IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 19, Issue 4, Ver. VI (Jul.-Aug. 2017), PP 51-56
www.iosrjournals.org

Optimization of Cache Size with Cache Replacement Policy for effective System Performance

Muralidharan Murugesan[1], Dr.E.Kirubakaran[2]
[1]*Head, Department of Computer ScienceNehru Memorial College, PuthanampattiTiruchirappalli, India*
[2]*Additional General ManagerBharat Heavy Electricals LimitedTiruchirappalli, India*
Corresponding Author: MuralidharanMurugesan

Abstract: *The exponential growth of information in the internet over the last decade forces the researchers to think alternate solutions to provide better user service, reduce network bandwidth and reduced user perceived latency. Web Cache is the most promising technique for improving the web performance. Cache replacement is one of the important attributes to be addressed for better results. This paper proposes traditional cache replacement polices LRU, LFU and RR and examines its effectiveness in conjunction with different Cache Size. The results shows that LRU is most promising than other two methods in terms of HR , BHR and LSR and cache Size is definitely influence the effectiveness of the Cache Replacement Algorithms.*

Keywords: *web log mining, caching, hit ratio, byte hit ratio, Latency Saving Ratio, LRU, LFU, RR*

Date of Submission: 09-08-2017 Date of acceptance: 17-08-2017

I. Introduction

World Wide Web plays an important role as a knowledge hub in the world. Every day, hour and even in seconds billions of people are seeking some information in the web. As requests in the network are increasing exponentially the response time is delayed due to congestion. As of now approximately 3.6 billion users are using the internetwith twelve billion websites [1]. The number of web content is increasing day by day with alarming proportion. So it is necessary to improve the performance of web to provide quick response to the consumers.[2]One solution to the problem is to introduce web caching. Caching reduces network bandwidth usage, user perceived delays and loads on the original server [3]. Web cache is a temporary storage area holding web objects. One of the important issue in web caching is the choice of Cache Replacement Strategies. Cache Replacement refers to removing old web objects from the cache when the cache is full to pave way to new web objects. As a number of Cache Replacement Algorithms are available, a better algorithms has to be selected for improving the performance of the web.But there are some limitations in the algorithms like replacement precision, cache storage size and cost. So researchers are trying various techniques to enhance the performance. This paper proposed to analysis traditional approaches and their outcome. This paper also suggests further area of research in improving the performance of the web server.

II. Related Work

A detailed survey on various Cache Replacement Strategies and their classification were discussed in detail by Stefan Podlipnig et.al [3]. Recency, Frequency, Size and Function based strategies and their methods have been elaborated. Abdullah Balamashet. al [4] proposes a detailed comparative study of different algorithms and their impact of web traffic. The paper also discuss how cache replacement algorithms behave in the dynamic web traffic environment. A qualitative assessment is made about Efficiency of Cache Replacement Algorithms. Kin-Yeung et. al [5] discusses how various factors like cache size, limited processing and band width affects the outcome of the algorithms. A quantitative study of recency and frequency based strategies were made by Sam Romano et. al [6]. Waleed Ali et. al [7] proposed a detailed survey on various Web Cache and Web Prefetching techniques and their advantages with performance metrics for measuring efficiency of the techniques. Kapil Arora et. al discusses LRU and LFU algorithms and compare their performances. [8]. Gerhard Hasslinger et.al proposed a new web caching strategy by combining LRU with score based object selection. [9] Mohammed Salah AbdalazizKhaleel et. al proposes Average Least Frequency Used (ALFUR) Cache Replacement Technology for improving the web performance. [10]. Chronological study of aforesaid works shows that various factors involved in deciding the efficiency of the Cache Replacement Algorithms. LRU and LFU are the two major traditional algorithms used in the Web Cache Replacement Policies.

III. Replacement Strategies

Web Cache

Web Cache is one of the main strategy used to improve the performance of the web server by keeping web objects that are likely to be called in future to a nearby location. Web cache is implemented in three levels: client level, proxy level and server level. Since cache involves limited storage space it is essential to use appropriate cache replacement algorithms for achieving better response time. Managing cache will result in improving hit rates, reduce network trafficand alleviate loads in the original server and thereby improving the scalability of the web system

Classification of Replacement Policies:

The replacement policies are generally classified as follows:

Recency-based Policies: Recency is the primary factor for Cache replacement since recently accessed objects are likely to be accessed again in the near future. This policies are good for users who are interested in the same web object at the same time.Eg: LRU

Frequency-based Policies: A small set of web objects which are frequently called are the popular objects to be cached. This will be useful for users who access standard web objects.Eg.LFU

Size-based Policies: Removing larger sized web objects pave way for smaller objects. This will be useful for users who seek smaller information.Eg. SIZE

Function-based Policies: considering more parameters for a better hit ratio. This will require more memory and other resources.Eg. GD-Size

Randomized: simple and choosing random web objects for replacement. Useful for smaller memory and other resources.Eg. RR

Cache Replacement Algorithms: Cache replacement policies plays an important role in shaping Web Caching algorithms. A detailed survey was made on web caching and the following are the traditional cache replacement algorithms.

