

Web Cache Page Replacement by Using LRU and LFU Algorithms with Hit Ratio: A Case Unification

Kapil Arora¹, Dhawaleswar Rao Ch²

¹M.Tech, Department of Computer Science & Engineering,
Lovely Professional University, Phagwara, India

²Assistant Professor, Department of Computer Science & Engineering,
Lovely Professional University, Phagwara, India

ABSTRACT- In the information age, due rapid growth of the web and its protocols, a huge transfer of information take place over the Internet, which leads to enormous transmission traffic, which increases the load, latency time, network traffic, and bandwidth consumption. These parameters can be considered as problems arise due to the enormous traffic over the Internet. To overcome these kinds of traffic problems, we can adopt several measures and web caching is part of them. Web caching enhances the performance and offers an efficient access to the web content. Different algorithms such as least recently used (LRU), least frequently used (LFU), most recently used (MRU) etc., can be used. In this paper, we present web cache page replacement algorithm and comparison between LRU and LFU using the caching with respect to pages and the system we has proposed.

Keywords--Architecture, Performance Metrics, Page Replacement Policy, Least recently used, Least frequently Used.

I. INTRODUCTION

The World Wide Web (WWW) is increasingly being used by people all over the world and this lead to an enlarge in traffic in network. Consequently, this high traffic will increase the server load, latency time and reduce the bandwidth for competing request. To overcome this situation, Web caching technique has been used. Web cache reduces the high traffic over the internet so that user can access the web content faster. The main purpose of cache is to place the copy of object near to the client, so that web user can access the object easily, without the request going to the web server. There are different points at which we can set up a cache such as browser, proxy server and close to server. When a user request for the web page, firstly it is checked in cache, if the requested web page is available then it send back to the user. If the web page is not found in the cache, then the request redirect the web server and sends the response to the client. In between cache store the requested web page.

Because of the partial size of memory of cache, it becomes so much hard to save all objects in the memory. So, to resolve this problem, we have to use the page replacement algorithms. These algorithms are being used to evict the object from the cache and create a space for the new web

object to store. Algorithm such as Least Recently Used (LRU), Least frequently Used (LFU), Lowest Relative Value (LRV), Most recently used (MRU) are most popular. These algorithms assist in improving the performance of cache. A brief description on various web cache page replacement algorithms will be discussed in this paper.

II. ARCHITECTURE

To monitor request from client , a web cache is placed between server and client. Instead of sending the request to the server, it will serve the request from the cache. This will improve the latency and saved the bandwidth as well. It will cache the web document for the future used.

In order to reduce the numerous cost metrics such as average latency, byte hit ratio, hit ratio, several caching algorithm have been proposed. Cache page replacement algorithm is classified into categories.

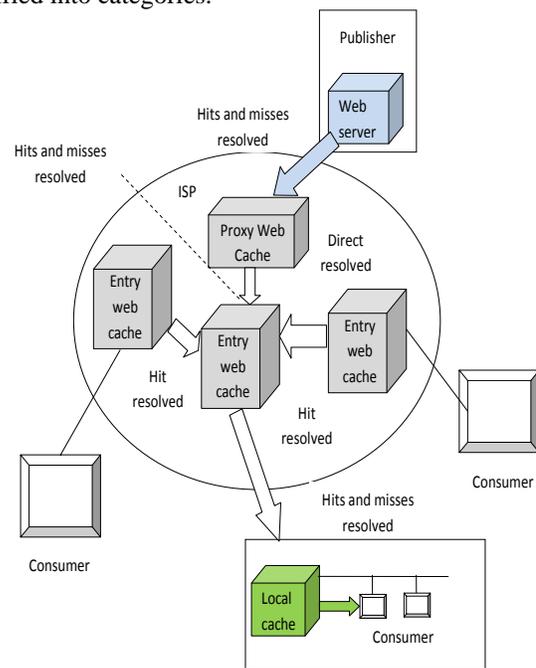


Fig1. Web Cache Architecture [2]

III. PERFORMANCE METRICS

To accomplish the objective, page replacement policy depends on the several key metrics. Based on this performance metric we can compare the performance of different algorithms. These performance metrics are plays a very important role in the web cache performance calculation. Such replacement policy aims to optimize performance metrics. To evaluate the performance of the cache, performance metrics are being used. The most commonly used are Hit rate, byte hit rate bandwidth saved, delay saving ratio are most commonly used.

