

플래시 메모리 기기를 위한 다중 버전 잠금 기법

Multi-version Locking Scheme for Flash Memory Devices

변 시 우*
Byun Si Woo

Abstract - Flash memories are one of best media to support portable computer's storages. However, we need to improve traditional data management scheme due to the relatively slow characteristics of flash operation as compared to RAM memory. In order to achieve this goal, we devise a new scheme called *Flash Two Phase Locking (F2PL)* scheme for efficient data processing. F2PL improves transaction performance by allowing multi version reads and efficiently handling slow flash write/erase operation in lock management process.

Key Words : Portable Device, Flash Memory, Locking, Multi-version, Lock Mode,

1. 서 론

각종 소형 정보기기들이 대중화됨에 따라, 정보 저장용 미디어로 플래시 메모리가 보편적으로 활용되게 되었다. 따라서 플래시 미디어에 저장된 데이터를 체계적으로 관리하는 임베디드 소프트웨어가 필요하게 되었다. Gartner社의 자료에 따르면, 정보기기용 임베디드 소프트웨어는 최근 100%이상으로 급성장하고 있다고 한다.[1] 특히, 세계적인 분석 기관인 IDC(International Data Corporation)社는 64%이상의 휴대용 정보기기의 어플리케이션이 임베디드 데이터베이스 시스템이 필요하다고 보고했다.[2] 따라서 소형 정보기기의 플래시 메모리 기반 데이터베이스 시스템은 핵심 임베디드 소프트웨어 분야로서 상당히 유망하며, 이와 관련된 데이터 관리 기술의 개발이 매우 시급하다고 할 수 있다.

본 논문은 휴대용 소형 정보기기의 데이터 저장 장치로 많이 사용되는 플래시 메모리를 기반으로 하는 플래시 메모리 데이터베이스 시스템에서의 새로운 트랜잭션 처리 기법을 제안한다.

2. 관련 연구

현재 플래시 미디어 자체에 대한 연구나 플래시 파일 시스템에 대한 연구는 성숙단계에 진입하였다. 또한, 기존의 디스크-기반의 상용DR(Disk Resident) 데이터 처리기나 메인 메모리-기반 MM(Main Memory) 데이터 처리기는 오래 전부터 인기 있는 연구 분야로 활발한 연구가 지속되고 있다. OBE[3], MM-DBMS[4], MARS[5], HALO[6], System M[7]

등의 시스템이 과거의 메모리 기반 데이터 처리기의 연구사례이다. 하지만 플래시 메모리를 활용하는 데이터베이스 시스템에 대한 연구는 아직 시작 단계에 불과한 실정이다.

IBM의 Office-By-Example(OBE) 데이터베이스 프로젝트와 연계하여 메인 메모리 데이터베이스가 제작되었는데, 주로 트랜잭션의 ad hoc 쿼리를 처리하는 것에 집중되어 있다. 데이터 표현은 상당히 많은 포인터를 사용하였으며, 릴레이션은 튜플들의 연결 리스트로 저장되었다.

MM-DBMS는 데이터 표현과 트랜잭션 관련 접근 처리를 위하여 OBE와 마찬가지로 많은 포인터를 사용하였으며, 인덱스 구조는 인덱스된 튜플을 직접 포인터 한다. 각 블록은 복구처리에 의하여 checkpoint 처리 되었으며, 트랜잭션의 동시성 제어를 위하여 two-phase 기법을 사용하였다. 또한 전체 릴레이션을 로킹하는 큰 단위의 로킹 기법을 사용하였다.

MARS는 메모리 상주의 휘발성 데이터에 대하여 고속의 트랜잭션 처리를 위하여 데이터베이스 처리기와 복구 처리기를 사용하였다. 동시성 제어는 two-phase 로킹과 전체 릴레이션 단위의 대단위 로킹 기법을 사용하였다.

MM-DBMS와 MARS가 트랜잭션 관련 로킹과 체크포인트를 위하여 전용처리기를 사용한 반면, HALO(Hardware Logging)는 트랜잭션 로킹을 위하여 특수 장치를 사용하여서 트랜잭션 처리기의 로킹 부하를 경감하였다. HALO는 처리기와 메모리 제어기 사이의 통신에 간섭하여 모든 메모리 수정에 대하여 워드 단위의 로그를 생성한다.

