Volume 3, Issue 11, November 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Study on web Caching Architecture: A Survey

Mukesh Dawar, Charanjit Singh
Department of Computer Science & Engineering
RIMT Institute of Engineering & Technology, Mandigobindgarh, India

Abstract— These days, the web has grown tremendously to fulfill the requests from huge number of users. Caching strategies are applied to minimize the retrieval latency of objects by a user. The contents are replicated using proxy servers closer to the users to reduce the network distance between the object and the user. In this paper, a study has been presented for the evaluation of several caching schemes and their issues. At the end, it has been observed that cache cooperation can improve the performance at a larger scale. The cache cooperation at each level of hierarchy also minimizes the client's retrieval latency. The paper also discusses some approaches to manage Scalability & Metadata Server, extra overhead at overloaded proxy servers by introducing dynamic functionality in distributed web caching.

Keywords: Web caching; proxy server; Meta data; Scalability; latency; Robustness.

I. INTRODUCTION

As the World Wide Web (WWW) is gaining more and more popularity, servers have to handle more requests accordingly. The more people (or simply clients) request resources from web servers, the faster servers have to accept and process the requests. To cope with these requirements programmers as well as system administrators must take countermeasures. The web has become the most challenging and successful application. A network has to deal with the problems of Scalability, low latency, frequent disconnections of servers, congestion at servers and other unmanageable conditions due to increase in the number of users day by day. Clients experience unexpected delays while accessing web pages. All these problems and latencies must be maintained under a tolerable limit. These latencies, congestions and server's load can be handled by storing multiple copies of web documents in geographically distributed web caches for most of the contents are being static. Caching reduces latencies to users and also web traffic.
Caching was introduced to reduce the latency of retrieval. Mosaic [3] was the first web browser that was capable of caching web objects for reference and thus reduces the latency to clients and network traffic. After that Web caching has grown rapidly from local cache of a single browser to large shared cache to serve multiple clients from a certain institution [14].

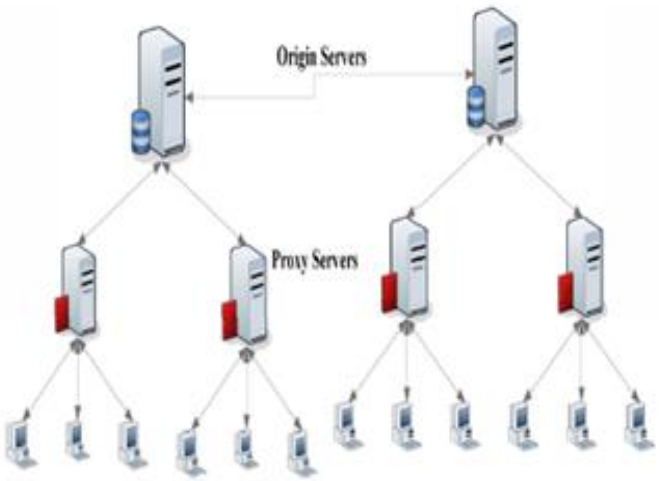

Fig-1 General Architecture of Web Caching.

General architectures of these kinds of networks are client-server architecture [19]. All servers are located remotely to fulfill the requests from the clients for accessing the web pages. This results in huge delays at every request if every time the requested page is to be fetched from the remote servers. A modification of client server architecture is proxy caching. If more users are accessing the same page then there will be high probability of hits. Two popular and common approaches for implementing cache cooperation scheme at a large scale are distributed [2] and hierarchical caching [10], [12], [2], [1]. We will discuss them in later sections.

In the remaining space, we will discuss architecture of cache in section 2. The Section 3 is dedicated for the issues and challenges related with caching in distributed environment. This is followed by conclusion and future research directions.

II. WEB CACHING ARCHITECTURE

Jacobson [6] proposed that data should be found through sources that are local to users rather than their place of origin for all attempts. With the exponential growth of internet in last decade, a main challenge remains to provide faster access to data and management of cached data. The question arises was, "how to architect lots of caches?" [23] Several architectures were proposed for fulfilling above objective some of these are:-

2.1 Single and Multiple caches:

Shim et al. [15] introduced a single cache algorithm that was relatively simple with only one proxy cache. They also considered both consistency maintenance and replacement polices for web proxies. Fan et al. [4] has described a protocol for multiple caching named "summary cache". In this each proxy maintains a summary of cache directory of every participating cache. It checks summary before sending data for any queries of potential hits. To handle misses and for reducing total traffic, these caches cooperate with each other.

