

工學碩士 學位論文

Java2 플랫폼 기반의 JDBC를 이용한 대규모
웹사이트 갱신시스템 설계 및 구현

指導教授 曹 佑 鉉

이 論文을 工學碩士 學位論文으로 提出함



2002년 2월

釜慶大學校 産業大學院

컴 퓨 터 공 학 과

金 用 洙

이 論文을 金用洙의 工學碩士
學位論文으로 認准함

2001년 12월 15일

主 審 工學博士 鄭 木 童



委 員 工學博士 徐 庚 龍



委 員 工學博士 曹 佑 鉉



목 차

Abstract	iv
제 1 장 서 론	1
제 2 장 관련연구	4
2.1 WWW 및 웹 기반의 기술-JAVA2	4
2.2 갱신시스템 모델	6
2.3 데이터베이스 연동	9
2.3.1 JDBC(Java Database Connectivity)-JAVA	9
2.3.2 ADO(ActiveX Data Object)-ASP	13
제 3 장 웹사이트 갱신시스템의 설계 및 구현	14
3.1 전체 시스템 설계	14
3.2 가상 DBMS의 분석	18
3.3 서버 시스템 구축	19
3.4 데이터베이스 설계 및 구현	20
3.4.1 시스템 모듈 구성	21
3.4.2 데이터베이스 테이블	29
제 4 장 웹사이트 갱신시스템 모의 실험	32
4.1 실험방법 및 환경	32
4.2 실험결과 및 평가	36
제 5 장 결론 및 향후과제	38
참고문헌	39

그 림 목 차

그림 1-1 국내 웹 페이지 구성 비율	1
그림 2-1 JAVA2 플랫폼 전체 구성	5
그림 2-2 일반적인 웹사이트 형태	7
그림 2-3 자동-동적 웹페이지	7
그림 2-4 JDBC에서의 데이터베이스 통신경로	11
그림 2-5 JDBC 드라이브 이용 환경	12
그림 2-6 ADO에서의 데이터베이스 이용환경	13
그림 3-1 3-Tire 클라이언트 서버 모듈	14
그림 3-2 JAVA2 플랫폼 기반의 갱신시스템	15
그림 3-3 갱신시스템 프로토콜(Protocol)	17
그림 3-4 웹서버 구성 및 연결도	20
그림 3-5 시스템별 모듈 동작	21
그림 3-6 사용자 시스템 모듈 코드내용	23
그림 3-7 서버시스템 모듈 코드내용	27
그림 3-8 갱신시스템 모듈 코드내용	28
그림 4-1 갱신시스템 모의실험용 웹페이지 화면	33
그림 4-2 동적 웹페이지 경매물품현황 실행 화면	33
그림 4-3 정적 웹페이지 신규종목등록 실행 화면	34
그림 4-4 정적 웹페이지 경매신청하기 응용 실행 화면	35
그림 4-5 실시간 조회 화면	35
그림 4-6 사용자 증가(스레드)에 따른 시스템 응답시간 측정	36

표 목 차

표 3-1 웹서버와 웹 데이터베이스서버의 환경	19
표 3-2 경매 테이블 구조	30
표 3-3 경매 기록테이블 구조	31
표 4-1 모의 실험에 사용된 시스템 사양	32

Design and Implementation of System for Refreshing Very Large Website using JDBC based on Java2 PlatForm

Yong-Soo Kim

Department of Computer Engineering
Graduate School of Industry
Pukyong National University

Abstract

The Internet Web sites have increased the contents and the numbers of pages about more and more basic data by virtue of the rapid development of the Internet which is shared with every information. Accordingly, the numbers of dynamic pages are quickly increasing which read and transact a lot of information from a database system. In this case, it is very important to handle and transfer the most recent information very fast and efficiently. In this thesis, we designed and implemented the most proper refreshing system that transact a real time information of a database system in Web sites. We applied this system to a specific model and simulated the results. This application based on basic data refreshing system was established with ASP using ORACLE 8.0.1 and Windows 2000 Server. Also the module of the user and a refreshing system was implemented by JDBC based on Java2. The optimum Web refreshing system is expected to play a important role and provide the users with enough satisfaction.

제 1 장 서론

우리가 살고 있는 정보화시대에서 우리는 여러 매체들을 통해서 통제할 수 없을 만큼 많은 정보를 쉽게 얻을 수 있다. 그리고, 이러한 정보들을 얼마나 잘 획득하고 사용하느냐에 따라 개인의 삶의 질이 달라질 수 있는 시대이다. 이러한 시대의 인터넷 열풍은 우리의 생활에서 여러 가지 것들을 획기적으로 바꾸어 놓았으며, 그 변화는 사회의 한 부분에서의 변화가 아닌 사회 전반에 걸친 커다란 변화를 가져왔다고 할 수 있다. 1990년대 중반부터 대중화 된 국내 인터넷 이용자 수는 2001년 6월말 현재 2,223만명(이용률 51.6%)에 이른다.

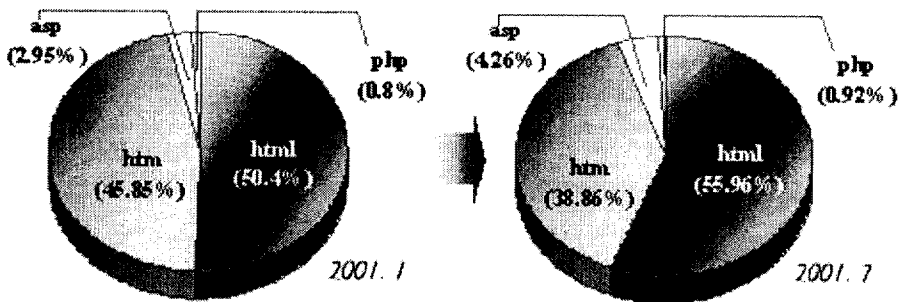


그림 1-1 국내 웹 페이지 구성비율

최근에는 웹사이트를 통한 기업과 개인들의 자체 광고가 점점 늘어가고 있다. 국내 최초로 산출 제시한 웹 관련 통계치를 살펴보면, 7월말까지 수집된 국내 웹 페이지 수는 17,782,299개(html, htm, asp, php 포함)이며, 그 중 정적인 웹 페이지(Static HTML)문서가 55.96%를 차지 하였다[1][2].

급속한 인터넷의 발전은 기존의 정보검색 서비스와는 또 다른, 그림과 동화상, 소리 등을 모두 지원할 수 있는 멀티미디어 정보검색 서비스와 동

적인 웹페이지(Dynamic HTML)로 바뀌어 가고 있다.

이러한 특성은 기본 데이터와 페이지수가 많아짐으로 인해 자료를 데이터베이스화하여 관리하고, 사용자가 웹 상에서 많은 자료요청을 계속적으로 요구함으로써 나타나고 있다. 최신 자료에 대한 정보전달의 문제와 데이터베이스의 부하가 생기는 것이 웹 서비스의 한계이다. 이러한 부분을 해결하기 위한 방법 중의 하나가 바로 이 논문에서 제안한 대규모 웹사이트에서 실시간으로 갱신되는 자료에 대한 동시성을 부여하고 데이터베이스의 부하를 최소화 시킨 갱신시스템이다. 이것은 사용자의 요구에 대한 정보전달시 발생하는 웹 페이지의 지속적인 갱신을 의미한다. 이미 사용한 웹페이지의 정보를 재사용하여 전달하는 방식으로 응용 데이터베이스와 이미 사용된 가상 데이터베이스와의 동기화를 통한 최신 정보를 전달하는 시스템이며, 중간에 동기화를 관여하는 리프레시 매니저를 사용하여 보다 효율적으로 정보를 동기화 하는 것이 그 주된 내용이다. 따라서, 본 논문에서는 대규모 웹사이트의 최신 정보를 유지하기 위한 갱신시스템을 구현하고, 사용자의 요구에 응하는 웹사이트의 지연 시간을 줄이고, 데이터베이스의 부하를 최소화하는 갱신시스템 프로토콜(Protocol)을 제안한다.

본 프로토콜에서는 갱신시스템의 주기적인 점검을 통해 이미 사용된 웹 정보와 신규로 생성된 최신정보 사이에서 발생하는 데이터의 일관성을 유지하고, 이미 사용된 웹정보를 무효화 시켰는지에 관한 정보도 관여하게 된다. 그리고 이 과정에서 무효화되는 정보는 사용자의 자료 요청 시 갱신시스템의 부하를 줄이게 되고, 사용자는 응용 서버를 통해 실시간으로 최신정보를 제공받게 된다. 본 논문에서는 실시간 경매 시스템을 통해 제안된 프로토콜(Protocol)을 구현하여, 동시 접속자수의 증가에 따른 웹사이트의 응답속도를 모의실험을 통해 그 성능 향상을 입증하였다.

이 논문은 다음과 같이 구성이 되어진다. 먼저, 제 2 장에서는 본 연구의 기반이 된 최신 관련 기술인 JDBC와 자료에 대한 갱신처리 방법에 대

해 살펴보고, 제 3 장에서는 대규모 웹사이트 갱신시스템의 설계와 구현에 대해 살펴보고, 제 4 장에서는 제안된 프로토콜의 성능을 모의실험을 통해 살펴본 후, 제 5장에서는 그 결론과 향후 연구과제를 밝힌다.

제 2 장 관련 연구

2.1 WWW 및 웹 기반의 기술-JAVA2

정보화시대의 지식의 획득 저장소라 불리는 인터넷은 방대한 기존의 연구 정보들을 효율적으로 공유하기 위한 목적으로 출발하였다. 처음에는 전화선을 통한 연결을 시도하였는데, 1969년 미 국방성의 지원으로 4개의 기관이 군사목적의 연구 프로젝트를 수행하기 위해 연결된 것이 ARPAnet(Advanced Research Project Agency)으로 인터넷의 시초라 할 수 있을 것이다.

