오디세우스 객체관계형 DBMS를 위한 JDBC 드라이버의 설계 및 구현

김민수[0] 이민재 이재길 황규영
한국과학기술원 전자전산학과 전산학 전공/첨단정보기술연구센터
{mskim, mjlee, ariel, kywhang}@mozart.kaist.ac.kr

Design and Implementation of a JDBC Driver for the ODYSSEUS Object-Relational DBMS

Min-Soo Kim[0] Min-Jae Lee Jae-Gil Lee Kyu-Young Whang
Department of Electrical Engineering & Computer Science
Division of Computer Science and
Advanced Information Technology Research Center
Korean Advanced Institute of Science and Technology

요 약

JDBC 는 Java 프로그램에서 표준 SQL 을 사용하여 데이타베이스에 접근하기 위한 응용 프로그램 인터페이스이다. 응용 프로그램 개발자는 JDBC API 를 사용하여 다양한 종류의 DBMS 에 적용 가능한 응용 프로그램을 쉽게 작성할 수 있을 뿐만 아니라 JDBC API 를 사용하여 작성된 기존의 응용 프로그램들을 재사용하는 것이 가능하다. 본 논문에서는, 객체관계형 데이타베이스 관리 시스템인 오디세우스를 위한 JDBC 드라이버를 설계하고 구현한다. 본 JDBC 드라이버는 DBMS 와 직접 연결하여 통신하므로 질의 결과 처리 성능이 좋고, 모두 Java 언어로 구현되기 때문에 웹 환경에서 사용하기 적합하다는 특징을 가진다.

1. 서 론

오늘날 기하급수적으로 늘어가는 정보를 효과적으로 사용하기 위해 다양한 데이타베이스 응용 프로그램들이 만들어지고 있다. 데이타베이스 응용 프로그램은 정보를 저장, 관리하고 있는 데이타베이스 관리 시스템(DBMS)으로부터 정보를 읽거나 기록함으로써 주어진 응용을 처리하는 프로그램이다. 이 프로그램은 DBMS 마다 응용 프로그램을 위해 제공하는 응용 프로그램 작성 인터페이스(Application Programming Interface: API)가 서로 다르기 때문에 사용하고자 하는 DBMS 에 따라 각각 다르게 작성된다[Mic97].

늘어가는 데이타베이스 응용 프로그램 작성에 있어 응용 프로그램 작성의 용이성과 사용의 상호 호환성을 위해 응용 프로그램 작성을 위한 표준 DBMS API 들이 제안되고 있다. 응용 프로그램 개발자는 표준 DBMS API 를 통해 다양한 종류의 DBMS 에 적용 가능한 응용 프로그램을 쉽게 작성할 수 있다. 대표적인 표준 API 로는 X/Open SQL Call Level Interface(CLI)[Ope95] 와 이를 기반으로 개발된 Open DataBase Connectivity(ODBC)[Mic97] 와 Java DataBase Connectivity (JDBC)[Whi99]가 있다. X/Open SQL CLI 는 응용 프로그램이 DBMS 에 표준화된 방법으로 접근하도록 제정된 API 명세로 다른 표준 API 들은 이를 바탕으로 개발되도록 하고 있다. ODBC 는 X/Open SQL CLI 명세를 바탕으로 개발된 API 로 C/C++ 언어를 기반으로 한 응용 프로그램을 위한 API 이다. JDBC 는 X/Open SQL CLI 와 ODBC 명세를 바탕으로 개발된 API 로 Java 언어를 기반으로 한 응용 프로그램을 위한 API 이다.

최근에는 인터넷의 폭발적인 발전과 함께 웹 환경 하에서 Java 를 사용하여 데이타베이스 응용 프로그램을 개발하는 경우가 증가하고 있으며 이와 더불어 JDBC 의 중요도도 점점 높아지고 있다. Java 데이타베이스 응용 프로그램에서 ODBC API 는 언어적 측면과 수행 환경적 측면에서 적합하지 않다. ODBC API 는 C/C++ 언어를 위한 인터페이스이기 때문에 Java 응용 프로그램에서 사용하기 힘들며 Java 응용 프로그램과 함께 네트워크로 전송되어 수행될 수 없다. 반면 JDBC 는 Java 언어의 특징을 고려하기 때문에 이러한 문제점이 없다.

