INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

Distributive Cooperative Caching Using LAN Network

S.Subramanian[1], M.Madan Mohan [2], G.Jagannathan[3]

[1]M.E CSE, Ranganathan Engineering College,
Coimbatore,
smanian30@gmail.com

[2]Assistant professor, Ranganathan Engineering College,
Coimbatore,
madhanaciet@gmail.com

[3]M.E CSE, Ranganathan Engineering College,
Coimbatore,
jaganjagan91@gmail.com

Abstract: *Network traffic increased day by day in internet. This project solves this problem by using the web caching and web prefetching. Web caching and prefetching generally used to improve the effective utilization of the internet. Web cache may cache the web document in their local proxy server. Web prefetching technique may perfected for late usage. To improve the cache access speed equal to the speed of CPU we can use multiple caches. Each level of cache need to updated regularly, for this process we can use page replacement policy like FIFO, LIFO and GDS. But this project use RGDS (Reversed Greedy Dual Size) instead of GDS. We can implement this project in multiple LAN's, so more than one client get benefits. This is done by set the individual proxy server for each LAN. Each LAN can communicate with another LAN through Inter Autonomous System routing protocol mechanism. Apart from the above concept we can also reduce the cost as well as time in the internet.*

Keywords: *Web caching, Web prefetching, FIFO, LIFO, GDS, RGDS, Inter AS routing.*

1. INTRODUCTION

The growth of network offers the benefits in terms of increased exchanged of information. In World Wide Web network traffic increased day by day. My objective is to reduce the network traffic by eliminating the network latency there by providing the quick response time between the end systems on network and hence the load on the web server is reduced with the help of specific algorithms, caching and prefetching with parallel processing technique. Web documents are cached at various network points. Then the cached documents are perfected this provide the way to reduce the traffic, cost as well as time.

The above problem can be solved using multiple LAN's. Each individual LAN with each proxy that could handle caching and prefetching of those data locally. The concept of Inter Autonomous System Routing can be used to share the web documents between the different LAN's which monitors all our web request and tune up our network data exchange both globally and locally. To replace the cache documents we can use RGDS technique.

1.1 Proxy server

A Proxy server is a server which services the requests of its clients by forwarding requests to other servers. A client connects to the proxy server, requesting some service, such as a file, web page, or other resmyce, available from a different server. The proxy server provides this by connecting to the specified server and requesting the service on behalf of the client. It may optionally alter the client's request or the server's response, and sometimes it may serve the request without contacting the specified server. In this case, it would 'cache' the first request to the remote server, and make everything as fast as possible. Caching proxies keep local copies of frequently requested resmyces, allowing to significantly reducing their upstream bandwidth usage and cost and increases performance. A proxy that focuses on World Wide Web (WWW) traffic is called a "web proxy".

1.2. Web caching and prefetching

Web caching and prefetching is a concept and technique generally used to improve the effective utilization of the internet. It caches the web document in their local proxy so that it can be perfected for later usage. This may cache either the web address or a web content that displayed on the client browser. It may also use any of the page replacement schemes for efficient caching and Prefetching.

According to the locations where objects are cached, Web caching technology can be classified into three categories.

- Client's Browser Caching
- Client-side Proxy Caching
- Server-side Proxy Caching

1.2.1 Client's Browser Caching

Web objects are cached in the client's local disk. If the user accesses the same object more than once in a short time, the browser can fetch the object directly from the local disk, eliminating the repeated network latency. However, users

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

are likely to access many sites, each for a short period of time which leads to increased hit ratio.

1.2.2 Client-side Proxy Caching

In client side proxy caching, objects are cached in the proxy near the clients to avoid repeated round-trip delays between the clients and the origin Web servers. To effectively utilize the limited capacity of the proxy cache, several cache replacement algorithms are proposed to maximize the delay savings obtained from cache hits.

1.2.3 Server-side Proxy Caching

Server-side Web caching, which distributes/routes the users' requests to the proper server-side proxies, is able to release the Web server's load and, thus, to shorten the user perceived response time.