A. Hit rate

The percentage of all requests object which are found in the cache instead of transferred from the requested server.

B. Byte hit

The percentage of all data that is transfer straight from the cache rather than from requested server.

Metric	Definition
Hit Ratio	$\frac{\sum_{i \in R} h_i}{\sum_{i \in R} f_i}$
Byte Hit Ratio	$\frac{\sum_{i \in R} S_i \cdot h_i}{\sum_{i \in R} S_i \cdot f_i}$
Saved Bandwidth	This is directly related to byte hit ratio.
Delay Saving Ratio	$\frac{\sum_{i \in R} d_i \cdot h_i}{\sum_{i \in R} d_i \cdot f_i}$
Average downloads time.	$\frac{\sum_{i \in R} d_i \cdot (1 - h_i f_i)}{ R }$
Notation: S_i =size of an object i. f_i =The total number of request for the object i h_i =a number of object hit for object i. d_i =The delay which are occur to retrieve the object from server. R = Set of object which are accessed. $ R $ =size of R	

Fig 2.Performance metrics [5]

C. Bandwidth Saved

Bandwidth saved is used to reduce the amount of the bandwidth consumed.

D. Delay Saving Ratio

The measure the latency of fetching an object. These performance metrics are the heart of web caching algorithm. These performance metrics are used to evaluate the performance of the replacement algorithms with respect to object which are present and requested in memory of cache, the saved bytes due to no retransmission, the decrease in latency to retrieve an object which is requested..

I. Performance metrics parameters

There are some parameters which we inured to evaluate the performance.

A. Throughput

A number of requests which is generate by the client per second

B. Hit ratio

Ratio of the total amount of object found in the cache to the total number of object is requested.

C. Byte hit ratio

Ration of the total amount of bytes which are served by the cache to total number of bytes which are sent to clients.

D. Cache age

Time after which the cache becomes full.

E. Downtime

Time taken to improve from the cache failures. [1].

II. Category of cache page replacement policy

Cache page replacement policies is classify in the following category:

A. Recency-based

This category of algorithm work on the time basis i.e. time to access the last references of object. The algorithm of this category is LRU (Least Recently Used) and has been functional in a number of proxy caching servers.

B. Size-based

The size based cache replacement policy consider the object size as the basic parameter. LFU-Size based algorithm is come under this category.

C. Frequency-based

The Frequency-based cache replacement policy is work on the frequency of the object means that the number of times an object is accessed. The algorithm of this category is LFU .

D. Function-based

A cost based function is used to determine the functional based algorithms. It involves the multiple parameters which are related to performance metric we used. Most indicative algorithm of this category is Greedy-Dual Size. [3].

VI. CACHE REPLACEMENT POLICY

Cache replacement policy plays a crucial part in the web caching. To achieve the high sophisticated cache mechanism, these replacement algorithms is necessary. These replacement policies helpful in evicted the object from the cache and build a new space for the incoming object. Due to limited size of the cache, a cache cannot store the entire requested object. So, we use the cache replacement policy to let the room for new document. This is applicable when the cache is full of object and then we have to insert the new object in to cache. So we have to evict object from the cache to make space. There are different cache replacement policies which are playing an important role in the web cache. These algorithms are:

A. Least Recently Used (LRU)

Least Recently Used page replacement policy is simple and easy to use. This algorithm is work on the time-stamp. It removes the last recently used object that was not used for a

longest time when the cache exceeds its maximum size and put a new object in place of evicted object. In case, if the cache size is not full then it will insert the object in the cache memory. For example, let we have “14” page, then the LRU work as:

7	0	1	2	0	3	0	4	2	3	0	3	2	1
		1	1	1	3	3	3	2	2	2	2	2	2
	0	0	0	0	0	0	0	0	3	3	3	3	3
7	7	7	2	2	2	2	4	4	4	0	0	0	1

Fig 3.Example of LRU page replacement policy

B. Least Frequently Used (LFU)

Least Frequently Used (LFU) page replacement policy which removes page which are least frequent used and then put a new requested page in free space. It is very simple and easy to use. In this, we have to maintain a counter which will count the frequency of the page. The page which is less frequent to use is replaced by the new incoming page. For example, we have “14” page and LFU work as:

7	0	1	2	0	3	0	4	2	3	0	3	2	1
		1	1	1	3	3	3	2	2	2	2	2	2
	0	0	0	0	0	0	0	0	0	0	0	0	0
7	7	7	2	2	2	2	4	4	3	3	3	3	1

Fig 4. Example of LFU page replacement policy

C. Most Recently Used (MRU) page replacement policy

The Most Recently used algorithm evicts the most recently used document from the cache. This algorithm is used where we have to access the historical information.[2].

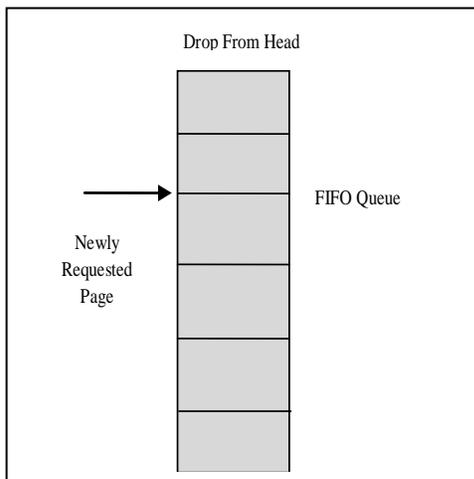


Fig. 5. Example of MRU page replacement policy [2]

D. Lowest Relative Value (LRV)

As the name suggest, lowest relative document value is first evict from the cache. To calculate an utility, to each object we assign an relative cost value. The object which is having a very low relative value is evicted first from the cache and a new entity will place in that space.

E. LRU-MIN

To minimize the replacement, this algorithm keeps a smaller size of object in the cache. If there is any object with Size S in the cache, then we follow the LRU algorithm to evict least recent used object from the cache. If there are no such an object which is having size S in the cache, then this algorithm evict the object of size S/2 in the Least recently used order.

LRU-Threshold

This is similar to LRU, but with a subtle difference is that, a object which is largest than a threshold size is not inert into the cache.

F. Greedy dual-Size (GD-SIZE)

It is a generalized version of LRU. This algorithm is concerned when we have an equal size of an object and the different cost is incurred to retrieve from the secondary storage [4].

VII. COMPARISON OF LRU AND LFU

As, we consider the page fault in the case of least recently used (LRU) and least frequently Used(LFU) in fig.3 and fig 4. There are 10 page fault in the Least recently used and there are 9 page fault in the Least Frequently used. So as we consider the performance, then we can conclude that LFU perform better than LRU due to the less page fault.

VIII. PROPOSED SYSTEM

In this, I have included a threshold value i.e. TSD to evict the historical object which are not been used by the long time. I have used both recency and frequency in this system. If any object which are placed in the memory and is not used since from a long interval of time, then it has to be remove that object to make a room for the new object.

Frequency division (F.D.) is used to calculate the average frequency. When a new object is entering, it checks the least time and check the priority with the frequency division. If the priority is greater than frequency division than we look for the second smallest timestamp. If the difference between both the time-stamp is greater than TSD, then we remove the first least document else if not, then we calculate the priority with the frequency division. If the priority is greater than the frequency division than remove the first document else second.