2.2 Hierarchical Caching:

Harvest project [2] proposed the concept of Hierarchical caching for web. It works with sharing of interests of large group of clients and also several countries have implemented this [9]. In this arrangement caches are placed at different levels of network with hierarchical caching as shown in figure-2. Client caches are at the bottom level of hierarchy [12]. The request is redirected to next level cache that is institutional cache if it was not satisfied by the client cache. For the case if document is not present at this level too then the request travels to the regional cache. Regional cache in turn forwards all the requests that are not satisfied to the national cache. If the document is not part of any cache level, then national cache is responsible for extracting the document directly from the origin server. After the document is found at origin server or any of cache level it travels down the hierarchy and also leaves copy of document at each intermediate level. All the time request travels up the hierarchy until the document of interest is found. But there are several problems associated with this scheme are: - Extra delay introduced at each level of hierarchy, there might be long query delays at higher levels and redundancy of data at different caching levels.

In [5] a Central Directory Approach (CRISP) is presented in which certain number of caches are ties together through a central mapping service. This approach could be deployed at any level of global or regional cache hierarchy to maximize capacity. According to their experience CRISP can scale up to hundreds or thousands of clients.

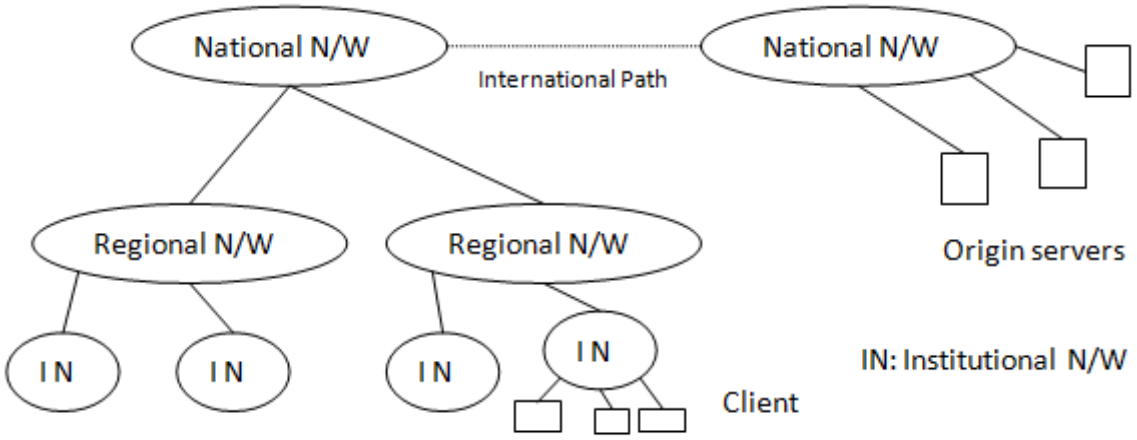

Fig-2 Hierarchical Caching Architecture

2.3 Distributed Caching:

In this type of architecture there are no intermediate caches, only institutional caches at the edges of the network. These caches cooperate with each other to serve the misses. As there are no intermediate caches so to centralize the documents requested by the lower level caches some another sharing mechanisms are used by institutional caches:

- Institutional caches can transfer the query to their cooperating institutional caches for all local misses. This all is done by using Inter Cache Protocol (ICP) [22]. But this approach increases the bandwidth consumption and latency.
- Institutional caches can maintain the summary of the contents of their cooperating caches. It eliminates the need for querying the cooperating caches. These summaries are exchanged periodically. For effective distribution of summary a hierarchical structure of intermediate nodes can be applied [10] that only distributes information about the location of the documents not the actual documents.
- For cooperating institutional caches, a hash function can also be used [20]. This function map requests by the clients to certain caches. This scheme ensures no duplicate copies for a single document in different caches. But this property limits this scheme to local environments only.

For distributed caching a protocol named ICP (Internet Cache Protocol) [22] and HTCP [21] was designed by NLANR. They supported management, retrieval and discovery of updated data from parent cache and neighboring caches as well.

Cache Array Routing Protocol (CARP) is another approach for distributed caching [20]. In this URL space is divided among an array whose elements are loosely coupled caches. Each document is hashed to a particular cache.

To resolve the URL space Karger et al. [7] proposed an approach similar to CARP that makes use of DNS. This scheme also allows replication of frequently used contents in several caches. A large scale distributed cache is proposed by Harrison and Povey [10], in this scheme directory server has replaced the upper level caches. These directory servers store the hints about the location of all documents kept in caches. These locations hints are maintained by hierarchical structure that results in scalable distribution. Tewari et al. [17] proposed a similar scheme for fully distributed internet cache architecture, in which location hints are maintained in the same way but replicated at all local institutional caches.

In Cache Digest [13] and in the Relais Project [8] all caches maintain local directories of contents of other caches for ease of locating documents in other caches. Also caches keep exchanging messages with each other indicating their contents.

