파일시스템 메타데이터 수정을 활용한 SQLite WAL 체크포인트 기법 연구

박대준°, 신동군
성균관대학교 전자전기컴퓨터공학과
pdaejun@skku.edu, dongkun@skku.deu

DB Remap: Remapping WAL for Checkpointing by Filesystem Metadata modification for SQLite databases

Daejun Park°, Dongkun Shin
Department of Electrical and Computer Engineering, Sungkyunkwan University

요 약

사용자에게 데이터베이스내의 데이터를 접근할 수 있도록 도와주는 소프트웨어인 데이터베이스 관리 시스템은 데이터베이스가 일관성을 가지도록 유지해야 한다. 데이터베이스는 비휘발성을 가지는 저장장치에 기록되는데, 저장장치에 데이터베이스를 기록하던 중 시스템에 장애가 일어나게 되면, 원자성이 훼손되어 해당 데이터베이스는 변경 전, 변경 후의 상태도 아닌 제 3의 상태가 되어 일관성에 문제가 생긴다. 원자성을 지키기 위해 사용되는 WAL 방식은 변경하고자 하는 데이터베이스의 내용만을 WAL 파일에 로그 형식으로 기록한 후 WAL 파일이 일정 크기가 되면, 체크포인트를 통해 데이터베이스 파일에 변경된 내용을 반영시키는 방법이다. 이때 체크포인트가 중복된 데이터에 대한 쓰기 명령을 발생시키기 때문에, 낸드 플래시 메모리와 같은 쓰기 횟수에 제한이 있는 저장장치에서는 수명에 영향을 끼칠 수 있다. 이러한 문제를 해결하기 위해 본 논문에서는 체크포인트 수행을 저장장치에 대한 쓰기 연산 없이 파일시스템의 메타데이터 수정만으로 수행하는 기법을 제안하고 있다. 제안한 기법을 통해 체크포인트 수행 시 쓰기 명령의 양이 효과적으로 줄어드는 것을 확인 할 수 있었다.

1. 서 론

데이터베이스 관리 시스템은 데이터베이스를 관리하여, 사용자의 요청에 따라 데이터를 처리하고 응답하는 방식으로 사용된다. 스마트 디바이스의 운영체제로 주로 사용되는 안드로이드에서는 SQLite를 기본 데이터베이스 관리 시스템으로 채택하여 안드로이드에서 사용되는 어플리케이션의 데이터와, 시스템 설정과 같은 정보들을 SQLite로 관리하고 있다. 스마트 디바이스에서 대표적으로 사용되는 저장장치인 eMMC는 낸드 플래시 메모리(NAND flash memory)를 기반으로 저 전력, 비휘발성, 충격 내구성이 큰 장점을 가지고 있다.

