

SQLite3 모바일 데이터베이스의 갱신 성능 비교 구현

최진오

부산외국어대학교

Implementation of Modification Performance Comparison of SQLite3 Mobile Databases

Jin-oh Choi

Busan University of Foreign Studies

E-mail : jochoi@bufs.ac.kr

요 약

모바일 IT 기기 환경에서 데이터베이스 엔진 사용이 보편화되고 있으며 모바일에 적합한 데이터베이스 엔진 개발과 발표가 뒤따르고 있다. 이 논문에서는 대표적이고 우수한 모바일 데이터베이스인 SQLite3를 선택하여 갱신 성능 및 특성을 테스트하기 위한 실험을 실시한다. 실험 결과를 평가하기 위하여 동일한 환경의 Oracle 데이터베이스의 결과와 비교한다.

ABSTRACT

In mobile IT device environments, it is becoming common to use database engine, and new mobile databases are continuously developed and announced. In this paper, the most common mobile database, SQLite3 is selected and experimented to test and understand the update performance and characteristics. The results of experiment are compared with the one of Oracle database at the same condition to evaluate the experiment.

키워드

SQLite3, Mobile Database, Update Performance, Oracle Test

I. 서 론

모바일 IT 환경에서 최근 가장 주목받는 변화는 계산 성능의 획기적 향상, 저장 용량의 대폭적인 증가, 디스플레이 기술의 정교한 발전으로 꼽을 수 있다. 이로 인하여 오늘날의 모바일 IT 기기는 모바일 인터넷 디바이스의 지위에서 모바일 고성능 컴퓨터로 그 위상이 바뀌었다.

따라서 스마트폰으로 대표되는 모바일 IT 디바이스는 단말기로 작동하던 지위에서 벗어나 서버나 미들웨어의 역할도 수행할 수 있는 상황을 만들고 있다. 그런데 그 역할을 수행하기 위해서는 데이터베이스의 설치 및 운용이 반드시 필요하다. 따라서 모바일 기기에 최적화된 모바일 데이터베이스의 필요성이 점점 커지고 있으며 새로운 모바일 데이터베이스가 지속적으로 새로 개발되고 발표되고 있다.

이 논문에서는 최근 관심의 초점이 모아지고 있는 모바일 데이터베이스들 중 가장 선두 주자로 꼽히는 SQLite3를 선정하여 갱신 성능을 테스트하는 실험을 진행하고자 한다. 선행 연구[1]에서 SQLite3의 검색 성능의 비교 분석에 이어, 여기에서는 SQLite3의 갱신 성능을 테스트하고 평가하기 위한 실험을 진행한다. 실험 결과를 비교 분석하기 위해 동일한 환경에서 동일한 조건으로 Oracle 데이터베이스에 대하여도 실험을 실시한다. 그리고 결과를 비교 분석하여 SQLite3의 갱신 성능을 평가한다.

II. 관련 연구

모바일 데이터베이스의 활용이 보편화되는 사례로 먼저 최근 라즈베리파이(Raspberry Pi)나 아두이

노(Arduino)를 이용한 데이터베이스 서버 구축이 많이 시도되고 있다[2]. 또한 모바일 장치의 센싱 기술이 발전하여 모바일 장치로 수집되는 센싱 데이터를 로컬 데이터베이스에 저장하였다가 추후 마스터 데이터베이스와 동기화하는 기술인 mobile edge computing[3] 기법이 등장하여 활용도가 점차 증가하고 있다. 그리고 e-commerce나 m-commerce 분야에서도 모바일 디바이스의 사용이 보편화되면서 모바일 디바이스에서의 파일로 처리 불가능한 데이터 처리에 대해 모바일 데이터베이스가 중요한 해법으로 등장하게 되었다[4].

[1]에서 대표적인 모바일 데이터베이스인 SQLite[5]의 검색 성능을 다양한 실험을 통해 수치화해서 보였다. 그리고 비교를 위해 Oracle 11g의 성능을 같이 평가해 보였다. 그러나 이 연구는 Linux 환경에서 실시하였고 동일한 모바일 데이터베이스와 비교하지 않아 본 연구와 차별화된다.

SQLite는 2000년에 발표된 만큼 일찍 개발되어 널리 사용되며 그 성능을 인증 받고 있다. 미국의 Richard Hipp에 의해 개발되었으며 오픈소스 프로젝트로 제작되어 현재 V3.23.1 버전까지 지원되고 있다.

