Various Techniques used to Improve the Performance of Proxy Caching

Jitendra Singh Kushwah[#1]
Department of Computer Science & Engineering
ITM Group of Institutions
Gwalior, India
jsk2474@gmail.com

Dr. Sitendra Tamrakar[#2]
Department of Computer Science & Engineering
NMREC
Hyderabad, India
drsitendra@gmail.com

Abstract*—Proxy caching is utilized to increase access of user to general web content. Multiple system utilizes multiple cache for well performance. Cache of Multi-level usually works via testing the minimum lead cache. Uncertainty you miss a minor cache than the next vast level cache. This paper will study L1 cache such as the primary memory with L2 cache like the secondary form of memory of the proxy server. LRU takes into account the replacement of page algorithm. Previously, LRU technique used to eliminate the cold caching pollution. But it is not much efficiently work so we overcome this problem. In this paper, there we used three techniques such as LRU, AVL Tree and Binary Search Tree. The performance has been done over these techniques on the basis of access time. The result shows in the form of tables and graphs that shows, AVL tree is best among these techniques.

Keywords*—Proxy Server, LRU, AVL tree, Binary Search Tree, Access time.

I. INTRODUCTION

This period is an Internet period. As number of the users' increases, each user can not be efficiently satisfied. So to overcome the problem connected to the server and the client between the proxy servers. The critical need for each user is to have a quick response time. Caching in the proxy server for a quick response to the user. The cache will help you to keep your network bandwidth correct and more efficient for network traffic. Helps quickly access the cache page handles on the proxy server. The rapid form of growth of the multimedia streaming applications, the servers of multimedia type on Internet and wireless network have a huge role in traffic. To get better efficiency of the content delivery service in multimedia form , there is a growing tendency for multimedia content distribution techniques deployment like multimedia proxy and media replacement. The multimedia proxy is an efficient technique for solving network traffic and reducing latency through caching the well known content around the client. To date, there are certain tasks related to the multimedia proxy in the Internet. [1] The caching proxy is being a kind of Internet / n/w caching system which make capable to a proxy server to protect the most recent as well as frequent webpage requests and data that require more than one client machines. This is a way to accelerate the web page as well as website requests by protecting content & resources locally in proxy server. Caching proxy which can be called web proxy caching. The caching proxy is primarily improving website access time, enabling data download reduction and decreasing bandwidth usage. Caching proxy works when a proxy server is an example or is used to analyze or store certain proportions of data for frequently used websites and / or Internet-based resources. The proxy server will recover and deliver the data immediately to any web page that matches the data stored locally in a proxy cache or when performing a client request for a resource. The source stored on a local proxy server will be delivered very quickly and requests less bandwidth to be downloaded.

II. CLIENT-SIDE WEB CACHING

Cache initiate in web intermediates between browsers, user agent & source server. Usually, a cache is in browser as well as proxy. A browser cache client. If you check dialog by Preferences of few type of modern web browser (such as Internet Explorer or Mozilla), you will most likely monitor a "Cache" setting. Because many users visit the same web site, this is useful for a browser to download the latest pages set. Users can also interact with the web browser in presence of the cache of web browser, with special buttons, not just web pages but also by back, forward, refresh as well as URL rewinding. On the other hand, a cache of proxy is being located on a proxy. It facilitates in similar principle, but it's a great deal. Proxy of around serve of hundreds or thousands of users in similar way. Because size of cache is being limited, a cache restore policy is required to knob the cache content. If an object is stored, the cache will be filled, so it will decide which object to replace the space with the new material. The availability policy goal the finest use of cache space available is to get better form of hit rates as well as reducing the load on original server [10].

III. ARCHITECTURE OF LRU-BASED WEB CACHE

The LRU-base containment of web cache of a cache manager and cache core. Displayed in architecture image A. A. Cache Manager (CM) is being responsible for creating a web cache, so it is called a factory of cache. When web cache service starts, CM that will start with an example of CM web cache, such as the CM cache, the default storage path (build sure the client data be accessed on the web cache or not) and the CM value. Web is an

integral part of the web cache system, which consist of 3 models:

- Cache Listener Model

Cache listener is been used mainly so as to respond to requests after client that offer both REST full (reference state form of transfer) services POST & GET services. While getting a GET / POST request, LRU will format the data being received for mode of cache, also send stream of data to client after calculating LRU type cache data.

