HYBRID WEB CACHING FRAMEWORK FOR REDUCTION OF WEB LATENCY

Ranju Khemka[1], Aruna Jain[2]

[1]*M.Tech Student, Dept. of Computer Science and Engineering, Birla Institute of Technology, Jharkhand, India*
[2]*Associate Professor, Dept. of Computer Science and Engineering, Birla Institute of Technology, Jharkhand, India*

Abstract

Distributed web caching and Hierarchical web caching are two important techniques to minimize the problem of web latency. Web latency is the time taken by user in retrieving the web documents. The main performance problems with distributed caching are longer connection times and overhead such as resolution delay, queuing delay etc in used bandwidth. Whereas the main issue with Hierarchical caching is longer transmission times. So there is a need of a sophisticated combination of a hybrid scheme to effectively reduce web latency. Studies show that combination of Hierarchical and Distributed web caching reduces transmission time and connection time thereby reducing the overall latency time. Our results show that we can improve hit ratio from Hierarchical and Distributed caching strategy by 55% and 42% respectively.

Keywords: *Distributed caching, Hierarchical caching, Hybrid caching, Web Latency, Hit ratio*

----------***----------

1. INTRODUCTION

The unparallel growth of Internet in terms of total bytes transferred among hosts, coupled with dominance of HTTP protocol shows much can be leveraged through World Wide Web Caching technology [1]. Web Caching is a mechanism that can not only enhance end users' experience by reducing latency time, server load and perceived lag but also at the same time save bandwidth for the Internet Service Providers (ISPs). In simple terms a Web Cache is a temporary storage place for data requested from the Internet. Data can be an HTML page, images or multimedia files. The first request for a particular data is fulfilled from the Internet, now the web cache stores copies of document passing through it. Subsequent requests for the same data can be satisfied from the cache, if certain conditions are met. Websites are composed of many WebPages and Web documents. These inturn are composed of many small parts like logos, images, tables, text and audio files. Each part is cached as a different object. And some of the parts may not be cached at all. For example- when we access a news website, if the logo object, some advertising bars and some static content can be cached, it will be easier to download just the dynamic news content [2]. A proxy server works by intercepting the connections between the sender and receiver.

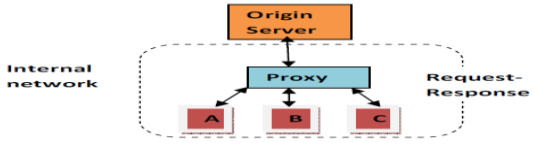

Fig- 1: A proxy Server

For client it acts as Origin Server and for Origin Server it acts as client. Proxies predominantly cache web pages. Each time an internal user requests a URL from outside , a temporary copy is stored locally. The next time any internal user requests for the same URL, the proxy can serve the local copy instead of retrieving the original across the network, thereby reducing latency and improving performance [3]. Proxy caching improves performance in many ways. Firstly Caching attempts to reduce web latency time required in obtaining web documents. Latency time can be reduced because the proxy cache is much closer to the end user than the original content provider or Origin Server. Secondly caching reduces the network traffic across the web server. Network load can be reduced because required document that are served from cache has to travel less on the network than when they are served by the Origin Server. Finally proxy caching can reduce the service demands on Origin Servers, since cache hits need not involve the Origin Server. It may also lower transit costs for access providers or ISPs. Furthermore, as it delivers cached objects from Proxy Servers; it reduces external latency and improves reliability as a user can obtain a cached copy even if the remote Server is unavailable. Due to the larger number of users connected to the same proxy server, object characteristics (popularity, spatial and temporal locality) can be better exploited increasing the cache hit ratio and improving Web performance [4]. With the exponential increase in interests towards dynamically generated content the need for more and faster Proxy Servers cache was created. As replacing these limited sized proxy caches every time was not a cost effective solution. So many proxy servers were grouped together to form a cluster. The clustering technology handled the issues of scalability, load balancing and fault