1.3 Page Replacement Schemes

The size of the cache memory is limited which allows us to store only a limited amount of data so that we have to update it with only of those data that are to be accessed and useful for future usage. We consider each cache document as a page and it can be replaced by using the same strategies as we use for page replacement policies.

1.3.1 Greedy Dual Size (GDS)

GD-size stands for greedy Dual size this is just an extension of LRU-threshold along with the rule for eviction of document from cache. The eviction rule of this scheme works as follows. This takes the decision based on the threshold size of the cached document. It is very much effective when we want to decide when there are more than one document which has same LRU value. In those conditions GD-size evicts the document which has larger size. This GD-size method has an disadvantage of removing large sized cache document that could require a high response time to fetch from the web server. The available solution for the above problem on GDS is replaced by some new cache replacement schemes such as Reversed Greedy Dual Size. It evicts the document which has smaller size. Compare the both GD-size and RGD-size, RGD-size efficiently reduce the network traffic, cost as well as time.

1.4 Multilevel cache

Multilevel cache is using more than one level of cache implementation in order to make the speed of cache access almost equal to the speed of the CPU and to hold a large number of cache objects. Based on the content provided in the both the level of the cache it can be classified into two major categories.

- Multilevel Inclusion.
- Multilevel Exclusion.

Multilevel Inclusion is a technique which contains data that are common to both the levels. Normally level 1 cache size is made smaller when compare to that of the level 2 caches. This concept of multilevel caching already exists and used by athelon micro devices in their microprocessor chips. The main problem with this type of caching is that for same data we need two different levels of cache and also it is the Wastage of memory store. This will also results in removal of any cache results in updating of both the levels which is tedious job to perform.

Multilevel Exclusion is a technique where only the data which is not in the level 2 are present in the smaller cache of level 1.This has a greater advantage of having very less associativity between the two level caches. Such a concept was successfully used in all the recent Pentium processors. In our implementation idea we have used this Mutual Exclusion method of caching because of it smaller memory conception and easy cache Updating policy.

2. LITERATURE REVIEW

Improving proxy cache performance, John dilley, Martin Arlitt. The performances [1] of the existing three algorithms are referred. There are LFU, LFU+aging, Greedy Dual Size. Efficient Randomized web cache replacement schemes, Konstantinos Psounis, Balaji Prabhakar. This paper done an [2] implements another new algorithm for cache replacement called RR (Randomized Replacement).An efficient cache replacement strategy for the hybrid cache consistency approach, Aline Zeitunlian and A. Haraty. This paper done [3] an implementing the web cache and web prefetching. Objective optimal algorithm for long term web prefetching, Bin Wu and Ajay D. Kshemkalyani. In this concept [4] limited size cache memory need to be updated regular intervals. A Multilevel cache model for runtime optimization of remote visualization, Robert Sisneros, Jian Huang, Nagiza. This paper had implemented [5] the concept of multilevel caching for web caching. An efficient cache Replacement strategy for hybrid [6] cache consistency approaches, Aline Zeitunlian and Ramzi A. Haraty. This paper [7] had implemented the concept of update the cache to maintain the speed access speed of cache equal to the speed of CPU.

3. ALGORITHMS

There are two algorithms are used in this project.
1. Border Gateway Proxy Caching
2. Reversed Greedy Dual Size

3.1 Border gateway Proxy Caching

Border gateway proxy caching algorithm is used for prefetching and caching client's web object in proxy. It categorizes the client's web data into images, audio, video and text and then caches it.. It reads the user requested URL and responds [8] to referred URL based on the URL content .This reduces the response time for each request as it can directly prefetching the documents from the corresponding web caches. It also implements the cache sharing mechanisms through flooding and above explained Autonomous routing algorithm. It sets the replacement policy for evicting the documents replacement policy is a decision rule used for evicting the web pages .Replacement policy is required for replacing a page from web cache to make room for the new page. It uses utility function. The utility function assigns each page a value based on the recentness of use, frequency of usage, size of page and cost of fetching the each web documents from cache.

