

바다-IV/JDBC 드라이버의 구현

김병섭, 이미영
한국전자통신연구원 컴퓨터소프트웨어연구소

Implementation of BADA-IV/JDBC Driver

Byoung-Seob Kim, Mi-Young Lee
Computer & Software Research Lab., ETRI
E-mail : powerkim@etri.re.kr, mylee@etri.re.kr

요약

자바 환경에서 데이터베이스 연동을 위한 표준 인터페이스를 위해서 Sun에서 JDBC를 제시하고 있으며, 본 연구에서는 객체지향 멀티미디어 데이터베이스 바다-IV 시스템에 대한 JDBC 인터페이스를 위한 바다-IV/JDBC 드라이버를 구현하였다. 바다-IV/JDBC 드라이버는 별도의 라이브러리 없이 순수한 자바 코드로 개발하였으며, 데이터베이스 서버에 직접 연동하는 방법을 사용하여 고성능을 추구하였다. 또한, 멀티미디어 데이터베이스를 위한 BLOB, CLOB 등의 변경 기능과 Collection 타입에 대한 기능을 확장 구현하였으며, 사용자는 바다-IV/JDBC를 사용하여 바다-IV 서버와 연결하여 SQL 문을 사용한 자바 응용 서비스의 개발이 가능하며, 동일한 응용 서비스를 JDBC를 지원하는 다른 데이터베이스를 사용한 응용 서비스로 쉽게 변경할 수 있다.

1. 서론

최근 보안성, 이식성, 이동성 등을 특징[1]으로 하는 자바 환경의 사용이 증가하는 추세이다. 자바는 시스템 플랫폼 및 운영체제에 상관없이 동일한 실행 환경을 제공하여 주며, "Write Once, Run Anywhere"[1]에서 의미하는 바와 같이 한 사이트에서 컴파일 하면 어떤 사이트에서라도 수행이 가능한 환경을 제공하여 준다. 이러한 자바 환경에서 기업형 응용 서비스에 필수적인 데이터베이스에 대한 접근을 위해서 표준 인터페이스를 제공하며, 이러한 인터페이스가 JDBC(Java Database Connectivity[2,3])이다. JDBC는 자바 환경에서 데이터베이스를 접근하는 표준 인터페이스를 제시함으로써 JDBC를 제공하는 모든 데이터베이스 시스템이 동일한 사용자 인터페이스를 제공함으로써 서로 다른 데이터베이스 시스템에 대한 접근 방법을 동일하게 제공하고자 하였다. JDBC는 이러한 접근 방법에 대한 자바 인터페이스만 제시할 뿐 실제 인터페이스에 대한 구현은 별도로 제공하지 않는다[3]. 인터페이스에 대한 실제 구현(드라이버)은 데이터베이스 시스템에 의존적이며, 데이터베이스에 따라 서로 다른 구현 코드를 갖는 드라이버를 제공한다.

결과적으로 사용자는 공통된 JDBC 인터페이스를 통하여 다양한 데이터베이스를 동일한 방법으로 접근

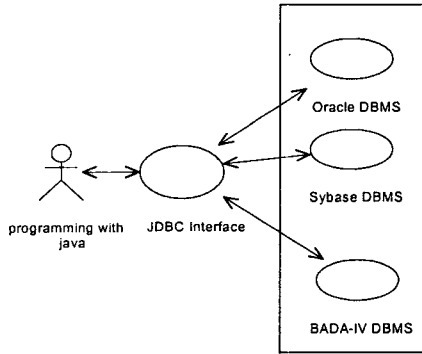
할 수 있게 된다.

본 연구에서는 자바 환경에서의 객체지향 멀티미디어 데이터베이스 시스템인 바다-IV[4] 사용자를 위해서 이러한 JDBC 표준 인터페이스를 제공하기 위한 바다-IV/JDBC 드라이버를 구현하였다.

본 논문의 구성은 다음과 같다. 2장에서는 바다-IV/JDBC 드라이버에 대한 개요를 설명하고, 3장에서는 설계를 4장에서는 구현에 대하여 설명하며, 끝으로 5장에서 결론을 맺는다.

