

DB 서버 부하를 고려한 EEIS 시스템의 설계 및 구현

최병진[†], 김창근^{††}, 김봉기^{†††}, 김홍준^{†††}

진주산업대학교

Design and Implementation of the EEIS Considering the Load of DB Server

Byeong-Jin Choi[†], Chang-Geun Kim^{††}, Bong-Gi Kim^{†††}, Heung-Jun Kim^{†††}

Chinju National University

E-mail: bjchoi@cjcc.chinju.ac.kr, cgkim@cjcc.chinju.ac.kr, bgkim@cjcc.chinju.ac.kr, thinkthe@cjcc.chinju.ac.kr

요 약

기존 인터넷에 의한 대학입시정보 시스템은 네트워크 및 DB 서버에 과부하가 발생하지 않게 하기 위하여 특정 검색키에 의존하거나, HTML 문서편집에 의한 지원 등의 획일화된 검색 시스템을 제공할 뿐 사용자의 욕구 충족에는 미흡한 부분이 많다. 이러한 문제점을 개선하기 위하여 본 논문에서는 다수의 사용자가 다량의 데이터를 DB 서버에 요청 시 발생하는 DB 서버 속도저하 및 과부하 문제를 웹서버와 DB 서버 사이에 VVS(Virtual View Server)를 이용하여 외부 Client와 DB 서버의 가교역할을 수행하며 DB 서버 과부하 방지기능을 수행하는 EEIS(Entrance Examination Information System) 시스템을 구축함으로써 해결하고, 사용자에게는 실시간 자료검색을 제공함으로써 이용에 대한 편리성 및 다양성을 고려한 시스템을 구축하였다. 또한 EEIS 시스템은 DB 서버의 과부하를 개선함으로써 내부사용자에 대하여도 네트워크 속도 및 효율성을 증대시켜 다양한 정보를 다량의 데이터와 함께 실시간 제공함으로써 대학의 신뢰도 제고는 물론 사용자의 욕구충족 및 편의성을 고려한 시스템으로 설계 및 구현하였다.

ABSTRACT

Current Internet system of the entrance exam information to the university that is depending on searching key to solve the overloaded problem in the network and DB server or other tools to support HTML edit, haven't satisfied user's wants by supplying uniformed searching system. So this thesis will establish EEIS(Entrance Examination Information System) to prevent database overload phenomena when many users request a great amount of data at the same time and improve the decrease of speed and overload problem in DB server. EEIS play a role of bridge between outside client and DB server by placing VVS(Virtual View Server) between web server and DB server. By that method this system give users several usefulness in convenience and variety by supplying realtime data searching function to user. EEIS also give inner system manager more efficiency and speed in control the management system by solving those problem. This system is design and implementation to satisfy user's desire and give them more convenience and bring up the confidence of university that adopt this system at the end.

Key Word : Overload, DB Server, Searching, VVS(Virtual View Server)

1. 서론

오늘날 인터넷의 보편화로 인하여 언제 어디서나 필요한 정보를 적시에 제공받을 수 있는 정보 시스템이 요구되고 있다[1]. 그러나 기존의 인터넷에 의한 다량의 DB 서버 연동을 요구하는 원격 검색지원 시스템은 네트워크 및 DB 서버에 과부하가 발생하지 않게 하기 위하여 특정 검색키에 의존하거나, HTML 문서편집에 의한 지원

등의 획일화된 검색 시스템을 제공할 뿐 사용자의 욕구 충족에는 미흡하다[5].

본 논문에서는 이러한 문제점을 개선하기 위하여 웹서버와 DB 서버 사이에 VVS(Virtual View Server)를 이용하여 구현하였으며, VVS는 EEIS(Entrance Examination Information System) 시스템에서 주요부분을 차지하고 있다.

VVS는 별도의 단독 시스템을 요구하는 게 아

나라 웹서비스가 가능하고 디스크의 여유공간만 있다면, 어떠한 웹서버라도 구축이 가능하다.

VVS의 필요기능은 먼저 DB 서버와 동일한 DB 엔진 및 관련 프로그램을 설치한 후 CGI 프로그램에서 필요로 하는 정보만 골라 View Table을 작성하여 DB 서버의 안전성 및 과부하를 예방하는 역할을 수행한다. 또한 EEIS 시스템은 내부사용자에 대하여도 네트워크 속도 및 효율성을 증대하였고, 외부 지원자의 검색정보 요구에 대하여는 다양한 정보를 실시간 제공함으로써 대학의 신뢰도 및 사용자 편의성을 향상시킨 시스템을 설계 및 구현하였다[8][10].

