A Study on a Design of Efficient Electronic Commerce System

고일석*, 신승수**, 최승권***, 조용환***
충북과학대학*, (주)시그마정보기술**
충북대학교***

Ko Il-Seok*, Shin Seung-Soo**,
Choi Seung-Kwon***, Cho Yong-Hwan***
Dept. of e-commerce, Chungbuk Provincial University, Assistant Professor*
Sigma Information Technology Co, Ltd**,
Chungbuk National Univ.***

요약

Now that the e-commerce users are explosively increasing, there is always followed by a sharp increase in load from the e-commerce system and the heavy traffic on the network, where leads to the delayed service for the client's request. The natural consequences contain decreasing customer satisfaction and weakening the business's competitive position in markets. Therefore, we'll need to study the e-commerce system in due consideration of the operational efficiency and response speed. This study includes a design of e-commerce system, with a hierarchical structure based on the local server, which has the capability of caching necessary to distribute the system load, makes a proposal concerning a split web-cache algorithm devoted to the local web server to finally give an analysis of the performance through the appropriate trial.

1. Introduction

The developed IT technology and the rapid spread of Internet are not only expanding the e-commerce system suddenly but also occasioning a sharp increase in the number of users. These current situations bring about the heavier network traffic in addition to a sudden increase of load to the e-commerce system, and also incur the unnecessary squandering of the system resources as the repeated requests for the same object occupy a large portion of the network bandwidth. Therefore, it's necessary to study the e-commerce system in consideration of the operational efficiency and response speed. There are several factors that have impact upon the user's response speed on the Internet as follows: size of the object, advantageous position, status of the network traffic, and performance of the server. To improve the physical factors such as traffic condition, performance of the server, the required cost will be relatively considerable. For that reason, we shall require improvement of the response speed through the dispersion of the load and traffic, rather than betterment of each individual system's own performance. Our efforts to fulfill the needs will just have to go through an innovative improvement in network traffic condition and server performance.

In this study, we designed an e-commerce system with the capability of web cache based on the split area in a hierarchical structure and then assessed the performance through an experimental trial. The local server for the proposed system is, as a replicated server for the e-commerce system, capable of shortening latency time by the system's advantageous location as well as improving response speed through the dispersion of the entire system's load. A Design was devised to decrease impact that the repetitive request for the same object have upon the network bandwidth through the Web Cache function of the local server.

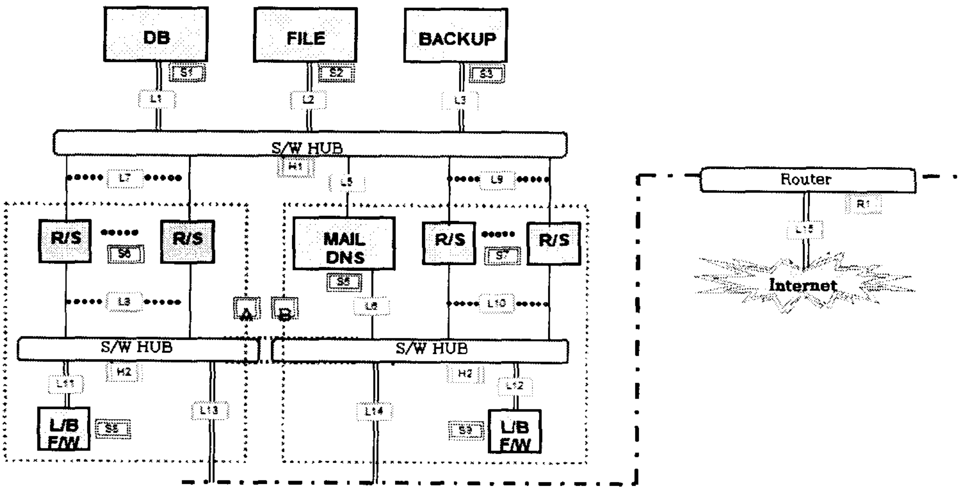
2. A Design of E-Commerce System

2.1 Configuration of System

The proposed system consists of multiple servers variably depending on the system load. The e-commerce system enables the distribution of traffic among the multiple servers - the component elements of a web server through load balancing (L/B). Figure 4 shows t he experimental model of e-commerce system.

