ISSN: 2321-7782 (Online)
Volume 3, Issue 3, March 2015
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Analysis of various techniques for improving 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 public Internet is a worldwide network of computers, in which millions of computers are interconnected throughout the world. Most of these computers are desktop PCs, UNIX workstations and servers that store and transmit information such as Web pages and E-mail messages. The World Wide Web is one of the most popular and important Internet applications and people in daily lives heavily dependent on it. Despite its importance, the current Web access is still limited for two reasons: (i) the Web has grown significantly as using social networking sites, streaming video, and file hosting Websites have become popular, forcing Internet Service Providers to provide more and more bandwidth to satisfy end-users, and (ii) the need for Web access has also grown, and many users in limited-bandwidth environments, such as people around world or mobile phone users, still suffer from poor Web access. This paper provides an overview of techniques for improving Web performance. For improving server performance, a number of Web servers can be used with efficient load balancing techniques. We examine content Delivery networks (CDN's) and the routing techniques that they use. While Web performance can be improved using caching, a key problem with caching is consistency. This paper presents different techniques for achieving varying forms of cache consistency. Content Delivery Network is an effective approach to minimize the network congestion and servers to improve the response to end-users.***

Keywords: WWW, Load balancing, Web caching, proxy caching, CDN, peer to peer

I. INTRODUCTION

The World Wide Web is one of the most popular and important Internet applications and people in daily lives heavily rely on it. Over the last few decades, the Internet has revolutionized modern society and economy. It has changed the way the people communicate with each other and the way business is conducted. The Internet has created a worldwide environment that is connecting people from all over the world. Due to exponential growth in World Wide Web leads to network congestion and increases the response time from end-users. In other words, end-users tend to wait a bit longer to be able to get information that they needed. These days Web requiring more and more bandwidth. This forces Internet Service Providers (ISP) to provide bandwidth to satisfy end-users. People read news, send/receive emails search, online shopping and watch video through the Web and in result it became very critical for people to have good Web access. By adding more bandwidth, more processing power and other mechanisms to improve the quality of service to the Internet infrastructure is one potential remedy for performance problems.

Web Caching is the widely used technique, used by Internet Service Providers (ISPs) all around the world, to save bandwidth and to improve Web performance. Caching is the most important and widely used performance improvement technique for Web-based systems [32]. Caching approach has been employed to improve the efficiency, reliability and Web traffic reduction over the Internet [12]. A nearby cache can serve a (cached) page quickly even if the originating server is swamped or the network is congested. The main issue with caching is to maintain consistency of the cache across the Web. This is the process by which cached copies are kept up-to-date with the originals. It is a less serious problem with static Web content

but for dynamic Web content, caching may deliver expired information. The attempt to maintain content consistency can be initiated by the server and the client.

This paper presents an overview of various techniques and components needed to support big volume traffic. These include number of servers at Web sites which can be scaled to accommodate high user request. Sandeep et al., [10] presented the performance analysis of static and dynamic load balancing algorithms. Load balancing algorithm is selected on the basis of situation in which work load is assigned i.e. at run time or compile time. Dynamic load balancing algorithms are proved to be less stable as compare to Static algorithms.

A number of Content Delivery Networks (CDNs) have developed to improve Web performance. Content Delivery Network (CDN) is an extension of cache optimization designed to supercharge Website performance aimed specifically around world to dispersed Web traffic. CDNs consist of a network of servers having cached copies of Web pages. Internet users requesting this information are directed to the nearest server within this network based on their geographic location. A Content Delivery Network is a shared network of servers or caches that deliver content to users on behalf of content providers. Content providers do not control the caches and the content is replicated as a function of user requests. This paper examines several issues related to CDN's including their techniques for routing requests.

II. Literature Review

Dimple & Atul [14] proposed an ant based framework to balance the load. In their research, the author proposed an active ant at both client side and the server side. The client ant is responsible for the user request whereas the server ant is responsible for replying the request. The authors improved the server performance in their research. Sandeep et al., [10] presented the performance analysis of static and dynamic load balancing algorithms. Comparison is done on the number of parameters such as overload rejection, fault tolerance, accuracy and stability etc. Load balancing algorithm is selected on the basis of situation in which work load is assigned i.e. at run time or compile time. Dynamic load balancing algorithms are proved to be less stable as compare to static algorithms.

