

# 맵 캐시 기법을 사용하는 낸드 플래시 저장장치의 맵 블록 개수에 따른 성능 관계 분석

이대용\*, 송용호\*\*

\*한양대학교 전자컴퓨터통신공학과

\*\*한양대학교 융합전자공학부

e-mail: {dylee, yhsong}@enc.hanyang.ac.kr

## Performance Relationship Analysis in Map Block Number of NAND Flash Storage Device Using Map Cache Techniques

Daeyong Lee\*, Yong Ho Song\*\*

\*Dept. of Electronic Engineering, Hanyang University

\*\*College of Engineering, Hanyang University

### 요 약

맵 캐시 기법을 사용하는 낸드 플래시 저장장치는 맵 데이터를 저장하기 위한 공간을 필요로 한다. 이 공간을 맵 블록이라 부르며 시스템 유지 및 성능 개선을 위해 사용되는 낸드 블록의 일부를 점유한다. 맵 블록의 개수가 너무 많을 경우 시스템 유지에 필요한 블록이 부족해지기 때문에 전반적인 성능이 하락하게 된다. 하지만 맵 블록이 너무 적은 경우에도 전체 맵 데이터를 유지하기 위한 동작이 과도하게 수행되어 성능이 크게 하락하는 문제가 발생한다. 본 논문은 맵 블록 개수에 따른 성능 변화를 분석하고 최적의 맵 블록 개수를 제안한다.

### 1. 서론

낸드 플래시 저장 장치는 작은 크기와 작은 무게, 적은 전력 소모량, 충격에 강한 특성 등의 장점에 현재 널리 사용되고 있다. 하지만 낸드 플래시는 덮어 쓰기를 지원하지 않고 블록 단위 지우기만 가능한 단점을 가지고 있다. 이를 보완하기 위해 낸드 저장 장치는 FTL (Flash translation layer)라는 시스템 소프트웨어를 사용한다.

FTL의 주요 역할은 호스트 시스템에서 사용하는 논리 주소를 낸드 플래시 내부 물리 주소로 변환하는 매핑 (Mapping) 기능과 무효 (Invalid) 페이지를 정리하여 사용 가능한 블록을 확보하는 가비지컬렉션 (Garbage collection) 기능이다. FTL이 매핑 동작을 수행하기 위해선 매핑 정보를 기록하고 항상 최신 상태를 유지해야한다. 보편적으로 FTL이 기록하는 매핑 정보는 테이블 구조이기 때문에 맵 테이블 (Map table)이라고 부른다. 맵 테이블은 휘발성 메모리를 점유하기 때문에 비용과 전력 소모량 상승 문제가 제기되어 왔다. 최근에는 낸드 플래시 저장장치의 고성능 대용량화가 진행되면서 맵 테이블의 크기가 더욱 커졌고 이러한 문제를 해결하기 위해 맵 캐시 (Map cache) 기법 [1], [2]의 연구가 활발하게 진행 되었다.

맵 캐시 기법은 전체 맵 테이블을 맵 블록 (Map block)이라 부르는 낸드 저장장치 영역에 저장하고 맵의 일부분만을 작은 휘발성 메모리에 캐싱

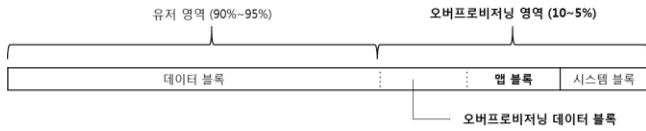
(Caching) 하여 사용하는 방법이다. 맵 블록은 유저가 사용 가능한 영역 외에 성능 개선 및 시스템 관리 목적으로 제공되는 영역에 저장된다. 이 영역을 오버프로비저닝 (Over-provisioning) 영역이라 부른다.

오버프로비저닝 영역은 맵 블록 외에도 시스템 유지 및 관리에 사용되며 이외에 남는 블록은 성능 향상을 위한 오버프로비저닝 데이터 블록으로 사용된다. 오버프로비저닝 데이터 블록은 일반 데이터 블록과 동일하게 사용되지만 유저 (User) 사용 영역에 포함되지 않는다. 오버프로비저닝 데이터 블록이 충분할 경우 블록당 유효 페이지 개수가 감소하기 때문에 많이 확보할 수록 가비지컬렉션 성능 개선 효과가 있다. 제조사는 최대한 많은 오버프로비저닝 데이터 블록을 확보하기 위해 시스템 블록과 맵 블록의 개수를 최소화 해야한다. 하지만 오버프로비저닝 데이터 블록을 최대화 하기 위해 맵 블록 개수를 과도하게 줄일 경우 오히려 맵 블록 가비지컬렉션이 과다 발생하여 성능이 감소하게 된다.

