

전자상거래를 위한 CORBA 기반 분산 DBMS의 설계 및 구현

박정인[°], 전순미
인제대학교 전산학과

The Design and Implementation of CORBA-based Distributed DataBase Management System in Electronic Commerce

Jeong-In Park[°], Soon-Mi Jun
Department of Computer Science, Inje University

요 약

최근 각광 받고 있는 웹에서의 전자상거래 구축 시 필수요소인 데이터베이스에 대한 접근 방법과 다수의 클라이언트가 하나의 데이터베이스 서버로의 접근 시 발생할 수 있는 접속 과부하 문제를 해결하기 위해 분산처리 기술인 CORBA를 이용하여 3계층 방식의 분산 DBMS를 구현하였다. 즉, 웹 상에서 전자상거래 시스템을 운용할 때 2단계 방식의 클라이언트/서버를 사용함에 따라 나타나는 여러 문제를 해결하고, 서버의 시스템 부하를 줄이기 위해 CORBA를 이용하여 3계층 방식으로 구현하였으며, 웹을 통해 고객이 주문한 상품을 서버의 구현체호출을 통해 분산된 데이터 베이스 서버에 저장되게 하였다.

1. 서 론

최근 각광받고 있는 인터넷을 이용한 전자상거래의 도입으로 인해 서 다루어져야 하는 정보의 형태는 단순 텍스트 문서에서 음성, 도면, 그래픽 데이터, 화상에 이르는 멀티미디어 정보로 그 형태가 변하게 되었고, 따라서 이러한 정보를 처리하기 위한 데이터베이스 기술과 자료압축 및 복원기술, 인터넷 및 Web 관련기술, 멀티미디어 관련기술 등이 필요하게 되었다. 그 중에서도 데이터베이스 기술은 중대되는 클라이언트의 요구에 가장 민감히 반응하는 전자상거래의 필수 구성요소가 되었다.[5]

분산시스템이 대두되기 이전의 기존 시스템은 호스트 중심의 중앙 집중식 방식이었다. 이는 중앙의 시스템에 모든 작업이 집중되는 등 여러 문제를 초래하였다. 특히 최근 웹에서의 전자상거래 시스템의 규모가 방대해짐에 따라 시스템의 성능 향상 또한 크게 요구되어졌고, 이러한 문제들을 해결하기 위해 이중 컴퓨터들 간에 프로그램을 분산시켜 부하를 줄임으로써 시스템의 성능 저하와 네트워크 병목 현상을 해결하려 하였다.[5] 이를 위해 클라이언트/서버 시스템이 대두하게 되었는데, 기존 2단계 모델에서는 클라이언트 측 사용자 수에 따라 서버의 세션을 설정하기 때문에 사용자의 수가 늘수록 서버에 걸리는 부하가 역시 중대하였다. 그러나 기존 클라이언트/서버 모델을 변형한 다단계 방식의 클라이언트/서버 모델은 클라이언트와 서버 사이에 상호 작용을 위해 미들웨어를 사용하여 하나의 시스템처럼 구성하여 여러 이질적인 데이터베이스 서버와 부하를 줄이며 일관되게 연결할 수 있게 해준다.[6]

이에 따라 본 연구에서는 이질적인 데이터베이스의 호출과 시스템 부하를 줄이기 위한 미들웨어로서 OMG(Object Management

Group)의 CORBA(Common Object Request Broker Architecture)를 이용하여 이러한 문제를 해결하고자 하였다.

이상의 연구에 대한 본 논문의 구성은 다음과 같다. 제 2장에서는 전자상거래 기술, 웹에서의 데이터베이스 접근 및 CORBA에 대한 관련연구를 살펴보고, 제 3장에서는 구현하고자하는 Web 전자상거래 분산데이터베이스 시스템의 설계 및 전체 구성도를 제시한다. 끝으로 제4장에서 결론 및 향후 연구과제를 알아보고 글을 맺는다.

2. 관련연구

2.1 전자상거래의 정보처리 기술

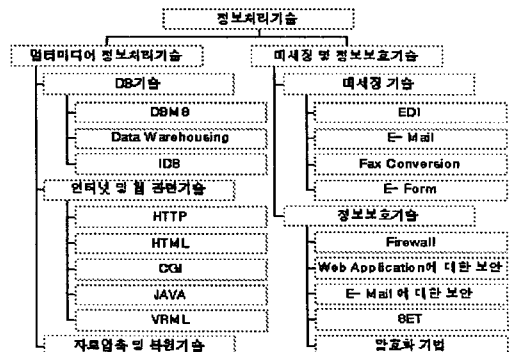


그림 1 정보처리 기술 체제도

그림 1은 전자상거래의 요소기술로 정보처리 기술의 체계를 나타내고 있으며 본 절에서는 그 내용에 대해 알아보기로 한다.