Realm[6]의 경우 2014년 새로운 모바일 데이터베이스로 출시되어 다양한 모바일 기업과 모바일 앱에서 사용되고 있으며 SQLite의 경쟁자로 등장했다.

이 연구에서는 SQLite3 모바일 데이터베이스에 대한 갱신 성능 평가 및 비교를 위한 구현과 실험을 진행한다. 그리고 비교 결과를 분석하여 그 의미를 해석하고자 한다.

III. 성능 테스트를 위한 갱신 쿼리 유형

쿼리는 갱신문을 대상으로 한다. 실험해서 평가하고자 하는 갱신 쿼리는 모든 가능한 갱신 유형을 포함하도록 구성하였다. 실험하는 갱신 쿼리 유형은 다음과 같다.

1. Insert Query

1) 한 레코드씩 insert, 100회 반복, 평균 속도 평가

2) 10만개 레코드 반복 입력, 1) 결과와 비교 평가

2. Update Query

- 1) Point Update, with no index
- 2) Point Update, with index
- 3) Range Update, with no index
- 4) Range Update, with index
- 5) Full Table Update

3. Delete Query

- 1) Point Delete, with no index
- 2) Point Delete, with index
- 3) Range Delete, with no index
- 4) Range Delete, with index

5) Full Table Delete

‘Point’ 쿼리는 키 값을 조건으로 한 레코드를 검색하여 Update하거나 Delete 하는 것이고 ‘Range’ 쿼리는 일정 범위안의 순차 데이터를 검색하는 유형이다.

Insert 쿼리에서 Index의 유무를 구분하여 실험하는 것은 무의미하여 생략하였다.

IV. 실험 결과

실험 환경은 다음과 같다.

1. 리눅스: CentOS 6.7 64bit, Kernel 2.6.32.
2. 실험 컴퓨터: CPU Intel Xeon E5-2690 Core 8 2.60GHz, 32Gb Memory
3. SQLite3: V3.25.2
4. Oracle: 11g (11.2.0.1.0)

SQLite3의 성능 실험을 위한 구현은 ESQL/C로 작성하여 gcc로 컴파일하였다. Oracle의 성능 실험을 위한 구현은 Pro*C로 구현하였고 Oracle의 proc 처리기와 gcc로 컴파일하였다. 실험 데이터는 임의로 생성한 1백 만 개의 레코드(8개의 필드)를 가진 테이블을 사용하였다. 키가 아닌 필드들은 동일한 값들을 최대 20%까지 가진다. 실험 결과에서 X축은 측정 방법의 종류를, Y축은 초(Sec.)의 단위를 가진다.

Insert 쿼리는 SQLite3와 Oracle 모두 일정 시간 간격으로 큰 지연 현상을 보였다. 이는 버퍼 처리 지연으로 보인다. Oracle에 비하여 SQLite3의 Insert 쿼리 처리 성능이 월등히 좋았다. 또한 SQLite3의 경우 한 Insert 쿼리 처리 평균 시간이 0.049sec이며 10만 개 레코드 입력 처리 시간이 4345sec로 거의 비례하였지만 Oracle은 평균 0.007sec이므로 10만개 레코드 처리 시간은 714초로 추정되었으나 실제 처리 시간은 6.6sec로 아주 우수하였다.

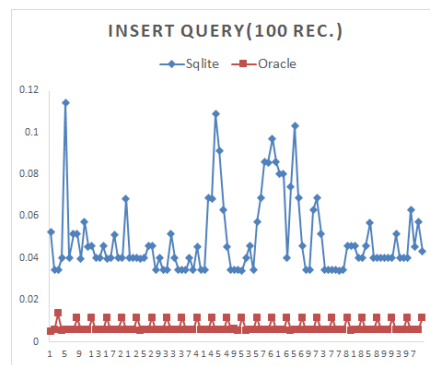


그림 1. Insert 쿼리 처리 성능

Update-Point 쿼리는 SQLite3와 Oracle 모두 비슷

한 성능을 보였다. Index 처리 오버헤드 때문에 Index가 없는 경우에 우수한 성능을 보였다.