본 논문은 맵 블록과 오버프로비저닝 데이터 블록의 성능 트레이드-오프 (Trade-off) 관계를 분석하고 실험을 통해 최적 비율을 탐색한다.

### 2. 배경

가) 논리적 블록 관리 기법



(그림 1) 논리적 블록 영역 구분

물리적으로 낸드 플래시의 블록은 모두 동일하다. 하지만 FTL 은 논리적으로 블록의 역할을 정하고 각 역할의 비율을 유지한다. 물리적 낸드 플래시 블록이 100% 라면 사용자가 사용 가능한 비율은 약 90~95% 정도이며 유저 영역이라고 부른다. 나머지를 오버프로비저닝 영역이라 부르며 맵 블록, 오버프로비저닝 데이터 블록, 시스템 블록으로 나뉘어 사용한다. 맵 블록은 맵 캐시 사용시 전체 맵 테이블을 저장하기 위한 공간으로 사용된다. 시스템 블록은 초기 베드 블록, 서든 파워 오프 대비 메타 저장 공간으로 사용된다. 오버프로비저닝 데이터 블록은 일반 데이터 블록과 동일하게 사용된다.

나) 가비지컬렉션

가비지컬렉션 [4] 은 무효 페이지를 정리하여 사용 가능한 클린 (Clean) 블록을 확보하는 기능이다. 보편적으로 가비지컬렉션은 수행 대상 (Victim) 블록의 유효 (Valid) 페이지를 클린 블록에 옮겨 쓴 뒤 대상 블록을 지우는 동작을 수행한다. 가비지컬렉션의 효율은 대상 블록의 유효 페이지 개수에 따라 달라진다. 만약 유효 페이지가 많을 경우 데이터를 옮겨야 할 뿐만 아니라 삭제 이후 얻게 되는 클린 페이지 개수도 적어진다. 이러한 이유로 가비지컬렉션 대상 블록은 유효 페이지가 적은 블록을 선정한다.

다) 맵 블록 가비지컬렉션

맵 캐시 기법 사용시 전체 맵 데이터를 저장하는 낸드 블록을 맵 블록이라 한다. 맵 블록은 제조사 또는 알고리즘에 의해 최대 개수가 제한된다. 맵 정보에 변화가 생길 경우 맵 캐시 알고리즘에 따라 새로운 맵 페이지에 맵 데이터가 기록되며 기존 맵 페이지는 무효 페이지가 된다. 만약 맵 블록의 무효 페이지 개수가 증가하여 최대 허용 맵 블록 개수를 초과할 경우 맵 블록 역시 가비지컬렉션을 통해 클린 페이지를 확보해야 한다. 만약 허용된 맵 블록 개수가 많을 경우 블록당 유효 페이지 개수가 적어지므로 가비지컬렉션 효율이 개선된다. 맵 블록은 데이터 블록에 비해 상대적으로 갱신 주기가 짧다. 만약 최대 맵 블록 허용 개수가 너무 적을 경우 잦은 맵 블록 가비지컬렉션으로 인한 성능 하락 문제를 유발한다.

라) 오버프로비저닝 데이터 블록

오버프로비저닝 데이터 블록은 오버프로비저닝 영역의 블록 중 데이터 블록으로 사용되는 블록을 의미한다. 오버프로비저닝 영역은 맵 블록과 시스템 블록

으로 사용되는 영역을 제외하고 모두 오버프로비저닝 데이터 블록으로 사용한다. 오버프로비저닝 데이터 블록은 유저 영역에 포함되지 않는다. 오버프로비저닝 데이터 블록이 많을 경우 여분의 데이터 블록 개수를 증가시켜 블록당 유효 페이지 개수를 감소시키는 효과가 있다. 유효 페이지 개수가 감소할 경우 가비지컬렉션 효율이 개선되어 성능이 향상되기 때문에 오버프로비저닝 데이터 블록은 성능 향상에 중요한 역할을 수행한다.

