웹 객체 크기 이질성 기반의 웹 캐시 대체 기법의 설계와 성능 평가

나 윤 지[†] · 고 일 석[††] · 조 동 욱[†††]

요 약

웹 캐시의 효율적인 사용은 웹기반 시스템의 효율성을 판단하는 중요한 척도가 되고 있다. 캐시의 성능은 한정된 공간에서 객체를 동적으로 관리하는 대체 알고리즘에 크게 영향을 받는다. 본 연구에서는 웹기반 시스템의 효율을 높이기 위한 웹 캐싱 알고리즘을 제안한다. 제안 알고리즘은 분할 영역에 대해 웹 객체의 크기 참조 특성과 이질성을 고려하도록 설계되었다. 또한 실험 환경에서 제안 알고리즘이 기존의 대체 알고리즘을 비교하여, 15% 이상 성능이 향상되었음을 확인하였다.

키워드: 웹 캐싱, 객체 이질성, 캐싱 시스템, 웹 객체, 참조특성

A Design and Performance Analysis of Web Cache Replacement Policy Based on the Size Heterogeneity of the Web Object

Yun Ji Na[†] · Il Seok Ko[††] · Dong Uk Cho[†††]

ABSTRACT

Efficient using of the web cache is becoming important factors that decide system management efficiency in the web base system. The cache performance depends heavily on the replacement algorithm, which dynamically selects a suitable subset of objects for caching in a finite cache space. In this paper, the web caching algorithm is proposed for the efficient operation of the web base system. The algorithm is designed based on a divided scope that considered size reference characteristic and heterogeneity on web object. With the experiment environment, the algorithm is compared with conservative replacement algorithms, we have confirmed more than 15% of an performance improvement.

Key Words : Caching Algorithm, Web Object, Object-hit Ratio, Reference Characteristics

1. Introduction

Cache performance depends on replacement algorithms, which dynamically selects a suitable subset of objects for caching in a finite cache space[1, 2]. Researches on the replacement algorithm have been conducted actively using web object at storage scope of a cache[3, 4, 7]. A replacement algorithm for web cache must reflect characteristics of web object, and web objects which user request have the following reference characteristics[5, 6]. First, web object size referred to by a user is greatly variable, and the variable object that web user requests must be efficiently supported by web cache. Second, reference characteristics are variable by an object preference of web user, and a variation of the size is also very large. Third, user reference characteristics have the temporal locality and the spacial locality.

Reducing total cost incurred due to cache-misses is more important in these caching environments. Therefore, a replacement policy is required that reflects these reference characteristics of web object. However, the traditional algorithm cannot reflect enough reference characteristics of these web object characteristics.

In this paper, we proposed a web cache replacement policy based on divided web cache scope. Emphasis is placed on improvement of the object-hit ratio and the response time. In the experiment results, the policy is compared with previous replacement policy, and the proposed

† 정 회 원 : 호남대학교 인터넷소프트웨어학과 전임강사
†† 정 회 원 : 충북과학대학 전자상거래과 조교수
††† 정 회 원 : 충북과학대학 정보통신과 교수
논문접수 : 2004년 9월 7일, 심사완료 : 2005년 2월 14일

policy improves the object-hit ratio and response speed.

2. Related works

Web caching is classified into several types : the client caching, the server caching, the proxy caching, the hierarchical caching, and the cooperation server caching[1]. The common goal of these web caching method is an efficient operation of the limited storage scope. We can increase the performance of web caching by saving the frequently used object in the storage scope of cache. Therefore, an efficient object replacement policy is the most important factor for performance improvement of web caching[8].

The following descriptions indicate how each web caching policies select a victim-object to purge from the cache space.

2.1 LRU(least recently used)

Removes the least recently referenced object first. LRU is a algorithm to replace an unused object in storage scope so that a new object gets a storage space.

2.2 LFU(least frequently used)

Removes the least frequently referenced object first[4]. LRU and LFU have applied the traditional replacement algorithm to web caching field to have the specialty that was an object of variable size among the traditional replacement algorithms.

