

SSD를 위한 최적화 파일시스템

박 제 호[†]

[†]단국대학교 컴퓨터학과

An Optimized File System for SSD

Je-Ho Park[†]

[†]Dankook University, Computer Science

ABSTRACT

Recently increasing application of flash memory in mobile and ubiquitous related devices is due to its non-volatility, fast response time, shock resistance and low power consumption. Following this trend, SSD(Solid State Disk) using multiple flash chips, instead of hard-drive based storage system, started to widely used for its advantageous features. However, flash memory based storage subsystem should resolve the performance bottleneck for writing in perspective of speed and lifetime according to its disadvantageous physical property. In order to provide tangible performance, solutions are studied in aspect of reclaiming of invalid regions by decreasing the number of erasures and distributing the erasures uniformly over the whole memory space as much as possible. In this paper, we study flash memory recycling algorithms with multiple management units and demonstrate that the proposed algorithm provides feasible performance. The proposed method utilizes the partitions of the memory space by utilizing threshold values and reconfigures the management units if necessary. The performance of the proposed policies is evaluated through a number of simulation based experiments.

Key Words : Flash Memory File System, Block Cleaning, Even Wear-Leveling, SSD

1. 서 론

시장 확대와 기술 발전으로 SLC(Single-level Cell) 구조인 경우 크기가 작은 경우는 64MByte, 확장된 크기에서는 8GByte, MLC(Multi-level Cell)는 16GByte의 용량을 가지는 NAND 플래시 메모리가 대량생산 가능하다. 이에 따라 NAND 플래시 저장매체를 응용한 제품 시장의 크기는 이미 초기 시장을 거쳐 확산 및 저장매체의 새로운 매체 시대를 열고 있다고 보아야 할 것이다.

초기 응용단계에서는 단일 플래시 메모리칩을 휴대폰, MP3플레이어, PMP(Portable Multimedia Player), 네비게이션, 디지털카메라 등 소형 모바일 기기의 보조 저장매체로서 사용하였으나, 지금은 다수의 플래시 메모리를 통합해 하나의 저장매체 시스템으로 사용하는

SSD(Solid State Disk)로 구성하여 소형 컴퓨터 또는 유비쿼터스 시스템의 저장매체로 이용하고 있다[1-3]. 이러한 장치들은 감소된 부피, 외부 충격에 대한 내구성, 낮은 소비전력, 빠른 응답시간이라는 플래시 메모리의 우수성을 효과적으로 응용한 제품들이다. 플래시 메모리칩을 이용한 저장매체는 플래시 메모리 드라이버, 플래시 변환 계층(FTL, Flash Translation Layer), 플래시 메모리 파일 시스템 등으로 구성되는 다계층 소프트웨어 구조의 지원이 필요하다[4].

기존의 저장매체와는 달리, 플래시 메모리는 같은 주소에 한번에 자료갱신을 할 수 없는 특성을 가지고 있다. 또한, 저장된 자료의 소거를 위해서는 소거가 필요한 부분뿐 아니라 유효한 데이터가 포함된 미리 크기가 설정된 영역(블록)을 일시에 소거해야 한다. 한 영역에 대한 소거횟수는 제한되어 있어, 전체 메모리 영역에서 특정 부분에 소거가 집중될 경우, 플래시 메모리는 장애를 일으킬 수 있는 단점을 가지고 있어,

[†]E-mail : dk_jhpark@dankook.ac.kr

FTL은 전체 영역에 대한 소거를 가능한 균형 있게 실시하는 균등소거 알고리즘을 포함하고 있다. 균등소거 알고리즘의 원활한 성능을 위해, 전체 메모리 영역을 고려하여 균등소거 알고리즘을 실행하는 것이 아니라, 분할된 하부영역에 독립적으로 균등소거 알고리즘을 실행하여, 성능 및 속도에 대한 개선을 모색 할 수 있다.

이 논문에서는 이러한 하부영역의 분할이 가져올 수 있는 특성을 분석하고, 이를 해결할 수 있는 방법을 제시하고자 한다. 또한, 성능 분석을 위해 수행한 실험의 결과를 예시한다.

2. 플래시 메모리

플래시 메모리는 읽기/쓰기 단위가 되는 2/4 Kbyte 크기의 페이지들로 구성되는 64/128 Kbyte 크기의 블록들로 구성되며, 관리영역은 블록의 집합으로 구성되는 것을 Fig. 1에 보였다. 플래시 메모리의 기본적인 동작과 특성에 따른 필요 소요시간은 Table 1에 보여지고 있다. 복사 동작은 특정 페이지를 다른 블록에 있는 페이지로 복사할 때 사용되며, 일부 MLC NAND 플래시 메모리를 제외한 대부분의 NAND 플래시 메모리 제품에서 지원된다.