정보처리 기술은 크게 멀티미디어정보 처리기술과 메세징 및 정보보호 기술로 나눌 수 있으며, 다시 멀티미디어 정보를 처리하기 위한 기술은 DB 기술, 인터넷 및 웹 관련기술, 그리고 자료압축 및 복원기술 등을 들 수 있다.

DB 기술은 DBMS와 데이터웨어하우징 및 IDB구축 등의 문제에 대해 생각해 볼 수 있는데, DBMS는 현재 상용화되어 많이 사용되고 있는 RDBMS와 현재 멀티미디어 관련 자료의 처리에 적용하기 위한 기술로 논의되고 있는 OODBMS를 들 수 있다.[5]

2.2 2단계 구조의 클라이언트 서버 시스템

2.2.1 원격 데이터베이스 접근 모델과 데이터베이스 서버모델

원격 데이터베이스 접근 모델은 원격지의 컴퓨터에 위치한 데이터베이스 서버에게 SQL같은 데이터베이스 질의어를 통해 서비스를 요청하고 이에 대한 결과를 제공받는 모델을 말한다. 실제 개발 시에는 주로 ESQL(Embedded SQL)방식을 사용하여 C나 C++코드에 SQL을 직접 삽입하여 서비스를 요청하고 이에 대한 결과를 일련의 테이블 형태로 전달받는다.[4][6]

이 모델의 단점으로는 클라이언트/서버간 대규모의 데이터 이동이 불가피하다는 점이다. 이점은 본 논문에서 구현하고자 하는 전자상거래에 있어서 가장 큰 문제점으로 지적되어진다. 예를 들어, 전자상거래에 있어 하나의 상품정보 비교에 관한 SQL요청 결과가 수십만개 라면 이 모든 데이터가 서버에서 클라이언트로 옮겨져야만 한다. 이에 따라 시스템의 성능 저하 및 네트워크 병목현상을 초래하게 된다. 또한 프로그램의 로직과 구조가 서버 시스템의 구조에 의존하기 때문에 서버의 데이터베이스 구조가 바뀌면 모든 클라이언트 프로그램을 재작성 해야 한다. 그 외 클라이언트의 수가 늘수록 그 수에 따라 서버의 세션을 설정하기 때문에 서버에 부하가 걸리는 단점이 있다.

이 외에 2단계 구조로 데이터베이스 서버 모델이 있는데, 위 모델의 문제점을 다소 해결하고자 개발된 모델이다. 이 모델은 데이터베이스 시스템의 스토어드 프로시저를 사용함으로써 응용 프로그램의 로직이 바뀌더라도 서버내의 프로시저만 변경하고 이를 클라이언트들이 공유함으로써 응용 프로그램 개발 후 유지 보수가 다소 유리하다. 그러나 이 두 방식 모두 소규모이고 비 정형화된 업무에 적합한 방식이기 때문에, 본 논문에서 구현하고자 하는 전자상거래 시스템이나, 은행업무 또는 증권전산처럼 대규모의 클라이언트를 요구하는 환경에서는 성능 저하 등의 문제를 유발할 수 있다.[6]

2.3 다단계 클라이언트 서버 시스템

2.2에서 언급한 모델들의 문제점들을 해결하는 방안으로써 대두된 것이 다단계 구조의 클라이언트/서버 시스템이다. 이 다단계 클라이언트/서버 시스템은 데이터베이스가 위치한 서버부분과 최종 사용자들이 사용하는 클라이언트 부분을 완전히 분리하고 이들 사이에 연결 기능을 제공하는 중간 소프트웨어(미들웨어)를 위치시킴으로써 좀 더 유연하고 확장 가능한 시스템을 구현할 수 있게 해 준다. 특히 데이터베이스의 위치나 공급업자에 상관없이 이질적인 DBMS를 계속 연결시켜 확장이 가능하며 서버에 많은 부하가 걸릴 때 이 부하를 균등하

게 분배하는 로드 밸런싱(Load Balancing) 기능이 가능하다.[3] 사실상 대규모의 분산 환경에서 다양한 서버들에 접근하기 위해서는 해당 서버의 개수만큼 응용 프로그램을 작성해야 하는 반면에, 미들웨어를 이용한 다단계 모델처럼 미들웨어가 복수 개의 클라이언트와 서버사이에서 이들의 인터페이스를 담당함으로써 이질적인 대규모의 분산 환경에서 별도의 인터페이스 프로그램 없이도 접근할 수 있다. 이러한 이유 때문에 분산 환경을 구축하는 데 있어 미들웨어의 도입과 활용은 가장 중요한 결정사항 중의 하나라고 할 수 있다.[2][4][6]