2.3 Size

Removes the largest object first. SIZE is a algorithm to replace the largest object among the objects saved in storage scope so that a new object gets a storage space. The web cache is different from traditional cache as a hardware cache and a file system cache. In web cache the unit of an exchange is a web object, and the size of an object is very variable. Therefore, in a case where the size of many objects is small, they are removed from storage scope by one object whose size is large[3]. The algorithm of SIZE improved this issue by replacing the greatest object among objects of cache storage scope for new object.

2.4 LRU-MIN

Removes the least recently referenced object whose size is larger than desired free space of size s[4]. If enough space is not freed, LRU-MIN sets s to $s/2$ and repeats the procedure until enough space has been freed. LRU-min is a transformation of LRU.

2.5 SLRU(size-adjusted LRU)

$Value(i) = (1/t_{ik})*(1/s_i)$, where s_i is the size of object i, and t_{ik} is the number of references since the last time i was referenced.

3. Log analysis of the web objet references characteristics and Replacement policy

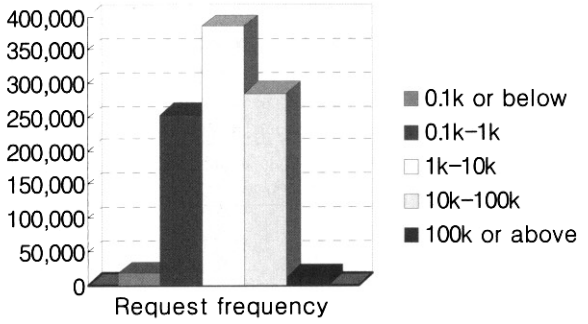
(Fig. 1-3) show the results of log analysis of web object reference characteristic. Many requests were frequently made for objects of 1k-10k bytes like (Fig. 1). And request frequency rate on web object of 1k-10k bytes was the highest like (Fig. 2). Frequency characteristics of the user have a locality on the object-size as shown in (Fig. 1). Web is affected by various physical factors. Therefore, object reference characteristics of specific time cannot reflect total reference characteristics of web object. So, we must reflect the request frequency and size characteristic of web object in order to reflect an influence of networks effectively. Total transmission quantity can reflect size characteristics of web object well as in (Fig. 3).

Total transmission quantity is the value that multiplied the request frequency by the requested object size. The ratio of the request frequency of an object of 1k-10k bytes is the highest in the request frequency side, but the rate of an object of a 10k-100k bytes is the highest in the total transmission quantity side. This means that cache-misses occurring in web object on 10k-100k bytessize will suddenly increase the network traffic. Analysis of the results shown in Figures led us following conclusions on influences of reference characteristics and heterogeneity. First, The request for a small size object (10k bytes or below) is frequent, and according to this, a small size object generates a frequent replacement than a large size object (10k bytes or above). Second, Requests for a large size object are few in frequency side but generate a suddenly increase network traffic.

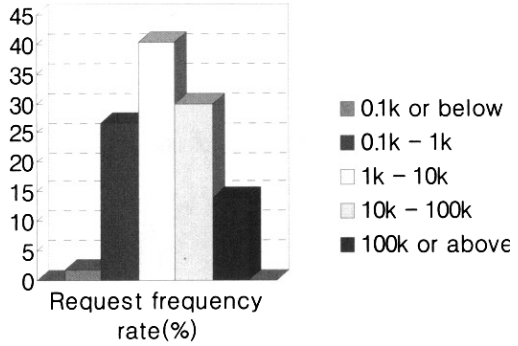
We can get the following conclusion to have an influence on performance of web cache with this log analysis.

1) The frequent cache replacement occurring by cache-miss of a small size object is a factor to lower the efficiency of a cache. Therefore we must reduce the number of frequent replacements of a small size object.
2) The cache replacement occurring by cache-miss of a large size object becomes a factor responsible for sudden increase of networks traffic and for the decrease in the network's efficiency. Therefore we should reduce the replacements of a large size object.