- LRU Cache Model

It's to calculate its component. In given model, LRU algorithm also enables the cache to upgrade cache capabilities, enhance the cache capacity and implement the in-cache objects so as to create space for the novel ones when an interface is available to the cache listener model. If the requested area from the cache.

- Cache Map

Responsible for just storing caching data (web materials). Cache data consist the content, full name, with the time of data storage. In paper, we have suggested as well as implemented a data structure for caching data retention. Since it's type of a map, we call it as cache-map. This can be explained in detail in next section [6].

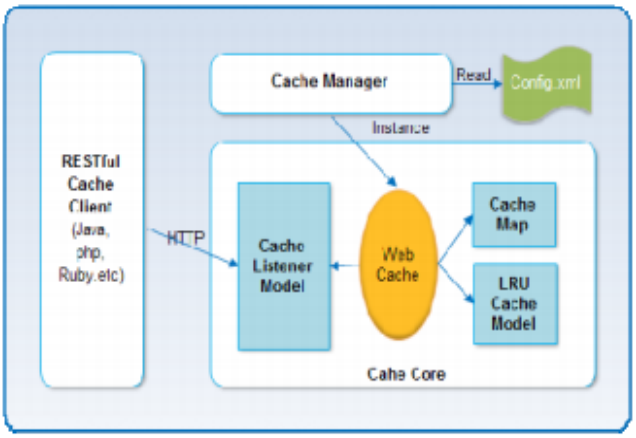

Fig 1.The Architecture of Web Cache System

IV. LRU-DISTANCE ALGORITHM

LRU-Far Algorithm is of a variant of the LRU, that attempts to avoid cold cache waste of proxy type caching. The LRU unit attempts to change cache behavior by disks of novel objects at the bottom of the stack, as shown in Figure 1. Only once accessible cache can be accessed only by giving the opportunity to be saved in cache. Distance D is based on the no: of objects at bottom of cache stack. For example, if D is 10, it will be 10^{th} item to be ejected from an object cache once available. However, if this material is being accessed again for second time, it migrates to most recent use into stack. Therefore, the LRU-away algorithm seeks to minimize the cool cache waste created by "one timers" by only removing once-only objects in a short time than in traditional LRU.

V. RELATED WORK

This paper analyzes the nature of the algorithm of various media. Instead of using alternatives (LRU), fewer uses (LFU) and first out (FIFO) are considered algorithms. The replacement algorithm of MFRR suggests for LNG, which works best for 16% better than the existing algorithms considered in this paper. This paper suggests a new replacement policy instead of the L2, which is called AF2LRU (the average frequency used, most recently). The simulation results show that MFMR and AF-LRU are about 28% more than the current pairing regent algorithm. [1].

In this paper, we provide wireless internet surveys for a variety of servers to provide cache transition for one server and server selection. Multiple projections of the video proxy are considered. We designed a unified price metric to measure video projection performance on wireless internet. The defined unified cost is based on the matrix. The new novel algorithm for a single server, and the new server selection of video servers, we intend to improve the performance of the latency and end load. Video quality as well as startup latency. Simulation o/p show that better performance than cache reboot algorithm and sever selection schemes. [2].

Gonzalez and et. al [3] three basic coaching algorithms developed specifically for GD Size, GDSF, and GD etc. LRU, LFU, LFUDA etc.

Tejaswi Agarwal et al. [5] Proxy servers are deployed more by organizations for operational benefits; however, major intervention on open source proxy servers still exists in decreasing client authentication in proxy mode. Technically, a block mode isn't designed for the authentication of client, but it should be implemented in some organizations. In given paper we have focused on WWW, with an authentication proposal for the transparent level proxy users making use of external scripts that display current transparency proxy authentication credentials and its defects and Internet protocol addresses. The most authentic Http source proxy server has been activated by this authentication system.