그러나 P/E(Program/Erase) cycle이 제한되어 있는 물리적인 특성도 가지고 있다. 한편 SQLite에서는 데이터베이스의 일관성을 유지하기 위해 저널 방식과 WAL[1](write-ahead log) 방식을 사용한다. WAL 방식은 데이터 변경 시에, 변경될 데이터를 WAL 파일에 먼저 기록 한 다음 체크포인트 수행 시에 변경될 데이터를 데이터베이스 파일에 기록하는 것이다. 시스템 장애 시에는 WAL 파일에 기록된 데이터를 데이터베이스 파일에 기록하는 방식으로 데이터베이스의 일관성을 보장 할 수 있다. 이때 WAL파일에 변경하고자 하는 데이터를 미리 기록하고, 체크포인트 수행 시에 이 데이터를 다시 한 번 데이터베이스에 기록되는 과정에서 같은 데이터에 대한 중복 기록은 쓰기 명령을 증가시키고, P/E cycle이 제한되는 낸드 플래시 메모리를 기반으로 하는 저장장치에서 수명을 줄이는 원인이 된다. 체크포인트 수행은 WAL 파일에 기록된 데이터를 데이터베이스 파일로 옮기는 작업이므로, 파일의 주소를 저장장치의 주소로 매핑해주는 역할을 하는 파일시스템에서 파일과 저장장치 사이의 매핑을 수정하게 되면, WAL 파일에 기록된 데이터를 실제 쓰기 없이 데이터베이스 파일로 옮길 수 있게 된다. 이를 통해 중복된 데이터에 대한 쓰기 명령이 줄어들게 되므로, 낸드 플래시 메모리를 기반으로 하는 저장장치의 수명 보장에 유리하다. 그러나 파일에 대한 매핑을 수정하기 위해 파일시스템의 메타데이터를 수정하던 중 시스템에 장애가 생기게 되면, 데이터베이스 뿐 만 아니라 파일시스템 전체의 일관성에 문제가 생기게 된다. 이를 방지하기 위해서 파일시스템에서 제공하는 저널링[2] 기법을 이용하면, 메타데이터 수정에 대한 원자성을 보장받을 수 있으므로, 데이터베이스와 파일시스템의 일관성을 유지할 수 있다. 본 논문에서는 파일시스템 메타데이터의 수정을 통해 WAL 방식을 사용하는 SQLite에서 체크포인트 시 유발되는 쓰기 명령을 줄이는 방법을 제안하고, 이때 생길 수 있는 데이터베이스의 일관성 문제를 저널링을 활용하여 해결하겠다.

이 논문은 2010년도 정부(미래창조과학부)의 재원으로 한국연구재단-차세대정보·컴퓨팅기술개발사업의 지원을 받아 수행된 연구임(No. 2010-0020724)

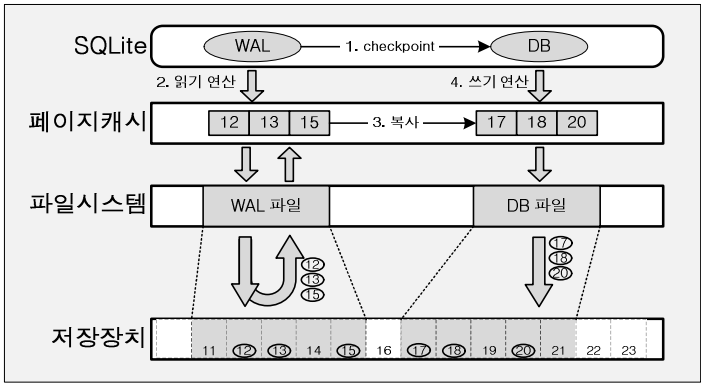

그림 1 WAL 체크포인트

2. 관련 연구

WAL로 인해 생기는 중복된 데이터에 대한 쓰기 명령을 피하기 위해 각 논문에서 여러 기법이 제안되었다.

Atomic write[3]는 FTL(flash translation layer)의 변경을 통해 여러 블록에 대해 원자적 쓰기를 지원하는 방식으로 추가적인 쓰기 명령을 줄이면서 데이터베이스에 대한 일관성을 유지한다. WAL은 저장장치가 데이터베이스 변경과 관련된 여러 블록에 대하여 원자적인 쓰기를 하드웨어적으로 제공하지 못하기 때문에, 데이터베이스 관리 시스템에서 사용하는 기법인데, atomic write는 하드웨어적으로 원자적인 쓰기를 제공하므로, WAL 파일에 대한 기록 없이 데이터베이스의 일관성을 제공한다. 그러나 atomic write를 사용하기 위해서는 해당 FTL이 적용된 저장장치가 필요하므로, 기존의 저장장치에는 사용 할 수 없다는 단점이 있다. 또한 atomic write를 사용 중일 때에는 다른 쓰기 명령을 동시에 수행할 수 없으므로 데이터베이스 관리 시스템은 자주 체크포인트를 수행하여, 체크포인트 시 처리해야하는 데이터의 양을 최소화시켜야 한다.