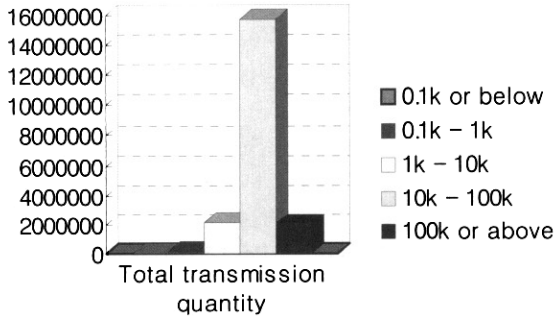
3) Various heterogeneities of an object make a size variation of an object seriously, and this becomes factors to generate a frequent cache replacement. Therefore the object replacement algorithm that considers size reference characteristics of an object is required in order to increase the efficiency of web caching.



(Fig. 1) Request frequency of object

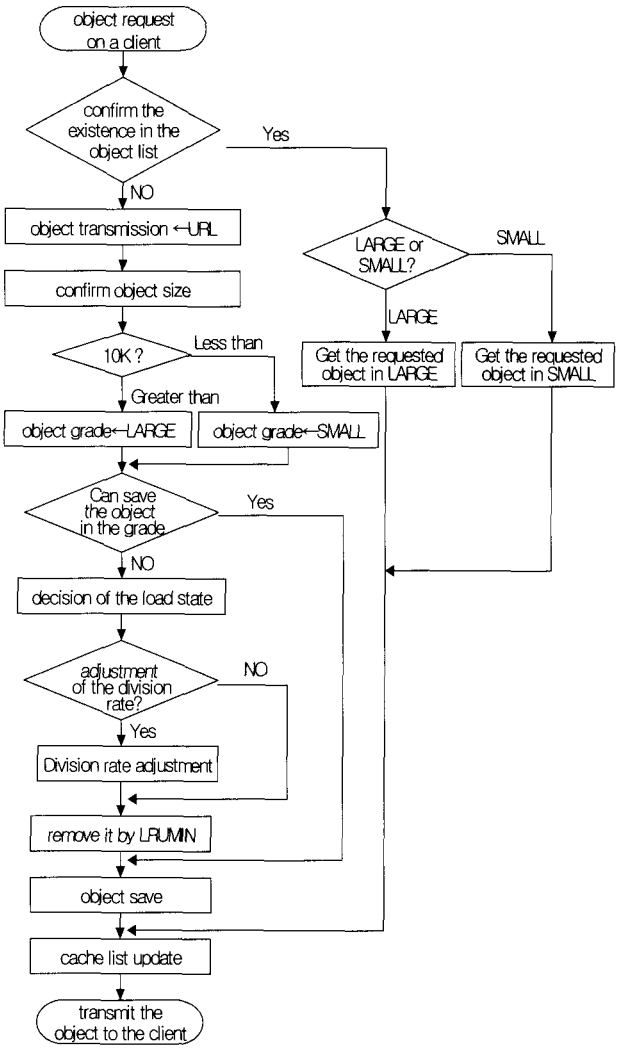


(Fig. 2) Request frequency rate



(Fig. 3) Total transmission quantity

The number of division and the volume of scope to be allocated to divided scope have important influence on the performance of web caching in proposed algorithm. The object storage scope of 10k bytes or above allocate grade LARGE, and the object storage scope of 10k bytes or less allocate grade SMALL. (Fig. 4) shows the proposed replacement algorithm. When object request of a client comes, the cache administrator confirms the existence of the object in the corresponding division scope. If a cache-hit occurs, the cache server transmit the object to a client. And the cache administrator updates the time record on when it was used so that high ranking is assigned to this object in an LRU-MIN replacement.



(Fig. 4) Object replacement policy

If there is a plentiful free space to save the object in the cache scope, the object is saved in this free space. But if it is not, the object is saved using LRU-MIN replacement policy by remove a certain object in the cache scope. Web objects which save on each grade scope are substituted among the same grade object. Also, high ranking is updated of the referenced object for the LRU-MIN replacement policy.

The proposed algorithm is similar to LRU and LRU-MIN, SIZE in the reference tendency. But there are differences in the following points.

First, LRU refer to the time record immediately before an object is referred to. And it didn't reflect a reference

frequency and the an object size.

Second, LRU-MIN operates with LRU similarly. But the LRU-MIN substitute a small sized-object unlike proposed policy. Bad cases on LRU-MIN, a lot of small-sized objects are removed by one large-sized object. The algorithm of SIZE improved this issue by replacing the greatest object among objects of cache storage scope for new object. But LRU-MIN and SIZE are not reflecting size heterogeneity of object, either.