Listed below are features and functions of the component servers listed in Figure 1

- DB Server : It manages a variety of Databases such as merchandise information, customer information, registration information server over e-commerce sites operating on the Real Server and provides services for a request of a Mail Server.
- Case Base Server : It is used to provide services for the customer who uses the e-commerce system through the case base. Its setup may be independent of or include a File Server.



▶▶ Figure 1 The E-Commerce System Configuration: An Experimental Model of e-Commerce System

- Backup Server : It is responsible for the second function to perform backup for a variety of DB servers and used to make clear the stability of data in the middle- or large-scale e-commerce system though being omitted sometimes in setting up a small-scale e-commerce system.
- Real Server : It plays a part in providing services to the customer directly through diverse capabilities over the web site such as membership, purchase, approval, etc.. Also it is set up with plural servers in consideration of distributing loads.
- Load Balancer : It distributes loads among the plural component servers. Most web servers are composed of multiple sub-servers, and as such can be fulfilled through the efficient distribution of traffic between the lower servers that construct a web server by way of load balancing .
- Local Server : It provides the capability of caching for a request of a user which connects with each local server in addition to the function of main system replication. This will provide a good remedy of latency time against the user's requirement.

Table 1 provides the list of system specifications and component s as applied for the experimental model. Experimental equipment was based on an alpha processor mounted system. As seen through Table 1, the experimental system was not prepared in the latest environment of equipment and network but optimal configuration of the system could make it clear to predict a potential improvement of its performance. As an experimental model of system implementation for e-commerce, available specifications are equivalent to Apache Web Server for the environment of Red Hat Linux7; hp4 for the server script language in Java Script and HTML; and Mysql for the database management system.

[Table 1] Component Elements and System Specifications of the Experimental Model

Component	Number	System Setup
DB Server	S1	CPU: Alpha 21264 667Mhz*2 Memory: 1GB HDD1: 9GB*1 HDD2: RAID Controller, 18.2GB*4 4 Ethernet (100Mbps)
File Server	S2	CPU Alpha 21264 667Mhz*2 Memory: 1GB HDD1: 9GB*1 HDD2: RAID Controller, 18.2GB*4 4 Ethernet (100Mbps)
Backup server	S3	CPU: Alpha 21264 667Mhz Memory: 500MB HDD1: 9GB*1 HDD2: RAID Controller, 18.2GB*4 4 Ethernet (100Mbps)
Real Server	S6,S7	CPU: Alpha 21264 600Mhz Memory: 256MB HDD1: 9.1GB*1 2 Ethernet (100Mbps)
Load Balancer	S8,S9	CPU: Alpha 21264 600Mhz Memory: 500MB HDD: 9.1GB*1 2 Ethernet (100Mbps)
Mail/DNS server	S5	CPU: Alpha 21264 667Mhz Memory: 1GB HDD1: 9GB*1 HDD2: RAID Controller, 18.2GB*4 4 Ethernet (100Mbps)
Switching Hub	H1,H2	24 Port Fast Ethernet switching HUB
Network	L15	100Mbps

Also, a speedy response is an absolute requirement to the DB server and file server for the customer and merchandise information service. Therefore, to improve the response speed, we have completed modeling a system with two mountable CPUs. Sever that Alpha Processor was loaded is a Processor product of

Samsung Electronics Company. In the prototype experiment, the server with two processors specified to Alpha 21264 667Mhz CPU is used as DB server and the file server Is equivalent to a model which is specifically mounted with Alpha 21264 600Mhz CPU, 256MB Memory, and 9.1GB*1 Hard Disk. For the real server and load balancing, we applied the same system model as this one. But we use neither mail/DNS server nor backup server that would have no effect on the assessment of performance.

2.2 Web Caching

The object replacement algorithm that considers an object size is required in order to increase the efficiency of web caching through the previous log analysis. The idea of a proposal algorithm is as follows. First, it is possible to classify objects based on size characteristics, and to manage the divided storage of a cache efficiently. Second, reference characteristics of an object are variable. Therefore, efficiency based on the divided size of cache storage is varied, too.

