

File System에 따른 SQLite3 모바일 데이터베이스의 갱신 성능 비교

최진오

부산외국어대학교 임베디드IT학부

Modification Performance Comparison of SQLite3 Mobile Databases for File Systems

Jin-oh Choi

School of Embedded IT, Busan University of Foreign Studies

E-mail : jochoi@bufs.ac.kr

요 약

모바일 환경에서 데이터베이스 엔진 사용이 보편화되고 있으며 모바일에 적합한 데이터베이스 엔진 개발과 발표가 뒤따르고 있다. 모바일 데이터베이스는 대부분 플래시 메모리로 구성된 다양한 파일시스템에서 관리된다. 이 논문에서는 대표적인 모바일 데이터베이스인 SQLite3의 파일 시스템별 갱신 성능을 테스트한다. 그래서 각 파일 시스템별 특성을 파악하고 비교하기 위한 시스템을 구현하고 성능 실험을 실시한다. 실험 결과로부터 각 데이터베이스 갱신 패턴에 따른 파일 시스템별 장단점을 분석할 수 있다.

키워드

File System, Modification Performance, SQLite3, Mobile Database

I. 서론

최근 모바일 IT 기기들은 계산 성능의 향상과 저장 용량의 증가로 인해 서버로 작동할 수 있을 정도로 그 지위가 변모되고 있다. 그래서 모바일 IT 기기가 서버 역할을 수행하거나 Edge Server로 많이 사용되는 추세에 있다.

이를 위해서는 모바일 IT 기기에 데이터베이스의 설치 및 운용이 필요하여 모바일 기기에 최적화된 모바일 데이터베이스의 필요성이 점점 커지고 있다. 그 결과로 최근까지 새로운 모바일 데이터베이스가 지속적으로 새로 개발되고 발표되고 있다.

그런데 모바일 IT 기기에서 모바일 데이터베이스는 대부분 플래시 저장장치에 설치된 다양한 파일시스템에서 작동한다. 그런데 파일 시스템의 종류에 따라 모바일 데이터베이스의 성능 차이가 발생할 수 있다.

이 논문에서는 모바일 데이터베이스의 파일시스템별 갱신 성능을 테스트하는 실험을 진행하고자

한다. 즉, 갱신 유형별 파일 시스템에 따른 성능을 측정하고 분석한다. 그 결과로서 각 파일시스템의 장단점과 특성을 파악할 수 있을 것으로 기대된다.

이 연구는 모바일 데이터베이스 중 가장 선두 주자로 꼽히는 SQLite3를 선정하여 진행한다. 연구 결과는 Realm, Machbase 등 다른 종류의 모바일 데이터베이스로 확장할 수 있다. 그리고 갱신 성능에 이어 다양한 검색에 대한 파일시스템들의 특성 파악에 활용될 수도 있을 것으로 기대된다.

II. 관련 연구

모바일 데이터베이스 제품군으로 SQLite[1]는 2000년에 발표되어 널리 사용되며 그 성능을 인증 받고 있다. Open Source SW이며 현재 V3.28 버전까지 지원되고 있다. Realm[2]의 경우 2014년 새로운 모바일 데이터베이스로 출시되어 다양한 모바일 기업과 모바일 앱에서 사용되고 있으며 SQLite의 경쟁자로 등장했다. Realm은 서

버, 에지, 그리고 메모리 데이터베이스로 활용이 가능하다. 또한 최근 국내 제품으로서 Edge 모바일 데이터베이스[3]가 Machbase에서 발표되었다. 이 모바일 데이터베이스는 IoT Gateway 장비에 사용되며 센서 등에 의해 생성되는 대용량의 데이터를 고속으로 저장하는 역할을 수행한다.

모바일 데이터베이스의 활용이 보편화되는 사례로 먼저 최근 라즈베리파이(Raspberry Pi)나 아두이노(Arduino)를 이용한 데이터베이스 서버 구축이 많이 시도되고 있다[4].

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) Point Insert
 - 2) Range Insert
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 한 레코드를 Insert하거나 키(Key) 값을 조건으로 한 레코드를 검색하여 Update하거나 Delete하는 쿼리를 말한다. ‘Range’ 쿼리는 복수 개 레코드 Insert 또는 일정 범위안의 순차 데이터를 검색하여 Update하거나 Delete하는 쿼리를 말한다. Range의 범위는 100개, 500개, 1,000개, 5,000개, 10,000개, 50,000개, 100,000개, 300,000개, 그리고 600,000개를 포함한다. 그리고 전체 테이블 갱신은 실험 데이터의 레코드 1,000,000개 전체를 갱신하는 쿼리를 말한다.

여기서 Insert 쿼리에서 Index가 없을 경우와 Index가 있을 경우를 구분하여 실험하는 것은 큰 성능 차이가 없어 생략하였다.

비교 파일시스템은 FAT32, NTFS, Ext2, Ext3로 한정한다.

IV. 실험 진행 계획

실험 환경은 다음과 같다.

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

V. 결론

이 논문에서는 SQLite3 데이터베이스의 갱신 쿼리 처리 성능을 파일시스템별로 실험하고 분석하기 위한 설계를 우선 진행하였다.

이 설계를 바탕으로 SQLite3를 대상으로 Insert, Update, 그리고 Delete 연산의 성능 특성을 파일시스템별로 측정하고 분석하고자 한다.

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

References

- [1] SQLite. Architecture of SQLite [Internet]. Available: <http://www.sqlite.org/arch.html>.
- [2] Realm. Realm Mobile Database [Internet]. Available: <http://realm.io/docs/#getting-started>.
- [3] MachBase Database. IoT Gateway for Edge Analytics [Internet]. Available: <http://machbase.com/product-iot-gateway>.
- [4] 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.