Rachid et al [7] A strategy called class-based LRU suggested. The C-LRU Rescue Base is based on a size-and-large basis, aimed at performing good and small materials in the cache of large and small documents. LRU based on the caching structural class is a change of the standard LRU.

Praveen Kalla et al. [8] A technology (LRU-SEQ) has been designed to minimize conversion energy in instructional cache sub-banks by redirecting the custom cache to the

final bank. By continuing to reinstate continuous entry, this policy will reduce the interbank transformation and the bank will likely increase the risk of shorts for a long time (thus lowering leakage energy). The project reduces the average energy to 23 percent.

[9] This paper suggests a novel kind of way to improve performance of proxy server by adjusting cache content in a certain way. This type of L1 cache is being used to retain its web pages and references of the web pages in L2 cache. This will reduce memory access time by average of proxy server.

In this study [11], we can evaluate web caching strategy performance , taking into consideration the successful rate earned for the realities of large consumer users. We focus on scores in a class of being recently used (SG-LRU) strategies. LRU Policy combines a simple update attempt to maintain the vast content of important type in cache based on the score function of predefined type . Several times-confirmed Zipf request patterns about access to be popular web form of platforms for the video streaming as well as further content types are evaluated by caching efficiency with simulations. In full coverage of basic parameters, we analyze the achievement of a specific web caching strategy of a typical independent request model (IRM) for a potential hit rate. In some cases, the results for specific caching tactics are confirmed by the fact that the results of the hit rate are 10% -20% higher than LRU. Furthermore, IRM Evaluations, over time, compare the results to the Dynamic Request Pattern with the use of Wikipedia statistics that has been recently added to over 1000 page requests. To demonstrate the influence of popularity of different objects in caching efficiency and to enhance the popularity and to activate score based caching techniques.

VI. PROPOSED METHODOLOGY

Previous methodology examined the LRU for the caching by reducing the pollution in cold cache. Instead of this, there are many drawbacks and these would be overcome by the AVL and Binary Search Tree. Firstly, there is a description regarding these techniques for better understanding.

AVL Tree:
AEL Tree GM a self-balancing binary search tree found. Adelson-Velsky, E.M. Landisin 1962. In honor of this discovery, this tree is named 'AVL'. An AVL In the tree, the height of two of the two sub-trees of a node may vary. Because of this behavior, AVL. The tree's height is also called a balanced tree. The ALL tree is going to improve a bigger performance, let's see how we can expand the process of adding a new key to the tree. We know that all the new keys add the leaves to the tree, and the rest of a new leaf element is zero, and there is no new need for the node added now. But once the new leaves are added, the parent balance should be updated. These new leaves influence how the parent balance component, and how the leaf node is the right child or a proper child. If node of new type is right child, the parent's balance sheet becomes less. If new node is being of the left child, the parents' balance sheet will increase. This relationship can be applied to the new grandparent's grandmother, perhaps to all the ancestors of the tree.
For the page replacement, there we used AVL Tree methods to eliminate the problem easily. Initially, generate two memories(i.e. primary & secondary) from the cache into equal parts. Distribution of the data performed into these memories according to their usage. In primary memory, the most useful data has been shifted using AVL Tree method for better and faster fetching of the data. In secondary memory, LRU has been used to store the least useful data. The proposed algorithm is as follows:

Proposed Algorithm:
Step 1: Start
Step 2: Divide cache into primary and secondary
Step 3: If (request for object)
Search object into primary memory
Call (AVL TREE)
Step 4: If (object found)
Then send them
Else
Search into secondary memory
Call (LRU)
Step 5: If (object found)
Move into primary memory
Then send them
Step 6: Repeat for every new request
Step 7: Exit

