

IPL 기반의 Berkeley DBMS

김강년*, 나갑주*, 이상원*

*성균관대학교 정보통신공학부

e-mail : {kangnuni.factory, swlee}@skku.edu

IPL based Berkeley DBMS

Kang-Nyeon Kim*, Gap-Joo Na*, Sang-Won Lee*

* Electrical and Computer Engineering, Sungkyunkwan University

요 약

최근 낸드 플래시 메모리가 차세대 저장장치로 부상하면서 수십 년간 DBMS 의 저장장치였던 하드디스크의 대안으로 주목 받고 있다. 낸드 플래시 메모리는 하드 디스크와 인터페이스가 다르기 때문에 일반적으로 플래시 변환 계층을 사용하여 기존 소프트웨어와 호환성을 유지한다. 하지만 플래시 변환 계층은 소량의 랜덤 쓰기가 빈번한 DBMS 환경에서 비효율적인 방식이다. 이러한 문제점을 극복하기 위해 DBMS 의 특성을 고려한 In-Page Logging(IPL) 기법이 제안되었다. IPL 기법은 우수한 성능과 복구의 용이성 외에도 DBMS 구조를 크게 변경하지 않고 구현이 가능한 것이 장점이다. 본 논문의 목적은 IPL 기법을 활용하여 상용 DBMS 에서 최소한의 변화만으로 낸드 플래시 메모리를 저장 장치로 사용 할 수 있음을 증명하는 것이다. 이를 위해 Berkeley DBMS 에 IPL 기법을 구현 하며 성능 평가를 통해 IPL 기법이 상용 DBMS 에서도 우수한 성능을 보이는 것을 확인한다.

1. 서론

낸드 플래시 메모리는 충격에 강하고 소형화가 용이하며 전력 소모가 적은 장점을 갖는 비 휘발성 메모리이다. 이러한 장점을 바탕으로 휴대폰과 PDA 같은 휴대용 기기에서 이미 하드 디스크를 대체하고 저장 장치로 널리 사용되고 있다. 디스크와 달리 덮어 쓰기가 불가능한 특성으로 인해 기존의 소프트웨어에서 낸드 플래시 메모리를 사용 하는 것이 어렵기 때문에 플래시 변환 계층(FTL)[6][7]을 주로 사용한다. 하지만 OLTP(On-Line Transaction Processing)와 같이 넓은 범위의 소량 갱신이 많은 환경에서는 버퍼 교체가 잦고 데이터 페이지 단위의 쓰기가 빈번하다. 이런 환경에서 FTL 을 사용하면 비용이 큰 삭제 연산이 자주 발생하는 문제가 있다.

이러한 문제점을 해결하고자 제안된 IPL 기법[2]은 DBMS 에서 데이터 페이지의 갱신을 로그로 기록하고 버퍼 교체 시점에 소량의 로그 데이터만 쓰기 때문에 삭제 연산이 빈번해지는 문제를 피할 수 있다. 또한 기존 DBMS 의 구조를 크게 바꾸지 않고 낸드 플래시 메모리를 저장 장치로 활용할 수 있기 때문에 구현이 용이하다. 낸드 플래시 메모리 기반 DBMS 에 관한 기존 연구로 IPL 기법을 인덱스에 적용한 IPL B⁺트리 [3]와 LGeDBMS[4]가 있지만 상용 DBMS 에서 낸드 플래시 메모리를 저장 장치로 활용하는 연구는 활발히 진행되고 있지 않다.

이번 연구에서는 비교적 간단하게 상용 DBMS 에서 낸드 플래시 메모리를 사용하는 것이 가능함을 보이기 위해 IPL 기법을 Berkeley DBMS 의 Data Store 모델에 구현한다. 또한 IPL 이 상용 DBMS 상에서도 우수한 성능을 내는지 확인하기 위한 성능 평가를 수행한다.

2. IPL Berkeley DBMS

Berkeley DBMS 는 Oracle 사에서 관리하는 오픈 소스 임베디드 DBMS 다. 일반적인 데이터베이스 서버와 달리 어플리케이션에 라이브러리 형태로 내장되어 API 를 제공한다. 또한 어플리케이션의 요구사항에 따라 Data Store(DS), Concurrent Data Store, Transactional Data Store, High Availability 의 네 가지 모드로 사용 가능한 유연한 구조를 갖는다. [5]

본 논문에서는 가장 기본적인 형태인 DS 모델에 IPL 을 구현한다. DS 모델은 크게 Access Method(B⁺트리) 계층, 버퍼 관리자 계층, 디스크 관리자 계층, OS 추상화 계층으로 구성된다.

