로컬서버와 웹캐싱을 이용한 전자상거래 시스템의 지연속도 개선에 관한 연구

윤 용 기[†]·고 일 석[††]·임 춘 성[†††]

요 약

전자상거래 사용자와 대용량 멀티미디어 데이터의 증가로 인한 전자상거래 시스템의 급격한 부하 증가와 네트워크 트래픽의 증가는 클라이언트의 요청에 대한 서비스의 지연을 가져와 고객에 대한 고객만족도 감소와 기업의 경쟁력을 떨어뜨리는 요인이 되고 있다. 따라서 전자상거래 시스템의 효율적인 관리와 응답속도를 고려한 연구가 필요하다. 본 연구에서는 전자상거래 시스템의 부하 분산을 위해 캐시 기능을 가진 로컬서버 기반의 계층적 구조의 전자상거래 시스템을 설계하고 로컬 웹서버를 위한 분할된 웹캐시 알고리즘을 제안하며 실험을 통해 성능을 분석하였다.

A Study on Latency Time Improvement of Electronic Commerce System Using Local Server and Web Cashing

Yong Ki Yoon[†]·Il Seok Ko[††]·Choon Seong Leem[†††]

ABSTRACT

Increasing of electronic commerce users and high-capacity multimedia data is the main causes of increasing of load and network traffic in electronic commerce system. Increasing of load and network traffic brings the delay of service about client's request. And it's becoming factor that decreases customer satisfaction and corporation's competitive ability. So it is needed to research that consider efficient management and response time of electronic commerce system. In this paper, we designed electronic commerce system based on local server with web cashing function in order to solve load problem of electronic commerce system. And we proposed web cashing algorithm for local web server and analyzed performance by an experiment.

키워드 : 로컬서버(Local Server), 웹캐싱(Web Cashing), 전자상거래 시스템(Electronic Commerce System)

1. 서 론

최근 인터넷의 급속한 확산으로 인터넷을 통한 거래 형태인 전자상거래(Electronic Commerce : EC)가 상거래를 위한 중요한 채널로 부각되게 되었다. 효율적인 전자상거래의 활용을 위해서는 사용자 중심의 인터페이스를 제공해야하며 쉽고 빠르면서도 사용자가 만족할만한 수준의 상품정보를 제공할 수 있는 적응성(adaptiveness)을 가지고 있어야 하며 사용자 측면이 고려된 시스템의 개발이 필수적이다. 이를 위해 지능적인 에이전트 기술을 도입한 전자상거래 지원시스템[1, 2]에 대한 연구와 사용자의 요구에 대한 응답속도의 개선을 위한 각종 연구가 필요하다.

인터넷을 통한 전자상거래 사용자의 폭발적인 증가는 전자상거래를 위한 시스템의 급격한 부하 증가와 네트워크 트래픽을 증가시키고 있으며 동일한 객체에 대한 중복된 요청이 네트워크 대역폭의 상당 부분을 차지하여 불필요한 시스템 자원의 낭비를 초래하고 있다. 이에 따라 인터넷 서비스 제공업자(ISP : Internet Service Provider)들은 시스템의 관리와 회선의 유지를 위해 막대한 비용을 지불하고 있는 실정이며 전자상거래 시스템에 대한 부하와 트래픽의 증가는 고객에 대한 응답 속도의 저하를 가져와 e-비지니스 고객에 대한 고객만족도를 떨어뜨리는 요인이 되고 있다. 따라서 전자상거래 시스템의 효율적인 관리와 응답속도

† 준 회 원 : 연세대학교 대학원 기술경영학협동과정
†† 정 회 원 : 충북과학대학 전자상거래과 교수
††† 정 회 원 : 연세대학교 정보산업전공 교수
논문접수 : 2002년 7월 27일, 심사완료 : 2002년 10월 23일

를 고려한 전자상거래 시스템에 대한 연구가 필요하다. 인터넷상에서 사용자의 응답속도에 영향을 미치는 요소는 객체의 크기, 지리적인 위치와 네트워크 트래픽의 상태, 서버의 성능이 있다. 트래픽의 상태나 서버의 성능과 같은 물리적인 요인의 개선에는 상대적으로 많은 비용이 소요된다. 따라서 개별 시스템 자체의 성능향상 보다는 부하와 트래픽의 분산을 통한 성능을 향상을 위해 많은 연구가 진행되고 있다. 전자상거래 시스템 자체에 대한 성능 개선을 통한 응답 속도의 개선은 네트워크 트래픽의 상태와 서버의 성능 개선을 통해 이루어질 수 있으며 이를 위해서는 시스템의 부하를 분산시키는 것과 인터넷 캐싱을 활용한 전자상거래 시스템의 설계 및 개발이 필요하다. 또한 전자상거래를 지원하는 시스템에서 다루어야할 동영상 자료가 전체 객체 요청에서 차지하는 비중이 점차 늘어나고 있으며, 고객의 선호도에 따라 특정한 객체에 대한 요청이 집중적으로 이루어진다. 따라서 웹캐시의 기능은 전자상거래를 지원하는 시스템의 성능 향상을 위해 필수적이다.

