

# 플래시 메모리 기반의 저장장치를 위한 비대칭적 버퍼 관리기법

이선복\*, 정호영\*, 심효기\*, 박성민\*, 차재혁\*, 강수용\*  
 \*한양대학교 전자컴퓨터통신공학과  
 {ashavaisht, horong, dahlia, syrilo, chajh, sykang}@hanyang.ac.kr

## ABM: Asymmetric Buffer Management for Flash Memory based Storage System

Seonbock Lee\*, Hoyoung Jung\*, Hyogi Sim\*, Sungmin Park\*, Jaehyuk Cha\*, Sooyong Kang\*  
 \*Dept. of Electronics Computer Engineering, Hanyang University

### 요 약

최근 플래시 메모리 기반의 저장장치가 하드디스크 영역에서 광범위하게 사용되고 있다. 하지만 단점을 가지고 있는데 느린 쓰기 및 지우기 속도가 그것이다. 본 논문에서는 플래시 메모리의 쓰기 및 지우기 동작을 줄일 수 있는 플래시 메모리 기반의 저장장치를 위한 새로운 버퍼 관리 기법을 제안한다. 제안된 ABM 기법은 플래시 메모리 기반의 저장장치에서 약 30%의 성능향상을 보이고 있다.

### 1. 서론

플래시 메모리는 저전력, 작은 크기, 경량, 높은 충격내구성 등 많은 장점을 가지고 있다[1]. 이러한 이유로 플래시 메모리는 간단한 이동형 저장장치나 휴대형 전자기기에 많이 쓰이고 있다. 최근에는 Solid-state Drive(SSD)의 형태로 개인용 컴퓨터와 서버에도 채택되고 있다[2].

플래시 메모리를 하드디스크와 같은 블록 디바이스로 사용하기 위해서는 Flash Translation Layer(FTL)라고 불리는 특별한 시스템 소프트웨어가 필요하다[3]. 플래시 메모리 기반의 저장장치는 저장장치로 사용하기 위해서는 특정한 제한을 가지고 있기 때문에 현존하는 버퍼교체정책은 적합하지 않다.

본 논문에서는 플래시 메모리의 성능과 수명을 늘리기 위해 쓰기와 지우기 동작을 줄이는 비대칭적 버퍼 관리기법(ABM)을 제안한다.

### 2. 관련 연구

#### 2.1. 플래시 메모리

NAND 플래시 메모리는 하드디스크와는 다르게 몇 가지 하드웨어적인 제한을 가지고 있다. <표 1>은 NAND 플래시 메모리의 특성을 나타낸 것이다.

<표 1> 플래시 메모리의 특성[4]

Operation	Time(us)
256KB Block Erase	2000
2KB Page Read	25
2KB Page Write	200

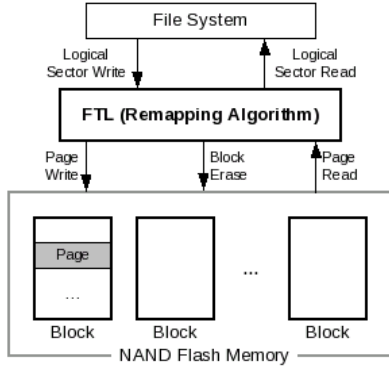
플래시 메모리의 쓰기 속도는 읽기 속도보다 느리며 지우기 속도는 쓰기 속도보다 현저히 느리다. 또한 읽기/쓰기의 단위는 페이지이지만 지우기의 단위는 여러 페이지로 이루어진 블록이다. 그리고 덮어쓰기가 불가능하기 때문에 페이지에 변경이 필요할 경우 지우기 작업이 선행되어야 한다.

데이터를 변경하기 위해서는 새로운 블록을 할당 받고 데이터를 포함하고 있는 페이지들을 이전 블록에서 새로운 블록으로 복사한 후 이전 블록을 지운다. 이러한 추가적인 작업을 머지(Merge) 작업이라 부른다. 또한 플래시 메모리는 지우기 횟수가 제한되어 있는데 보통 블록당 10,000 ~ 100,000 번 정도이다.

#### 2.2. Flash Translation Layer(FTL)

FTL 은 (그림 1)에서와 같이 플래시 메모리를 블록 디바이스로 변환할 때 사용된다. FTL의 주요 기능은 플래시 메모리의 한계를 극복하는 것이다. 이러한 목적을 이루기 위해 FTL은 LSN(Logical sector number, 파일시스템)을 PPN(Physical page number, 플래시 메모리)로 변환한다. FTL은 맵핑(mapping) 알고리즘에 따라 페이지 맵핑 FTL, 블록 맵핑 FTL, 로그 블록 FTL의 3가지 그룹으로 분류될 수 있다.

