An Adaptive Web Caching Method based on the Heterogeneity of Web Object

Il-Suk Ko*, Yun-Ji Na**, Chun-Seong Leem***
*,***Dept of Computer and Industrial Eng., Yonsei Univ.
**Korea COD Engineering

웹 객체 이질성 기반의 적응형 웹캐싱 기법

고일석*, 나윤지**, 임춘성***
*,***연세대학교 컴퓨터산업시스템공학과
**한국 COD 공학
e-mail:isko0326@korea.com

Abstract

The use of a cache for storing and processing of Web objects is becoming larger. Also, many studies on the efficient management of the storing scope of caches are being done. Web caching algorithms have many differences from traditional algorithms. Particularly, heterogeneity of Web objects that are processing units of Web caching, and a variation of Web object reference characteristic with time are the important causes of the decrease the performance of existing algorithms. In this study, we proposed the new web-caching algorithm. A heterogeneity variation of an object can be reduced as the proposed method dividedly managing Web objects and a cache scope with heterogeneity, and it is adaptively reflecting a variation of object reference characteristics with the flowing of time.

In the experiments, we verified that the performance of the proposed method was more improved than existing algorithms through the two experiment models which considered heterogeneity of an object.

1. Introduction

With the progress of network technology and advancement of production technology of digital contents, the heterogeneity of Web objects has gradually increased and has gradually had many influences an Web efficiency[1,2,3]. The heterogeneity increase of objects more frequently substitutes Web objects, but traditional substitute techniques cannot reflect enough these characteristics. Also, the reference locality and other reference characteristics vary depending on the flow of time, and are major factors decreasing the performance of previous caching techniques[4][5]. This study proposed Adaptive Caching Algorithm with Size Heterogeneity, which is a Web caching technique which adaptively reflects the heterogeneity of Web objects. The proposed method approached Web caching problems as follows.

□ The heterogeneity of Web objects has a close relationship with size variation, and the identification of objects according to heterogeneity is able to divide objects according to object size.

□ The SIZE technique and LRUMIN algorithm can decrease the cases in which a big size object eliminates many small size objects from the storage domain.

□ By managing objects by unit size, the variation of object size in each storage domain can be decreased compared to management by one unit. We can reduce the number of small size objects needed to substitute a big size object with the decrease of size variation.

□ The reference characteristics of objects is very variable, and the variation of heterogeneity also is variable. An adaptive caching technique needs to reflect variable reference characteristics.

Also, to evaluate the performance, this study presented two

kinds of object use models which reflected the heterogeneity of objects: it analyzed the hitting ratio and response time of objects, and the profit ratio by comparing a developed substitute technique with pervious substitute techniques (LRU, LRUMIN, SIZE). By these experimental results, the developed substitute technique could confirm a greater improvement of performance than the previous substitute techniques.

2. Proposed Method

The highest heterogeneity of the object generates more frequent cache replacement and creates large variations in object size. A division based on the object size can decrease the heterogeneity of the object by reducing the variation in object size in each divided domain.

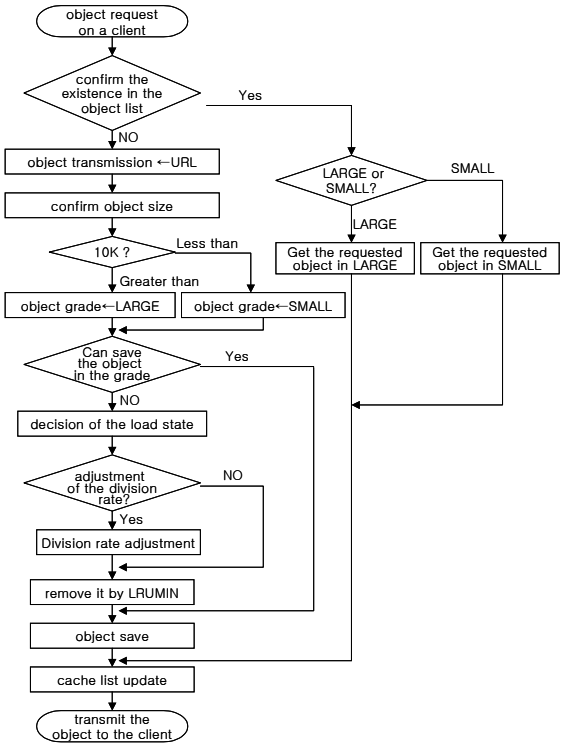

Figure 1 Flow Chart

We divides the storage space of a Web cache into two kinds of domain according to object size. Then the following points are considered:

□ It is best to store web objects as often as possible in a Web cache and to save small-sized web objects in the cache with this perspective.

□ As above research, the replacement of a big-sized object generates the replacement of many small-sized ones at same time. This largely decreases the performance of a cache. The absence of a cache for a big-sized object highly increases network traffic and reduces the performance of a system. Finally, to save a big-sized object in a cache is better with respect to the network.

