
Linux환경에서 SQLite 데이터베이스의 검색 성능 실험

김수환*, 최진오*

*부산외국어대학교 임베디드소프트웨어학과

Search Performance Experiments of SQLite Database on Linux Environments

Soo-hwan Kim*, Jin-oh Choi*

*Department of Embedded Software, Busan University of Foreign Studies

E-mail : shkim@bufs.ac.kr, jochoi@bufs.ac.kr

요 약

SQLite, Realm 등은 리소스 제약이 큰 소형 기기에 적합한 데이터베이스 엔진들이며 리눅스 기반 모바일 기기에 많이 사용되고 있다. 이 엔진들은 대부분 오픈소스 프로그램들이며 범용 데이터베이스에 비해 가볍고 속도가 빠른 장점을 지닌다. 이 논문에서는 리눅스 기반 환경에서 SQLite 데이터베이스의 검색 성능을 파악하기 위한 테스트 프로그램을 구현하고 성능 실험을 실시한다. 실험은 같은 환경에서 실행되는 Oracle 데이터베이스와 비교하여 진행한다.

ABSTRACT

SQLite, Realm, etc. are database engines which are suitable for small devices that have large resource restriction and are used a lot in mobile devices based on Linux. These engines are open source programs and have advantages which are lighter weight and faster speed than general purpose databases. This paper implements performance test program and performs performance experiment of SQLite database to test search efficiency in Linux environments. The experiment is processed by comparison with the results of a Oracle database working out at the same environment.

키워드

SQLite, Mobile Database, Performance Test, Linux

I. 서론

모바일 장치의 보편화와 함께 메모리 장치의 가격 하락으로 인한 대용량화가 급속히 진행되고 있는 현실이다. 따라서 비약적으로 모바일 데이터베이스의 필요성이 커지고 있으며 모바일 기기에서 처리하는 데이터베이스가 점점 많아지고 있다.

모바일 데이터베이스의 활용이 보편화되는 사례로 먼저 최근 라즈베리파이(Raspberry Pi)를 이용한 데이터베이스 서버 구축이 많이 시도되고 있다. 또한 모바일 장치의 센스 기술이 발전하여 모바일 장치로 수집되는 센싱 데이터를 로컬 데이터베이스에 저장하였다가 추후 마스터 데이터베이스와 동기화하는 기술이 등장하고 있다. 그리고 mobile edge computing[1] 기술이 보편화되면서 활용도가 점차 증가하고 있다.

이러한 환경을 바탕으로 이 논문에서는 모바일 데이터베이스의 선두 주자로 꼽히는 SQLite의 성

능에 주목하고자 한다. 모바일 환경의 대세인 리눅스 환경에서 검색 성능에 초점을 맞추어 분석한다. 성능 실험을 위한 테스트 프로그램을 구현하여 실험을 실시한다. 테스트 프로그램은 모든 가능한 쿼리 유형에서의 성능을 실험할 수 있도록 구성된다. 성능 분석은 같은 환경에서 실험한 Oracle 11g와 비교하여 실시한다.

II. 관련 연구

SQLite는 2000년 8월에 미국의 Richard Hipp의 주도의 오픈소스 프로젝트로 제작되어 V1.0을 발표하였고 현재 V3.12.2를 제공하고 있는 대표적인 모바일 데이터베이스이다. SQLite는 모바일 데이터베이스 분야의 선두주자라 할 수 있다.

최근의 경우 2014년 새로운 모바일 데이터베이스인 Realm[3]이 출시되어 다양한 모바일 기업과

모바일 앱에서 사용되고 있다.

ORMLite(Object Relational Mapping)는 Java와 관계 데이터베이스 사이의 매핑을 제공하는 오픈 소스 프레임워크로서 안드로이드 플랫폼에서 JDBC를 지원한다. Java와 SQLite의 연결도 지원한다.

Core Data는 Apple에서 제공되는 객체 저장 프레임워크로서 OS X와 iOS에서 구동된다. 데이터를 관계형 모델로 저장할 수 있으며 추상화된 엔티티-릴레이션 구조로 표현할 수도 있다. 저장은 XML, binary 또는 SQLite에 할 수 있다.

그 외에도 페이스북에서 제공하는 모바일 데이터베이스 플랫폼인 Parse[4], Apache 프로젝트 중 하나인 CouchDB[5] 등이 등장했다.

이 중에서 현재 가장 많이 사용되고 있는 SQLite는 현재 SQLite V3.12 까지 출시되어 있다. 안드로이드 OS를 비롯한 다양한 플랫폼에서 작동되며 다른 데이터베이스 제품에 비해 가볍고 속도가 빠르다고 한다. 주요 특징으로는 단일 데이터베이스 파일을 이용하여 서버로 구동되지 않고 라이브러리 형태의 API로 작동한다는 것이다. 동적 자료형을 지원하여 편리한 코딩이 가능한 장점도 지니고 있다.



그림 1. SQLite Site

III. 성능 테스트를 위한 검색 쿼리 유형

쿼리는 검색문을 대상으로 한다. 검색 쿼리에 대한 유형은 다음과 같다.

