A Proxy Load Reduction Technique to Improve Web performance

Ayaz Ahmad Sofi[1]
Research Scholar
MMICT&BM (MCA)
Maharishi Markandeshwar University
Mullana, Haryana – India

Atul Garg[2]
Associate Professor
MMICT&BM (MCA)
Maharishi Markandeshwar University
Mullana, Haryana – India

Abstract: **The rapid growth of World Wide Web has caused serious performance degradation on the Internet. Web traffic has increased tremendously since the beginnings of the 1990s and overloading conditions of proxy servers occur repeatedly on Internet. Many people have started calling the WWW the World Wide Wait. In order to solve the problem of overloading on proxy servers, a new approach by extending cache memory close to the proxy server which stores a list of the most popular Websites, called the hot-site list, to reduce the unnecessary requests reaching to proxy server caches has been proposed in this work. These unnecessary client requests called the miss requests which cause the proxy caches to store copies of the requested information but the copies will not be accessed again. If the requests are not available in the hot- site list, then the forward proxy will forward the request to the original server. This is made possible only when the browsers are informed about the list of popular Websites. In this research, the extended cache memory on server side is used which may be in the form of hard disk to reduce load and response time on the proxy server which in turn improves the performance of Web. This extended cache memory placed at proxy server side. This research also focused on the working of the existing Web cache proxy server.**

Keywords—Peoxy server, Extended Cache Memory response time, Web traffic, Web performance.

I. INTRODUCTION

With the growth of Web, the user is able to look for and retrieve all kinds of information from the network without having any awareness about the network. The client requests according to their choice and doesn't matter if the requested document is a video clip, an image or the information on a computer in the next room or on the other side of the world. This leads to huge increase of traffic on the backbone of the network. As the use of the Web is increasing exponentially, it is been expected that the Web traffic will also grows exponentially with raising latency. In result the Web traffic increasing regularly and the client requests to the proxy servers are also increasing.

This in turn will increase the workload of the proxy server caches and results frequent overloading of proxy servers. Due to this overloading on proxies the performance is decreasing day by day. Liu et al., [9] states that proxy server performance is sensitive to traffic load and when it gets overloaded, its performance will degrade quickly. The authors in [10] stated that when a proxy gets overloaded, there are several connection time-outs which are reported as errors. A good solution is to install cooperating proxy servers for improving the overall caching performance. However, researchers still have to face one major problem of extra load and traffic. It has been proposed by Krishnan et al., [16] that the extra load could cause up to 300% overhead.

A proxy cache server receives clients request for a Web application and if it finds the requested information in its cache, it returns the same to the client without disturbing the upstream target server. If it is not present in the cache, the proxy fetches the object directly from the target server. Finally the target server, which has the original file, executes it and responds the result to the client through proxy server. The usage of proxy caching are supposed to reduce network traffic and reduce average latency. Proxy caches provide the clients to get quicker access to the document that are cached locally. Proxy caches are often placed very close to network gateways in order to reduce the bandwidth required over costly Internet connections. The proxy caches serve many clients with cached Web applications from many servers. They are also used to filter requests, for example, to prevent Web clients from accessing a specific set of Web sites. Proxy cache are typically used in saving bandwidth, improve response time and increase the availability of static Web based data and objects. In this research, an external cache memory on proxy side server is proposed to tackle the overloaded proxies. By, Meenakhsi and Atul the Web performance increases, if the server is near to the request sender [22]. Storing the popular Websites in the external cache which is placed near the proxy server and need not be forwarded to the origin server, the workload of the proxy cache can be reduced significantly without affecting the performance of the proxy server. In its result, it minimizes the traffic at the original server which in turn reduces the response time. Hence, it improves the performance of Web.

