

WWW과 데이터베이스 연동기술의 조사분석

청주대학교 구흥서*

1. 서 론

WWW(World Wide Web)의 급속한 성장은 1990년대 중반의 획기적인 기술적 사건들 중 하나로써 웹은 거의 모든 분야에서 적은 비용으로 정보를 전세계로 전파시킬 수 있도록 만들었다. 이러한 웹이 수없이 많은 기업들의 컴퓨팅 환경을 변화시키고 있지만, 극복해야 할 요소들 중 하나가 데이터베이스와의 통합이다. 이 문제에 대한 효율적인 해결책이 웹이 e-비즈니스 응용을 효과적으로 지원할 수 있도록 만들 것이다.

전통적인 웹 기술은 웹브라우저가 웹서버를 통하여 HTML 문서나 실행파일들을 접근하도록 허용한다. 그러나 대부분의 기관에서는 중요한 비즈니스 데이터와 관리 데이터를 모두 데이터베이스에 저장하므로, 웹 시스템과 데이터베이스 시스템의 연동은 매우 중요하다. 이러한 요구는 인터넷과 웹의 대중성이 증가할수록 더욱 빠르게 증가하고 있다. 특히, 웹은 데이터베이스 전위 시스템으로서 많은 장점을 제공할 수 있다. 웹브라우저가 거의 모든 플랫폼에서 작동하기 때문에 다중 플랫폼 문제를 해결할 수 있다. 더욱이 웹의 하이퍼미디어-기반 모델은 대부분의 사용자에게 친숙하다. 그러므로 웹 데이터베이스 연동 솔루션이 주어지면, 웹은 데이터베이스 전위 응용(database frontend application)을 개발하기 위한 개방형 플랫폼으로 사용될 수 있다[1].

따라서 본 고에서는 웹 시스템과 데이터베이스 시스템을 통합하는 여러 메카니즘을 구조적으로

살펴보고, 최근에 웹 응용에 많이 사용되는 Perl, PHP(Professional Home Page), ASP(Active Server Page), 그리고 자바기반의 데이터베이스 연동구조를 고찰한다.

2. WWW과 데이터베이스의 연동

2.1 WWW과 데이터베이스 통합의 장점

웹과의 연동을 통한 데이터베이스 응용의 개발은 다양한 플랫폼을 지원하고 멀티미디어와 GUI 환경을 쉽게 구축할 수 있는 장점이 있다. 그리고 웹은 문서관리 측면에서 지니는 약점을 데이터베이스의 뛰어난 데이터관리 기능을 통해 보완할 수 있다[2, 3, 4, 5]. 그러므로 데이터베이스 시스템의 유용한 기능과 웹의 편의성과 멀티미디어 서비스 능력을 상호 보완적으로 통합하면 바람직한 대규모 데이터 서비스 응용의 개발이 가능하다.

2.2 WWW 데이터베이스 응용 작성에 필요한 지식

데이터베이스 서버를 접근하는 웹응용을 작성하려면 다음 요소에 관한 지식을 필요로 한다[6].

- 응용이 웹에서 동작하는 메커니즘
- 웹클라이언트로부터 웹서버로 사용자 입력 데이터를 전송하는데 사용되는 메소드들(GET, POST)
- 동적 HTML 문서를 생성하는 응용을 작성하는데 활용 가능한 웹의 지원 도구

* 정회원

2.3 WWW과 데이터베이스 연동시 고려사항

현재의 HTTP 프로토콜을 이용하는 웹 시스템은 웹서버가 클라이언트에 대한 상태 및 연결 정보 등을 유지하지 않으므로 웹 상에서 안정된 온라인 트랜잭션 처리(OLTP)를 보장할 수 없다. 즉 가상금융 및 전자상거래 등의 응용에서 응용 프로그램 개발자가 많은 위험 부담을 지게 되는 것이다. 따라서 이러한 문제들을 해결하기 위해서는 트랜잭션을 지원하는 도구(예, 트랜잭션 서버)를 이용해야 한다.

물론 단일 페이지 환경의 웹 응용 프로그램의 경우에는 별다른 상태유지가 필요 없지만, 여러 페이지를 접근하는 복잡한 웹 데이터베이스 응용 프로그램의 경우에는 상태를 유지하지 않으면 문제가 발생할 수 있다. 가상쇼핑몰은 각 분야별로 여러 페이지에 걸쳐서 구성되고, 사용자는 여러 페이지를 이동하면서 구매를 하는 형태를 갖는다. 사용자가 최종적으로 계산을 하는 페이지로 갔을 때, 응용 프로그램은 사용자가 이전 페이지에서 구입을 결정한 물품들에 대한 정보를 알고 있어야 할 것이다. 따라서 클라이언트의 상태에 관한 정보를 관리하지 않는 현재의 HTTP 프로토콜 하에서는 응용 프로그램이 클라이언트에 대한 상태를 별도로 관리해야 한다.