The number of division scope, the volume of scope to be allocated to divided scope, and the determination of size to classify an object have an important influence on the performance of web caching in this algorithm. Storage scope of the object that has an influence on web caching must be assigned in order to increase the hit ratio of cache.

The object storage scope of 10k or above is divided into scope LARGE, and the object storage scope of 10k or less is divided into scope SMALL. Figure 3 shows the replacement algorithm that has divided cache scope.

When the object requests of a client arrive, the cache administrator confirms the existence of an object in the corresponding division scope according to the size of an object. The service of the object that a client required would be provided if a cache-hit occurs. Then, 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.

If a cache miss occurs, the cache server requests the corresponding URL for the service of the object, and the object is transmitted to the cache server. Then, a transmitted object is classified into a corresponding grade according to size, and a cache administrator confirms whether there is a space for this object to be saved in a cache scope of a corresponding grade.

If there is a space to save in the cache scope, this object is saved, and this object is saved by an LRU-MIN replacement algorithm if it is not there. Then, the web object saved in each scope is substituted among the objects of the same grade. Also, a time record of the newly arrived object is saved, and a high ranking is assigned to this object in an LRU-MIN replacement process.

As was mentioned in the previous reference characteristics analysis of an object, the reference characteristics and the heterogeneity of web objects would be affected by the characteristics of web service and the user's aging characteristics and user's academic background, as well as a timing factor. The web service that includes many different kinds of multimedia data and the object reference of a comparatively young age user increase the object reference to large size. According to this, the object reference characteristic has an extreme variation. Therefore, the size of cache scope division must be varied.

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 is referring to the time immediately before an object was referred to and to the size of an object among the past object reference information, and it is not reflecting a reference frequency and the heterogeneity of an object. Second, basically, LRU-MIN operates with LRU equally. But the point that substitutes a small size object for it at first in order to substitute the minimum object for it is different in reflection on a size of an object. In the worst case, 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 heterogeneity of object, either. Third, the proposed algorithm can reflect the size and heterogeneity of an object sufficiently.

This study experimented on the performance of a cache scope divided by 6:4 and 7:3. But the efficiency of web caching may be increased more if the division scope of variable size is used according to reference characteristics of an object than by division scope of size that has been fixed.

3. Analysis

Generally, for the performance evaluation measure of the caching algorithm, hit ratio is used. And for the performance evaluation of the e-commerce system, response speed is mostly used.

In this experiment, the performance of the proposed algorithm is evaluated by comparing an average gain ration of object-hit. And the performance of the proposed system is evaluated by comparing a response speed.

Response speed RT have the several response speed notations, RT_{ch}(Response speed of cache-hit) on the cache-hit and response speed RT_{cm}(Response speed of cache miss) on the cache miss. Response speed for an object request of a client has the following delay factors.

① $TDT_{client\_to\_cache}$: Transmission delay time that occurs when a client requests an object to cache server

② DDT_{cache} : Delay time required for a determination of cache-hit or cache miss of cache server

③ SDT_{cache} : The delay time required for a search of an object saved in Large or Small scope of cache

④ $TDT_{cache\_to\_client}$: Delay time required when an object is transmitted from cache server to a client

⑤ $TDT_{cache\_to\_URL}$: Transmission delay time required when cache server requests an object to URL

⑥ $TDT_{URL\_to\_cache}$: Delay time needed when an object is transmitted from URL to cache server

⑦ RDT_{cache} : Delay time that occurs in cache server when an object is replaced

1) A case of cache-hit

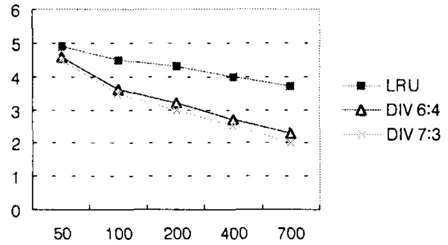
$RT_{ch}=TDT_{client\_to\_cache}+DDT_{cache}+SDT_{cache}+TDT_{cache\_to\_client}$ - Formula (1)