Proxy servers are commonly used to allow the Internet users to access the Internet within a firewall. It is used for security reasons because various companies run a special server called "proxy" on their firewall machines. These proxy servers usually process the client requests from within a firewall by sending them to the destination servers, catch the responses and returning the required data back to the clients. Since the identical proxy servers are commonly shared by all clients inside of the firewall, this leads to the effectiveness of using these proxies to cache Web applications. Clients belong to the same organization can likely share universal ideas. They would most likely access the common set of documents and each client tends to browse back and forth within a less period of time. Web caching at proxy server not only saving the network bandwidth but also reduce access latency for the Internet users.

There are two types of miss requests; good miss and bad miss. Good requests will cause the proxy to keep copies of the miss objects and the copies will be accessed again when requested by clients in future. Bad requests are those requests that too cause the proxy to keep copies of the miss objects but the copy will not be accessed again before it is being ejected from the cache. The corresponding stored copies of the bad-miss requests are called one-timer objects [17]. The caching of the good-miss requests reduces both network load and the response delay to the clients. However, bad-miss request cause new objects to remain in the cache for a certain period of time without being requested again. They could greatly waste system resources and degrade the cache performance.

Web cache proxy server could be the most well-known techniques for improving the performance of Web-based system by keeping Web objects that are likely to be used again when requested in the future in location closer to Internet user. It is placed between the Internet and the end-users that act to provide services between end-users and the Internet by reducing the number of requests sent across the Internet to the original servers [2]. Also, it must contain an up-to-date data or fresh data for the popular Web sites that frequently accessed by local clients. By making implementation in a proxy server, Internet bandwidth can be saved and the end-user response time can be reduced.

Before applying Web proxy caching approach the following problems can't be ignored:

- In traditional architectures every proxy server keeps records for data of all other proxy servers. This will lead in increasing the size of cache and if cache size becomes large then it will be a problem because as cache size is larger, it is difficult to manage Meta data. [13]
- Cache Consistency should be verified to avoid Cache Coherence problem. Cache Consistency means when a client send requests for data to proxy server that data should be updated regularly. [5]
- There must be a limit for number of connections to certain proxy server to avoid the problem of overloading only one server than the other in case load balancer is used. [6]
- When all the proxy servers keep the records for data of all the other proxy servers, this will lead to extra overload in the system which already produces congestion on all proxy servers. This extra overload due to each proxy server in the system must check the validity of its data with respect to all other proxy server caches. [7].

The structure of the paper is organized as follows: Section II reviews the related work of the Web proxy caching, Section III explains the proposed work of proxy server caching using extended cache memory and also discussed working of existing proxy servers, in section IV the advantage of using extended cache memory at the proxy server is discussed. Section V presents the conclusion.

II. LITERATURE REVIEW

From the evolution of the Web, its performance is the main research area which attracts the researcher to do better. In this section, the research work done by few researchers is discussed. Authors of [8] argued that an adaptive, highly scalable and robust Web caching system is needed to effectively handle the exponential growth and extreme dynamic environment of the World Wide Web. The system must evolve towards a more scalable, adaptive efficient and self-configuring Web-caching system in order to effectively support the phenomenal growth in demand for Web content on the Internet.

According to Liu et al. [9] , there work states that proxy server performance is sensitive to traffic load, and when a proxy is overloaded, its performance will degrade quickly. In contrast, J. Almeida et al [10] show that when a proxy is overloaded, there are various connection time-outs which are reported as errors. A standard solution is to install cooperating proxy servers for improving the overall caching performance. However, researchers following this line have to face one major problem; extra load and traffic.

Tiwari et al. [13] 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 neighboring Proxy Server. To avoid cache coherence problem the metadata has to be updated periodically which leads to extra overhead and network traffic congestion. . In [21] the author has refined the scheme of [13] to handle more delays and frequent disconnections of proxy servers. This can result in fastest response to the clients and also provide load balancing. Even these schemes suffer from the scalability problem. If size of the cluster grows, size of metadata grows as well then metadata at every proxy servers can become unmanageable. This problem can be overcome by new proposed architecture.