플래시 메모리의 상대적으로 느린 쓰기 및 복사의 소요시간, 10,000~100,000 번의 갱신한도라는 물리적 특성은 효율적인 플래시 메모리의 응용에 제한적인 문제가 될 수 있다. 특히, 동일 페이지에 자료 갱신을 허락하지 않는 특성은 플래시 메모리 효율성을 감소시켜 플래시 메모리의 다양한 응용에 부정적인 영향을 미칠 수 있다. 페이지 단위 소거는 불가능하고, 블록단위 소거만 가능하다는 특성은 기존의 메모리에서 사용하는 방법과는 다른 갱신(update)과정을 필요로 한다.

갱신을 위한 방법론 중 하나인 In-place 갱신은 갱신 블록에 포함된 유효한 페이지들을 시스템 버퍼에 임시로 저장 하고, 해당 블록을 소거한 후, 버퍼에 저장된 페이지들과 새로 갱신되는 페이지를 함께 동일 블록에 쓰기를 통해 갱신 작업을 수행한다. 따라서, 매 갱신마다 소요 시간이 긴 소거 작업이 수행되어야 하므로 플래시 메모리 저장 매체의 효율성에 영향을 미친다.

블록의 갱신 횟수가 한도에 도달할 경우, 해당 블록은 손상되어, 해당 영역에 대한 쓰기는 장애를 발생시켜, 복구 절차를 거쳐야 하기 때문에 적절한 반응 속도를 보장할 수가 없게 된다. 이를 방지하기 위해 발생하는 갱신들을 전체 플래시 메모리 공간에 균등하게 분포하는 방법을 필요로 한다. Non-in place 방법론은 갱신 과정에 포함된 읽기/쓰기 동작을 분리하여 해당 블

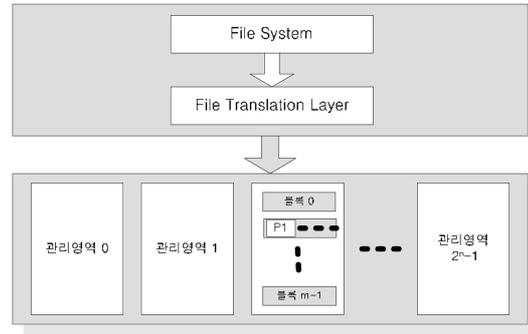


Fig. 1. Configuration of Flash Memory Storage.

Table 1. Characteristics of Flash Memory

동 작	단 위	특성값
읽 기	페이지	25 microsec
쓰 기	페이지	200~300 microsec
복 사	페이지	200~300 microsec
소 거	블 록	2000 microsec

록의 즉각적인 소거가 없이 쓰기 작업을 수행한다[5].

플래시 메모리 재활용에 사용되는 메모리 공간을 확보하기 위해서는, 갱신으로 인해 무효화된 페이지가 포함된 블록에서 유효한 페이지들을 다른 블록으로 이동시킨 뒤, 해당 블록은 후에 자유 공간이 필요 할 때, 블록 재활용(cleaning) 정책에 따라 순서적으로 소거를 한다. 블록 소거는 소요 시간이 클 뿐 아니라 전력 소모량이 다른 동작에 비해 많다. 블록 재활용 정책의 목적은 소거 횟수를 최소화하여 전력 소비와 반응 속도 측면에서 시스템 성능을 개선하는 동시에 일부 블록에 소거가 집중되는 것을 방지하여 전체 메모리 영역의 생명 주기를 연장하기 위해 균등소거(even wear leveling)를 유지하는 것이다. 균등소거 문제는 비결정 다항식(NP-Hard: Non-deterministic Polynomial Hard)문제로 증명되었다[6,7].

균등소거를 고려할 때, 플래시 메모리 공간을 두 개의 공간으로 분할을 한다: 정적 공간과 동적 공간. 정적 공간에는 플래시 메모리 운영에 필요한 자료 또는 파일들이 저장되며, 이러한 파일들은 한 번 저장되면 거의 갱신이 되지 않는 특성을 가지고 있다. 동적 공간은 갱신 횟수가 많은 사용자 파일들을 저장하게 된다. 정적 균등소거법(static wear leveling)은 정적 공간과 동적 공간을 모두 균등소거의 대상으로 하는 방법이며, 동적 공간만을 소거 대상으로 하는 것을 동적 균등소거법(dynamic wear leveling)이라고 한다[8].