2.4 Hybrid Caching:

This scheme is combination of hierarchical and distributed caching scheme. In this, caches at every level of hierarchy cooperate with each other at the same level or with higher level caches of hierarchy using distributed caching. This reduces latency of retrieving popular documents. This approach handles the inefficiencies of previous approaches. In [19], they presented the dynamic behavior of proxy servers for handling cache's misses on the basis of their loads that is if a proxy server is congested it can deny more requests. Those requests will be handled by some other proxy servers at the same or other level. It will result in reduced transmission time, connection time and also lower retrieving latency.

In [22], authors have presented hybrid scheme, with the concept of caches cooperation at each level in hierarchy. Rabinovich et al. [11] have altered this scheme by limiting the cooperation between the neighboring caches. This scheme advantages by avoiding fetching of documents from the slower or distant caches if that document could be retrieved at a lower cost directly from the origin server.

The author in [18] provides a solution for robustness and scalability problem in web caching due to heavy load. They have used the concept of clustering along with the feature of dynamic allocation of requests by maintaining metadata of neighbors. They provide the concept of managing the load of overloaded server by transferring requests to less loaded proxy servers. In [19] the author has refined their scheme of [18] for hybrid caching to handle more delays and frequent disconnections of proxy servers. This can result in fastest response to the clients and also provide load balancing.

2.5 Other Caching Structures:

Another transparent structure called en-route caching was developed around 2002. In this each request is intercepted by the en-route cache that passes through it. The object is transfer to the client if it is present in the cache otherwise the request is forwarded to the next node along the regular routing path [16]. This scheme has certain advantages. It reduces the network delay for cache miss as the requests are not detoured and also eliminate extra overhead of broadcasting the queries. This scheme is also transparent to both the client and the content server.

III. SOME PROBLEMS AND ISSUES OF CACHE MANAGEMENT

In this section some general problems and issues related with distributed web caching architecture are discussed.

3.1Extra Overhead

Overhead at proxy server increases when it maintains records of all other proxy servers. It also results into more congestion at proxy servers. Each server must keep checking the validity of their data and this leads to extra overhead on servers. Updating and exchanging of Meta data on proxy servers keeps them unnecessarily busy.

3.2 Size of Cache

If size of cache grows it results into larger Meta data that becomes unmanageable at proxy servers as they also keeps Meta data of all other proxy servers too. In this way cache's size become bottle neck for the maintenance of larger Meta data.

3.3 Cache Coherence Problem

Client must always receive an up-to date data from the proxy server when ever requested. This requirement can result into cache coherence problem. All changes of data at the proxy server cache must be simultaneously reflected to the main server too. If this is not possible there should be some provision to update the cache data for maintaining their coherency.

3.4 Scalability

As numbers of clients are growing day by day and also clients of one region can request for connectivity, scalability can be an issue as every server has an upper limit to support the client's connections. If the servers provide the connectivity to more clients then they will become congested and will not be able to provide further connectivity. So scalability should be as high as possible. Clustering can be solution for scalability problem. Through this more number of clients can be served and client's data is maintained on cluster basis.

3.5 Robustness

As proxy servers can serve a limited number of requests, they hang up if the request exceed their limit and started get down the links. In these cases the client request are not fulfill for connections as they always requests to the same proxy server they are connected. These links need to be reset. Clustering can be a solution to this problem and can help to serve most to requests.

3.6 Hit Ratio

If the requested document is present in the proxy cache, that is called a hit. The hit ratio in web caching should be high enough so that user's request can be served from the pages cached on the cache server instead of forwarding the request to other proxy server. In case of congestion hit ratio will decrease drastically as all the users wait for the requested documents. This degrades the network performance so an approach that can ensures high hit ratio is always an issue.

3.7 Balancing Of Load

The scheme of balancing the loads on the proxy servers is a major issue in web caching. If there is no predefined criteria is set the clients can connect to same proxy servers, if there is no limit on the number of clients to proxy servers. This means one server will be overloaded and other proxy server will remain ideal, the busy proxy server get congested and may be down later, so a proper load balancing strategies for proxy server is a major requirement.

3.8 Low Latency

As caching provides easy handling of clients request by the proxy servers and speedup the reply process for the clients, so techniques should provide lower delays for all requests as possible.

3.9 Frequent Disconnections

Interrupts to continue service can arise due to some unmanageable situations that lead proxy servers to be disconnected. So scheme must be provided that can observe smaller disconnection of proxy server and should be able to maintain consistency and recover back all the Meta data.

IV. CONCLUSIONS

Web services have become very popular today, but server overloading and network congestion etc. have become major issues. They directly affect the performance of web. Web caching has come up as a great solution for all these problems and issues. A number of caching schemes already exits. Some effective techniques such as hierarchical web caching, distributed web caching and hybrid web caching and etc. schemes have been discussed in this paper. These techniques can handle server's extra overhead and reduce network traffic. We have also discussed some of the problems affecting the performance of web caching and major issues related with distributed web caching. These all issues and problems related to distributed web caching must be resolved to improve performance, hit rate and for lower access latency.