이후 이를 기반으로 미국 국립과학재단(NSF : National Science Foundation)이 미국에 있는 5개의 슈퍼컴퓨터 센터들을 서로 연결하면서 전국적인 네트워크를 형성할 수 있었는데, 이 NFSnet은 미국의 인터넷 기간망으로 역할을 하였고, 이에 따라 인터넷은 민간부문을 중심으로 발전하게 되었다. 그러나, 초창기의 이러한 네트워크는 소수의 사람들만이 사용했고, 실질적으로는 1989년 WWW(World Wide Web) 서비스가 개발된 후에야 일반 사용자들이 쉽고 편리하게 사용하게 되었다. 그렇지만 초기에는 문자와 자판만을 이용한 인터넷 사용만이 가능했기 때문에 사용자 수가 적었지만, 1993년에 들어서서 마우스와 GUI(Graphic User Interface)를 지원하는 Mosaic 이라는 웹 브라우저가 개발되었으며, 1994년에 출현한 넷스케이프 네비게이터와 인터넷 익스플로러 같은 고성능 웹브라우저의 등장으로 인해 인터넷 사용자 인구는 폭발적으로 늘어나게 되었다.

웹브라우저의 출현 이후로 WWW를 이용한 다양한 멀티미디어 정보와 서비스가 제공되어 이제는 누구라도 쉽게 인터넷을 사용할 수 있게 되었고, 인터넷은 사회 전반에 걸쳐 널리 이용되어지고 있다. 현재 웹을 중심으로

다양한 인터넷 기술들이 발달하고 있으며 학술, 교육, 상업, 취미 생활 등 매우 폭넓은 영역에서 이용되고 있다. 또한 전 세계적으로 미래의 정보시대에서 뒤지지 않기 위하여 초고속 정보 통신망 사업 등을 벌이고 있으며 인터넷을 이용한 사업들도 활성화되고 있다.

자바는 HTML과 함께 인터넷을 통해 혁명을 불고 온 주역이다. 자바는 네트워크나 소형기기를 위해서 기획된 플랫폼의 독립적인 언어이자 보완성을 가진 프로그램 또는 개발 플랫폼이다. 이러한 간단한 정의에서도 볼 수 있듯이 자바란, 자바 실행 환경, 자바 언어, 자바 OS를 동시에 지칭한다. 보통 플랫폼의 독립성, 보안, 작고 효율적인 프로그램, 사용하기 쉬운 언어의 특성을 가진다. SUN은 JDK 1.1x 버전을 개발하면서 자바 자체의 개념을 크게 바꾸기보다는 자바의 약점을 보완하는 방향으로 가닥을 잡았고, JDK 1.2의 개발에 즈음해서는 자바 플랫폼의 계층별 분화도 이루어진다. 즉, 표준자바 이외에 소형기기에 사용할 목적인 퍼스널 자바(Personal Java), 임베디드 자바(Embedded Java), 자바 카드(Card)가 생겼다. 물론 각 플랫폼은 각각 상위 플랫폼의 서브 플랫폼 성격을 가지며 호환성이 있다. 이러한 변화를 거쳐 현재, 자바는 JDK 1.2x~1.3 또는 JAVA2 플랫폼 시대를 맞이하고 있다. JAVA2 플랫폼의 전체적인 구성은 다음과 같다.

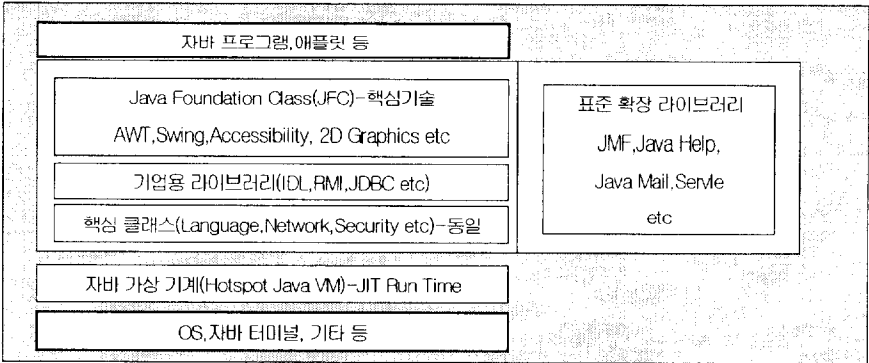


그림 2-1 JAVA2 플랫폼 전체 구성

JAVA2 플랫폼이 갖는 의의로 몇 가지가 있다.

첫째, 이제까지 서드 파티(Third Party)에 의존했던 많은 라이브러리를 자바 표준 라이브러리에서 제공하게 된다.

둘째, 빠른 확장성과 안정적인 환경이라는 두 가지 상반된 문제에 대해서 어느 정도 해결책을 제시한다.

셋째, 기업용 API(Enterprise API)를 많이 제공하여 실 업무에 사용할 수 있는 자바 프로그램을 제작할 기반을 제공하기 시작했다는 점이다.

이는 분산 환경의 표준이라고 할 만한 IDL, DB 업계 표준의 SQL 질의를 가능하게 한 JDBC, 원격 함수 호출이 가능하게 한 RMI 등을 지원함으로써 이루어 졌다. 이외에 JAVA2 플랫폼은 속도 향상을 위하여 향상된 자바 가상 기계인 HotSpot 가상기계를 사용하게 된다. 이는 자바 코드 중 특히 많은 프로세스를 소모하는 부분을 분석하여 가속시키는 기술이다.

2.2 갱신시스템 모델

WWW의 급속한 성장으로 인해서 데이터베이스로부터 정보를 읽어 절반만 동적인 페이지의 수가 급속히 늘어나고 있다. 이는 기본 데이터의 내용이 많고, 페이지수가 많아짐으로 인해서 웹 상에서는 새로운 정보전달의 문제가 되고 있다. 이러한 경우 웹 상의 문제에 대해 시간상으로 가장 최근의 정보를 다룰 경우에 발생하는 중요한 사항이라고 할 수 있다.

일반적인 웹페이지는 그림 2-2와 같이 개인 홈페이지처럼 거의 변화하지 않는 페이지인 정적인 웹페이지와 CGI를 통한 항상 주기적인 정보 요청하여 사용자에게 최신 정보를 전달하는 동적인 웹페이지, 그리고, 절반만 동적인 웹페이지로 나눌 수 있다. 절반만 동적인 웹페이지란 데이터베이스로부터 전달되어지는 내용을 가지며, 업데이트 시 그 내용이 변화하는 것을

말한다. 이러한 일반적인 웹사이트 형태는 페이지의 최신정보를 획득 할 수 없고, 정보전달 시 많은 부하를 발생하게 된다.

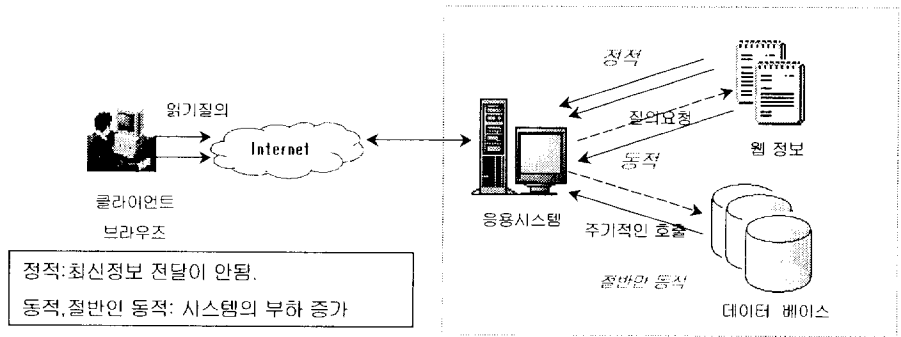


그림 2-2 일반적인 웹사이트 형태

그림 2-3은 절반만 동적인 웹페이지를 보완한 것으로 동적 웹페이지 (Dynamic HTML)와 정적 웹 페이지(Static HTML)를 적절하게 구현한 자동-동적 웹 페이지(Auto Dynamic HTML) 의미한다. 이런 방식에는 갱신시스템이 필요하다.

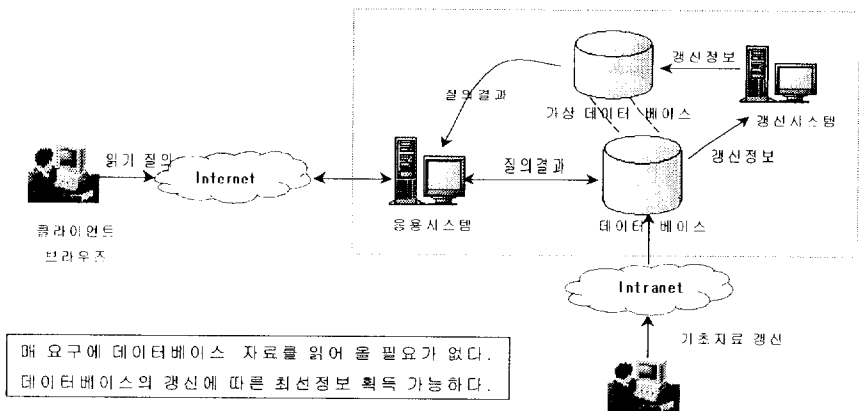


그림 2-3 자동-동적 웹 페이지

이 논문에서 데이터베이스의 갱신시스템은 두 가지 일을 포함하고 있다. 하나는 여러 사용자에게 자료를 실시간으로 제공하는 일이고, 다른 하나는 데이터베이스와 사용자간에 전달할 데이터를 가상 데이터베이스 상에 존재하게 될 실 데이터베이스 복사본들에 대한 일관성을 책임지는 일이다.

