

κ-평준화: 플래시 메모리의 효율적인 소거 횟수 평준화 기법*

김도윤⁰ 유현석 박성환 박원주 박상원

한국의국어대학교 컴퓨터및정보통신공학부

{dykim⁰, hsyoo, shpark, wjpark, swpark}@islab.hufs.ac.kr

κ-leveling: An Efficient Wear-leveling Scheme for Flash Memory

Do Yun Kim⁰ Hyun-Seok Yoo Sung-Hwan Park Won-Joo Park Sangwon Park

Computer Science & Information Communication Engineering Div., Hankuk University of Foreign Studies

요 약

최근 이동성이 중요한 요소로 차지하는 기기들이 등장하면서 플래시 메모리가 각광을 받고 있다. 플래시 메모리의 소형화, 대용량화, 저전력화, 비휘발성, 고속화 그리고 충격에 강한 장점으로 인하여 많은 응용에서 디스크를 대체할 것으로 예상된다. 하지만 이런 플래시 메모리는 데이터를 기록하기 전에 해당 블록이 미리 소거가 되어야 하는 제약 조건을 가지고 있으며 각 블록들의 최대 소거 횟수가 제한되어 있다는 한계가 있다. 이때 소거 연산이 특정 블록에 집중되어 특정 블록의 수명이 단축되는 문제점을 해결하기 위하여 블록에 대한 소거 횟수 평준화 기법(wear-leveling)이 필요하다. 기존에 제안된 소거 횟수 평준화 기법은 각 블록의 소거 횟수를 유지해야하는 비용이 필요로 하거나 플래시 메모리가 대용량일 경우에는 블록 영역을 이동시키는데 비용이 발생하는 문제가 있다. 본 논문에서는 플래시 메모리의 소거 횟수 평준화를 위하여 해당 블록의 소거 횟수에 대한 정보의 유지의 부담을 줄이고 플래시 메모리의 대용량화 및 디스크 대체 시에 효율적인 소거 횟수 평준화 기법을 제안하고, 실험을 통하여 성능의 우수함을 보인다.

1. 서론

최근 디지털 카메라, MP3 플레이어, 핸드폰 등 이동성이 중요한 요소로 차지하는 기기들이 많이 등장하였다. 이에 따라 소형화, 대용량화, 저 전력화, 비휘발성, 고속화 그리고 충격에 강한 메모리가 필요하게 되었다. 이런 장점들로 인하여 기존 디스크 저장 장치에 비해 빠른 데이터 접근 성능을 보장하게 됨으로써 많은 응용에서 플래시 메모리가 디스크를 대체할 것이라고 예상된다.

하지만 플래시 메모리는 특정 블록에 쓰기 연산을 하기 전에 해당 블록은 미리 소거(erase)되어 있어야 하는 제약이 있다. 이에 따라 플래시 메모리에 저장된 데이터를 직접 갱신하는 것은 불가능하기 때문에 빈 블록을 확보하고 변경된 데이터를 갱신하는 작업이 필요하다. 또한 각 블록은 소거될 수 있는 한계 횟수(보통 10만 번)를 가지고 있기 때문에 각 블록에 대한 소거 횟수를 평준화하는 기법이 필수적이다. 즉 플래시 메모리는 소거 연산이 한 블록에 집중이 되면 일부 블록을 사용할 수 없게 되어 플래시 메모리의 수명이 오래 가지 못하게 된다. 따라서 플래시 메모리는 특정 블록에 있어서 소거 연산이 집중되지 않고 각 블록에 대하여 균등한 소거 횟수를 가질 수 있도록 소거 횟수 평준화(wear-leveling) 기법이 필수적이다. 이런 소거 횟수 평준화 기법은 플래시 메모리를 안정적으로 오래 사용할 수 있도록 하는 방법이 되며 플래시 메모리가 디스크를 대체할 때 매우 중요한 요소가 된다.

기존의 알고리즘은 각 블록의 소거 횟수를 기록하여 빈 블록을 할당하는 과정에서 소거 횟수가 적은 블록을 할당하는 방법으로 소거 횟수를 평준화하고 있다. 하지만 이런 방법은 복잡한 연산을 필요로 하며 빈 블록을 할당하기 위해 전체 블록을 검색해야하는 오버 헤드가 발생하게 된다. 이 외에도 트랜잭션 기반 플래시 메모리 파일 시스템(Transaction Flash Memory File System, TFFS)[1]에서 사용하는 영역 이동 기법은 고정된 영역에 블록을 이동해야 하는 제약이 있다. 또한 TFFS의

영역 이동 기법은 메모리가 작은 소형 이동 기기에 적합하기 때문에 대용량의 플래시 기반 디스크에서는 연산의 복잡도를 줄이면서 좀 더 효과적으로 소거 횟수를 평준화하는 기법이 필요하다.

