STUDY OF THE WEB CACHING ALGORITHMS FOR PERFORMANCE IMPROVEMENT OF THE RESPONSE SPEED

Dhawaleswar Rao. CH,

Department of Computer Science & Engineering,
Lovely Professional University,
Jalandhar (Punjab), India,
dhawaleswar.14671@lpu.co.in
dhawaleswarrao@gmail.com

Abstract

The increasing demand for World Wide Web (WWW) services has made document caching a necessity to decrease download times and reduce Internet traffic. To make efficient use of caching, an instructive conclusion has to be made as to which documents are to be ejected from the cache in case of cache saturation. A processing unit of web caching contains web objects, and changes in the heterogeneity of these web objects and their reference characteristics become a major factor of decreased performance.

A provocation of web object heterogeneity more frequently produces a replacement of a web object, but cannot reflect enough characteristics of web object in these traditional replacement techniques. Especially size heterogeneity of an object has an influence on performance of a web caching seriously.

In this study, a Response Time Gain Factor is included in this web object replacement algorithm with Size heterogeneity of a web object for performance improvement of the response speed. This factor gives amount of advantage in web cache response time. This is because of Cache. This factor is designed for the average response time gain. With these results, we make a qualitative comparison between these policies and its performance object-hit ratio and an improvement of response speed.

Keywords: World-Wide Web, Size variation, caching, web object

1. Introduction

The World Wide Web growth created as huge increase in network loads and user response time. Web application complexity increased in day by day. Web browsing dominates today's Internet. More than two thirds of the traffic on the Internet today is generated by the Web. In looking at how to improve the quality of service delivered by the Internet, a very productive way to start is examining the performance evaluation of Web transactions.

Until now, different web caching replacement algorithms related to this were researched. A provocation of web object heterogeneity more frequently generates a replacement of a web object, but cannot reflect characteristics of web object in these traditional replacement techniques. Especially size heterogeneity of an object has an influence on performance of a web caching seriously. Therefore, a study on web caching algorithm with size heterogeneity of an object is required. A proxy server is a computer that is often placed near a gateway to the Internet and that provides a shared cache to a set of clients. Client requests enter at the proxy regardless of the Web servers that host the required documents. The proxy either serves these requests using previously cached responses or obtains the required documents from the original web servers on behalf of the clients. It optionally stores the responses in its cache for future use.

In this study, a Response Time Gain Factor (Formula-ii) is included in this web object replacement algorithm with Size heterogeneity of a web object for performance improvement of the response speed. This factor gives amount of advantage in web cache response time. This is because of Cache. This factor is for the average response time gain. With these results, we make a qualitative comparison between these policies and its performance object-hit ratio and an improvement of response speed

2. Related work:

The key feature of the effectiveness of proxy caches is a document placement/replacement algorithm that can produce high hit rate. A number of cache replacement algorithms have been proposed in recent studies, which attempt to minimize various cost metrics, such as hit rate, byte hit rate, average latency, and total cost. They can be classified into the following three categories [1].

2.1 *Traditional replacement policies and its direct extensions:*

- Least Recently Used (LRU): ejects the object which was requested the least recently.
- Lease Frequently Used (LFU): ejects the object which is accessed least frequently.
- Pitkow/Recker : ejects objects in LRU order, except if all objects are accessed within the same day, in which case the largest one is removed.

2.2 Key-based replacement policies:

(i.e. the replacement policies in this category eject objects based upon a primary key)

- Size ejects the largest object.
- LRU-MIN ejects smaller objects. If there are any objects in the cache which have size being at least, LRU-MIN ejects the least recently used such object from the cache. If there are no objects with size being at least, then LRU-MIN starts ejecting objects in LRU order of size being at least. i.e., at the first the object that has the largest size and is the least recently used object among all objects with the same size will be ejected.

- LRU-Threshold is the same as LRU, but objects larger than a certain threshold size are never cached.
- Hyper-G is a refinement of LFU, break ties using the recency of last use and size.
- Lowest Latency First minimizes average latency by ejecting the document with the lowest download latency first.

2.3 Cost-based replacement policies:

(i.e. the replacement policies in this group utilize a potential cost function derived from different factors such as time since last access, entry time of the object in the cache, transfer time cost, object expiration time)

- GreedyDual-Size (GD-Size) associates a size with each object and ejects object with the lowest size.
- Hybrid associates a utility function with each object and ejects the one that has the least utility to reduce the total latency. Lowest utility value. Least Normalized Cost Replacement (LCN-R) makes use of a rational function of the access frequency, the transfer time cost and the size.
- Size-Adjusted LRU (SLRU) orders the object by ratio of cost to size and choose objects with the best cost-to-size ratio.
- Hierarchical GreedyDual (Hierarchical GD) does object placement and replacement in a hierarchy.

In brief, a great deal of effort has been made to maximize the hit rate. However, the performance of replacement policies depends highly on traffic characteristics of WWW accesses.