3. 실험 환경

가) 시뮬레이션 환경

<표 1> 낸드 플래시 환경

전체 저장 장치 용량	64 GB
셀 당 비트 수	2 (MLC)
다이 개수 (다이 당 플레인 개수)	8 다이 (2 플레인)
버스 모델	2 채널 - 4 웨이
페이지 크기 (블록 당 워드 라인)	8 KB (256)
블록 크기 (플레인 당 블록 개수)	4 MB (1024)
슈퍼 블록 [5] 크기 (전체 개수)	64 MB (1024)

<표 2> FTL 환경

FTL 알고리즘	S-FTL[2] 맵 캐시 기법
매핑 크기	4 KB (sub-page mapping)
SRAM 크기 (using only FTL)	256 KB
가비지컬렉션 알고리즘	On-demand 가비지컬렉션

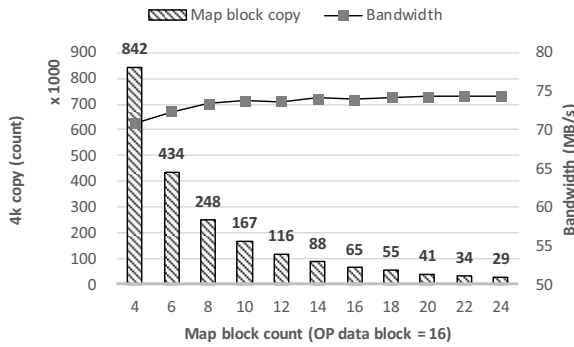
<표 3> 블록 환경 (슈퍼 블록 기준)

전체 블록 개수	1024
유저 영역 블록 개수	944 ~ 960 (약 93%)
오버프로비저닝 블록 개수	64 ~ 80 (약 7%)

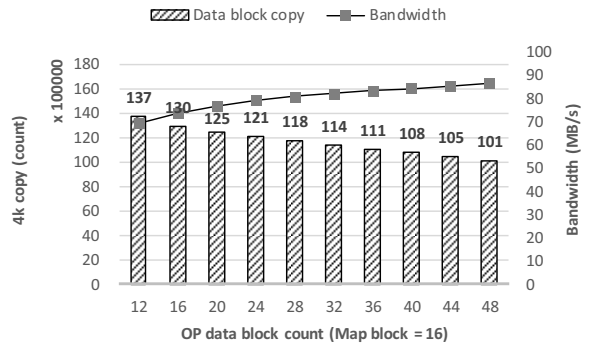
실험은 이벤트-기반 시뮬레이션 (Event-driven simulation) 기법 [3] 이 적용된 시뮬레이터를 사용하여 수행되었다. 현재 모바일 저장 장치로 주로 사용되는 낸드 플래시 환경을 모델링 하였으며 4kbyte 서브-페이지 (Sub-page) 매핑, 슈퍼 블록 (Super block) [5] 등 엔터프라이즈 수준의 기법이 적용되어 실험의 신뢰성을 높였다.

나) 워크로드 (Workload)

실험에 사용된 워크로드는 크게 가상 워크로드와 실제 워크로드로 나뉜다. 가상 워크로드는 실험을 위해 인위적으로 제작된 워크로드를 의미한다. 특수한 상황이나 확보하기 어려운 워크로드를 테스트하는 용도로 사용된다. 실제 워크로드는 실제 시스템 사용 환경에서 추출한 워크로드를 의미한다. 가상 워크로드와 달리 실제 워크로드는 로컬리티 (Locality) [10] 의 영향을 받는다. 본 논문의 검증은 총 네 개의 위