With AVL Tree:
1. Initially put new data
2. Each data should be placed at height with value 1
3. if new node added
    1. its parents height increased by value 1
    2. each node's height get updated
4. Calculate Balance Factor (BF) using formula:
   BF = Height of left subtree –Height of right subtree
5. If BF = -1/0/1
   Then tree is balanced
   Else
   Perform rotation
6. Exit

Now, if we want to compare AVL tree with simple binary search tree (BST) without balancing, then AVL will consume more memory (each node has to remember its balance factor) and each operation can be slower (because you need to maintain the balance factor and sometimes perform rotations). Since there is the added overhead of

checking and updating balance factors and rotating nodes, insertion and deletion in AVL trees can be pretty slow when compared to non-balanced BST's.WE can perform any operation in O(log(n)) only so the data retrieval rate is also fast as compared to binary search tree.AVL tree is also self-balancing tree.

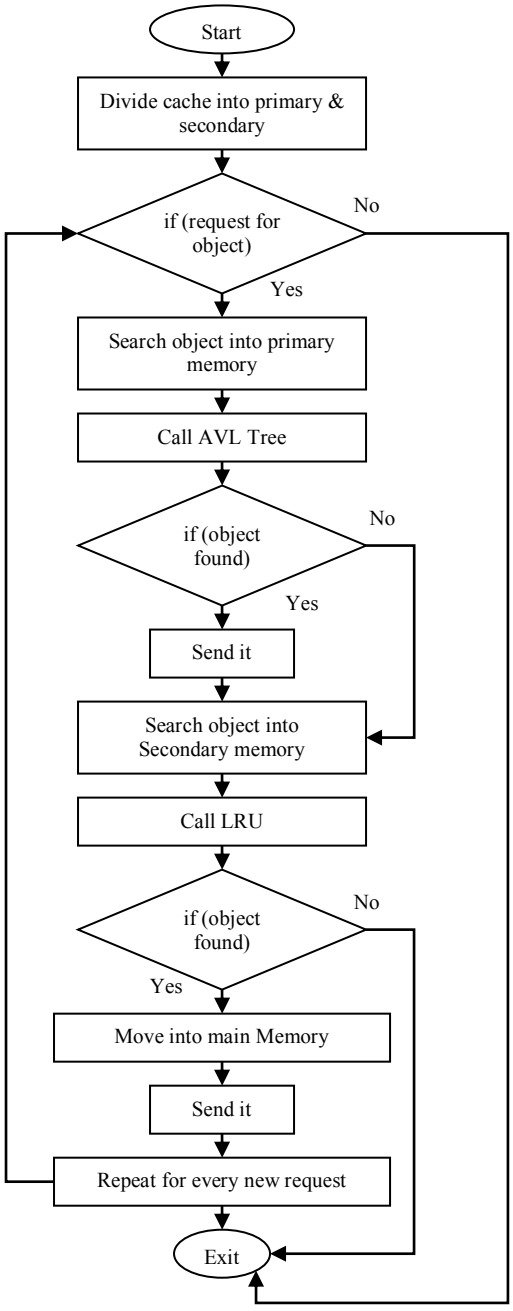

Fig. 2 Flowchart of Proposed Algorithm

VII. Analysis & Discussion

In the result analysis, table formed for considering various cache sizes. There are multiple attributes taken to illustrate the access time of the base and propose technique. Links contains URL of the websites and their access time is calculated for both the techniques. There 14 websites are taken to evaluate the access time and page number is used for consideration. Here, we consider two techniques such as LRU, AVL tree.

- **Cache Size=10**

Table 1: Access Time for Cache size = 10

S. No.	Page No.	Links	LRU Access Time	AVL Access Time
1	1	Google.com	2	1
2	2	Gmail.com	2	0
3	3	Facebook.com	2	0
4	4	Twitter.com	2	1
5	5	W3schools.com	2	1
6	6	Irctc.co.in	5	1
7	7	Youtube.com	5	0
8	8	Hotstar.com	4	1
9	2	Gmail.com	7	1
10	1	Google.com	6	1
11	3	Facebook.com	7	1
12	9	Instagram.com	6	0
13	10	Linkedin.com	8	0
14	4	Twitter.com	4	1
Total			62	9
Average			4.43	0.64