갱신시스템은 비교적 구현이 쉽고 확장성이 우수하여 많은 시스템에 채택이 가능하다. 그러나, 기초자료에 대한 일관성 유지가 모두 가상 데이터베이스를 거쳐야 한다는 단점이 있다. 이로 인하여 대규모 시스템에서 소수의 요구사항을 가진 다수의 사용자일 경우 전체 사용률이 점차 증가되기 때문에 읽기 실패에 대한 전체 부하가 더욱 커질 수 있다. 이러한 경우 성능의 손실을 줄이기 위한 방법이 필요하게 된다.

공유된 웹 정보의 모든 결과 복사본은 사용정책에 따라 성능을 향상시키기 위해 무효화가 되어 부담이 적게 드는 방법을 채택한다. 이는, 무효화를 당한 자료가 바로 다음 사용자가 다시 읽기를 요청했을 경우 초기 설정치인 공유된 모든 복사본의 자료를 새로운 값으로 갱신하는 방법으로 인하여 지연 시간이 증가된다는 점을 감안하여 가상 데이터베이스의 갱신시스템 설계 시 아주 중요한 사항이다.

일반적으로 자료의 무효화와 자료의 갱신을 이 시스템에서는 주로 사용한다. 또한, 데이터 동기화(Data Synchronization)도 매우 중요하다. 동기화란, 둘 이상의 응용 프로그램이 사용하고 있는 데이터의 일치성을 보장하는 기술이다. 한 쪽에 어떤 데이터의 변화, 즉 입력, 수정, 삭제와 같은 변경이 일어났을 경우 상대방 데이터도 이러한 변경 사항을 적용해야 하는데 이것이 데이터 동기화 기술이다.

데이터 동기화 시점에 따라 응용 프로그램에 의해 일정 시간 간격이나 특정 시간대에 데이터 동기화 작업이 일어나게 하는 방법과 한쪽에서 데이터 변경이 일어나면 실시간으로 상대방 데이터의 변경을 적용하는 두 가지 방식이 일반적으로 쓰이고 있다. 두 가지 방식 모두 각각 장단점이 있어 도

메인에서는 이 둘을 혼용해 다양한 시나리오에 따른 동기화 동작을 핸들링하고 있다. 데이터 동기화는 단순한 데이터 가공 기술로 생각해서는 안 된다. 데이터 변환이 동일한 스키마 내의 정보를 상대방으로 전부 이동하는 대량의 기계적인 작업이라고 한다면 데이터 동기화는 스키마와 데이터 접근 방식에 영향을 받지 않고 안정적으로 상대방 정보를 변경할 수 있는 좀 더 지능적인 기술이라고 할 수 있다.

2.3 데이터베이스 연동

불과 수 년 전만 해도 데이터베이스를 이용하는 개발은 데이터베이스에 의존하여 작업하는 고도의 기술 작업이었다. 근래 웹 개발에서 데이터베이스 연동은 빼놓을 수 없는 요소이며, 웹 페이지의 중요한 기술 중의 하나가 웹에서 데이터베이스와 연동하는 것이다.

2.3.1 JDBC(Java Database Connectivity)-JAVA

웹과의 연동하는 기술은 매우 다양하며, 본 논문에서 주로 사용하는 기술은 자바에 관련된 기술과 JDBC(Java Database Connectivity)이다. 자바는 다른 개발 환경보다도 훨씬 쉽고 강력한 데이터베이스 조작 인터페이스를 제공하는데 이것을 JDBC라고 한다. 즉, SQL문들을 JDBC 인터페이스에 있는 메소드의 인수로서 사용할 수 있게 해 주는 SQL수준의 API이다. SQL 수준의 API라는 것은 자바 API를 호출함으로써 주어진 SQL을 사용할 수 있다는 것을 의미하는데, 쉽게 말해서 기본적으로 SQL을 사용하여 데이터베이스와 연동한다는 뜻이다.

JDBC Application은 데이터베이스의 특수한 특성으로부터 잘 분리되어 있으므로 특정 데이터베이스에 대해 다시 설계할 필요가 없다. 자바환경에서 Database를 연결할 수 있으려면 다음 세 가지 입장에 있는 사람들이 서

로 같은 표준을 가지고 Program을 개발해야 한다. 첫째, JVM 환경과 둘째, 응용 Program 개발자(일반 Programmer)와 셋째, Database Software 개발자 (Oracle, Sybase, Informix 등)이다. 이러한 표준이 JDBC이다.

개발자는 자신이 사용하는 데이터베이스 엔진을 위한 JDBC 드라이브를 사용해야 하고, JDBC Driver 유형은 보통 다음과 같이 4가지로 나뉜다.

유형 1 : ODBC Driver를 연결하여 작동하는 JDBC Driver를 의미한다. 이것은 MicroSoft의 표준인 ODBC Driver를 연결하고 이를 통하여 데이터베이스에 접근하는 방식이기 때문에 JDBC-ODBC Bridge라고도 한다. 유형 1 JDBC Driver를 사용하려면 Client 컴퓨터에 ODBC Driver와 JDBC Driver가 설치되어 있어야 한다. 이 때문에 동적으로 Driver가 Download 되어 사용하는 환경에서는 사용하기 어려우며 고정된 Driver가 설치되어 운영되는 환경에 적합하다. SUN에서 JVM에 기본적으로 제공하는 JDBC-ODBC Driver는 sun.jdbc.odbc.JdbcOdbcDriver이다.

유형 2 : 이 타입은 일부가 자바로 되어 있는 JDBC Driver를 의미한다. 따라서 유형 3과 마찬가지로 Client 컴퓨터에 설치되어 있는 Local Driver가 필요하다. 다만, 이 Local Driver가 ODBC 형식이 아니고, 데이터베이스를 만든 Vendor가 제공하는 다른 형태의 Driver이며, Vendor의 Local Driver와 JDBC Driver가 Client 컴퓨터에 설치되어 있어야 사용 가능하다. 이러한 형식은 Oracle, Informix, DB2 등 DBMS Vendor들이 제공하는 API를 다시 호출하게 되는데 유형 1 과 마찬가지로 Client 컴퓨터에 정적으로 Driver를 설치하여 데이터베이스와 연동하는 환경에 적합하다.

유형 3 : 이 타입은 완전한 자바기술 및 코드로 이루어진 JDBC Driver를 의미한다. 이 타입은 JDBC API 표준에 의하여 만들어졌기 때문에

DBMS의 종류에 상관없이 사용할 수 있다. 4가지 유형 중에서 가장 융통성이 뛰어난 Driver 타입이기 때문에 동적으로 Download되는 Applet 같은 환경에 더욱 적합하며, Client 컴퓨터가 정적으로 이루어졌다고 하더라도 유형 3을 사용하면 시스템 유지보수가 더욱 쉬워진다. 따라서 특별한 요구사항이 필요한 경우 즉, 특정 DBMS 만에 제공하는 기능을 사용할 필요가 있는 경우 또는 DBMS 자체가 유형 3을 지원하지 않는 경우를 제외하고는 이것을 사용하는 것이 가장 적절하다.

유형 4 : 완전한 자바기술 및 코드로 이루어진 JDBC Driver를 의미한다. 하지만 유형 3과는 다르다. JDBC API 표준을 기준으로 만들어졌다고 보다는 DBMS Vendor가 표준을 기준으로 기능을 추가 또는 삭제하여 독자적인 형태로 만든 유형이다. 따라서 특정 DBMS에 의존적인 반면 해당 DBMS만이 제공하는 기능들을 사용할 수 있다는 장점이 있다. Client가 동적인 환경에서 특정 DBMS만이 제공하는 기능들을 사용할 경우 적합하다.

이 네 가지 유형 중에서 유형 3이 JDBC로부터 데이터베이스에 접근하는 우선적이 방법이 될 것이고, 그림 2-4는 JDBC에서 데이터베이스로의 경로이다.

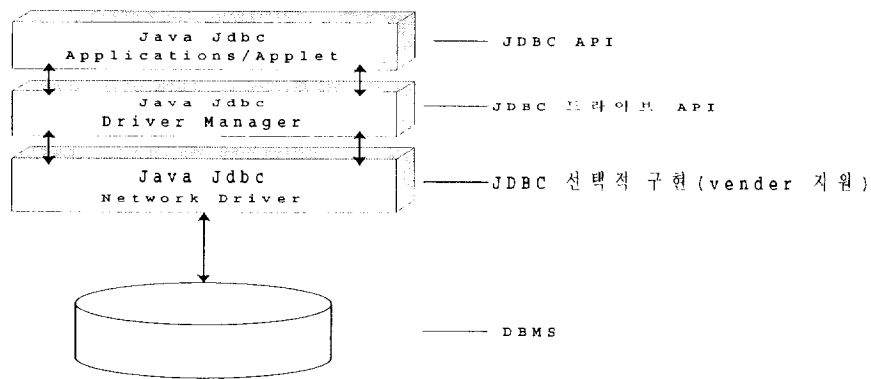


그림 2-4 JDBC에서의 데이터베이스 통신경로

이런 구성에 따라 JDBC 프로그램은 다음 Logic의 절차를 따른다.

1. Driver를 Loading 한다.
2. DB와 연결을 하여 Connection Object를 얻는다.
3. Connection으로부터 SQL을 실행하기 위한 Statement Object를 얻는다.
4. Statement Object의 API(Method)를 이용하여 SQL을 실행한다.
5. SQL을 실행한 결과인 ResultSet Object를 얻는다.
6. ResultSet Object에 있는 결과 자료를 Looping 을 이용하여 처리한다.

그림 2-5은 클라이언트-서버 환경에서의 데이터베이스와 어떻게 연결되는지를 보여준다.

