
Wear-leveling 향상을 위한 플래시 메모리의 그룹단위 관리 방법

장시웅* · 김영주** · 유윤식***

A group based management method of flash memory for enhancing wear-leveling

Si-Woong Jang* · Young-Ju Kim** · Yun-Sik Yu***

본 연구는 지식경제부 및 정보통신연구진흥원의 지원을 받아 수행되었음.
(09-산업원천, IT특화연구소 : "부산IT융합부품연구소" 설립 및 운영)

요 약

플래시 메모리는 데이터 변경 시에 덮어쓰기가 되지 않아 해당 데이터를 새로운 영역에 쓰기 이전에 존재하던 데이터는 무효화시켜야 하고 가비지 컬렉션 시 소거해야 한다. 플래시 메모리의 빠른 발전속도에 힘입어 플래시 메모리의 용량은 급속도로 증가하고 있다. 플래시 메모리의 용량이 대용량화됨에 따라 소거대상 블록을 선택할 때 전체 플래시 메모리를 대상으로 탐색을 실시하면 CPU의 수행시간이 많이 증가하는 문제점이 있다. 이 단점을 개선하기 위해 플래시 메모리를 그룹으로 분할하여 관리하고 소거대상 블록을 선택할 때 해당 그룹을 대상으로 탐색을 수행한다. 접근 지역성을 가지는 작업부하에 대해 hot 데이터는 hot 그룹에 배치하고, cold 데이터는 cold 그룹에 배치하여 그룹 내에서의 wear-leveling을 개선하고, 주기적으로 hot 그룹의 역할과 cold 그룹의 역할을 교환하여 전체 플래시 메모리의 wear-leveling과 성능을 개선하였다.

ABSTRACT

Since flash memory can not be overwritten on updating data, new data are updated in new area and old data should be invalidated and erased for garbage collection. With develop of flash memory technology, capacity of flash memory is rapidly increasing. It increases rapidly execution time of CPU to search an entire flash memory of large capacity when choosing the block to erase in garbage collection. To solve the problem that is increasing execution time of CPU, flash memory is partitioned into several groups, the block to erase in garbage collection is searched within the corresponding group. In workload of access locality, we enhanced wear-leveling within group by allocating hot data to hot group and cold data to cold group respectively and enhanced wear-leveling among groups by exchanging periodically hot group and cold group.

키워드

Flash File System, Wear-leveling, Garbage Collection, Database, Flash Memory

* 동의대학교 컴퓨터과학과/부산 IT융합 부품연구소

접수일자 2008. 08. 21

** 신라대학교 컴퓨터공학과

*** 동의대학교 물리학과/부산 IT융합 부품연구소

I. 서 론

플래시 메모리 기술의 빠른 발전은 플래시 메모리의 대용량화를 가능하게 했으며, 이에 따라 플래시 메모리는 PDA와 같은 임베디드 시스템뿐만 아니라 일반 컴퓨터 환경에 필요한 요건을 만족시키고 있다[1-2]. 플래시 메모리가 대형화됨에 따라 플래시 파일시스템에 관한 많은 연구가 진행되고 있다[3-7]. 플래시 메모리는 디스크와는 달리 덮어쓰기가 불가능하여 데이터가 변경되면 변경된 데이터에 해당하는 이전 영역은 쓸모없게 되고, 새로운 데이터는 새로운 영역에 갱신된다.

데이터가 빈번히 변경되면 새로운 영역을 확보하기 위해 가비지 컬렉션을 통하여 새로운 영역을 확보하게 된다. 데이터를 읽고 쓰는 것은 페이지 단위로 수행하나 데이터를 지우는 연산은 블록단위로 수행된다[8]. 데이터에 대한 가비지 컬렉션을 수행할 때 유효한 페이지는 새로운 블록으로 이동한 후 해당 데이터 블록을 지우는데, 이때 지움(erase) 연산을 수행할 수 있는 회수는 플래시 메모리의 특성에 의해 한정된 회수(약 100,000회)로 제한을 받게 되므로 플래시 메모리의 모든 블록은 고르게 쓰여 지고 지워져서 균등한 wear-leveling이 제공되어야 한다. 플래시 메모리의 용량이 대용량화됨에 따라 소거대상 블록을 선택할 때 전체 플래시 메모리를 대상으로 탐색을 실시하면 CPU의 수행시간이 많이 증가하는 문제점이 있다. 이 단점을 개선하기 위해 플래시 메모리를 그룹으로 분할하여 관리하고 소거대상 블록을 선택할 때 해당 그룹을 대상으로 탐색을 수행하는 방법이 제안되었다[10]. [10]에서는 단순히 그룹으로 분할하여 CPU의 수행 시간을 줄였지만 접근지역성을 가지는 작업 부하에 대해서는 고려하지 않았다.