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

Routers within the same LAN are collectively known as Autonomous System. Each Autonomous system has a router called Border gateway router through which the communication [9] takes place. Inter Autonomous System routing protocol is used to communicate with the two different Autonomous systems. It sets the replacement policy for evicting the documents replacement policy is a decision rule used for evicting the web pages .Replacement policy is required for replacing a page from web cache to make room for the new page. It uses utility function. The utility function assigns each page a value based on the recentness of use, frequency of usage, size of page and cost of fetching the each web [10] documents from cache. Figure 1 shows the way data or cache shared by a LAN from neighbors LAN, if the requested web page not in the particular LAN.

```
If (new request) then
   flood it to neighbor gateway proxies
  if(not in neighbor)  then
   Connect to original web server.
   Display it on client's browser and cache it.
  else
   found so send to requested proxy.
End if
Else
   web object found in local cache,
   display in browser.
   End if
   if (time or tsize exceeds) then\
      evicts the document based on RGDS
   End if.
```

Figure 1: sample pseudo code.

3.2 Reversed Greedy Dual Size

Reversed Greedy Dual Size evicts the web document which has smaller size. Compare the both GD-size and RGD-size, RGD-size efficiently reduce the network traffic, cost as well as time.

4. IMPLEMENTATION

4.1 Experimental setup

Figure 2 shows the proxy server can act as a simple message replicator for individual clients. It catches all multicast messages sent by applications and replicates the messages to all [11] connected clients. Similarly, each time a client sends a message, the proxy server will replicate the message to all other connected clients and to the multicast group.

In my general architecture of implementation we use the concept of Border Gateway Proxy Caching, which share their cache [12] content among the various different LAN's using the Inter Autonomous System routing protocol. Each autonomous system is controlled by the single proxy designated for each LAN.

Whenever the cache is not available to prefetch for clients in one[13] Autonomous system then it immediately checks over the other Border gateway proxy systems of the neighboring Autonomous System by sending the request through flooding mechanism to all Border gateway proxies.

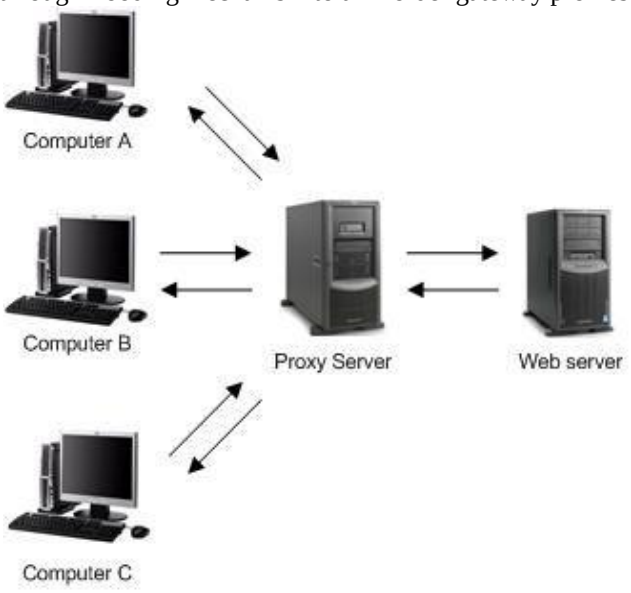

Figure 2: Experimental setup of Proxy operation

A figure 3 show, the proxy system is the border gateway system where we implemented my two proxy algorithms. They simultaneously perform their cache and prefetch operation for their respective autonomous system LAN.s. Whenever the cache is not available [14] to prefetch for clients in one Autonomous system then it immediately checks over the other Border gateway proxy systems of the neighboring Autonomous System by sending the request through flooding mechanism to all Border gateway proxies. The diagrammatical representations of this implementation are shown below.