2. 바다-IV/JDBC 드라이버 개요

바다-IV/JDBC 드라이버는 바다-IV 데이터베이스 시스템을 위한 JDBC 드라이버이며, 드라이버 전체 코드가 순수한 자바 코드로 구현되어 있어 플랫폼에 독립적이며 자바 언어가 갖는 특징을 포함한다. 또한, 바다-IV/JDBC 드라이버가 표준 JDBC 인터페이스를 지원하고 있기 때문에 그림 1과 같이 동일한 방법으로 다른 데이터베이스에 대한 접근이 가능하여 바다-IV/JDBC 드라이버를 이용한 응용 서비스를 다른 데이터베이스를 사용한 응용 서비스로의 변경이 용이하다.



(그림 1) JDBC를 사용한 데이터베이스 접근

바다-IV/JDBC 드라이버의 기능은 Sun에서 정의한 JDBC 표준 규격을 기본으로 하며, 객체지향 멀티미디어 데이터베이스 지원을 위한 기능을 추가하였다. 이에 따라 바다-IV/JDBC 드라이버의 구성은 핵심 부분과 확장 부분으로 구성하였으며 각 구성은 다음과 같다

- (1) 바다-IV/JDBC 드라이버 핵심 부분
 - JDBC 1.2 규격 지원
 - 바다-IV/JAVASTUB
(바다-IV 내부 구현 코드)
- (2) 바다-IV/JDBC 드라이버 확장 부분
 - JDBC 2.0 규격 중 일부 지원
 - SQL3 타입
 - ARRAY,
 - BLOB(Binary Large Object),
 - CLOB(Character Large Object),
 - REF
 - Batch 수행
 - Collection 타입 (바다-IV 전용)
 - SET, LIST, MULTISSET 지원

핵심 부분은 드라이버 대부분의 기능을 포함한 JDBC 1.2 표준 규격을 구현하였으며, 바다-IV 서버와 자바 환경 연결을 위한 자바 스템트 코드를 포함한다.

확장부분은 SQL3 타입 지원과 Batch 수행을 위한 JDBC 2.0 규격의 일부 구현과 바다-IV 서버에서 지원하는 Collection 타입을 위한 인터페이스를 추가 제공한다. SQL3 타입에 있어서는 멀티미디어 데이터베이스 서버(바다-IV)의 원활한 지원을 위해서 JDBC 2.0 규격에서는 지원하지 않는 BLOB, CLOB에 대한

변경 기능을 추가 지원한다.

이러한 바다-IV/JDBC 드라이버 사용자는 자바 환경에서 바다-IV 서버에 쉽게 접근할 수 있으며, 서버에 SQL 질의 문 수행을 요구하여 그에 대한 결과를 얻을 수 있으며, 결과 데이터는 자바 환경에서 쉽게 처리가 가능하다.

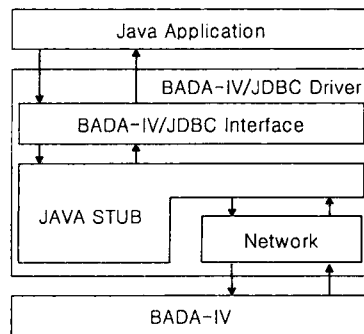
3. 바다-IV/JDBC 드라이버의 설계

일반적으로 JDBC는 구현방법에 따라 다음 4가지 종류로 구분된다[6].

- (1) JDBC-ODBC Bridge plus ODBC Driver
- (2) Native-API partly-Java Driver
- (3) Pure Java Driver for Database Middleware
- (4) Direct-to-Database Pure Java Driver

이러한 4가지 종류 중 바다-IV/JDBC는 고성능을 위해서 "Direct-to-Database Pure Java Driver" 형태로 설계하였으며, 다른 종류와는 달리 별도의 라이브러리가 필요하지 않고 드라이버 전체가 순수 자바로 구현되어 속도가 빠르고 코드 이동성이 뛰어나다.

바다-IV/JDBC 드라이버의 내부 구조는 그림 2와 같이 바다-IV 서버에 의존적인 내부 인터페이스를 위한 자바 스템트 모듈과 바다-IV 서버와의 프로토콜 및 하부 데이터 송수신을 위한 네트워크 모듈로 구성 되어있다.