4. Experiments and Analysis

In experiments, we measure object-hit ratio and average object-hit ratio, response time. The cache-hit ratio can indicate an object-hit ratio in the web cache. The average object-hit ratio can calculate an average value of an object-hit ratio on the requested objects as following Equation (1).

$$\frac{\sum_{i=0}^{k}(OB_{hit(i)} \times SO_{hit(i)})}{\sum_{i=1}^{n}(OB_{req(i)} \times SO_{req(i)})} \text{ - Equation} \quad (1)$$

$OB_{req(i)}$: number of the requested object
$SO_{req(i)}$: size of the requested object
$OB_{hit(i)}$: number of the hit object
$SO_{hit(i)}$: size of the hit object

An average object-hit ratio is calculated by a ratio of hit object size in size of the total object that a client requested. It means that a high object-hit ratio provides a faster response regarding a request of a client in web caching, and is capable of efficiently reducing the load of a system. Also, it can decrease the traffic of the main system and the local server system.

And the response time RT has the several response time notations, RT_{ch}(Response Time of cache-hit) on the cache-hit and response time RT_{cm}(Response Time of cache-miss) on the cache-miss. Response time for an object request of a client has delay factors like <Table 1>.

4.1 A case of cache-hit

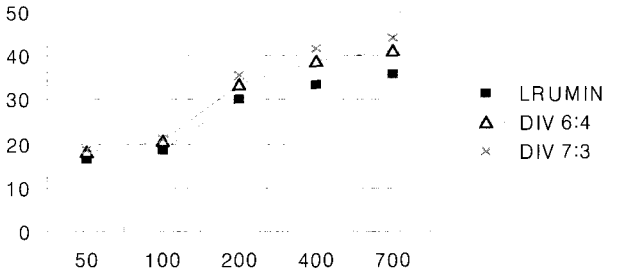
$$RT_{ch}=TDT_{client\_to\_cache}+DDT_{cache}+SDT_{cache}+TDT_{cache\_to\_client} \text{ - Formula} \quad (2)$$

4.2 A case of cache-miss

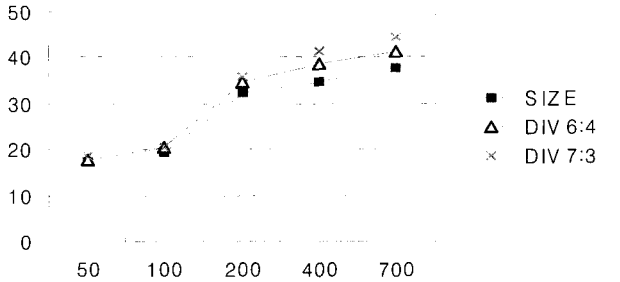
$$RT_{cm}=TDT_{client\_to\_cache}+DDT_{cache}+TDT_{cache\_to\_URL}+TDT_{URL\_to\_cache}+RDT_{cache}+TDT_{cache\_to\_client} \text{ - Formula} \quad (3)$$

〈Table 1〉 Delay factors

Delay factors	Cause of the delay
$TDT_{client\_to\_cache}$	Transmission delay time occured when a client requests an object to cache server
DDT_{cache}	Delay time required for a determination of cache-hit or cache-miss of cache server
SDT_{cache}	The delay time required for a search of an object saved in Large or Small scope of cache
$TDT_{cache\_to\_client}$	Delay time required when an object is transmitted from cache server to a client
$TDT_{cache\_to\_URL}$	Transmission delay time required when cache server requests an object to URL
$TDT_{URL\_to\_cache}$	Delay time needed when an object is transmitted from URL to cache server
RDT_{cache}	Delay time that occurs in cache server when an object is replaced

