

# SQL를 지원하는 Java와 Mnesia의 연동 연구

문찬호<sup>0</sup>

(주)뱅크타운 기술연구소  
chmoon@banktown.com

## A Study on Interoperability with Java and Mnesia Supporting SQL

ChanHo Moon<sup>0</sup>

Dept. of R&D Center, Banktown Inc.

### 요 약

Erlang/OTP(Erlang Open Telecom Platform)의 Mnesia는 실시간 고장 허용 분산 MMDBMS이다. Mnesia는 메모리 상에서의 질의 처리와 자동화된 데이터 분산 처리로 신속한 트랜잭션 처리가 가능하지만 표준화된 DB 인터페이스와 SQL을 지원하지 않기 때문에 활용상에서의 한계를 갖고 있다. 본 논문에서는 SQL을 지원하는 Java와 Mnesia의 연동에 대해 연구한다. 이를 위해 SQL 컴파일러와 Mnesia 인터페이스를 제공하는 Java 라이브러리를 개발하였다. SQL을 이용한 Java와 Mnesia의 연동에 대한 성능 분석 결과 성능의 우수성과 사용의 편리성을 모두 갖추게 되었다.

### 1. 서론

컴퓨터 하드웨어 및 통신 기술의 발달과 인터넷의 확산 등으로 데이터가 양적/질적으로 증가함에 따라 이들을 효율적으로 관리하고 활용하기 위한 DBMS의 중요성은 날로 증가하고 있다. 그리고 Oracle, SQL Server, DB2 등의 상용 DBMS는 국외뿐만 아니라 국내의 DBMS 시장을 독점하고 있다. 그러나 이들 DBMS는 대부분 1970년대에 제안된 디스크 기반의 데이터 저장 관리 구조를 갖는다. 한편 최근의 반도체 메모리 가격의 하락으로 말미암아 메모리 기반의 DBMS(MMDBMS)의 활용이 점차 커져가고 있고, 국내 MMDBMS 기술이 상당히 발전되어 있는 것으로 알려져 있다. 또한 최근 MMDB는 다양한 최신 기술을 적용함으로써 디스크 기반 DB와 동일한 수준의 안정성을 보장하고 있다. 따라서 MMDB는 통신이나 금융 거래 등과 같이 동시에 다수의 사용자에 대한 데이터 접근을 신속히 처리해야 하는 분야에 매우 적합하다.

Erlang/OTP(Erlang Open Telecom Platform)은 오픈 소스로서 통신 응용을 개발하고 운용하기 편리하도록 지원하는 플랫폼으로 내부적으로 Mnesia라는 DBMS를 포함한다[1]. Mnesia는 실시간 고장 허용 분산 메인 메모리 DBMS(real-time fault-tolerant distributed main memory DBMS)이다[2][3]. 메모리 상에서의 질의 처리와 자동화된 데이터 분산 처리로 트랜잭션의 신속 처리가 가능하기 때문에 Mnesia를 이용한 데이터 저장 관리는 고 가용성(HA : high availability)을

획득할 수 있다. Erlang/OTP는 소켓 통신을 통해 다른 응용 프로그램과 쉽게 데이터를 교환할 수 있다. 그러나 Mnesia에 대한 ODBC나 JDBC와 같은 표준화된 DB 인터페이스를 제공하지 않기 때문에 프로그래머가 Mnesia 응용 프로그램을 개발하기에 매우 불편하다. 또한 Mnesia는 SQL을 지원하지 않기 때문에 Mnesia를 범용적으로 사용하기에는 한계점을 가지고 있다. 따라서 Mnesia가 SQL을 지원하고, 표준화된 DB 인터페이스를 제공하면 Mnesia의 활용 분야는 다양해질 수 있다. 본 논문은 분산 환경에서 SQL을 지원하는 Java와 Mnesia의 연동에 관해서 연구한 것이다. 이를 위해 SQL을 처리할 수 있는 컴파일러와 Java로 접근 가능한 Mnesia 인터페이스를 개발하였다.

본 논문의 구성은 다음과 같다. 2절에서는 Erlang/OTP와 Mnesia에 대해 간략히 기술한다. 3절에서는 SQL을 지원하는 Java와 Mnesia의 연동에 대해 기술한다. 4절에서는 SQL을 이용한 Java와 Mnesia의 연동 성능에 대해 분석한다. 마지막으로 5절에서 결론을 맺는다.