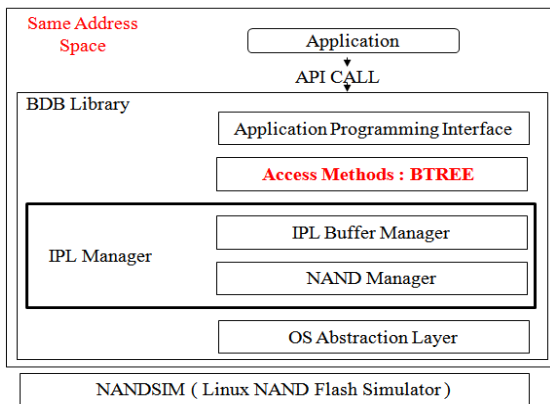
<그림 1>은 IPL Berkeley DBMS 의 아키텍처를 나타낸다. 기존의 버퍼 관리자와 디스크 관리자를 각각 IPL 기법을 적용한 IPL 버퍼 관리자와 낸드 플래시 메모리의 특성을 고려한 낸드 관리자로 변경한다.

B⁺ 트리 계층에서 IPL 기법 적용을 위해 추가적으로 필요한 작업은 데이터 페이지의 변경을 유발하는 모든 연산들을 로그로 기록하는 것이다. 삽입과 삭제 연산뿐 만 아니라 분할 연산, 데이터 페이지 초기화 연산도 데이터 페이지의 구조를 물리적으로 변경한다. 이러한 변경 내용은 512 바이트의 메모리 로그 섹터

본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT 연구센터 지원사업(NIPA-2009-(C1090-0902-0046))과 2008 년 정부(교육과학기술부)의 재원으로 한국학술진흥재단(KRF-2008-0641)의 지원을 받아 수행하였음.

에 로그로 기록하는 작업을 추가한다.

다음으로 IPL 버퍼 관리자 계층에서는 IPL 의 읽기, 쓰기, 병합 연산을 구현한다. 쓰기 연산 수행 시 기존 DBMS 처럼 데이터페이지를 쓰지 않고 메모리 로그 섹터만 쓴다. 메모리 로그 섹터는 데이터 페이지가 속한 블록의 로그 영역에 순차적으로 쓴다. 쓰기 연산 발생 시점은 메모리 로그 섹터가 꽉 찼을 때나 커밋 시점 혹은 버퍼 교체 정책에 의한 데이터 페이지 방출 시점이다. 쓰기 연산 과정에서 로그 영역에 빈공간이 없으면 병합 연산을 통해 최신 데이터 페이지를 생성하고 깨끗하게 지워진 새 블록의 데이터영역에 쓴다. 읽기 연산은 기존의 DBMS 처럼 원본 데이터 페이지를 읽는 과정은 동일하지만 추가적으로 로그영역에 저장된 로그를 모두 읽어 데이터 페이지에 반영하는 작업을 수행한다.



(그림 1) IPL Berkeley DBMS 아키텍처

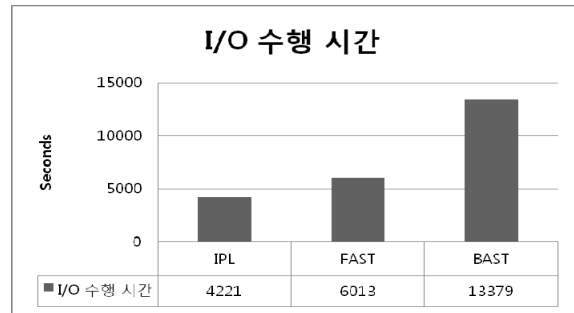
낸드 관리자에서는 기존의 디스크 공간 관리에 블록 매핑 기법을 적용하고 블록 내의 로그 영역을 관리한다. 마지막으로 OS 추상화 계층에는 삭제 연산을 위한 인터페이스를 추가한다.