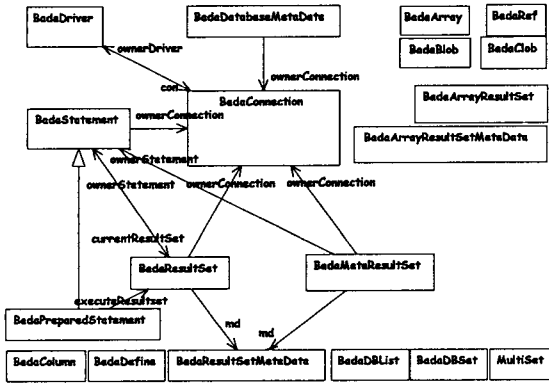


(그림 2) 바다-IV/JDBC 내부 구조

하부 네트워크 모듈을 포함한 자바 스템트 코드는 바다-IV 서버를 자바 환경에서 지원할 수 있도록 제공하는 바다-IV 전용 인터페이스를 제공하며, 자바 클래스 라이브러리 형태로 제공된다. 이 클래스 라이브러리는 원격지의 바다-IV 서버와 자바 환경에서 직접 연결되며, 자바 사용자와 JDBC 드라이버 개발을

위해 필요한 인터페이스를 제공한다. 자바 스타브를 사용한 드라이버 개발은 통신 환경 및 바다 서버와의 복잡한 프로토콜을 단순화 시켜 주고, 추후 기능 확장이 용이하도록 해준다.

바다-IV/JDBC 드라이버는 Sun에서 제시하는 JDBC 인터페이스를 자바 구현(implements)하였으며, 클래스 구조 및 연관관계는 그림 3과 같다.



(그림 3) 바다-IV/JDBC 드라이버 클래스 구조

클래스별 상세 연관관계는 다음과 같다.

- BadaDriver : BadaDriver 클래스는 BadaConnection 클래스 변수 con을 소유하여 서버와의 연결을 관장한다.
- BadaConnection : BadaConnection 클래스는 현재 소속된 드라이버 정보를 유지 하기위해서 BadaDriver 클래스 변수 ownerDriver를 갖는다.
- BadaDatabaseMetaData : 사용자에게 현재 연결된 서버 정보를 제공하기 위해서 BadaConnection 클래스 변수 ownerConnection을 유지한다.
- BadaStatement : 현재 연결된 서버 정보를 유지 하기 위해서 BadaConnection 클래스 변수 ownerConnection을 소유하며 질의 수행에 대한 결과를 사용자에게 제공하기 위해서 currentResultSet을 소유한다.
- BadaPreparedStatement : BadaPreparedStatement는 BadaStatement 클래스로부터 상속되어 동적 질의 처리 부분을 추가 지원한다. BadaStatement 클래스와 유사하게 현재 질의 수행에 대한 결과를 사용자에게 제공하기 위해서 BadaResultSet 클래스 변수executeResultset 를 소유한다.
- BadaResultSet : BadaResultSet은 현재 생성된 연결 정보를 유지 하고자 BadaConnection 클래스

스 변수 ownerConnection을 소유하며, 결과 집합을 요구한 BadaStatement 객체를 유지 하기 위해서 ownerStatement 변수를 소유하며, 결과 집합에 대한 메타 정보를 유지 하기 위해서 BadaResultSetMetaData 클래스 변수 md를 소유한다.