If issues of extra overhead, scalability, latency and coherency are handled carefully, hit ratio can be improved. Although some new research works have proposed new and improved architectures for distributed web caching to handle all these issues, but there are still some problems in distributed web caching such as cache routing, fault tolerance, proxy placement, security, and dynamic data caching, etc which need to be addressed carefully for making cache management a perfect one.

REFERENCES

[1] V. Cardellini, M. Colajanni, P.S. Yu, Geographic Load balancing for scalable distributed Web systems, Proc. Of MASCOTS'2000

[2] A. Chankhunthod *et al.*, A hierarchical internet object cache, in *Proc. 1996 USENIX Technical Conf.*, San Diego, CA, Jan. 1996.

[3] K. Claffy, H.W. Braun (1994).Web traffic characterization: An assessment of the impact of caching documents from NCSAs web server, in Electronic Proc. 2nd World Wide Web Conf.94: Mosaic and the Web.

[4] L. Fan, P. Cao, J. Almeida, and A. Broder, Summary cache: A scalable wide-area web cache sharing protocol, in *Proc. SIGCOMM'98*, Feb. 1998, pp. 254–265.

[5] S. Gadde, M. Rabinovich, and J. Chase, Reduce, reuse, recycle: An approach to building large internet caches, in *Proc. 6th Workshop on Hot Topics in Operating Systems (HotOS–VI)*, May 1997

[6] V. Jacobson (1995). How to kill the internet, presented at SIGCOMM'95 Middleware Workshop, [Online]. Available: ftp://ftp.ee.lhl.gov/talks/vj-webflame.ps.Z

[7] D. Karger, A. Sherman, A. Berkhemier, B. Bogstad, R. Dhanidina, K.Iwamoto, B. Kim, L. Matkins, and Y. Yerushalmi, Web caching with consistent hashing, in *Proc. 8th Int. World Wide Web Conf.*, May 1999

[8] M. Makpangou, G. Pierre, C. Khoury, and N. Dorta, Replicated directory service for weakly consistent replicated caches, in *Proc. ICDCS'99 Conf.*, Austin, TX, May.

[9] National Lab of Applied Network Research (NLANR). [Online]. Available: http://ircache.nlanr.net/

[10] D. Povey and J. Harrison, A distributed Internet cache, in *Proc. 20th Australian Computer Science Conf.*, Sydney, Australia, Feb. 1997

[11] M. Rabinovich, J. Chase, and S. Gadde, Not all hits are created equal: Cooperative proxy caching over a wide-area network, in *Proc. 3rd Int. WWW Caching Workshop*, Manchester, U.K., June 1998

[12] Pablo Rodriguez, Christian Spanner, and Ernst W. Biersack, Analysis of Web Caching Architectures: Hierarchical and Distributed Caching, IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 9, NO. 4, AUG 2001.

[13] A. Rousskov and D. Wessels, Cache digest, in *Proc. 3rd Int. WWW Caching Workshop*, June 1998, pp. 272–273.

[14] A. Rousskov, On performance of caching proxies, in *Proc. ACM SIGMETRICS*, Madison, WI, Sept. 1998.

[15] J. Shim , P. Scheuermann , R. Vingralek (1999). Proxy Cache Algorithms: Design, Implementation, and Performance, IEEE Transactions on Knowledge and Data Engineering, v.11 n.4, p.549-562.

[16] X. Tang , S.T. Chanson (2002). Coordinated En-Route Web Caching, IEEE Transactions on Computers, v.51 n.6, p.595-607.
[17] R. Tewari, M. Dahlin, H. M. Vin, and J. S. Kay, Beyond hierarchies: Design considerations for disturbed caching on the Internet, in *Proc. ICDCS '99 Conf.*, Austin, TX, May 1999.
[18] Rajeev Tiwari, Lalit Garg, Robust Distributed Web Caching Scheme, in International Journal of Engineering Science and Technology in ISSN : 0975-5462 Vol. 3 No. 2 Feb 2011,pp 1069-1076.
[19] Rajeev Tiwari and Neeraj Kumar, Dynamic Web Caching: For Robustness, Low Latency & Disconnection Handling, 2nd IEEE International Conference on Parallel, Distributed and Grid Computing, 2012.
[20] P. Vixie and D. Wessels, RFC 2756: Hyper text caching protocol, (HTCP/0.0), Jan. 2000.
[21] Jing Zhang, A Literature Survey of cooperative caching in content distribution networks, arXiv: 1210.0071v1 [cs.NI], Sep 2012.