tolerance to an appreciable extent. For clustering also different architectures like hierarchical and distributed were proposed. Hierarchical Web caching cooperation was first proposed in the Harvest project [5]. Other examples of hierarchical caching are Adaptive Web caching, Access Driven caching; Push caching, Active Caching, Cooperative caching etc. A hierarchical architecture is more bandwidth efficient especially in case of low speed connectivity. However, there are several problems associated with a caching hierarchy as well. Every hierarchy level introduces additional delay . And higher level caches may become bottlenecks and have long queuing delays [6]. In distributed Web caching architecture [7], no intermediate caches are set up, and there are only peer proxy caches which serve each others' misses. Here in distributed caching systems most of the traffic flows through low network levels, which are less congested .Moreover it allows better load sharing and is more fault tolerant. However a large scale deployment of distributed caching has issues of high connection times, higher bandwidth usage and other administrative issues [7, 8]. A recent research work [8] shows that hierarchical caching has shorter connection times than distributed caching, It's also shown that distributed caching has shorter transmission times and higher bandwidth usage than hierarchical caching. So a well configured hybrid scheme can combine the advantages of both hierarchical and distributed caching reducing the connection time as well as the transmission time. So motivated by this work we proposed a hybrid web caching framework that clubs the strengths of both above mentioned architectures and minimizes their limitations. The remaining parts of this paper are organized as follows: literature review is presented in Section 2, in section 3 we have introduced the proposed strategy "Hybrid Web Caching Framework for Reduction of Web Latency", while Section 4 elucidates the results based on simulations performed on trace of data. Finally, Section 5 concludes the paper with future work.

2. RELATED WORK

Srinath et al. [9] discusses about the need of web caching and different web caching architectures like single level architecture, multilevel architecture, parallel and load balancing architecture .Though this paper explains about the existing techniques of web caching but does not throw light on a new architecture.

Sosa et al. [10] proposes a novel architecture based on distributed and cooperative web caching system. This paper just shows the conceptual proof of the adaptive cooperative web caching system. It requires further research and experiments to improve mapping function and scaling of the error tolerated during sequence matching.

Barish et al. [11] presents with several caching architectures, deployment options and specific design techniques. The paper failed to deal with issues like content security and practicality of handling dynamic and real time data.

Che et al. [12] aims to develop an analytical modelling technique to characterize an uncooperative two level hierarchical caching system where Least Recently Used (LRU) algorithm is run at each cache server. The same design principles are used to guide the design of a cooperative hierarchical architecture. And performance of later is found to be better. The performance of two is taken just on the basis of LRU which does not perform the best all the time.

Tang et al. [13] presents analytical framework for coordinated management of cascaded caches in hierarchical and enroute caching structure. It also proposes a novel caching scheme that incorporates both object placement and replacement strategies.

Tiwari et al. [14] devised an algorithm for Distributed Web Cache concepts with clusters of Proxy Servers based on Geographical Region.. Although the strategy provides load balancing of Proxy Server s dynamically to other less congested Proxy Servers. But metadata management becomes very difficult and each Proxy Server has to maintain the metadata of its neighbouring Proxy Server. To avoid cache coherence problem the metadata has to be updated periodically which leads to extra overhead and network traffic congestion.

Wijesundara et al. [15] proposes a feature in distributed web caching concept where each client acts as a cache server and share its contents with the neighbouring node. Here Hit rate and Miss rate are considered but no discussion on cache size and latency time is done.

Tiwari et al. [16] developed an algorithm for Distributed Web Cache which incorporates cooperation among proxy servers of one cluster. Though congestion and scalability problems are being dealt in this paper, but the proposed architecture fails to handle congestion and metadata management efficiently when number of Proxy Server in a particular cluster increases.