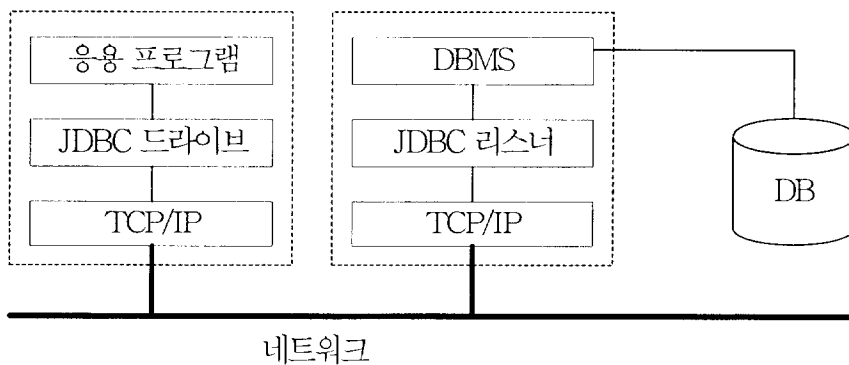


그림 2-5 JDBC 드라이브 이용 환경

2.3.2 ADO(ActiveX Data Object)-ASP

ADO(ActiveX Database Object)는 ODBC기반의 데이터베이스 관리 기능을 가진 고성능의 객체이다. 웹 콘텐츠 개발자들이 쉽게 DB에 접근해 서버차원에서의 Dynamic Html을 제작하고 데이터를 관리할 수 있도록 해 준다. ODBC 기반이므로 액세스, 오라클, 사이베이스, 인포믹스, 마이크로소프트 SQL서버 등 다양한 DB를 사용할 수 있다. ADO는 마이크로소프트가 갖고 있는 DAO(Data Access Object)와 RDO(Remote Data Object)기술을 통합하고, 인터넷에서 구동될 수 있도록 ActiveX 기술을 접목시킨 새로운 데이터베이스 객체이다. 즉, ASP(Active Server Page)에서는 이 ADO를 사용하여 데이터베이스와 연결하고 질의를 처리하고 그 결과를 받아와 동적인 페이지를 만들어 낸다. 특히 WWW에 사용자 인트라넷에서 정보를 효율적으로 다양한 형태로 처리할 수 있고, ADO를 통하여 안정적인 데이터베이스 액세스 기능을 제공하지만, 모든 플랫폼에 독립적이지 않고 윈도우 계열 플랫폼에 의존적인 것이 단점이다. 본 논문은 ODBC에 호환되는 다양한 형태 데이터베이스 내용을 제어하고 검색이 쉬운 ADO기법을 갱신시스템 모의실험의 인트라넷 작업에 사용하였다.

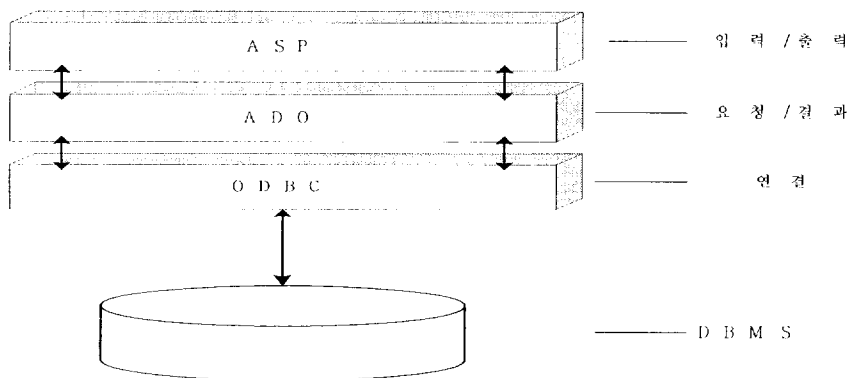


그림 2-6 ADO에서의 데이터베이스 이용환경

제 3 장 웹사이트 갱신 시스템의 설계 및 구현

3.1 전체시스템 설계

본 연구에서는 전통적인 클라이언트-서버 모델인 2-tier구조를 개선한 3-tier 클라이언트 서버 모델을 사용하였다. 이것은 3-tier로도 가능하기 때문이다. 3-tier구조에서는 응용처리를 전담하는 Middle-tier를 두고 클라이언트는 모든 서비스를 이 Middle-tier에게 요청하고 Middle-tier는 데이터베이스와 같은 특정한 작업만 수행하고 서버와 통신한 후 그 결과를 클라이언트에게 전달하는 구조로 구성된다.

다음은 3-tier의 클라이언트-서버 모델의 개념적 그림이며, 기능을 정의하면 다음과 같다.

- Tier 1 : 사용자 인터페이스를 담당하는 클라이언트
- Tier 2 : 응용처리 서버
- Tier 3 : 데이터베이스 서버와 같은 최종 기능을 수행한 서버

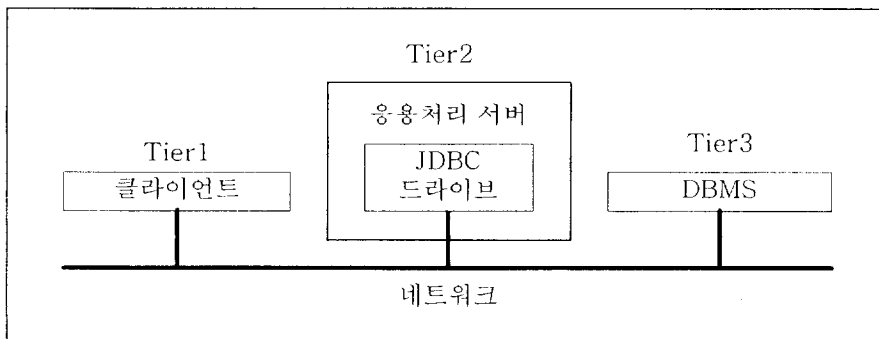


그림 3-1 3-Tire 클라이언트-서버 모델

대용량 서버에 적합한 3-tire서비스는 확장성(Scalability), 안정성과 효율면에서 아주 유리하고, 인터넷 사용자의 요청에 따른 처리를 직접 처리하지 않기 때문에 서버의 부담이 클라이언트에 비례하여 증가하지 않아 효율적이다. 이와 반면에 클라이언트 수가 적거나 클라이언트가 단순히 하나의 자료에 접근하는 경우는 2-tire 구조가 더 안정적이다.

그림 3-2는 JAVA2 플랫폼을 사용한 웹사이트 갱신시스템의 전체 구조도 이다.

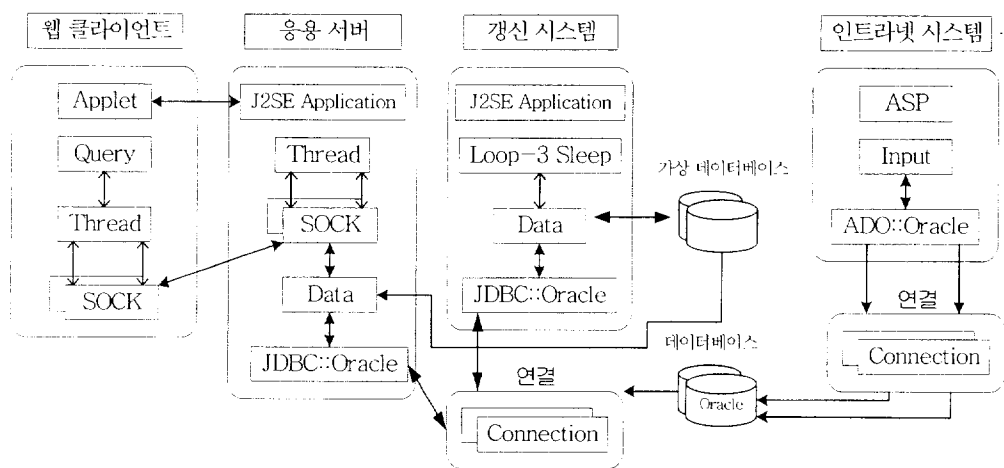


그림 3-2 JAVA2 플랫폼 기반의 갱신시스템

웹 정보에 따른 사용자 목적과 특성을 고려하여, 최소한의 패킷으로 정보를 보낼 수 있게 하고, 사용자가 처음 자료 요청 시 조금 속도를 늦추고, 연결 후에는 사용자가 많은 경우와는 관계가 없는 실시간으로 웹 정보를 전달하는 것을 설계 기본 방향으로 설정한다.

시스템의 개략 설계를 요약해서 시스템 기본 전략을 다음과 같이 설정하였다.

첫째, 자동 인터넷·인트라넷 시스템 : 모든 시스템이 웹 기반 클라이언트·서버 모듈로 변환되고 있는 추세이기 때문에 모든 조회 프로그램을 웹 기반(JAVA2 Standard Edition)으로 설계하였다. 그러므로 모든 사용자들은 별도의 프로그램이 필요 없이 웹브라우저로 접근이 가능하다.

둘째, 온라인 실시간 조회 : 웹브라우저로부터 정보를 요구하여 인터넷으로 접근하고 전송, 정보 획득, 정보 갱신, 가상 정보 열람까지 모두 실시간으로 웹에서 가능하다.

셋째, 쉽고 빠른 접근 : 사용자 접근 환경을 최대한 고려하여 정보가 많은 경우 정보 전달을 최소로 분할하여 과다정보는 배제, 최적 질의를 자동 분류하는 등 속도개선에도 배려했다.

넷째, 갱신시스템 분산 저장 : 전체 정보 처리는 서버에 일괄 처리하여 사용자의 데이터베이스의 자료접근을 최소화하여 네트워크 병목현상이 발생하지 않도록 했다.

다섯째, 자료의 융통성 제공 : 한번 데이터베이스에서 가져온 정보는 사용자 정책에 따라 구성되어 갱신시스템의 특성을 살리고, 갱신시스템의 전략이 웹기반에 고속처리로 반영되도록 배려했다.

여섯째, 단일 사용자와 같은 효과 : 이 시스템에서 한번 열람된 정보를 사용자가 보관하여, 모든 사용자에게 정보 요청에 따라 최소의 부하로 전달하고, 다른 작업을 할 수 있게 부족하기 쉬운 인간적인 상호작용을 향상시키기 위한 실시간 열람기능을 제공한다.