Chankhunthod et al., in [12] proposed Hierarchical Web caching scheme in the Harvest Project, that shares the interest of a large number of clients and also several countries have implemented this scheme. In this architecture caches are placed at the different levels of hierarchy and client's caches are at the bottom level [11]. For every miss the request is redirected to the next upper level caches in the hierarchy. If the document is not present in any of the level, request is forwarded to the origin server and on reply path copy of document is maintained at each intermediate level proxy server. But this scheme incurred many problems such as redundancy of data at each level and longer queries delay.

Atul & Kapil [18] proposed Portable Extended Cache Approach (PECA) to store frequently used data at client-side in an extended cache memory to enhance the computational performance of Web service. The extended cache memory may be in the form of pen drive, compact disk (CD), Digital Versatile Disk (DVD) or any other secondary storage devices. The problem with PECA approach used at client-side caches is that it serves the cache Internet documents for a single client only from many servers.

Dharmendra Patel et. al., [14] introduced one prediction model which predicts sequences of Web pages in advance and stores all Web pages in cache memory of proxy server when user starts a session and as a result access latency to access Web pages can be reduced. This prediction model consists of several components to do correct prediction. The components of prediction models are Pre-processing, User Session Identification, Pattern Generation and Pre-fetching.

Ayaz Ahmad & Atul Garg [24] analyzed various techniques to improve performance of Web. Their work states that by storing popular documents close to the users, caching proxies can save network traffic and reduce Web latency. According to Atul Garg et al. [23], the authors states that the purpose and methods are changed by using the Web from the beginning to this era and they also analyze and compare various approaches of Web Service Life Cycles which are needed in this era.

P. Somrutai et al., [15] explained that Proxy servers have been used widely to reduce the network traffic by caching frequently requested Web pages by using Web caching. Proxy server acts as an intermediary between the Web server and the Web user requesting the Web page. The proxy servers try to serve as many requests at the proxy server level. Proxy servers first fetch the requested Web pages from the origin Web servers and store the Web pages in the proxy server's cache. If a user makes a request to a Web page already stored in the cache, the proxy server accesses the local copy of the Web page stored in the cache and serves it to the user who requested the Web page. The proxy server's cache has limited capacity in terms of size of Web pages that can be stored in the cache at any given time. Once the cache capacity is reached, the temporally stale Web pages in the cache are discarded and replaced by newly requested Web pages. The Web pages stored in the proxy server cache are managed by the cache replacement algorithms. This approach of caching is called as Web caching. Web caching has been used to reduce the network traffic by caching Web pages at the proxy server level.

III. PROPOSED WORK

Web caching is the caching of Web documents, such as HTML pages and images, to minimize bandwidth usage and traffic load on server. A Web cache stores copies of documents passing through it; subsequent requests may be satisfied from the cache if certain conditions are met. Web cache optimization is used to get fast retrieval of user query results.

i. Traditional Proxy Server

The proxy server performance is measured by the factors like response time, filtering, handling the http request and caching. The main problems in proxies are traffic overload and response time, when load increases in local intranet the proxy server waiting time becomes higher. The only performance constraint that lies within the software is the congestion in the network [19]. This is a very common problem, which can occur any time in the network. If the network is congested and more clients login to the network simultaneously, it will degrade the performance quickly.

Proxy cache: A proxy cache is installed near the Web users. In the presence of a proxy server, there is no direct connection between the client and the target server. Instead, the client contacts to the proxy server and sends requests for resources such as a Web document, Web page or a file that actually resides on a remote server. The proxy server takes this request by fetching the required resources from the remote server and forwarding the same to the client. The proxy cache intercepts and handles all the requests for Web applications from a Website. If the requests are not available in the cache, the proxy gets these requests from the Web server itself. The key benefits of proxy caching are to reduce network traffic and reduce average latency. The proxy server offers a cache for all users so that commonly accessed content can be retrieved across the Internet once and then shared with many clients, improving network usage.

Due to the increase of Websites and increase in the number of Internet users, the network traffic increases which in result causes huge overloading on proxy servers. This huge overloading might result in the loss of data packets and also diminish the speed of the Internet. This indicates that the target server gets overloaded most of the time. The diagram of basic working proxy server is shown in figure 3.1.