In Proxy Caching, the end-users stores recently accessed Web contents in cache memory of proxy servers. The upcoming requests for these contents are satisfied through proxy server rather than sending these requests to origin server. This reduces network traffic, load on Web server and response time [16]. Atul & Kapil [12] proposed Portable Extended Cache Approach (PECA) to store frequently used data at user-end 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. Joseph et al., [31] examined that client-side caching is complementary to data scheduling in improving the performance of real time information dispatch systems. An effective caching mechanism retains data items that are most likely to be accessed by clients and reduces the number of requests submitted to the server over the wireless communication channel. This saves the narrow bandwidth and reduces the workload on the server; also it helps in reducing access latency by serving requests locally with data cached at the clients.

According to Hossam et al., [17], Web caching is a popular technique to improve the performance and scalability of the Web by increasing document availability and enabling download sharing. Using cache cooperation, a mechanism for sharing documents among caches can improve performance of the system. Further, it can improve performance by providing a shared cache to a large user population. According to Liu et al., [11], "An Overview of world wide Web Caching" proposed that an adaptive technique are used for reducing Web traffic and to access the Web fsites efficiently. The proposed algorithms at client side and server side are efficient to reduce the Web traffic in adaptive manner. Since it is a hybrid technique, latency is reduced to 20 – 60 % and cache hit ratio is increased 40 – 82 %. In [29] the author has refined their scheme of [30] to handle more delays and frequent disconnections of proxy servers. In its outcome fastest response to the clients is provided with load balancing. Even these schemes suffer from the scalability problem.

Many studies concentrate on the user-level behaviour such as the size and number of request/response messages and Web application-specific properties such as page complexity and document referencing [22]. Flow-level properties of Web traffic have also been studied in [23]. There are many studies of popular P2P systems in the literature, including Napster [24], Gnutella [24, 21], and eDonkey [20, 26]. These studies have focussed on different aspects of P2P systems such as query traffic [27], data traffic [21], flow characteristics [20, 26], peer behaviour [21]. Atul et al., [28] presented the comparison of various evolutionary search methods which are developed on Web servers for query optimization. Evolutionary algorithms are analyzed and compared on the basis of their behaviour, flexibility and transmission methods etc. In their work, they show that performance of PSO is better as compared to other algorithms but lacks behind in processing time.

III. Analysis Of Various Techniques To Improve Web Performance

Due to tremendous increase in Websites, it becomes very critical for clients to have good Web access. The volume of Web traffic is increasing continuously which results to increase in response time and increase latency. In order to improve the Web access, there are two different techniques used to improve the Web performance. First is to increase the hardware and second is to improve the software. Using first technique, each and every client and service provider has to improve the hardware at their own end. But, using second technique, client or server does not change their hardware. In this research the second technique is analyzed. Few of the techniques are discussed below:

a) Image Optimization:

Access time of Web page depends on the total size of content assets being downloaded from host servers to the request browser. High quality images are the largest contributors to Website page size, degrading page speed and users eagerly waiting to load the Web page. The image optimization practices to reducing the negative impact of images on Website speed are:

- ***Format Selection:*** Use JPGs when quality is a high priority and image modifications are not required before uploading it. The PNG format is used for the images with icons, logos, text etc. GIFs format are only used for small or simple images and avoid BMPs or TIFFs.
- ***Proper Sizing:*** Save valuable bytes of image payload and match the dimensions (width) with the template of Web page. By making images responsive with browsers resize capabilities by setting fixed width and auto-height instructions.
- ***Compression:*** Image compression should be a thoughtful tradeoff between image size and quality. A compression of 60-70 percent produces a good balance in JPGs Formats.
- ***Fewer Images*:** Keep the number of images to an absolute minimum.

b) Minimize HTTP Requests:

Most of this time is tied up in downloading all the components such as images, style sheets, Flash, etc in the page:. Minimizing the number of components in turn reduces the number of HTTP requests required. This practice makes pages to display faster.