본 논문에서는 대규모 웹사이트의 최신 정보를 유지하기 위한 갱신시스템을 구현을 위한 전략에 따른 프로토콜(Protocol)은 그림 3-3과 같다.

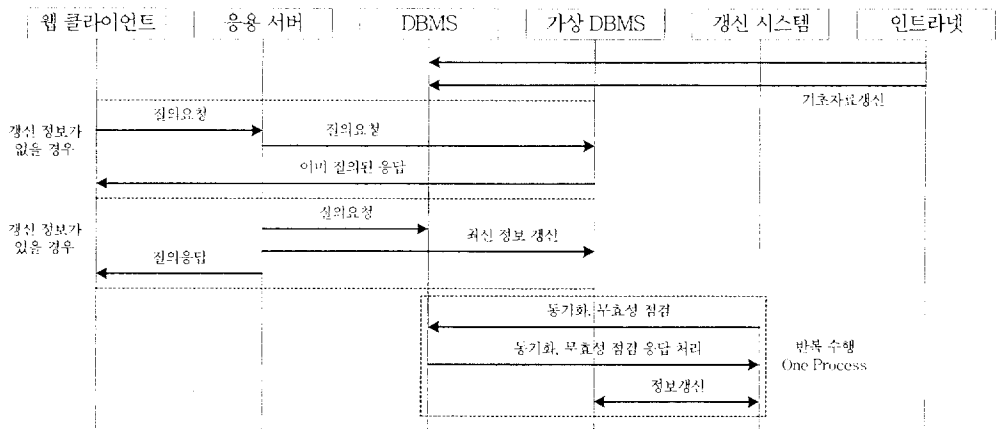


그림 3-3 갱신시스템 프로토콜(Protocol)

진행과정을 살펴보면 사용자는 웹 페이지를 통한 질의 요청 시 데이터 베이스에 질의 결과를 점검하여 가상 DBMS에 자료가 있을 경우 가상 DBMS를 통해 전달하고, 없을 경우 데이터베이스에서 질의 결과를 전달한다. 이때 갱신시스템은 일정 주기로 동기화 및 무효성 점검을 통하여 한번 질의된 내용은 가상 DBMS에 존재하게 된다. 응용서버는 실시간으로 점검하여 분류된 검색결과를 사용자에게 실시간으로 전달한다. 모든 기초자료의 갱신은 인트라넷 사용자만 자료의 변경이 가능하고, 변경된 내용도 갱신시스템이 자동으로 데이터베이스와 가상 DBMS의 자료를 동기화 한다.

3.2 가상 DBMS의 분석

갱신시스템의 설계와 제작은 두 가지 방법으로 하였다. 사용자의 검색정보와 서버에서의 데이터베이스의 정보를 처리하는 방법은 두 가지이다.

사용자의 검색정보에 의하는 경우는 검색을 선택하여 검색 정보를 갱신시스템에 전달함으로써 자료의 일관성이나 인터페이스의 편리성 같은 장점을 가진다. 그러나, 처음 사용자의 검색일 경우, 갱신시스템의 신규정보를 구축함으로써, 조금의 시간이 소요되며, 두 번째 열람했을 때에는 상당히 많은 직접적인 상호작용을 해야 하는 단점이 있다.

갱신시스템은 신규정보가 많아질 경우 서버에서의 처리 범위가 넓어지고, 사용자 인터페이스에서 자료의 변경이 발생할 경우 일관성과 다양성이 향상되어야 하기 때문에 별도의 사용자 인터페이스가 필요하다. 아주 작은 자료의 변경이 발생할 경우는 갱신시스템에서 소요시간이 적게 든다는 장점을 가지고 있다. 본 시스템에서의 자료전달 특성을 고려하여 여기에 적합한 가상 셀 부분을 나타낸다. 사용자의 정보를 열람하는 부분에서 이 부분이 데이터베이스의 역할을 담당한다.

다음과 같이 가상DBMS를 구성하기 위한 갱신시스템 모델을 정의한다.

(1) 기초자료(Base Data)

인트라넷 응용프로그램에 의해 독립적으로 웹에서 운영되며, 실시간으로 데이터베이스에 반영된다.

(2) 웹 셀 베이스(Web Cell Base)

관계형 데이터베이스의 테이블로부터 생성된 정보를 포함하고, 기초 자료의 정보의 최적화된 질의에 대한 결과를 가지고 있다.

(3) 웹 페이지(Web Page)

웹 공간에서 사용자의 정보에 의해 보여지는 페이지로서 실시간으로 웹 셀 베이스의 정보를 리프레시 한다.

(4) 리프레시 질의(Refresh Query)

최신정보를 얻기 위한 모음으로써 결과의 내용으로 기초자료에 보관된다.

(5) 리프레시 매니저(Refresh Manager)

기초자료와 웹 셀 베이스의 정보를 동기화하고, 웹 페이지에 최신 정보를 일정한 패턴으로 전송하고, 무효성 데이터에 대해 점검하고, 데이터에 대한 신뢰성을 보장한다.

3.3 서버 시스템 구축

웹 상에서 갱신시스템의 서버 시스템은 웹서버와 웹 데이터베이스 서버로 구성되었다. 인트라넷 사용자가 데이터베이스를 관리 및 갱신하기 위해 ORACLE 8.0.1 을 이용하여 Windows 2000 Server 기반에서 프로그램 기법은 ASP(Active Server Page) 구축하였다.

표 3-1 웹 서버와 웹 데이터베이스 서버의 환경

네트워크 프로토콜		TCP/IP SOCK
하드웨어 환경	CPU	Pentium III 800 MHz
	RAM	128Mbyte
소프트웨어 환경	OS	Windows 2000 Server
	Web Server	IIS 5.0
	DB Server	Oracle 8.0.1
	DB Driver	JDBC 2.0, ODBC Drive
	개발언어	ASP, Java Applet, Java Application

사용자가 질의요구에 대한 자료 검색은 플랫폼과 상관없는 J2SE(Java2

Standard Edition)으로 구성하였다. 웹 서버는 IIS 5.0으로 구현하였고, 데이터베이스 서버는 ORACLE 8.0.1로 구현하였다. 웹 응용 프로그램과 데이터베이스 서버간에는 ADO와 ODBC를 이용하고, 웹 조회프로그램은 JDBC와 소켓으로 연동하였다. 웹 서버는 인트라넷에서 정보의 갱신을 위해 개인에 정보를 맞는 전달을 한다. 서버의 사양과 서버구축 환경은 표 3-1과 같다.

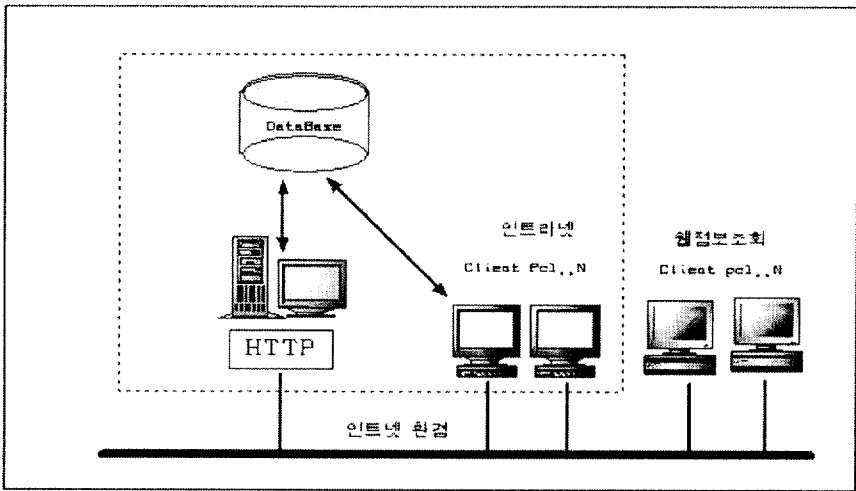


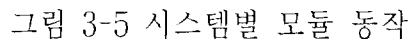
그림 3-4 웹 서버 구성 및 연결도

3.4 데이터베이스 설계 및 구축

갱신된 데이터베이스란 가상 데이터베이스(셀 베이스)를 의미한다. 이것은 자료의 일관성이나 무효성에 이르기까지 전체적으로 다양하게 사용될 수 있어야 한다. 이러한 데이터베이스의 가장 큰 역할은 동시성과 신뢰성을 만족하게 하는 것이다. 사용자가 무엇을 원하는지를 잘 판단하여 자료를 어떻게 줄 것인가를 정해야 한다는 의미이다. 자료에 대한 구체적인 갱신 시스템을 위한 가상 데이터베이스를 구축하려면 우선 어떻게 설계 할 것인가를 결정하여야 한다. 즉, 어느 정도의 규모로 하고 어떤 정보를 수록해야 하며

완성도가 높은 많은 정보를 담기 위해서는 아주 중요하다고 판단되는 내용 부분만 사용해야 한다. 바람직하지 못한 설계를 할 경우 전체 웹사이트의 장애를 가져 올 수 있고, 불필요한 자료에 대한 전달은 정보의 낭비일 수 있다.

본 시스템은 기능에 따라 4개로 모듈화 하였으며, 그림 3-5 시스템별 모듈동작과 같이 하였다. 사용자 시스템 모듈, 서버 시스템 모듈, 갠신 시스템 모듈, 응용프로그램 모듈로 구분된다.



사용자 시스템 모듈은 그림 3-5와 같이 웹 클라이언트를 의미한다. 사용자가 자료를 요청하면 이 정보를 서버 시스템 모듈로 전송하는 역할을 하며, 서버에서 자동 갱신된 정보가 있을 경우 웹 요청자에게 갱신시스템은

갱신된 정보를 실시간으로 사용자에게 맞게 정보를 부여하는 모듈이다. 웹 페이지를 설계하기 위해서 다음 설계 방향을 설정해야 한다.

첫째, 인터넷에 익숙하지 못한 사용자도 이용할 수 있게 편의성에 중점을 둔 인터페이스를 제공한다.

