

# 전자상거래에서 메모리 상주 DBMS를 이용한 웹 데이터베이스의 성능 향상 방안

공희경<sup>1</sup>, 장인기<sup>2</sup>, 이현우<sup>1</sup>, 조완섭<sup>3</sup>, 조현성<sup>4</sup>

충북대학교 정보산업공학과<sup>1</sup>

충북대학교 전자계산학과<sup>2</sup>

충북대학교 경영정보학과<sup>3</sup>

한국전자통신연구원<sup>4</sup>

[khkong@just.chungbuk.ac.kr](mailto:khkong@just.chungbuk.ac.kr), [cig2005@shinbiro.com](mailto:cig2005@shinbiro.com).

[leehw72@just.chungbuk.ac.kr](mailto:leehw72@just.chungbuk.ac.kr)

[wscho@trut.chungbuk.ac.kr](mailto:wscho@trut.chungbuk.ac.kr), [hsc@etri.re.kr](mailto:hsc@etri.re.kr)

**요약** 본 논문에서는 인터넷을 이용한 전자상거래에서 메모리 상주 DBMS를 이용하여 사용자 수가 증가하는 경우에도 최종 사용자에게 대한 응답 시간을 빠르게 유지하는 새로운 캐싱 기법을 제안한다. 지금까지 인터넷 서버 시스템 구조의 성능에 관한 연구에서는 멀티-티어(multi-tier)시스템 구조가 제안되었으나 메모리 상주 DBMS를 이용한 새로운 서버 시스템 구조의 연구는 시작단계이다. 본 연구에서는 전자상거래에서 최종 사용자에게 빠른 응답시간을 보장하기 위해 전자상거래용 웹 데이터베이스에서 다수의 사용자가 공통으로 빈번하게 사용하는 hot data를 메모리 상주 DBMS가 관리하도록 하는 성능 향상 방안을 제시한다.

## I. 서론

최근 발전하는 웹 기술과 사용하기 편리한 브라우저의 출현으로 인터넷 사용이 급격히 증가하면서 인터넷을 비즈니스 측면에서 활용하고자 하는 노력이 계속되고 있으며, 그 일환으로 전자 상점(Electronic Mall)을 이용한 전자상거래가 점차 확산되고 있다. 빠르게 발전하는 통신망과 데이터베이스 등의 정보기술과 24시간 항상 접속 가능한 개방된 인터넷 환경을 고려해 본다면 전자상거래의 출현은 당연한 결과라고도 할 수 있다. 인터넷의 확산으로 인해 전자상점을 이용한 전자상거래가 현재 기업과 개인간 거래뿐 아니라 기업 대 기업간의 거래에서도 빠르게 확산 될 것으로 예상된다.

이러한 전자상거래의 급속한 보급과 이

에 따른 경쟁의 심화에 따라 전자상거래 시스템은

인터넷으로 접속한 최종 사용자에게 빠른 응답시간을 보장하는 것이 전자상거래 시스템 연구의 중요한 이슈가 되고 있다[1,2,3,8]. 인터넷을 통하여 한꺼번에 많은 사용자가 접속하는 경우 통신망이나, 서버 시스템 구조 및 데이터 관리 기법 등 여러 가지 이유로 급격한 응답 시간 지연 현상이 발생하며, 이로 인한 고객 이탈 현상이 심각한 문제로 대두되고 있기 때문이다. 이러한 성능 저하 문제를 개선하기 위하여 통신망의 고속화, 인터넷 서버 시스템 구조 개선을 통한 성능 향상, 고성능의 데이터베이스 관리 기법 등의 연구가 진행되고 있으나[1,2,3,4,5,6] 이들 중 인터넷 서버 시스템 구조 개선을 통한 성능 향상은 중요한 분야임에도 불구하고 별다른 연구결과가 없는

상황이다.

이 논문에서는 메모리 상주 캐싱 기법을 이용하여 인터넷 서버 시스템의 구조를 개선하고, 그 결과 서버 시스템의 성능을 향상시키는 방안을 제안하고자 한다. 또한 메모리 상주 DBMS(MMDB:main memory database manager system)가 전자상거래용 데이터베이스에서 다수의 사용자가 공통으로 빈번하게 사용하는 hot data를 관리하는 서버 시스템 구조를 제안한다[1]. 메모리에 캐싱되는 hot data를 메모리 상주 DBMS를 사용하여 관리함으로써 여러 응용에서 이들 데이터를 공유할 수 있으며, 캐쉬된 데이터가 디스크에 있는 원본 데이터와 항상 일치하도록 보장함으로써 기존의 화일 시스템 기반 캐싱 기법과는 차별화된 기능을 제공할 수 있다.