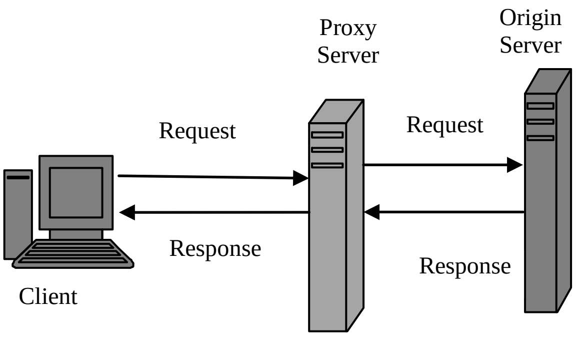

Figure 3.1: Basic Working of Web cache proxy server

As shown in the figure 3.1, whenever the client connects to a Web proxy server and makes a request for the resources that actually reside on a remote server, the proxy server forwards this request to the target server on behalf of the client, so as to get the requested information and deliver it back to the client. It plays an important role in copying a Web objects close to the clients and maintain the Web objects up-to-date. So the clients do not have to get the objects from the actual server.

A caching proxy can improve response times; reduce network traffic and the effective bandwidth available to end-users by returning local copies of objects at any time [3]. A cache replacement policy is implement in caching proxies including Least Recently Used (LRU), Least Frequently Used (LFU), and Hyper-G to determine when Web objects have to be removed from their archive [1][3]. Basically, *LRU* policy works to remove the documents that have not been accessed for the longest time whereas *LFU* policy acts to delete the documents that are least frequently asked by end-users [4]. At last, *Hyper-G* performs by removing the documents, which are LFU, and if it found there are two items that have identical LFU and then it will delete the one that is LRU. If the two documents that have identical LFU and LRU, Hyper-G will perform task to remove one that is larger [4].

Atul & Kapil [18] proposed Portable Extended Cache Approach (PECA) to store frequently used data at client-side in an extended cache memory to enhance the computational performance of Web service. Client side caches are built to cache Internet Web applications for a single user from many Web servers. The extended cache memory may be in the form of pen drive, compact disk, Digital Versatile Disk or any other secondary storage devices. The below figure 3.2 [18] shows the extended cache memory at the client side, which can be a pen drive, compact desk or any of the storage device that provides the path to the clients request. In this proposed technique the heavy data in the form of images, videos or audio resides in the client side and this information is fetched from the local machine in the particular Web application. In this technique, this heavy data or information did not move from server to client, which saves network load and improve the Web performance. Further this information is updated at the client side automatically whenever network load permits.

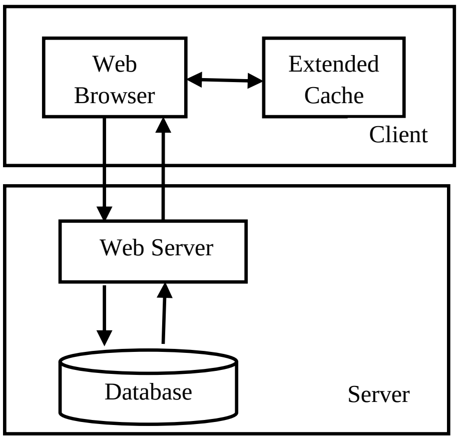

Figure 3.2: Extended cache memory at client side

ii. *Scenario of traditional Proxy Server*

There are three scenarios that involve in Web caching technology including cache-hit and cache-miss (see figure 3.3). When the request from the clients found the fresh copy of data on the cache and sends it directly to the clients, it is called Cache hit [2]. While, cache-miss happen when the request from the clients could not find the fresh copy of data that is requested on the cache. The next job is proxy server will get the copy from the origin server, save it on the cache for further usage and send it to the clients [2]. From user prospective, as a result of cache-hit, the response time is faster than cache-miss, because when the cache-miss happened the proxy server has to perform task to find the fresh data to origin server and send it to the user.