LRU	The least recently used objects are removed first
LFU	The least frequently used are removed first
SIZE	Big objects are removed first
GD – SIZE	With key value being assigned to each object and lowest key value is replaced
GDSF	Extension of GDS by integrating the frequency factor

The following factors influence the cache replacement process.

Recency: Time of Last Reference

Frequency: number of requests to an object

Size: Size of the Object

Cost: Cost of fetching the object from the original server

Access latency of the Object

Performance Metrics:

There are standard metrics available for measuring the efficiency and the performance of the web caching algorithms.

Hit Ratio (HR): Percentage of requests that can be satisfied by the cache.

Byte Hit Ratio: Number of bytes stratified by the cache as a fraction of total bytes requested by the user.

LSR: Ratio of the sum of download time of objects satisfied by the cache over the sum of all downloading time

Let N be the total number of requests and$\delta_i = 1$ if the request i is in the cache, while $\delta_i = 0$ otherwise.

$$HR = \frac{\sum_{i=1}^{N} \delta_i}{N}$$

$$BHR = \frac{\sum_{i=1}^{N} b_i \delta_i}{\sum_{i=1}^{N} b_i}$$

Where b_i is the size of the ith request

$$LSR = \frac{\sum_{i=1}^{N} t_i \delta_i}{\sum_{i=1}^{N} t_i}$$

Where t_i is the time to download the ith referenced object from the server to the cache.
Many researches have made on traditional cache replacement algorithms and results showed a considerable improvement in performance of the web server.

LEAST RECENTLY USED: [LRU]
LRU method keeps recently used items near the front end of the cache. Whenever a new item is accessed the LRU places it at the top of the cache. When the cache is filled items that have been accessed recently will be removed.

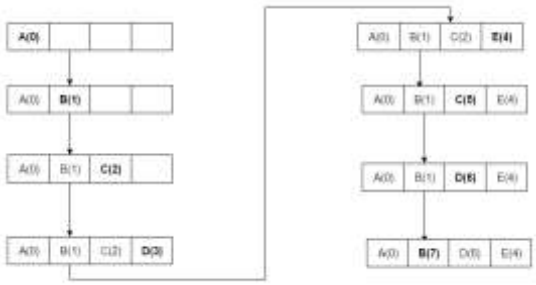

Figure 1-LRU Algorithm

Here when E(4) is inserted the Least Recently Used Object A(0) is removed.
Procedure:
1. Create a url list of records for the items in cache
2. Find out recently used item for that url
3. Append it to the recent used rows list
4. Repeat the steps (I) to (III) for all the items in the cache
5. Now recent used rows contain recently used url records. From that select least recently used url record based on time. Create a list if the list items have equal least time.
6. Remove least recently used row or recently used records obtained from step (5)

LEAST FREQUENTLY USED: [LFU]:
This method uses a counter to check how often an object is used. The value of repeated access is stored and block of objects which are least frequently accessed are removed from the cache. E.g., if A was used (accessed) 5 times and B was used 3 times and others C and D were used 10 times each, we will replace B.

Procedure:
1. Create a list for the items in cache
2. Count number of occurrences of each url item in the list
3. Create a dictionary with url as a key and number of occurrences as its value
4. From the dictionary, select the url with least number of occurrences
5. Remove that url from cache

RANDAM REPLACEMENT (RR):
Randomly selects a candidate item and discards it to make space when necessary. This algorithm does not require keeping any information about the access history.

Procedure:
1. Create a url list for the records in cache
2. Select any one item randomly from the list using random.randint function
3. If the size of the selected item is greater than or equal to the size of requested url, Remove the selected item from cache.
4. Otherwise repeat steps (II) and (III) to select one more record and remove it

IV. Proposed System

This paper analysis the choice of optimum Cache Size so that the traditional Cache Replacement Algorithms will yield better results. LRU,LFU and RR algorithms are used to evaluate the web performance with various cache sizes. The performance metrics used in the proposed system is Hit Ratio, Byte Ratio and Latency Saving Ratio

Experimental Setup:

In order to evaluate the performance of Cache Replacement algorithms we choose a dataset consists of traces contain two month's worth of all HTTP requests to the NASA Kennedy Space Center WWW server in Florida (ftp://ita.ee.lbl.gov/traces/NASA_access_log_Aug95.gz). The data set contains Aug 04 to Aug 31, ASCII format, 21.8 MB gzip compressed, 167.8 MB uncompressed raw data. The log was collected from 00:00:00 August 1, 1995 through 23:59:59 August 31, 1995, a total of 7 days. In this two week period there were 3,461,612 requests.The logs are an ASCII file with one line per request, with the following columns:

1. host making the request. A hostname when possible, otherwise the Internet address if the name could not be looked up.
2. timestamp in the format "DAY MON DD HH:MM:SS YYYY", where DAY is the day of the week, MON is the name of the month, DD is the day of the month, HH:MM:SS is the time of day using a 24-hour clock, and YYYY is the year. The timezone is -0400.
3. request given in quotes.
4. HTTP reply code.
5. bytes in the reply.

Standard pre processing methods are used to pre process the data and essential data are retained for analysis. Cache Size is fixed between the range 10KB to 200 KB and then the three algorithms are applied to evaluate the Hit Ratio, Byte Hit Ratio and Latency Saving Ratio.

V. Results And Discussion

For each algorithm LRU, LFU and RR respectively Hit Ratio, Byte Hit Ratio and Latency Saving Ratio are calculated and the results are tabulated as follows:

Size	HR	BHR	LSR
10KB	0.0513	0.0429	0.1136
20KB	0.0513	0.0432	0.1059
30KB	0.1026	0.0627	0.1178
40KB	0.1026	0.607	0.1164
50KB	0.2051	0.1274	0.0975
60KB	0.2821	0.215	0.0839
80KB	0.2821	0.215	0.0767
100KB	0.3333	0.2934	0.0969
150KB	0.7692	0.695	0.0521
200KB	0.7692	0.695	0.0316

Table 1.1 LRU

Size	HR	BHR	LSR
10KB	0	0	0.307
20KB	0.0526	0.0326	0.29
30KB	0.1023	0.0514	0.2866
40KB	0.1053	0.0507	0.3239
50KB	0.1053	0.0617	0.3131
60KB	0.1579	0.1118	0.3408
80KB	0.2127	0.1056	0.248
100KB	0.3158	0.2351	0.2807
150KB	0.6842	0.5176	0.1265
200KB	0.7632	0.6917	0.1525

Table 1.2 LFU

Size	HR	BHR	LSR
10KB	0.0256	0.0041	0.267
20KB	0.0769	0.0487	0.1992
30KB	0.1026	0.0627	0.1966
40KB	0.1282	0.0871	0.2058
50KB	0.1538	0.1184	0.2272
60KB	0.2564	0.2084	0.2885
80KB	0.3846	0.256	0.2429
100KB	0.5385	0.4918	0.1596
150KB	0.7692	0.695	0.0812

Table 1.3 RR

Table 1.1 to 1.3 shows the Hit Ratio, Byte Hit Ratio and Latency Saving Ratio obtained for various Cache Sizes for the algorithms LRU,LFU and RR Respectively