X-FTL[4]은 SQLite에서 체크포인트 시 유발되는 중복된 데이터에 대한 쓰기 명령을 피하기 위해, WAL파일을 기록하지 않고, 각 트랜잭션에 해당하는 데이터에 대한 FTL의 X-L2P(transactional logical to physical) 매핑테이블을 따로 유지하여, 체크포인트 시에는 해당 매핑테이블을 수정하는 방식으로 데이터베이스 파일에 변경된 데이터를 원자적으로 적용시키는 기법이다. X-FTL에서 사용하는 X-L2P 매핑테이블에 트랜잭션에 대한 정보를 함께 기록하므로, 이를 통해 atomic write에서 각각의 트랜잭션을 순차적으로 처리 할 수밖에 없는 문제점을 해결하였지만, 특수한 FTL을 사용하여야 하므로 기존의 저장장치에는 적용하기 힘들다는 단점이 있다.

본 논문에서는 WAL 파일을 기록하여 변경된 데이터를 체크포인트 시 파일시스템상의 매핑을 수정하여 데이터베이스 파일에 적용시키는 방식을 사용하고 있다. 위의 기법들과는 달리 FTL의 변경이 필요 없는 방법을 사용하기 때문에 기존의 저장장치에 쉽게 적용 가능하다는 장점이 있다.

3. DB remap

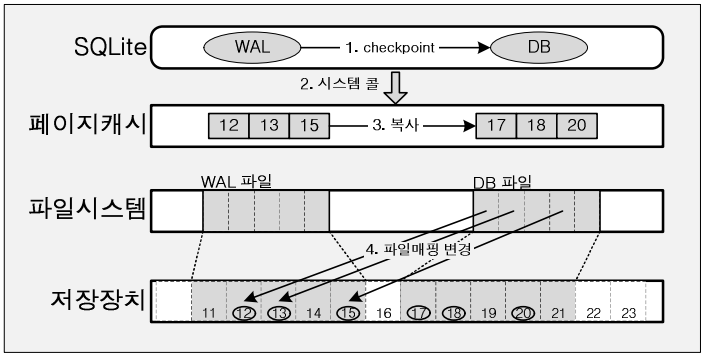

그림 2 DB remap의 동작 원리

3.1 기존 WAL 기법

데이터베이스 관리 시스템에 대한 사용자의 요청으로 인한 트랜잭션은, 데이터베이스에 여러 데이터를 변경하게 된다. 이러한 변경은 여러 블록으로 이루어져 있는데, 기존의 저장장치는 여러 블록에 대한 쓰기 명령을 원자적으로 수행 할 수 없다. WAL 기법은 이를 보완하기 위해 데이터베이스 관리 시스템에서 트랜잭션의 적용을 원자적으로 수행하여, 데이터베이스에 대한 일관성을 유지하기 위해 사용되는 기법이다. 그림 1은 WAL 기법에서 체크포인트에 대한 설명이다. WAL 파일에는 트랜잭션이 변경한 데이터들을 기록되어 있고, 체크포인트 동작 시에 WAL 파일에서 해당 데이터를 읽어서 데이터베이스 파일에 옮겨 적게 된다. WAL의 내용을 데이터베이스 파일에 적용하던 중 시스템에 장애가 생기게 되면, 복구 과정에서 해당 WAL의 내용을 다시 데이터베이스에 적용시키는 방식으로 데이터베이스의 일관성을 유지할 수 있고, WAL에 기록되지 않은 채 시스템에 장애가 생기게 되면 복구 과정에서 이를 무시하여 데이터베이스의 일관성을 유지할 수 있게 된다.