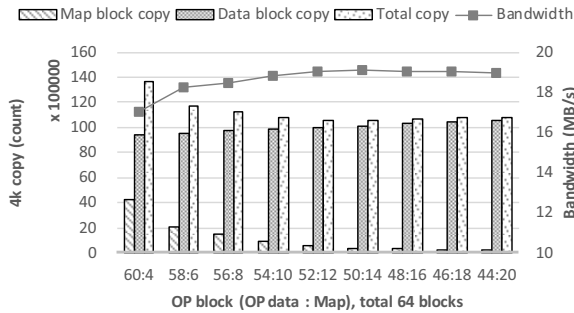


(a) 맵 블록 증가에 따른 성능 변화

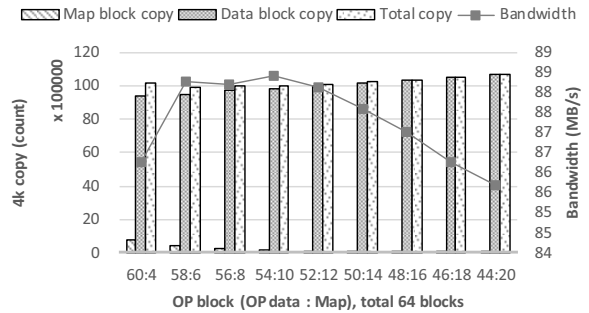


(b) 오버프로비저닝 데이터 블록 증가에 따른 성능 변화

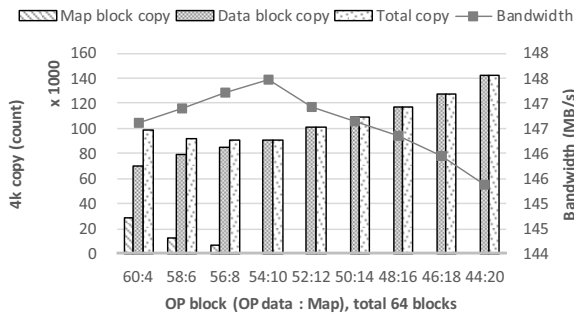
(그림 2) 시퀀셜 (80%) + 랜덤 (20%) 가상 워크로드 수행 결과



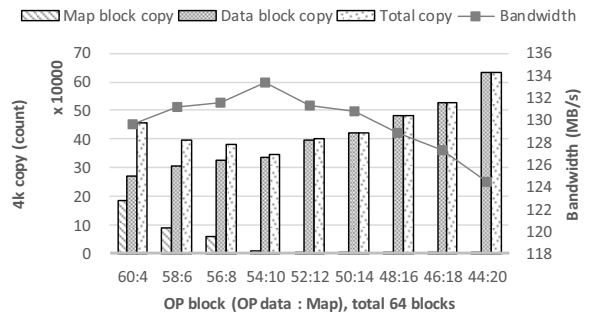
(a) 랜덤(100%) 트레이스



(b) 시퀀셜 (80%) + 랜덤 (20%) 트레이스



(c) 파이낸셜 트레이스



(d) MSR 트레이스

(그림 3) 오버프로비저닝 블록의 할당 비율에 따른 성능 변화

크로드로 수행 되었으며 두 개의 가상 워크로드와 두 개의 실제 워크로드를 사용하였다.

가상 워크로드는 트레이스 제너레이터 (Trace generator) [7] 를 활용하여 생성하였으며 랜덤 (Random), 시퀀셜 (Sequential) 비율을 조정할 수 있다. 사용된 가상 워크로드는 워스트 케이스 (Worst case) 를 가정한 랜덤 100% 워크로드와 일반적인 사용 환경을 가정한 시퀀셜 80%와 랜덤 20%를 혼합한 워크로드가 실험에 사용되었다. 실제 워크로드는 파이낸셜 (Financial) 워크로드 [8] 와 MSR 워크로드 [9] 를 사용한다.

다) 실험 환경 설정

모든 실험은 유저 영역 중 80%에 프리-필링 (Pre-filling) 을 수행 하였다. 프리-필링은 테스트 환경을 실제 저장 장치를 사용하는 환경처럼 만들기 위해 일

부 공간을 미리 유효 데이터로 채워두는 동작을 의미한다.

해당 실험의 전체 맵 테이블은 약 128 MB 크기 이다. 이는 logical to physical 테이블과 physical to logical 테이블을 모두 포함한 크기이다. 맵 테이블 128 MB 는 최소 2 개의 맵 블록 할당 시 시스템 유지가 가능 하지만 성능 하락 폭이 크기 때문에 실험에 사용된 맵 블록 최소 개수는 4 개로 정한다.

4. 실험 결과

가) 블록 개수 증가에 따른 성능 변화 분석

(그림 2) 는 가상 워크로드를 활용하여 맵 블록 증가와 오버프로비저닝 데이터 블록 증가에 따른 개별 성능 변화를 확인하기 위해 수행 되었다. (a) 맵 블록

개수가 증가함에 따라 맵 블록 페이지 카피 횟수가 크게 줄어든다. 이는 맵 블록 내 유효 페이지 감소에 따른 효과이며 맵 블록 개수가 충분히 클 경우 감소 폭은 크게 감소하여 포화 되는 경향을 보인다. (b) 오버프로비저닝 데이터 블록은 개수 증가에 따라 데이터 블록 카피 횟수가 일정하게 감소한다. 카피 횟수 감소량이 일정하기 때문에 성능도 거의 일정하게 증가하는 것으로 알 수 있다. 맵 블록과 달리 성능 포화 경향이 적다.