균등소거 방법론은 또한 메모리 영역을 특정 사이즈의 관리영역(management unit)으로 나누어, 한 개의 관리영역 내부에서만 균등소거를 고려하는 방법과 관리영역의 구분 없이 전체 메모리 영역을 소거 대상으로 하는 방법으로 분리된다.

3. 균등소거를 위한 소거 대상 선택

3.1. 기존의 방법론

기본적으로 소거 대상 블록을 선택하는 방법은 무작위 선택(random selection)이나 특정 순서에 따라 순환하는 방법(round-robin)이 있다. 다른 방법론들은 블록의 속성값을 기반으로 전체 공간을 탐색한다. 탐색적 방법론은 대상 블록들 중에서 최대 (블록 페이지 수 - 유효 페이지 수)를 가지는 블록을 소거 대상으로 한다. 비용/편익 기간 방법론은 특정 블록에 대해 마지막 페이지 무효화 시간과 계산 수행까지의 시간을 age 라 정의하고, 해당 블록의 유효 자료 비율을 u 라고 할 때 최대 $(age * (1-u)) / (2 * u)$ 을 가지는 블록을 소거한다[4].

한 개의 관리영역에 포함된 페이지 수를 PageMUnit, 적어도 전체 메모리 공간에서 한 개 이상 유효 자료를 포함하는 블록들의 백분율을 UtilMUnit 이라고 정의하며, 한 개의 관리영역에 전역 탐색을 적용할 때, 최적값 탐색에 필요한 블록의 수 SearchMU 는 식 (1)과 같이 정의된다. 또한, 다수의 n 개의 관리 영역으로 구성되는 시스템 전체에 대한 검색 영역 SysSpace는 식 (2)와 같이 정의된다.

$$SearchMU = [PageMUnit * UtilMUnit] \quad (1)$$

$$SysSpace = \sum_{i=0}^{n-1} [SearchMU_i] \quad (2)$$

플래시 메모리 공간이 정적 공간과 동적 공간으로 분할되어, 갱신이 빈번한 동적 공간에 집중적으로 갱신이 수행될 경우, 동적 균등소거법은 전체 메모리 영역에 대한 균등소거의 불균형을 가져온다. 반대로 자료의 갱신이 거의 없는 정적 공간을 소거 대상 검색에 포함시키는 정적 균등소거법을 채택할 경우, 균등소거는 개선될 수 있지만, 검색 영역의 확장과 갱신이 되지 않는 페이지의 이동이 빈번하게 발생하여, 성능 개선에 대한 필요 비용은 높아지게 될 것이다. 이 문제를 효과적으로 접근하기 위해, 본 논문에서는 정적/동적 공간 개념과 관리 영역을 하나의 모델에 구현하는 것을 제안한다. 이 모델은 정적/동적 공간 개념은 논리적 파일 시

스템 계층이고, 관리 영역은 플래시 메모리의 물리적 계층이라는 점에 바탕을 두고 있다.

본 논문에서 제안하는 방법론은 두 단계로 이루어진다. 관리영역 내부에 적용되는 소거 대상 방법론은 검색 영역을 축소하여 기존 방법론이 필요로 하는 검색 비용을 감소시키고, 동시에 전역 검색이 아닌 집합 개념에 기반을 둔 방법론이다. 또한 이 방법론은 잠재적인 성능의 감소의 최소화를 통해 효율 대비 비용 효과를 최적화하는 것이다.

또한 제안하는 방법론은 논리적 관리영역 계층 구성을 위해, 서로 다른 관리영역간의 균등소거 정도의 차이가 현저해졌을 때, 관리영역의 논리적 구성을 재편성하여 관리영역 간 균등소거 분포를 최적화하는 것을 목표로 한다.