이 논문의 구성은 다음과 같다. 제 2장에서는 메모리 상주 DBMS를 이용하여 시스템의 성능을 향상시키는 새로운 인터넷 서버 시스템 구조를 제시한다. 제 3장에서는 메모리 상주 DBMS를 이용하여 메모리에 캐싱되는 hot data를 관리하는 기법을 설명한다. 제 4장에서 결론 및 향후 연구 과제를 기술한다.

## II. 메모리 상주 DBMS를 이용한 인터넷 서버 시스템 구조

인터넷 전자상거래 시스템에서 대용량 자료를 관리하기 위하여 웹 데이터베이스의 사용이 필수적이다. 웹 데이터베이스를 관리하는 DBMS로서는 기존의 디스크 기반 DBMS와 메모리 상주 DBMS가 활용될 수 있다. 여기서는 디스크 기반 DBMS 만을 활용하는 경우의 문제점을 살펴보고, 메모리 상주 DBMS를 이용한 개선방안을 논의한다.

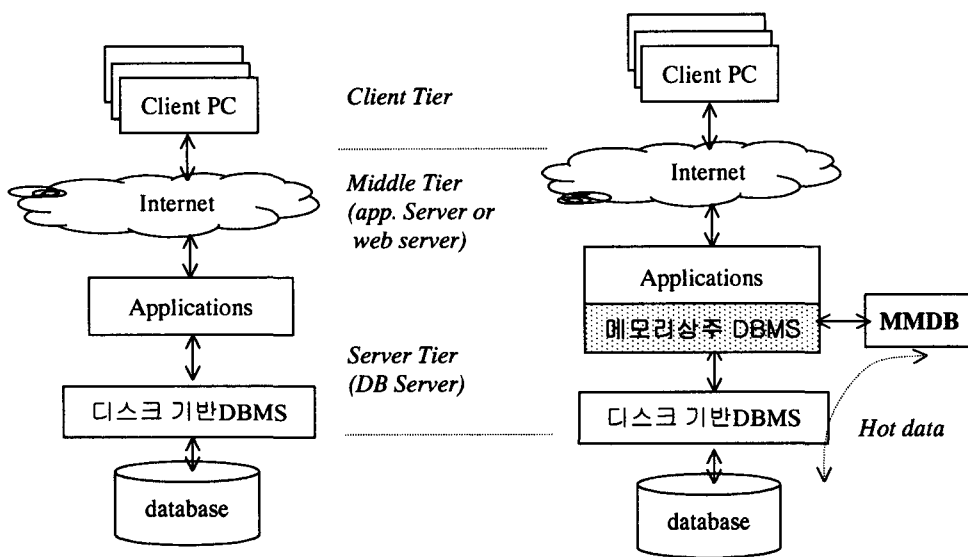
인터넷의 보편화로 전자상거래 환경에서 고객의 요청을 신속하게 처리하는 요구가 증대되고 있다. 이러한 환경에서 기존의 디스크 기반 DBMS를 사용하여 웹 데이터베이스를 관리하면 디스크 접근 부분에서 병목 현상이 발생하여 서버 시스템의 성능이 저하될 수 있

다. 더구나 최종 사용자의 요구에 대한 응답 시간이 중요한 경쟁 요소로 되고 있는 전자상거래 환경에서는 이러한 성능의 저하가 심각한 결과를 초래할 수 있다. 인터넷 환경의 전자상거래에서 주 기억 장치의 용량을 최대한 활용하여 사용자가 공통으로 빈번히 요구하는 hot data를 메모리에 상주시켜 웹 데이터베이스의 성능을 향상시키는 인터넷 서버 시스템의 구조를 개선하는 것이 중요하다.

그림1 (a)는 웹 데이터베이스에서 디스크 기반 DBMS 만을 사용하는 서버 시스템 구조를 보여준다. 반면 그림1 (b)는 메모리 상주 DBMS 시스템과 디스크 기반 DBMS를 모두 사용하는 시스템 구조를 보여주고 있다.

