

스토리지 시스템의 계층 통합을 통한 플래시메모리 기반 DBMS의 성능 향상에 대한 연구

심효기[○]* 윤경훈* 박성민* 정호영* 강수용** 차재혁***

한양대학교 전기컴퓨터통신학과*

한양대학교 컴퓨터교육과 교수**

한양대학교 정보통신공학부 교수***

dahlia[○]@hanyang.ac.kr, rumimaru@hanyang.ac.kr, syrilo@hanyang.ac.kr, horong@hanyang.ac.kr,

sykang@hanyang.ac.kr, chajh@hanyang.ac.kr

A Study of Performance Improvements of Flash-Based DBMS through Merging the Layers of the Storage System

Hyokey Sim[○]*, Kyoungnon Yoon*, Sungmin Park*, Ho-young Jung*,

Sooyong Kang**, Jae-hyuk Cha***

Dept. of Electronics and Computer Engineering, Hanyang Univ.

Dept. of Computer Science Education, Hanyang Univ.

Dept. of Informations and Communications, Hanyang Univ.

휴대용 기기나 디지털 미디어 기기와 같은 소형 컴퓨터는 저장 매체로 NAND 타입의 플래시 메모리를 사용한다. 플래시 메모리는 비휘발성 기억장치로서 하드디스크에 비해 I/O 연산의 속도가 빠르고 외부 충격에 강하며 전력소모가 적고 가볍다는 장점이 있다. 하지만 읽기/쓰기 연산의 기본 단위인 페이지에 대한 재쓰기를 위해서는 반드시 지우기 연산이 필요하다. 게다가 지우기 연산은 페이지의 집합인 블록 단위로만 가능하고 읽기/쓰기 연산에 비해 보통 수십배 느리다. 또한 한 블록에 대한 지우기 연산의 횟수는 10,000번에서 100,000번 정도로 제한이 있다.[1][2] 따라서 플래시 메모리를 저장장치로 사용하기 위해서 FTL(Flash Translation Layer)[3] 계층을 추가하거나 플래시 전용의 파일 시스템[4][5]을 이용한다. 하지만 이렇게 구성된 플래시 메모리 기반의 시스템에서 DBMS가 이용될 때 DBMS는 물리적인 저장 매체인 플래시 메모리를 직접 제어할 수 없다. DBMS의 물리적 I/O 관리 계층 아래에 OS의 페이지 캐시, 블록 디바이스 관리 계층, 파일시스템 그리고 플래시 디스크의 FTL이 존재하기 때문에 DBMS에서 플래시 메모리를 고려한 정책을 사용한다 해도 실제 플래시 메모리까지 그 영향이 미치지 힘들다.

본 논문에서는 플래시 메모리 기반의 DBMS에서 물리적 I/O 관리 계층이 플래시 메모리를 직접 통제할 수 있도록 계층 구조가 단일화된 시스템을 구현하였다. 제안된 시스템에서 DBMS의 물리적 I/O 관리 계층을 플래시 메모리의 특성을 고려해 플래시 메모리를 직접 제어할 수 있도록 재작성 하였다. 제안된 시스템은 WiSS (Wisconsin Storage System)[4]의 스토리지 시스템에 MiniSQL[5]의 쿼리 프로세서를 통합한 DBMS에 구현되었으며 리눅스 환경의 임베디드 평가보드에서 테스트되었다.

기존 DBMS의 물리적 I/O 관리 계층은 요청된 하드디스크의 블록 주소에 페이지를 읽고 쓰는 일만 담당했다. 하지만 제안된 DBMS에서는 물리적 I/O 관리 계층이 요청된 블록 주소를 플래시메모리 내 물리적 페이지 주소로 사상시켜 주는 기능이 필요하다. 테스트 용 평가보드는 64MB의 내장형 플래시 메모리를 사용했으며 저장 장치 크기에 대한 확장성이 필요치 않았으므로 페이지 단위로 사상은 알고리즘을 이용하였다. 각 페이지에 대한 사상 정보는 플래시 메모리의 스페어 영역에 저장했으며 시스템 초기화시 모든 페이지를 스캔하는 시간을 단축시키기 위해서 플래시 메모리의 특정 블록에 저장할 수 있도록 했다. DBMS에 지속적으로 레코드가 입출력 될 때의 물리적 I/O 요청 트레이스를 분석한 결과