본 논문에서는 블록의 소거 횟수에 대한 레벨 κ 를 두어서 소거 횟수가 편차 이내에서 이루어지게 하여, 많은 소거 연산이 들어오더라도 각 블록의 소거 횟수가 균등한 수준이 될 수 있도록 하는 기법을 제안한다. 이 기법에서는 각 블록의 소거 횟수를 적은 메모리에 기록하여 소거 횟수 평준화를 보장하며 빈 블록을 할당하는 기간 내에 평준화 작업을 위한 블록의 교체 작업이 이루어지므로 계산 비용을 줄인다. 또한 이 기법에서는 블록에 대한 교체 작업을 제한함으로써 교체 작업에 대한 비용을 최소화하고 효과적으로 소거 횟수 평준화 기능을 수행한다.

2. 관련 연구

소거 횟수 평준화를 통해 플래시 메모리 저장 장치의 내구성을 개선하려는 연구는 초기의 블록 관리 기법의 개발과 함께 있어 왔다. eNVy 시스템[2]에서는 블록 삭제 비용과 삭제 빈도의 곱으로 표시되는 삭제 색인(cleaning index)을 이용하여 블록 삭제 작업을 수행한다. 이때 삭제 색인 값이 큰 블록이 우선적으로 선택되기 때문에 삭제 빈도가 높은 블록에 기록과 소거 동작이 집중되는 문제가 발생한다. eNVy 시스템은 소거 횟수가 많은 블록과 소거 횟수가 적은 블록에서 유효한 데이터를 서로 교환함으로써 문제를 해결하였다. 그러나 매번 삭제 작업 때 마다 삭제 색인을 계산하여 전체 블록과 비교하므로 계산 비용이 많이 든다. TFFS에서는 상대적으로 사용 빈도가 높은 맵 영역과 교대 영역의 위치를 주기적으로 이동시킴으로써 특정한 블록들이 맵과 교대 영역으로서 집중적으로 사용되는 것을 방지한다. 이 때 영역 이동에 의해 변경되는 교대 영역과 데이터 영역의 위치는 맵 영역에 기록되면, 변경되는 맵 영역의 위치는 별도의 고정된 블록에 기록하는데 플래시 메모리가 대용량화되면 영역 이동 기법에 따른 비용이 커지게 되는 문제가 있다.

* 본 논문은 2005년도 한국의국어대학교 학술연구비 지원에 의해서 연구되었음

3. κ -평준화(κ -leveling) 기법

본 논문에서는 κ -평준화 기법으로 효율적인 소거 횟수 평준화 기법을 제안하고 구현한다. κ -평준화 기법은 다음의 4가지의 주안점을 가지고 구현한다.

3.1 균등한 블록 할당

κ -평준화 기법은 소거 횟수 평준화를 위해 공평하게 빈 블록을 할당 하는 것이 중요하다. 그림 1에서 보는 바와 같이 공평하게 빈 블록을 할당하기 위해서 순차적으로 할당하는 방식(round-robin)을 사용한다. κ -평준화 기법은 빈 블록을 할당할 때 블록에 대한 정보를 알기 위해서 물리 블록 맵핑 테이블을 사용한다.

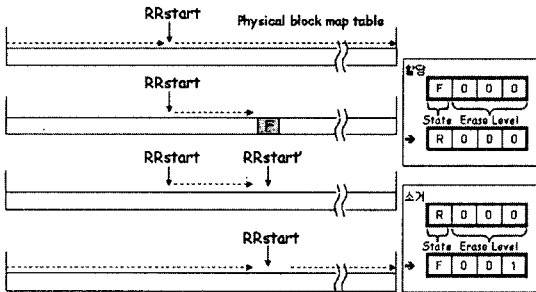


그림 1. 순차적인 블록 할당

그림 1에서 보는 바와 같이 물리 블록에 대한 정보에는 해당 블록의 상태(state)와 소거 횟수에 대한 레벨(erase level)이 있다. 빈 블록이 할당되면 상태는 빈 상태(free)에서 할당된 상태(reserved)로 바뀌게 된다. 반대로 블록이 소거되면 블록의 상태는 할당된 상태에서 빈 상태로 변하게 되며, 소거 횟수에 대한 레벨은 증가하게 된다. 빈 블록에 대한 공평한 할당은 블록에 대한 소거도 공평하게 한다. 빈 블록에 대한 할당과 할당된 블록의 소거에 따른 상태는 그림 2에서 보는 바와 같다.