System M은 Mach운영체제에서 쓰레드 기반의 서버상에서 구현되었다. 메시지 서버가 트랜잭션 요청을 받고 수행결과를 클라이언트에게 돌려주며, 트랜잭션 서버가 요청된 트랜잭션을 수행한다. 트랜잭션 관련 동시성 제어를 위하여 two-phase 로킹을 사용하며, 사전 커미트와 그룹 커미트를 사용한다.

(주)코스모에서 개발한 메모리 상주형 DBMS인 RtPlus[8]는 한국전자통신연구원의 DREAM-S의 후속 모델로, 범용

저자 소개

* 正 會 員 : 安養大學校 디지털미디어學科 助敎授 · 工博

* 본 연구는 한국학술진흥재단의 지역대학 연구지원으로 수행되었음(R05-2004-000-10961-0).

멀티미디어 처리를 위하여 객체관계형 데이터 모델을 기반으로 개발된 초소형 실시간 DBMS이다. RtPlus의 플랫폼으로는 Solaris, Linux 뿐 아니라 VxWorks 등 상용 실시간 운영체제도 지원하고 있다. RtPlus는 독립적으로 소형 DBMS로 사용될 수 있으며, 사용자 및 관리자 지원도구도 갖추고 있다.

플래시 메모리를 저장 매체로 사용한 연구에는 eNVy 시스템이 있다.[9] 이 시스템은 플래시 메모리를 일반 중소형 컴퓨터 및 하드 디스크의 대용으로 사용하였다. 플래시 메모리를 다량으로 연결하여 2GB의 저장 영역을 구성하였다.

그러나 순수하게 플래시 미디어 기반의 데이터 처리기의 연구는 시도된 적이 거의 없는 미개척 분야이다. 이유는 그동안 플래시 메모리는 주 메모리의 성능과 비교하면 느리면서도 고가이며, 일반적인 시스템에는 사용되지 않는 특수 부품이며, 디스크에 비하여 저장 비용이 고가였기 때문이다.

그러나 최근 플래시 메모리 기술의 발전으로 대부분의 단점들이 해소된 상태이며, 저렴하고 성능 좋은 미래의 저장 매체로서 크게 각광 받고 있다. 특히, 대부분의 임베디드 시스템에서는 공간 제약, 전력 소모, 중량 및 내충격성 문제로 디스크는 사용할 수 없기 때문에, 비휘발성(Non-Volatile) 저장 장치로서 플래시 메모리를 반드시 사용하여야만 한다. 다만, 일반 메인 메모리와는 달리, 쓰기와 지우기 연산에 10배 정도의 많은 시간이 소요되며, 쓰기 회수가 100,000번 정도로 제한된다는 특성을 감안하여, 효과적인 휴대형 정보 기기의 데이터 처리기를 개발하여야 한다.

3. 플래시 메모리의 특성을 고려한 처리 기법 제안

플래시 메모리는 일반 메인 메모리와는 달리, 쓰기와 지우기 연산에 10배 이상의 많은 시간이 소요되며, 쓰기 연산에 앞서 세그먼트 단위의 저속도의 지우기 연산이 먼저 수행되어야 하는 부품 특성상의 약점이 있다. 이러한 고유한 약점을 고려하여 기존의 트랜잭션 처리 기법을 개선하지 않으면, 플래시 메모리를 주 저장장치로 사용하는 소형 정보 기기에서 충분한 성능을 기대하기는 어렵다. 다음은 본 연구에서 성능 개선을 위하여 제안하는 플래시 기반의 트랜잭션 처리 기법의 기본적인 설계 목표이다.

- ① 트랜잭션의 직렬가능성을 유지한다.
- ② 트랜잭션의 처리 성능을 높인다.
- ③ 트랜잭션의 철회율을 낮춘다.

일반적으로 트랜잭션 처리 관련된 동시성 제어 기법에서 데이터 일관성 유지를 위하여 직렬가능성을 보장하게 된다. 보통 트랜잭션이 임출력을 위하여 기다리는 동안 중앙처리장치는 대기상태로 되므로 이용률이 저하된다. 이때 일관성을 유지하면서 다른 트랜잭션을 동시에 수행하게 되면 처리 성능은 자연히 높아진다. 즉, 트랜잭션의 직렬화가 가능하게 되면 정확성과 동시 실행성의 장점을 모두 얻을 수 있다.