According to the analysis of web object reference characteristics in prior research, we can identify a large amount of difference between the total transmission quantity and the number of frequency references based on a 10K size object. Therefore, ACASH manages web objects by dividing LARGE (over 10K in object size) from SMALL (below 10K in object size).

Then, the reference characteristics of web objects gain the adaptability according to time flow by managing the division rate of each available cache domain. If there is an object requested by a client, a cache manager checks the existence of the object in the division domain constituted by object size. Generating a cache hit, it thereby provides the object service for a client request, and the cache manager updates the used time record for which this object receives a high priority in a LRUMIN replacement.(see Figure 1)

After generating a cache miss, the cache manager receives the transmission of the object by requesting service from the URL Internet server that generated the cache miss. The transmitted object then divides the object on the basis of object size. The cache manager then checks the storage space to save this object in the cache domain in the proper object level. If there is space to save it, the object is stored in the cache, and if there is no space to save it, the object is assigned to free space by LRUMIN and is saved in the cache. The web object saved in each space level can be placed between same level objects, and high priority is assigned by saving the time record of the newly stored object. At this time, because the proposed method manages the storage space of object by dividing object size into SMALL and LARGE, it has a relatively smaller variation of web object size compared with LRU, LRUMIN and SIZE. Therefore, compared with the previous algorithm, the number of small-sized objects generated by a big size object can be reduced.

Also, the proposed method adaptively reflects change according to the reference characteristics time of a web object. In order to do this, we use ADAPTOR as follows in Figure 2. ADAPTOR is used to ascertain the adaptability of both division rate and action as follows:

1) Each domain is divided into a lightly loaded state, lightly overloaded state, and an overloaded state. The basis rate of a load state (SMALL and LARGE domains) reflects the variation of object size, these difference are presented in Figure 2.

2) The load state of each domain increases according to the reference object for each division domain. This is first divided into 5:5.

3) According to the time flow, each divided ADAPTOR load state is changed by reflecting the reference characteristics of an object. If the load state of a domain arrives at an overloaded state, ADAPTOR checks the load state of different domain.

3-1) If the load state of other domains is smaller than the overloaded state, the total cache division rate for overload state is increased by 5%.

3-2) If the load state of another domain is in a slightly overload state, the total cache division rate operates without an adjustment to the division rate.

4) The rate of one domain cannot be over 70% in a division rate adjustment. In other words, the maximum rate of division is 7:3 or 3:7.

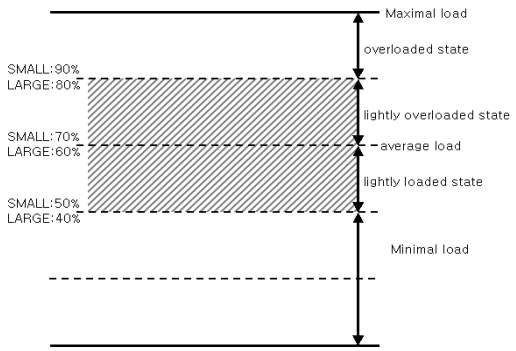

Figure 2 ADAPTOR

3. Experiments and Analysis

We established an experimental model that has two kinds of object reference characteristics, as in Table 1, to evaluate the performance of the proposed method. Experimental model 1 has a generally small variation in the object and many small-sized object references. Experimental model 2 has an almost small variation in the object and many big-sized object references. 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, LRUMIN and SIZE of previous algorithm.

Table 2 indicates the change of the load state of the ADAPTOR in each time interval for models 1 and 2 in the experimental model. Time interval is not an important factor in this experiment because it is affected by both the cache capacity and the reference characteristics of the object. In Table 3, the change in the division ratio of each domain is more significant than is the value of time interval. According to the experiment results of the division change, the division change in experimental model 2 is much quicker and larger than it is in experimental model 1. This is because experimental model 2 is larger than experimental model 1 in regard to the the size of reference objects and fills the cache rapidly.

In light of the experiment results of the object-hit ratio in Figure 3, MODEL 1 resulted in more of a decrease than did the previous algorithm in the object hit-ratio. This is because the object characteristics of MODEL 1 generate frequent object requests for small-sized objects, and the size decrease in the small domain due to the division of large domain is why the new algorithm is better than was the previous algorithm not using the division domain.

In the experiment we considered object size as Figure 5, and the new algorithm indicates an average better performance improvement of 15% against the previous algorithm due to the increase in the object-hit ratio for big-sized objects.