전체 페이지 쓰기 요청의 90%가 시스템 카탈로그 테이블의 업데이트를 위한 것이었다. 게다가 시스템 카탈로그 테이블의 경우 DBMS의 버퍼 관리 계층을 거치지 않는 구조였기 때문에 모든 쓰기 요청이 실제 플래시 메모리에서 페이지 쓰기 연산을 발생시켰다. 플래시 메모리에서 같은 페이지에 대한 지속적인 쓰기 요청은 많은 플래시 메모리 페이지를 소비해 지우기 연산을 발생시키며, 지우기 연산의 경우 읽기나 쓰기 연산에 비해 느린 연산이므로 전체 시스템 성능이 감소되었다. 제안된 시스템에서는 이러한 문제를 극복하기 위해 물리적 관리 계층에서 플래시 메모리 페이지에 대한 캐시를 추가해 시스템 카탈로그 테이블 업데이트를 위한 쓰기 연산을 감소시킬 수 있도록 했다.

본 논문에서는 제안된 DBMS 통합 시스템을 Linux-2.6.8을 이용하는 S3C2410 임베디드 보드[6]에 구현하고 테스트 하였다. 실험용 보드에는 삼성 K9F1208R0B NAND 플래시 메모리[7]가 내장형으로 포함되어 있었다. 구현된 DBMS가 MTD를 이용해 직접 플래시 메모리를 제어할 때와 같은 DBMS가 플래시 파일시스템인 YaFFS[8] 위에서 동작할 때의 성능을 비교했다. 성능 비교를 위해서 플래시 메모리에 1,000,000개의 INSERT 쿼리를 발생시켰다. 각각의 INSERT 쿼리 요청은 데이터 테이블에 대한 쓰기 요청과 시스템 카탈로그 테이블에 대한 업데이트 요청을 발생시키게 된다.

실험 결과 플래시 메모리에서의 읽기 요청, 쓰기 요청 횟수가 반 이상 감소된 것을 확인할 수 있었으며 블록 지우기 연산도 50% 이상 감소되었다. 그리고 수행시간도 70%이상 감소되었다.

본 논문에서는 하드디스크 기반의 DBMS와 OS의 스토리지 시스템 계층을 통합해 플래시 메모리를 DBMS에서 직접 제어할 수 있는 시스템을 제안했으며 실제 구현해 기존 시스템과의 성능을 비교해 실험 결과 기존 시스템에 비해 성능이 70% 이상 향상된 것을 확인할 수 있었다. 향후 연구로 플래시 메모리를 위한 버퍼관리 알고리즘, 인덱스 구조, 비용 계산을 통한 access path 선택, external sorting 알고리즘 등을 플래시 메모리에서 효과적으로 동작하도록 개선시키는 연구를 수행할 계획이다.

참고문헌

- [1] H. Kim and S. Lee., "A New Flash Memory Management for Flash Storage System," In 32nd Annual Intl. Computer Science and Applications Conference, October 1999.
- [2] Chanik Park, Jeong-Uk Kang, Seon-Yeong Park, Jin-Soo Kim, "Energy-aware demand paging on NAND flash-based embedded storages," Proceedings of the 2004 international symposium on Low power electronics and design table of contents, pp338 - 343, 2004
- [3] Intel. Understanding the Flash Translation Layer (FTL) Specification. Application Note AP-684, Intel Corporation, December 1998.
- [4] Hong-Tai Chou, David J. DeWitt, Randy H. Katz, Anthony C. Klug: Design and Implementation of the Wisconsin Storage System. Softw., Pract. Exper. 15(10): 943-962(1985)
- [5] Hughes Technologies, MiniSQL available from: <http://www.hughes.com.au/products/msql/>
- [6] FALINUX, EZ-S2410 available from: <http://falinux.com/zproducts/ez-s2410.php>
- [7] Samsung Elec., "64Mx8 Bit NAND Flash Memory", available from: <http://www.datasheetarchive.com/pdf/1880138.pdf>
- [8] YAFFS, A Flash Filesystem for Embedded Use. Available from: <http://www.yaffs.net/>