- ***Combined files*** are a way to reduce the number of HTTP requests when combining all the scripts into a single script and all CSS into a single style sheet.
- ***Image maps*** combine multiple images into a single image. The total size is near about same, but minimizing the number of HTTP requests speeds up the page. If the images are contiguous in the page, Image maps only work.

c) Reduce DNS Lookups:

DNS lookups take a meaningful amount of time to look up the IP address for a host. The browser can do anything if and only if the lookup is complete. **Decreasing the number of unique hostnames** may **increase response times**

Various other techniques have been proposed by researchers in the past. Some of them are discussed below.

d) Load balancing:

Load balancing aims to optimize increase throughput, decrease response time, and reduce overload of any single resource. Using number of components with load balancing instead of a single component may increase reliability through redundancy. The load balancer distributes requests among the Web servers (see in Figure 1). A DNS server is one of the methods of load balancing requests to servers. DNS servers provide clients with the IP address of one of the site's content delivery. For example, a client sends a request to a Web server such as http://www.research.ibm.com/compsci/, "www.research.ibm.com" must be translated to an IP address and this translation is performed by DNS servers. A name associated with a Web site can map to various IP addresses, each attached with a different Web server. A round robin [4] technique is used to select these servers with DNS servers. In this technique, a single domain name is associated with multiple IP addresses, clients are expected to choose which server to connect.

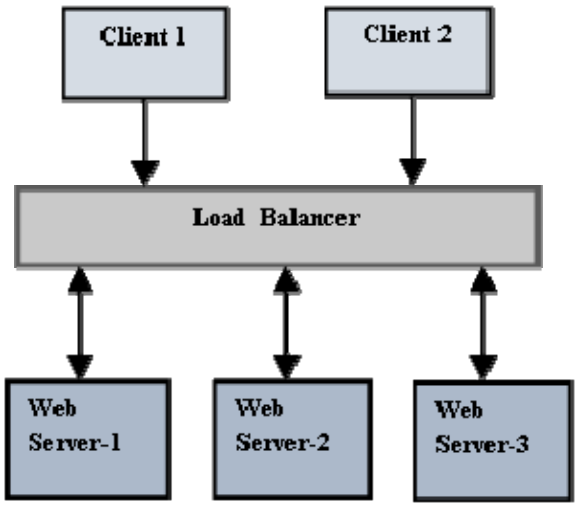

Figure-1: Load balancing

One of the problems with load balancing using DNS is that name-to-IP mappings resulting from a DNS lookup, may be cached anywhere along the path between a client and a server. This load imbalance can be caused because client requests can then bypass the DNS server entirely and go directly to a server [6]. The problem with this approach is that it can increase response times [7]. A request which comes from a client with a high request rate would receive a name-to-IP address mapping with a shorter lifetime than that assigned with a low request rate to a client.

Another method using in load balancing in front of several back-end servers is a connection router. Connection routers hide the IP addresses of the back- end servers. A DNS server can route requests to multiple connection routers. Connection routers also simplify the management of a Web site because back-end servers can be added and removed transparently. Two approaches of load balancing are static load balancing and dynamic load balancing. In *static load balancing*, the performance of the processors is determined at the beginning of the execution. According to [19], work load is delivery in start without considering the current load. This approach is used to reduce the overall execution time of a concurrent program while minimizing the communication delays.This approach requires less communication hence reduces the execution time [10]. In *dynamic load balancing*, at runtime the work load is delivery among the processors. The Web request must be spread across the Web server's node to balance the load, so as to reduce the response time and provide WWW user the best available quality of service.

e) Web caching:

A cache is a temporary storage area that keeps data available for fast and easily accessible. For example, the files which automatically request by looking at a Web page are stored on the hard disk in a cache subdirectory under your browser's directory. When return a page that have recently accessed and viewed the browser then get those files from the cache rather than from the original server, saving time and saving the network with the burden of additional traffic. Web Caching is the widely used technique, used by Internet Service Providers (ISPs) all around the world, to save bandwidth and to improve Web performance. The concept of caching has found its way into almost every aspect of computing and networking systems. Computer processors have both data and instruction caches. Caching is the most important and most widely used performance improvement technique for Web-based systems [32]. The main issue with caching is to maintain consistency of the cache across the Web. This is the process by which cached copies are kept up-to-date with the originals. There are two types of cache consistency, strong cache consistency and weak cache consistency. They have been proposed and investigated for caches on the World Wide Web. The performance improvement gained by introducing a cache is through satisfying requests directly from the cache instead of generating traffic to and from the server. For effective work, the footprint of a cache should cover a large population of users. By this, two or more users will request the same resource that can then be returned from the local cache at least once.