본 연구에서는 계층적 구조를 가진 전자상거래시스템을 설계하고 실험을 통해 그 성능을 평가하였다. 제안한 시스템에서의 지역서버는 전체 시스템의 부하를 분산시켜 지역적인 요인과 메인 시스템의 부하로 인한 응답속도를 개선할 수 있다. 또한 지역서버의 웹캐시 기능을 통해 동일한 객체에 대한 중복된 요청이 네트워크 대역폭에 미치는 영향을 줄일 수 있게 하였으며 분할 영역을 기반으로 한 웹캐시 알고리즘을 제안하고 실험을 통해 성능을 분석하였다. 실험 결과 본 연구에서 제안한 분할영역 기반의 웹캐싱은 기존의 LRU 기법에 비해 클라이언트의 객체 요청에 대한 객체 적중률을 향상시켰으며, 이는 전자상거래 시스템의 고객 요구에 대해 개선된 응답 속도와 빠른 적응성을 가질 수 있다.

2. 관련 연구

단일 서버로서의 성능향상은 물리적인 확장의 한계로 인하여 네트워크(Network) 트래픽의 증가추세를 따라오지 못하고, 성능대비 투자비용이 기하급수적으로 증가되어서 효율성이 매우 떨어졌다. 근래에는 시스템의 성능을 향상시키기 위한 방법으로 동적인 로드밸런싱[3] 기법과 네트워크에 접속된 여러 개의 서버를 하나로 연결하여 부하를 분산[4]시키는 클러스터(Cluster)[6] 기법이 대두되고 있으며 월드와이드 웹서비스 기반하의 클러스터 기법을 웹 클러스터(Web Cluster)라 한다. 부하 분산 기법은 크게 사용자 측면에서의 부하 분산과, 서버 측면에서의 부하 분산 기법으로 나누어 볼 수 있다. 사용자측에서의 부하 분산 기법은 사용자가 클러스터링된 웹서버들 중 부하가 상대적으로 적게 걸리는 서버를 동적으로 선택할 수 있도록 하는 방법으로 Berkely smart client[9]와 Bandwidth Probing을 이용한 동적인 서버 선택 등이 있다[10]. Berkely smart client는 사용자에게 클러스터링된 웹서버들에 대한 부하 정보를 얻어올 수 있는 애플릿을 제공해 사용자의 접속을 분산시킴으로써 웹서버의 부하를 분산시키는 기법을 이용하고 있다. 그러나 사용자 측면에서의 부하 분산은 사용자의 응용프로그램을 바꾸어야 한다는 점과 네트워크 부하를 증가시킨다는 단점 때문에 대부분의 연구는 서버 측면에서의 부하 분산 문제에 대해 다루고 있다.

서버 측면에서의 부하 분산은 사용기법에 따라 RR-DNS (Round Robin Domain Name System), 응용계층 스케쥴링 기법, IP 계층 스케쥴링 기법으로 나누어 볼 수 있다[11]. 서버측에서 부하 분산을 사용하는 방법은 RR-DNS 기법을 사용하는 NCSA 웹서버가 대표적이며[12], RR-DNS 기법과 응용계층 스케쥴링 기법을 사용한 것으로는 SWEB[13], 그리고 IP 계층 스케쥴링 기법을 채용한 것으로는 IBM사의 TCP_ROUTER[8]와 Cisco사의 Local Director[14]등이 대표적이다.

또한 웹 클러스터를 구현하는 방식에는 구현 아키택처(Architecture)에 따라 라운드 로빈 DNS(Round-robin DNS), 브로드케스팅(Broadcasting), 직접 라우팅(Direct Routing), 디스패처(Dispatcher) 방식 등으로 나눌 수 있다. 라운드 로빈 DNS 방식[15]은 사용자가 해당 서비스의 도메인 주소(Domain Address)를 DNS 서버를 통해 IP주소로 변환하는 과정에서 변환을 요청할 때마다 여러 서버의 다른 IP를 알려 주어 부하를 분산하는 방식이다. 브로드케스팅[16] 방식은 물리적으로 네트워크에 도달한 트래픽을 모든 서버에 전달시키고 그 중 특정한 한 서버만이 응답을 하는 방식으로, 이더넷(Ethernet)의 브로드케스팅(Broadcasting) 특성을 이용한 방식이다. 직접 라우팅[12] 방식은 요청을 중계해주는 서버가 존재하여 클라이언트의 요청을 서비스 서버에 연결하여 주는 방식이다. 클라이언트의 최초 요청은 중계서버를 거쳐 서버에 연결이 되지만 요청에 대한 서버의 응답은 클라이언트와 직접 통신하는 특성을 갖는다. 이외에도 디스패처 방식 등이 있다.