II. 관련연구

2.1 CGI

CGI(Common Gateway Interface)는 웹서버와 프로그램간에 서로 데이터를 주고받기 위한 방법을 정의한 표준규약을 말한다.

여기에서 Common Gateway는 이질적인 두 프로그램간에 데이터를 주고받기 위한 공통된 데이터 입출력 통로를 말하는 것이며, Interface는 이러한 입출력 통로에 대한 프로그래밍 인터페이스를 말한다.

공통된 데이터 입출력 통로란, 웹서버와 CGI 프로그램이 서로 데이터를 주고받는 입출력 통로를 의미하며, 보통 환경변수나 표준 입출력을 이용해 이러한 수행을 한다[4][6][7][9][11].

그림 1은 전체적인 HTML & CGI & DB 인터페이스 흐름을 나타낸다. 즉 웹브라우저가 어떤 자료의 URL을 요청하면, 웹브라우저는 요청헤더(Request Header)를 발생시켜 웹서버에 전달한다.

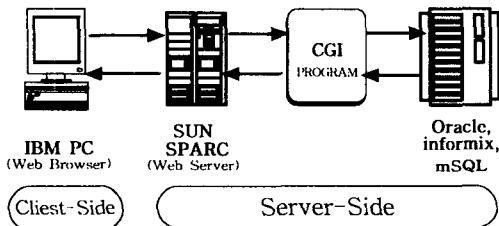


그림 1 HTML & CGI & DB 인터페이스 흐름도

이때 웹서버는 웹브라우저의 요청을 받아들여, 그 요청이 자체 내의 HTML 문서인지 CGI 프로그램의 실행인지 분석한다. 이때 CGI 프로그램 실행이라면 웹서버는 CGI 프로그램 실행을 위해 별도의 프로세스를 생성시켜 실행한다[4].

CGI 프로그램과 일반 프로그램의 차이점은 첫째, 일반 프로그램들은 자체 시스템에서 실시간으로 실행되어 프로그램이 메모리 상에 항상 저장

된 상태를 유지하며, 다음 단계를 진행할 때 메모리에 기록된 변수를 참조하는데 비해, CGI 프로그램은 HTTP를 기반으로 하는 웹의 특성으로 인해 어떤 자료를 요청(Request)하고 응답(Response)을 기반으로 하는 형식으로 처리하며, 최종 응답 후에 접속을 종료한다.

둘째로 일반 프로그램들은 클라이언트 쪽에서 실행되는 프로그램인 데 반해, CGI 프로그램은 웹서버가 자신의 기능을 확장하고자 실행하는 웹서버 종속 프로그램이다.

이러한 특성으로 웹서버와 CGI 프로그램이 서로 통신할 때에는 통신할 수 있는 규약이 필요한데 이때 CGI를 사용하게 된다. CGI 프로그램 개발도구로는 CGI 프로그램 자체가 단순히 웹서버와 인터페이스 할 수 있는 기술이기 때문에 특별히 CGI 프로그래밍을 위한 언어는 없고, 일반적으로 널리 사용되는 C나 C++ 같은 컴파일러 언어에서부터 Perl이나 쉘 같은 스크립트 언어까지 표준 입출력 기능을 제공한다면 어떤 언어든 문제가 되지 않는다[4][6][7][9].

2.2 CGI 와 DB 시스템과의 연계

CGI 와 DB 시스템의 연동 흐름도는 그림 2 와 같은데, CGI 와 DB 시스템을 연결시킴으로써 사용자는 복잡한 명령들을 익힐 필요가 없고, 단지 어떤 정보가 필요할 때 필요한 정보에 대한 요청으로 DB 자원들을 웹 상에서 서비스 받을 수 있다.

DB 시스템에 접근하는 방법은 크게 두 가지로 DB 시스템에서 자체 지원해 주는 API 함수를 이용하는 방법과, 중간에 DB 게이트웨이를 하나 더 두는 방법으로 나뉘 볼 수 있다.

첫째 DB API 함수를 이용하는 방법은 CGI 프로그램이나 웹서버 API 프로그램에서 직접 DB 함수를 호출해 DB 서버에 접근할 수 있도록 만들어 줄 수 있다. 대표적으로 인포믹스는 ESQL/C를 이용해 C 프로그램 안에 필요 DB를 정의하고, 정의된 DB를 다시 C로 확장한 다음 컴파일 해서 실행 파일을 생성시킨다

둘째 게이트웨이를 두는 방법은 각 DB 회사에서 제공하는 backend 프로그램을 실행시켜 결과를 얻는 방법이다.

backend 프로그램이란 DB 시스템을 실제 조작하는 시스템에서 DB 핸들링을 쉽게 해주기 위해 중간에 설치하는 독립된 DB 처리 프로그램을 말한다.