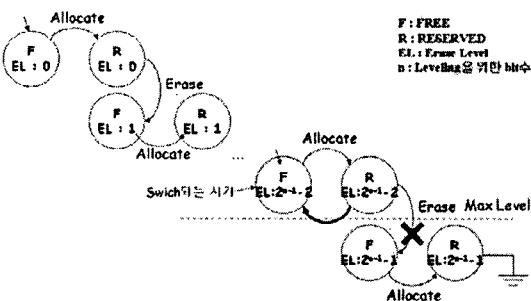


그림 2. 블록의 소거 레벨 상태도

3.2 소거 횟수 레벨

κ -평준화 기법은 각 블록에 대한 소거 횟수를 가지는 것이 아니기 때문에 소거 횟수 편차값(κ)에 따른 레벨을 가지게 된다. 소거 횟수 레벨은 소거 레벨의 비트수에 따라 변하게 되며 각 블록들의 소거 횟수에 대한 편차값을 뜻한다. 그림 2에서 최대 레벨(max level) 상태의 블록은 최소 레벨(min level) 상태의 블록과 교체되며 최대 레벨을 넘어서게 되면 상태에 대한 명확성을 잃게 되므로 소거 연산 시에는 이전 상태로 돌아가게 한다.

블록에 대한 소거 레벨이 상태에 따라 증가하게 되면 전체 레벨을 낮추는 작업이 필요하다. 이는 전체 블록이 어떤 동일

한 레벨 이상이 되면 전체 소거 횟수 레벨을 낮추어 최대 레벨(max level)로 상승하는 것을 막는다.

3.3 소거 레벨에 따른 블록의 교체

블록의 레벨이 최대이면서 비어있는 블록을 바쁜 블록(hot block)이라고 하며 레벨이 최소이며 할당된 블록을 수면 블록(cold block)이라고 한다. 바쁜 블록은 소거 연산이 많이 이루어져 높은 레벨이므로 빈 블록으로 할당하지 않고, 수면 블록과 교체하게 한다(forced-erase). 수면 블록은 할당된 블록 중에서 소거 레벨이 최소인 블록으로써 소거에 대한 연산이 드문 블록(read-only)이고 대기큐에 쌓이게 되며, 바쁜 블록을 만나게 되면 수면 블록은 깨어나서(wake-up) 바쁜 블록과 교체되어 바쁜 블록의 소거 가능성이 낮아진다. 교체작업을 하게 되면 강압적으로 소거 연산(forced-erase)이 이루어지므로 추가적인 비용을 초래하며, 수면 블록의 대기큐 크기가 비용의 크기를 결정한다. 블록에 대한 교체 작업은 특정 블록에 대한 소거 연산의 집중을 막아서 소거 횟수 평준화를 하게 한다.

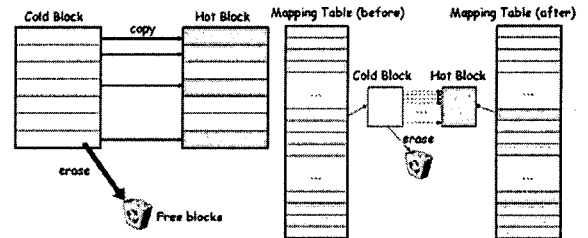


그림 3. 소거 레벨에 따른 블록의 교체

바쁜 블록과 수면 블록과의 교체는 그림 3에서 보는 바와 같이 수면 블록에 대한 내용을 바쁜 블록에 복사(copy)하게 되며 수면 블록은 소거하게 된다(switch). 또한 각 해당 블록의 맵핑 테이블도 바꾸어 준다. 하지만 FTL(Flash Translation Layer) 알고리즘에 따라서 블록에 대한 정보가 다르게 되므로 블록의 교체 작업도 달라질 수 있다.