### 2. Erlang/OTP와 Mnesia

Erlang은 스웨덴 에릭슨(Ericsson)사 전산 연구소에서 개발된 병행 및 분산 시스템을 위한 선언형 언어(declarative language)이고[4][5], OTP는 통신 상에서의 응용 구축을 지원하는 개발 시스템 플랫폼이며, 제어 시스템 플랫폼이다[1]. OTP 전체는 Erlang으로 구현되어 있다.

Erlang/OTP는 모놀리식(monolithic) 플랫폼이 아니라 여러 개의 블록 구성 요소(building block)로 이루어져 있다. 이 중 Mnesia는 MMDB로서 실시간 고장 허용 분산 기능을 지원한다[2][3].

통신 응용에서는 전통적인 DBMS가 제공하는 기능과는 다른 요구 사항들이 존재한다. Mnesia는 이들을 만족하기 위해 다음의 기능상 특징을 갖는다.

- 신속한 실시간 키/값으로 검색<sup>1</sup>
- 동적 재구성(re-configuration) : 실행 시간(run-time)동안 DB 스키마를 동적으로 재구성할 수 있다.
- 고성능 고장 허용 : 여러 노드<sup>2</sup>에 테이블을 복제(replication)함으로써 고장을 허용한다.

Mnesia는 자체 내부 함수(데이터 정의(data-definition)와 데이터 조작 처리(data-manipulation)를 수행) 이외에 SQL과는 다른 Mnymosyne[6]이라는 독자적인 질의어를 사용한다. Erlang/OTP는 메시지 교환 시 소켓 통신하기 때문에 java, c와 연동이 용이하나 Mnesia는 ODBC나 JDBC와 같은 표준화된 DB 인터페이스를 제공하지 않는다. 표 1은 분산 환경에서 디스크 기반 DBMS인 MySQL과 Mnesia의 성능을 비교한 것이다. 하나의 테이블을 대상으로 레코드 검색 TPS(transaction per second)를 나타낸 것이다.

표 1 MySQL과 Mnesia의 검색 성능 비교(TPS)

| MySQL | Mnymosyne | mnesia 내부 함수 이용 |
|-------|-----------|-----------------|
| 213   | 874       | 1509            |

표 1에서 나타난 바와 같이 디스크 기반 DBMS보다 Mnesia의 검색 성능이 최소 4배에서 최대 8배로 좋게 나타났다. 그러나 Mnymosyne(또는 Mnesia 내부 함수)를 사용할 경우 조인과 같은 복잡한 질의에 대해서는 Mnesia의 성능이 떨어질 수 있다. 그러나 최소한의 테이블을 대상으로 한 질의 처리에 대해선 고 가용성을 획득할 수 있다. 예를 들어, 웹 미들웨어 상에서의 제어 정보 공유, 전자 상거래나 금융 서비스에서의 사용자 로그인 정보 검색, 이기종 시스템 간의 메시지 큐, 온라인 게임 중 맵 상에서의 위치 정보 검색 등은 이들의 데이터를 저장한 최소한의 테이블을 다수의 사용자가 매우 빈번하게 접근하게 된다. 따라서 이와 같은 시스템에서 Mnesia를 활용하면 시스템의 성능이 좋아질 수 있다.

### 3. SQL을 지원하는 Java와 Mnesia의 연동

본 절에서는 SQL을 지원하는 java와 Mnesia의 연동을 위해 SQL 컴파일러와 Mnesia 인터페이스를 제공하는 java 라이브러리에 대해 기술한다. 또한 처리 가능한 SQL 질의

<sup>1</sup> Mnesia는 해쉬 인덱스(hash index)는 제공하나 순서 인덱스(ordered index)를 제공하지 않는다. 따라서 일치(exact match) 검색의 복잡도는 O(1)이나 범위(range) 검색의 복잡도는 O(n)이다(n은 레코드의 총 수).

<sup>2</sup> 노드는 다른 Erlang/OTP 시스템과 통신할 수 있는 Erlang/OTP 시스템.

유형에 대해 기술한다.

#### 3.1 시스템 구성

SQL을 지원하기 위해 Erlang/OTP 측에는 SQL 컴파일러<sup>3</sup>를 개발하고, Java와 Mnesia의 연동을 위해 java 측에는 java 라이브러리를 개발하였다. 그림 1은 SQL을 이용한 java와 Mnesia의 연동 시스템 구조를 나타낸 것이다.