인터넷상에서 사용자의 응답속도에 영향을 미치는 요소는 객체의 크기, 지리적인 위치와 네트워크 트래픽의 상태, 서버의 성능이 있다. 하지만 트래픽의 상태나 서버 자체 성능의 향상을 통한 물리적인 요인의 개선은 상대적으로 많은 경비가 요구된다. 캐싱(Caching)은 인터넷의 객체를 저장하여 시스템의 성능을 향상시킬 수 있다. 캐싱을 통해 서버의 부하와 전체 네트워크의 트래픽을 감소시킬 수 있으

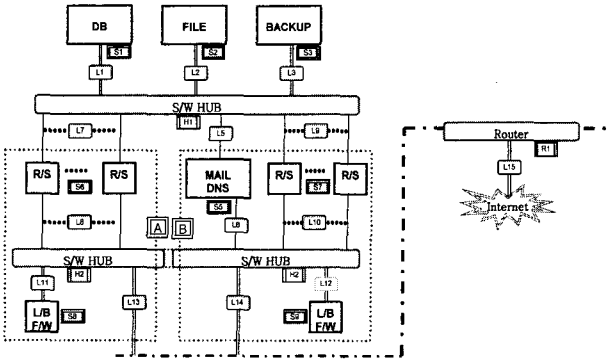
며, 사용자에게 문서의 빠른 응답 시간을 제공할 수 있다[4, 5]. 캐싱을 위한 대체 알고리즘의 성능은 캐싱 기법의 중요한 성능요소이며 FIFO(First In First Out), LRU(Least Recently Used), LFU(Least Frequently Used), SIZE와 같은 여러 알고리즘들이 연구되었다[4,5]. 인터넷 캐싱은 가변 크기의 인터넷 객체를 지원해야 하기 때문에 가상 메모리에서 사용되는 전통적인 대체 기법과는 다르다. 캐싱의 대상이 되는 인터넷 객체의 크기는 다양하다[6]. 경우에 따라 단 한번 또는 거의 사용되지 않을 큰 객체의 대체는 작은 크기의 자주 사용될 객체 여러 개를 제거할 수 있으며 이로 인해 캐싱의 효율은 매우 떨어지게 된다. 인터넷 캐싱의 관건은 한정된 캐시의 저장공간의 효과적인 관리이며 캐싱에 사용되는 시스템의 성능과 지리적인 위치는 대체 알고리즘의 성능과 함께 사용자의 지연 속도를 줄일 수 있는 중요한 요인이다. 또한 캐시서버로만 동작하는 별도의 시스템의 설치를 통해 사용자의 응답속도를 개선하는 캐시전용서버의 설치로 응답지연시간의 개선을 가질 수 있으며 캐시 전용서버를 사용하는 경우 이에 사용되는 효율적인 대체알고리즘의 연구가 필요하게 된다.

본 연구에서는 시스템의 지리적인 요인으로 인한 응답속도의 저하를 줄이기 위해 로컬서버를 이용한 계층적인 전자상거래 시스템을 설계하였다. 또한 객체의 크기로 인한 지연속도를 개선하기 위해 전용캐시서버를 위한 웹캐시 알고리즘을 제안하고 그 효용성을 입증하였다.

3. 제안시스템

3.1 시스템의 구성

전자상거래 시스템의 구성은 기본적으로 웹서버의 구성과 동일하다. 전자상거래를 위한 웹자료들은 점차 멀티미디어화 되어 가고 있으며 이는 시스템의 구성에서 다음과 같은 요소를 고려하게 하고 있다. 첫째는 네트워크의 제한된 대역폭 문제이다. 대부분의 ISP는 이 문제의 해결을 위해 많은 비용과 노력을 기울이고 있다. 둘째는 고객의 정보 요청 시에 발생하는 지연 시간의 문제이다. 이는 특히 동시에 집중적으로 발생하는 다수의 사용자에 대해 더욱 심각한 문제가 되고 있다. 셋째는 신뢰성과 안정성의 문제이다. 오류가 없는 정보의 교환을 위해서는 상대적으로 네트워크 상에서의 부가적인 오버헤드가 발생하며 이는 지연시간의 증가를 가져와 시스템의 성능을 저하시키는 요인이 되고 있다. 이러한 점들을 고려하면 전자상거래 시스템은 안정적이고 고객의 요구에 대해 발생할 수 있는 지연 시간을 최소화할 수 있어야한다. (그림 1)은 본 실험에서 사용된 시스템 모델이다.