3.2. 관리영역 기반 소거

특정 관리영역에서의 소거 대상 선택은 대상블록 집합을 동일한 부분집합으로 분할한다. 이 때 최적분할을 위해 필요한 부분집합의 개수 K 값은 실험적으로 결정한다. 각 계층별 하부 탐색영역의 이상적인 크기 SetSize는 다음과 같이 정의된다.

$$SetSize = \left\lceil \frac{SearchMU}{K} \right\rceil \quad (3)$$

따라서, 소거 대상 블록을 선택할 때, 탐색영역은 SearchMU 보다 작은 하부영역 SetSize로 한정되고, 각 하부영역이 가지는 대표속성값을 토대로 탐색 대상 하부영역을 결정하게 된다. 관리영역 m 에 속하는 소거 대상 블록들은 K 개의 하부영역 $\{R_m(K), \dots, R_m(1)\}$ 로 분할된다. 각 하부영역에 속하는 블록들은 분할의 최저/최대한계치의 범주에 속하는 속성값을 가진다. 전역 검색 방법론에 사용되는 속성값을 하부영역 구성에도 같이 적용한다고 가정할 때, 최상위 대표 속성값에 상응하는 하부영역을 탐색 결과는 적용된 알고리즘에 따라 무작위 선택과 전역 탐색 기반 선택을 사용할 수 있다.

특정 관리영역에 대한 탐색결과는 전체메모리 영역을 탐색 대상으로 하는 방법론의 결과와는 차이가 있을 것으로 예상할 수 있다. 최적의 속성값을 가진 블록 소거는 소거 균등성을 개선한다는 점을 고려할 때, 전체 대상 블록 중에서 무작위 선택을 적용하는 것보다 하부 탐색영역 $R_m(K)$ 에서 무작위 선택을 하는 것이 개선된 효과를 가져올 것이다. 이는 하부영역 $R_m(K)$ 에 속하는 블록들은 높은 최적성을 가지고 있어, 전체 소거 대상 집합에서 최적성의 고려 없이 무작위 선택을 적용한 것보다 개선된 결과를 보여줄 것이다.

3.3. 하부영역의 분할

특정 관리영역의 소거 대상 페이지들을 속성값에 따라 K 개의 하부영역으로 분할할 때 기본이 되는 속성값 G 는 식 (4)와 같이 정의한다.

$$G = Page_{invalid} + Page_{unused} \quad (4)$$

따라서, $Pages_{Block}$ 를 한 블록에 속하는 페이지수라고 정의하면, G 는 최소 1, 최대 ($Pages_{Block} - 1$)의 영역에 속하게 된다. 이 구간을 K 개로 분할하면, 각기 하부영역은 최대 다음과 같은 속성값 구간(W)을 가지게 된다.

$$W = \left\lceil \frac{Pages_{Block} - 2}{K} \right\rceil \quad (5)$$

$$LowerB = 1 + (P - 1) * W \quad (6)$$

$$UpperB = LowerB + (W - 1) \quad (7)$$

각 하부영역의 대표속성값 P 는 1부터 K 까지의 값을 가지게 되며, 대표속성값이 P 인 하부영역은 식 각기 (6)과 (7)에서와 같이 최소속성값 $LowerB$ 와 최대속성값 $UpperB$ 를 가지게 된다. 특히, 하부 영역 대표속성값이 K 인 경우는 상한 임계값을 ($Pages_{Block} - 1$)로 정의한다.

3.4. 관리영역의 재구성

하나의 관리영역 m 에 속하는 페이지 집합 $ManUnit_m$ 는 다음과 같이 하부영역의 집합으로 나타낼 수 있다.

$$ManUnit_m = \bigcup_{i=0}^k R_m(i) \quad (8)$$

특히, $R_m(0)$ 은 해당 모든 페이지가 유효한 블록들의 집합을 나타낸다. 두 개의 서로 다른 관리영역 m 과 n 에 속하는 블록 중 우선적으로 소거의 대상이 블록 중 우선 대상이 되는 하부영역을 $R_m(i)$ 와 $R_n(j)$ 라고 할 때, 이 두 하부영역에 속하는 블록들은 각기 관리영역의 균등소거를 개선하는 측면에서 선택될 것이다. 만일, 관리영역 m 의 $R_m(i)$ 에 속하는 블록들의 속성값이 $R_m(i)$ 와 $R_n(j)$ 의 합집합에서도 우선적으로 소거 대상으로 선택된다면, $R_m(i)$ 에서 소거 대상을 선택하는 것은 해당 관리영역 뿐 아니라 관리영역 외부에 대한 균등소거에도 공헌할 것이다. 하지만, 합집합에서 선택의 낮은 우선순위를 가지는 $R_n(j)$ 은 해당 관리영역의 균등소거에는 공헌을 할 수 있지만, 전역적인 측면에서는 공헌도가 낮게 된다. 이를 보정하기 위해서는 관리영역 뿐 아니라, 전체 영역에서의 균등소거를 할 수 있는 블