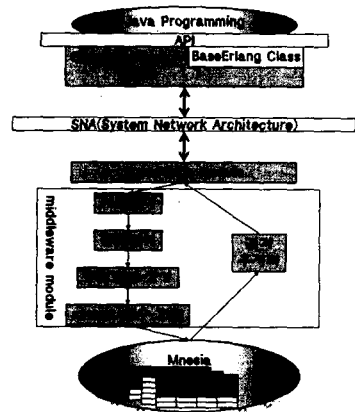


그림 1 SQL을 이용한 java와 Mnesia의 연동 시스템 구조

SQL 컴파일러 모듈의 구성은 다음과 같다.

- 연결 모듈(connection module) : java 측과의 통신을 담당한다
- 미들 모듈(middle module) : 어휘 분석, 구문 분석, 중간 코드(immediate code) 생성, 중간 코드의 내용에 따른 Mnesia 함수 호출로 질의를 수행하고 처리 결과를 후처리하여<sup>4</sup> java 측으로 전달한다.

java 라이브러리(클래스)의 구성은 다음과 같다.

- ErlangConnection 클래스 : java 노드를 설정하고, Erlang/OTP와의 연결자를 획득한다. SQL을 Erlang/OTP 객체로 변환하고 질의 결과를 얻는다.
- BaseErlang 클래스 : 질의 결과를 java 데이터 타입으로 변환하고, 검색 결과에 대한 커서 기능을 제공한다<sup>5</sup>.

<sup>3</sup> [7]은 Erlang/OTP에서 SQL 컴파일러를 개발하였다. 그러나 이는 Erlang 모듈 중 하나로 기능을 가질 뿐 다른 언어를 위한 DB 인터페이스를 제공하지 못한다. 또한 SQL 92 기능 전체를 지원하는 것을 목표로 하기 때문에 성능이 우수하지 못하다.

<sup>4</sup> Mnesia에 대한 질의 결과는 항상 결과 메시지와 결과셋의 쌍으로 구성된다(예를 들어 {ok, {{person, 1, '홍길동', ...}, {person, 2, '김영희', ...}, ...}} 임). 따라서 이 결과 중 결과 셋만을 추출하는 후처리가 필요하다.

<sup>5</sup> Erlang에서 java로 변환가능한 데이터 타입은 byte, int, long, float, double, String 이며, first, last, next, prev에 대한 커서 기능을 제공한다.

### 3.2 처리 가능한 SQL 질의 유형

SQL을 이용한 java와 Mnesia의 연동 시스템은 연구 초기 단계이기 때문에 처리할 수 있는 SQL 질의 처리 유형은 다음과 같다.

- 검색 대상 테이블은 하나이다.
- 데이터 정의에 해당되는 테이블 생성, 삭제 그리고 데이터 조작 처리에 해당되는 레코드 삽입, 삭제, 변경, 검색의 SQL 문을 처리할 수 있으나 조인 연산과 같은 복잡한 SQL 문은 처리하지 못한다.
- 순차적인 일련번호를 생성할 수 있는 sequence를 생성하고 획득할 수 있다. 예를 들어 "select dtd\_id\_seq.nextval from dual"로 sequence를 획득할 수 있다.

### 4. 성능 평가

본 절에서는 개발된 SQL를 이용한 java와 Mnesia의 연동 시스템의 성능을 분석한다.

표 2는 실험 환경을 정리한 것이다.

표 2 성능 평가를 위한 실험 환경

|          | Erlang/OTP 측                             | Java 측                                  |
|----------|--|---|
| 시스템 사양   | Intel Celeron dual CPU 2.4GHz, 512MB 메모리 | Intel Pentium 4 CPU 2.41 GHz, 512MB 메모리 |
| OS       | Red Hat Linux 9                          | Windows XP                              |
| 네트워크 환경  | LAN                                      |   |
| 실험 데이터   | 40byte 크기의 레코드 1,000,000개                |   |
| 동시 사용자 수 | 1 ~ 10명                                  |   |