3. 성능 평가

IPL Berkeley DBMS 의 성능 평가를 수행하여 IPL 기법이 실제 상용 DBMS 에서도 우수한 성능을 보이는지 검증한다. 워크로드를 생성하기 위해 어플리케이션에서 100 만 건의 키와 데이터를 삽입 한 후 검색을 요청한다. 어플리케이션의 요청을 처리하기 위해 IPL Berkeley DBMS 를 NANDSIM 상에서 구동한다. 또한 성능 비교를 위해 디스크 기반의 Berkeley DBMS 를 BAST[6]와 FAST[7] FTL 시뮬레이터상에서 각각 구동하고 I/O 수행 시간을 계산 하여 비교 한다. 각 FTL 의 로그 영역의 크기는 데이터 영역의 크기 대비 30%이며 플래시 메모리의 읽기, 쓰기, 삭제 연산의 지연 시간을 각각 131us, 306us, 2ms 로 가정한다.[1]

<그림 2>는 IPL Berkeley DBMS 와 BAST, FAST 의 I/O 수행 시간을 비교한 결과를 나타낸다. 실험 결과 IPL Berkeley DBMS 는 BAST 방식 대비 약 3 배, FAST 방식 대비 약 1.4 배의 성능 향상을 보인다. 30%의 상당한 로그영역을 할당한 FTL 방식보다 IPL 기법이 우수한 성능을 보이는 이유는 랜덤 워크로드가 유발하는 잦은 버퍼 교체시점에 IPL 버퍼 관리자가 데이터

페이지 전체를 쓰는 대신 소량의 로그만을 순차 기록 하기 때문이다.



(그림 2) 100 만건 랜덤 삽입/검색 처리 결과

4. 결론

본 연구에서는 상용 DBMS 구조를 크게 바꾸지 않고 낸드 플래시 메모리를 저장 장치로 사용할 수 있음을 보이기 위해 IPL 기법을 Berkeley DBMS 에 적용했다. 성능 평가 결과 IPL 방식을 사용한 Berkeley DBMS 는 FTL 방식 보다 최대 3 배 이상의 성능 향상을 보여 IPL 기법이 실제 DBMS 에서도 좋은 성능을 보였고 결과적으로 낸드 플래시 메모리 기반 DBMS 에 적합한 기법임을 증명했다. 이번 연구에서는 Berkeley DBMS 의 Data Store 모델에 한해 IPL 기법을 적용했지만 향후 연구로 DBMS 의 핵심 기능인 동시성 제어, 트랜잭션 관리, 복구 기능을 지원하는 Transactional DS 모델에도 IPL 을 적용할 계획이다.

참고문헌

- [1] A. Inoue and D.Wong. Nand flash applications design guide. Technical Report Revision 2.0, Toshiba America Electronic Components, Inc., 2004.
- [2] Sang-Won Lee and Bongki Moon. Design of Flash-Based DBMS: An In-Page Logging Approach. In Proceedings of the ACM SIGMOD, pages 55–66, Beijing, China, June 2007.
- [3] Gap-Joo Na, Bongki Moon, and Sang-Won Lee. In-page logging b-tree for flash memory. In DASFAA, Lecture Notes in Computer Science, pages 55–66. Springer, April 2009.
- [4] Gye-Jeong Kim, Seung-Cheon Baek, Hyun-Sook Lee, Han-Deok Lee, and Moon Jeung Joe. LGeDBMS: A Small DBMS for Embedded System with Flash Memory. In Proceedings of the 32nd International Conference on Very Large Data Bases, Seoul, Korea, September 12-15, 2006, pages 1255–1258. ACM, 2006.
- [5] Himanshu Yadava. The Berkeley DB Book. Apress
- [6] KIM, J. S., KIM, J. M., NOH, S. H., MIN, S. L., AND CHO, Y. K. 2002. A space-efficient flash translation layer for compactflash systems. IEEE Transactions on Consumer Electronics 48, 366-375.
- [7] S. Lee, D. Park, T. Chung, D. Lee, S. Park, and H. Song. A. Log Buffer based Flash Translation Layer Using Fully Associative Sector Translation. IEEE Transactions on Embedded Computing Systems, 6(3):18, 2007. ISSN 1539-9087.