록들을 대상으로 삼아야 할 것이다. 본 논문에서 제안하는 관리영역 간의 균등소거 보정을 방법론은 관리영역의 균등소거의 현저한 차이를 보이는 두 관리영역의 최우선 순위의 하부영역을 교체하여 관리영역의 구성을 재편성하는 것이다. 이를 통해 관리영역 내부의 균등소거를 유지할 뿐 아니라, 전체 메모리 영역의 균등소거를 개선하고자 하는 것이다. 본 논문에서 제안하는 플래시 메모리 관리의 유효성은 시뮬레이션 기반 실험을 통해 검증하였다.

4. 구현 사례 및 결과 분석

4.1. 구현사례

실험의 목표는 제안하는 플래시 메모리 기반 저장매체를 위한 파일시스템의 유효성을 실험을 통해 검증하는데 있다. 실험에 사용된 플래시 메모리 모델은 전형적인 모델을 구성하였으며, 전체 플래시 메모리 기반 저장매체의 모델링은 일반적 모델을 사용하여 구현하였다.

본 논문에서 사용한 시스템의 구성은 Table 2에 제시하였다. 특히, 메모리 유효데이터 비율은 저장매체의 성능에 크게 영향을 미치는 것으로 알려져 있다[3,8]. 성능 측면에서 병목현상이 나타나는 유효데이터 비율은 80%로 설정하였다.

Table 2. Configuration of Simulation System

시스템 속성	설정 값
메모리 크기	8 GByte
블록 크기	256 KByte
페이지 크기	4 Kbyte
관리영역 당 블록 수	2048 blocks
유효 데이터 비율	80 %

접근 빈도수가 높은 영역(Hot Access Area)과 사용 빈도수가 낮은 영역(Cold Access Area)을 설정하고, 접근 발생 시 접근의 종류를 조절하여 접근 집약성을 구현하였다. Hot Access 영역에 대한 페이지 접근을 전체 페이지 접근의 80%로 설정하였다. Hot Access 영역은 특정 관리영역들을 이용해 설정하였다[9].

관리영역 내부에서 소거 대상 블록을 선택하는 알고리즘으로 무작위 선택(Random), 탐욕적 선택(Greedy)을 구현하고, 또한 제안하는 분할 기반(PSetBased) 방법론을 구현하여 유효성 분석을 위하여 사용하였다. 분할 기반 방법론에 사용된 분할 수는 메모리 모듈 내부

에 대한 기존의 성능 분석을 통해 결정된 값 6을 사용하였다. 관리영역 간의 하부영역 교환을 위해서는 최상의 균등소거를 보이는 관리영역과 최하의 균등소거를 보이는 관리영역을 선택하였다.

4.2. 성능 분석을 위한 실험

Fig. 2는 균등소거율을 예시하며, 범례 PSetBased는 관리영역 간의 균등소거의 차이가 현저할 때, 관리영역의 구성을 새로이 하였을 때의 균등소거를 보여준다. 균등소거율이 높은 관리영역은 Hot Access 영역에 속하는 데이터가 많은 관리영역을 나타낸다. 무작위 선택 방법과 탐욕적 선택 방법은 관리영역의 구성이 고정되어, 균등소거율에서 분할 기반 방법론의 성능보다 떨어지는 것을 볼 수 있다.

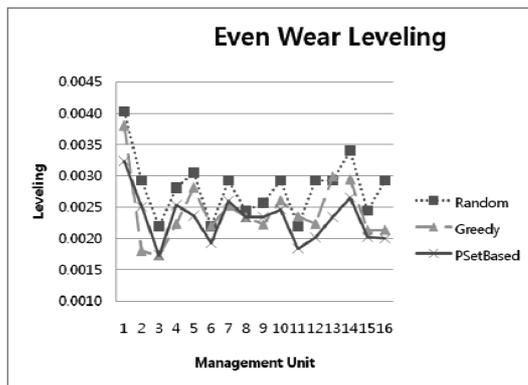


Fig. 2. Even Wear Leveling.

정적 균등소거 방법론은 전체 메모리 영역의 균등소거를 유지한다는 측면에서는 장점이 있지만, 갱신이 적은 데이터들이 전체 메모리에 균등하게 분포하지 않고 일부 관리영역에 집중적으로 위치할 때는 동적 균등소거 방법론과 같은 균등소거율의 차이가 현저하게 나타난다. 따라서, 관리영역의 구성을 동적으로 재구성 가능한 분할 기반 방법론은 결과치에서 보는 바와 같이 일정 부분에서 일어나는 저하된 균등소거율을 회복시키는 효과를 가져올 수 있다.