따라서, 본 논문에서는 접근 지역성을 가지는 작업 부하에 대해 hot 데이터는 hot 그룹에 배치하고, cold 데이터는 cold 그룹에 배치하여 그룹 내에서의 wear-leveling을 개선하고, 주기적으로 hot 그룹의 역할과 cold 그룹의 역할을 교환하여 전체 플래시 메모리의 wear-leveling과 성능을 개선하는 CB-MG (Cost-Benefit between Multi Group) 방법을 제안하고 시뮬레이션을 통해 wear-leveling과 성능을 비교 분석하였다.

II. 플래시 메모리 관리 방법

플래시 메모리는 여러 개의 블록으로 구성되며 하나의 블록은 일정수의 페이지로 나뉘어 관리된다. 한 페이지의 크기는 하드 디스크의 최소 단위인 섹터의 배수로 구성된다. 플래시 메모리는 읽기와 쓰기 연산을 페이지 단위로 수행한다. 한편, 쓰기 연산은 지워진 영역에만 가능하므로 데이터를 쓰기 위해서는 해당 영역을 미리 지워야하는데 지움 연산은 블록 단위로 수행된다.

플래시 메모리는 덮어쓰기(overwrite)가 되지 않으므로 데이터를 갱신하기 위해서는 갱신할 데이터에 해당하는 페이지를 무효화시킨 후 갱신된 데이터를 새로운 블록의 페이지에 써야 한다. 많은 데이터에 대해 갱신 연산이 발생하면 무효화된 데이터가 많아져서 유효한 블록이 소진되게 된다. 따라서 자유블록의 수가 임계치 이하로 떨어지면 가비지 컬렉션 정책에 의해 특정 블록을 선정하여 유효 페이지를 새로운 페이지를 할당 받아 이동한 후 지움 연산을 수행한다.

2.1 가비지 컬렉션 정책

가비지 컬렉션 정책에는 무효 페이지가 가장 많은 블록을 선정하는 Greedy 알고리즘과 블록의 age와 이용율을 고려한 Cost benefit 알고리즘이 있다. Cost benefit 알고리즘은 블록 이용율(u)과 age를 고려하여 $\frac{age(1-u)}{2u}$ 를 최대로 하는 블록을 선정[8]하여 지움 연산을 수행한다. 이 방법을 이용하면 cold 블록들은 hot 블록보다 낮은 이용율을 가짐에도 불구하고 age가 클 경우에는 지움 연산을 위한 블록으로 선정된다.

종래의 Cost benefit 방법은 균등한 wear-leveling을 제공하지만 접근 지역성을 가지는 작업부하에서는 Greedy 방법처럼 좋지 못한 성능을 제공한다.

2.2 CB-S 방법

접근 지역성을 가지는 데이터에 대해 cost-benefit 방법의 단점을 개선하기 위해 hot 데이터와 cold 데이터를 분리하여 관리하는 CB-S (Cost Benefit with Separation) 방법이 제안되었다. 이 방법은 hot 데이터와 cold 데이터를 분리하여 관리하는 Cost benefit 방법으로 하나의 논리적 블록에는 hot 데이터의 페이지와 cold 데이터의

페이지가 공존하는 형태로 존재한다. 그러나 논리적 페이지 데이터들을 물리적 블록에 할당할 때는 hot 페이지 데이터들은 hot 페이지만 포함하는 물리적 블록에 저장하고, cold 페이지 데이터들은 cold 페이지만 포함하는 물리적 블록에 저장한다. 하나의 물리적 블록에 hot 페이지만 포함하는 블록과 cold 페이지만 포함하는 블록이 공존하므로 wear-leveling을 저하시키는 결과가 된다.