Base and Propose both Page Fault Counter=10

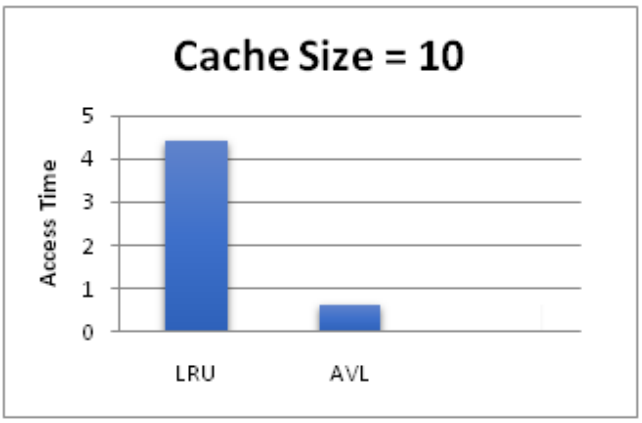

Fig 3. Comparison graph at cache size 10

- **Cache Size=15**

Table 2: Access Time for Cache size = 15

S.No.	Page No.	Links	LRU Access Time	AVL Access Time
1	1	Google.com	1	1
2	2	Gmail.com	1	0
3	3	Facebook.com	2	0
4	4	Twitter.com	4	0
5	5	W3schools.com	3	0
6	6	Irctc.co.in	4	0
7	7	Youtube.com	2	2
8	8	Hotstar.com	0	0
9	9	Instagram.com	1	3
10	10	Linkedin.com	2	3
11	11	Onlinesbi.com	21	0
12	12	Wikipedia.org	20	4
13	13	icicibank.com	2	1
14	14	Rediff.com	2	2
15	15	Paytm.com	25	0
16	3	Facebook.com	24	0
17	12	Wikipedia.org	28	0
18	2	Gmail.com	27	0
Total			169	16
Average			9.39	0.89

Base and Propose both Page Fault Counter=15

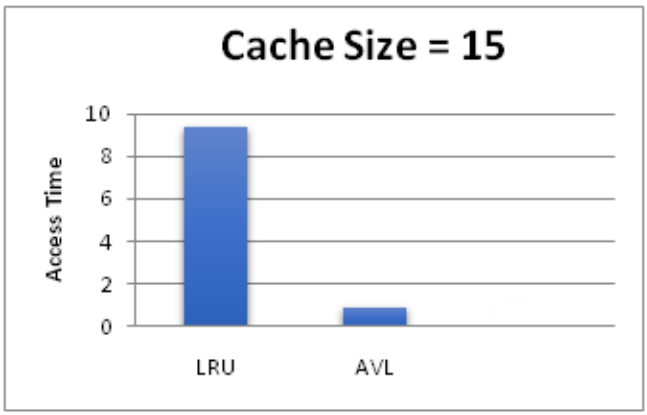

Fig 4. Comparison graph at cache size 15

- **Cache Size=20**

Table 3: Access Time for Cache size = 20

S.No.	Page No.	Links	LRU Access Time	AVL Access Time
1	1	Google.com	1	7
2	2	Gmail.com	1	0
3	3	Facebook.com	0	0
4	4	instagram.com	3	0
5	5	hotstar.com	1	0
6	6	Rediff.com	8	0
7	7	wikipedia.com	3	0
8	8	linkedin.com	8	0
9	9	icicibank.com	8	0
10	10	stackoverflow.com	9	0
11	11	W3schools.com	4	1
12	12	irctc.co.in	19	1
13	13	amazon.com	20	1
14	14	quora.com	11	1
15	15	whatsapp.com	10	0
16	16	naukri.com	25	0
17	17	Paytm.com	17	1
18	18	Wordpress.com	14	1
19	19	Yatra.com	1	0
20	20	Jeevansathi.com	9	0
21	13	Amazon.com	3	0
22	1	google.com	3	0
23	6	rediff.com	3	1
24	10	Stackoverflow.com	1	1
Total			182	15
Average			7.58	0.625