3. CGI 연동구조

3.1 CGI

그림 1에 나타난 것처럼 CGI(Common Gate-

way Interface)는 웹서버와 외부 응용을 연동하기 위한 표준 인터페이스로서, 기본적으로 CGI를 이용하면 웹과 데이터베이스 연동이 가능하다. 즉 웹을 통하여 데이터베이스에 대한 질의, 갱신, 심지어 새로운 객체나 테이블을 생성하는 프로그램을 작성할 수 있다.

일반적인 HTML 문서는 정적 문서이기 때문에 항상 동일한 내용을 출력하지만, CGI 프로그램은 매번 새로 실행되므로 동적인 정보를 출력할 수 있다. 웹을 통해 데이터베이스를 접근하려면 웹서버가 전송한 질의정보를 데이터베이스 엔진으로 전달하고, 그 실행결과를 반환 받아서 웹서버를 통하여 클라이언트로 출력할 수 있는 CGI 프로그램을 작성하면 된다. 이것이 웹 데이터베이스 게이트웨이(database gateway)이다. 데이터베이스 게이트웨이는 어떠한 프로그래밍 언어로도 작성이 가능하며 C, C++, Java, Perl, Python, Tcl 등을 많이 사용한다.

3.2 데이터 전송방법(GET/POST 메소드)

사용자의 입력 데이터를 서버쪽 CGI 프로그램으로 전달하려면 GET과 POST의 두 가지 메소드를 사용해야 한다. CGI 프로그램을 호출하는 문서, 클라이언트의 IP 주소, 질의처리 방법 같이 공통되는 부분은 환경변수로 전달되고, 질의 데이터는 GET과 POST 메소드에서 서로 다른 방법으로 전달된다.

GET은 질의 데이터를 환경변수 QUERY_STRING을 통해서 전달한다. 환경변수를 통해 전달되기 때문에 데이터의 끝을 쉽게 알 수 있으

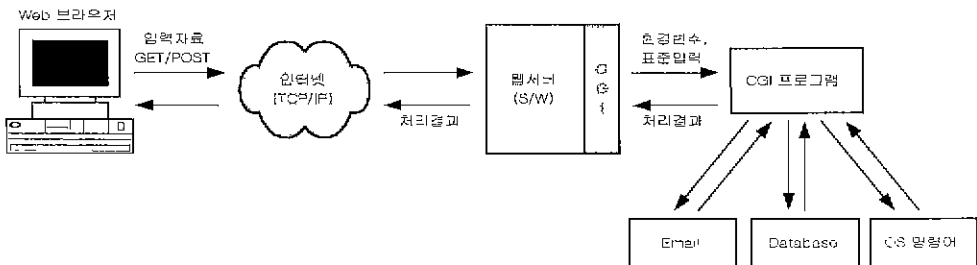


그림 1 CGI 기본 연동구조

나, 환경변수가 저장할 수 있는 최대길이가 256 개 문자로 제한된다는 단점이 있다. POST의 경우는 웹서버가 질의 데이터를 시스템의 내부 입력장치(stdin)를 통해서 CGI 프로그램으로 전송한다. 그러므로 GET 메소드와 달리 전달 데이터의 길이에 제한을 받지 않는다. 또한 POST 메소드는 URL에 사용자 입력 데이터를 첨부하지 않기 때문에 사용자 입력 데이터가 노출되는 잠재적인 보안 위험을 방지할 수 있다.

3.3 CGI 프로그램의 출력

CGI 프로그램에서 생성되는 출력은 헤더(Header) 부분과 데이터 부분으로 구성된다. 키워드 'Header'는 다음에 나오는 데이터의 형태를 나타내며 'Content-type', 'Location', 'Status'가 올 수 있다. 'Location'은 데이터 대신 다른 문서나 프로그램의 위치를 직접 가리킬 때 사용되고, 'Status'는 이상이 생겼을 때 오류코드와 함께 표시해 준다. 'Content-type'은 MIME 타입을 따르며 주형식/부형식으로 구성된다. 헤더 다음에 따라오는 데이터가 HTML 문서일 경우 'Content-type'은 text/html이 된다. HTML 문서 외에도 단순한 텍스트 형식, 이미지, 포스트 스크립트 등의 'Content-type'이 있다.

3.4 HTML과 CGI 프로그램 연동방법

그림 2는 HTML의 폼을 통해 입력된 질의 데이터가 서버쪽의 CGI 프로그램(데이터베이스 게

이트웨이)을 거쳐서 데이터베이스 서버로 전달되는 과정과 데이터베이스 서버의 실행결과가 웹브라우저에 전달되는 과정을 나타낸 것이다.