2.3 CB-MB 방법

hot 데이터와 cold 데이터를 서로 다른 블록에 분할하여 저장하는 CB-MB[11]는 hot 데이터와 cold 데이터를 분리하여 비용효율적인 방법으로 관리되 hot 데이터의 페이지와 cold 데이터의 페이지를 서로 다른 물리적 블록에 나누어 저장하고 관리하는 방법으로 hot 데이터의 페이지와 cold 데이터의 페이지가 공존하는 형태로 존재한다. 그러나 논리적 페이지 데이터들을 물리적 블록에 할당할 때는 hot 페이지 데이터들은 hot 블록만 포함하는 물리적 블록에 저장하고, cold 페이지 데이터들은 cold 블록만 포함하는 물리적 블록에 저장한 후, 일정한 주기로 hot 블록과 cold 블록의 역할을 교환되도록 하여 블록 간에 균등한 부하를 받도록 되어 있다. 이 방법에서는 hot 블록만 포함하는 블록과 cold 블록만 포함하는 블록을 둬으로써 블록내에서는 균등한 wear-leveling을 제공할 수 있으나 블록간에는 균등한 wear-leveling을 제공하지 못한다. 따라서 주기적으로 블록을 교환함으로써 전체적으로 균등한 wear-leveling을 제공할 수 있지만 hot 블록과 cold 블록간의 데이터 양이 많이 차이 나면 플래시 메모리의 낭비를 초래한다.

III. 플래시 메모리의 그룹 단위 관리 방법

플래시 메모리의 그룹 단위 관리 방법(CB-MG)은 hot 데이터와 cold 데이터를 분리하여 비용효율적인 방법으로 관리되 hot 데이터의 페이지와 cold 데이터의 페이지를 서로 다른 물리적 그룹에 나누어 저장하고 관리하는 방법이다.

그림 1의 (a)와 같이 하나의 논리적 블록에는 hot 데이터의 페이지와 cold 데이터의 페이지가 공존하는 형태로 존재한다. 그러나 논리적 페이지 데이터들을 물리적 그룹에 할당할 때는 그림 1의 (b)와 같이 hot 페이지 데이터들은 hot 블록만 포함하는 물리적 그룹에 저장하고, cold 페이지 데이터들은 cold 블록만 포함하는 물리적 그룹에 저장한다. 한편, hot 블록만 포함하는 그룹과 cold 블록만 포함하는 그룹 모두 균등한 wear-leveling을 제공할 수 있지만 hot 블록만 포함하는 그룹이 빨리 소진되는 현상이 나타날 것이다. 따라서 본 논문에서는 그림 2의 알고리즘에서와 같이 일정한 주기로 hot 그룹과 cold 그룹의 역할을 바꾸도록 하였다. 실험 결과, 그룹 교환 주기를 짧게 하는 경우 wear-leveling과 성능이 좋지 않은 결과를 나타냈으나 그룹 교환 주기를 평균 write 회수가 20이상 이 되도록 잡았을 때는 균등한 wear-leveling과 좋은 성능을 보였다.

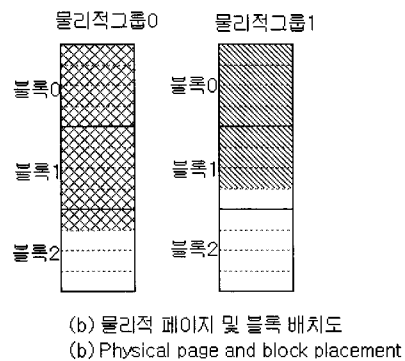
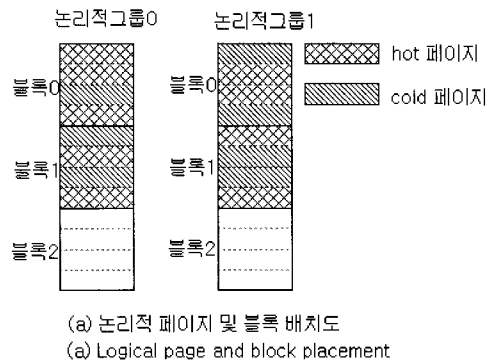