Tiwari et al. [17] discussed effective Distributed Web caching (DWC), Distributed Web Caching with Clustering (DWCC) and Robust Distributed Web Caching (RDWC). It designs a scheme to overcome frequent disconnections of Proxy Server. Clustering of Proxy Servers is used with dynamic allocation of requests to less congested servers to achieve load balancing and robustness. Though the proposed scheme showed improvement upon RDWC and other previous techniques in terms of Hit Ratio and Robustness but could not show significant improvements in scalability and metadata management.

3. PROPOSED WORK

In this section we propose our Hybrid Web Caching Framework for reducing Web Latency. The proposed framework is a combination of hierarchical and distributed Web caching taking advantages of both the architectures and enhancing the clustering concepts of peer proxy servers. Here the proxies are clustered together based upon geographical location. Each cluster, along with the peer proxy servers, consists of a Local Cache (LC). This LC is logically divided into two major segments. First one store the most recently and frequently accessed documents/pages/object and also pages from neighbouring clusters brought in demand. The second segment of Local Cache is used as Metadata Repository (MDR). MDR stores the information of the pages cached by each proxy servers in that particular cluster along with the corresponding proxy id. For reducing latency time and improving cache hit ratio a hybrid cache replacement policy is employed in the LC and sibling proxy caches when they are full. Page replacement algorithm proposed by Sirshendu Sekhar et al. [18] has come out to be the most efficient of all among the other trivial ones like LRU, LFU etc. Each cluster in the network has the same configuration.

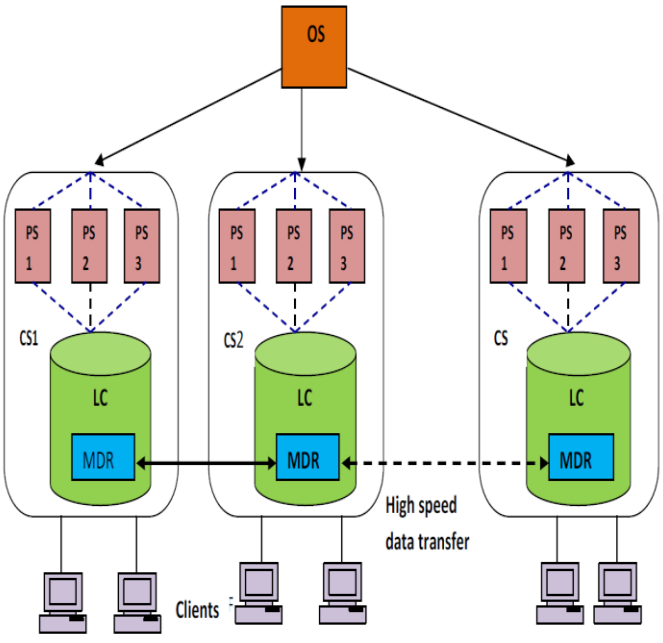

Fig-2: Hybrid Web Caching Framework

Our proposed model can be justified by the following Client Request Handling Algorithm (CRHA).

Client Request Handling Algorithm (CRHA)

```
1. begin
2. send Page Request to CS[i]
3. Call search_fn (CS[i]);
4. if (flag!=1)
5. Begin
6.      Call search_fn (CS [i-1]/CS [i+1])
7.       if (flag==0)
8.       Send Page Request to OS
9. End
10. End


search_fn (CS[x])
{
Search CS[x].LC;
if Page Hit
  {
     flag =1;
     Send Page Response;
    }
Else
 {
     Search CS[x].MDR
     if Link Hit for PS[m]
     {
       flag=1;
       Send Page Response from CS[x].PS[m];
      }                          //end of inner if
    }                            //end of else
}                               //end of search_fn()
```

Fig-3 Client Request Handling Algorithm

In this Hybrid Web caching Framework whenever a client sends a request, the request is forwarded to the near most cluster say cluster CS (i). The request of the client is handled in following way:

1. Whenever client issues a request first of all it comes to a near most cluster CS (i) and the corresponding local cache LC of CS (i) is searched. If there's a Hit, requested page is sent to the client.
2. In case of a miss in local cache LC of cluster CS (i), the requested page information is searched in MDR of that particular cluster. MDR (i) has information about all proxies' content in that particular cluster.