둘째, 한번의 조회로서 최신 데이터의 결과를 가져온다.

Step 1 서버의 주소와 포트를 연다.

Step 2 Connect를 연결한다.

Step 3 소켓으로 InputStream과 OutputStream을 생성한다.

Step 4 최적화된 질의를 통하여 정보를 요구한다.

Step 5 수신된 정보를 화면에 갱신한다. Step 4가 반복된다.

코딩 내용에서 주요 부분은 그림 3-6과 같다.

```
public class AppletClient extends Applet implements
Runnable, ActionListener
{
    -- 생략 --
    // Step 1,2,3초기화
    public void init()
    {
        -- 생략 --
        this.initGraphics(); // 화면구성
        try{                // 소켓 생성
            clientSocket = new Socket(this.getCodeBase().getHost(),20001);
            oos = new ObjectOutputStream(clientSocket.getOutputStream());
            ois = new ObjectInputStream(clientSocket.getInputStream());

            d = new Data(schatziname, "Connecting....", Data.FIRST_AIR);
            oos.writeObject(d);

        } catch(Exception e){System.out.println("소켓생성에러" + e);}
    }
}
```

```

// Step 5. 데이터 처리
public void run()
{
    while(!exit){
        try{
            d = (Data)ois.readObject();
            //메시지를 받아서 집어넣음
            list.add("[ " + d.getName() + "]: " + d.getMessage());
            //메세지 출력 리스트박스가 40줄을 초과하면
            //가장 위에 있는 것을 삭제
            if(list.getItemCount()>40)
                list.remove(0);
            list.select(list.getItemCount()-1);
        }catch(Exception e){
            System.out.println("데이터 읽기 예러" + e);
            exit = true;
        }
    }

    // 화면 처리 Step 4.
    public void actionPerformed(ActionEvent e)
    {
        Object obj = e.getSource();
        if(obj==tf1) {
            try{
                d = new
                Data(schatziname, tf1.getText(), Data.ON_AIR);
                oos.writeObject(d);
                tf1.setText("");
            }catch(Exception ex){System.out.println("기록하는데 예러");}
        }

        -- 생략 --
    }
}

```

그림 3-6 사용자 시스템 모듈 코드 내용

(2) 서버 시스템 모듈

서버 시스템 모듈은 그림 3-5와 같이 서버 ORACLE 데이터베이스의 정보 결과를 사용자 모듈에 전달하는 부분이다. 질의를 요청한 사용자의 정보를 받아서 내부 조회코드로 분류하여 가상 데이터베이스 가능 여부를 확인하여 자료를 갱신 시스템이 작동을 할 수 있게 하는 역할을 한다. 즉, 사용자 시스템 모듈을 모든 정보를 관리하는 모듈이다. 데이터베이스와 가상 데이터베이스를 위한 설계방향을 설정해야 한다.

첫째, 사용자 모듈과 서버모듈의 최적 질의를 서로 분류를 시킨다.

둘째, 데이터베이스 검색을 우선으로 하여 가상 데이터베이스 처리 여부를 결정한다.

Step 1. 서버 소켓 생성(특정포트)

Step 2. 서버소켓으로 클라이언트 접속허용(accept)

Step 3. 클라이언트가 접속을 했을 때 서버 소켓은 클라이언트와 통신할 수 있는 하나의 소켓하나를 리턴해 준다.

Step 4. Accept는 계속해서 while문을 돌면서 클라이언트의 접속에 대기한다.

Step 5. 접속 할 때마다 접속한 클라이언트와 통신을 할 수 있는 소켓을 하나 리턴해 준다.

Step 6. 하나의 서버 소켓에 많은 클라이언트들이 접속할 때마다 통신을 하기 위한 하나의 소켓이 만들어진다.

Step 7. 스트림을 열어 InputStream과 OutputStream을 생성하고 JDBC를 통하여 정보를 전달한다.

Step 8. 갱신된 내용이나 변경된 질의에 대해서 Step 7을 반복한다.

코딩 내용에서 주요 부분은 그림 3-7과 같다.

```

public class SchatziServer
{
    -- 생략 --
    //소켓,소켓동적배열,질의동적배열 정의
    while(true){-- 생략 --
        tempSocket = serverSocket.accept();
        // 읽고 쓰기 소켓 정의
        ois = new ObjectInputStream(tempSocket.getInputStream());
        oos = new ObjectOutputStream(tempSocket.getOutputStream());
        /*****클라이언트 저장*****/
        buffer.addElement(new Client(tempSocket, ois, oos));//백트저장
        buffer_room.addElement("0000");    //백터 저장
        /*****클라이언트에 관련된 스래드생성*****/
        CommonThread ct = new CommonThread(tempSocket, ois, oos);
        ct.start(); -- 생략 --
    }
}

//클라이언트에서 보낸 것을 감시해서 소켓,oos,ois를 가져서
//브로드캐스트를 하는 역할

public class CommonThread extends Thread
{
    // 소켓 정의
    -- 생략 --
    // 데이터베이스와 가상데이터베이스를 생성한다.
    public void DbQuery(String sQuery)
        throws SQLException, ClassNotFoundException
    {
        -- 생략 --
        // 1. JDBC로 DB연결 및 질의
        // 2. 결과를 생성한다.
    }

    //갱신된 데이터를 읽어서 클라이언트에게 보낸다.
    public void dBrefresh(String sSendinfo)
    {
        -- 생략 --
        // 1. 정보를 읽는다.
        // 2. 사용자의 구분에 따라 결과를 소켓으로 전송한다.
    }
}

```

```

public void replaceClient(String sGubun)
{
    // 화면출력 Vector을 키값을 변경한다.
    -- 생략 --
    for(int i=0; i<size; I++)
    {
        Client c = (Client)v.elementAt(i);
        if(c.socket == this.socket){
            v_room.removeElementAt(i);
            v_room.insertElementAt(sGubun,i);
        }
    }
}

-- 생략 --
// 소켓으로 자료를 받아
replaceClient(sQuery); // 질의값을 동적배열로 치환한다.
this.broadcast(sQuery); // 화면 표시
try{
    while(true) {
        File f1 = new File(sPathdata+sQuery+".txt");
        if(!f1.exists()){
            System.err.println("File Not Found.");
            DbQuery(sQuery) ;
        }
        // 이미 읽은자료를 비교하여 반복 전송을 제어한다.
        if(f1.lastModified()!= lfilesize){
            dBrefresh(sQuery);
            // 자료를 리프레시한다.
            Thread.sleep( 2000 ); // 2초대기
        }
        lfilesize=f1.lastModified();
        try{
            socket.setSoTimeout(3000);//3초동안 소켓 반응없을
            if(null !=(d = (Data)ois.readObject()))
            {
                sQuery= d.getMessage() ;
                // 소켓에서 질의를 수신
            }
        }
        -- 생략 --
    }
}

```

```

// 화면 전송
public synchronized void broadcast(String gubun)
{
    Enumeration e = SchatziServer.buffer.elements();
    Enumeration e_room=SchatziServer.buffer_room.elements();

    while(e.hasMoreElements()){
        try
        { Client c = (Client)e.nextElement();
          //클라이언트의 매개변수는 3개이다.
          if(gubun.equals( e_room.nextElement() ))
            c.oos.writeObject(d);

            -- 생략 --
        }
    }
}

```

그림 3-7 서버 시스템 모듈 코드 내용

(3) 갱신시스템 시스템 모듈

갱신시스템 모듈은 그림 3-5와 같이 데이터베이스의 데이터의 변경이나 갱신이 있을 경우 가상데이터베이스와 동기화 하는 모듈이다. 갱신시스템 모듈의 두 가지 설계방향을 설정해야 한다.

첫째, 데이터베이스의 기초 자료와 가상 데이터베이스의 동기화를 처리한다.

둘째, 가상데이터베이스의 무효성을 점검하고, 관리한다.

Step 1 데이터베이스의 갱신정보를 읽어온다.

Step 2 데이터베이스의 정보를 가상 데이터베이스로 검사한다.

Step 3 정보를 동기화 시키고 주기적으로 Step 1을 반복 수행한다.

코딩 내용에서 중요 부분은 다음 그림 3-8과 같다.

```

class managerq
{
    -- 생략 --

    // Step 1. DB 접속한다.
    // Step 2. 3초마다 작업을 반복한다.

    -- 생략 --

    while(true) {
        //Step 3. 변경값이 있는지 데이터를 가져온다.
        v_rset = v_stmt.executeQuery (
            "select * from jk_history where user_info = 0 order by data_add " );

        //Step 4. 변경값이 있는지 데이터를 가져온다.
        if(v_rset.next()){
            sMem_info      = v_rset.getString(1);
            try{
                File f1 = new File("../data\\"+sMem_info);
                System.out.println(" f1.delete(): "+f1.delete());
                // 처리된 결과를 데이터베이스에 갱신한다.
                v_stmt.executeUpdate(
                    " update jk_history set user_info =1 where " +
                    jh_gubun +' ' + sunique_value      + " and user_info =0 " );
            } catch (Exception e) {
                System.out.println(e);
            }
            } else {
                System.out.println ("Query not Found... ");
            }
            try{
                Thread.sleep( 3000 );
            } catch (InterruptedException e) { }
        }
    }
}

```

그림 3-8 갱신시스템 모듈 코드 내용

(4) 응용프로그램 모듈

응용프로그램은 그림 3-5과 같이 인트라넷 사용자를 위한 자료의 변경 내용을 처리하는 모듈이다. 웹에서 자료를 갱신하면, 변경된 자료가 별도로 갱신시스템 모듈이 처리할 수 있는 방식으로 구성된다. 응용프로그램 모듈 두 가지 설계방향을 설정해야한다.

첫째, 데이터베이스의 기초 자료를 입력한다.

둘째, 갱신시스템이 동기화 모듈을 구동할 수 있게 기능을 제공한다.