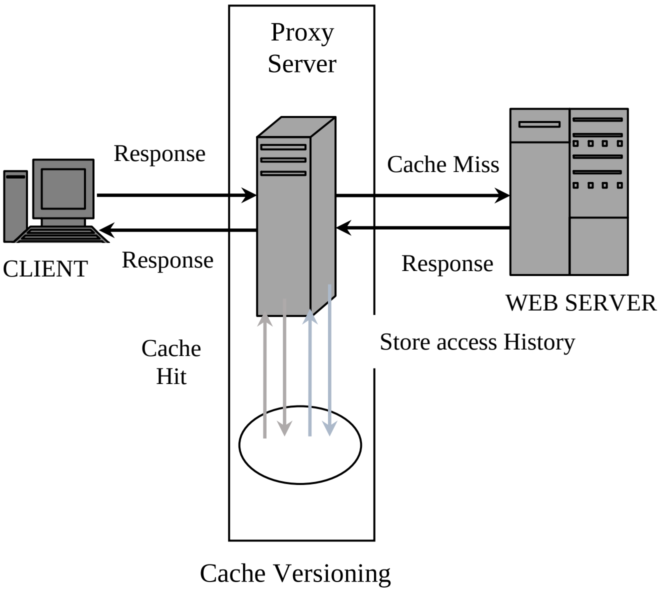

Figure 3.3: Scenario of Web cache proxy server [6]

iii. Extended Cache Memory at Proxy Server

The proposed new approach of proxy server cache is based upon two major factors which reduces the performance of the existing proxies, that is, traffic overload and response time. With the traditional method the cache memory at the proxy server is updated automatically when the new requests of the clients. The automatic change in the cache memory at the server side can cause a problem. For e.g. in a university there is a proxy server. In the university, the administrative staff and faculty have to access few common sites daily. So, these are in the cache of proxy and the proxy respond to the clients. In case, if other peoples say students, guests uses more sites other than common sites used by the staff, then the cache of proxy will be updated. Next time, if the staff will send a request then it may possible the proxy have to send that request to the main server for the response. This case may happen in the current era. To, overcome on this problem, the authors in [18] proposed a portable external cache memory at the client side and their results showed better results. But, the basic problem with their work is that their approach can be used individually.

In this proposed work an extended cache memory is placed at proxy server side which serves multiple clients. The high level view of the proposed approach of proxy server cache is shown in figure 3.4. Here the external cache memory placed at the proxy server which stores the most frequently accessed Websites by the local clients. When a proxy server receives a request from the client, it immediately checks the external cache memory to know whether the requested document in its cache is available or not. If the object is available, it will be immediately sent to the client. Otherwise, a proxy server forwards the request to the remote Web server and the requested document is in turn sent to the client when it is received. In this process, the traffic overload at the proxy cache and response time to the client can be reduced. The main objective of new approach is to place the frequently accessed Websites close to the Internet users so that they can be visited in less time. The frequently used websites are almost very large sites for example online shopping sites such as www.flipkart.com consisting of multimedia data such as large images, videos and other information which takes longer time to download for the client.

According to the author in [20], some statistics, for example, 30-50% of requests are satisfied for a fairly large site which handles 2000 requests from 60 clients each day and 25-35% of requests are satisfied even for a small site with a few hundred requests from 20 clients per day.