플래시 메모리 기반의 저장장치가 하드디스크와 같이 덮어쓰기가 가능하기 위해서는 블록 지우기와 머지 작업이 필요하다. 이러한 작업들은 추가적인 지우기와 쓰기 작업을 발생시키고 이는 성능하락의 주요요인이다. 따라서 플래시 메모리 기반의 저장장치의 성능을 증가시키기 위해서는 지우기와 쓰기 작업을 줄이는 것이 필요하다.



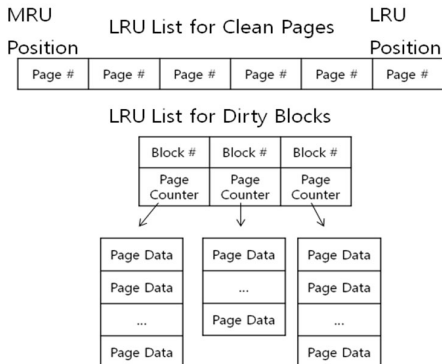
(그림 1) NAND 플래시와 FTL

### 3. ABM-LRU-FAB

FAB[5]와 같은 현존하는 플래시 메모리를 위한 버퍼교체 알고리즘들은 쓰기과 지우기를 줄이는데 목적을 두고 있으나 히트율(Hit Ratio)을 고려하지 않았기 때문에 과도한 읽기 작업을 유발한다. 그래서 단지 특정한 경우에만 좋은 성능을 보이고 있다. 일반적으로 플래시 메모리 버퍼 매니저의 성능을 높이기 위해 고려해야 할 사항은 다음과 같다 : (1) 지우기 작업 감소, (2) 쓰기 작업 감소, (3) 히트율의 하락 방지.

이러한 목적을 달성하기 위해서 ABM 은 다음과 같은 전략을 사용한다 : (1) 읽기 버퍼와 쓰기 버퍼의 분리, (2) 각각의 버퍼에 다른 버퍼교체알고리즘 적용, (3) 각 버퍼교체알고리즘별 다른 작업 단위 사용. 읽기 버퍼의 버퍼교체알고리즘은 히트율을 최대화하기 위해 LRU, ARC[6], LIRS[7]와 같이 하드디스크를 위한 알고리즘을 채택한다. 또한 쓰기 버퍼를 위한 버퍼교체알고리즘은 쓰기과 지우기 작업을 최소화하기 위해 FAB, BPLRU[8]와 같은 알고리즘을 채택한다. 그리고 읽기 버퍼의 단위는 페이지이고 쓰기 버퍼의 단위는 로그블록 FTL의 성능을 위해 블록을 사용한다.

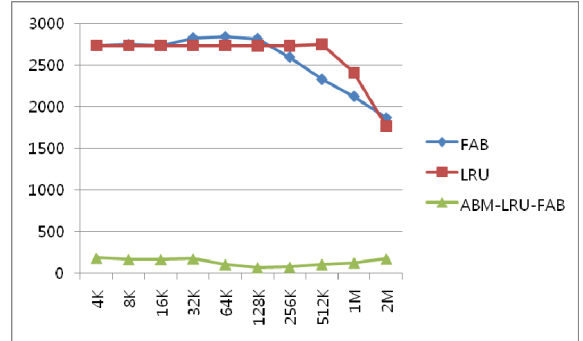
(그림 2)는 ABM의 데이터 구조이다. 그림에서는 읽기 버퍼는 LRU, 쓰기 버퍼는 FAB를 선택하였다.



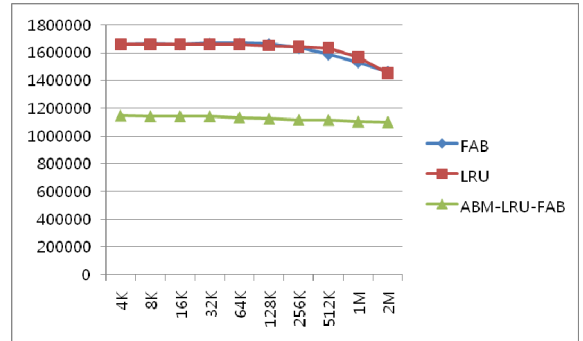
(그림 2) ABM의 데이터 구조

### 4. 성능 평가

ABM 성능 평가를 위해 트레이스(Trace) 데이터를 이용한 시뮬레이션[9]을 사용하였으며 트레이스 데이터는 데스크톱 애플리케이션에서 추출하였다.