그림1 (a)와 같이 기존의 멀티 티어(multi-tier)구조와 디스크 기반 DBMS를 사용하는 서버 시스템에서는 인터넷 사용자의 요구로 발생한 웹 트랜잭션 수행 시 모든 트랜잭션 처리마다 디스크 I/O(input/output)가 발생하므로 큰 오버헤드를 발생하게 된다. 그러나 본 논문에서 제안된 그림1 (b)의 메모리 상주 DBMS 및 디스크 기반 DBMS를 모두 사용하는 인터넷 서버 시스템 구조에서는 기존의 디스크 기반 DBMS와 함께 어플리케이션(application) 단계에 메모리 상주 DBMS 구조를 사용함으로써 사용자의 요구가 자주 발생하는 트랜잭션에 대한 data를 메모리 상주 DBMS에서 관리하므로 디스크 I/O에 대한 오버헤드를 줄일 수 있다. 따라서 제안된 인터넷 서버 시스템 구조는 전자상거래에서 사용자 수가 증가하여 발생하는 디스크 접근 부분에서의 병목 현상으로 인한 서버 시스템의 성능 저하를 개선하게 된다.

이 논문에서 제안된 인터넷 서버 시스템의 구조는 기존의 디스크 기반 DBMS 만을 사용하는 경우와 달리 메모리 상주 DBMS를 추가시킴으로써 사용자의 요구에 의한 데이터 관리 수행을 효율적으로 하여 성능을 향상시킬 수 있다. 전자상거래에서 자주 발생하는 트랜잭션에 대한 데이터를 메모리 상주 DBMS 단계에서 관리함으로써 사용자의 요구에 신속하게 응답할 수 있다. 이로 인해 전자



상거래에서 자주 발생하는 구매, 지불, 제품정보와 장바구니 등의 정보(hot data)에 관한 관리를 메모리 상주 DBMS가 효율적으로 처리

시스템 구조를 가정한다. 사용자는 웹 페이지를 이용하여 전자 쇼핑몰에서 전시된 상품에 대하여 검색, 구매, 주문, 지불, 기타 서비스 등

(a)디스크 기반 DBMS 만을 사용하는 경우

(b)디스크 기반 DBMS와 메모리 상주 DBMS를 모두 사용하는 경우

[그림-1] 웹 데이터베이스에서 서버 시스템의 구조

할 수 있도록 하여 디스크 출입 없이 빠른 응답시간을 제공할 수 있게 된다. 따라서 제안된 메모리 상주 DBMS 응용 방식에서는 사용자가 빈번히 요구하는 hot data의 캐싱과, 관리가 중요하게 요구되어 진다.

을 요구한다. 머천트 서버는 상품을 구매하려는 고객의 요구 처리를 담당하는 서버 시스템 모듈이다. 머천트 서버에서는 전자 상거래용 웹 데이터베이스에 저장된 데이터를 검색하고, 변경하고, 관리하는 database interface를 가진다. cache manager와 hot data를 관리하는 MMDB, 그리고 디스크 DB와 메모리 상주 DB간의 update 명령을 관리하는 refresh manager가 존재한다. 그리고 DB 서버 단계에서 전체 data를 저장하고 관리하는 디스크 DB가 존재한다.

### III. 메모리 상주 DBMS를 이용한 웹 데이터베이스 캐싱 기법

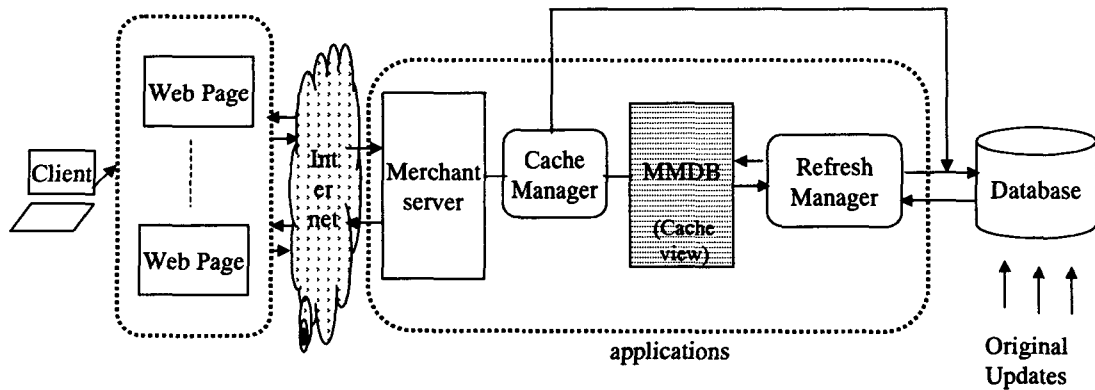
이 논문은 전자상거래용 응용들이 디스크 데이터베이스 대신에 메모리 상주 데이터베이스를 이용함으로써 사용자에게 대한 응답시간을 줄이는 데 초점을 맞춘다.