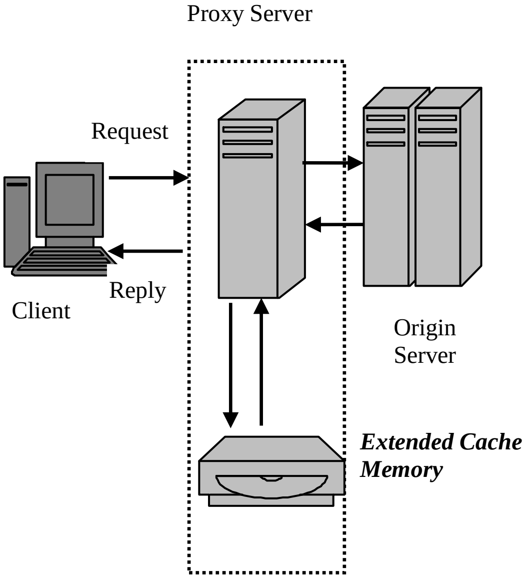

Figure 3.4: High level view of proposed Proxy Server.

iv. The Working Flow of proposed Proxy Server

The flowchart of the proposed proxy server is designed with help of the figure 3.4. The flowchart is described in the following steps.
Step 1: Web client sends a request for the Web application.
Step 2: The proxy server receives the request.
Step 3: If the request is valid, it performs the searching operation in the cache memory.
Step 4: If the request is not valid, an error message is sent to the client.
Step 5: Proxy server checks the frequently requested data in the extended cache.
Step 6: If requested data is available in it, then immediately responds to the client.
Step 7: If the requested data is not in the extended cache memory, proxy server forwards the request to the origin server and then sends to the client.
Step 8: The heavy data which is available in the extended cache is updated automatically without disturbing the client at the proxy server.

IV. ADVANTAGE OF THE PROPOSED WORK

The new proposed model of proxy server has the following advantages than the previous existing proxy servers.

- Handling HTTP requests: The proxy server handles multiple HTTP requests from the clients.
- Faster delivery of Web objects to the end user.
  The clients get the requested information in a less time without waiting for a longer time.
- Cashing is one of the few mechanisms that are preventing the Internet from overloading. By already storing frequently accessed sites in the extended cache memory, it significantly reduces the network traffic on target servers; this automatically reduces the

consumption of bandwidth. It also gives the appearance of a faster response time and save employees time and connectivity Expenses.

- The proposed approach reduces the workload on the origin Web server by caching requested data locally on the proxy servers over the Internet.

V. CONCLUSION

In this paper, a new approach is proposed for reducing the overload on the proxy server and minimizing the response time in the Web caching system. By using the extended cache memory such as hard disk at the proxy server cache having popular sites stored in it which are frequently visited by the clients can improve the performance of the Web. In its result, Web performance can be better as compared to the traditional proxy server. Therefore, traffic on the proxy server gets reduced and the clients can access the requests in a minimum response time and can achieve the higher cache hit rate at the proxy server.

REFERENCES:

1. Dutkiewicz E., 2005, “4th week lecture material of content servers and caching technologies”, School of Electrical, Computer and Telecommunication Engineering, University of Wollongong, p.4-40.
2. I Memic, 2001, “A primer of Web caching technology and benefits”. p,1-5 [online] Available: http://www.imimic.com/documents/WPBackgroundTechBenefits.pdf accessed:
3. Powell J.’1997, “Web Caching: questions and answers,” p.1-3, [online] available: http://scholar.lib.vt.edu/digilib/reports/dlcachetalk.pdf accessed:
4. Wooster R.P.,1996, “Optimizing response time, rather than hit rates of www proxy caches,” (Master of Science, Virginia Polytechnic Institute and State University), [online] Available: http://scholar.lib.vt.edu/thesis/available/etd.pdf accessed:
5. Ali, W., Shamsuddin, S.M., Ismail, A.S, 2011: “Web proxy cache content classification based on support vector machine”, Journal of Artificial Intelligence 4(1), 100–109.
6. Tiwari Rajeev andKhan Gulista, 2010, “Load Balancing in Distributed Web Caching : A Novel Clustering Approach” , Proc. of ICM2ST-10, International Conference on Methods and models in science and technology pp. 341-345, November 6, vol.1324,
7. Rajeev Tiwari, Gulista khan, 2010, “Load Balancing through distributed Web Caching with clusters”, Proceeding of the CSNA 2010 Springer, pp 46-54, Chennai, India.
8. S. Michael, K. Nguyen, A. Rosenstein, L. Zhang, S. Floyd and V. Jacobson, 1997, “Adaptive Web Caching: towards a new global caching architecture”, In Proceedings of the Symposium on Internet Technologies and Systems.
9. B. Liu, G. Abdulla, T. Johnson, and E.A. Fox, Nov. 1998, “Web Response Time and Proxy Caching,” WebNet98, Orlando.
10. J. Almeida, and P. Cao, Nov. 1998, “Measuring proxy performance with the Wisconsin proxy benchmark,” Computer Networks and ISDN Systems, vol. 30, issue 22-23, pp. 2179-2192.
11. Pablo Rodriguez, Christian Spanner, and Ernst W. Biersack, AUG 2001, “Analysis of Web Caching Architectures: Hierarchical and Distributed Caching”, IEEE/ACM Transactions On Networking, Vol. 9, NO. 4.
12. Chankhunthod et. al., Jan. 1996, “A hierarchical Internet object cache”, in Proc. 1996 annual conference on USENIX Annual Technical Conference, San Diego, CA.
13. Rajeev Tiwari, Lalit Garg, 2 Feb 2011, “Robust Distributed Web Caching Scheme: A Dynamic Clustering Approach”, in International Journal of Engineering Science and Technology in ISSN : 0975-5462 Vol. 3 No, pp 1069-1076.
14. Dharmendra Patel, Atul Patel and Kalpesh Parikh, July-December 2011, “Preprocessing Algorithm of Prediction Model for Web Caching and Perfecting”, International Journal of Information Technology and Knowledge Management, vol. 4, no. 2, pp. 343-345.
15. P. Somrutai, , 2011 “Improving the Performance of a Proxy Server using Web log mining,” M.S. thesis, San Jose State University.
16. P. Krishnan and B. Sugla, 1998, “Utility of co-operating Web proxy caches,” Computer Networks and ISDN Systems, vol. 30, issue 1-7, pp. 195203.
17. B. Guangwei and C. Williamson, Oct. 2002, “Workload characterization in Web caching hierarchies,” Proc. of 10th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems, MASCOTS 2002, 11-16, pp. 13-22.
18. Atul Garg and Anil Kapil, , 2010, “Potable Extended Cache Memory to Reduce Web Traffic”, International Journal of Engineering Science and Technology, Vol. 2(9), pp. 4744-4750.
19. Sieminski A, 2004,: “The Potentials of Client Oriented Prefetching”,, in Intelligent Technologies for Inconsistent Knowledge Processing, Advanced Knowledge International, Dikaiakos M.: “Intermediary Infrastructures for the WWW”, url: "citeseer.ist.psu.edu/dikaiakos02intermediary.html"
20. S. Glassman, May 1994, “A Caching Relay for the World Wide Web”, First International World Wide Web Conference.
21. Rajeev Tiwari, Neeraj Kumar, 2012, Dynamic Web Caching: for Robustness, Low Latency & Disconnection Handling, 2nd IEEE international conference on Parallel, Distributed and Grid Computing.
22. Meenakshi Gupta & Atul Garg, “Content Delivery Network Approach to Improve Web Performance: A Review”, International Journal of Advance Research in Computer Science and Management Studies Volume 2, Issue 12, December 2014 pg. 374-385.
23. Atul Garg & Jyoti Parashar, “A comparative study on Web Life Cycle Activities and Composite Web Sercices”, International Journal of Innovative Research in Computer and Communication Engineering Vol. 2, Issue 12, December 2014 pg. 7238-7247.
24. Ayaz Ahmad & Atul Garg,”Analysis of various techniques to improve Web performance”, International Journal of Advance Research in Computer Science and Management Studies Volume 3, Issue 3, March 2015 pg. 271-278.