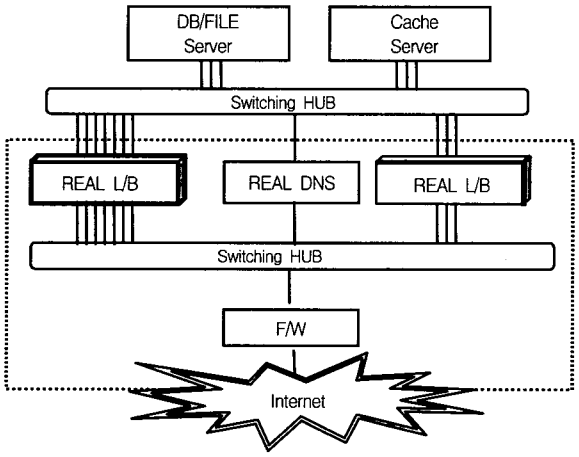


(그림 1) 실험 모델

(그림 1)에서 S는 서버를 나타내며 H는 허브, L은 라인을 나타내는 네트워크를 나타낸 것으로 이에 대한 자세한 설계는 <표 2>에 나타나 있다. DB Server는 Real Server에서 동작하는 전자상거래 사이트에서 상품정보와 고객 정보, 등록 정보 등과 같은 각종 데이터베이스를 관리하는 서버이며, Real Server와 Mail Server의 요청에 대한 서비스를 제공하게 된다. 전자상거래 시스템은 고객의 구매정보와 같은 히스토리 기능을 사례로 저장하고 이를 기반으로 타겟 마케팅(target marketing), 크로스셀링(cross selling), 업셀링(up selling)과 같은 기능을 통해 원투원 마케팅을 통해 고객에게 특화된 서비스를 제공할 수 있으며 이를 위한 Case Base Server는 사례기반을 통한 맞춤형 서비스의 제공을 위해 사용되며 File Server와 별도로 구성이 되거나 함께 구성이 된다. Backup Serve는 각종 DB Server의 백업을 위한 2차 기능을 담당하며 소규모 전자상거래 시스템의 구성에서는 생략되기도 하지만 대부분의 중대규모 전자상거래 시스템에서는 자료의 안정성을 위해 사용된다. Real Server는 회원가입, 물품구매, 결재 등과 같은 웹사이트의 각종 기능을 통해 고객에게 서비스를 직접 제공하는 역할을 담당하며, 부하의 분산을 고려하여 다수의 서버로 구성된다. Load Balancer는 다수 개로 구성된 서버 사이에서 부하의 분산을 위해 사용된다. 대부분의 웹서버는 다수의 서버로 구성이 되며 로드밸런싱을 통해 웹서버를 구성하는 서버간의 트래픽의 효율적인 분산을 통해 웹서버의 부하를 분산시킬 수 있다[3]. Local Server는 메인 시스템의 복제 시스템으로서의 기능과 각 로컬서버에 접속하는 사용자의 요청에 대한 캐싱을 제공하여 사용자가의 요구에 대한 지연시간의 향상을 가질 수 있게 한다. Local Server는 (그림 1)의 시스템 구조에서 라우터를 통해 연결되는 별도의 서버 시스템으로 구성되며 로컬서버의 구성은 (그림 2)와 같다.

인터넷상에서 사용자의 응답속도에 영향을 미치는 요소는 객체의 크기, 지리적인 위치와 네트워크 트래픽의 상태, 서버의 성능이 있다. 트래픽의 상태나 서버 자체 성능의 향

상을 통한 물리적인 요인의 개선은 상대적으로 많은 경비가 요구된다. 따라서 개별 시스템 자체의 성능향상 보다는 부하와 트래픽의 분산을 통한 성능을 향상을 위해 많은 연구가 진행되고 있다. 시스템 자체에서 발생하는 부하는 먼저 로드밸런싱을 이용하여 다수 개로 구성된 서버들을 통해 부하를 분산하게 되며, 다수개의 로컬서버로 구성된 복제 서버를 통해 시스템 전체의 부하와 네트워크 트래픽을 분산시키게 된다. 본 시스템은 다수 개의 지역서버로 구성된 계층적 구조의 전자상거래 시스템이며 트래픽의 분산과 지리적 요인으로 인한 응답 속도의 저하를 개선할 수 있다. 로컬서버는 또한 인터넷 캐싱 기능의 제공을 통해 서비스의 지연 시간을 줄일 수 있다. 캐싱은 인터넷 서버, 프락시 서버(Proxy Server), 클라이언트 브라우저와 같은 네트워크의 여러 지점에서 구현할 수 있으며 이 경우 캐싱되는 위치는 다르지만 캐싱 기법은 결국 어떻게 한정된 인터넷 캐시의 공간을 효율적으로 사용할 것인가에 대한 공통적인 문제로 접근을 하고 있다. 캐싱을 통해 서버의 부하와 전체 트래픽을 감소시키며 인터넷 사용자에게는 캐싱된 문서를 통해 빠른 응답 시간을 제공할 수 있게된다. 캐시 기능은 (그림 2)와 같은 로컬서버에서 구현이 된다.



(그림 2) Local Server

웹서버 시스템 자체에 대해서는 로드밸런싱 기능을 통해 부하의 분산이 가능하며 로컬서버(Local Server)는 (그림 1)에서 2계층의 기능을 담당하는 서버이며 (그림 2)와 같이 구성된다. 로컬 서버는 트래픽의 양과 사용자의 수에 따라 다수 개의 서버로 구성이 가능하다. 로컬서버에서 DB/File Server는 로컬서버의 각종 데이터를 저장하는 기능을 담당하며 캐시서버는 웹캐싱을 통해 반복적인 클라이언트의 요청으로 인한 응답에 대한 효율을 높일 수 있다. 다음절에서 여기에 사용되는 캐시 알고리즘을 제안하며 실험을 통해 성능을 분석한다. Real Server는 메인 서버시스템의 복제 서버로서 로컬서버에서 1차 적으로 고객에게 서비스를 직접 제공하는 역할을 담당하며, 로컬 서버에서 사용되는 Real Server는 부하의 분산을 고려하여 다수의 서버로 확장할 수 있다. Load Balancer는 Load를 다수 개로 확장된 각 Real Server에게 배분하는 기능을 담당한다.