그림 1. 페이지 및 블록 할당
Fig. 1 Allocation of page and block

```

if (the data == hot data) then {
    allocate a page in hot group for the data;
    write the data to the allocated page;
}
else {
    allocate a page in cold group for the data;
    write the data to the allocated page;
}
if (group exchange period is timeout) then {
    choose a cold group block to change its status;
    set the cold group block to hot group;
    choose a hot group block to change its status;
    set the hot group block to cold group;
}
    
```

그림 2. CB-MG의 블록관리 알고리즘
Fig 2. Block Management Algorithm of CB-MG

IV. 실험

본 논문에서 제안한 플래시 메모리의 그룹 단위 관리 방법의 성능 및 wear-leveling을 CB-S 기법과 비교하기 위해 시뮬레이션을 수행하였다.

4.1 실험 방법

실험은 1개 블록의 크기에 해당하는 논리적 페이지에 대한 페이지 번호를 발생시키되 발생된 홀수 페이지 번호는 번호가 하나 작은 짝수 페이지로 바꾸어 논리적인 홀수 페이지는 없도록 하여 1개의 물리적 블록에 할당되는 물리적인 페이지의 수는 물리적 블록의 절반만 차지하도록 하였다. 이때 새로운 물리적 블록이 일정개수 (1/100) 이하가 되면 가비지 컬렉션이 수행되도록 하여 dirty 페이지가 많은 블록들이 클리어 되도록 하였다. 이와 같은 방법으로 작업부하를 10000회 연속 발생시켜 실험을 수행하였다. 플래시 메모리의 그룹 단위 관리 방법의 성능을 평가하기 위해 플래시 메모리를 10개의 그룹으로 나눈 후, 발생된 페이지 번호에 따라 그룹으로 나뉘어 관리되도록 하였다.

실험 수행시 사용된 파라미터는 표 1과 같다. 표 1의 실험 파라미터는 블록의 크기가 크고 대용량인 플래시 메모리의 사양으로 사용되는 파라미터이다.

본 논문에서 제안하는 플래시 메모리의 그룹 단위 관리 방법(CB-MG)의 성능을 상대적으로 평가하기 위해

기존에 제안된 CB-S 방법의 성능을 함께 평가하여 비교한다. 성능비교시 CB-S 및 플래시 메모리의 그룹 단위 관리 방법의 CPU 수행시간을 고려하였다. 본 실험은 플래시 메모리를 사용하는 환경을 PDA와 같은 임베디드 시스템을 고려하므로 CPU의 수행시간을 PC 수행시간의 5배로 환산하여 계산하였다.

표 1. 실험 파라미터
Table 1. Experimental parameters

특성	값	특성	값
읽기	0.025ms	페이지 크기	2 KBytes
쓰기	0.200ms	블록 크기	64 KBytes
지우기	1.5ms	뱅크 크기	8 GBytes

작업부하는 [8]에 있는 것과 유사한 방법으로 지역성을 가지는 작업부하를 2개의 등급으로 나누어 실험한다. 참조의 지역성을 표시하기 위해 x/y 로 표현하는데, 이것은 $y\%$ 의 데이터에 $x\%$ 의 접근이 이루어지는 것을 의미한다. 따라서, 90/10의 작업부하는 10%의 데이터에 90%의 참조가 이루어지고, 90%의 데이터에는 10%의 참조가 발생하는 것을 의미한다.

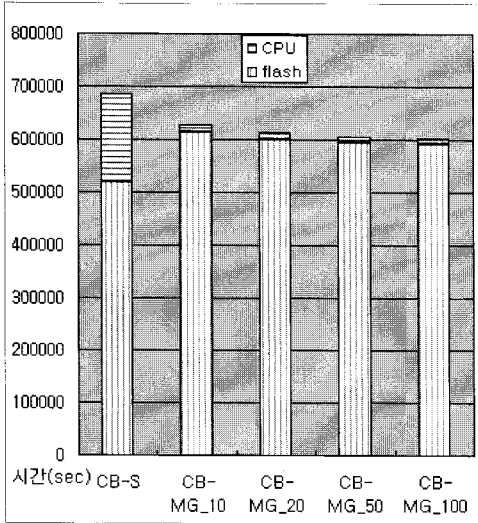
4.2 성능 평가 결과

4.1절의 실험 방법으로 CB-S 및 CB-MG에 대해 실험한 전체 수행시간과 wear-leveling을 비교하여 분석하면 각각 그림 3 및 표 2와 같다.

