types of data being requested from the server. Cache has been used to reduce the internet traffic and improve the performance. The present architecture could be improved by the Proposed distributed cache management architecture. This architecture will reduce the network traffic, distribute the load of proxy server, and improve the user experience.

REFERENCES

[1] Lucian Popa, Ali Ghodsi, and Ion Stoica. "HTTP as the Narrow Waist of the Future Internet". in Proc. 9th ACM Workshop on Hot Topics in Networks (Hotnets- IX), Monterey, CA, October 2010.

[2] Craig Labovitz, Scott Iekel-Johnson, Danny McPherson, Jon Oberheide, and Farnam Jahanian. "Internet inter-domain traffic." in Proc. ACM SIGCOMM, New Delhi, India, August 2010.

[3] Mary Meeker , Liang Wu, " Internet Trends", in D11 Conference, May 2013

[4] Steffen Gebert1, Rastin Pries, Daniel Schlosser, and Klaus Heck, "Internet Access Traffic Measurement and Analysis" in Proc. 4th international conference on Traffic Monitoring and Analysis, Pages 29-42, 2012.

[5] Geoff Huston, Telstra, "Web Caching" on The Internet Protocol Journal - Volume 2, No. 3, september 1999.

[6] Thomas Shinder, "Understanding Web Caching Concepts for the ISA Firewall", on isaserver.org , september 2008.

[7] Andrzej Sieminski, "The impact of Proxy caches on Browser Latency" on International Journal of Computer Science & Applications, *Vol. II, No. II, pp. 5 – 21, 2005*