인터넷 데이터 접급 속도 향상을 위한 WWW 캐시 교환 알고리즘 개발

허윤정, 이양원, 김철원

호남대학교

ywlee@honam.ac.kr

Development of WWW cache replacement algorithm for the Internet data access

Yeun-Jeong Heo, Yang-weon Lee, Chul-Won Kim

Honam University

E-mail : ywlee@honam.ac.kr

요 약

인터넷 사용자들의 증가와 함께 네트워크에서 데이터 교환량이 증가하고 있다. 따라서 네트워크 트래픽과 서버의 부담으로 인하여 WWW 서버로부터 응답 시간의 하락은 많은 문제로 제기되고 있다.

본 논문에서는 WWW cache 서버를 이용하여 응답시간을 줄수 있는 방안을 제시한다. 이런 목적을 위하여 www cache 서버를 원격으로 교체할 수 있는 알고리즘을 제안한다.

I . Introduction

In recent years, information services on wide area network using WWW has been widely spread. As a result, the delay of the response time of WWW services, because of the network traffic and the burden on the WWW server, has become a big problem. The network traffic and burden on the WWW server are encouraged by redundancy of requesting the same pages by many people, even though they browse the same ones.

One of the solutions for these problems is utilization of WWW cache server. For instance, squid, apache server, CERN httpd server and Delegate server are known as proxy server.

The outline of the function of the WWW cash server is as follows. When a user demands page data from WWW and WWW cache server holds the object that the client required in the cache, the cache server transmits the object to the client instead of requesting it to WWW server. And when the required object is not in the cache of the WWW cache server, while acquiring an object from the original WWW, it stores the object also in own cache. By realizing reuse of the same object with the cache function, the amount of communications data between the WWW server and the WWW cache server are cut down, and average response time is also shortened.

On the other hand, there is no useful cache replacement algorithm, which is used commonly by the WWW server, taking the information of network-distance into account to store and manage the page data. Though, some existing cache replacement algorithms consider the size of the page and the access frequency of the response time, existing algorithm are not sufficient.

The purpose of the study is to realize a WWW server that reduction of the stress on the improvement of the response time. To achieve the end, we propose the new cache replacement algorithm by focusing on the utilizable information of network distance from the WWW cache server to WWW server that possessing the page data of the user requesting.

II. Existing Cache Replacement Algorithms

Cache replacement algorithm is the method that determines page data to be evicted if there is little space for storing new ones. In this section, we explain three major cache replacement algorithms, Least Recently used(LRU), Least Frequently Used with Dynamic Aging(LFUDA) and Greedy-Dual Size Frequency(GDSF). Moreover, we sort out the problems of each algorithms.

2.1 LRU

LRU is one of the most commonly used cache replacement policies. LRU deletes the object that has been used (read or written) less recently than any other object. This simple algorithm uses only reference history of FIFO to evaluate the value of the data. However, there are rooms for improvement in the byte hit rate and the hit rate in this algorithm, because LRU does not consider the frequency of the reference and size of the data.

Key = (Last referred time)

2.2 LFUDA

LFUDA is a variant of Least Frequently Used that uses a dynamic aging policy to accommodate shifts in the set of popular data. In the dynamic aging policy, the cache age factor is added to the reference count when a data is added to the cache or an existing data modified. LFUDA used reference history and the frequency of the reference to evaluate the value of the data. With the use of frequency of the reference, LFUDA is expected to cache the dta having possibilities of being accessed compared with LRU. However there are rooms for improvement in byte-hit rate, because LFUDA takes no account of the dta size.

Key = (Cache retension period)
+ (reference count)
- (formula for adjustment)

2.3 GDSF

GDSF is a variant of the Greedy Dual-Size policy that considers frequency of the reference. GDSF is optimized for caching more popular and smaller objects in order to maximize object hit rate. GDSF policy assigns a value to each object computed as the object's reference count divided by its size, plus the cache age factor. By adding the cache age factor, we limit the influence of previously popular documents, as secured above in the part of LFUDA. Since GDSF considers the size of the data besides the reference history and the frequency of the reference, LFUDA is expected to achieve the improved hit rate compared with LFUDA. However, there still are rooms for improvement in byte-hit rate, because GDSF caches smaller-sized data.