3.2 웹 캐싱

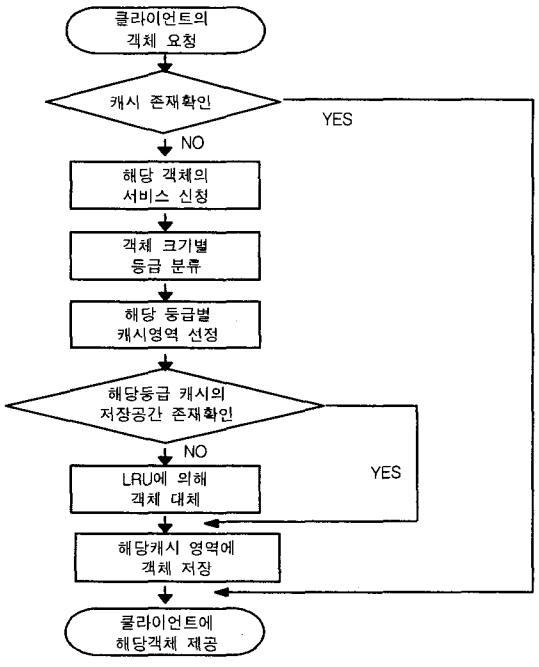
전자상거래 시스템에서 동영상을 포함한 자료가 점차 늘어나고 있으며 이에 따라 효율적인 웹캐싱을 위한 방법이 절실히 요구되고 있다. 인터넷 캐싱은 문서에 대해 발생하는 시간 지역성에 의존된다. 인터넷 객체의 크기는 매우 가변적이며 이에 따라 객체의 크기를 기반으로 한 분할된 캐시영역의 사용이 필요하다. 이 경우 캐시의 적중률을 높이기 위해서는 자주 사용되는 객체의 크기가 속한 영역의 캐시의 크기를 그렇지 않은 캐시의 영역에 비해 크게 할당하여야한다. 본 시스템에서는 캐시 영역의 분할을 위해 로그분석을 통해 인터넷 객체의 특성을 분석하였고, 사용자가 요청한 객체의 크기의 비율에 대한 산술적인 값을 구하여 그 결과 값을 활용하였다. 로그 분석 결과 100B에서 100KB까지의 객체에 대한 요청 횟수가 가장 빈번히 일어났으며, 그 중에서 1KB에서 10KB까지의 객체에 대한 요청이 가장 많았다. 이것은 실험에 사용한 전자상거래 사이트가 멀티미디어 자료를 많이 포함하지 못했기 때문이며, 멀티미디어 자료가 많은 사이트에 대해서는 그 경향 또한 다를 것으로 판단된다. 객체 크기별 요청 횟수를 기초로 하여 객체의 크기별로 총전송량과 비율을 구하면 <표 1>과 같다.

〈표 1〉 객체의 크기별 총전송량과 비율

객체 크기(KB)	요청횟수	요청횟수 비율	총전송량	총전송량 비율
0.1 이하	17,823	1.86	891.15	-
0.1~1	253,654	26.53	139,509.7	0.7
1~10	385,428	40.31	2,119,854	10.6
10~100	285,673	29.88	15,712,015	78.3
100 이상	13,567	1.19	2,102,885	10.5
TOTAL	956145	-	20,075,154.85	-

<표 1>에서 총전송량은 요청된 객체크기의 평균값에 요청 횟수를 곱한 값이며 요청횟수 비율은 전체 객체의 요청에 대한 횟수 비율을 백분율로 나타낸 것이다. 또한 총전송량 비율은 총전송량에 대한 비율을 백분율로 나타낸 것이다. <표 1>에서 보는 것과 같이 요청횟수 면에서는 1KB에서 10KB의 객체의 요청 횟수에 대한 비율이 가장 높지만, 총전송량 면에서는 10KB에서 100KB의 객체의 비율이 가장 높게 나타나고 있다. 따라서 다음과 같은 두 가지 결론을 얻을 수 있다. 첫 번째, 크기가 큰 객체(10KB 이상)보다

작은 크기의 객체(10KB 이하)에 대한 요청 횟수가 빈번하다. 두 번째, 크기가 큰 객체에 대한 적은 횟수의 요청이 한꺼번에 네트워크의 트래픽을 증가시킨다. 이 분석 결과를 바탕으로 캐시의 효율을 높이기 위해서는 다음과 같은 점을 고려해야 한다. 첫 번째 크기가 작은 객체의 비적중 증가에 의한 대체는 캐시의 효율을 떨어뜨리는 요인이 된다. 두 번째, 크기가 큰 객체의 비적중에 의한 캐시 대체는 한꺼번에 네트워크의 트래픽을 떨어뜨리는 요인이 된다. 실험을 통해 작은 크기의 객체가 가장 빈번한 요청이 일어나고 있으며, 따라서 작은 크기의 빈번한 캐시 비적중은 캐시의 객체 대체 횟수를 증가시키며 이로 인한 캐시 부하를 증가시키게 되기 때문이다. 또한 크기가 큰 객체인 경우 적은 횟수의 요청이지만 이에 대한 비적중은 용량면에서 캐시에서 큰 영역의 대체를 일으키게 하여 이로 인한 캐시대체 시간이 증가하게 되기 때문이다.