Base and Propose both Page Fault Counter=20

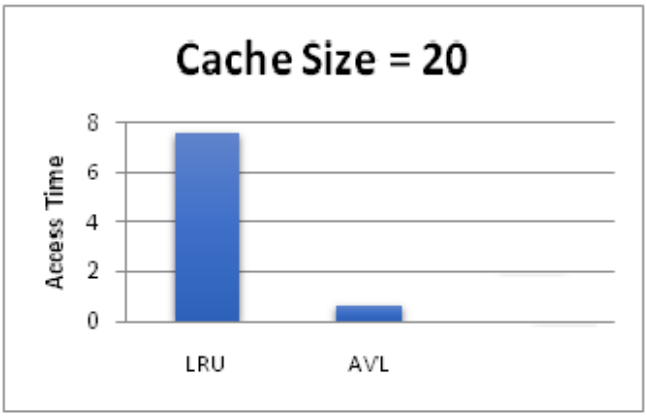

Fig 5. Comparison graph at cache size 20

- **Cache Size=30**

Table 4: Access Time for Cache size = 30

S.No.	Page No.	Links	LRU Access Time	AVL Access Time
1	1	Quikr.com	1	1
2	2	Ibibo.com	0	0
3	3	Answers.com	3	0
4	4	Clickindia.com	13	0
5	5	Airtel.in	1	0
6	6	Hindu.com	1	0
7	7	Guruji.com	1	0
8	8	Justdial.com	13	0
9	9	Shaadi.com	17	0
10	10	Twitter.com	20	0

11	11	Jeevansathi.com	23	0
12	12	Yatra.com	40	1
13	13	Wordpress.com	15	0
14	14	Paytm.com	47	0
15	15	naukri.com	16	0
16	16	whatsapp.com	32	0
17	17	quora.com	6	0
18	18	amazon.com	7	0
19	19	irctc.co.in	4	0
20	20	W3schools.com	3	0
21	21	stackoverflow.com	4	1
22	22	icicibank.com	4	0
23	23	linkedin.com	7	0
24	24	wikipedia.com	2	0
25	25	Rediff.com	3	1
26	26	hotstar.com	3	1
27	27	instagram.com	2	0
28	28	Facebook.com	3	1
29	29	Gmail.com	4	0
30	30	Google.com	5	0
31	12	Yatra.com	3	1
32	23	Linkedin.com	4	1
33	11	Jeevansathi.com	4	1
34	6	Hindu.com	4	1
35	29	Gmail.com	9	1
36	28	Facebook.com	4	1
Total			328	12
Average			9.11	0.33