3. If there is a Hit, request is forwarded to respective Proxy server and requested page is sent to the client. Also a copy of the cached document is stored in Local cache (LC). This is called First Level Caching. It is the caching from proxy servers to Local Cache (LC).
4. Else in case of a miss MDR of CS(i) contacts MDR of cluster CS(i-1) and CS(i+1) .
5. If the requested page information is present in any of MDRs, then request is forwarded to the respective cluster's corresponding proxy server and requested page is sent back to client through the high speed data transfer connection between the clusters. Also a copy of the requested page is kept in LC of requesting cluster CS (i). This is First Level Caching.
6. In case of miss in neighboring clusters also, the request is forwarded to the Origin Server. And a copy is cached in any of proxy servers in that cluster. This is called Second Level Caching. It is the caching of pages from Origin Server to proxy server.
7. If the requested page is not present even in the Origin Server then a "Page Not Found" message is flashed back to the client.

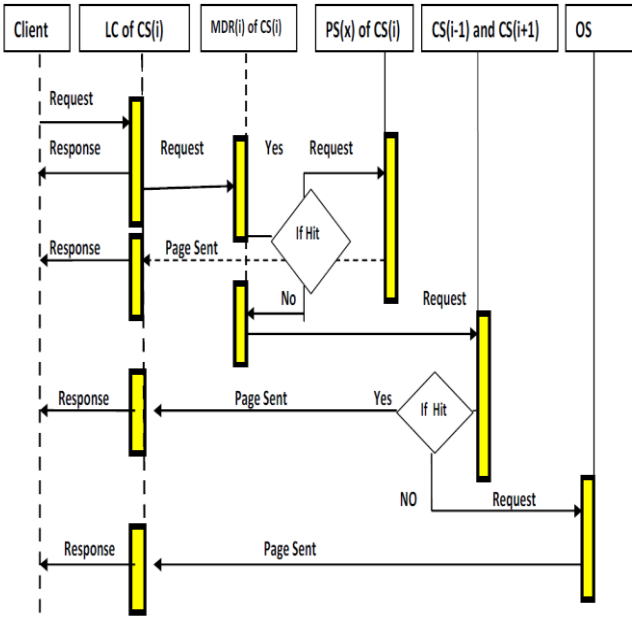

Fig- 4: Sequence diagram for handling clients' request

4. RESULTS AND DISCUSSION

The data set for testing our proposed architecture "Hybrid Web Caching Framework for Reduction of Web Latency" is obtained from proxy server of Birla Institute of Technology (BIT), Mesra, Ranchi, Jharkhand which is extremely popular among students, faculty members and staffs of as many twenty five departments along with various administrative sections, hostels and quarters. We constructed a trace-driven simulation to study our proposed model using the proxy workload of student/faculty traces from the university. In our experiment the proxy traces refer to the period from 12/Sept/2013:11:45:04 to 26/Sept/2013:00:00:02 of two weeks. The trace is composed of 11,388 nodes and 1,165,845 Web requests with average of 2,300 users per day. The simulations were performed at different network loads.

The performance of the proposed algorithm is evaluated in comparison with the existing distributed and hierarchical schemes. Figure 5 shows the impact of the proposed scheme on hit ratio with an increase in size of the cache. As shown below, with an increase in size of the cache the proposed hybrid web caching has highest hit ratio compared to other schemes. The reason behind is that in the proposed scheme information is cached and maintained at different levels. So if the requests from one site fail then it can be satisfied from the other sites simultaneously. Hence there is an increase in hit ratio of the proposed hybrid scheme compared to other schemes.