Strong consistency. It means that cached data is always validated before it is used. CPU and file system caches require strong consistency. However, some of the caches those in routers and DNS resolvers, are most effective even if they return stale information.

Weak consistency. Weak consistency means that the cache sometimes returns outdated information.

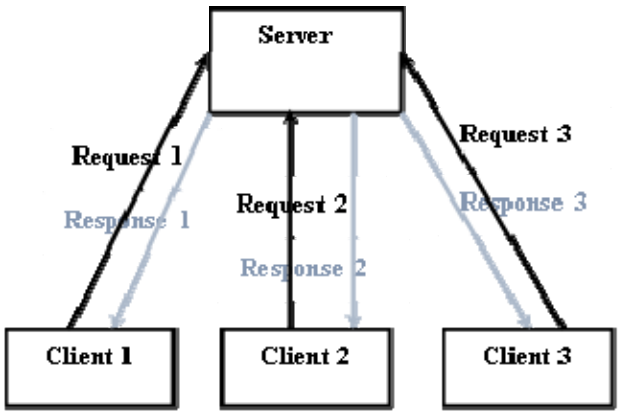

Figure 2(a): Client-Server Model

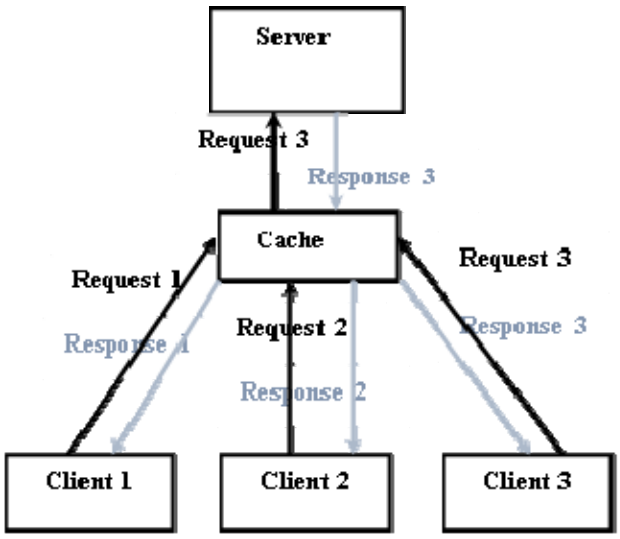

Figure 2(b): Client-Server Model with cache

Cache is between service providers and service users; it can benefit multiple service providers and service users, and improve the Web Services framework system performance much more. The application of cache includes client-side caching and server- side caching.

Client Cache: Client Cache can effectively improve the corresponding rate of the services and the consistence between data and server side data cache can be maintained by using asynchronous communication and multithreading technology, when the client mode is idle. It can reduce the remote interaction with services side. The performance of services is improved by more than 800% by using client side caching [3]. Atul & Anil [12], proposed to store frequently used data at user-end in an extended cache memory to increase the computational performance of Web service. However in cache memory, contents are placed after the users request them and the cache must be properly updated otherwise the users may get old contents. The basic problem of

using this technique is security. These days users usually access the Web to find new things, as a result cache hit ratio tends to be low. This hinders in improving the performance of Web content delivery to end-users.

Server cache: A Server-Side Web Proxy Caching (SSPC) is a server that acts as an intermediary for requests from clients seeking resources from other servers. The proxy server evaluates the request depending on its filtering guidelines. For example, it may pass traffic by IP address. If the filter shows valid request, the proxy by connecting to the relevant server provides the resource and requesting the service on behalf of the client. A proxy server may change the client's request or the response of server's, and sometimes it serves the request without contacting the origin server.