예를 들어 mSQL에서는 W-mSQL 이라는 backend 프로그램을 제공하는데, 직접 프로그래밍을 하지 않고도 backend 프로그램이 직접 mSQL DB에 질의해서 결과를 받아 사용자에게 넘기는 작업을 수행해 준다[4][8][10]

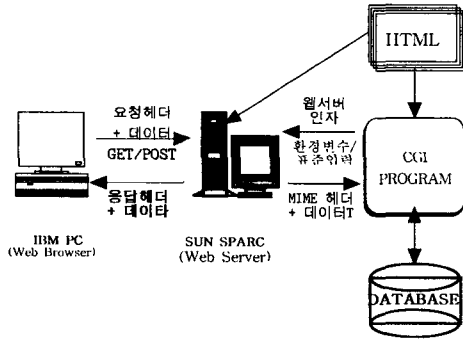


그림 2 CGI 와 DB 시스템과의 연동 흐름도

그림 3은 mSQL API를 배경으로 실제 mSQL API가 어떤 순서로 C 프로그램에 내장될 수 있는지에 대해 알려준다.

가장먼저 msqlConnect()로 mSQL DB 서버에 TCP/IP상으로 접속한다. 그런 다음 사용자 DB를 msqlSelectDB()로 선택하고 실제 원하는 자료에 대한 질의를 msqlQuery()에 실어 전달한다.

질의 수행이 완료되면 원하는 자료에 대한 값을 캐시 버퍼에서 실제 메모리로 가져오는 작업을 해야 하는데, 이때 msqlStoreResult() 함수를 사용하게 된다.

실제 메모리로 가져온 테이블의 행을 msqlNumRows()나 msqlFetchRow()함수로 화면에 출력한 다음 위해 할당되었던 메모리를 msqlFreeResult()로 해제시키고 DB 접속을 종료한다 [4][10].

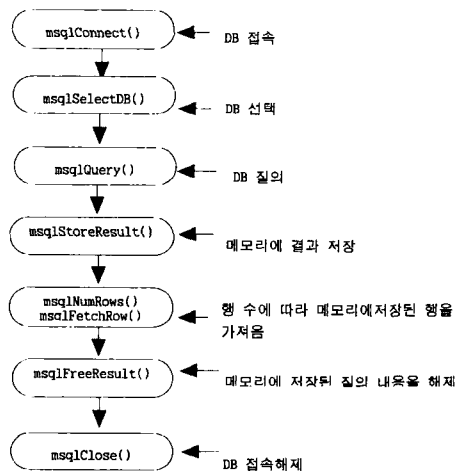


그림 3 mSQL API 흐름도

2.3 XML

XML(eXtensible Markup Language)은 W3C(World Wide Web Consortium)에서 기존의 HTML과 SGML(Standard Generalized Markup Language)이 갖는 단점을 보완하여 SGML이 갖는 장점인 구조화된 문서를 정의하여 태그도 자유롭게 정의할 수 있고, HTML이 갖는 장점인 인터넷상에서 손쉽게 하이퍼미디어 문서를 제공할 수 있는 장점을 갖도록 제안한 마크업 언어(Markup Language)이다[2][3][11][12].

III. 시스템 분석 및 설계

3.1 시스템 구현 환경

EEIS 시스템은 크게 웹서버와 VVS, DB 서버로 나눌 수 있는데, EEIS 시스템의 구현 환경은 그림 4와 같다.

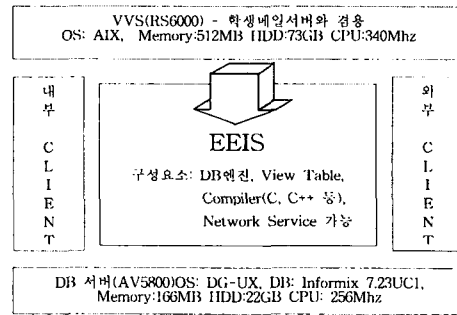


그림 4 EEIS 시스템 환경

먼저 모든 지원 가능한 Master DB를 포함하고 있는 DB 서버와, DB 서버를 중심으로 내부에서 응용되어지는 내부 Client, 또한 다수의 정보요청자인 외부 Client로 나누어진다. 내부 Client는 주로 자료처리를 목적으로 활용되어지며, 외부 Client는 정보요청 처리로 사용되어진다.