직렬 가능성을 보장하는 보편적인 방법이 2단계 로킹(two-phase locking:2PL)[10]이며, 동시에 실행되는 트랜잭션의 상호 간섭을 방지하기 위해 데이터 항목에 록(lock)을 걸게 된다. 실질적으로, 2단계 로킹의 변형 기법들 중에서 가장 많이 사용되는 것이 엄격한 2단계 로킹(strict two-phase

locking: S2PL)기법[11]이다. S2PL은 트랜잭션 T가 완료되거나 철회될 때까지 T가 보유한 록들 중 어떠한 록도 해제하지 않는다. 따라서 트랜잭션 T가 완료되지 않았으면 어떠한 다른 트랜잭션도 T가 쓴 항목을 읽거나 쓸 수 없다. 이 S2PL 기법은 플래시 메모리 데이터베이스 환경에 적용할 경우 설계 목표 ①의 직렬가능성 조건을 만족한다. 그러나 설계 목표 ②와 ③을 만족하기 위해서는 플래시 메모리 데이터베이스 환경을 고려하여 더 개선할 필요가 있다.

먼저, 설계 목표 ②와 설계 목표 ③과 관련된 이슈에 대하여 살펴보자. 읽기 트랜잭션의 경우에는 플래시 메모리의 연산 속도가 일반 RAM 메모리에 비하여 크게 뒤지지 않으므로 별 문제가 없다. 하지만, 플래시 메모리의 쓰기 연산의 경우에는 속도가 상대적으로 매우 느리며, 더욱이 쓰기 연산 전에 상당히 느린 소거 연산을 반드시 수행해야 하므로 연산의 부담이 상당히 큰 것이 문제이다.

이 문제를 해결하고자, 본 논문에서는 기존의 트랜잭션 처리 방식을 개선한 플래시 2 단계 로킹 기법(Flash two-phase locking: F2PL)을 제안한다. 즉, 트랜잭션 관련 데이터가 플래시 메모리에 위치한 경우에 플래시 메모리의 접근 특성을 고려하여 효율적인 트랜잭션 관리를 제공하는 기법이다. 이 기법을 통하여 설계 목표 ②와 설계 목표 ③을 만족시킬 수 있다.

플래시 메모리에 쓰기 트랜잭션을 수행시킬 경우, 실제 특정 번지에 쓰기 연산을 수행하기 전에 2ms의 매우 느린 속도로 16KB 크기를 가진 세그먼트 단위의 소거 연산이 먼저 수행되어야 한다. 또한, 소거 연산 수행후의 쓰기 연산도 한 바이트 단위가 아닌 512 바이트의 블록 단위로 수행되어야 하는 제약이 있다. 따라서 쓰기 연산의 비중이 높을수록 빈번한 트랜잭션 충돌이 심각하게 발생하여 트랜잭션 처리 성능을 크게 저하시키게 된다. 이 때, 접근하고자 하는 세그먼트 블록을 미리 점유하지 않고 바로 쓰기 록을 진행하면 높은 충돌 특성에 의하여 다수의 트랜잭션이 세그먼트 점유 과정에서 중도에 철회되게 된다. F2PL 기법은 쓰기 연산의 특을 쓰기-의도 록과 쓰기 록으로 분리하여 쓰기 연산 진행시 미리 쓰기-의도 록을 획득하여 진행하게 제어하였다. F2PL 기법은 이 두 가지의 록을 효율적으로 관리하여서 가장 심각한 쓰기 트랜잭션 관련 철회율을 개선하게 된다.

또한, 쓰기 트랜잭션은 장시간 록을 점유하기 때문에 이 점유 시간에 발생하는 다수의 읽기 트랜잭션은 모두 대기하거나 철회되어야 한다. 이로 인한 트랜잭션 처리 성능의 저하도 심각하므로 F2PL 기법은 읽기 트랜잭션에 한하여 다중 버전을 허용하였다. F2PL 기법은 기존의 다중 버전 로킹 기법[11,12]을 기반으로 확장하여 같이 읽기 록(Read Lock), 쓰기-의도 록(Write-Intention Lock), 인증 록(Certify Lock)의 4가지 록 모드가 있다. 표1은 4가지 록 모드와 상호간의 호환성을 나타내고 있다. 일반적인 록 관리 기법에서는 한 트랜잭션이 특정 오브젝트에 대하여 쓰기 록을 점유하게 되면, 추후에 발생하는 읽기 및 쓰기 트랜잭션은 이 특정 오브젝트에 대한 연산을 수행할 수 없다. F2PL의 다중버전 허용 2단계 록 관리는 어떤 트랜잭션이 특정 오브젝트에 대하여 쓰기 록을 걸고 있을 때, 이 오브젝트에 접근하고자 하는 다른 읽기 트랜잭션을 허용하는 것이다. 이 F2PL 기법은 한 오브젝트에 대하여 두 가지 버전을 관리하여 제어한다. 한 버전 오