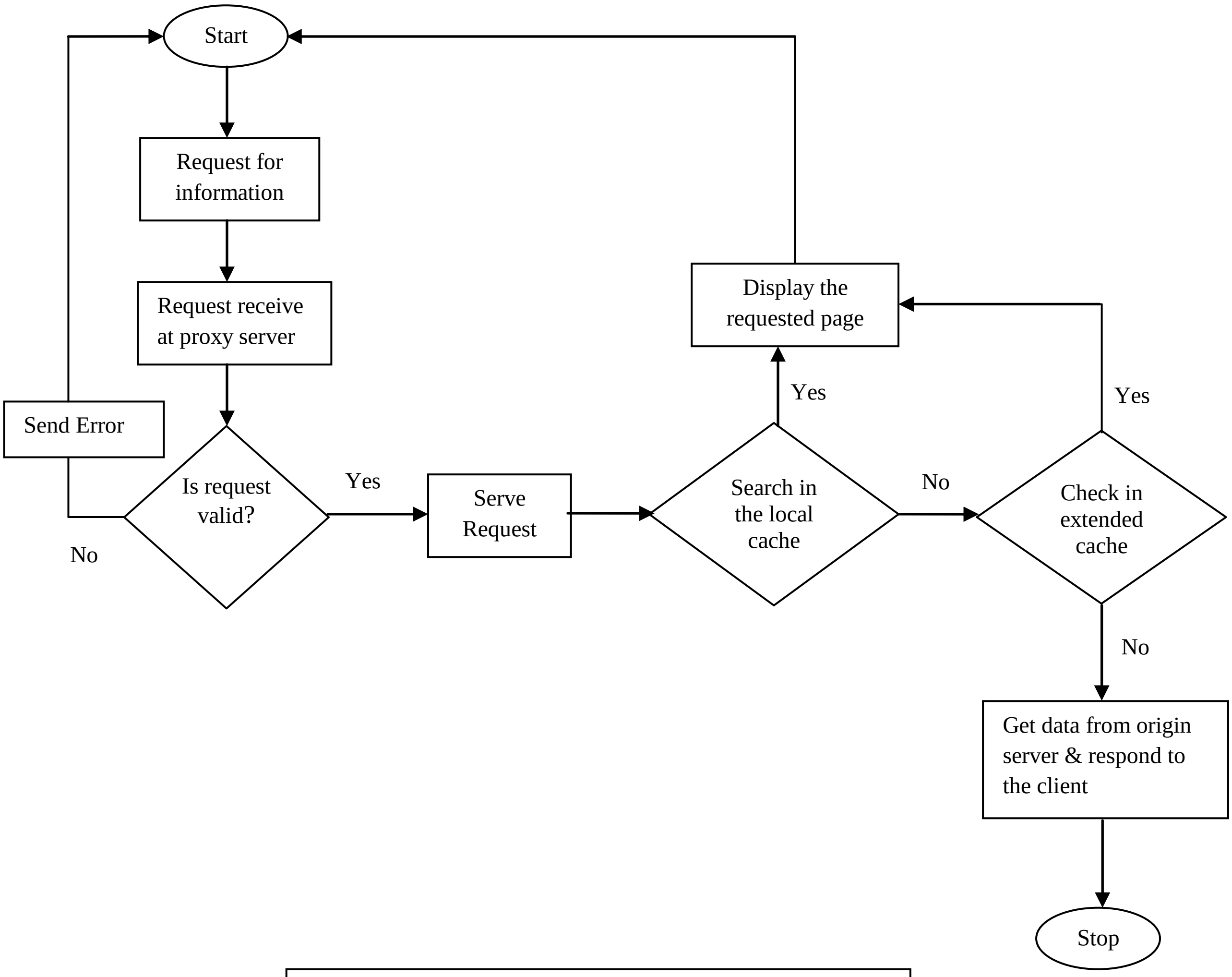
Start
Request for information
Request receive at proxy server
Send Error
Is request valid?
No
Yes
Serve Request
Search in the local cache
Yes
Display the requested page
No
Check in extended cache
Yes
No
Get data from origin server & respond to the client
Stop

Figure 3.5: Flowchart of the proposed Proxy Server