Base and Propose both Page Fault Counter=30

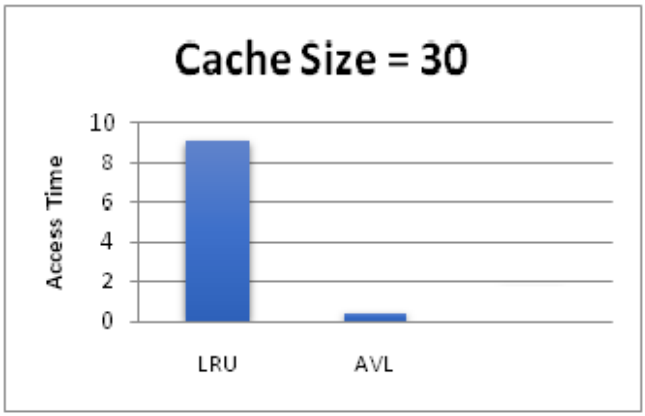

Fig 6. Comparison graph at cache size 30

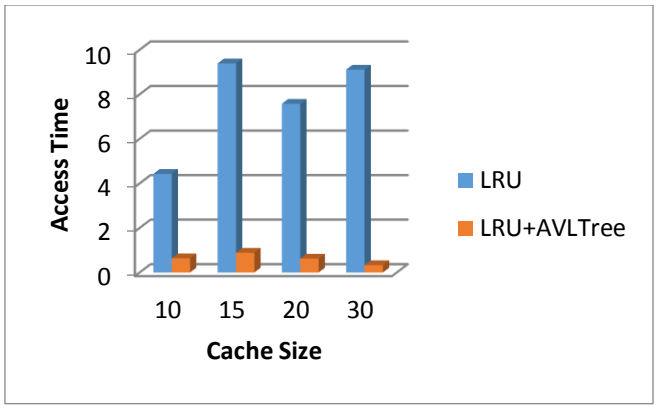

Fig 7: Average Access Time of LRU and LRU+AVLTree

Conclusion

The World Wide Web or the Internet is being global system of being computer n/w of being interconnected. WWW is a n/w of networks involving millions of the private form, public form, academic, as well as government n/w. Because of cache memory on the proxy server, the client may be satisfied with speed access speed. Large caches have better hit races. Several computers make use of different cache levels to address this temptation, small fast caches reserved for short-term caches. A proxy server that generates all requests as well as replies has been typically c/d a gateway / tanning proxy sometimes. A proxy server can be established in various points of local computer user's, user or targeted servers or Internet. From the above, we get that AVL generate better result in less access time.

References

[1] Gupta, R., &Tokekar, S. (2009). Pair of Replacement Algorithms MFMR and AF-LRU on L1 and L2 Cache for Proxy Server. 2009 Annual IEEE India Conference.doi:10.1109/indcon.2009.5409421.

[2] Zhe Xiang, Qian Zhang, &Wenwu Zhu. (n.d.). Cache replacement and server selection for video proxy across wireless Internet. Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS '03. doi:10.1109/iscas.2003.1206102.

[3] F.J. Gonzalez- Canete, E Casilari, Alicia Trivino - Cabrera, "Characterizing Document Types to Evaluate Web Cache Replacement Policies",International conference on Information Technology ITNG 2007.

[4] K.SureshBabu,SwethaMadireddy "Survey of Proxy Creates Continuous Location Based Spatial Queries for Mobile Clients by Exploiting Spatial and Temporal Locality" IJCSMC, Vol. 3, Issue. 9, September 2014, pg.141 – 147, ISSN 2320–088X.

[5] Tejaswi Agarwal, Mike Leonetti, "Design and Implementation of an IP based authentication mechanism for Open Source Proxy Servers in Interception Mode", Vol.4, No.1, Advanced Computing: An International Journal (ACIJ), January 2013.

[6] N. P. Jouppi, "Improving Direct mapping Cache Performance by the Addition of a Small Full Associative Cache and Prefetch Buffers," In Proceedings of the 17th International Symposium on Computer Architecture, Seattle, USA, 1990, pp.364-373.

[7] Boudewijn R. Haverkort, Rachid El Abdouni Khayari, Ramin Sadre: A Class-Based Least Recently Used Caching Algorithm for World-Wide Web Proxies. Computer Performance Evaluation / TOOLS 2003: 273-290.

[8] P. Kalla, X. S. Hu and J. Henkel, "LRU-SEQ: A Novel Replacement Policy for Transition Energy Reduction in Instruction

Caches", in Proceedings of the 2003 IEEE/ACM International Conference on Computer-aided design, pp. 518 – 522, November 2003.

[9] Niranjan, Y., Tiwari, S., & Gupta, R. (2013). Average memory access time reduction in multilevel cache of proxy server. 2013 3rd IEEE International Advance Computing Conference (IACC).doi:10.1109/iadcc.2013.650681.

[10] Krishnamurthy, B., Rexforrd, J.: Web Protocols and Practice: HTTP/1.1, Networking Protocols,Caching and Traffic Measurement. Addison-Wesley, Reading (2001).

[11] Gerhard Hasslinger KonstantinosNtougias Frank Hasslinger Oliver Hohlfeld, Performance Evaluation for New Web Caching StrategiesCombining LRU with Score Based Object Selection, 978-0-9883045-1-2/16 $31.00 © 2016 ITC.