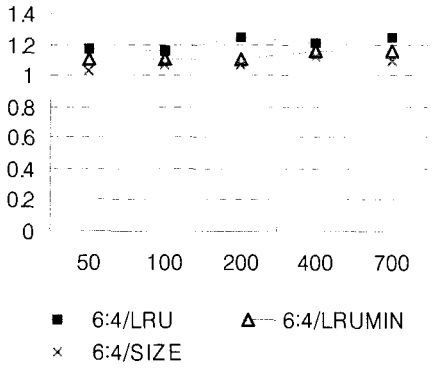


(Fig. 5) Object-hit ratio(%) : compared with LRU-MIN

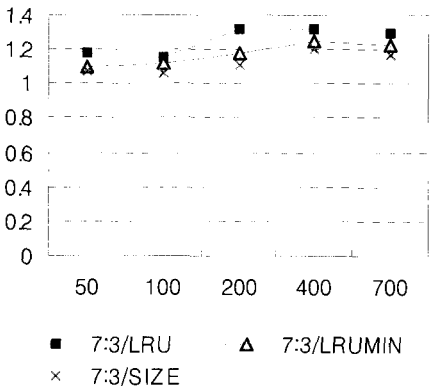


(Fig. 6) Object-hit ratio(%) : compared with SIZE

The response speed has a close relationship with the object hit ratio. Four delay factors occur with response time of a hit object in cache as in Formula (2), but six delay factors occur following the pattern of Formula (3) in the response time of a miss object. Among these delay factors, $TDT_{client\_to\_cache}$, $TDT_{cache\_to\_client}$, $TDT_{cache\_to\_URL}$, and $TDT_{URL\_to\_cache}$ are affected by the physical environment of networks. Therefore, delay time gets longer than cache-hit for the cache-miss because of the many influences of the physical environment. Also, the RDT_{cache} is the delay time that occurs in cache server when objects replaced have a lot of influences on the performance of web caching in Formula (3). Therefore, if we increase the object-hit ratio in experiment one, we can improve the response time in experiment two.



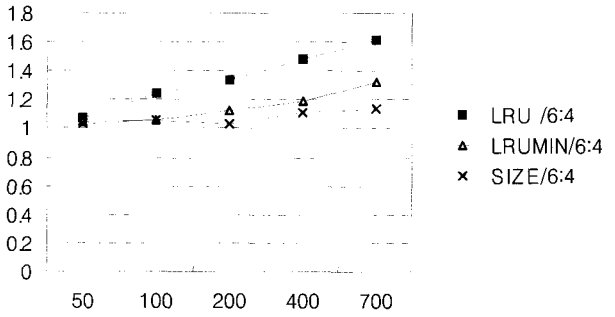
(Fig. 7) Gain ratio of object-hit on 6 : 4



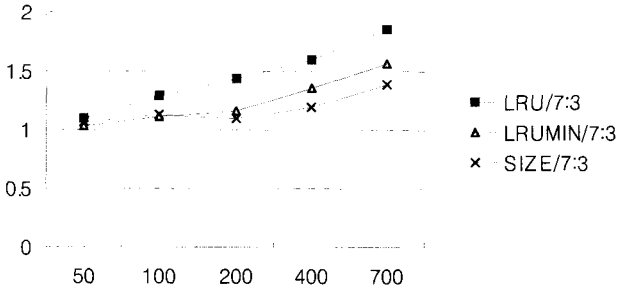
(Fig. 8) Gain ratio of object-hit on 7 : 3

Firstly we measured object-hit ratio. The experiment method is as follows. 70% on the cache scope was assigned first to a LARGE grade, and 30% was assigned to a SMALL grade. And the experiments were conducted on object-hit performance of this algorithm and LRU-MIN, SIZE. Second, 60% on the cache scope was assigned to a LARGE grade, and 40% was assigned to a SMALL grade. Also, we experimented on the performance of these algorithms. In the figure which showed the examination result, DIV6 : 4 stands for 6 : 4 division, and DIV7 : 3 stands for 7 : 3 division. The experiment results of object-hit ratio are shown in (Fig. 5) and (Fig. 6). In experiment result figures, x-axis denotes the cache space size and y-axis denotes the hit ratio.