3.2 DB remap의 동작

기존 WAL 기법은 체크포인트 수행 시 WAL 파일의 데이터를 데이터베이스 파일로 복사한다. 제안하는 기법에서는 체크포인트 수행 시 생기는 쓰기 연산을 줄이기 위해 WAL 파일에 있는 데이터에 대한 파일시스템의 매핑을 수정하여 데이터베이스 파일의 데이터로 이동시킨다. 이를 통해 체크포인트에서 생성되는 데이터에 대한 많은 양의 쓰기 연산이 파일시스템의 메타데이터에 대한 쓰기 연산만으로 줄어들게 된다. 그림 2는 DB remap의 동작 원리에 관한 설명이다. 체크포인트 수행 시 SQLite에서 WAL파일 중 옮겨야 할 데이터에 대한 정보를 파일시스템에 시스템 콜을 통하여 전달한다. 이 후 파일시스템에서는 전달 받은 정보를 통해 WAL파일의 데이터에 대한 매핑을 데이터베이스 파일에 대한 매핑과 서로 바꾼다. 이때 생기는 파일시스템 메타데이터의 변경은 여러 블록으로 이루어지는데, 체크포인트 수행 중에 시스템에 장애가 생기게 되면 데이터베이스 파일과 WAL 파일 뿐만 아니라 파일시스템 전체에 일관성을 깨뜨리게

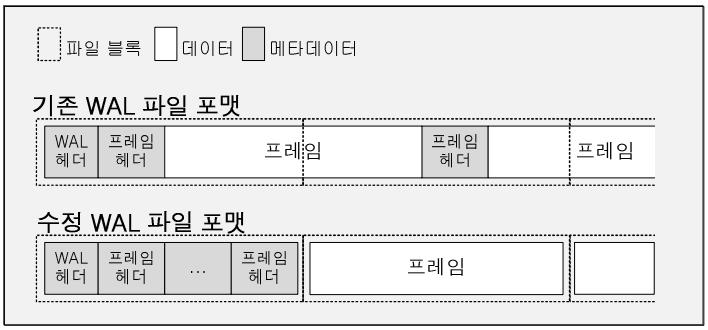

그림 3 WAL 포맷

된다. 그러므로 매핑의 변경으로 인해 생기는 파일시스템의 메타데이터들에 대해서 원자적 쓰기가 필요하다. 원자적 쓰기를 수행하기 위해 DB remap은 파일시스템에서 사용하는 저널링을 활용한다. 저널링은 WAL 기법과 같은 원리로, DB remap을 수행하는 시스템 콜에서 발생하는 모든 파일시스템의 메타데이터를 저널 영역에 미리 기록하는 방법을 통해 DB remap을 수행하는 시스템 콜에서 변경하는 모든 파일시스템의 메타데이터를 원자적으로 수정할 수 있다.

3.3 WAL 파일 포맷 수정

그림 3은 기존 WAL 파일 포맷과 DB remap을 위해 수정한 WAL 파일 포맷에 대한 그림이다. WAL 파일에는 데이터베이스로 복사되는 데이터인 프레임 외에 데이터를 관리하기 위해 기록되는 메타데이터도 함께 존재한다. 프레임의 크기는 파일 블록크기와 같지만, 메타데이터로 인해 WAL 파일 내부에 프레임이 파일 블록 단위로 정렬되어 있지 않다. 파일시스템 매핑 정보를 파일 블록 단위로 관리 하므로 기존 WAL 파일 포맷에서는 DB remap의 적용이 불가능하게 된다. 그래서 WAL 파일 포맷을 일부 수정하여 프레임에 대한 위치를 파일 블록에 대해 정렬되도록 하였다.

4. 실 험

DB remap 기법으로 인한 쓰기 연산의 감소를 측정하기 위해 안드로이드 젤리빈 버전이 적용된 갤럭시 S4를 사용하여 SQLite에 대한 벤치마크인 mobibench[5]를 수행하였다. ext4를 파일시스템으로 사용하여 벤치마크에서 트랜잭션의 수를 1000개의 단위로 총 5000개까지 수행하였다. 각 트랜잭션은 8KB 크기의 INSERT 요청으로 이루어져 있다. 기존의 WAL 기법을 비교 대상으로 하였다.