브젝트는 완료된 트랜잭션이 쓰기를 수행한 것이고, 다른 하나의 버전 오브젝트는 어떤 트랜잭션이 그 오브젝트에 대해 쓰기 록을 점유할 때 만들어 진다. 따라서 어떤 트랜잭션이 쓰기 록을 점유하고 있어도 추후 발생된 읽기 트랜잭션은 완료된 버전의 오브젝트를 계속 읽을 수 있다. 쓰기 트랜잭션은 완료된 버전의 값에 영향을 주지 않고 원하는 대로 오브젝트의 값을 갱신할 수 있다. 그러나 일단 트랜잭션이 완료할 준비가 되면 완료하기 전에 현재 쓰기 록을 보유하고 있는 모든 항목들에 대하여 인증 록을 확보해야 한다. 인증 록은 읽기 록과 호환성이 없으므로 자신이 쓰기 록을 가지고 있는 오브젝트들을 읽고 있는 다른 트랜잭션이 없을 때까지 트랜잭션 완료가 연기된다.

(표1) F2PL의 록 모드 및 호환성 테이블

록 모드	읽기(R)	쓰기-의도(WI)	쓰기(W)	인증(C)
읽기(R)	Y	Y	Y	N
쓰기-의도(WI)	Y	N	N	N
쓰기(W)	Y	N	N	N
인증(C)	N	N	N	N

4. 결론 및 향후 과제

최근 소형 정보기기의 데이터 저장 장치로 많이 사용되는 플래시 메모리와 관련된 기존의 데이터 처리 기술을 분석하고, 트랜잭션 처리 성능과 응답 성능을 개선할 수 있는 F2PL 기법을 제안하였다. F2PL 기법은 쓰기 연산과 소거 연산이 상대적으로 매우 느린 플래시 메모리 연산의 단점을 고려하여 록 관리 기법에 반영하였으며, 다중버전 읽기를 허용하여 전반적인 트랜잭션 처리율을 낮추었다. 이로부터 트랜잭션 응답 성능과 처리 성능을 높일 수 있다.

참 고 문 헌

[1] 이정배, 이두원, "임베디드 시스템 동향," 정보처리 제9권 제1호, pp. 13-27, 2002.
 [2] 유재정, "Mobile Database란?", "http://www.mobilejava.co.kr/bbs/temp/lecture/j2me/mdb1.html, 2003.
 [3] Ammann A. C., Hanrahan M. B., and Krishnamurthy R., "Design of a memory resident DBMS", in Proc. IEEE COMPCOM Conf., 1985
 [4] Lehman T. J. and Carey M. J., "Query processing in main memory database management systems", in Proc. ACM SIGMOD Conf., Washington, DC, May, 1986.
 [5] Eich M. H., "A classification and comparison of main memory database recovery techniques", in Proc. Int. Con. On Data Engineering, pp. 332-3339, Feb. 1987.
 [6] Garcia-Molina H. and Salem K., "High performance transaction processing with memory resident data", in Proc. Int. Workshop on High Performance Transaction

Systems, Paris, Dec. 1987.
 [7] Garcia-Molina H. and Salem K., "Main Memory Database Systems: An Overview" IEEE Trans. Knowl. Data Eng., vol.4, no. 6, pp. 509-516, Dec. 1992.
 [8] 조주현, "임베디드 실시간 시스템의 개발 환경", 정보처리 논문지, 제9권 제1호, 2002
 [9] 민용기, 박승규, "이동컴퓨터를 위한 플래시 메모리 클리닝 정책", 한국통신학회논문지, vol. 24 no. 5A, pp. 657-666, 1999. 5.
 [10] Tamer Ozsu M. and Patrick Valduriez, "Principles of Distributed Database Systems", Prentice Hall, 1991.
 [11] Bernstein P., Hadzilacos V., Goodman N. "Concurrency control and recovery in database systems", Addison-Wesley, 1987.
 [12] 황규영, 홍의경, 음두현, 박영철, 김진호, "데이터베이스 시스템", 생능출판사, 2000