Proxy Caching: A proxy cache server receives HTTP requests from clients for a Web object and if the requested object is available in its cache, it returns the object to the user without causing any disturbance to the upstream network connection or destination server. If not available in the cache, the proxy tries to fetch the object directly from the origin server. Finally the origin server, which has the Web application get the request, then it process the application and respond to the client through the proxy server (see in figure 3). Next time if the client sends the same request again, then the proxy will provide the result. Proxy caching are very good to reduce network traffic and latency because popular objects are requested once, and served to a large number of clients.

Proxy caches are often located very close to network gateways to reduce the bandwidth required over expensive Internet connections. The proxies serve many clients with cached objects from many servers. Proxy servers are also used to filter requests, for example, to prevent users from accessing a specific set of Web sites. Proxy cache are typically used in saving wide-area bandwidth, improve response time and increase the availability of static Web based data and objects. Caching proxies are essential services for many organizations, including ISPs, corporations and schools.

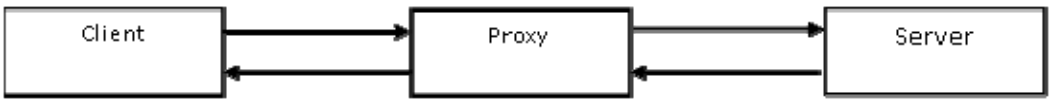

Figure 3: Proxy Cache

f) Content Delivery Network:

Content Delivery Networks (CDNs) are a popular and effective means of increasing the performance and reliability of high traffic Websites while reducing total cost of ownership. Content delivery network is an extension of cache optimization designed to supercharge Website performance aimed specifically at globally dispersed Web traffic. Internet users requesting this information are directed to the nearest server within this network based on their geographic location. Content providers do not control the caches and the content is replicated as a function of user requests. A recent study of CDN-served content found that 96% of the objects served were images [2]. However, the remaining little objects accounted for 40–60% of the bytes that are served, indicating a less number of very large objects.

i. Request Routing technique:

The request-routing approach has a direct impact on the performance of the CDN. This method improves the response time over accessing the origin server. The best request routing strategy is to direct the client to a CDN server that hosts the content being requested. However, if the request router does not know the content being requested, for example, if request-routing is done in the context of name resolution and then the request contains only a server name (e.g., www.greaterkashmir.com) as opposed to the full HTTP URL. The request-routing system operates as shown in Figure 4. Clients access content from the CDN servers by first contacting a request router (step 1). The request router makes a server selection decision and returns a server as segment to the client (step 2). Finally, the client retrieves content from the specific CDN server (step 3).CDN is normally used to serve static content such as images or multimedia objects. However, the use of CDN techniques to serve dynamic data is

increasing. Several research studies have recently tried to quantify the extent to which CDNs are able to improve response-time performance. An early study by John- son et al., [15] focused on the quality of the request-routing decision. The aim was to measure the response time to download a single object from the CDN server assigned by the request router and the time to download it from all other CDN servers that could be identify.

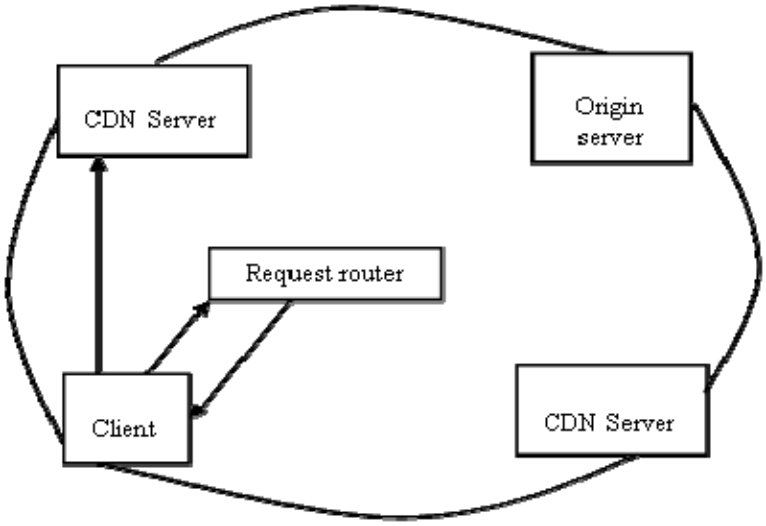

Figure 4: CDN Request Routing

ii. Surrogate Server:

Content delivery networks use surrogate server to replicate information at many different locations. Typically, clients are directed to the nearest surrogate that has a given resource. In this manner, it seems like all users are closer to the origin server. A content delivery network is a system of delivery surrogate servers (also called replica servers) to deliver Web contents to end-users on behalf of the origin server. The contents of the origin server are replicated on the surrogate servers. The requests from end-users are redirected to surrogate severs closer to them. As a result load on the origin server is reduced and network bandwidth expands. Content delivery architecture consists of a set of surrogate servers that deliver copies of content to the clients. The delivery system consists of mechanisms to move contents from the origin server to the surrogates. Some Web servers are slow because they generate pages dynamically and these slow Web servers can be accelerated by using Surrogates.

g) Peer-to-Peer application:

Recently, there have been a lot of excitement surrounding *peer-to-peer* applications, such as Napster. In these systems, clients share files and other resources (e.g., CPU cycles) directly with each other. In [24] Napster, which enables people to share MP3 files, does not store the files on its servers. Rather, it acts as a directory and returns pointers to files so that two clients can communicate directly. In the peer-to-peer realm, there are no centralized servers; every client is a server. P2P usage has grown steadily since its inception, and recent empirical studies indicate that Web and P2P together dominate today's Internet traffic [20, 21]. The peer-to-peer movement is relatively young but already very popular. It's likely that a significant percentage of Internet traffic today is due to Napster alone. Peers that connect to the system typically behave as servers as well as clients: a file that one peer downloads is often made available for upload to other peers. A recent study [13] has shown that most content-serving hosts are run by end-users, differ with less availability, and have relatively less capacity network connections (modem, cable modems, or DSL routers). *Peer-to-peer* applications have better utilization of bandwidth, storage capacity, other resources and user contributes resources to network.

IV. CONCLUSION

In this paper, various techniques to improve Website performance are discussed. As Web services become more and more popular, users will suffer from network congestion and server overloading. Web caching is recognized to be one of the effective

techniques to reduce latency and reduce network traffic. By storing popular documents closer to the users, caching proxies save network traffic and reduce Web latency. Cache response time is an important performance metric because for many users it represents the speed of the Internet. The main challenges in Web caching are proxy placement, dynamic data caching and cache routing. In future work a cache based technique will be proposed to improve the Web performance.