나) 오버프로비저닝 데이터 블록과 맵 블록의 할당 비율에 따른 성능 변화 분석

(그림 3) 는 전체 64 개의 오버프로비저닝 블록을 맵 블록과 오버프로비저닝 데이터 블록으로 사용 할 경우 최적의 비율을 확인하기 위해 수행되었다.

실험은 2 개의 가상 워크로드와 2 개의 실제 워크로드 총 4 가지 워크로드로 수행되었다. 그래프의 좌측 실험으로 갈수록 맵 블록이 적고 오버프로비저닝 데이터 블록이 많은 비율이며 우측으로 갈수록 맵 블록이 많아지는 대신 오버프로비저닝 데이터 블록이 적어진다.

(a)-(d) 모두 맵 블록 비율이 낮을 경우 많은 맵 블록 카피가 발생하지만 맵 블록 개수가 증가할 수록 카피가 크게 감소하여 약 10~14 개 에서 카피가 거의 발생하지 않게 된다. 오버프로비저닝 데이터 블록은 비율이 증가할 수록 꾸준히 데이터 블록 카피 횟수가 감소하는 경향을 보이지만 변화 폭이 맵 블록 카피 변화 보다 작다.

최고 성능은 (a) 의 경우 오버프로비저닝 데이터 블록 대 맵 블록 비율 50:14 에서, (b), (c), (d) 는 약 54:10 에서 나타난다. 최적 성능이 54:10~50:14 에서 나온 이유는 맵 블록의 성능 포화 개수가 10~14 개 이기 때문이다. 이는 맵 블록의 성능 영향이 오버프로비저닝 데이터 블록보다 크지만 성능 포화 개수인 10~14 개 이상을 할당할 경우 성능상 이득이 없음을 의미한다.

결과적으로 오버프로비저닝 블록의 최적 할당 비율은 최적의 맵 블록 개수에 따라 달라지게 된다. 해당 실험 환경에선 맵 블록 약 10~14 개가 최적 개수인 것으로 나타났다. 이는 전체 블록의 1% 정도이며 전체 맵 데이터가 저장 되기 위한 최소 블록 개수의 5 배 크기임을 고려하여 환경에 따라 적용이 가능하다.

## 5. 결론

맵 캐시 기법을 사용하는 환경에서 맵 블록의 개수와 오버프로비저닝 데이터 블록의 관계는 성능 최적화에 매우 중요하다. 최적 환경과 최저 환경간의 성능 차이는 최대 28.7% 이며 환경에 따라 더욱 커질 수 있다.

향후 본 논문에서 다루지 않은 시스템 블록의 영향과 다양한 맵 캐시 기법 적용 시 성능 관계를 연구할 필요성이 있다.

## 참고문헌

- [1] Kim, Y. (2009). DFTL: A Flash Translation Layer Employing Demand-based Selective Caching of Page-level Address Mappings, 229–240.
- [2] Jiang, S., Zhang, L., Yuan, X., Hu, H., & Chen, Y. (2011). S-FTL: An Efficient Address Translation for Flash Memory by Exploiting Spatial Locality.
- [3] Jang, M., & Song, Y. H. (2015). Dynamic Resource Assignment Model for Event-Driven Simulator, 539–540.
- [4] Bux, W., & Iliadis, I. (2010). Performance of greedy garbage collection in flash-based solid-state drives. *Performance Evaluation*, 67(11), 1172–1186. <http://doi.org/10.1016/j.peva.2010.07.003>
- [5] Micron Technology. (2005). TN-29-28: Memory Management in NAND Flash Arrays Technical Note Memory Management in NAND Flash Arrays Performance Benefits of Combining NAND Flash Memory Arrays, 1–10.
- [6] JEDEC Standard – Solid-State Drive(SSD) Endurance Workloads. 2010, (<http://www.jedec.org>)
- [7] Lee, S., Lee, D., & Song, Y. H. (2016). Wear-leveling Experiments of Workload Generator Design and Implementation, 5–6.
- [8] UMass Trace Repository: Financial trace. <http://traces.cs.umass.edu/index.php/Storage/Storage>
- [9] SNIA IOTTA Repository: MSR Cambridge traces. <http://iota.snia.org/traces/388>
- [10] Lee, S., Kim, Y., & Kim, J. (2008). LAST: Locality-Aware Sector Translation for NAND Flash Memory-Based Storage Systems, 36–42.