2.4 CORBA

CORBA는 지능형 컴포넌트들이 서로를 발견하고, 객체 버스 상에서 상호 운용될 수 있게 설계되어 있어, 객체를 생성 및 삭제하고, 객체에 이름을 통해 접근하며, 지속적인 저장장소(Implementation Repository, Interface Repository)에 객체를 저장하고, 객체의 상태를 의형화 하고, 그들간의 특별한 관계를 정의하기 위하여 버스 관련 서비스의 확장된 집합을 지정한다.[3]

CORBA는 일반 객체를 생성한 다음 생성된 객체를 적절한 서비스로부터 다중 상속관계 함으로써 트랜잭션, 보안성, 폐쇄성 및 지속성을 갖게 한다. 이는 일반적인 컴포넌트를 일반 함수를 제공할 수 있도록 설계한 다음, 컴포넌트 구현 시 또는 런타임에 적절한 미들웨어 믹스를 삽입할 수 있다는 것을 의미한다. 기존의 클라이언트/서버 환경에서는 이러한 것을 할 수 없다.

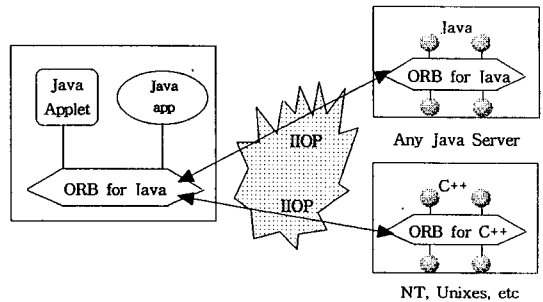


그림 2 ORB : The Client and Server Story

3. 제안된 시스템의 설계

3.1 데이터베이스 연결

본 시스템 설계를 위한 요구사항으로는 CORBA를 기반으로 하여 Java 애플리케이션과 데이터베이스를 연동하는 기술에 관한 것이다.

CORBA기반 하에서 Java애플리케이션으로 데이터베이스에 접근하기 위해서는 Java에서 데이터베이스에 연결하기 위한 JDBC를 이용하여 연결한다. 연결 방식은 2단계와 3단계 방식이 있는데, 2단계 방식을 적용하게 되면, 각 클라이언트 어플리케이션마다 JDBC를 두어야 한다. 이는 서버의 DBMS가 변경되거나 서버 어플리케이션 변경 시 클라이언트마다 일일이 드라이버를 변경시켜 주어야 하는 단점을 가지고 있었다. 하지만 3단계 방식을 이용하게 되면 2계층에 JDBC를 두어 DBMS에 연결하는 드라이버를 하나씩만 구현해 주면 된다.[2][4]

다음의 소스코드는 본 논문의 구현을 위한 JDBC를 이용한 데이터베이스 접근방법을 나타내며, 먼저 데이터베이스에 연결하기 위해

JDBC 드라이버를 연결하였다. JDBC를 사용하면, 어떠한 관계 데이터베이스로도 SQL문을 전송하기 쉽다. 즉, JDBC API를 사용하면 여러 이질적인 DB에 접근하는 프로그램을 따로 만들 필요가 없다. 단지 하나의 프로그램을 작성하고 그 프로그램에서 SQL문을 적당한 데이터베이스에 전송할 수 있다.

```
public void connect(String datasource, String db,
String id, String pw)
throws Exception
{
try {
//드라이버 로딩
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
String url = "jdbc:odbc:" + datasource + ";database=" + db;
System.out.println("Connecting to " + url);
con = DriverManager.getConnection(url, id, pw);
} catch(Exception e)
{ System.err.println("System Exception in connect");
System.err.println(e);
throw e;
}
}
```

그림 3 데이터베이스 연결

3.2 시스템 전체 구성도 및 동작 메커니즘

본 논문에서 제안하는 시스템의 구성은 쇼핑몰에서 상품구매를 원하는 클라이언트 측과 이러한 상품들에 대한 정보를 저장하고 있는 데이터베이스 서버, 그리고 각 DBMS에 접근하기 위한 JDBC 드라이버들을 포함하고 있는 미들웨어로 분류된다. 여기서 상품정보를 담고 있는 데이터베이스 서버는 여러 개의 DBMS 엔진들로 구성될 수 있으며, datasource 이름과 db 이름을 참조하여 JDBC 드라이버를 이용하여 접근 가능하도록 하였다. 아래 그림 4는 제안된 시스템의 전체 구성도이다. 웹을 통한 전자상거래 시스템은 다수의 클라이언트 접속이 이루어지기 때문에 서버가 받게되는 부하가 큰 처리시간의 요구, 다량의 데이터 전송 등의 문제를 해결하기 위해 CORBA를 이용하여 3단계의 구조로 설계해 보았다. 3계층에 해당하는 DBMS에 접근하기 위해 Java에서 데이터베이스 연결 시 필요한 JDBC를 사용하였다.