본 논문에서는 객체관계형 데이타베이스 관리 시스템인 오디세우스[Par94]를 위한 JDBC 드라이버를 설계하고 구현한다. JDBC API 를 사용하기 위해서는 각 DBMS 별로 JDBC 드라이버가 필요한데, 오디세우스를 위한 JDBC 드라이버의 개발은 웹 환경 하의 오디세우스 응용 프로그램을 보다 쉽게 작성하도록 해줄 뿐만 아니라 JDBC API 를 사용하여 작성된 기존의 응용 프로그램들을 재사용할 수 있도록 해준다.

본 논문의 구성은 다음과 같다. 제 2 장에서는 관련 연구로서 JDBC 에 대해 간략히 소개하고 구현 방법에 따른 JDBC 드라이버의 타입들에 대해 설명한다. 제 3 장에서는 오디세우스를 위한 JDBC 드라이버의 설계 및 구현방법에 대해 설명한다. 끝으로 제 4 장에서는 결론을 내리고 논문의 공헌을 요약한다.

2. 관련 연구

본 장에서는 데이타베이스 응용 프로그램의 호환성을 위한 표준 API 로서 Java 언어 구현 환경을 위해 개발된 JDBC 에 대해 설명한다. 제 2.1 절에서는 JDBC 를 정의하고 JDBC 아키텍처를 설명한다. 그리고 제 2.2 절에서는 구현 방법에 따른 JDBC 드라이버의 종류에 대해 설명한다.

2.1 JDBC

JDBC 는 Java 응용 프로그램이 DBMS 의 종류에 무관한(independent) 방법으로 데이터베이스에 접근하도록 Sun 사에 의해 X/Open SQL CLI 명세를 바탕으로 개발된 API 로서[Whi99] 현재 최신 버전은 JDBC 2.0 이다. JDBC 는 ODBC 와 마찬가지로 X/Open SQL CLI 명세를 바탕으로 하고 있으며 ODBC 설계사항의 일부분을 유지하고 있기 때문에 기능적인 측면에서 ODBC 와 많은 유사점을 가지고 있다. 그러나

* 본 연구는 첨단정보기술연구센터를 통하여 과학재단의 지원과 웹 기반 정보검색 OODBMS 를 위한 Web-DBMS 통합기술의 개발 과제를 통해 과학기술부의 지원을 받았음.

JDBC 는 Java 언어의 객체 지향 개념이 반영되어 있으며 ODBC 에서의 불필요하게 복잡했던 기능을 단순화시켜서 사용상의 편리성을 추구한다는 점에서 ODBC 와 차이점을 가지고 있다[Whi99].

JDBC 의 아키텍처는 크게 응용 프로그램, JDBC 드라이버 매니저, JDBC 드라이버, DBMS 로 구성되며 그림1 은 이러한 JDBC 의 아키텍처를 보여준다. 응용 프로그램은 JDBC API 를 호출함으로써 SQL 문장을 JDBC 드라이버를 통해 DBMS 로 보내고 그 결과를 얻는다. JDBC 드라이버 매니저는 응용 프로그램과 JDBC 드라이버를 중계하는 모듈로서 응용 프로그램에서 필요로 하는 드라이버를 응용 프로그램과 연결해 주고 응용 프로그램의 JDBC API 호출을 드라이버에게 전달해 준다. JDBC 드라이버는 JDBC API 호출을 실제로 처리하는 부분으로서 주어진 SQL 문장을 해당 DBMS 에 맞게 수정하여 DBMS 로 전송하고 그 결과를 받아 응용 프로그램에게 돌려준다. JDBC 드라이버는 DBMS 의 종류에 무관한 JDBC API 의 호출을 해당 DBMS 에 맞게 처리해야 하는 모듈로서 각 DBMS 마다 다르게 구현되어야 한다.