Figure 6 shows the impact of the proposed scheme on latency with varying the cache size. As shown in figure, with an increase in cache size, the latency decreases. But in the proposed scheme the latency reduces more than the other schemes. Because clients requests are not satisfied from one site then other clusters' proxy server is used for satisfaction of their requests and hence there is a reduction in latency in the proposed scheme with an increase in the size of the cache.

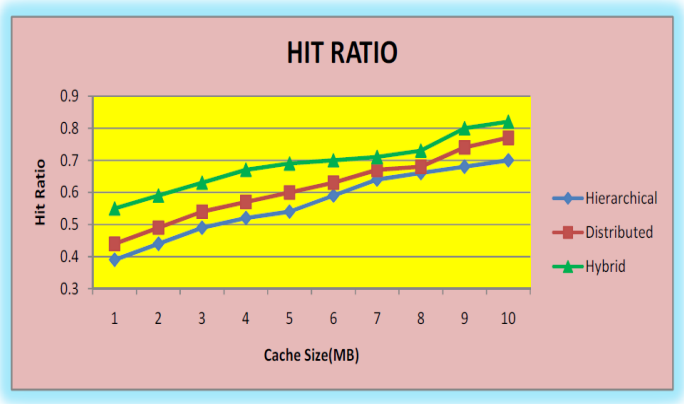

Fig-5: Comparison of Hit ratio in Hybrid , distributed and hierarchical caching strategy .

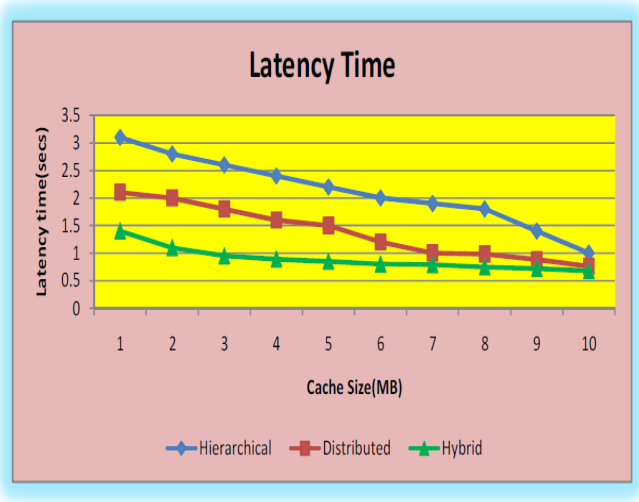

Fig-6: Latency time taken in Hybrid Web caching Framework, Distributed and Hierarchical caching approach

5. CONCLUSIONS AND FUTURE WORK

Hierarchical Web Caching achieves shorter connection times and also minimizes the bandwidth usage. However, the performance requirements for the higher level caches are very high. On the other hand, distributed Web caching achieves shorter transmission times as it distributes the network traffic away from the congested links. Above all, it gives good overall performance without expensive hardware, but the bandwidth usage is higher. Not for large scale, but it can be a good alternative for small, well interconnected areas. A hybrid caching scheme created with inter cluster communication is an optimal solution for reduction of web latency, improvement of cache hit ratio and efficient metadata management for big Organizations with clients in very high number.

Local Cache is logically divided into two parts where one stores most recently and frequently accessed pages and other part stores the metadata of documents in peer proxies of a cluster. This arrangement of cache in one hand reduces network congestion, latency time and network overhead. And on the other hand improves cache hit ratio, load balancing, bandwidth utilization, scalability and robustness. Here latency time is reduced from Hierarchical and Distributed caching strategy by 56% and 31% respectively and hit ratio is improved from the above mentioned strategies by 55% and 42% respectively.

The proposed strategy" Hybrid web caching framework for reduction of web latency" provides optimal solution to the issues of hierarchical and distributed web caching schemes. Herein we have clustered the proxy servers based on geographical location. However, in future we aim to cluster the proxy servers based on users' choice and behavior.