4. WWW 데이터베이스 게이트웨이

4.1 게이트웨이의 유형

웹 시스템에 대한 데이터베이스 게이트웨이의 구조는 데이터베이스를 접속하는 응용 프로그램의 위치에 따라 그림 3과 같이 분류할 수 있다[7].

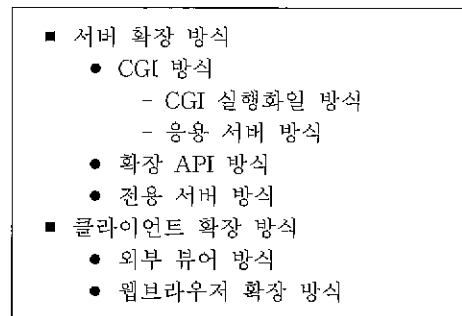


그림 3 게이트웨이의 유형

이번 절에서는 웹 데이터베이스 응용에 대해 주로 많이 사용되는 4가지 기본적인 접근방법을 살펴본다.

4.1.1 CGI 실행파일 방식

CGI 실행파일 방식은 WDB[8]나 GSQL[9]같은 초창기의 웹 데이터베이스 게이트웨이에서 많

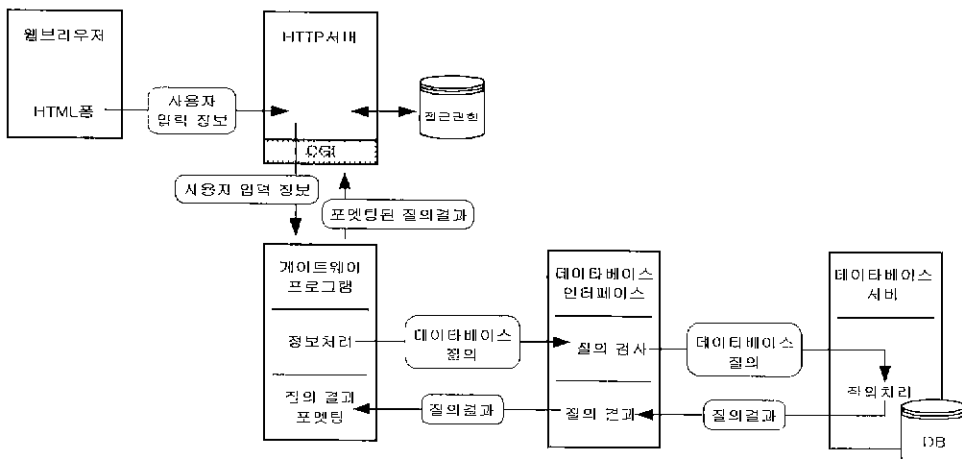


그림 2 웹브라우저와 데이터베이스 서버의 연동구조

이 사용되었다. 이 방식은 단순하기 때문에 구현이 용이하고 어떠한 웹브라우저나 웹서버에서도 사용될 수 있다는 장점을 가진다. 또한 빠르게 발전하고 있는 웹 표준에 종속되지 않으며 유지관리에 복잡한 지식을 요구하지 않는다. 그러나 그림 4에 나타난 것처럼 사용자로부터 요구가 들어올 때마다 새로운 CGI 프로세스를 생성하기 때문에 대규모 사용자 접속환경에서는 성능저하 현상이 발생한다.

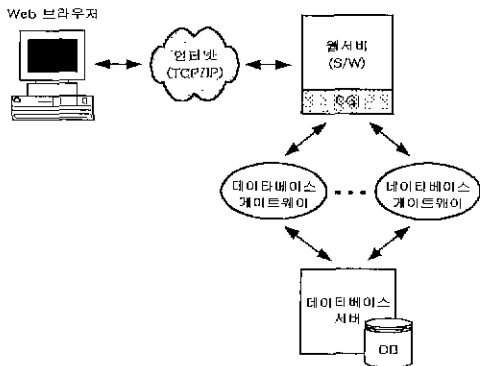


그림 4 CGI 실행화일 연동구조

4.1.2 CGI 응용서버 방식

CGI 응용서버 방식은 게이트웨이를 데몬(daemon) 프로세스로 동작시켜서 CGI 실행화일 방식의 문제점인 다수의 프로세스 생성으로 인한 성능저하를 덜어준다. 그림 5에 나타난 것처럼 CGI 프로세스는 클라이언트로부터 들어온 요구를 응용서버에게 전달하고, 응용서버로부터의 결과를 받아 이를 다시 클라이언트로 전달하는 단순한 일만을 수행한다. 따라서 프로세스 크기가 아주 작아서 많은 수의 프로세스가 동작하더라도 시스템에 큰 무리를 주지 않는다. 또한 한 사용자가 여러 요구를 연속해서 하는 경우에도 효율성을 높일 수 있다. CGI 실행화일 방식에서는 사용자로부터 요청이 들어올 때마다 매번 데이터베이스와의 연결 및 연결해제를 수행해야 한다. 하지만 이 방식은 데이터베이스와의 연결을 계속 유지하기 때문에 뒤이어 발생하는 연속되는 사용자 요구는 새로 데이터베이스를 연결하지 않고 무하 없이 처리할 수 있는 장점을 가진다.