(그림 3) FAB, LRU, ABM의 지우기 횟수



(그림 4) FAB, LRU, ABM의 수행시간

(그림 3)과 (그림 4)는 각 버퍼교체알고리즘의 지우기 횟수 및 수행시간(Total Elapsed Time)을 보여주고 있다. X-축은 실험에서 사용한 버퍼의 크기이다. 수행시간은 읽기/쓰기/지우기 비율을 1:10:80[10]으로 설정하여 각각의 연산횟수에 비율을 곱셈하여 계산하였다.

(그림 3)에서 FAB는 읽기와 쓰기 페이지 모두를 블록단위로 관리하고 교체 블록을 선정할 때 페이지가 가장 많은 블록을 선택한다. 이때 읽기 페이지와 쓰기 페이지가 섞여있는 블록이 선택될 수 있고 이로 인해 임의의 쓰기가 발생되어 성능이 하락할 수 있다. 이에 반해 ABM은 쓰기 페이지만 블록단위로 관리하여 임의의 쓰기보다는 순차쓰기가 늘어나 지우기 횟수가 현저하게 줄어들어 LRU와 FAB보다 더 나은 성능을 보이고 있다. 지우기 횟수의 감소로 인해 플래시 메모리의 수명도 증가할 것이라고 예상할 수 있다.

(그림 4)는 (그림 3)에서 지우기 횟수가 줄어든 결과로 ABM이 25%~30%의 수행시간 성능향상을 보이고 있다.

### 5. 결론 및 향후 과제

플래시 메모리 기반의 저장장치의 성능과 수명을

위해 지우기와 쓰기 작업을 줄이기 위한 버퍼교체정책인 ABM 을 제안하였다. ABM 은 기존의 알고리즘보다 높은 성능을 나타냈으며 플래시 메모리의 수명을 늘리고 있다.

추후 실제 환경하에서 여러 가지 알고리즘에 ABM 을 적용하여 테스트할 예정이다.

### 참고문헌

- [1] Fred Douglis, Ramon Caceres, M. Frans Kaashoek, Kai Li, Brian Marsh, and Joshua A. Tauber. "Storage alternatives for mobile computers. "In *Operating Systems Design and Implementation*, 1994, pp. 25-37
- [2] Hyojun Kim and Seongjun Ahn, "BPLRU: A buffer management scheme for improving random writes in flash storage", FAST '08, 2008, pp. 239-252
- [3] Intel Corporation. "Understanding the Flash Translation Layer(FTL) Specification", 1988, <http://www.embeddedfreebsd.org/Documents/Intel-FTL.pdf>
- [4] Samsung Electronics, "128M x 8 Bit / 256M x 8 Bit NAND Flash Memory", <http://pdf1.alldatasheet.com/datasheet-pdf/view/109725/SAMSUNG/K9F1G08U0A.html>
- [5] Heeseung Jo, Jeong-Uk Kang, Seon-Yeong Park, Jin-Soo Kim, and Joonwon Lee, "FAB: Flash-Aware Buffer Management Policy for Portable Media Players," IEEE Transactions on Consumer Electronics, Vol. 52, No. 2, pp.485-493, May 2006.
- [6] Nimrod Megiddo, Dharmendra Modha, "ARC: A Self-Tuning, Low Overhead Replacement Cache", Proc. 2nd USENIX Conference on File and Storage Technologies (FAST 03), 2003.
- [7] Song Jiang, Xiaodong Zhang, "LIRS: an efficient low inter-reference recency set replacement policy to improve buffer cache performance", ACM SIGMETRICS Performance Evaluation Review archive Vol. 30, Issue 1 pp31-42, June 2002.
- [8] Hyojun Kim, Seongjun Ahn, "BPLRU: A Buffer Management Scheme for Improving Random Writes in Flash Storage", 6<sup>th</sup> USENIX Conference on File and Storage Technologies(FAST 08), Feb, 2008
- [9] 윤경훈, 정호영, 박성민, 심효기, 차재혁, 강수용, "IPSiNS: 낸드 플래시 메모리 기반 저장 장치를 위한 입출력 성능 시뮬레이션 도구", 한국정보과학회 2007 한국컴퓨터종합학술대회 논문집 제 34 권 제 1 호(A).
- [10] Samsung Electronics, "128M x 8 Bit / 256M x 8 Bit NAND Flash Memory", <http://pdf1.alldatasheet.com/datasheet-pdf/view/109725/SAMSUNG/K9F1G08U0A.html>