(그림 3) 캐시 대체 및 저장 알고리즘

(그림 3)은 캐시 대체 및 저장 알고리즘을 나타낸 것이다. 캐시 관리자는 캐시에 저장된 인터넷 객체의 리스트를 관리한다. 클라이언트의 객체 요청이 들어오면 캐시 관리자는 요구된 객체의 크기에 따라 분류하여 해당 캐시 영역에 존재하는 지를 확인한다. 이때 클라이언트에서 요청된 객체가 캐시 영역에 존재하여 적중이 될 경우 해당 클라이언트에 서비스를 제공하게 되며 캐시 관리자는 이용된 시간 기록을 저장하여 LRU에 의한 객체의 대체에서 높은 순위를 할당받을 수 있게 한다. 만일 적중하지 않으면 관리자는 해당 인터넷 서버에 서비스를 요청하여 해당 객체를 전송 받는다. 전송 받은 객체는 크기에 따라 해당 등급을 분류 받은 후 캐시 관리자는 해당 등급의 캐시 영역에 이 객체를 저장할 공간이 있는지를 확인한다. 저장할 공간이 있을 경우 객체를 캐시에 저장하게 되고, 없을 경우 해당 캐시 영역에 LRU에 의해 여유 공간을 배정하고 객체를 저장한다.

4. 실험결과 및 분석

실험 장비는 <표 2>와 같이 Alpha Processor가 탑재된 시스템을 기반으로 하고 있으며 실험에서와 같은 구성을 통해 성능의 향상을 예측할 수 있다.

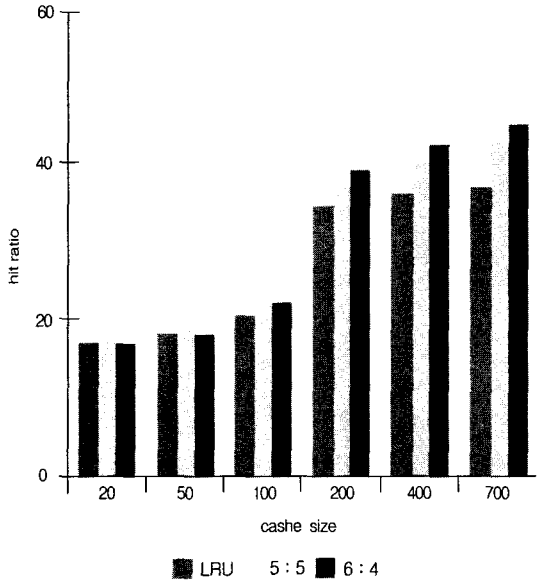
〈표 2〉 실험 모델 구성

구성요소	Number	시스템구성	비고
DB Server	S1	CPU : Alpha 21264 667Mhz×2 Memory : 1GB HDD1 : 9GB×1 HDD2 : RAID Controller, 18.2GB×4 4 Ethernet (100Mbps)	Raid 5
File Srever	S2	CPU : Alpha 21264 667Mhz×2 Memory : 1GB HDD1 : 9GB×1 HDD2 : RAID Controller, 18.2GB×4 4 Ethernet (100Mbps)	Raid 5
Backup Server	S3	CPU : Alpha 21264 667Mhz Memory : 500MB HDD1 : 9GB×1 HDD2 : RAID Controller, 18.2GB×4 4 Ethernet (100Mbps)	Raid 5
Real Server	S6, S7	CPU : Alpha 21264 600Mhz Memory : 256MB HDD1 : 9.1GB×1 2 Ethernet (100Mbps)	Raid 5
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 FastEthernet switching HUB	
Network	L15	100Mbps	

또한 실험에서 전자상거래를 위한 시스템의 구현 모델은 Red Hat Linux7 환경에서 Apache Web Server를 사용하였고 Java Script와 HTML에서 Server Script 언어로 php4, Data Base Management System은 Mysql을 사용하였다. 시스템의 설계에서 보는 것과 같이 고객 및 상품정보를 위한 DB 서버 및 File Server의 경우에는 빠른 응답속도를 요구하며 응답속도의 향상을 위해서 2개의 CPU가 탑재 가능한 시스템 모델로 설계하였다. 하지만 실제 실험에 사용된 프

로토타입에서는 데이터베이스 서버는 Alpha 21264 667Mhz 프로세서가 두 개 탑재된 서버를 사용하였고 File 서버는 Alpha 21264 600Mhz CPU와 256MB Memory 및 9.1GB×1의 하드디스크가 탑재된 모델을 사용하였다. 실험에서는 리얼 서버, 로드밸런스 또한 이와 동일한 시스템 모델로 사용하였으며 성능 평가에서 영향을 미치지 않는 메일/DNS 서버와 백업서버는 사용하지 않았다.