REFERENCES

[1] http://citeseerx.ist.psu.edu
[2] http://www.visolve.com/uploads/resources/ViSolve_Web_Caching.pdf
[3] Michael Piatek , J.Jackson, P.Juola ,"Distributed Web Proxy Caching in a Local Network Environment"ACM.src.acm.org/subpages/papers/piatek.src.2004.pdf
[4] http: //mae.engr.ucdavis.edu/d'souza/cdnVakali.pdf
[5] C. Grimm and J. S. Vockler , "DFN Cache Project", http://www-cache.dfn.de/.
[6] Chankhunthod et. al., "A hierarchical internet object cache", in Proc. 1996 annual conference on USENIX Annual Technical Conference, San Diego, CA, Jan. 1996.
[7] Povey and J. Harrison, "A distributed Internet cache" , in Proc. 20th Australian Computer Science Conf., Sydney, Australia, Feb. 1997
[8] Christian Spanner "Evaluation of Web Caching Strategies" Institut Fu¨ R Informatik ,Der Technischen Universita¨ Tmu¨ Nchen
[9] Harsha Srinath and Shiva Shankar Ramanna "Web Caching : A Technique to Speed up Access to Web Contents " Springer 2002.
[10] V.J. Sosa , G.Gonzalez, L.Navarro "Building a flexible Web Caching system" Computer Science, 2003. IEEE2003, Proceedings of the Fourth Mexican International Conference.
[11] Greg Barish ,K.Obraczke "World Wide Web caching: trends and techniques" IEEE2000 Communication Magazine.
[12] Hao Che, Zhijung Wang, and Ye Tung"Analysis and Design of Hierarchical Web Caching Systems " IEEE INFOCOM 2001
[13] Xueyan Tang & Samuel T. Chanson "Coordinated Management of Cascaded Caches for Efficient Content Distribution" Proceedings of 19th International Conference on Data Engineering (ICDE'03)
[14] Rajeev Tiwari, Gulista Khan, Lalit Garg, "Robust Distributed Web Caching", in International Journal of Engineering Science and Technology. ISSN : 0975-5462 Vol. 3 No. 2 Feb 2011
[15] M. N. Wijesundara, T.T. Tay "DISTRIBUTED WEB CACHING" IEEE 2002.
[16] R.Tiwari, K.Kumar, G.Khan,"Load Balancing in Distributed Web Caching : A Novel Clustering Approach", Proceedings of ICM2ST, International Conference on Methods and Models in science and technology.
[17] R.Tiwari, Neeraj Kumar, "Dynamic Web Caching : For robustness , Low Latency , Disconnection Handling", IEEE 2012.
[18] Sirshendu Sekhar Ghosh and Dr. Aruna Jain, "Hybrid Cache Replacement Policy for Proxy Servers" International Journal of Advanced Research in

Computer and Communication Engineering , e-ISSN: 2278-067X, p-ISSN: 2278-800X, Volume 6, Issue 11 (April 2013), PP. 15-22.

BIOGRAPHIES

Ranju Khemka is pursuing her M. Tech in Information Security from Birla Institute of Technology ,Mesra, Ranchi, India since 2012 and going to complete in June 2014. She has completed her MCA in 2010 and completed B.Sc. from University of Rajasthan, Jaipur, India. Her research interest includes Internet technologies, Web Engineering, and Algorithms.

Dr. Aruna Jain has received her Ph.D. from BIT Mesra, Ranchi, Jharkhand, India in the year 2009. She has done M.Tech in Computer Science and M.Sc. in Physics. She has published around 30 papers in reputed Journals and National and International Conferences . She has acted as resource person in various National and International Conferences and editorial board member in reputed Journals. Her fields of research are Computer Networks and Security, Data Mining, Soft Computing and Web Engineering. She has more than 20 years of teaching experience. Currently She is working as Associate Professor in Department of Computer Science and Engineering, BIT, Mesra, Ranchi, Jharkhand, India and guiding Ph.D. research scholars.