Key = (Cache retention period)
+ (reference count)/size
- (formula for adjustment)

All the algorithm introduced above does not utilize the information of the network-distance to evaluate the page data, therefore there still are rooms for improvement in the response time.

III. Cache Replacement Algorithm considering the network distance

As we mentioned in section 2, existing cache replacement algorithm mainly focus on the access frequency of each data, and no cache replacement algorithm considers the network distance between the WWW cache server and the WWW server. Therefore the data that is accessed fewer and placed further network tends to be evicted rather than the data accessed more often and placed nearer. this is one of the reasons of the fall of the response time. In the other words, the existing cache replacement algorithm are not designed for improving the response time. thus we can improve the response time by considering the information of the network distance to the WWW sever.

In our study, we suggest a new cache replacement algorithm, which is stressed on the improvement of the response time, considering the network distance besides other factors. To prove the effectiveness of it, we examined the relationship between the network distance and the response time.

3.1 Analysis of log in a WWW cache server

We calculated network distance of all WWW servers, which are picked from an access log on a WWW cache server running in the University of Honam. To this end, we measured the average response time of a transmitted ICMP echo packet to the WWW servers. As for some servers that did not

permit the use of the ICMP echo packets, we used approximate value of the average response time to measurable place by traceroute command as the network distance to the WWW server, which had the demand from the user, was approximately 120.2ms. The request whose network distance ws less than 67ms occupied about 50% of the whole, and the request whose network distance was less than 200ms occupied about 80%, and other 20% was dispersed between 200ms and 5545.5ms. As for the ratio of the completion time of acquisition of the dta from the WWW server, the request whose network distance was less than 67ms accounted for 16%, and even the request to the WWW server placed nearer than 200ms was only 28% of the whole.

Judging from the result above, the request to the WWW server whose network distance is more than 200ms, though it held only 20% of the number of the total access, accounted for no less than 72% of the overall completion time of acquisition of the object. Therefore, we can say that most of the access to the WWW server are accounted by the request whose network distance is more than 200ms. thus, we can improve the whole response time by caching the object whose network distance is more than 200ms preferentially.

3.2 Cache Replacement Algorithm by consideration of Network distance

From the above suggestions, we utilize the information of the network distance in addition to access frequency and the size of the object. We set higher priority to the object whose data size is larger and network distance from the WWW cache server is further, and set same degree of priority to the larger sized object that is in the further the WWW server and the smaller sized object that is in the nearer one, when the object is referred y the client, we arrange the page data in the priority in the reference order list to adjust the length of the stay of the page data. Moreover, we use the distance time of 200ms as the basis of the priority value, because it is thought to be effective judging from the log analisys.

Though, we use the ICMP echo packet to measure the network distance, it is inefficient if we send it to all the servers in the log list. therefore, we only send the ICMP echo packet to the newly accessed the WWW server, and as for the WWW server accessed in the past, we does not send it if it is accessed in 5 minuets. The reasons why we set up the period of measure to 5 minutes are that there are little difference of the response time of the ICMP echo packet sent within 5 minutes, and the result of taking the load of the WWW cache server caused by sending the ICMP echo packets into consideration. As the basic value of performance evaluation, the hit rate, which is calculated by dividing the number of the pages by the number of requested object, and the byte hit rate, which is calculated by dividing the data total size of whole page that is hit by the number of total requested object are used under normal conditions.

suggestion method 1:

Key = (Cache retention period)
+ (The network distance)
- (formula for adjustment)

suggestion method 2:

Key = (Cache retention period)
+ (reference count)/size
+(The network distance)
- (formula for adjustment)

3.3 Implementation

To implement suggesting method and verify the effectiveness of the cache replacement algorithm, we used Apache web server, a powerful and high efficiency WWW cache server program that is distributed under GPL and the current major release version is Apache 2.4. We can mount the new algorithm and verify it easily.

IV. Conclusion

In this paper, we prove the effectiveness of the information of the network distance from the WWW cache server to the WWW server based on the experimental analysis of a log in WWW cache data running in a real network. Then, we propose the new cache replacement algorithm and experimentally mounted this algorithm that utilizes the information of the network distance. To be more precise, we mounted the cache replacement algorithm which utilized the network distance of the cache object besides the dta size and access frequency as indication of its efficiency, and then validate the effectiveness by system operation data.

V. References

[1] "Apache software foundations",
http://www.apache.org