응용서버 방식은 기본적으로 CGI를 이용하기 때문에 CGI 실행화일 방식의 장점을 그대로 가

지면서, CGI 실행화일 방식의 여러 가지 성능상의 문제점을 해결할 수 있다. 한국전지통신연구원(ETRI)에서 개발한 BADA/Web[10], 서울대학교에서 개발한 SWeS[3], 한국컴퓨터통신(주)의 UniWeb[7], 그리고 IBM사의 DB2WWW Connection[6]이 여기에 속한다.

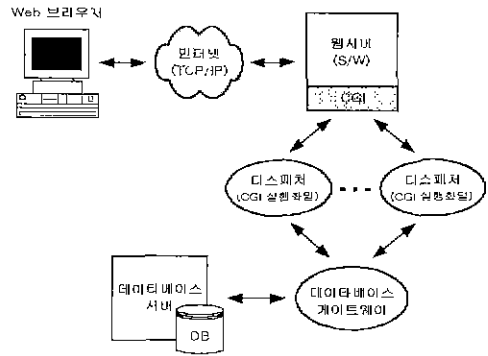


그림 5 CGI 응용서버 연동구조

4.1.3 확장 API 방식

앞에서 기술한 CGI를 이용한 2가지 연동방식의 성능 문제를 개선하기 위하여 웹서버에서 데이터베이스 서버로 직접 접속할 수 있는 확장 API를 지원하는 방법이 HTTP 제작사들로부터 제시되었다. Microsoft사의 ISAPI와 Netscape사의 NSAPI가 여기에 속한다. 응용 프로그래머들은 웹서버에서 지원하는 이러한 확장 API를 이용하여 데이터베이스 응용을 작성할 수 있다. 그림 6은 확장 API를 이용한 웹 데이터베이스 게이트웨이의 구조를 나타낸 것이다. 확장 API를 이용하여 구축된 데이터베이스 응용은 웹서버에 동적으로 링크되어 하나의 프로세스로 수행되므로 에 비하여 성능이 우수하다. 반면, 특정 웹서버에서만 실행되므로 이식성이 떨어지는 단점이 있으며, 일반적으로 웹서버에 대한 많은 경험을 필요로 한다. 그러나 최근에는 Jasmine WebLink[1]와 같이 NSAPI와 ISAPI를 모두 지원하는 제품들도 발표되고 있다. WebLink를 구성하는 세 개 프로그램(odbc-login, odbc-get, odbc-logout)이 NSAPI와 ISAPI의 확장 공유 라이브러리로 구현되어서 CGI 대신 이들을 사용할 수 있다.

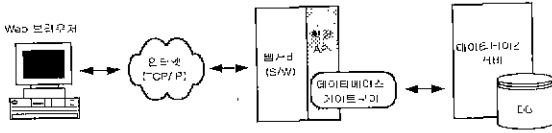


그림 6 확장 API 연동구조

4.1.4 전용서버 방식

웹서버는 HTML 문서를 웹브라우저로 전송하는 특수 목적의 파일서버인데, 최근에 발표되는 일부 웹서버는 데이터베이스 연결 기능을 포함하여 단순 웹서버보다는 응용서버(application server)처럼 동작한다. 데이터베이스 서버와 연결 기능을 제공하는 이러한 전용서버를 이용하면 웹 데이터베이스 응용을 작성할 수 있다. Oracle사의 OWS(Oracle WebServer)와 AOL사의 AOLserver가 여기에 속한다. 그림 7은 전용서버 방식의 웹 데이터베이스 연동구조를 나타낸 것이다.

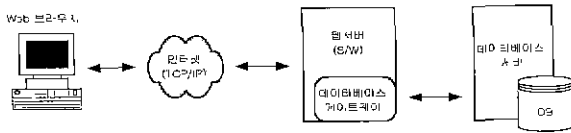


그림 7 전용서버 연동구조

4.2 성능

[11]에서는 4.1절에서 살펴본 접근방식들 중에서 처음 세 가지 방식에 대해 성능평가를 보여주고 있는데, 자체 함수 호출의 부하가 외부 CGI 프로그램을 실행시키는 것에 비해 대단히 적기 때문에 확장 API 방식이 CGI 방식(CGI 실행화일 방식, CGI 응용서버 방식)보다 성능이 우수한 것으로 나타났다. 그러나 이것은 성능과 이식성 간의 상충관계를 포함하므로 선택에 주의해야 한다. CGI 응용서버 방식은 이 방식이 지나는 복잡한 프로세스 구조가 성능을 크게 저하시키지 않으면서, 일반 웹서버를 그대로 이용할 수 있기 때문에 성능과 이식성을 모두 만족시킬 수 있다.