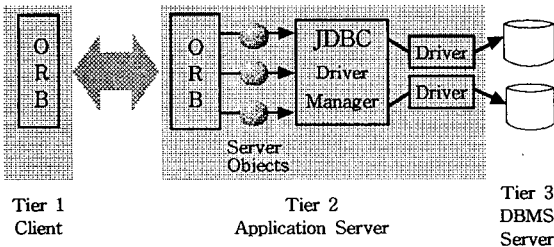


그림 4 3-tier 방식의 전자상거래 전체 구성도

그림 4의 구성에 따른 이 시스템의 특징을 살펴보면, 서버와 접속하기 위한 JDBC 드라이버는 미들웨어 속에 같이 포함되어 구현객체 내에 정의되어 있으므로 클라이언트 측에서는 어떤 드라이버를 이용하여 어떤 서버에 접속해야 하는가에 대해서는 전혀 알 필요가 없다. 다만 미들웨어 내에 드라이버들을 정의하고 데이터베이스 이름을 참조하여 해당 데이터를 찾아가는 것이다. 데이터베이스 서버는 하나 이상의 DBMS들로 구성하였고, 향후 여러 대의 서버머신(machine)

으로 확장 가능하다.

3.3 데이터베이스 스키마

데이터베이스 테이블의 구조는 크게 고객 테이블, 상품 테이블, 주문 상품 테이블, 주문 결과 테이블로 구성된다.[1] 각각의 구성을 살펴보면 다음과 같다.

· 고객 테이블

클라이언트는 자바 애플릿을 통해 상품을 주문하고 이때 상품 코드와 함께 고객의 정보가 전송된다. 이때 고객에 대한 정보와 주문 상품에 대한 정보는 각각 다른 테이블에 저장된다.

· 상품 테이블

상품에 대한 유일한 코드번호가 존재해야 하며, 클라이언트에 의해 전송되어진 코드번호는 해당 상품과 조인하여 전자데이터베이스를 만들게 된다.

· 주문 상품 테이블

클라이언트에서 주문한 상품의 코드가 저장되며 주문한 고객이 누구인지를 식별할 수 있는 키가 저장된다.

· 주문 결과 테이블

고객의 정보를 갖고 있는 키를 이용하여 출력하고 주문한 상품의 코드에 해당하는 상품을 표시한다.

4. 결론 및 향후 연구과제

본 논문에서는 CORBA를 이용하여 최근에 각광받고 있는 전자상거래 시스템을 제안하였고 이를 운용함에 있어 다수의 클라이언트가 하나의 데이터베이스 서버로의 접근 시 발생할 수 있는 접속 과부하 문제를 해결하기 위해 3계층 방식의 분산 DBMS를 구현하였다.

이는 CORBA를 기반으로 함에 따라, 플랫폼에 독립적으로 수행되며, 서버가 받는 부하를 줄이기 위해 서버의 분산처리가 용이하게 하였다. 또한 각 DBMS의 위치 투명성을 보장하기 때문에 메소드 호출의 동일성을 보장할 수 있다. 하지만 이러한 장점 외에 단점으로는 ORB 통신 시 클라이언트 측 통신객체들을 로딩 하는데 많은 시간이 걸리므로 빠른 결과를 원하는 사용자에게는 호응을 얻지 못하는 요인이 될 수가 있다.

앞으로의 연구방향은 상품정보를 디스플레이 할 때 음성 및 동영상 형태의 멀티미디어 정보 전송에 관한 연구가 행해져야 할 것이다.

참고문헌

- [1]C. J. Date, "An Introduction To Database Systems", Addison-Wesley Publishing Company, 1990.
- [2]Gary Cornell and Cay S. Horstmann, "Core JAVA", The SunSoft Press, 1997.
- [3]Jon Siegel, "CORBA Fundamentals and Programming", John Wiley and Sons Inc., 1996.
- [4]Robert Orfali and Dan Harkey, "Client/Server Programming with JAVA and CORBA", John Wiley and Sons Inc., 1998.
- [5]연세대학교 산업정보시스템 연구실, "전자상거래", <http://www.ecrc.or.kr>
- [6]왕창중 외 1인, "분산 객체 컴퓨팅 기술, CORBA 프로그래밍", 대림, 1998.