또한 EEIS의 주요부분인 VVS는 외부 Client와 DB 서버의 가교 역할을 수행하며 DB 서버의 과부하 방지 기능을 수행하게 되는데, VVS는 별도의 비용으로 시스템을 구축하는 것이 아니라, 웹서비스가 가능하고 DB 엔진을 구축할 수 있는 디스크의 여유 공간만 있으면 어느 웹서버나 가능하다.

다음은 VVS 시스템의 원활한 기능을 수행하기 위하여 변경이 되어야 할 사용자 프로 파일, 원격지 DB 접근 관리 파일, 웹서비스 및 CGI 관련 파일이다.

. profile :
INFORMIXDIR=/usr/informix

```
export INFORMIXDIR
PATH=$INFORMIXDIR/bin:/usr/webroot/usr-bin:$PATH
#INFORMIXSERVER=ncjav /* DB Server */
INFORMIXSERVER=cjst /* VVS Server name */
```

```
. sqlhosts :
cjst  onsoctcp  cjst      sqlexec
ncjav onsoctcp  cjav      sqlexec
```

. httpd.conf :

```
ScriptAlias /cgi-bin/ "/usr/webroot/http-docs/cgi-bin/"
<Directory "/usr/webroot/http-docs/cgi-bin">
    AllowOverride None
    Options ExecCGI
    Order allow,deny
    Allow from all
</Directory>
```

3.2 시스템 구성

DB 서버를 고려한 EEIS 시스템의 구성도는 그림 5와 같이, EEIS 정보를 검색할 수 있는 웹서버와 정보를 제공하는 DB 서버, 그리고 웹서버와 DB 서버 사이에서 DB 검색 정보를 제어하는 VVS로 구성되어 있다

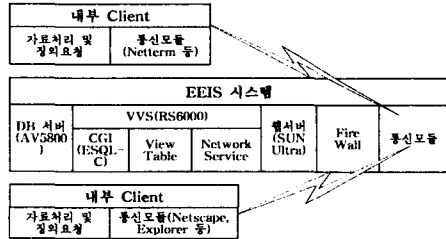


그림 5 EEIS 시스템의 구성도

3.3 시스템 설계

그림 6, 그림 7과 같이 기존의 입시정보 시스템은 네트워크 및 DB 서버 속도저하 등을 우려하여 대부분 수험번호, 주민번호, 성명 등의 단순한 키에 의한 검색지원이나 아니면 HTML 문서로 편집하여 제공하는 것이 대부분이다.

이렇게 DB 서버를 연동하여 사용하는 경우 DB 서버에 부담되는 과부하를 고려할 수밖에 없는 주요 원인은 DB 서버 접근에서부터 결과를 가져오는데 까지 소요되는 시간과 서버에 가중시키는 부담 때문이다

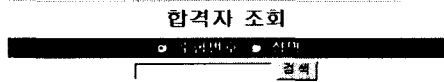


그림 6 키 입력 방식에 의한 검색 화면

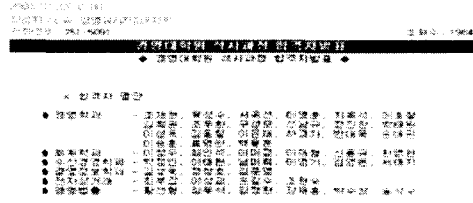


그림 7 HTML로 작성된 검색결과 화면

일반적으로 정보요청자인 Client 측면에서는 HTML 문서에 비해 DB 서버를 통한 정보수급이 실시간 데이터 획득 및 신뢰성에 대한 장점이 있는 반면 보다 많은 시간과 비용이 소요되는데, 이러한 이유는 DB 종류에 관계없이 DB접속을 통한 결과처리에서는 DB에서 제공되는 API 함수 또는 이와 유사한 단계가 필수적으로 사용되어야 하기 때문이다.

API 함수를 사용하는 경우에는 CGI 프로그램 내에서 DB 종류를 결정하여 접속하고, DB를 선택한 다음 요청된 Client의 질의결과에 따라 질의하고 이렇게 질의한 내용을 메모리에 가져와 결과를 저장하게 한다. 각 Client는 이러한 과정을 반복함으로써 요청된 결과를 화면으로 받아보게 되는데, 이러한 부분이 DB 서버 과부하를 초래함은 물론 내부사용자에게도 DB 접속 속도저하 문제를 초래한다.