이 장에서는 웹 데이터 베이스 관리에서 메모리 상주 DBMS를 이용한 웹 데이터베이스 캐싱 기법과, 메모리 상주 DBMS 이용을 위해 필요한 각 모듈을 설명하고, 메모리에 캐싱되는 hot data를 메모리 상주 DBMS가 관리하는 기법을 설명한다.

그림 2의 전자상거래 시스템은 다음과 같이 동작한다. 인터넷을 통한 전자상거래 트랜잭션의 결과로 머천트 서버에서 발생한 database request에 대하여 cache manager는 요구되는 data가 디스크 DB에 저장된 자료인지 메모리 상주 DB인지를 구분한다. 자료의

#### 1. 전자상거래 시스템 구조

본 연구에서는 그림 2와 같은 전자상거래 시



위치가 확인되면 cache manager는 MMDB 혹은 disk DB에 접근하여 자료를 검색하여 client에게 전달한다. 이 과정은 cache

타 서비스를 담당하는 트랜잭션 처리 등을 담당한다.

[그림-2] 전자상거래 시스템 구조

manager가 별도의 시스템 카탈로그 정보를 가지거나 혹은 SQL에 대하여 cache view 개념을 갖도록 확장한 후 파싱할 때 위치를 인식할 수 있도록 하는 두 가지 방식이 사용될 수 있다. 여기서는 후자를 가정한다. 트랜잭션 처리에서 발생한 데이터베이스의 update 명령은 refresh manager를 통하여 base database와 MMDB간의 일치성 문제를 해결하게 된다 [9]. 이 경우 MMDB에 있는 자료를 변경하는 경우 먼저 디스크 기반 DB에서 해당 자료를 update 한 후 성공한 경우에만 MMDB에서 update 명령을 실행한다.

전자상거래 시스템 구조의 각 모듈에 대한 자세한 설명은 다음과 같다.

### 1.1 Client

전자상거래를 통해 상품을 검색하고 구매하며, 주문, 지불등의 기타 서비스를 요청하는 사용자가 사용하는 PC의 웹 브라우저를 통해 머천트 서버에 접근한다.

### 1.2 Merchant Server

웹 환경에서 상품을 구매하려는 시도의 시작 부분부터 최종 지불 완료시점까지의 각각의 사용자 요청을 관리하는 전자상거래 솔루션으로 고객의 상품 구매 요구처리를 담당하는 모듈이다. 머천트 서버에서는 상품 및 카탈로그 관리, 주문 및 온라인 지불 처리, 검색 및 기

### 1.3 Cache Manager

Cache Manager는 MMDB에서의 cache view의 생성, 관리, 삭제 기능을 담당한다. cache view는 디스크 기반 DB에 대하여 정의되는 실체화된 뷰(Materialized view)로써 MMDB에 table형태로 저장된다. 또한 인터넷을 통해 머천트 서버에서 발생한 database request를 분석하여 disk DBMS에서 처리할 것인지 메모리 상주 DBMS에서 처리할 것인지를 결정한다. 트랜잭션이 요구한 data의 저장위치를 찾는 과정은 cache manager용 catalog를 이용하거나 SQL 내에서 cache view에 특별한 표기를 한 후 파서가 인식하도록 할수 있다. 이때 cache manager용 catalog는 MMDB에 table로 구현되며, MMDB에 의하여 동시성 제어가 제공되므로 cache view는 다수의 응용에서 공유될 수 있다.

### 1.4 MMDB(Main Memory DataBase)

현재 널리 사용되고 있는 디스크 기반 데이터베이스 시스템(disk resident database system)에서는 데이터를 다루기 위한 디스크 액세스의 오버헤드가 지나치게 크므로[10] 빠른 처리 속도를 요구하는 응용에 적합하지 않다. 해결책으로 메모리 상주 DBMS가 연구되어 상용 제품들이 보급되고 있다[1]. 메모리 상에 데이터베이스를 상주시키는 방법을 전자상거래 internet computing architecture에 적

용시킴으로써 빠른 응답시간을 요구하는 트랜잭션처리를 효율적으로 관리할 수 있다. 사용자들이 자주 사용하는 hot data를 application 내에 MMDB에 따로 관리함으로써 사용자의 요구에 대한 머친트 서버의 응답시간을 줄여주는 역할을 한다. MMDB에서는 hot data를 cache view 형식으로 저장하여 관리하게 된다.