그림 2는 Mnesia를 대상으로 Mnesia의 내부 함수(read, dirty\_read<sup>6</sup>)와 Mmemosyne 그리고 java 인터페이스를 이용한 SQL 처리(이하 Java\_SQL)의 TPS를 나타낸 것이다. 그림에서 나타난 바와 같이 dirty\_read의 성능이 가장 우수하고 Java\_SQL, read, 그리고 Mmemosyne 순으로 성능이 우수하게 나타났다. 따라서 Java\_SQL는 성능의 우수성과 사용의 편리성을 모두 갖추었음이 입증된다. 결국 2절에서 언급한 최소한의 테이블을 다수의 사용자가 매우 빈번하게 접근하는 시스템에서 SQL과 java와 연동을 지원하는 Mnesia를 이용하면 고 가용성을 얻게 될 수 있다.

6 read는 트랜잭션 처리를 지원하는 검색 함수이고, dirty\_read는 트랜잭션의 처리를 신속히 하기 위해 ACID 특성 중 일부를 위반하는(atomicity와 isolation을 보장하지 못함) 트랜잭션 처리를 지원하는 검색함수이다. dirty\_read를 적절히 이용하면, 트랜잭션을 포함한 응용의 신속한 처리가 가능하다.

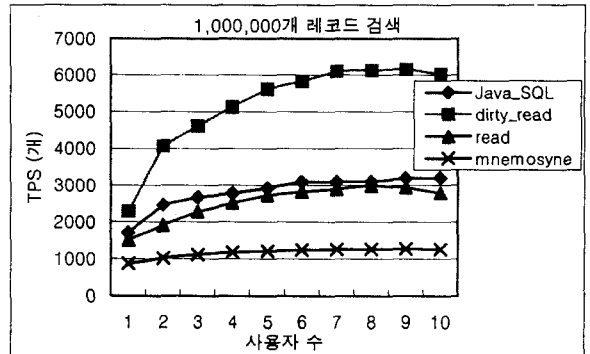


그림 2 사용자 수의 증가에 따른 레코드 검색 TPS

### 5. 결론

Mnesia는 우수한 성능을 지원함에도 불구하고 범용성과 사용의 편리성을 제공하지 못하기 때문에 광범위하게 사용되기 힘든 MMDBMS이다. Mnesia가 표준 SQL을 지원하고 Mnesia 응용 프로그램을 개발을 위한 다양한 프로그래밍 언어로 구성된 인터페이스를 제공한다면 Mnesia는 널리 사용될 수 있다. 본 논문에서는 분산 환경에서 SQL을 지원하는 java와 Mnesia의 연동에 관해서 연구하였다. SQL 컴파일러와 java로 접근가능한 Mnesia 인터페이스를 개발하였다. Mnesia를 대상으로 SQL로 질의 처리한 경우 성능 및 사용의 편리성을 모두 갖추어 줄 수 있게 되었다. 향후 연구로는 조인 연산, 중첩 연산을 지원하는 다양한 SQL을 처리할 수 있는 컴파일러를 개발하고 최적화시키는 것이고, java로 접근가능한 Mnesia 인터페이스를 확장하여 JDBC 인터페이스를 제공하는 드라이버를 개발하는 것이다.

### 참고문헌

- [1] S. Torstendahl, "Open Telecom Platform," Ericsson Review, No. 1, 1997, pp. 14-23, [http://www.ericsson.com/about/publications/review/1997\\_01/index.shtml](http://www.ericsson.com/about/publications/review/1997_01/index.shtml).
- [2] H. Nilsson et al., "Mnesia - A Distributed Robust DBMS for Telecommunications Applications," Int'l Workshop of Practical Aspects of Declarative Languages 1999(PADL99), Jan. 1999, pp. 152-163.
- [3] Erlang Org., "Mnesia Reference Manual version 4.1.4," <http://www.erlang.org/doc/r9c/lib/mnesia-4.1.4/doc/html/index.html>, 2003.
- [4] J. Armstrong et al., "Concurrent Programming in ERLANG," Prentice Hall Europe, 1996, <http://www.erlang.org/doc.html>.
- [5] Erlang Org., <http://www.erlang.org/>.
- [6] Erlang Org., "Mmemosyne Reference Manual version 1.2.5," <http://www.erlang.org/doc/r9c/lib/mmemosyne-1.2.5/doc/html/index.html>, 2003.
- [7] R Andersson, "SQL-compiler," Master thesis, Chalmers Univ. of Technology and Goteborg Univ., 1998.