1. Point Query : 단일 레코드 검색
2. Set Query : 무작위로 복수 레코드 검색
3. Range Query : 순차적으로 레코드들 일부분 검색
4. Multiple Condition Query : 복수 조건 검색
5. Join Query : 조인 검색

Point 쿼리는 키 값을 조건으로 한 레코드를 검색하는 것이고 Set 쿼리는 키가 아닌 값을 조건으로 무작위의 여러 레코드가 검색되는 유형이다. Range 쿼리는 일정 범위안의 순차 데이터를 검색하는 유형이고 다중 조건 쿼리는 where절의 조건이 2 이상일 경우의 쿼리로서 통계정보를 이용한

질의 최적화 여부를 판별할 수 있다. Join 쿼리는 조인 성능을 판별하기 위한 유형이다.

IV. 실험 결과

실험 환경은 다음과 같다. 리눅스는 Fedora 11, Kernel 2.6.29.4 버전을 사용하고 실험 컴퓨터 CPU는 Intel Core Duo, 2.83Gh, 그리고 2Gb Memory를 가진다. SQLite는 V3.12.2 버전을 이용한다. Oracle은 11g 버전을 사용하여 테스트를 진행한다.

그림 2는 Point Query에 대한 실험 결과이다. SQLite가 더 우수한 것은 가벼운 데이터베이스이기 때문이다.

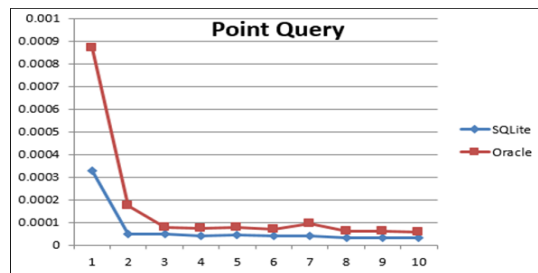


그림 2. Point Query 실험 결과

그림 3은 Set Query의 실험 결과로서 Oracle이 우수하였다. 통계 정보의 사용으로 Oracle이 우수한 것으로 파악된다.

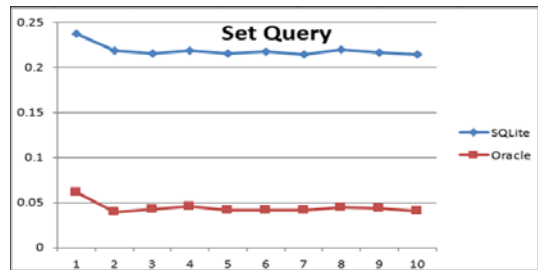


그림 3. Set Query 실험 결과

그림 4는 Range Query의 실험 결과로서 서로 유사한 성능을 보였다.

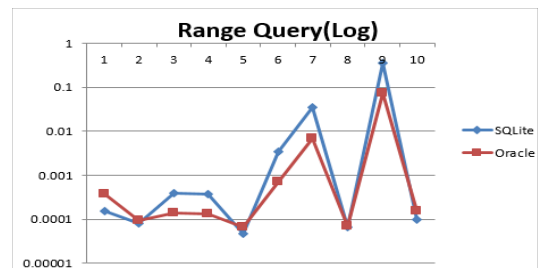


그림 4. Range Query 실험 결과

그림 5는 Multi-Conditions Query의 실험 결과로서 Oracle이 우수하였다. 통계 정보의 사용으로 Oracle이 우수한 것으로 파악된다.

[5] <http://couchdb.apache.org>

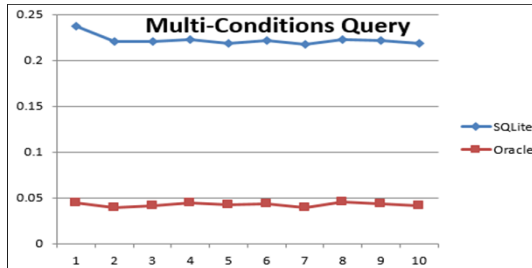


그림 5. Multi-Conditions Query 실험 결과

그림 6은 Join Query의 실험 결과로서 인덱스의 사용으로 성능 차이가 거의 나지 않는 것으로 측정되었다.



그림 6. Join Query 실험 결과

V. 결론

이 논문에서는 Linux 환경에서 SQLite 데이터베이스의 검색 쿼리 유형에 따른 성능을 측정하고 Oracle 데이터베이스와 결과와 비교 분석하였다.

성능 분석 결과 SQLite가 Point Query에서 Oracle보다 우수하였으며, 통계 정보를 이용하는 Oracle의 우수한 점도 관찰되었다. 그러나 대부분의 경우 유사한 성능을 보였다.

향후 Realm 등 경쟁 모바일 데이터베이스와의 성능 비교도 필요해 보인다. 그리고 안드로이드 등 다양한 모바일 플랫폼 환경에서 동일한 실험으로 성능 비교 분석이 필요하다.

참고문헌

- [1] M. Beck, M. Wemer, S. Feld, "Mobile Edge Computing: A Taxonomy," 6th Conference on Advances in Future Internet, pp. 48-54, 2014
- [2] <http://www.sqlite.org>
- [3] <http://realm.io/kr>
- [4] <http://www.parse.com>