4.3 데이터베이스 접근방법의 유형

WWW 데이터베이스 게이트웨이는 여러 가지

방법으로 데이터베이스를 접근할 수 있다. 주요한 데이터베이스 접근방법은 다음과 같으며 이 방법들이 서로 배타적이지는 않다[2].

- 확장 HTML 방식:SQL문을 HTML 문서에 내장시키는 방식으로 게이트웨이는 해당 문서를 파싱하여 SQL 질의를 데이터베이스로 전송한 다음, 질의결과를 반환 받아서 표준 HTML문서로 포맷팅하여 웹서버로 전송한다. IBM사의 DB2WWW Connection, CA사의 Jasmine WebLink, W3-msql에서 이 방식을 사용하고 있다.
- 템플릿 파일 방식:데이터베이스에 저장된 테이블과 필드들에 관한 정보를 외부 템플릿 파일에 저장한 다음, 사용자가 요청하면 이 정보를 이용하여 동적 SQL 질의를 생성하는 방식이다. DB2WWW Connection, GSQL, WDB, Jasmine WebLink 등이 이 방식을 사용한다.
- 데이터베이스 저장 프로시저 방식:일련의 SQL 명령어들을 데이터베이스에 저장한 다음 필요할 때 외부 프로그램에서 호출할 수 있는 데이터베이스 저장 프로시저 기능을 이용하는 방식이다. Oracle WWW Interface Kit 내에 포함된 WOW(Web Oracle Web) 게이트웨이가 이 방식을 사용한다.
- 표준 인터페이스 방식:동일한 데이터베이스 응용 프로그램으로 다양한 데이터베이스를 접근할 수 있도록 산업계 표준 인터페이스인 ODBC(Open DataBase Connectivity)를 이용하는 방식이다. 이 방식은 ODBC를 지원하는 다중 데이터베이스를 접근해야 하는 경우 매우 유용하다. ODBC는 X/Open SAG(SQL Access Group)의 CLI(Call Level Interface)를 기반으로 하는 개방형 데이터베이스 접근 API이다.
- 고유 API 방식:특정 데이터베이스 엔진이 제공하는 고유의 API를 이용하는 방식으로서 단일 제품에 대한 접근만을 허용한다. ODBC API와는 달리 이식성은 떨어지지만 특정 데이터베이스 엔진이 제공하는 모든 기능을 활용할 수 있는 장점이 있다.

4.4 게이트웨이 선택시 고려사항

웹 데이터베이스 응용을 개발하기 위해 게이트웨이 도구를 선택하려면 다음 요소들을 고려해야 한다[2].

- 데이터베이스 호환성
- 지원 플랫폼
- 지원되는 SQL 질의 및 갱신 범위
- 데이터 보안 기능
- 성능
- 데이터베이스 접근방식

5. Perl과 데이터베이스의 연동

Perl은 웹 환경에서 가장 인기 있는 CGI 스크립트 언어로서 데이터 조작 기능이 탁월하기 때문에, 다양한 데이터베이스 서버와 직접 연결할 수 있는 확장 모듈들이 많이 개발되어 있다. Perl 4에서는 Oracle에 대한 인터페이스인 oraperl, Sybase에 대한 인터페이스인 sybperl, 그리고 Ingres에 대한 인터페이스인 ingperl같은 인터페이스들이 개발되었으며, 이들 인터페이스들은 각각의 고유한 구문을 가지며 호환이 되지 않았다. 그러나 Perl 5에서는 객체지향 기능들을 이용하여 보다 이식성이 높고 강력한 방법으로 이 문제를 해결하였다. 이러한 예가 Perl DBD/DBI (database dependent/database independent interface) 데이터베이스 라이브러리이다.

Perl DBD/DBI 기술은 Microsoft사의 ODBC 기술과 유사한 개념을 사용한다. DBI 모듈은 모든 데이터베이스 서버와 통신할 수 있도록 단일 인터페이스를 지원하므로, DBI 모듈을 사용하면 각각의 데이터베이스 엔진에 대해 별도의 Perl

스크립트 프로그램을 작성할 필요가 없다. DBI 모듈로 전송된 SQL 명령들은 변환되어 각각의 데이터베이스 서버에 대해 개발된 다양한 DBD 라이브러리를 통하여 데이터베이스 서버로 전달되어 처리된다. 그림 8은 Perl DBI/DBD 데이터베이스 라이브러리와 데이터베이스 서버가 연동하는 구조를 나타낸 것이다[12].