3. Experiment and Analysis

We established an experimental model that has two kinds of object reference characteristics to evaluate the performance of the proposed method.

3.1 *Comparison:*

Experimental model has a variation in the object. In these experiments, we measured the object-hit ratio, the average object-hit ratio, and the response time, and evaluated them by comparing the proposed algorithm with the LRU, LFU and SIZE of previous algorithm.

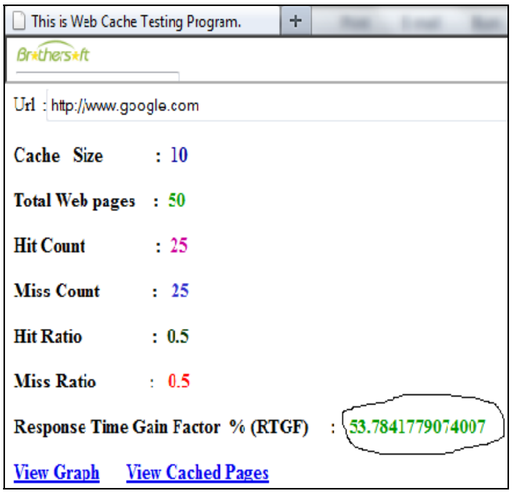


Fig 1: LRU RTGF

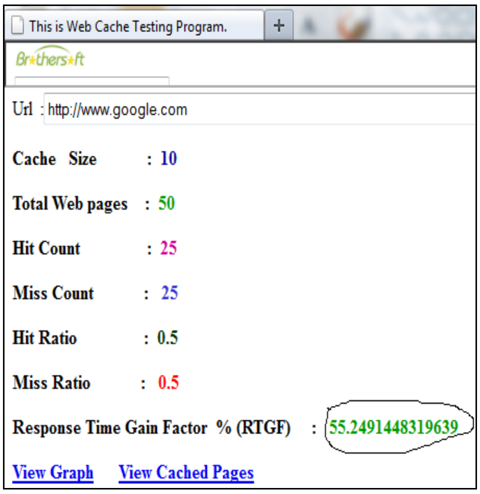


Fig 2: LFU RTGF

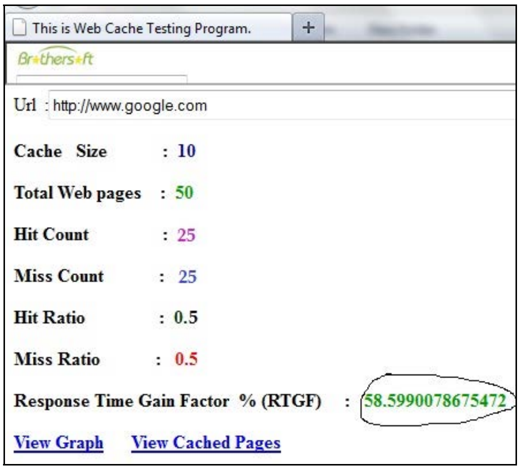
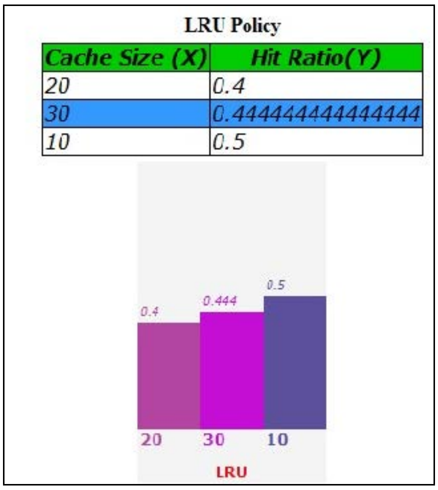


Fig 3: SIZE RTGF

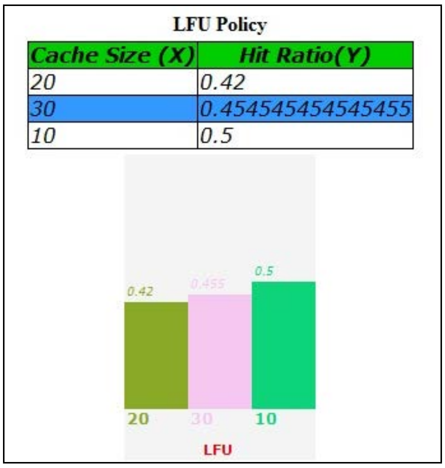
GRAPH:



LRU Policy

Cache Size (X)	Hit Ratio(Y)
20	0.4
30	0.444444444444444
10	0.5

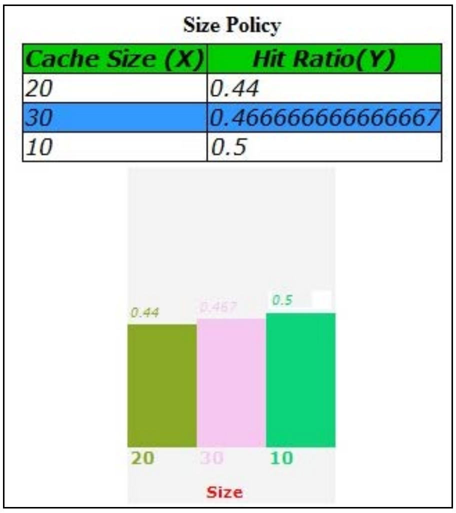
Fig 4: LRU Hit Ratio



LFU Policy

Cache Size (X)	Hit Ratio(Y)
20	0.42
30	0.454545454545455
10	0.5

Fig 5: LFU Hit Ratio



Size Policy

Cache Size (X)	Hit Ratio(Y)
20	0.44
30	0.466666666666667
10	0.5

Fig 6: SIZE Hit Ratio

To reflect the physical environment of the network, we have to consider factors influencing traffic. Of various factors influencing traffic, object size is a factor of the objects themselves. Hence, we can reflect the size factor of web object. An average object size hit ratio reflects the factor of object size to object-hit ratio.

Average-object Hit Ratio: The cache-hit ratio can indicate an object-hit ratio in web caching. The average object-hit ratio can calculate an average value of an object-hit ratio on a requested page the performance is evaluated by comparing an average object-hit ratio and response time [1].

$$\frac{\sum_{i=1}^{n} S\_O_i \times n\_hit(i) \times 100}{\sum_{i=1}^{n} S\_O_i \times n\_req(i)}$$

……Formula (1)

Where

s_oi : size of object i

n_hit(i) : number of hits for object i

n_req(i) : number of requests for object i

Response Time gain factor (RTGF): This factor give you amount of advantage in web cache response time. This is because of Cache. Here RTGF is proposition to Hit Ratio.

$$\text{Response Time Gain Factor} = \frac{(\text{Time Without cache} - \text{Time With cache}) \times 100}{\text{Time Without cache}}$$

…..Formula (2)

This is the Average response time gain for LRU, LFU and Size policy.

We discussed the parameters they use in decision making and made a qualitative comparison between them. We compared policies in the same group irrespective of the performance metric. The web pages are used where some are repeatedly and some are randomly by maintaining same web site access order for LRU, LFU and SIZE policies.

4. Conclusion and Future Work

In this study, a Response Time Gain Factor is included with the web object replacement algorithm with heterogeneity of a web object for performance improvement of the response speed. This factor gives you amount of advantage in web cache response time. This is because of Cache. Here RTGF is proposition to Hit Ratio.

The experiment results were variable depending on the diverse object reference characteristics and various traffic conditions of the network. Further researches are needed on the division-ratio of storage scope and the operation method of cache that considers this diversity dynamically.

5. References

[1] Yun Ji Na1 and Il Seok KO[2] .(2007). *Design and performance Analysis of the Web Caching Algorithm for Performance Improvement of the Response Spee,*. T.-J. Cham et al. (Eds.): MMM 2007, LNCS 4352, Part II, pp. 686 – 693, 2007. Springer-Verlag Berlin Heidelberg 2007.

[2] Korupolu .M. R.; and Dahlin .M. (1999). *Coordinated placement and replacement for large-scale distributed caches*, Proceedings of the IEEE Workshop on Internet Applications.

[3] CISCO Systems (1997). *Web Cache Control Protocol*, White Paper.

[4] Vinod Valloppillil; Keith Ross .W. (1998). *Cache Array Routing Protocol*, v1.0. Internet draft.

[5] Ross K. W. (1997). *Hash-routing for collections of shared Web caches*, IEEE Network Magazine, 11(6):37–44.

[6] Arlitt, M. ; Friedrich . R.; Jin. T.. (1998). *Performance Evaluation of Web Proxy Cache Replacement Policies*, Performance Tools, Palma de Mallora, Spain.

[7] Abrams. M. et al.(1995). *Caching Proxies: Limitations and Potentials*, Proc. 4th Int'l. World Wide Web Conf., Boston University..
[8] Wang . J. (1999). *A Survey of Web Caching Schemes for the Internet*, ACM Comp. Commun. Review, vol. **29**, no. 5, pp. 36–46.
[9] Duane Wessels; Claffy. K. (1998). *ICP and the Squid Web cach*,. IEEE Journal on Selected Areas in Communication, **16(3):**345–357.
[10] Rodriguez. Pablo ; Spanner Christian; Biersack. Ernst W. (1999). *Web caching architectures:Hierarchical and distributed caching*, In Proceedings of the 4th International Web Caching Workshop.
[11] Seok, K. (2006). *ACASH: An Adaptive Web Caching method based on the Heterogeneity of Web Object and Reference Characteristics*, Vol.176, Issue 12, 1695-1711.
[12] Seok, K. (2005). *An Adaptive Web Caching Method based on the Heterogeneity of Web Object*, LNCS3758, pp.853-858.