그림 4는 벤치마크 수행을 통해 제안한 기법과 기존의 WAL 기법을 비교한 결과이다. 제안한 기법에서는 WAL 파일에 기록한 데이터를 데이터베이스 파일에 다시 기록하는 쓰기 연산이 필요하지 않기 때문에, 기존 WAL 기법에 비해 일반데이터의 경우 약 40%정도의 적은 쓰기 연산량을 보이고 있다. 그러나 제안한 기법에서는 파일시스템의 매핑을 수정할 때 저널링 기법을 활용하기 때문에 해당 메타데이터들의 원자성을 보장해 주기 위해

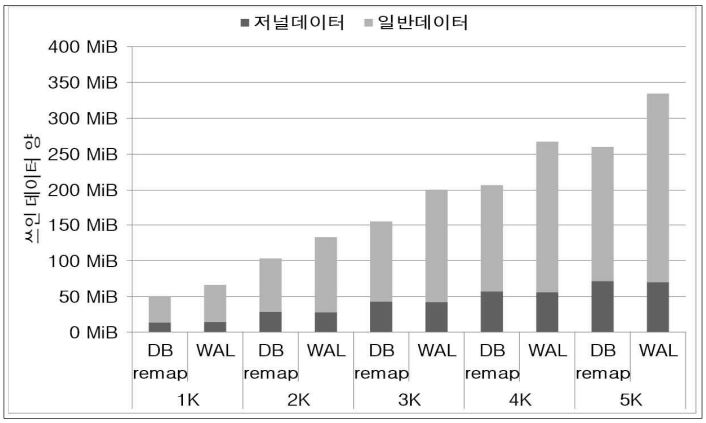

그림 4 벤치마크 실험결과

저널 영역에 추가적인 쓰기 연산이 유발된다. 하지만 수정되는 메타데이터가 적기 때문에, 이는 매우 적은 수준으로, 제안된 기법에 의한 저널 데이터에 대한 쓰기 증가량은 약 1% 내외로 측정되었다. 스마트 디바이스에서 주로 쓰이는 운영체제인 안드로이드에서 SQLite가 기본 데이터베이스 관리 시스템인 점과 낸드 플래시 메모리를 기반으로 한 eMMC를 주로 사용한다는 점을 살펴볼 때, 제안하는 기법을 통해 스마트 디바이스에서 SQLite로 인해 생기는 쓰기 연산이 줄어들어, eMMC의 수명향상을 기대할 수 있다.

5. 결 론

기존 저장장치와는 달리 낸드 플래시 메모리 기반의 저장장치에서는 제한된 P/E cycle로 인해, 잦은 쓰기 명령은 수명을 감소시키는 원인이 된다. 본 논문에서는 WAL기법을 사용하는 SQLite에서 데이터에 대한 중복기록을 개선하기 위해, 파일시스템의 파일 매핑을 변경하여, 실제로 데이터를 기록하지 않고도 체크포인트를 수행 할 수 있게 하였다. 이를 통해 낸드 플래시 기반의 저장장치에서 중복된 쓰기 명령으로 인한 수명 감소를 줄이면서, 체크포인트를 수행 할 수 있었다. 향후 연구로는 제안한 기법을 통해 쓰기 연산 양 뿐만 아니라 성능도 향상시키는 방법에 대해 연구할 계획이다.

참 고 문 헌

[1] http://www.sqlite.org/wal.html
[2] Tweedie, Stephen C. “Journaling the Linux ext2fs filesystem.” The Fourth Annual Linux Expo. 1998.
[3] Ouyang, Xiangyong, et al. “Beyond block I/O: Rethinking traditional storage primitives.” High Performance Computer Architecture (HPCA), 2011 IEEE 17th International Symposium on. IEEE, 2011.
[4] Kang, Woon-Hak, et al. "X-FTL: transactional FTL for SQLite databases." Proceedings of the 2013 international conference on Management of data. ACM, 2013.
[5] Jeong, Sooman, et al. "Androstep: Android storage performance analysis tool." ME13: In Proc. of the First European Workshop on Mobile Engineering, Aachen, Germany. 2013.