### 1.5 Refresh Manager

disk database와 메모리 상주 database와의 consistency를 유지시켜주는 기능으로 base data에 original updates가 발생하면 MMDBMS의 해당 data도 동일하게 변경해서 MMDB의 cache view가 최신 버전임을 보장해야 한다. 즉, 여러 사용자가 접속하며, 여러 트랜잭션이 발생하는 상황에서 다른 사람이 cache data를 변경하는 즉시 나머지 사람들이 변경된 data를 볼 수 있도록 해야 한다. 또한 database와 MMDB간의 refresh에 따른 부담을 최소화해야 하며 Push 기법과 Pull 기법을 적절하게 사용해야 한다[9].

### 1.6 Base Data

일반적으로 전자상거래를 지원하는 머친트 서버에서 사용되는 데이터베이스로 전자상거래 템플릿(template) 정보와 product catalog 정보 등의 static한 정보로 자주 업데이트 되지 않는 정보와 전자상거래 session 정보나 고객 정보 등 dynamic한 정보가 함께 저장되고 업데이트되는 데이터베이스이다. 전자상거래 거래 발생 시 웹 트랜잭션의 생성으로 인해 disk database가 original update되면 refresh manager를 통해 MMDBMS에 존재하는 data와도 일치시켜야 한다.

## 2. 웹 데이터 캐싱 기법

지금까지의 웹 데이터 캐싱 기법은 대부분 캐쉬 데이터를 클라이언트의 디스크에 위치시키는 방식이나[2,3,4,5,6], 최근 들어 메모리에 캐쉬 데이터를 상주시키는 메모리 상주 방식[1,8]이 제안되고 있다. 메모리 상주 방식의 경

우 메모리 상주 DBMS를 사용하는 방식과 그렇지 않은 방식으로 나뉘어진다.

기존의 메모리 상주 DBMS를 사용하지 않는 메모리 caching 기법의 문제점은 caching data에 대한 정교한 DBMS 기능이 제공되지 않으며, 각 어플리케이션마다 자신이 요구하는 data를 별도로 캐싱하고, 캐쉬된 data는 여러 어플리케이션에서 동시에 공유할 수 없다는 한계가 존재한다. 동일한 자료가 중복 캐쉬되어 시스템 자원의 낭비가 초래되고, 자료 변경시 일치성 유지가 매우 어렵게 되는 경우가 발생한다.

전자상거래 환경에서 캐싱되는 웹 데이터는 시간에 따라 변화하며, 머친트 서버를 이용하는 사용자마다 달라질 수 있기 때문에 DB에서 여러 가지 요인에 따라서 수시로 변화하는 특징을 갖고 있다. 예를 들면 고객정보나 상품 정보 등의 전자상거래에서 중요하면서도 빈번히 발생하는 트랜잭션에 속하는 데이터들이다. 본 논문에서 제안하는 웹 데이터 캐싱 기법은 웹 트랜잭션이 발생하면 cache manager를 통하여 DB 스키마(schema)의 일부를 cache view로 지정하여 MMDB에 cache manager를 위한 system catalog에 view형식으로 저장하는 기법이다. 이러한 캐싱에 요구되는 기술적인 문제점은 캐쉬된 데이터가 항상 최신 버전인 것을 보장해야 하며, 임의의 사용자가 또 다른 사용자가 사용하는 캐쉬된 data를 변경하는 즉시 변경된 새로운 버전의 data를 볼 수 있어야 하고, MMDB에 존재하는 cache view를 drop하는 경우 그것을 참조하는 reference count가 0인 경우에만 drop시킬 수 있도록 해야 한다.

그림 3은 cache manager가 cache view를 생성하고, 이용하는 예를 보여주고 있다.

- (가) Create cache view cv1 as SQL1;
- Drop cache view cv1;
- (나) Select \* from cv1 where ... ;
- (다) Update ... from cv1 ;
- (라) Insert ... into cv1 ;
- (마) Delete ... from cv1 ;

### [그림 3] cache view의 생성과 이용

위의 SQL문에서 그림 3의 (가)는 cache view cv1을 생성하는 문장으로서 system catalog에 cv1(create view)에 관한 정보를 저장한 후 disk DB에 SQL을 보내서 결과를 얻고, 이를 MMDB에 저장한 뒤 웹 트랜잭션 발생으로 인한 변경 가능성을 판정하게 된다. (나)는 파싱 후 cv1이 MMDB에 있는 cached view라고 판정되면 query를 MMDBMS에서 처리하게 된다. (다)~(마)는 cache view에 대해 update, insert, delete 가능성을 검사한 후, disk DB에서 update commit 명령이 수행된 후에 MMDB를 update하게 된다. 이 때 cache manager가 사용자가 요청한 트랜잭션에 대해 관련된 데이터의 위치를 system catalog 정보를 사용하여 저장된 위치를 결정하고 질문을 보내어 처리하도록 한다.