Fig. 3은 전체 관리영역에 대한 균등소거율의 표준편차를 보여준다. 관리영역 단위 내부에서 균등소거 방법론을 적용할 경우는 전체 메모리 영역에 균등소거를 적용할 때와 비교할 때, 소거 대상 블록을 검색할 때 필요한 시간이 감소한다는 장점이 있지만, 고정된 관리영역으로 인해 생기는 관리영역 간의 균등소거의 불균형은 전체 시스템의 장애를 가져올 수 있는 심각한 문제이다. 이를 해결하기 위한 동적인 관리영역의 재구성

은 결과 측면에서 우수한 면을 보이지만, 관리영역의 재구성을 결정하기 위해 시스템의 성능분석을 수행한다는 점과 재구성이 필요한 경우 관리영역에 속하는 블록들에 대한 정보를 서로 교환하는데 필요한 비용은 절대적으로 필요하고, 이는 최적화하지 않는다면 결과적인 성능에 영향을 미칠 수 있다.

관리영역의 재구성에 필요 비용을 줄이기 위해서는 분할에 대한 정보를 효과적으로 운영할 필요가 있으며, 이를 통해 필요 경비를 감소시키고 또한 재구성의 효과도 개선할 수 있을 것이다.

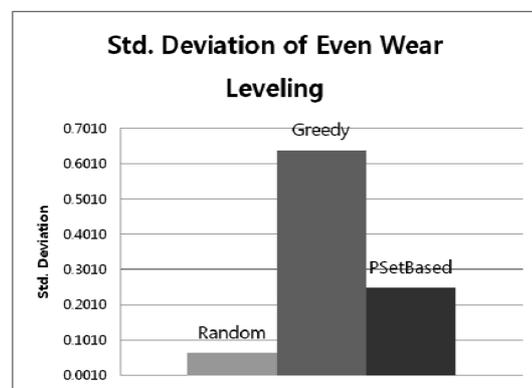


Fig. 3. Standard Deviation of Even Wear Leveling.

5. 결 론

본 논문에서 제안하는 균등소거 방법론은 메모리 재 활용을 위한 소거 대상 블록 선택에 소비되는 비용을 줄이기 위해 부분적 최적값이 아닌 근사 최적값에 기반하여 균등소거의 불균형을 가져오는 고정된 관리영역의 구성을 동적으로 재구성할 수 있는 기능을 부여하여, 전체 메모리 영역에서 불균형적인 소거 분포를 완화한다. 유효가능성 검증을 위해 본 논문에서는 시뮬레이션 기반 환경을 구축하여 실행한 실험 결과를 통해 제안된 방법론의 유효성을 검증하였으며, 제안하는 방법론이 우수한 성능을 가지고 있음을 보였다.

감사의 글

이 연구는 2009년도 단국대학교 대학연구비의 지원으로 연구되었음.

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. Brian Dipert, “Solid-state Drives Challenge Hard Disks”, EDN, Vol.53, No.23, 2008.
 3. Sang Lyul Min and Eyee Hyun Nam. “Current Trends in Flash Memory Technology” In Proc. of ASP-DAC 2006, Yokohama, Japan, January 24-27, 2006.
 4. Eran Gal and Sival Toledo, “Algorithms and Data Structures for Flash Memories”, ACM Computing Surveys, Vol.27, No.2, pp.138-163, 2005.
 5. M. Rosenblum and J. K. Ousterhout. “The Design and Implementation of a Log-Structured File System.” ACM Trans. Computer Systems, Vol.10, No.1, pp.26-52, 1992.
 6. Lin-Pin Chang and Tei-Wei Kuo, “An Efficient Management Scheme for Large-Scale Flash-Memory Storage Systems”, Proc. Of the 2004 ACM symposium on applied computing, pp.862-868, 2004.
 7. A. Jiang et. Al, “Storage Coding for Wear Leveling in Flash Memories”, CoRR, abs/0911.3992, 2009.
 8. Amir Ban, “Wear Leveling of Static Areas in Flash Memory”, United States Patent, no.0184432.
 9. 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.
-
- 접수일: 2010년 5월 28일, 심사일: 2010년 6월 7일
게재확정일: 2010년 6월 15일