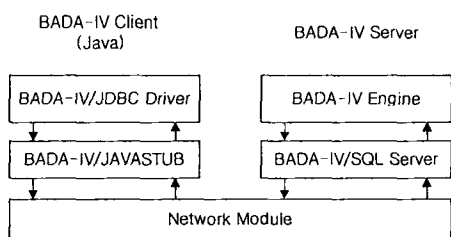
- BadaResultSetMetaData : BadaResultSetMetaData는 사용자에게 현재 마인딩된 클래스의 결과에 대한 메타 정보를 제공해준다.
- BadaMetaResultSet : BadaMetaResultSet은 BadaResultSet과 사용자 인터페이스가 동일하나 "ResultSet BadaDatabaseMetaData::getTypeInfo()"의 반환 값인 결과 셋을 지원하기 위해 별도 제작된 클래스이다. BadaResultSet과 동일하게 md 클래스 변수를 소유한다.
- BadaArrayResultSet : BadaArrayResultSet은 "ResultSet BadaArray:: getResultSet()"의 반환 값인 결과 셋을 지원하기 위해 별도 제작된 클래스이다. 바다-IV 서버의 Array 타입 정책이 로케이터(Locator)를 사용한 방법이 아닌 직접 값을 가져오는 방식이므로 이에 대한 ResultSet 인터페이스 데이터 설정이 기존 결과 셋과는 차이가 생긴다. 이러한 차이를 사용자가 인식하지 않고 사용할 수있도록 BadaArrayResultSet 클래스를 제공한다.
- BadaArrayResultSetMetaData : BadaArrayResultSetMetaData 객체는 "ResultSetMetaData BadaArrayResultSet:: getMetaData()" 호출 시 생성되며 BadaArrayResultSet에서 설명한 바와 같이 기존 ResultSetMetaData 인터페이스의 데이터 처리 방법이 다르므로 별도의 클래스를 제작하였으며 사용자는 동일한 ResultSetMetaData 인터페이스를 통해서 서비스 받는다.
- BadaArray : BadaResultSet 객체가 소유한 결과 셋 중에서 컬럼 타입이 Array 타입인 경우 사용자는 Array 인터페이스를 사용하여 데이터를 자바 환경에서 조작할 수 있다.
- BadaBlob : BadaResultSet 객체가 소유한 결과 셋 중에서 컬럼 타입이 Blob 타입인 경우 사용자는 Blob 인터페이스를 사용하여 데이터를 자바 환경에서 조작할 수 있다.
- BadaClob : BadaResultSet 객체가 소유한 결과 셋 중에서 컬럼 타입이 Clob 타입인 경우 사용자는 Clob 인터페이스를 사용하여 데이터를 자바 환경에서 조작할 수 있다.

- BadaRef : BadaResultSet 객체가 소유한 결과 셋 중에서 컬럼 타입이 Ref 타입인 경우 사용자는 Ref 인터페이스를 사용하여 데이터를 자바 환경에서 조작할 수 있다.
- BadaDBList/BadaDBSet/MultiSet : 바다-IV 서버는 JDBC2.0 표준에 정의되어 있지 않은 Collection 타입을 지원한다. 이에 따라 바다-IV/JDBC 드라이버는 LIST, SET, MULTISSET 세가지 Collection 타입을 제공해야 한다. 이에 대한 사용자 인터페이스 지원을 위해서 BadaList/BadaSet/BadaMultiSet을 제공한다. 사용자는 BadaResultSet 객체가 소유한 결과 셋 중에서 컬럼 타입이 LIST/SET/MULTISSET 타입인 경우 List/Set/MultiSet 인터페이스를 사용하여 해당 데이터를 자바 환경에서 조작할 수 있다.
- BadaColumn/BadaDefine : 바다-IV/JDBC 드라이버의 구현을 위해서 내부에서 사용된다.

4. 바다-IV/JDBC 드라이버의 구현

바다-IV/JDBC 드라이버는 J2SE(Java2 Platform, Standard Edition)[5] 버전 1.2 이상 환경에서 구현하였으며, J2SE의 JDBC 인터페이스를 바다-IV에 맞게 구현하였다.

바다-IV/JDBC 드라이버는 그림 4와 같이 바다-IV 자바 스템 인터페이스를 사용해서 구현하였으며, 자바 스템은 바다-IV 네트워크 모듈을 통해서 서버에 질의 처리와 이에 대한 결과를 요구한다.

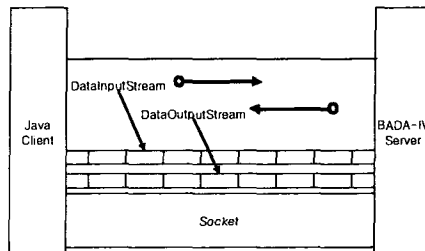


(그림 4) 바다-IV/JDBC 데이터 흐름

클라이언트가 요구한 질의는 바다-IV 서버의 SQL 서버가 분석하여 엔진이 처리하여 필요 시 결과를 클라이언트에게 전달한다.