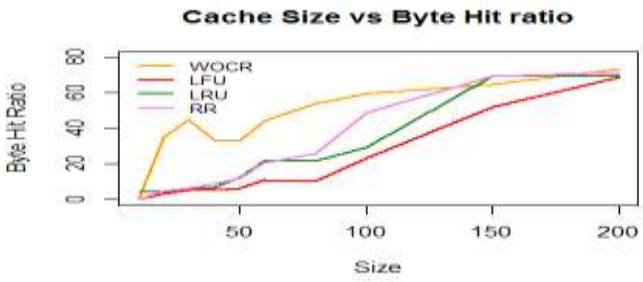

Figure 2

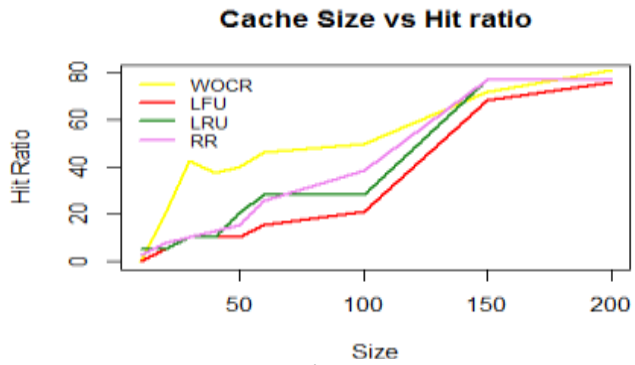

Figure 3

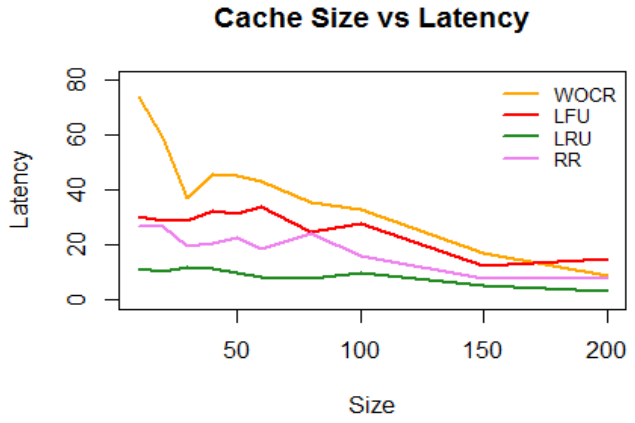

Figure 4

VI. Result

The above said figures represents three different algorithms i.e LRU,LFU and RR and their respective Hit Rations, Byte Hit Ration and Latency Saving Time. In all the three methods increase in Cache Size the HR's are also increased..

An interesting pattern noted in the result is in LRU HR is balanced between 50 to 80 KB cache size. But in LFU and RR it is still in lower cache size i.e 30 to 50 KB

Between 100 to 150KB HR and BHR are steadily increasing according to Fig No.

Between 50 to 80 KB Cache Size LRU shows consistent Hit Ratio and have better user satisfaction than LFU. But RR produces inconsistent result in that category.

In BHR also LRU produces better result than LFU and RR. LRU produces excellent Latency Saving Ratio. It shows that LRU improves network Bandwidth and reduce user perceived latency

VII. Conclusions

Improving the performance of the Web server either in server level or in proxy level is one of the prominent ongoing research area in web mining. Traditional Cache Replacement algorithms like LRU,LFU,RR are used to increase the HR or BHR. In this paper these three algorithms are tested by using a dataset and the outcome shows that LRU shows better HR,BHR and LSR than other two methods. But interesting observation got from the results is that Cache Size is influencing the efficiency of the algorithms as plot shows that LRU is consistent between 50 to 80 KB Cache Size Range. At the outset when we choose appropriate Cache Size with better cache replacement algorithm definitely we can expect good user hit ratio, reduced network bandwidth and reduced user perceived latency.However the same methodology can be applied to large cache size with large size datasets. In addition the same may be applied to hybrid scheme by combining LRU and LFU and the same may be investigated further.

References

[1] 17 06 2017 http://www.internetlivestats.com/
[2] Sofi, Ayaz Ahmad, and AtulGarg. "Analysis of various techniques for improving Web performance." (2015).
[3] Podlipnig, Stefan, and Laszlo Böszörmenyi. "A survey of web cache replacement strategies." ACM Computing Surveys (CSUR) 35.4 (2003): 374-398.
[4] Balamash, Abdullah, and Marwan Krunz. "An overview of web caching replacement algorithms." IEEE Communications Surveys & Tutorials 6.2 (2004).
[5] Wong, Kin-Yeung. "Web cache replacement policies: a pragmatic approach." IEEE Network 20.1 (2006): 28-34.
[6] Romano, Sam, and HalaElAarag. "A quantitative study of recency and frequency based web cache replacement strategies." Proceedings of the 11th communications and networking simulation symposium. ACM, 2008.
[7] Ali, Waleed, SitiMariyamShamsuddin, and Abdul Samad Ismail. "A survey of web caching and prefetching." Int. J. Advance. Soft Comput. Appl 3.1 (2011): 18-44.
[8] Arora, Kapil, and Dhawaleswar Rao Ch. "Web Cache Page Replacement by Using LRU and LFU Algorithms with Hit Ratio: A Case Unification." (IJCSIT) International Journal of Computer Science and Information Technologies 5.3 (2014): 3232-3235.
[9] Hasslinger, Gerhard, et al. "Performance evaluation for new web caching strategies combining LRU with score based object selection." Computer Networks (2017).
[10] Khaleel, Mohammed Salah Abdalaziz, SaifEldinFattoh Osman, and Hiba Ali Nasir Sirour. "Average Least Frequency Used (ALFUR) cache replacement technology USING intelligent agents." Communication, Control, Computing and Electronics Engineering (ICCCCEE), 2017 International Conference on. IEEE, 2017.
[11] Wikipedia: Cache Replacement Algorithms

IOSR Journal of Computer Engineering (IOSR-JCE) is UGC approved Journal with Sl. No. 5019, Journal no. 49102.

Muralidharan Murugesan. "Optimization of Cache Size with Cache Replacement Policy for effective System Performance." IOSR Journal of Computer Engineering (IOSR-JCE), vol. 19, no. 4, 2017, pp. 51-56.