## IV. 결론 및 향후 연구

본 논문에서는 전자상거래 환경에서 새로운 인터넷 서버 시스템 구조를 제안하고, 전자상거래 시스템 구조에서 메모리 상주 DBMS를 이용하여 웹 트랜잭션 처리의 성능을 개선하는 기법을 제안하였으며, 디스크 데이터베이스의 일부를 메모리에 캐싱하여 관리하는 방안을 제시하였다. 메모리 상주 DBMS를 이용한 웹 캐싱 기법은 향후 인터넷을 통한 웹 응용의 폭발적인 증가와 더불어 성능 향상을 위하여 필연적으로 요구되는 웹 서버 기술이다. 인터넷의 발전과 더불어 새로운 형태의 전자상거래 솔루션이 창출되고 있는 가운데, 대규모 사용자의 접속에서도 적정 수준의 응답 시간을 보장해 줄 수 있는 전자상거래 시스템의 성능 향상 기술의 개발이 중요한 관건이라 보여진다. 특히 이러한 환경에서 응용 서버나 웹 서버가 존재하는 중간계층(middle tier)에 메모리 상주 DBMS를 사용한 캐싱 기법을 활용함으로써 성능을 향상시키는 요구는 더욱 증가할 것으로 예상된다.

앞으로 메모리 상주 DBMS를 이용한 웹

데이터 캐싱 기술의 연구는 실제 프로토타입 시스템의 구축과 이를 통한 성능 평가 및 성능 조율에 관한 문제를 다룰 것이다.

## 참고 문헌

- [1] TimesTen, White Paper, <http://www.timesten.com>.
- [2] Dongwon Lee, Wesley W.Chu, "A Semantic Caching Scheme for Wrappers in Web Databases," In *Proc. Int'l Conf. on International Conference on Information and Knowledge Management*, Kansas City, Missouri, USA, pp. 77-85, Nov.1999.
- [3] Dongwon Lee, Wesley W. Chu, "Conjunctive Point Predicate-based Semantic Caching for Wrappers in Web Databases," In *Proc. Int'l Conf. Workshop on Web Information and Data Management*, Washington, DC, USA, November 1998.
- [4] Shaul Dar, Michael J.Franklin, Bjorn T.Jonsson, Divesh Srivastava, Michael Tan, "Semantic Data Caching and Replacement," In *Proc. Int'l journal on Very Large Data Bases* Mumbai(Bombay), India, pp. 330-341 Sept. 1996.
- [5] Michael Franklin, Stan Zdonik, "Data In Your Face: Push Technology in Perspective," In *Proc. Int'l Conf. on Management of Data*, ACM SIGMOD, pp. 516-519, June. 1998.
- [6] Michael J. Franklin, Michael J. Carey, Miron Livny, "Transactional Client-Server Cache Consistency: Alternatives and Performance," *ACM Trans. on Database System*, vol. 22 No.

3, pp. 315-363, Sept. 1997.

- [7] Swarup Acharya, Michael Franklin, Stanley Zdonik, "Balancing Push and Pull for Data Broadcast," In *Proc. Int'l Conf. on Management of Data*, ACM SIGMOD Arizona, pp. 315-363, May. 1997.
  
- [8] Evangelos P. Markatos, "Main Memory Caching of Web Documents," In *Proc. Int'l Conf. on Second International World-Wide Web Conference*, Chicago, Illinois, USA, Vol. 28 No.(7-11) pp, 893-905, May 1996.
  
- [9] Haifeng Liu, Wee-Keong NG, Ee-Peng Lim, "Mode and Research Issues for Refreshing A Very Large Website," In *Proc. Int'l Conf. on Web information System Engineering*, 2000.
  
- [10] 황규영, 장지웅, 이영구, 김원영, "주기억 장치 데이터베이스를 위한 저장 시스템," 정보과학회지 제 14 권 제 2 호 1996.2