2) A case of cache-miss

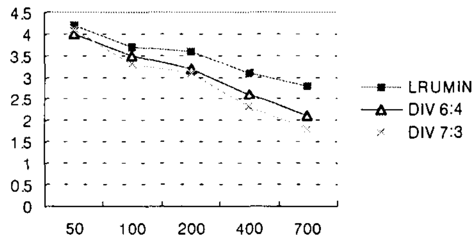
$RT_{cm}=TDT_{client\_to\_cache}+DDT_{cache}+TDT_{cache\_to\_URL}+TDT_{URL\_to\_cache}+RDT_{cache}+TDT_{cache\_to\_client}$ - Formula (2)

The response speed has a close relationship with the object-hit ratio. Four delay factors occur with response speed of a hit object in cache as in Formula(1), but six delay factors occur following the pattern of Formula(2) in the response speed 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(2). Therefore, if we increase the object-hit ratio, we can improve the response speed.

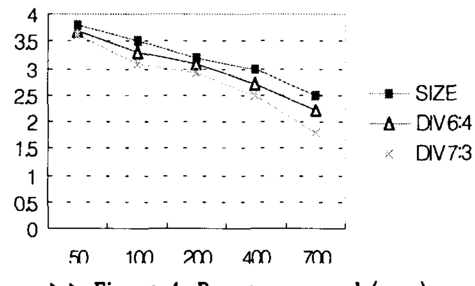
First, we measured object-hit ratio. The experiment method is as follows. 70% on the cache scope was assigned first to a LARGE grade, and 30% was assigned to a SMALL grade. And the experiments were conducted on object-hit performance of this algorithm and LRU, LRU-MIN, SIZE. Second, 60% on the cache scope was assigned to a LARGE grade, and 40% was assigned to a SMALL grade. Also, we experimented on the performance of these algorithms.



▶▶ **Figure 2. Response speed (sec.) : compare with LRU**



▶▶ **Figure 3. Response speed (sec.) : compare with LRU-MIN**



▶▶ **Figure 4. Response speed (sec.) : compare with SIZE**

Response speed is the time required to provide the requested web object to a client(customer). Figure 2 - Figure 4 show the results of the experiment on response speed. And Figure 10 - Figure 11 show the gain ratio on response speed.

Experiments were conducted on response speed performance of the supposed system with LRU, LRU-MIN, SIZE. We reached the following conclusion by the experiments results of a response speed.

1) As the capacity of cache grew larger, the response speed performance of the proposed system is more efficient than traditional replacement algorithms.
2) As for the gain ratio of response speed, we can get 30% or above performance enhancement than LRU. Also, 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 speed affected the delay time of the network greatly.

6. Conclusions

This study includes a design of hierarchical e-commerce system using the local server in order to decrease the decline of the system's response speed due to various geographical factors; proposes a partition-based web cache algorithm for the exclusive cache server in order to remedy latency due to object size; and verifies the usefulness of the system and algorithm.

Based on the experiment results, the proposed web cache algorithm indicates a higher hit ratio than the existing cache algorithm and therefore guarantees improved response speed and more flexibility for the client's various requests. It also prevents a decline of response speed caused by the multiplex requests from multiple users concurrently using the e-commerce system.

References

[1] Abrams M., Standridge C. R., Abdulla G., Williams S. and Fox E. A., 1995. Caching Proxies: Limitations and Potentials, Proc. 4th Int'l World Wide Conf.

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

[3] Bolot J. C. and Hoschka P., 1996. Performance engineering of the World-Wide Web: Application to dimensioning and cache design," Proc. of the 5th Int'l WWW Conf.

[4] Cardellini V., Colajanni M. and Yu P. S., 1999. Dynamic Load Balancing on Web-Server Systems, IEEE Internet Computing, pp.28-39.

[5] Kangasharju J., Ross K. W., 1999. A Clustering Structure for Reliable Multi-casting, Computer Communications and Networks, 1999. Proceedings, Eight International Conference, pp.378-383.

[6] Leem C. S. and e-Biz Lab., 2000. e-Business File. YoungJin Biz.com, Seoul, Korea

[7] Zwass V., 1996. Electronic commerce: structures and issues. International journal of electronic commerce.