6. PHP와 데이터베이스의 연동

PHP(Prpessional Home Page)는 HTML 문서에 내장하여 사용할 수 있는 서버쪽 실행 스크립트 언어이다. 사용자가 PHP와 HTML로 작성한 웹 문서를 접속하면 ASP나 ColdFusion처럼 웹서버쪽에서 PHP 명령어들을 처리하여 그 결과를 웹브라우저로 전송한다. 사용자 질의가 웹브라우저를 통하여 웹서버로 전달되면, 서버쪽의 PHP가 그 질의를 해당 데이터베이스 서버에 대한 확장 모듈을 통하여 데이터베이스 엔진으로 전송한다. PHP와 데이터베이스 서버간의 연동구조는 그림 9에 나타나 있다.

PHP는 윈도우즈 NT와 많은 유닉스 플랫폼을 지원하며, Apache 웹서버와는 하나의 모듈로서 구성될 수 있기 때문에 가볍고 빠르게 동작한다. PHP의 가장 큰 강점 중 하나는 다양한 데이터베이스 서버에 대해 쉬운 인터페이스를 제공한다는 점이다. 현재 PHP에서 제공하는 데이터베이스 모듈은 다음과 같다 - Informix, Microsoft SQL Server, MySQL, mSQL, ODBC, Oracle, Sybase, PostgreSQL 등[13, 14].

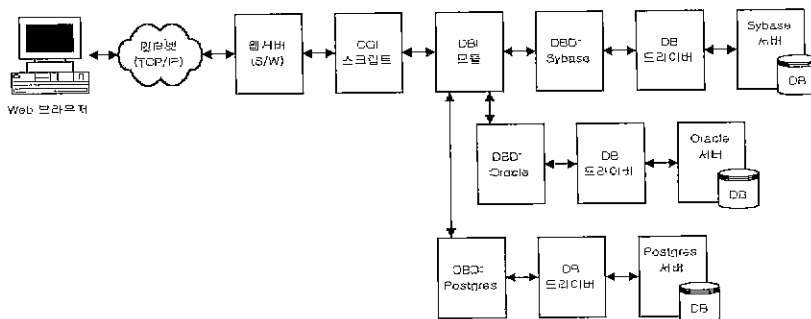


그림 8 Perl DBD/DBI 연동구조

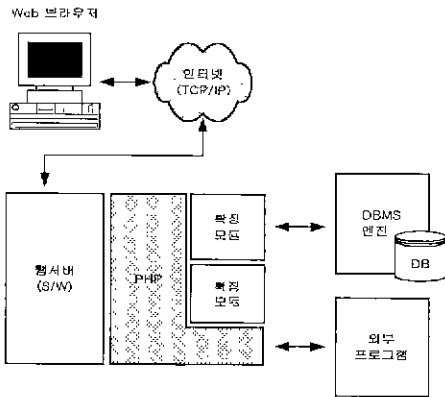


그림 9 PHP와 데이터베이스 연동구조

7. ASP와 데이터베이스의 연동

IIS 3.0과 함께 제공되는 Microsoft사의 ASP(Active Server Page)는 웹응용을 위해 ODBC 지원 데이터베이스 서버와 상호작용 할 수 있도록 유연하고 사용법이 쉬우면서 확장성이 있는 메소드들을 제공한다. 그림 10은 ASP와 데이터베이스 서버가 연동하는 구조를 나타낸 것인데, ASP는 OLE DB를 기반으로 한 ADO (Active Data Obejct) 컴포넌트를 이용하여 데이터베이스에 연결한다.

ASP에서는 데이터베이스를 접근하기 위해 두 가지 방법을 제공한다. 첫 번째 방법은 데이터베이스에 접근할 때마다 새로 데이터베이스 연결을 설정한다. 즉, 매번 사용자 요구가 있을 때마다 데이터베이스 연결이 설정된 다음, SQL 질의가

실행되고, 질의결과가 생성되면 관련된 모든 객체를 반환한다.

두 번째는 하나의 세션동안은 계속 데이터베이스 연결을 유지하는 방법이다. 이것은 "Global.asa" 파일에 의해 제어된다. ASP를 이용한 웹응용은 응용의 루트 디렉토리에 한 개의 "Global.asa" 파일을 가질 수 있는데, 이 파일이 4가지 이벤트(event)를 포함한다 - 응용시작, 세션시작, 응용종료, 세션종료[15].

ASP는 서버쪽에서 HTML 문서를 동적으로 생성하는 웹페이지 엔진이며, Microsoft사의 MTS(Microsoft Transaction Server)와 결합하여 웹 응용서버로서 역할을 수행할 수 있다.