하지만 본 논문에서 구현한 EEIS 시스템은 다수의 사용자가 다량의 데이터를 DB 서버에 요청시 발생하는 DB 서버의 속도저하 문제를 해결하기 위해 VVS를 구축하여 다수의 정보요청 Client 중 항목 별 최초접속자만 DB 서버에 접속하고 나머지는 VVS에서 정보제공을 받을 수 있도록 구현하였다. 이렇게 함으로써, 내부사용자에 대하여는 네트워크 및 DB 서버 접속 속도를 증대하여 효율성을 증대함은 물론, 지원자의 검색정보 요구에 대하여는 지원 학과별 전체 검색, 지원 학교별 전체현황, 지원자의 후보순위 등 일목요연한 다양한 정보를 실시간 제공함으로써 대학의 신뢰도 제고 및 사용자 편의성을 고려한 EEIS 시스템을 설계하였다.

3.3.1 DB 서버 부하를 고려하는 CGI 제어 정책

그림 8은 CGI 내부에서 DB 서버의 부하를 감소하기 위한 제어 알고리즘인데, 다수의 정보검색 요청자인 Client들이 웹서버를 통해 DB 서버에 실시간으로 자료를 요청 시 항목별 최초의 DB 서버 접속 Client가 DB 서버에 접속하여 데이터를 VVS에 HTML 문서로 저장을 하면, 나머지 Client들은 DB 서버에 접속하지 않아도 VVS에

저장된 HTML을 참고함으로써 실시간 데이터를 열람하는 효과와 같은 기능의 제어정책을 설명한 것이다.

즉 다수의 Client들이 지원자의 지원학과에 대한 모든 정보를 요청하거나, 지원고교에서 출신학교 지원자들에 대한 종합검색을 요청할 때 일반적 상황에서는 다량의 자료요청 때문에 DB 서버 및 내부 사용자들에게 업무 마비상태까지 초래하기도 한다. 이러한 문제가 발생하는 주요 원인은 유 종류에 관계없이 DB 접속을 통한 결과처리에서는 Client에서 데이터를 요청할 때마다 DB에서 제공되는 API 함수 또는 이와 유사한 흐름을 반드시 수행하여야 하기 때문에 필수적으로 소요되는 시간이 많으며, 이러한 과정이 Client마다 발생된다면 다수의 Client가 다량의 데이터를 요청할 때는 DB 서버에 주는 부담은 상당하며, 지나치면 DB 서버에 과부하가 발생하여 치명적인 접속 오류를 발생할 수도 있다. 하지만 본 논문에서 구축한 EEIS 시스템은 CGI 제어 정책으로서 DB 서버 접속을 통제하고, 효율적인 관리를 병행함으로써 이러한 문제를 해결한다.

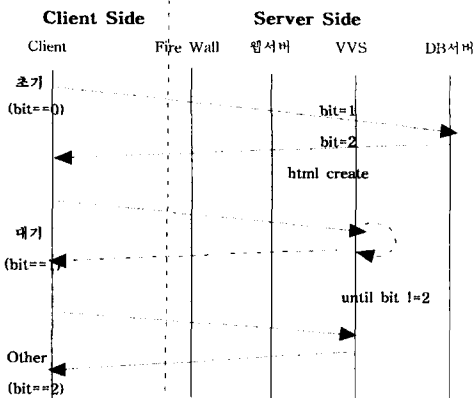


그림 8 Client와 EEIS 간의 상호동작 프로토콜

즉 분야별 초기접속 Client만 DB 서버에 접속하며, 초기접속 Client가 VVS에서 HTML 코드를 생성시키는 임무를 완료하기 전까지는 나머지 접속 요청 Client는 VVS에서 대기 상태가 된다. 이후 초기접속 Client가 HTML 코드 생성을 완료하게 되면, 나머지 대기 상태의 Client 및 모든 요청 Client는 VVS의 HTML 코드만을 참조하게 되는데, 이러한 모든 과정을 자동적으로 시스템에서 처리함으로써 DB 서버의 과부하를 해소한다..

IV. 구현 및 평가

4.1 EEIS 시스템 구현

본 논문에서 제안한 EEIS 시스템의 구현 방법으로는 그림 9와 같이 일반적인 인터넷상에서의 DB 접근 방식인 웹서버를 경유하여 DB 서버에 접근하는 방식에서, 웹서버와 DB 서버 사이에 VVS를 구축(그림 10)하였다.