Table 1 Charactristic of Experimental Model

Division	Reference Characteristics	Variation
MODEL1	- Includes a high quantity of HTML text. - High ratio of small-sized GIF, JPEG, AVI	Relatively small
MODEL2	- High ratio of big-sized graphic objects and the objects of moving images such as MPEG	Relatively large

Table 2 Change of Division

Division	Domain	T1	T2	T3	T4	T5	T6	T7	T8
MODEL1	LARGE	5	5	4.5	4.5	4.5	4	4.5	4
	SMALL	5	5	5.5	5.5	5.5	6	6.5	6
MODEL2	LARGE	5	5.5	5.5	6	6.5	6.5	6.5	7
	SMALL	5	4.5	4.5	4	3.5	3.5	3.5	3

Figure 4 is the experiment result of the object-hit ratio for MODEL 2. The object-hit ratio for MODEL 2 has more of an increase than it does for MODEL 1. This is because the average value of the object-hit ratio is raised by many requests for big-sized objects. However, we can ascertain that the object-hit ratio has little decrease or is much the same as the previous algorithm for the same reasons as the experiment results for MODEL 1. The new algorithm indicates an average 30% more than the previous algorithms do, as per the experiment results in Figure 6 showing the improvement of the object-hit ratio for big-sized objects. The difference however of the object hit ratio between the proposed method and the previous algorithms will decrease with the increase in use of a cache exclusive server and cache capacity.

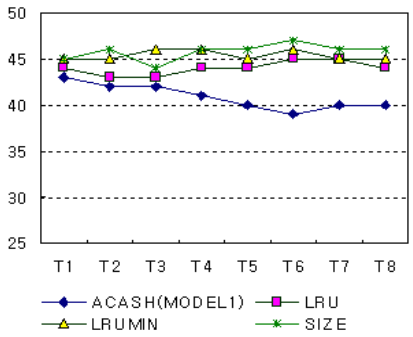

Figure3 object hit ratio(MODEL1)

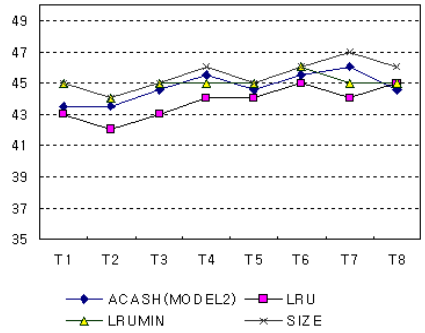

Figure4 object hit ratio(MODEL2)

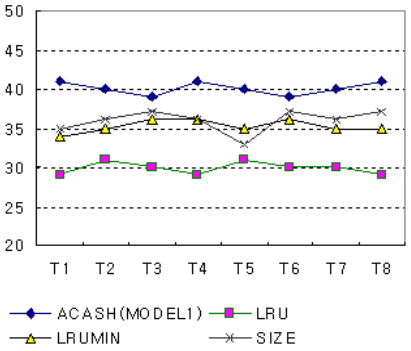

Figure5 object size hit ratio(MODEL1)

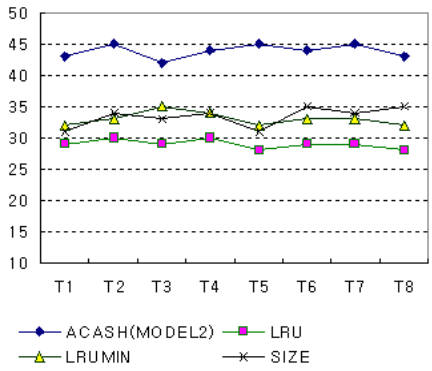

Figure6 object size hit ratio(MODEL2)

4. Conclusion

The various reference characteristics of web users indicated the heterogeneity of web objects, but the traditional replacement algorithm did not reflect this fully. Hence, a web caching algorithm that reflects variations in the reference characteristics of the object is needed. This study proposed the proposed method web-caching algorithm as a tool for adaptively reflecting the heterogeneity and reference characteristics of web objects. The proposed method reflected this heterogeneity based on object reference characteristics and indicated user reference characteristics within the flow of time. In order to evaluate the performance of the proposed method, we performed the experiment by comparing the object-hit ratio, the object size hit ratio and the response time for the previous replacement algorithms, and two experiment models reflected heterogeneity of objects. According to the experiment results, The proposed method verified the improvement of performance in the scale in which considered object size for the previous replacement algorithms is considered.

The proposed method, as proposed by this study, is a replacement algorithm reflecting object reference characteristics based on a key. In future research, an algorithm reflecting the transmission expense on the network and the heterogeneity of web objects need to be developed by studying the mixed form algorithm based on key and expense.

Reference

[1] G. Barish, K. Obraczka, World Wide Web Caching: Trends and Techniques. IEEE Communications, Internet Technology Series, May 2000.

[2] H. Bahn, S. Noh, S. L. Min, and K. Koh, "Efficient Replacement of Nonuniform Objects in Web Caches," IEEE Computer, Vol.35, No.6, pp.65-73, June 2002.

[3] L. Rizzo, L. Vicisano, "Replacement Polices for a Proxy Cache," IEEE/ACM Trans. Networking, vol.8, no.2, pp.158-170, 2000.

[4] C. D. Murta, Virgilio A. F. Almeida, Jr. W. Meira, "Analyzing Performance of Partitioned Caches for the WWW," In Proceedings of the Third International WWW Caching Workshop, Manchester, England, June, 1998.

[5] Annie P Foong, Yu-Hen Hu, Dennis M Heisey, Web Caching: Locality of References Revisited, IEEE International Conference on Networks (ICON'00), Singapore, September 05-08, 2000.