바다-IV 네트워크 모듈은 그림 5와 같이 클라이언트/서버 세션 생성 시에 자바의 DataInputStream과 DataOutputStream으로 소켓을 바인딩하여 각각 입력, 출력 채널을 형성하여 드라이버와 서버와의 데이터

송수신을 효율적으로 관리할 수 있도록 하였으며, 바다-IV 서버 프로토콜을 해석하여 자바 스템에서 데이터를 의미 단위로 송수신 할 수 있도록 하였다.



(그림 5) 클라이언트/서버 데이터 채널

이러한 네트워크 모듈 위에 바다-IV 자바 스템을 구현하였으며, 주요 클래스 및 기능은 다음과 같다.

- ConnectionHandle
 - 서버와 연결 관리 기능
 - 트랜잭션 관리 기능
 - SQL 문의 수행 기능
 - 커서 관리
 - 동적 SQL 관리
- OmxBlob, OmxBlobHandle
 - BLOB 자료에 대한 지원
- OmxClob, OmxClobHandle
 - CLOB 자료에 대한 지원
- OmxCollection
 - Collection 자료에 대한 지원
- Omid, OmidHandle
 - 객체 ID에 대한 지원
- ResultInfo, ResultInfoList, ValueSpec, ValueSpecList
 - 질의 처리 결과에 대한 지원

이러한 바다-IV 자바 스템 인터페이스를 기반으로 바다-IV/JDBC 드라이버를 구현하였다.

현재 바다-IV/JDBC 드라이버의 구현을 완료하였으며, J2SE 1.2 환경이 가능한 Solaris, Linux, Windows에서 자체 개발한 바다-IV/JDBC Test Suite을 사용하여 시험 완료하였다.

5. 결론

본 연구에서는 바다-IV 데이터베이스 시스템의 자바 개발 환경을 위한 사용자 인터페이스인 바다-IV/JDBC 드라이버 설계 및 구현에 대하여 기술하였

다. Sun사에서는 JDBC 드라이버에 대한 표준안을 제시하였으며, 이에 대한 자바 인터페이스를 제공하고있다. 대부분의 데이터베이스 시스템들은 이러한 인터페이스에 대한 구현으로서 JDBC 드라이버 클래스 라이브러리를 제공하고 있으며 바다-IV/JDBC 드라이버 또한 이러한 구현의 결과이다. 바다-IV/JDBC는 순수한 자바 코드로만 구현이 되었으며, 별도의 라이브러리를 요구하지 않는다. 그러한 이유로 플랫폼에 독립적이며 자바가 갖는 특징을 포함한다. 또한, 멀티미디어 데이터베이스의 원활한 지원을 위해서 BLOB, CLOB에 대한 변경 기능을 추가 지원하고 있다.

Sun에서 배포한 자바 인터페이스가 객체 지향 기반으로 설계되었으며 대부분의 기능 수행은 자바 스타브와 바다-IV 서버에서 진행된다. 이러한 이유로 바다-IV/JDBC 드라이버의 설계 및 구현은 비교적 이해하기 쉽고 확장이 용이하다.

바다-IV/JDBC 드라이버 개발로 인하여 객체지향 멀티미디어 데이터베이스 시스템인 바다-IV 시스템에 대한 자바 환경에서의 응용 서비스 개발이 보다 쉽고 일반화되었다.

[참고문헌]

- [1] <http://www.javasoft.com/docs/books/tutorial/getStarted/intro/>
- [2] <http://java.sun.com/products/jdbc/>
- [3] <http://java.sun.com/j2se/1.3/docs/guide/jdbc/>
- [4] 채미옥, 이미영, 박순영, 김영균, 손덕주, "바다-IV OK 기능 규격 정의서 Rev. 2.0", 한국전자통신연구소
- [5] <http://java.sun.com/j2se/>
- [6] <http://java.sun.com/products/jdbc/datasheet.html>
- [7] van Hoff, Sami Shaio, Orca Starbuck. "Hooked on Java", Sun Microsystems, 1996
- [8] Seth White, Maydene Fisher, Rick Cattell, Craham Hamilton, Mark Hapner, "JDBC API Tutorial and Reference, Second Edition", ADDISON-WESLEY, 1999