복사 기법을 사용하는 M-Systems의 FMAX에서 각 블록은 원본 데이터 블록(data block)과 해당 블록에 대한 복사 블록(mirroring block)으로 구성된다. κ -평준화 기법을 적용한 FMAX는 원본 데이터 블록과 복사 블록을 합병 연산을 통해서 최신 정보만 바쁜 블록에 복사한다. 로그 블록 기법(Log Block Scheme)의 경우에는 데이터 블록(data block)과 로그 블록(log block)으로 블록이 구성된다. 로그 블록 기법 역시 데이터 블록과 로그 블록의 합병 연산을 통해서 바쁜 블록에 복사하게 되며 데이터 블록과 로그 블록은 소거하게 된다.

3.4 정보의 유지

블록에 대한 정보를 가지고 있는 물리적인 블록 테이블(physical block table)은 램에 기록되어 정보의 손실이 발생할 수 있다. 정보의 손실은 블록에 대한 소거 횟수를 가지고 있지 않으므로 이후에 소거 횟수 평준화를 할 수 없게 된다. 따라서 물리적인 블록 테이블에 대한 정보는 정보 블록(information block)에 상주시키는 방법을 이용하며, 각 블록의 κ 값은 그 블록의 0번째 섹터의 여유 공간(spare area)에 저장한다. 이렇듯 두 가지 방법을 두는 이유는 정보의 안정성을 높이기 위한 것이며 갱신하는 시기는 정보 블록의 경우에는 전체 블록의 소거 레벨을 낮추는 과정(down-leveling)에서 갱신을 하게 되며 한 블록에만 정보를 쓰는 것을 막는다. 두 번째 각 블록의 여유 공간에는 소거 연산이 수행될 경우에 소거 레벨을 기록한다. 두 경우에 대해서 정보의 일관성(consistency) 문제에서는

두 정보 중 낮은 값을 선택하게 하여 최신 정보를 알아낸다.

표 1. 테스트 작업의 구성

로그파일 형태	FAT/NTFS file system JPG, MP3 순차적 데이터 랜덤형 데이터(JPG/MP3)
테스트 FTL 알고리즘	로그 블록 기법, FMAX
플래시 메모리 크기	223,027,200 Bytes(13,200 blocks)
소거 레벨	3 bit(K: 8개 레벨)

4. 성능 평가

소거 횟수 평준화 기법의 성능을 비교 평가하기 위하여 기존의 순차적으로 빈 블록을 할당하는 방법과 K-평준화 기법의 결과를 비교하였다. 또한 K-평준화 기법과 순차적 할당에 대한 소거 횟수 평준화 기법을 비교하기 위해 표 1과 같은 테스트 작업을 수행하였다. 플래시 메모리는 13,200개의 블록을 사용하였으며 로그 파일에 대한 것은 FAT 시스템과 NTFS 시스템으로 나누어서 순차적인 데이터와 임의의 데이터에 대한 I/O를 수행하여 실제 물리적인 메모리에 대해서 각 블록 당 소거 횟수를 살펴보고 순차적 할당 방법과 성능의 차이를 테스트하였다.

표 2. K-평준화 로그 블록 기법 성능평가

파일시스템	평준화기법	로그파일	소거총합	소거평균	표준편차
FAT	라운드 로빈	랜덤	316,434	23.8	6.33
		JPG	359,900	27.3	8.91
		MP3	260,318	19.1	3.03
	K-평준화	랜덤	326,361	24.7	1.70
		JPG	378,327	28.7	1.82
		MP3	271,740	20.6	1.90
NTFS	라운드 로빈	랜덤	283,996	21.5	3.61
		JPG	258,056	19.5	2.56
		MP3	258,056	35.2	4.65
	K-평준화	랜덤	294,093	22.3	1.84
		JPG	352,350	26.7	2.11
		MP3	267,650	20.3	2.05

표 3. K-평준화 FMAX 성능평가

파일시스템	평준화기법	로그파일	총합	평균	표준편차
FAT	라운드 로빈	랜덤	445,332	33.7	10.79
		JPG	447,362	33.9	14.77
		MP3	468,462	35.5	7.04
	K-평준화	랜덤	485,494	36.8	0.67
		JPG	496,732	37.6	0.76
		MP3	497,368	37.7	0.77
NTFS	라운드 로빈	랜덤	452,478	34.3	6.48
		JPG	467,712	35.4	4.25
		MP3	465,292	35.2	4.65
	K-평준화	랜덤	480,075	36.4	0.54
		JPG	491,718	37.3	0.69
		MP3	485,502	36.8	0.66