3.4.2 데이터베이스 테이블

(1) 사용자 정보 테이블(jk_info)

이 테이블은 사용자의 최신 정보를 보관한다. 구분(JL_GUBUN)이 기본 키이며, 무효성 검사(USER_INFO)가 0일 경우 현재 정보를 사용하고 있는 것이고 1일 경우는 갱신시스템에 의해 사용되어지지 않는 것을 의미한다.

이 정보에 따라 사용자가 웹브라우저로 자료의 변경여부를 관리하며, 이에 따라 응용프로그램에 의해 등록관리 되면, 서버 시스템 모듈이 관련 정보를 클라이언트에게 보내게 된다.

이때, 서버와 클라이언트를 오가는 정보는 실시간을 고려하여 최소한의 정보를 보낼 수 있게 설계 및 정의한다.

표 3-2 경매 테이블 구조

KEY TYPE	COLUMN NAME	DATA TYPE	COMMENT	SAMPE DATA
PK,FK	ji_gubun	varchar(4)	구분	A001
	ji_jongmok	varchar(2000)	품목명	SAMSUNG COMPUTER
	ji_high	number(7)	최고가	10000
	ji_low	number(7)	최저가	1000
	ji_new	number(7)	현재가	8000
	user_info	varchar(1)	사용여부 0:사용,1:미사용	0
PK	data_add	data	마감시간	2001-10-04 10:05:05

(2) 사용자 갱신관리 기록 테이블(jk_history)

이 테이블은 인트라넷에서의 데이터베이스에 정보변경 기록을 보관한다. 응용프로그램 모듈에서 입찰가를 입력하면 자동으로 등록이 된다. 이때 주기적으로 발생하는 갱신시스템이 이 정보에 따라 데이터베이스와 가상 데이터베이스와의 동기화를 유지한다. 예를 들어, 사용자가 자신의 웹 정보를 조회하고 있다고 가정했을 경우, 인트라넷 사용자가 입찰금액을 신청하면, 데이터베이스의 한 레코드가 추가된다. 이때, 갱신시스템 모듈이 작동하고, 웹 요구자는 신뢰성 데이터를 가지게 된다. 주기적으로 갱신시스템은 기록 테이블을 참조하기 때문에 무효성 (USER_INFO)검사에 따라 가상 데이터베이스의 자료도 갱신이 된다.

표 3-3 경매 기록 테이블 구조

KEY TYPE	COLUMN NAME	DATA TYPE	COMMENT	SAMPE DATA
PK	jh_gubun	varchar(4)	구분	A001
PK	jh_new	number(7)	입찰금액	1000
	user_info	varchar(1)	갱신유무 0:사용,1:미사용	0
PK	data_add	data	기록일자	2001-10-04 12:01:01

제 4 장 웹사이트 갱신시스템 모의실험

본 시스템은 대규모 웹사이트의 갱신시스템을 구현하기 위한 것이다. 인터넷 사용자의 자료 갱신에 있어 보안에 관련된 인증과정 및 사용자, 관리자, 열람자 화면은 별도로 구성 및 구현하지 않고, 상위메뉴 형식으로 구현했다. 시스템을 시뮬레이션 하기 위해서는 주요 평가항목을 선정해야 한다. 첫째로, 인터넷 사용자가 데이터베이스를 변경할 경우 주기적인 정적-동적 웹 페이지를 통한 결과와 갱신시스템에서 구현되는 결과의 처리를 대상으로 결과를 확인한다. 둘째, 동일 질의에 대한 사용자 수의 증가에 따른 데이터베이스 처리 지연시간을 대상으로 평가를 통해서 성능을 검정한다.

4.1 실험방법 및 환경

본 논문에서 구현된 대규모 웹사이트 갱신시스템의 성능을 분석하기 위해서 대상선정에 따라 실험을 구현 측정하며, 질의에 대한 결과 시간을 일정 횟수만큼 반복 수행하여 갱신된 내용을 검정한다. 클라이언트 수를 증가해서 처리 프로그램 부하를 측정하여 증감되는 것을 확인하고, 이러한 실험을 하기 위한 서버 환경은 표 4-1과 같다.

표 4-1 모의 실험에 사용된 시스템 사양

네트워크 프로토콜		TCP/IP SOCK
하드웨어 환경	CPU	Pentium III 700 MHz
	RAM	128Mbyte
소프트웨어 환경	OS	Windows 2000 Server
	Web Server	IIS 5.0
	DB Server	Oracle 8.0.1(타 서버)

갱신 시스템을 구현한 TEST용 웹페이지는 경매물품 현황, 신규종목 등록, 물품경매 신청, 실시간 조회 등으로 구성되며, 그림 4-1과 같다.



그림 4-1 갱신시스템 모의실험용 웹페이지 화면

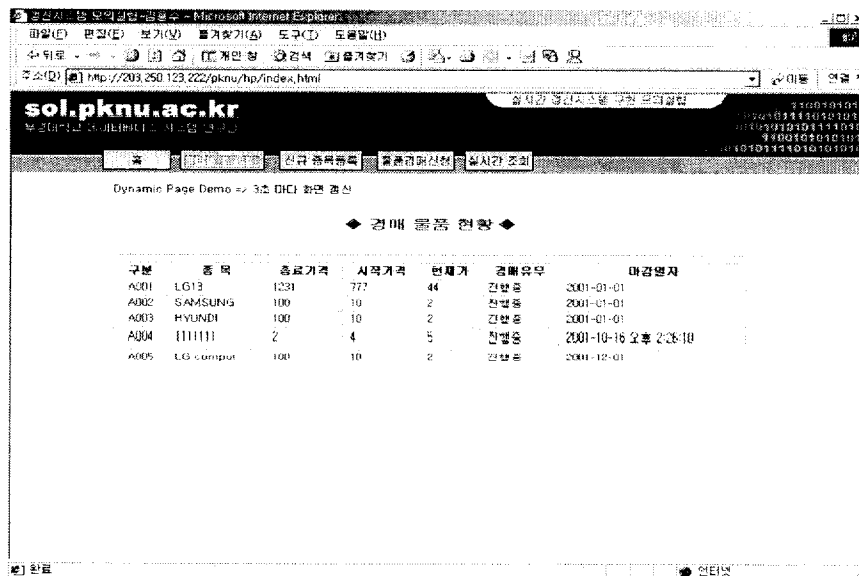


그림 4-2 동적 웹페이지인 경매 물품현황 실행 화면

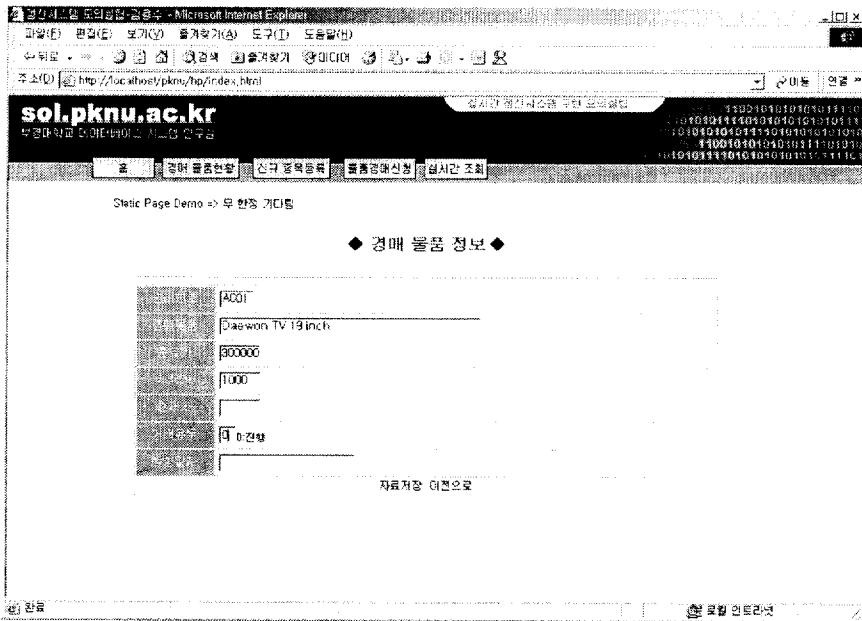


그림 4-3 정적 웹페이지인 신규종목등록 실행 화면

그림 4-2는 인트라넷 사용자를 위해 실시간으로 데이터베이스에 접속하여 전체 경매물품현황을 모니터링 하기 위한 화면이다. 그리고 경매에 대한 신규 종목이 발생할 시 관리자에 등록되는 부분은 그림 4-3과 같다.