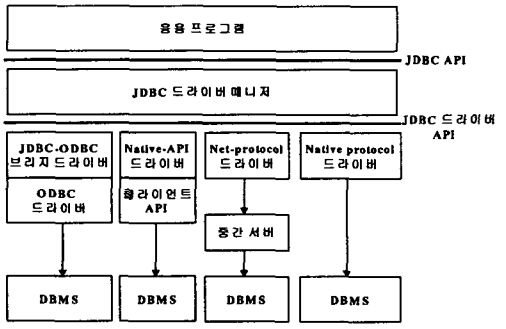

그림 1. JDBC 의 아키텍처

2.2 JDBC 드라이버 타입

JDBC 드라이버는 구현되는 방법에 따라 4 가지 타입으로 분류된다[Whi99]. JDBC 드라이버는 각각 JDBC-ODBC 브리지 드라이버, Native API partly-Java 드라이버, Net-Protocol All-Java 드라이버, 그리고 Native Protocol All-Java 드라이버의 4 가지 타입이 있다. 각 타입별 드라이버의 동작 방식과 특징은 다음과 같다.

- **Type 1 : JDBC-ODBC 브리지 드라이버**
  이미 구현된 ODBC 드라이버를 사용하여 주어진 DBMS 에 접근하는 드라이버다. 이 타입의 드라이버는 응용 프로그램의 JDBC API 호출을 ODBC API 호출로 변환함으로써 해당 DBMS 에 접근한다. 이 타입의 드라이버는 Sun 사에 의해 이미 구현되어 있기 때문에 접근하고자 하는 DBMS 를 위한 ODBC 드라이버가 갖추어져 있다면 이를 통해 JDBC 를 사용할 수 있다.
- **Type 2 : Native API partly-Java 드라이버**
  접근하고자 하는 DBMS 가 제공하는 전용 클라이언트 라이브러리를 사용하여 DBMS 에 접근하는 드라이버다. 이 타입의 드라이버는 응용 프로그램의 JDBC API 호출을 전용 클라이언트 API 호출로 변환함으로써 해당 DBMS 에 접근한다.
- **Type 3 : Net-Protocol All-Java 드라이버**
  Net 서버가 제공하는 Net 프로토콜을 사용하여 Net 서버에 접근한 후 Net 서버를 통해 DBMS 에 접근하는 드라이버다. Net 서버란 JDBC 드라이버와 DBMS 사이에 위치하면서 Net 프로트콜을 DBMS 가 제공하는 네트워크 프로트콜로 변환하는 미들웨어로서 Net 프로토콜은 Net 서버의 개발사마다 다르다. Net 서버가 Net 프로토콜을 여러 종류의 DBMS 에서 제공하는 전용 네트워크 프로토콜로 변환하는 기능을 갖추고 있다면 이 타입의 드라이버를 사용하여 여러 종류의 DBMS 에 접근할 수 있다.
- **Type 4 : Native Protocol All-Java 드라이버**
  접근하고자 하는 DBMS 가 제공하는 네트워크 프로토콜을 사용하여 DBMS 에 접근하는 드라이버다. Type 2 드라이버와의 차이점은 Type 2 드라이버는 이미 구현되어 있는 전용 클라이언트 API 를 사용해서 DBMS 와 통신을 하는 반면에 이 타입의 드라이버는 자체적으로 DBMS 와 통신할 수 있는 프로토콜을 설계하고 구현해야 한다는 것이다. 이 타입의 드라이버는 응용 프로그램의 JDBC API 호출을 접근하고자 하는 DBMS 용의 네트워크 프로토콜로 변환함으로써 직접적으로 해당 DBMS 에 접근한다.

JDBC 드라이버는 4 가지 타입의 드라이버 중에서 어느 타입으로 구현된다 하더라도 모두 동일한 API 를 제공해야 한다. JDBC API 는 모든 DBMS 에 공통된 부분과 DBMS 의 종류에 따라 다르게 구현되야 하는 부분으로 구성되는데, 공통된 부분은 Java Development Kit(JDK)에 Java 언어의 클래스 타입으로 이미 구현되어 있고 다르게 구현되야 하는 부분은 JDK 에 Java 언어의 인터페이스 타입으로 선언만 되어있다. JDBC 2.0 API 는 공통 부분으로 이미 JDK 에 구현되어 있는 10 개의 클래스를 갖고 있다. DBMS 종류에 따라 다르게 구현해야 하는 16 개의 인터페이스 타입은 다시 질의 처리 기능을 위한 8 개의 인터페이스와 SQL3 확장 데이타 타입을 지원하기 위한 8 개의 인터페이스로 구성되어 있다.