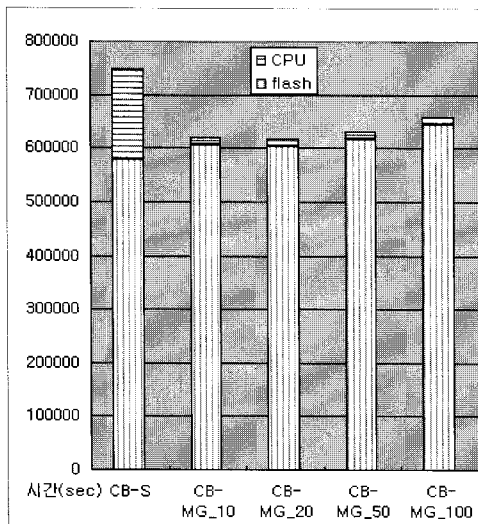
그림 3에서 보는 것처럼 CPU 수행시간을 포함한 CB-MG의 성능은 CB-S에 비해 좋은 성능을 나타내는 것으로 보인다. 이는 그림 2에서 보는 것처럼 CB-MG의 flash memory time은 CB-S보다 오히려 다소 증가하지만 CB-S는 가비지 컬렉션시 전체 플래시 메모리에서 대상 블록을 찾는 반면 CB-MG는 그룹내부에서 대상 블록을 찾기 때문에 CPU time이 1/7정도로 줄어들어 전체 수행시간은 상당히 줄어들 것으로 분석되었다.

접근 지역성이 높은 경우(90/10)에는 hot 데이터를 포함한 그룹과 cold 데이터를 포함한 그룹의 교환은 교환 주기가 10회에서 100회까지 교환 주기가 증가할수록 전체 시스템의 성능이 좋아지는 것으로 분석되었다. 접근 지역성이 비교적 낮은(80/20) 경우에는 하나의 그룹에

hot 데이터와 cold 데이터가 혼합되는 확률이 높아지므로 교환 주기가 중요한 의미를 가지지 않는 것으로 분석되었다.



(a) 접근지역성이 높은 경우 (90/10)
(a) Case of high access locality



(b) 접근지역성이 비교적 낮은 경우(80/20)
(b) Case of low access locality

그림 3. 그룹교환주기에 따른 수행시간의 비교
Fig 3. Comparison of execution time according to exchange period of group

접근 지역성이 높은 경우(90/10)에는 교환주기가 증가할수록 표준편차가 감소하여 균등한 wear-leveling을 제공한다. 그러나 CB-MG_100인 경우에는 표준편차는 적지만 전체 블록의 평균 지움연산 회수가 상대적으로 증가하는 문제점이 발생한다. 접근지역성이 낮은 (80/20) 경우에는 그룹 교환의 주기가 50까지는 표준편차와 평균 지움연산 회수 모두 감소하지만 100이상인 경우에는 표준편차와 평균 지움연산 회수가 모두 증가하는 현상이 발생한다.

표 2. 그룹교환주기에 따른 wear-leveling의 비교
Table 2. Comparison of wear-leveling according to exchange period of group

	10/90	20/80
CB-S	22.36 (1152 ~ 1287)	14.22 (1196 ~ 1283)
CB-MG_10	9.11 (1277 ~ 1335)	5.44 (1269 ~ 1303)
CB-MG_20	6.41 (1259 ~ 1299)	4.54 (1271 ~ 1299)
CB-MG_50	4.53 (1248 ~ 1276)	3.88 (1272 ~ 1296)
CB-MG_100	3.41 (1269 ~ 1291)	4.13 (1396 ~ 1421)