그림 4-4에서와 같이 인트라넷 사용자가 경매물품 금액을 입력한 후 저장 (신청)을 선택하면 경매 기록 테이블에 신청 금액이 기록된다. 이때 주기적으로 실행되고 있는 갱신시스템의 모듈인 리플래쉬 매니저가 모든 과정을 감시하여 이 정보를 가지고 사용자 경매테이블을 최신 내용으로 갱신한다. 이와 동시에 가상테이블 기록 정보를 갱신하기 위해 이전 자료에 대한 복사본을 삭제한다. 이때, 그림 3-5와 같이 서버시스템 모듈이 변경된 부분에 대한 신뢰성 있는 자료 전송을 위해 기록테이블을 생성하고, 경매 정보 요청자에게 갱신된 사항을 실시간으로 PUSH한다. 이러한 작업은 서버머신에서 주기적으로 모든 과정을 감시하는 리플래쉬 매니저가 갱신정보에 대한 처리를 담당한다. 그림 4-5는 경매정보를 열람하는 화면이다. 갱신시스템의

정책에 따라 실시간 자료가 서버시스템 PUSH에 의해 사용자에게 전달된다. 이 시스템은 ONE Process로 운영된다

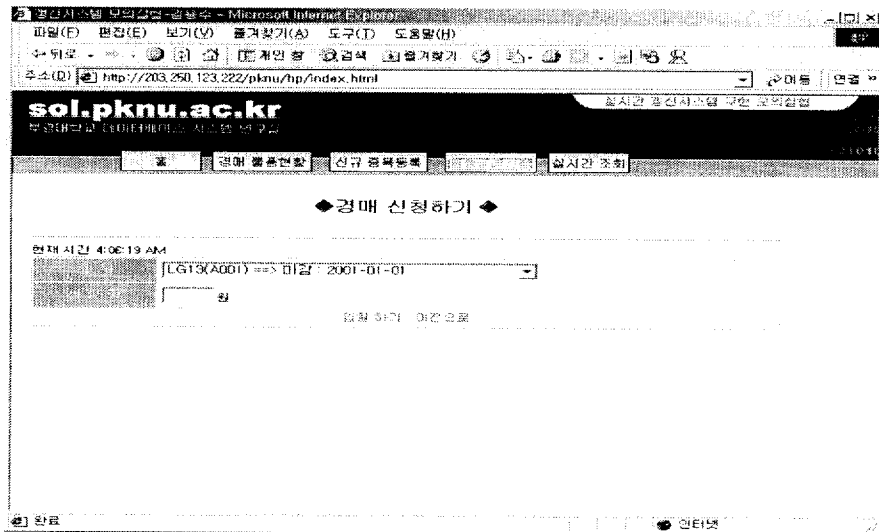


그림 4-4 정적 웹페이지인 경매 신청하기 실행 화면

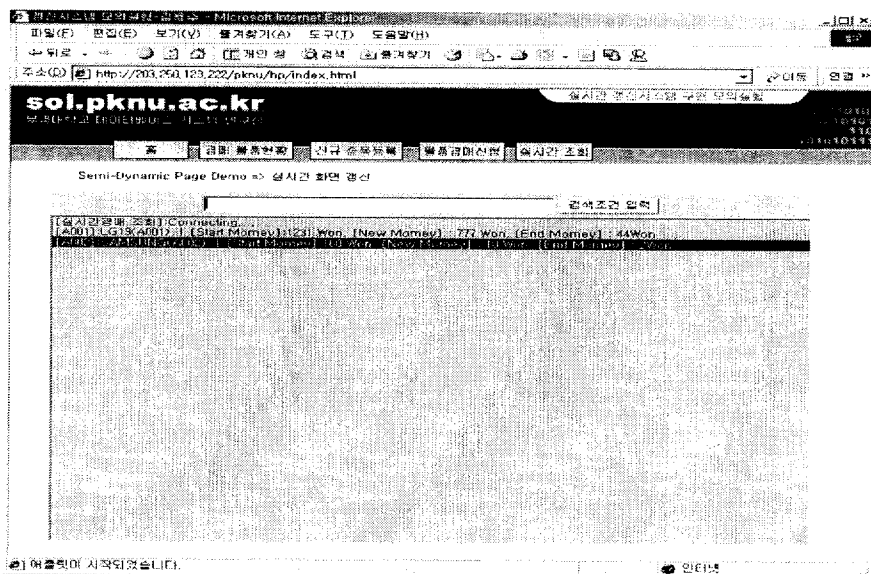


그림 4-5 실시간 조회 화면

모의실험 환경은 표 4-1과 같이 클라이언트는 Pentium III 700MHz, 128M Byte 메모리와 운영체제는 Windows 2000 서버로 하고, 데이터베이스는 다른 서버쪽에 구축해서 연결하고 실험한다.

4.2 실험결과 및 평가

실험결과 사용자의 접속수가 많아질수록 응답속도는 데이터베이스일 경우 점점 증가하고, 가상 데이터베이스에서 전송하는 자료는 점점 감소함을 알 수 있다. 보다 빠른 실험결과 성능 향상을 위해서 최적화하는 질의를 분류하는 방법과 실시간 동기화 처리 방법의 개선이 문제 시 되었다. 매번 데이터 기초정보가 바뀔 때마다 웹 페이지에 관련된 가상 데이터베이스가 갱신되어야 하므로 신뢰성이 부족했다.

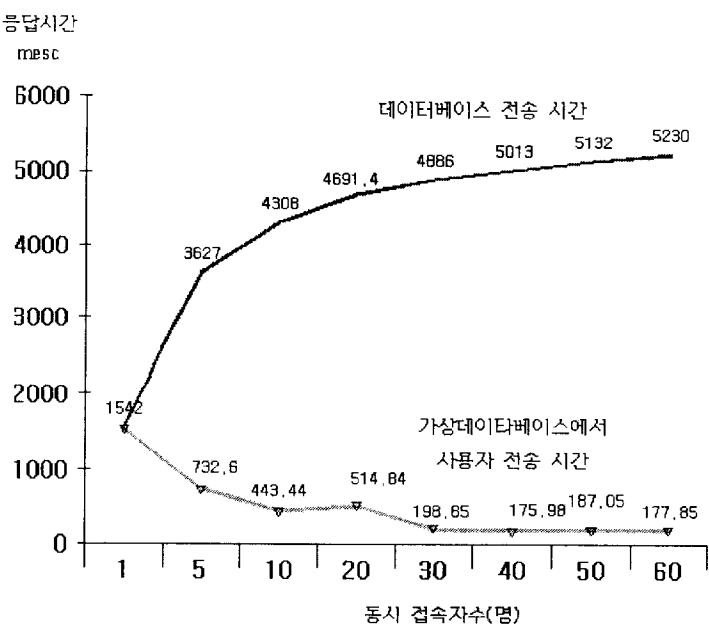


그림 4-6 사용자 증가(스래드)에 따른 시스템 응답 시간측정

따라서 성능 최적화를 위해 상기 결과를 적용하고, 소스를 최소화해야 하며, 리프레시 매니저의 역할도 최소화하는 것이 매우 중요하다.(본 논문에서는 수치적인 결과를 위해 500msec 처리 여유 시간을 두었다.)

제 5 장 결론 및 향후과제

본 논문에서는 JAVA2 플랫폼 기반의 JDBC를 이용한 경매시스템을 설계하여 대규모 웹사이트의 실시간 갱신시스템이 가능하도록 구현해 보았다. 최근 웹사이트는 점차 대형화되고 있는 추세이므로, 본 갱신시스템은 다음과 같은 특징을 갖는다.

첫째, 많은 사용자의 질의로 인한 트래픽 증가로 최신자료 획득을 해결한다.

둘째, 한번의 질의로 최신정보를 사용자에게 가장 빠르고 효율적으로 제공한다.

셋째, 대규모 웹사이트의 성능을 크게 향상시킨다.

이러한 연구의 배경과 필요성을 기초로 스포츠 중계 실시간 처리시스템, 투표 처리 시스템, 대학입시 실시간 접수현황, GIS와 관련한 위치추적 시스템, 주식정보, 각종 예매 서비스 등 대규모 웹사이트에 갱신시스템을 적용할 수 있다. 이를 이용하면, 인터넷 정보의 인프라를 실시간으로 획득할 수 있다. 웹사이트의 점차 대형화 추세에 있기 때문에 향후 가상 데이터베이스 압축 기술 등 다양한 디스크 알고리즘을 적용하여, 데이터베이스 시스템 부하를 최소화하기 위해 지속적으로 연구할 계획이다.

참고문헌

- [1] 최윤철, 한탁돈, 조성배 공저, “실용 인터넷” pp. 16-20, 1999
- [2] 한국인터넷 정보센터, “2001년 인터넷통계 월보”, 2001.10
- [3] 김화중 저, “컴퓨터 네트워크 프로그램” 2000.12
- [4] 이현우, 김형국, 홍성민 공저, “Java Programming Bible 2” 2000.2
- [5] Java Book, “<http://www.jabook.org>”
- [6] Kevin Mukhar, “Beginning Java Database”, 2001
- [7] George Reese, “Database Programming with JDBC and Java, 2nd Edition”, 2000
- [8] Madhu Siddalingaiah, Stephen D. LockWood, “JAVA HOW-TO”, 1997
- [9] Reese, Georg, “Database Programming with JDBC and JAVA”, 1998
- [10] Don Balse, “Java Programming with Oracle JDBC”, 2001
- [11] Y. N. Ko, T. N. Kim, Y. I. Kim, D. M. Lee, “Implementation and Performance Analysis of Wired & Wireless Real-time Auction System based on Java 2 Platform”, K.I.S.S., 2001.10
- [12] H. F. Liu, W. K. Ng, and E. P. Lim. Keeping a Very Large Website Up-to-date: Some Feasibility Results. International Conference on Electronic Commerce and Web Technologies (EC-Web '2000), Greenwich, UK, September 2000.

감사의 글

오늘 이 글을 남길 수 있게 도와주신 모든 분들께 깊은 감사의 뜻을 전합니다. 지난 2년의 대학원 생활동안 지속적인 보살핌과 배려 그리고 성심성의껏 지도 편달해 주신 지도교수이신 조우현 교수님께 진심으로 감사를 드립니다. 부족한 저의 논문을 심사하시면서 세심한 지도와 많은 관심을 보여주신 정목동 교수님, 서경룡 교수님께도 깊은 고마움을 표합니다.

또한 대학원 과정을 마칠 수 있도록 항상 격려와 편의를 제공 해 주신 동명정보대학교 종합정보센터 정정수 소장님, 문수근 과장님께 감사를 드리며, 서만승 교수님, 김정인 교수님 그리고, 종합정보센터의 모든 가족에게 진심으로 감사를 드립니다.

부경대학교 데이터베이스 연구실 정섭, 호택, 석희, 종우, 주민, 형국 모든 분과 박창수 선배와 동명정보대학교 동영, 태남, 영남과 친구 허태식에게 여러모로 고마움을 표한다.

끝으로 오늘이 있기까지 물신양면으로 도와주신 부모님과 식구들, 사랑하는 나의 아들 도환, 동현 그리고, 아내에게도 다시 한번 고마움을 전합니다.

2001 년 12 월

김 용 수 올림