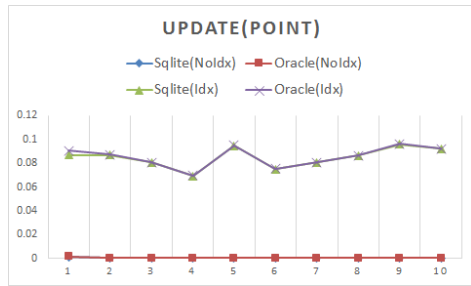


그림 2. Update-Point 쿼리 처리 성능

Update-Range 쿼리는 SQLite3가 연속된 대량 범위의 데이터를 수정할 때 Oracle보다 우수한 성능을 보였다. 실험 번호가 커질수록 수정 대상이 커지며 10번째 실험 결과는 full table을 수정하는 쿼리의 경우이다.

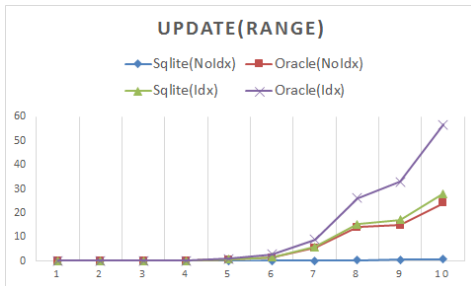


그림 3. Update-Range 쿼리 처리 성능

Delete-Point 쿼리도 SQLite3와 Oracle 모두 비슷한 성능을 보였다. Index 처리 오버헤드 때문에 Index가 없는 경우에 우수한 성능을 보였다.

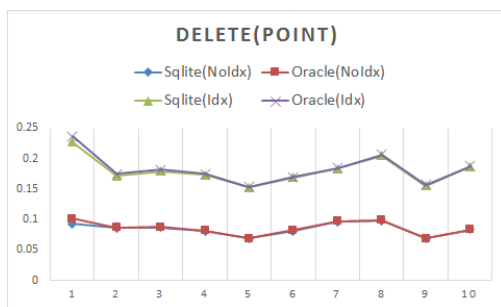


그림 4. Delete-Point 쿼리 처리 성능

Delete-Range 쿼리도 Update-Range와 같이 SQLite3가 Oracle보다 대량 데이터 처리에서 우수한 성능을 보였다. Index가 있는 경우는 모두 처리 성능이 떨어졌다.

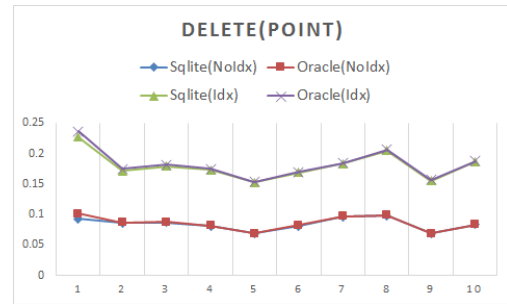


그림 5. Delete-Range 쿼리 처리 성능

V. 결론

이 논문에서는 SQLite3 데이터베이스의 갱신 쿼리 처리 성능을 실험하고 분석해 보았다. 실험 결과를 평가하기 위해 동일한 환경의 Oracle과 비교하여 살펴보았다.

실험결과 SQLite3는 Insert에서 Oracle과 큰 성능 차이를 보였지만 Update와 Delete에서는 대등하거나 보다 우수한 결과를 보였다.

SQLite3의 자체 갱신 쿼리 성능은 Insert 쿼리는 평균 0.049sec, Update-Point 쿼리는 0.0002sec, 그리고 Delete-Point 쿼리는 0.084sec로 측정되었다.

실험 결과를 바탕으로 향후 Android 환경에서 새로운 모바일 데이터베이스와의 성능 실험을 실시하고 분석 결과를 도출할 계획이다.

References

- [1] 최진오, SQLite 모바일 데이터베이스의 검색 성능 분석을 위한 구현 및 실험, 한국정보통신학회논문지, 2017.2.
- [2] S. Kim, J. Lee, M. Sin, S. Kim and Y. Kim, "Development of music & AV Server with Raspberry PI," in Proceedings of KSMTE, pp. 117-117, 2015.
- [3] M. Beck, M. Wemer and S. Feld, "Mobile Edge Computing: A Taxonomy," in Proceedings of 6th Conference on Advances in Future Internet, pp. 48-54, 2014.
- [4] A. Sripriya and R. Dhanapal, "Mobile software solutions using mobile database design methodology," in Proceedings of International Conference on ICCNT(IEEE), pp. 1-5, 2010.
- [5] Architecture of SQLite [Internet]. Available: <http://www.sqlite.org/arch.html>.
- [6] Realm Mobile Database [Internet]. Available: <http://realm.io/docs/#getting-started>.