그리고 VVS 시스템 내의 CGI 프로그램에서는 DB 서버 접속 시 통제 기능을 부여함으로써 초기 사용자 이외에는 DB 서버에 접속하지 않아도 데이터를 실시간 Client에서 조회가 가능하도록 시스템을 구현하였는데, 이렇게 함으로써 DB 서버의 직접적인 부하 감소는 물론, 외부 노출을 피함으로써 DB 서버 보안 문제도 개선하였다.

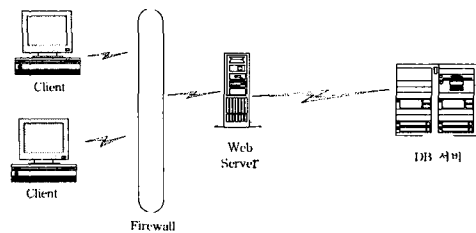


그림 9 인터넷상에서의 일반적인 DB 접근방식

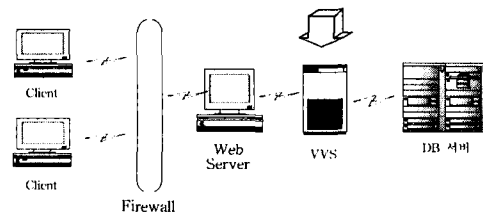


그림 10 EEIS 시스템의 DB 접근방식

그림 11은 EEIS 시스템의 메인 메뉴를 보여주는데 본 논문에서 구축한 EEIS 시스템은 단순한 키에 의한 입력방식이 아니라 그림 12와 같이 학과별, 지원 학교별 등 다량의 자료를 다수의 Client들에게 제공하여도 DB 서버에는 과부하가 발생하지 않는 시스템이 되도록 구현하였다.

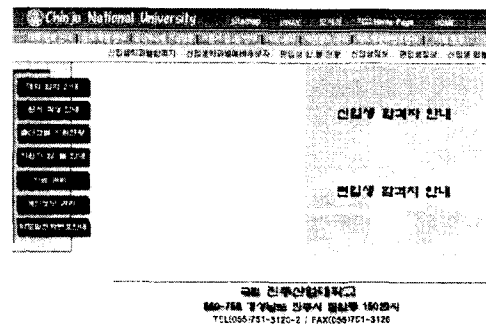


그림 11 EEIS 시스템의 메인 메뉴

● 합격자 수: 1차 합격자 발표일자: 2001년 02월04일

순위	성명	학번	성명	학번	성명	학번
1	주건	양민재현	장동원	1011052	김영진	1차원주과정
2	주건	양민재현	이동우	1011053	김영진	1차원주과정
3	주건	양민재현	김동원	1011054	김영진	1차원주과정
4	주건	양민재현	김동원	1011055	김영진	1차원주과정
5	주건	양민재현	김동원	1011056	김영진	1차원주과정
6	주건	양민재현	김동원	1011057	김영진	1차원주과정
7	주건	양민재현	김동원	1011058	김영진	1차원주과정

● 합격자 수: 2차 합격자 발표일자: 2001년 02월09일

순위	성명	학번	성명	학번	성명	학번
1	주건	양민재현	김동원	1011059	김영진	1차원주과정
2	주건	양민재현	김동원	1011060	김영진	1차원주과정
3	주건	양민재현	김동원	1011061	김영진	1차원주과정
4	주건	양민재현	김동원	1011062	김영진	1차원주과정
5	주건	양민재현	김동원	1011063	김영진	1차원주과정

그림 12 EEIS 시스템의 검색결과

4.2. CGI 내부 서버 제어

EEIS 시스템은 VVS 시스템 내부에서 DB 서버의 접속에 대한 서버접속 제어 정책을 구현하고 있는데, 서버제어 알고리즘은 그림 13과 같다. 우선 본 대학의 입시를 예로 들면, 먼저 관리자는 내부 Client에서 합격자 및 미 등록자에 대한 불합격 및 추가 합격자를 처리하면서 제어 파일의 io bit를 초기 상태(0)로 Set 시킨다. 합격자 발표 후 다수의 Client들이 한꺼번에 다량의 자료를 요청하면 io bit가 0인 상태에서 DB