8. 자바기반 접근방법

자바기반의 연동방법으로 서블릿(servlets)을 이용하는 방법이 있는데 이것은 CGI 접근방법에서 발생하는 성능문제를 해결할 수 있다. 서블릿은 서버쪽에서 실행되는 애플릿(applet)으로 자바 서버 API의 서브셋이다. 서블릿은 CGI와 유사하게 HTML 폼(form)이나 클라이언트 애플릿을 통해서 PUT/POST 메소드로 사용자 질의를 전달받는다. 질의를 전달받은 서블릿은 JDBC를 통해서 데이터베이스 서버에 연결하고, 실행결과를 반환 받아서 HTML 문서로 포맷팅한 다음 클라이언트로 전송한다. 그림 11은 이러한 연동구조를 나타낸 것이다 서블릿은 자바 멀티쓰레드를 이용해 실행되기 때문에 CGI 방식처럼 다수의 프로세스 생성으로 인한 성능 저하 현상이 발생하지 않는다.

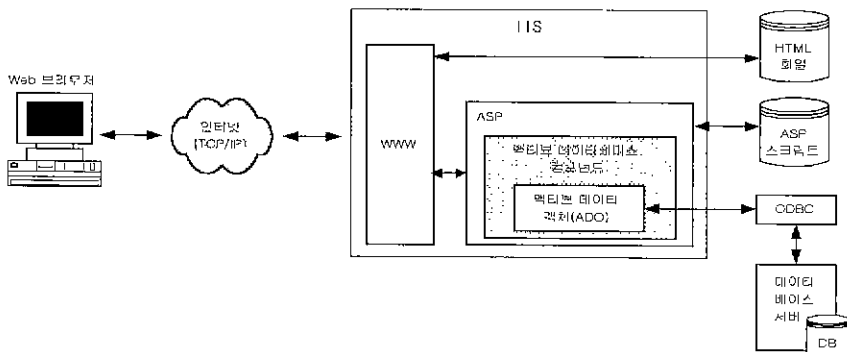


그림 10 ASP와 데이터베이스 연동구조

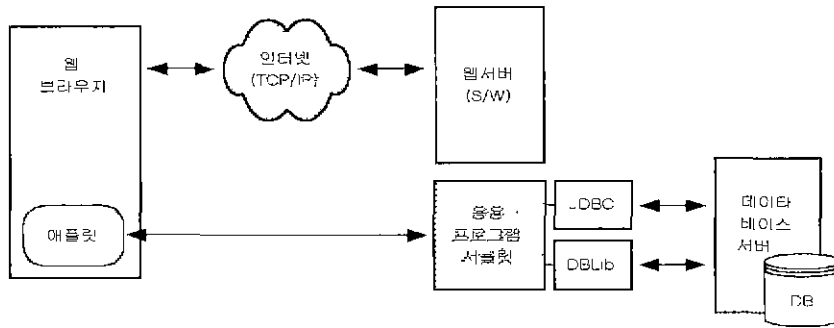


그림 11 자바기반 연동구조

서블릿을 이용한 자바기반 접근방법은 상태유지가 가능하고 플랫폼에 독립적이며 JDBC를 통한 데이터베이스 연결이 용이하다는 장점을 가진다. JDBC는 Sun Microsystems사에서 개발한 SQL 데이터베이스 접근 인터페이스로서, JDBC API는 데이터베이스 연결, SQL 명령문, 질의결과 집합, 메타 데이터 등을 표현할 수 있는 자바 클래스들을 정의한다.

9. 결 론

본 고에서는 다중 플랫폼을 지원하고 사용자 인터페이스가 친숙한 웹 시스템과 데이터관리 기능이 뛰어난 데이터베이스 시스템의 연동 방법들을 살펴보았다. 먼저 가장 단순한 CGI를 이용한 연동방법을 살펴보고, 데이터베이스를 접속하는 응용 프로그램의 위치에 따라 분류된 데이터베이스 게이트웨이의 여러 유형을 구조적 측면에서 고찰하였다. 그런 다음, 최근에 웹 응용에 많이 사용되는 Perl, PHP, ASP, 그리고 자바기반의 데이터베이스의 연동구조를 각각 살펴보았다.

그러므로 각자의 환경에 적합한 연동구조를 선택하면 바람직한 웹기반 데이터베이스 응용 개발이 가능할 것이다. 웹 시스템은 매우 빠르게 발전하는 성장기에 위치한 기술이기 때문에 웹 데이터베이스 연동구조를 설계할 때 성능뿐 아니라 차후의 확장성도 고려해야 한다는 점을 명심해야 한다.