표 2와 표 3은 로그 블록 기법과 FMAX에 대한 각각의 순차적 할당 방법 및 K-평준화 기법에 대한 소거 연산의 성능평가를 표로 나타낸 것이다. 순차적 할당 방식보다는 K-평준화 기법의 표준편차들이 낮아 K-평준화 기법이 우수하다는 것을 알 수 있다. 하지만 그림 4에서도 알 수 있듯이 블록의 교체로 인한 강제적 소거로 약 3~5%정도의 성능 저하가 나타나며 소거 횟수 평준화는 향상되는 결과를 알 수 있다.

그림 5와 그림 6은 로그 블록 기법과 FMAX에서 NTFS 파일 시스템의 임의 파일에 대한 로그 파일을 실험한 결과로써

왼쪽은 라운드 로빈에서 오른쪽은 K-평준화 기법에서의 각 블록에 대한 소거 횟수를 나타낸 것이다. 그림에서 보는 바와 같이 각 블록에 대한 소거 횟수가 K(단계에 대한 편차)값 이내에 분포되어 있는 것을 알 수 있다.

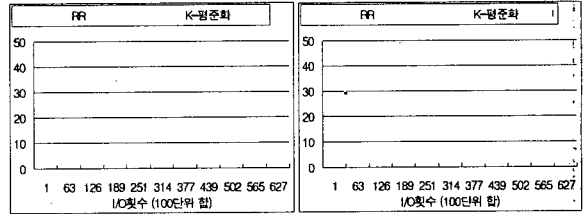


그림 4. NTFS 랜덤형 로그 파일에 대한 소요시간 누적

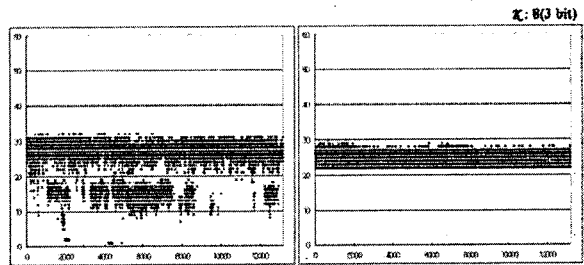


그림 5. 블록 당 소거횟수(로그 블록 기법/NTFS/랜덤형)

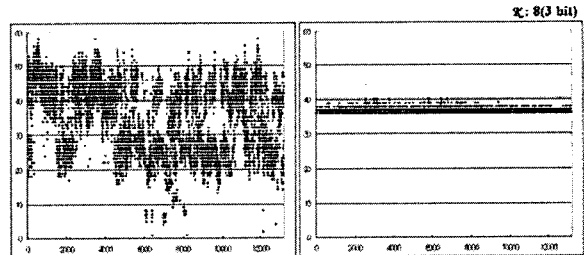


그림 6. 블록 당 소거횟수(FMAX/NTFS/랜덤형)

5. 결론

플래시 메모리는 각 블록에 대해 소거 횟수의 한계를 가지고 있기 때문에 특정 블록에 대해서 소거 연산이 집중이 되면 수명이 단축하게 되므로 플래시 메모리에 대한 안정적인 사용이 어렵게 된다. 따라서 플래시 메모리에 대한 소거 횟수 평준화 기법이 필수적이며 플래시 메모리의 대용량화에 따른 전체 블록의 계산 비용을 최소화하고 소거 횟수에 대한 정보를 적은 공간을 이용하여 관리하는 기법이 필요하다. 본 논문은 전체 블록에 대한 소거 횟수 정보에 레벨을 두어 공간을 적게 차지한다. 또한 제한된 기간 내에 교체 작업을 수행하여 교체 작업에 대한 비용을 최소화하며 효과적인 소거 횟수 평준화 기능을 제공한다.

[참고문헌]

[1] Young Hyun Bae, "An Efficient Wear-leveling Scheme for Flash Memory File System", 한국정보과학회 가을 학술발표논문집, 2004.
 [2] M. Wu, and W.Zwaenepoel, "eNvy: A Non-Volatile, Main Memory Storage System," In Proceedings of the 6th International Conference on Architectural Support for Programming Languages and Operating Systems(ASPLOS-6), pp. 86-97, Oct. 1994.