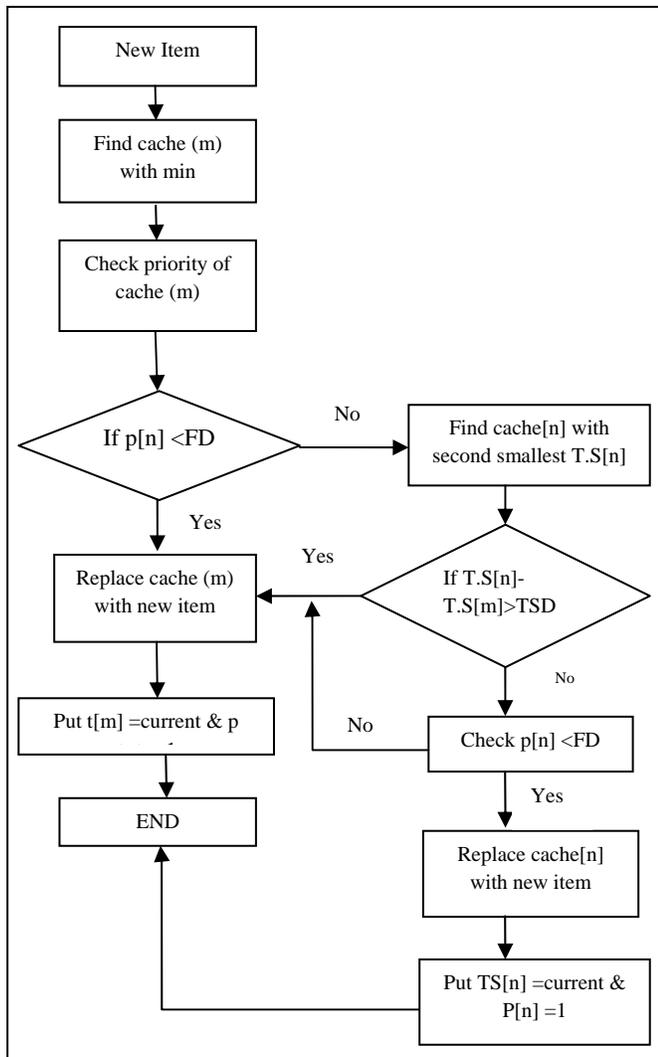


Fig.6. Proposed System

IX. CONCLUSION

Web cache is used to increase the performance of the system by reduce the server load in term of the requested object, latency time. In this paper, we reviewed the performance metrics to measure the performance of the cache and parameters which are meant for evaluate the performance of a Web cache. The page replacement algorithm categories such as Recency based, frequency based and functional based and page replacement policy such as LRU, LFU and the application related to the web cache has been discussed and we have concluded that LFU is perform better than the least recently used (LRU).Our system removed the drawback of LRU and LFU.

REFERENCES

- [1] S.V.Nagaraj, "Web Caching and Its Applications", Kluwer Academic Publishers, 2004.
- [2] Vinit A. Kakde, Prof. Sanjay K.Mishra , Prof. Amit Sinhal, Mahendra S. Mahalle"Survey of Effective Web Cache Algorithm", International Journal of Science and Engineering Investigations, Volume 1, Issue 1, February 2012.
- [3] George Pallis, Athena Vakali, Eythimis Sidiropoulo," FRES-CAR: An Adaptive Cache Replacement Policy", IEEE, 2005.
- [4] K.Geetha, Dr.N.Ammasai Gounden, Monikandan S.,"SEMALRU: An Implementation of modified web cache replacement algorithm", IEEE, 2009.
- [5] Abdullah balamash and Marwan krunz, "An Overview of Web Caching Replacement Algorithms", IEEE Communications Surveys & Tutorials, 2004, Vol.6 (2).
- [6]. Harshal N. Datir, Yogesh H. Gulhane, P.R. Deshmukh,"Analysis and Performance Evaluation of web caching algorithms", International Journal of Engineering Science and Technology, feb.2011.
- [7]. Naizheng Bian and Hao chen, (2008), "A Least Page Replacement Algorithm for web Cache Optimization", Workshop on Knowledge Discovery and Data Mining.
- [8]. Brad Whitehead, Chung-Horng Lung, Amogelang Tapela, Gopinath Sivarajah,(2008), "Experiments of Large File Caching and Comparisons of Cache Algorithms", IEEE international Symposium on Network Computing and Applications.