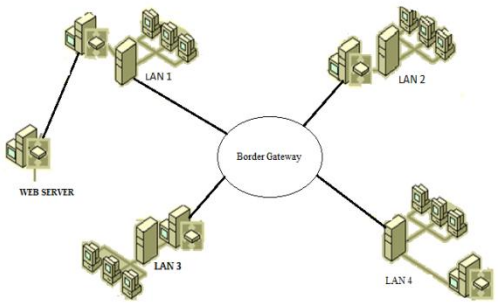

Figure 3: Experimental setup of Inter Autonomous system routing for proxy caching

Sending and receiving of the request/response can be done through by using the either TCP/UDP protocol for connection and FTP protocol for receiving cache object. the internal working of the proxy caching /prefetching involves the client request[15]validation, which checks whether the requested web object had already been fetched from the web server or not. If fetched and cached then in which border gateway system it was cached and making it to available to other autonomous system on request using the above discussed mechanism.

5. CONCLUSION

We have developed a proxy server, which runs with mentioned features, which inherently helps speed up browsing of web pages with use of randomized algorithms. As mentioned before the server is more reliable, more advantages than the existing one which uses the new concept of virtual proxy with randomized algorithms. Hence the server is successfully implemented with a few numbers of clients.

REFERENCES

[1] "Distributed Cooperative Caching in Social Wireless Networks" Mahmoud Taghizadeh, Kristopher Micinski, Charles Ofria, Eric Torng, and Subir Biswas, June 2013.

[2] "Integrating Web Caching and Web Prefetching in client side Proxies" IEEE transaction on parallel and distributed systems, May 2005.

[3] "Efficient Randomized Web-Cache Replacement Schemes Using Samples from Past Eviction Times" Konstantinos Psounis, Student Member, IEEE, and Balaji Prabhakar, Member, IEEE.

[4] "Improving proxy cache performance–Analyzing Three cache replacement policies" John Dielly, Martin Arlitt-HP Laboratories, Palo Alto, CA,USA. Hewlett-Packard company 1999.

[5] "Multilevel cache model for runtime optimization of remote Visualization" Robert sisneros, student member, IEEE, Chad Jones, Jian Huang, member, IEEE Byung-hoon park, and Nagiza F.samatora. IEEE Transaction on visualization and computer graphics, Vol.13, No.5, SEP 2002.

[6] "An efficient cache Replacement strategy for hybrid cache consistency approaches" Aline Zeitunlian and Ramzi A. Haraty. World Academy of Science, Engineering and Technology Vol: 39 2010-03-25.

[7] S. Podlipnig and L. Boszormenyi, "A Survey of Web Cache Replacement Strategies," ACM Computing Surveys, vol. 35, pp. 374- 398, 2003.

[8] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of Human Mobility on Opportunistic Forwarding Algorithms," IEEE Trans. Mobile Computing, vol. 6, no. 6, pp. 606-620, June 2007.

[9] "BU-Web-Client - Six Months of Web Client Traces," http://www.cs.bu.edu/techreports/1999-011-usertrace-98.gz, 2012.

[10] A. Wolman, M. Voelker, A. Karlin, and H. Levy, "On the Scale and Performance of Cooperative Web Caching," Proc. 17th ACM Symp. Operating Systems Principles, pp. 16-31, 1999.

[11] S. Dykes and K. Robbins, "A Viability Analysis of Cooperative Proxy Caching," Proc. IEEE INFOCOM, 2001.

[12] M. Korupolu and M. Dahlin, "Coordinated Placement and Replacement for Large-Scale Distributed Caches,"IEEE Trans Knowledge and Data Eng., vol. 14, no. 6, pp. 1317-1329, Nov. 2002.

[13] L. Yin and G. Cao, "Supporting Cooperative Caching in Ad Hoc Networks," IEEE Trans. Mobile Computing, vol. 5, no. 1, pp. 77-89, Jan. 2006.

[14] Y. Du, S. Gupta, and G. Varsamopoulos, "Improving On-Demand Data Access Efficiency in MANETs with Cooperative Caching," Ad Hoc Networks, vol. 7, pp. 579-598, May 2009.

[15] C. Chow, H. Lenog, and A. Chan, "GroCoca: Group-Based Peerto-peer cooperative caching in Mobile environment," IEEE J. Selected Areas in comm., vol.25, no.1, pp.179-191,Jan 2007.