본 실험에서는 평균객체적중률을 통해 성능을 평가하였다. 평균객체적중률은 클라이언트에서 요청한 전체 객체의 크기에서 적중한 객체 크기 양의 비율로 구해진다. 웹캐싱에서 평균객체적중률이 높다는 것은 클라이언트의 요청에 대해 좀더 빠른 응답을 제공할 수 있게 하며, 메인시스템의 부하를 효율적으로 줄일 수 있게 한다. 또한 메인 시스템과 로컬서버 시스템과의 트래픽을 감소시킬 수 있다. 일반적으로 인터넷을 기반으로 한 전자상거래 시스템에서는 환경의 특수성으로 인해 응답 시간의 향상은 실험을 통해 예측할 수는 있으나 정확하게 응답시간을 예측하는 것은 어렵다. 이것은 전자상거래 시스템에 대한 인터넷상의 응답 속도는 객체의 크기뿐만 아니라 지리적인 위치와 환경에 따른 네트워크 상황과 시간적이고 공간적인 네트워크의 트래픽 상황에 따라 크게 영향을 받기 때문이다. 본 시스템에서는 웹캐시 기능을 가진 로컬서버 시스템의 구성을 통해 지리적이고 공간적인 영향에 의한 응답 속도의 저하를 줄일 수 있다. 또한 실험에서는 웹캐싱의 성능 평가를 위해서 캐시의 영역을 10KB 이상의 영역 1과, 10KB 이하의 영역 2로 나누었다. 또한 분할된 캐시의 영역에 (그림 4)와 같이 각각의 영역에 50%, 50%을 할당한 경우와 60%, 40%을 할당한 경우를 비교 실험하였다.



(그림 4) 적중률의 비교

실험결과는 먼저 캐시를 가득 채울 때까지 클라이언트에서 객체를 요구한 후 측정한 것이며 캐시를 채울 때까지 발생하는 캐시의 비적중은 고려하지 않은 것이다. 성능 분석에서는 클라이언트에서 요구한 객체에 대한 캐시의 적중률만을 고려하고 속도는 분석하지 않았다. 이것은 속도는 네트워크의 상태나 시간적인 트래픽 여건과 같은 환경에 의한 영향을 많이 받기 때문이다. 실험결과 LRU의 경우 캐시 공간이 늘어날수록 효율이 높아짐을 알 수 있다. 이에 따라 현재 시중에서 사용되고 있는 캐시전용 서버의 경우 저장공간의 크기가 수기가 바이트에서 수십 기가 바이트까지 다양하며 점차 용량이 커지고 있다. 실험에서는 700 메가 바이트까지의 캐시 공간만을 사용하였다. 또한 시중에서 사용되고 있는 캐시전용 서버의 적중률이 20%에서 50%정도까지 요구되는 객체에 따라 가변적이나 대부분의 경우 40%~50% 내외의 적중률을 보이고 있어 본 실험에 사용된 캐시 또한 그 효용성이 있어 보인다. 적은 용량의 캐시를 할당한 경우 LRU나 5 : 5, 6 : 4로 할당된 캐시의 적중률이 거의 차이가 없음을 알 수 있다. 용량이 크질수록 LRU 방법에 비해 분할된 캐시방법이 3%~10% 정도의 적중률 향상이 있는 것으로 나타난다. 또한 본 실험은 트래픽의 여건과 성격에 따라 많은 차이가 있을 것으로 보인다. 예를 들어 mp3 파일이나, 각종 멀티미디어 자료의 빈번한 요청과 같은 사용자의 선호도에 따른 집중적인 요청은 객체의 종류는 상기 비교 방법들의 성능에 큰 영향을 미칠 것이다. 사이즈가 큰 객체의 비적중으로 인한 객체의 대체에 소요되는 시간은 사이즈가 작은 객체의 대체에 걸리는 시간에 비해 클라이언트의 응답 시간을 지연시키는 요인이 된다. 따라서 동영상 자료와 같이 용량이 큰 객체를 많이 포함한 경우 5 : 5보다는 6 : 4 할당에 의한 캐시의 분할이 좀 더 효율적일 것이며 객체 요청에 따른 캐시의 분할에 대한 추가적인 연구가 필요하다. 또한 전자상거래 시스템에 대한 객체의 요청은 고객의 선호도를 분석을 통해 집중적으로 요청되는 객체의 분석이 가능하다. 따라서 이러한 고객의 선호도를 캐시 알고리즘에 고려한다면 제안한 캐시의 효율성을 더욱 높일 수 있을 것이다.

5. 결 론

본 연구에서는 전자상거래 시스템의 효율을 높이기 위한 시스템을 제안하고 그 효용성을 실험을 통해 보였다. 제안한 시스템은 시스템의 지리적인 요인으로 인한 응답속도의 저하를 줄이기 위해 로컬서버를 이용한 계층적인 구조를 기반으로 하고 있으며, 객체의 크기로 인한 지연속도를 개선하기 위해 사용자 요청의 특성을 고려한 분할 영역 기반

의 웹캐시 알고리즘을 제안하였다. 실험 결과 본 연구에서 제안한 웹캐시 알고리즘이 기존의 LRU 기반의 캐시에 비해 적중률이 향상되어 클라이언트의 다양한 요구에 대해 개선된 응답 속도와 빠른 적응성을 가질 수 있으며, 특히 전자상거래 시스템에 대해 동시에 발생하는 다수의 다중 사용자의 요구로 인해 발생되는 응답 속도의 저하를 막을 수 있을 것으로 기대된다.