서버에 분야별 접속하는 Client는 VVS에서 io bit를 확인한다. 확인 후 io bit가 초기 상태이면 io bit를 1로 set 후 DB 서버에 접속한다. 이때 io bit를 1로 set 하는 것은 다른 Client의 DB 서버 동시접속을 제한하기 위해서이다. 그리고 DB 서버에 접속하여 데이터 처리가 끝나면 정상작업 완료코드인 io bit를 2로 set 후 VVS 시스템에 HTML 코드를 생성시켜 이후의 접속 Client가 참조하도록 한다. 이때 만약 초기 접속 Client가 DB 서버에 접속을 정상적으로 완료하지 않고 또한 HTML 코드를 생성시키기 전의 상태(io bit==1)라면, VVS 시스템에서 대기상태가 되어 초기 접속 Client의 정상종료를 기다리게 되는데, 이때 실제로 대기하는 시간은 미미하다. 이렇게 함으로써 이후에 접속되는 Client들은 다량의 정보를 실시간으로 조회하면서도 DB 서버에 직접 접속을 하지 않기 때문에 DB 서버에 대한 접속 속도는 고려되지 않고, 단지 네트워크의 부하만 고려되며, 내부 사용자들은 평상시와 같은 DB 서버 접속 속도를 유지하면서 업무를 진행할 수 있고, 정보요청 Client들은 빠른 속도의 정보제공을 받을 수 있다.

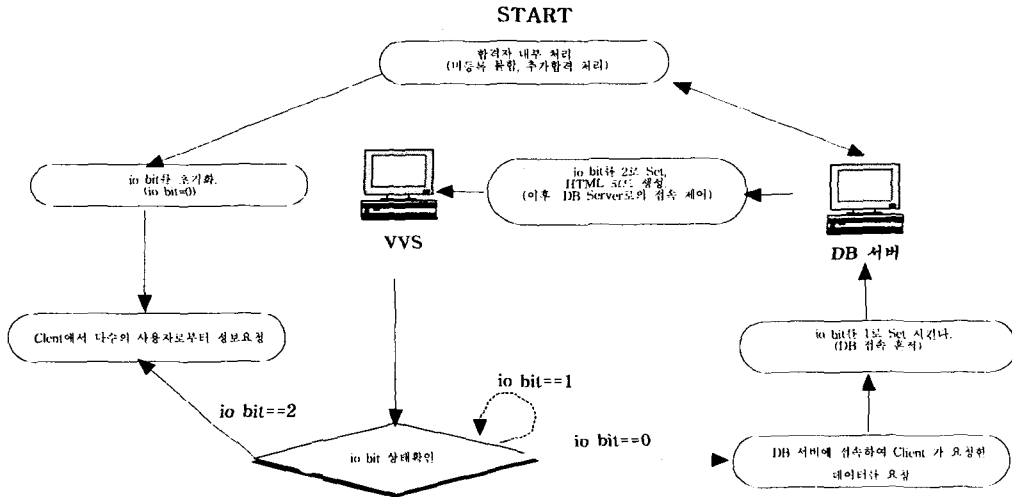


그림 13 CGI 내부 DB 서버 제어 알고리즘

4.2. 시스템 평가

본 논문에서 제안한 EEIS 시스템은 일반적인 CGI 기반에서 Client가 DB 서버로부터 요청한 자료를 얻기까지의 소요되는 시간이 평균 1,000건을 기준으로 10배 이상의 탁월한 속도차이를 보여 주었다.

시스템 평가는 DB 사용이 한가한 야간을 이용하여, 외부의 DB 사용자 접근을 통제된 상태에서 Client 수, 요청되어지는 자료건수 등에 변이를 주어가며 실험하였다. 평가에 대한 결과 처리는 Client IP Address,

Client 요청시간, Client 결과 출력시간들을 파일로 저장하여 평가하였다.

평가 결과에 의하면, 일반적인 DB 서버 접근방식을 수행하는 것에 비해 VVS를 통한 EEIS의 Client접속은 속도 차에서 평균 10배 이상의 결과 차이를 보여 주었는데, 이때 Client 측에서는 거의 Off-line 상태에서의 HTML을 열람하는 것과 같은 수준으로 자료를 받아볼 수 있었다.

그림 14의 그래프처럼 일반적인 CGI 기반으로 DB 서버에 접속 시 요청된 질의에 응답된 자료가 100 - 300건만 되어도 평균 5~6초의 응답시간이 소요되는 것에 비해 EEIS 평균 접속시간은 평균 1초 이내에 처리된다.

그러나 EEIS 시스템도 접속자 수가 아주 적거나, EEIS 시스템의 제일처음 접근하는 Client에게서는 일반방식과 별다른 차이를 나타내지 않았다.