3. JDBC 드라이버의 설계 및 구현

본 장에서는 객체관계형 데이타베이스 관리 시스템인 오디세우스를 위한 JDBC 드라이버의 설계에 대해 설명한다. 제 3.1 절에서는 JDBC 드라이버의 전체적인 시스템 구조에 대해 설명하고, 제 3.2 절에서는 JDBC 드라이버의 사용자 인터페이스에 대해 설명한다.

3.1 시스템 구조

JDBC 드라이버의 시스템 구조는 구현하고자 하는 드라이버 타입에 의해 대부분 결정되므로 시스템 구조를 설계하기 전에 먼저 드라이버 타입을 결정하는 것이 필요하다. 제 2.2 절에서 설명한 4 가지 타입의 JDBC 드라이버들은 저마다의 장단점을 가지고 있지만 본 논문에서는 특히 웹 환경에서 사용되기 적합한지와 성능이 좋은지를 고려하여 구현할 드라이버의 타입을 결정한다.

본 논문에서는 4 가지 JDBC 드라이버의 타입 중에서 타입 4 인 Native Protocol All-Java 드라이버를 구현한다. 타입 1 인 JDBC-ODBC 브리지 드라이버는 클라이언트에 ODBC 드라이버와 ODBC 드라이버 매니저를 별도로 설치해야 하기 때문에 웹을 통해 자동으로 설치될 수 없어 웹에서 사용되기 적합하지 않고, ODBC 를 거쳐서 DBMS 와 연결하므로 속도가 느린 문제를 가지고 있다. 타입 2 인 Native API partly-Java 드라이버는 드라이버가 모두 Java 언어로 구현되는 것이 아니기 때문에 웹을 통해 자동으로 설치될 수 없어 웹에서 사용되기 적합하지 않다는 문제를 가지고 있다. 타입 3 인 Net-Protocol All-Java 드라이버는 드라이버가 모두 Java 언어로 구현되기 때문에 웹을 통해 자동으로 설치될 수 있어 웹에서 사용되기 적합하지만 프로토콜의 변환을 위한 중간 서버의 사용으로 인해 속도가 느린 문제를 가지고 있다. 반면 본 논문에서 구현하는 타입 4 인 Native Protocol All-Java 드라이버는 드라이버가 모두 Java 언어로 구현되기 때문에 웹을 통해 자동으로 설치될 수 있어 웹에서 사용되기 적합하고 DBMS 와 직접 연결하여 통신하기 때문에 성능이 좋다는 장점을 가지고 있다[Yan98].

Native Protocol All-Java 드라이버는 드라이버가 모두 Java 언어로 구현되고 DBMS 와 직접 연결하여 통신하므로 Java 언어를 사용한 JDBC 드라이버와 DBMS 간의 독자적인 통신 프로토콜이 필요하며 본 논문에서는 Java RPC(Remote Procedure Call)를 통신 프로토콜로 사용한다. Java RPC 란 기존의 C RPC 와 같은 기능을 가지면서 Java 인터페이스를 제공하는 RPC 를 말한다. 본 논문에서 이용한 Java RPC 는 업계 표준인 Sun 사의 Open Network Computing RPC(ONC RPC)[Sri95]를 바탕으로 구현한 Java RPC[Lee99] 이다. JDBC 드라이버와 DBMS 사이의 통신을 위한 방법으로 Java RPC 를 사용하는 이유는 Java RPC 는 RFC 에 기술된 업계 표준을 따르고 있으며, DBMS 가 Java 언어 인터페이스를 제공하지 않아도 DBMS 와 통신할 수 있기 때문이다.