References

1. Iyengar, A., Nahum, E., Shaikh, A., and Tewari, R. (2002), "Enhancing Web Performance", In the 2002 IFIP World Computer Congress (Communication Systems: The State of the Art), Montreal, 25-30 August 2002.
2. B. Krishnamurthy, C. Wills, and Y. Zhang, "On the use and performance of Content Delivery Networks", In Proceedings of ACM SIGCOMM Internet MeasurementWorkshop, November 2001.
3. K. Devaram and D. Andresen, "SOAP optimization via parameterized client-side caching", In Proceedings of the IASTED International Conference on Parallel and Delivery Computing and Systems (PDCS 2003), pages 785-790, Marina Del Rey, CA, Nov. 2003.
4. T. Brisco, " DNS Support for Load Balancing", IETF RFC 1794, April 1995.
5. V. Cardellini, M. Colajanni, and P. Yu, "DNS Dispatching Algorithms with State Estimators for Scalable Web Server Clusters", World Wide Web, 2(2), July 1999.
6. D. Dias, W. Kish, R. Mukherjee, and R. Tewari, "A Scalable and Highly Available Web Server", In Proceedings of the 1996 IEEE Computer Conference (COMPCON), February 1996.
7. A. Shaikh, R. Tewari, and M. Agrawal, " On the Effectiveness of DNS-based Server Selection", In Proceedings of IEEE INFOCOM 2001, 2001.
8. G. Hunt, G. Goldszmidt, R. King, and R. Mukherjee, " Network Dispatcher: A Connection Router for Scalable Internet Services", In Proceedings of the 7th International World Wide Web Conference, April 1998.
9. E. M. Nahum, T. Barzilai, and D. Kandlur, "Performance issues in WWW servers", IEEE/ACM Transactions on Networking, 10(2):2–11, Feb 2002.
10. Sandeep Sharma, S.Singh and Meenakshi, "Performance analysis of load balancing algorithms", World academy of science, 2008.
11. M. Liu, F. Wang, D. Zeng and L.Yang, "An Overview of world wide Web Caching", *International conference on Systems Man and Cybernetics, IEEE*, 2001, pp.3045-3050.
12. Atul Garg and Anil Kapil, "Potable Extended Cache Memory to Reduce Web Traffic", International Journal of Engineering Science and Technology, Vol. 2(9), pp. 4744-4750, 2010
13. S. Saroiu, P. K. Gummadi, and S. D. Gribble, "A measurement study of peer-to-peer file sharing systems", In *Proc. of Multimedia Computing and Networking 2002*, Jan. 2002.
14. Dimple Juneja and Atul Garg, "Collective Intelligence based framework for load balancing of Web servers", IJICT, Vol 3 No-1 Jan-2012.
15. K. L. Johnson, J. F. Carr, M. S. Day and M. F. Kaashoek, "The measured performance of content Delivery networks", In International Web Caching and Content Delivery Workshop (WCW), Lisbon, Portugal, May 2000. http://www.terena.nl/conf/ wcw/Proceedings/S4/S4 1.pdf.
16. Radhika Malpani, Jacob Lorch, David Berger, "Making World Wide Web Caching Servers Cooperate", In Proceedings of the Fourth International World Wide Web Conference, pp. 107-117, 1995.
17. Hossam Hassanein, Zhengang Liang and Patrick Martin, "Performance Comparison of Alternative Web Caching Techniques", *Procceedings of the seventh International Symposium on Computers and Communications, IEEE*, 2002.
18. Stallings, W. (2005), "Business Data Communications", 5th ed. Upper Saddle River: Pearson Education.
19. Daniel Grousa, Anthony T., "Non-Cooperative load balancing in Delivery systems". Journal of Parallel and Distributing Computing, 2005.
20. L. Plissonneau, J. Costeux, and P. Brown, "Analysis of Peer-to-Peer Traffic on ADSL", In *PAM*, 2005.
21. S. Sen and J. Wang, "Analyzing Peer-to-Peer Traffic across Large Networks", *ToN*, 2004.
22. M. Arlitt and C. Williamson, "Internet Web Servers: Workload Characterization and Performance Implications", *ToN*, 1997.
23. M. Crovella and A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes", *ToN*, 1997.
24. S. Saroiu, P. Gummadi, and S. Gribble, "Measuring and analyzing the characteristics of Napster and Gnutella hosts", *Multi. Sys.*, 2003.
25. K. Tutschku, "A Measurement-Based Traffic Profile of the eDonkey File sharing Service", In *PAM*, 2004.
26. A. Klemm, C. Lindemann, M. K. Vernon, and O. P. Waldhorst, "Characterizing the Query Behavior in Peer-to-Peer File Sharing Systems', In *IMC*, 2004.
27. Atul Garg and Dimple Juneja,"A Comparison and analysis of various extended techniques of query optimization", IJICT Vol-3 no-3 July-2012.
28. Rajeev Tiwari, Neeraj Kumar, "Dynamic Web Caching: for Robustness, Low Latency & Disconnection Handling", 2nd IEEE international conference on Parallel, Delivery and Grid Computing, 2012.
29. Rajeev Tiwari, Lalit Garg, "Robust Delivery Web Caching Scheme: A Dynamic Clustering Approach", in International Journal of Engineering Science and Technology in ISSN : 0975-5462 Vol. 3 No. 2 Feb 2011,pp 1069-1076.
30. Jospeh Kee Yin Ng and Chui Ying Hui, " Client-Side caching strategies and On demand broadcast Algorithms for Real Time Information", IEEE , March 2008.
31. Killelea, P. (2002), "Web Performance Tuning", Sebastopol: O'Reilly & Associates.