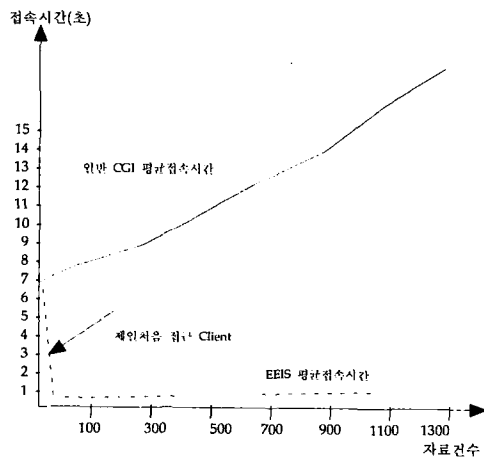


그림 14 CGI 일반접속과 EEIS 시스템의 Client 요청에 대한 답변속도 차이

V. 결론

기존 대학의 인터넷에 의한 대학입시정보 시스템은 지원자 및 주위관계자의 관심과 열정에 반해 네트워크 및 DB 서버의 부하를 고민하여 주민번호, 성명, 수험번호 등의 특정 검색키에 의존하거나, 아니면 HTML로 직접 작성하여 제공하는 등, 획일화된 검색 시스템을 제공하는 경우가 대부분이다. 특히 HTML 문서로 편집하여 제공 할 때에는 작성 오류로 신뢰성을 저하시키는 경우가 많으며, 작업 시간도 많이 소요된다. 이러한 문제점을 개선하기 위하여 본 논문에서는 DB 서버 부하를 고려한 EEIS 시스템을 구축함으로써, 다수의 사용자가 다량의 데이터를 DB 서버에 요청 시 발생하는 DB 서버 속도저하 문제를 해결하고 실시간 자료검색을 제공함으로써 이용자에 대한

편리성 및 다양성을 고려한 시스템이 되도록 노력하였다. 그리고 이러한 EEIS 시스템을 구축함으로써 DB 서버의 근원적인 과부하를 개선하였고, 내부사용자에 대하여는 네트워크 속도 및 효율성을 증대함은 물론, 타 업무도 원활히 진행될 수 있도록 하였으며, 지원자의 검색정보 요구에 대하여도 다양한 정보를 실시간 제공함으로써 대학의 신뢰도 제고 및 사용자 편의성을 고려한 시스템을 설계 및 구현하였다.

하지만 본 시스템은 다량의 Client가 다량의 자료를 요청할 때는 상당히 효율적이나 반면 개별 단위 자료의 검색 등에는 기존방식의 운용시스템과 큰 차이가 없다. 앞으로 이러한 분야에 광범위하게 활용이 될 수 있도록 지속적인 연구 및 관심이 요구된다.

참고문헌

- [1] 임종선, 주경수, "인터넷을 이용한 원격 학사정보 관리 시스템 개발", 한국멀티미디어학회 춘계학술발표논문집, 1998.
- [2] Fabio Arciniegas 저, 류성열 역, "XML 개발자 가이드", 인포북, 2001.
- [3] 김승훈, 김성환, 김성동, 최기호, "XML(SMIL) 기반의 멀티미디어 메일 저작도구의 설계 및 구현", 한국멀티미디어학회 추계학술발표논문집, 1999.
- [4] 김홍남, "CGI 파워 프로그래밍", 도서출판 대림, 1997.
- [5] 최준형, 김상권, 정유현, "인트라넷 구축 및 활용방안", 1998.
- [6] 이준석, "웹 브라우저와 CGI 프로그램 사이의 보안 통신을 지원하는 시스템 설계 및 구현", 한국정보처리학회 제6권, 1998.
- [7] 김은경, 황병연, "WWW 데이터베이스 인터페이스를 위한 UCM(United CGI Management) 시스템의 설계", 한국정보처리학회논문지 제6권, 1999.
- [8] KMK 정보산업연구원 편저, "SQL Server 7, 21일 완성", 삼각형프레스, 1999.
- [9] Ed tittel, Mark Gaitjer, Sebastian Hassinger, Mike Erwin, 저, 권영진, 황인호 편저, "CGI 바이블", 영진출판사, 1997.
- [10] Disan Petkovic 저, 문일보 역, "알기쉬운 마이크로소프트 SQL 서버 7", 정보문화사, 1999.
- [11] Dan Connolly and Jon Bosak, "Extension Markup Language(XML)", 1997, <http://www.w3.org/XML/>
- [12] Jon Bosak, "XML, Java, and the Future of the Web, 1997, <http://sunsite.unc.edu/pub/suninfo/standards/XML/why/xmlapps.htm>
- [13] Dan Connolly and Jon Nosak, "Extensible Markup Language(XML)", 1997, <http://www.w3.org/XML/>