본 논문에서 실험한 시스템은 로컬서버의 구성에 필요한 경비의 추가적인 소요가 발생하지만, 기존 시스템에서 발생하는 서버의 성능 증가를 위한 교체비용과 네트워크의 성능 향상에 요구되는 비용의 Trade off가 가능하다. 향후 본 연구에서 제안한 시스템을 실제 전자상거래 현장에서의 활용을 통해 시스템 적응성에 대한 검증이 필요하며 전자상거래를 위한 지능형 에이전트 구조에 대한 응용을 통해 사용자 중심의 시스템 성능 향상이 필요할 것이다. 또한 전자상거래 시스템에 대한 객체의 요청은 고객의 선호도를 분석을 통해 집중적으로 요청되는 객체의 분석이 가능하다. 따라서 이러한 고객의 선호도를 캐시 알고리즘에 고려한다면 제안한 캐시의 효율성을 더욱 높일 수 있을 것이다.

참 고 문 헌

[1] Soe-Tsyr Yuan, A. Liu, "Next-generation Agent-enabled Comparison Shopping," Expert Systems with Applications 18, pp.283-297, 2000.

[2] Elizabeth A. Kendall, "Role Modeling for Agent System Analysis, Design, and Implementation," IEEE Concurrency, Vol.8, No.2, April-June, 2000.

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

[4] Cardellini V., Colajanni M., Yu P. S., "Redirection Algorithms for Load Sharing in Distributed Web-server Systems," Distributed Computing Systems, 1999. Proceedings, 19th IEEE International Conference, pp.528-523, 1999.

[5] S. Williams, M. Abrams, C. R. Standridge, G. Abhulla, and E. A. Fox, "Removal Policies in Network Caches for World Wide Web Documents," Proc. ACM SIGCOMM, pp.293-304, 1996.

[6] Jian Liu, Longlu Xu, Baogen Gu, Jing Zhang, "A Scalable High Performance Internet Cluster Server," High Performance Computing in the Asia-Pacific Region 2000. Proceedings, The Fourth International Conference/Exhibition, Vol.2, pp.941-944, 2000.

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

[8] D. Dias, W. Kish, R. Mukherjee and R. Tewari, "A Scalable and Highly Available Server," COM-RCON 1996, pp.85-92, 1996.

[9] Chad Yoshikawa, et. als., "Using Smart Clients to Build Scalable Services," http://now.cs.berkeley.edu, USENIX'97, 1997.

[10] Robert L. Carter, Mark E. Crovella, "Dynamic Server Selection Using Bandwidth Probing in Wide Area Net-works," http://www.ncstrl.org, Boston University Technical Report, 1996.

[11] Wensong Zhang, Shiyao Jin, Quanyuan Wu, National Laboratory for Parallel & Distributed Processing, "Creating Linux Virtual Servers," http://proxy.iinchina.net/~wensong/ppfvs/linuxexpo.html.

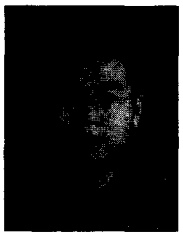
[12] Dongeun Kim, Cheol Ho Park, Daeyeon Park, "Request Rate Adaptive Dispatching Architecture for Scalable Internet Server," Cluster Computing, 2000. Proceedings, IEEE International Conference, pp.289-296, 2000.

[13] Daniel Anderson, Tao Yang, Veard Holmedahl, and Oscar H. Ibarram, "SWEB : Towards a Scalable World Wide Web Server On Multicomputers," Proceedings of International Conference on Parallel Processing, pp.15-19, April, 1996.

[14] Cisco System, "Cisco Local Director," http://www.cisco.com/warp/public/751/lodir/index.html.

[15] Kangasharju J., Ross K. W., "A Replicated Architecture for the Domain Name System," INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, Vol.2, pp.660-669, 2000.

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



윤 용 기

e-mail : yongki@yonsei.ac.kr

1994년~2000년 전)대우전자 주임연구원

1992년 경북대학교 전자공학(공학사)

1995년 경북대학교 전자공학(공학석사)

2001년 전)기업정보화지원센터연구원

2001년~2002년 전)KOREC(주) IT기획/전략본부장

현재 연세대학교 박사과정(기술경영학협동과정)

관심분야 : 비즈니스 프로세스 표준화, 전자상거래 시스템, ISPM

고 일 석

e-mail : isko@ctech.ac.kr
경북대학교 전자계산기공학전공(공학사)
경북대학교 컴퓨터공학과(공학석사)
2000년 성균관대학교 경영대학원 앤소프 전략스쿨 수료
2000년 미)USIU 경영학과(MBA, Ansoff Strategic Management전공)
연세대학교 컴퓨터산업시스템공학과 박사 수료
현재 충북과학대학 전자상거래과 교수
관심분야 : 전자상거래 시스템, 에이전트기반시스템, CBD

임 춘 성

e-mail : leem@yonsei.ac.kr
1985년 서울대학교 산업공학과(공학사)
1987년 서울대학교 산업공학과(공학석사)
1992년 미)UC Berkeley 산업공학과(공학박사)
1993년~1995년 미)Rutgers University 산업공학과 조교수
1995년~현재 연세대학교 정보산업전공 부교수
1997년~현재 기업정보화지원센터장
관심분야 : 전자상거래, 기업정보화방법론, 기업정보시스템평가