(Fig. 7) shows gain ratio of object-hit on 6 : 4 compare with LRU and LRU-MIN, SIZE. And (Fig. 8) shows gain ratio of object-hit on 7 : 3 compares with LRU-MIN and LRU-MIN, SIZE. In experiment result figures, x-axis denotes the cache space size and y-axis denotes the gain ratio. In these experiments of a hit ratio, we can get conclusions as follow. First, In the cache with small-sized capacity, the hit ratio of LRU and LRU-MIN, SIZE and the proposed algorithm were almost similar. Second, As the cache capacity grew larger, the performance of LRU and LRU-MIN, SIZE is improved. In experiment result, we can know the performance of SIZE and LRU-MIN was better than the performance of LRU. Third, And as the capacity of cache grew larger, the hit ratio performance of the proposed algorithm is more efficient than traditional replacement algorithms, and we can get 10%-15% performance enhancement.



(Fig. 9) Gain ratio of response time on 6 : 4



(Fig. 10) Gain ratio of response time on 7 : 3

Response time is the time required to provide the requested web object to a client. (Fig. 9) and (Fig. 10) show the gain ratio on response time. The experiments were also performed as an object-hit ratio in experiment one. Also, experiments were conducted on response time performance of this algorithm and LRU-MIN, SIZE. We reached the following conclusion by the experiments results of a response time. First, As the cache scope grows larger, the proposed algorithm and LRU-MIN, SIZE all improved their response time. Second, The performance of traditional algorithms and the proposed algorithm were almost similar about small-sized cache. Third, As the capacity of cache grew larger, the response time performance of the proposed algorithm is more efficient than traditional replacement algorithms. Forth, As for the gain ratio of response time, we can get 15% or above performance enhancement than LRUMIN and SIZE. The reason that performance enhancement of gain ratio is higher than performance enhancement of object hit ratio originated in size of the object which user refer to. There was comparatively

a lot of reference on large-sized object in the experiment. According to this, response time was affected delay time of network greatly.

5. CONCLUSION

In this study, we can get the following conclusions about the influencing factors on the performance of web caching. User's reference characteristics take great influences on the performance of web caching. The proposed algorithm was based on reference characteristics, it can improve the performance of web caching.

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.

REFERENCES

[1] G. Barish, K. Obraczka, World Wide Web Caching : Trends and Algorithms. 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 Transaction of Networking, Vol.8, No.2, pp.158-170, 2000.

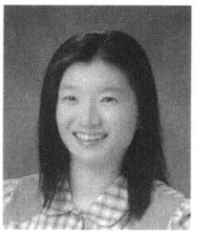
[4] S. Williams, M. Abrams, C. R. Standridge, G. Abhulla and E. A. Fox, "Removal Policies in Network Caches for World Wide Web Objects," In Proceedings of 1996 ACM Sigcomm, pp.293-304, 1996.

[5] V. Almcida, A. Bestavros, M. Crovella, and A. Oliveira, "Characterizing Reference Locality in the WWW," In Proceedings of the 4th Int'l Conference on Parallel and Distributed Information Systems, 1996.

[6] J. C. Bolotand P. Hoschka, "Performance engineering of the World-Wide Web : Application to dimensioning and cache design," In Proceedings of the 5th Int'l WWW Conference, 1996.

[7] M. Abrams, C. Standridge, G. Abdulla, S. Williams and E. Fox, "Caching Proxies : Limitations and Potentials," In Proceedings of 4th Int'l World Wide Conference, 1995.

[8] C. Aggarwal, J. Wolf and P. Yu, "Caching on the World Wide Web," IEEE Transaction of Knowledge and Data Engineering, Vol.11, No.1, pp.94-107, 1999.



나 윤 지

e-mail : yjna2967@korea.com
충북대학교 컴퓨터공학(공학박사)
뉴욕공대(NYIT) 대학원 Communication ART학과
충북대학교 컴퓨터공학(공학석사)
경북대학교 생명공학(이학사)
현재 호남대학교 인터넷소프트웨어학과 전임강사

고 일 석

e-mail : isko@ctech.ac.kr
연세대학교 컴퓨터산업시스템공학(박사수료)
미)USIU 경영학과(MBA)
경북대학교 컴퓨터공학(공학석사)
경북대학교 컴퓨터공학(공학사)
현재 충북과학대학 전자상거래과 조교수

조 동 욱

e-mail : ducho@ctech.ac.kr
한양대학교 전자통신공학과(공학박사)
한양대학교 전자공학과(공학석사)
한양대학교 전자공학과(공학사)
전)서원대학교 정보통신과 교수
현재 충북과학대학 정보통신과 교수