최근에는 웹 환경에서 개발자들이 비즈니스 로직(logic)을 격리시켜서 3계층 응용을 쉽게 개발할 수 있는 다양한 응용서버들이 시장에 발표되고 있다. 이들은 웹 환경의 특성을 보완해 주면서 데이

터 처리를 위한 응용 계층과 웹서버 계층이 통합되어 관리될 수 있도록 설계되어 있어서, 대규모의 웹기반 데이터베이스 응용 개발의 생산성과 효율성을 향상시킬 수 있다. IBM사의 WebSphere, Oracle사의 OAS(Oracle Application Server), Sybase사의 Enterprise Application Server 등이 여기에 속한다.

웹 데이터베이스 연동 시스템은 사용법이 쉽고, 가격대성능 등에서 많은 장점이 있어서 기존의 OLTP 시스템을 웹 환경으로 확장하는데도 많이 적용되고 있지만 신중한 판단이 요구된다. 왜냐하면 이 시스템은 기본적으로 웹을 기반으로 하기 때문에 대규모 OLTP 시스템이 요구하는 안정성과 신뢰성을 항상 제공할 수 있는 것은 아니기 때문이다[16]. 웹기반 데이터베이스 응용 시스템들이 OLTP의 충분한 지원을 필요로 하는 중요한 비즈니스 응용 시스템으로 역할을 수행하려면 다음 문제들이 해결되어야 할 것이다.

- 예측 가능한 응답시간: 대부분의 응답시간이 큰 편차 없이 일정해야 하는데 인터넷은 무수한 중간노드들을 거쳐 데이터가 전송되므로 응답시간이 일정하지 않을 수 있다.
- 연속적인 가용성: 극히 적은 비가동시간 이외에는 24시간, 365일 연속 가동되어야 한다.
- 데이터 보안: 전체 시스템이 비인가된 사용으로부터 보호되어야 하는데 인터넷은 보안이 취약하다는 단점을 가진다.

참고문헌

- [1] S. Khoshafian, et al., The Jasmine Object Database Multimedia Applications for the

Web, Morgan Kaufmann Publishers, 1999.

[2] J. Rowe, Building Internet Database Servers with CGI. New Riders, 1996.

[3] 최일환 외 2인, "SRP RDBMS를 위한 Web 게이트웨이", 한국정보과학회 논문지, 1999.2.

[4] 김완석 외 2인, "웹과 DBMS의 연동 기술," ETRI 주간기술동향 97-39, 1997.

[5] 문장원, WWW 환경에서의 데이터베이스 게이트웨이 설계 및 구현, 석사학위논문, 한국과학기술원, 1996.

[6] T. Nguyen, V. Srinivasan, "Accessing Relational Database from the World Wide Web," Database Technology Institute IBM Santa Teresa Laboratory, ACM SIGMOD, 1996.

[7] 김평철, "UniWeb - 웹을 이용한 클라이언트-서버 데이터베이스 응용 개발 환경," 데이터베이스 저널, 제3권, 제2호, 1996.

[8] B. F. Rasmusen, B. Pirenne, "WDB: A WWW to Sybase Interface," Proc. of the Workshop on Network and System Management, 1995.

[9] Jason Ng., "Gsql-a mosaic sql gateway," <http://www.ncsa.uiuc.edu/SDG/People/jason/pub/gsql/>, 1993.

[10] W.-S. Kim, "BADA-III/Web: Integration of the Web and an OODBMS", Proceedings of the 32nd Hawaii International Conference on System Sciences, 1999.

[11] 김평철, 민영훈, "월드와이드웹용 데이터베이스 통로의 성능 평가," 데이터베이스 연구회지, 제13권, 제2호, 1997.

[12] S. Sol, Part Three: CGI Interfaces to SQL Databases. <http://www.wdvl.com/Authoring/DB/Intro>, 1998.8.

[13] R. Lerdorf, "Dynamic Web Pages with PHP3", Web Techniques, 1998.2.

[14] C. Knudsen, Script Web databases quickly with PHP scripting language, <http://www-4.ibm.com/software/developer/library/web-php/index.html>, 1999.9.

[15] A. Saldanha, ASP 101 - A Database Interfacing Primer, <http://vallin.com/pub/1/aspl.asp>

[16] B. Black, "OLTP on the Internet, Internet Systems", 1996.10.

구 흥 서



1985 인하대학교 전산학과 학사
 1989 인하대학교 전산학과 석사
 1993 인하대학교 전산학과 박사
 1984~1986 현대전자산업(주) 연구원
 1994~현재 청주대학교 컴퓨터정보공학과 조교수
 관심분야: 데이터베이스 시스템, 웹정보 시스템, 정보검색, 컴포넌트 소프트웨어
 E-mail: hskoo@chongju.ac.kr

● 2000년도 한국데이터베이스 학술대회 ●

- 일 자 : 2000년 5월 12 ~ 13일
- 장 소 : 대덕 롯데호텔
- 내 용 : 튜토리얼, 논문발표 등
- 문 의 처 : 동국대학교 컴퓨터공학과 엄기현 교수
 Tel. 02-2260-3339