본 논문에서 Java RPC 를 사용해서 구현한 JDBC 드라이버의 시스템 구조는 크게 오디세우스 JDBC 드라이버 API, 오디세우스 Java 클

라이언트 API, Java RPC 클라이언트 Stub 으로 구성되며 그림 2 는 이러한 JDBC 드라이버의 시스템 구조를 보여준다. Java RPC Stub 이 사용하는 네트워크 프로토콜은 Sun 사의 ONC RPC 와 마찬가지로 표준 규약인 External Data Representation(XDR) 프로토콜을 따르고 있다. JDBC API 의 호출은 Java RPC Stub 에서 최종적으로 이 XDR 프로토콜로 변환되어 오디세우스 DBMS 에 보내지고, 이에 따라 오디세우스 API 가 실행된 결과는 다시 XDR 프로토콜을 통해 JDBC 드라이버로 돌아온다. 오디세우스 Java 클라이언트 API 는 Java 클라이언트 RPC Stub 을 바탕으로 구현되는 오디세우스 API 이다. 오디세우스 JDBC 드라이버 API 는 JDBC API 중 DBMS 의 종류에 따라 다르게 구현해야 하는 인터페이스 부분을 오디세우스 Java 클라이언트 API 를 사용하여 오디세우스에 맞도록 구현하는 API 이다.

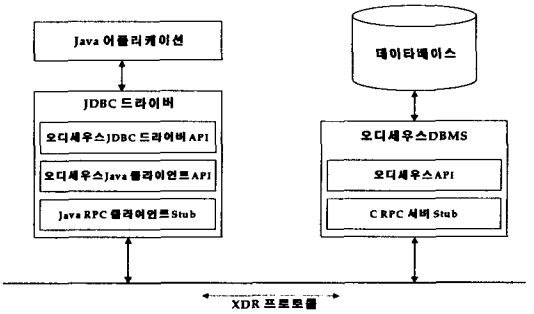

그림 2. 오디세우스 JDBC 드라이버의 시스템 구조

3.2 사용자 인터페이스

본 논문의 JDBC 드라이버의 사용자 인터페이스는 JDBC API 중 DBMS 의 종류에 따라 다르게 구현해야 하는 인터페이스 부분에서 질의 처리를 위한 부분을 구현한다.

본 논문에서 구현하는 8 개의 클래스들은 데이타베이스에 연결하기 위한 2 개의 클래스, 질의문을 수행을 위한 3 개의 클래스, 질의 결과를 처리하기 위한 1 개의 클래스, 그리고 메타 데이타를 얻기 위한 2 개의 클래스로 구성된다. JDBC 드라이버 API 를 구성하는 클래스의 객체를 생성하기 위해서는 생성 관계에 따라 지정된 메소드를 호출하면 된다. 예를 들어 OdysseusStatement 클래스의 객체는 OdysseusConnection 클래스의 createStatement() 메소드를 호출하여 생성한다. 오디세우스 JDBC 드라이버 API 의 각 클래스들에 대한 자세한 설명은 다음과 같다.

- **OdysseusDriver 클래스**
  오디세우스와 연결을 맺고 그 결과로서 OdysseusConnection 클래스 객체를 생성하는 기능과 드라이버에 대한 정보를 관리하는 기능을 가지고 있다.
- **OdysseusConnection 클래스**
  트랜잭션의 완료나 철회와 같은 트랜잭션을 관리하는 기능과 SQL 질의문을 생성하고 준비하는 기능을 가지고 있다.
- **OdysseusStatement 클래스**
  파라미터가 없는 SQL 질의문을 수행하는 기능을 가지고 있다.
- **OdysseusPrepareStatement 클래스**
  OdysseusStatement 클래스의 서브 클래스로서 사전에 미리 준비된(prepared) SQL 질의문을 수행하는 기능을 가지고 있다. 이 클래스에서 관리하는 SQL 질의문은 하나 이상의 입력 파라미터를 가질 수 있으므로 숫자나 문자열의 값만 바꾸면서 똑같은 SQL 질의문을 여러 번 실행해야 할 경우 유용하게 사용될 수 있다.
- **OdysseusCallableStatement 클래스**
  OdysseusPrepareStatement 클래스의 서브 클래스로서 내장 프로시져(stored procedure)를 호출하는 SQL 질의문을 수행하는 기능을 가지고 있다. OdysseusPrepareStatement 클래스를 상속받았기 때문에 수행하는 SQL 질의문은 하나 이상의 입력 파라미터를 가질 수 있다.
- **OdysseusResultSet 클래스**
  SQL 질의문을 수행한 결과 집합을 가져오는 기능과 결과 집합에서 원하는 투플을 갱신하는 기능을 가지고 있다.
- **OdysseusDatabaseMetaData 클래스**
  데이타베이스에 대한 메타 데이타(meta data)를 가져오는 기능을 가지고 있다. 데이타베이스에 대한 메타 데이타의 예로는 테이블들에 대한 정보, 테이블을 이루는 컬럼들에 대한 정보, 컬럼에 대한 인덱스 정보 등이 있다.
- **OdysseusResultSetMetaData 클래스**
  SQL 질의문을 수행한 결과 집합에 대한 메타 데이타를 가져오는 기능을 가지고 있다. 결과 집합에 대한 메타 데이타의 예로는 결과 집합의 컬럼 개수, 각 컬럼의 이름, 각 컬럼의 타입 등이 있다.