V. 결론

본 논문은 hot 데이터와 cold 데이터를 분리하여 관리하는 Cost benefit 방법(CB-S)에서 플래시 메모리가 대용량일 경우 가비지 컬렉션시 소거대상 블록을 검색하는 CPU time이 많이 소모되는 것을 개선하기 위해 플래시 메모리를 그룹으로 분할하여 관리하는 방법을 적용하고, 지역성을 가지는 작업부하를 고려하여 hot 그룹과 cold 그룹을 주기적으로 교환하도록 함으로써 전체 플래시 메모리의 wear-leveling을 개선하였다. hot 데이터와 cold 데이터를 블록단위로 분리하여 저장하여 관리하는 Cost benefit 방법은 좋은 성능을 보이지만 균등하지 않은 wear-leveling을 제공하는 것이 단점으로 지적된 반면, 본 연구에서 제안한 CB-MG 방법은 접근지역성을 보이는 작업부하에 대해 CB-S 방법에 비해 좋은 성능을 보임과 동시에 균등한 wear-leveling을 보이는 것으로 나타났다.

다 본 논문에서 제안한 플래시 메모리의 그룹 단위 관리 방법은 플래시 메모리의 크기가 4GBytes 이상이며, CPU의 성능이 일반 PC보다 좋지 않은 임베디드 시스템에 적합할 것으로 판단된다.

참고문헌

[1] Mei-Ling Chiang and Ruei-Chuan Chang, "Cleaning Policies in Mobile Computers Using Flash Memory." Journal of Systems and Software. Vol.48, No.3, pp.213-231, 1999.

[2] Fred Dougllis et al. "Storage Alternatives for Mobile Computers," In OSDI, November 14-17, Monterey, California, USA.

[3] Jen-Wei Hsieh, Li-Pin Chang and Tei-Wei Kuo. "Efficient On-line Identification of Hot Data for Flash-Memory Management", In SAC, pp.838-842, 2005.

[4] Y. Ryu and K. Lee. Improvement of space utilization in nand flash memory storages. Lecture Notes in Computer Science, 3820:766-775, 2005.

[5] Chang, L, Kuo, T., and Lo, S., "Real-Time Garbage Collection for Flash-Memory Storage Systems of Real-Time Embedded Systems," In ACM Trans. on Embedded Computing Systems, November 2004, pp. 837-863

[6] Li-Pin Chang, Tei-Wei Kuo and Shi-Wu Lo. "A Real-Time Garbago Collection for Flash-Memory Storage Systems of Real-Time Embedded Systems." ACM Trans. in Embedded Computing Systems, Vol.3, No.4, pp.837-863, 2004.

[7] ERAN GAL AND, SIVAN TOLEDO, "Algorithms and Data Structures for Flash Memories", ACM Computing Surveys, Vol.37, Issue.2, pp.138-163, June, 2005.

[8] L.Z Han, Y.S Ryu, T.S Chung, M.H Lee, S.W Hong, "An Intelligent Garbage Collection Algorithm for Flash Memory Storages," Lecture Notes in Computer Science, 3980:1019-1027, 2006.

[9] Atsuo Kawaguchi, Shingo Nishioka, and Hiroshi

Motoda, "Flash Memory Based File System," Proceedings of USENIX Technical Conference, New Orleans, LA, pp.155-164, 1995.

[10] 박제호, "K 분할 기반 플래시 메모리 균등소거 방법론," 한국정보처리학회 논문지D, 제13-D권, 제3호, pp.377-382, 2006.

[11] 장시웅, "접근 지역성을 가지는 작업부하에서 플래시 파일시스템의 wear-leveling 향상 기법," 한국해양정보통신학회, 제12권, 제1호, pp.108-114, 2008.

저자소개

장 시 웅(Si-Woong Jang)

1996년~현재 동의대학교 컴퓨터과학과 교수
(제12권 제11호 참조)

김 영 주(Youngju Kim)

2000년~현재 신라대학교 컴퓨터공학과 교수
(제10권 제12호 참조)

유 윤 식(Yun-Sik Yu)



1977년 2월 부산대학교 물리학과
(이학사)

1979년 2월 부산대학교 물리학과
(이학석사)

1990년 2월 부산대학교 물리학과(이학박사)

1983년 3월~현재 동의대학교 물리학과 교수

2008년 6월~현재 부산IT융합부품연구소 소장

※관심분야: IT융합, 광통신