3.3 성능 측정

본 논문에서는 구현한 JDBC 드라이버의 성능을 측정하는 기준을 얼마나 많은 질의 결과를 빠른 시간 안에 처리하여 응용 프로그램에서 사용할 수 있도록 준비하는가에 초점을 둔다. 성능 측정 환경은 서버는 Sun UltraSparc2, SunOS 5.6 이고, 클라이언트는 Intel Celeron 466MHz, RAM 128M, Windows 2000 이며, 네트워크는 10M bps LAN 이다. 속도를 측정하는 범위는 JDBC 드라이버에서 DBMS 로 질의어를 전송하는 시점부터 DBMS 로부터 질의 결과를 받아서 응용 프로그램이 사용 가능하도록 만드는 시점까지이다. 속도 측정 결과 평균 10 만건의 레코드를 2 초 안에 처리하는데, 이러한 처리 속도는 일반적으로 드라이버가 DBMS 에 따라 영향을 받는 부분이 있기 때문에 타 JDBC 드라이버들과 직접적인 비교는 할 수 없지만 처리 시간에 있어서 더 빠른 속도를 보여주고 있다.

4. 결 론

본 논문에서는 Java 응용 프로그램에서 오디세우스 객체관계형 데이타베이스 관리 시스템에 연결하여 데이타베이스를 사용할 수 있도록 JDBC 드라이버의 설계 및 구현을 수행하였다.

본 논문에서 설계하고 구현한 JDBC 드라이버를 사용하면 웹 환경하의 오디세우스 응용 프로그램을 쉽게 작성할 수 있을 뿐만 아니라 JDBC API 를 사용하여 작성된 기존의 응용 프로그램들을 재사용할 수 있게 된다.

본 JDBC 드라이버는 JavaRPC 를 사용하여 DBMS 와 직접 연결하여 통신하므로 질의 처리 성능이 좋고, 모두 Java 언어로 구현되기 때문에 웹 환경에서 사용하기 적합하다는 특징을 가지고 있다.

참고 문헌

[Lee99] 이국희, 한옥신, 이민재, 황규영, "ONC RPC 표준을 지원하는 Java RPC 의 설계 및 구현," 한국 정보과학회 추계 학술대회 발표논문집(III), pp. 206--208, 1999.

[Mic97] Microsoft, ODBC 3.0 *Programmer's Reference*, Microsoft Press, Vol.1, 1997.

[Ope95] The Open Group, *Data Management: SQL Call Level Interface (CLI)*, Open Group Technical Standard, 1995.

[Par94] 박종목, 심재균, 이종학, 우준호, 조완섭, 황규영, "ODYSSEUS: UNIX 용 다사용자 객체지향 데이타베이스 시스템," 한국 정보과학회 춘계 학술발표 논문집,Vol. 21, No. 2, pp. 32--34, 1994.

[Sri95] Srinivasan, R., RFC 1831 - Open Network Computing RPC: Remote Procedure Call Protocol Specification Version 2, Sun Microsystems, Aug. 1995.

[Whi99] White, S., Fisher, M., Cattell, R., Hamilton, G., Hapner, M., *JDBC API Tutorial and Reference, 2nd ed.*, Addison Wesley, 1999.

[Yan98] Yang, A., Linn, J., and Quadrato, D., "Developing Integrated Web and Database Applications Using JAVA Applets and JDBC Drivers," In *Proc. ACM SIGSCE*, pp. 302--306, 1998.