한국산학기술학회논문지
Vol. 5, No. 5, pp. 389-394, 2004

K-집합 플래시 메모리 관리 성능 분석

박 제 호*

Performance Analysis of K-set Flash Memory Management

Je-ho Park*

요 약 이 논문에서는 플래시 메모리의 특성에 따른 메모리 재활용 방법론을 제안하여 비용을 감소시키면서 동시에 성능의 감소를 방지하는 목적을 만족시키고자 한다. 새로운 방법론은 재활용 대상 메모리 세그먼트 공간을 K 개의 하부 공간으로 분할하여 전체 검색을 대신 부분 검색을 실행하여 비용의 최소화를 추구한다. 아울러, 새로운 메모리 공간 배정 결정 시 전체 플래시 메모리 공간의 균등 소거에 대한 개선책을 제안한다. 제안된 방법론의 최적화를 위하여 하부 공간 분할 크기를 실험을 통해 취득 하였다. 실험적 자료는 제안된 방법론이 기존의 방법론과 비교하였을 ãö 비용은 감소하고 성능은 개선되는 것을 예시한다. 또한, 실험을 통해 메모리 공간 배정이 메모리 공간의 균등 소거에 많은 영향을 미친다는 사실을 확인할 수 있다.

Abstract In this paper, according to characteristics of flash memory, a memory recycling method is proposed in order to decrease the necessary cost preventing performance degradation at the same time. In order to optimize the demanding costs, the new approach partitions the search space of flash memory segments into K segment groups. A method for memory space allocation, in addition, is proposed in order to satisfy the goal of even wearing over the total memory space. The optimized configuration of the proposed method is achieved through experiments. The fact that the newly proposed methods outperform the existing approaches regarding cost and performance is evaluated by simulations. Furthermore the experimental results demonstrate that the memory allocation method affects even wearing in great deal.

Key Words : Flash memory, Segment cleaning, Even leveling, Ubiquitous memory, Performance analysis

1. 서 론

플래시 메모리는 비휘발성 저장 매체로 상대적으로 적은 에너지를 필요로 하는 특성을 가지고 있다. 응용 분야로는 이동 기기, 전자기기, 내장시스템등을 들수 있다. 아울러 유비쿼터스에 대한 연구가 활성화되면서 플래시 메모리의 특성이 유비쿼터스 응용 기기들에 필수적이기 때문에 플래시 메모리에 대한 관심은 더욱 가속화되고 있다[1-3, 5, 7]. 이러한 경향과 함께 대용량 플래시 메모리의 상품 가능성이 가시화되면서 차세대 스토리지로서의 역활에 대한 기대가 크다. 이에 따라 플래시 메모리의 성능 향상을 위해 통계적 방법을 사용한 연구[4] 나 실시간 시스템에 대한 플래시 메모리의 응용도 고려되고 있다[6]. 하지만, 플래시 메모리는 그 하드웨어 특성이 일반 디스크나 메인메모리와는 상이한 메모리 소거 관리를 필요로 한는 단점을 특성을 지니고 있다[3, 5].

플래시 메모리 공간은 하드웨어 생산자가 정의하는 세크먼트들로 분할 되어지며, 한 개의 세그먼트는 여러 개의 블록들로 구성된다. 블록은 읽기/쓰기를 수행하는 단위이다. 덮어쓰기를 위해서는 소거가 먼저 선행되어야 하며 소거는 세그먼트 단위로 수행되는 특성이 있다. 하지만, 세그먼트 소거는 그 속도가 느리고 많은 에너지 소모를 필요로 한다. 또한, 세그먼트 소거 횟수는 한정되어 있을 뿐 아니라, 한 세그먼트에 한정된 소거가 이루어지면, 전체 플래시 메모리를 사용할 수 없다는 특성을 가지고 있다. 이런 특성을 완화시키기 위해서는 소거 횟수를 최소화하여 전력 소비를 절약하고 개선된 시스템 성능과 장기간 플래시 메모리 사용을 가능하게 하여야 한다. 아울러, 일부 세그먼트에 소거가 집중되어 플래시 메모리의 수명을 짧게 만드는 것을 방지해야 하는 문제(wear leveling)가 존재한다.

*단국대학교 컴퓨터과학과

Table 1. Flash Memory Feature Values

특성	특성값
블록 읽기 시간	120~250 nanosec
블록 쓰기 시간	6~8 microsec/byte
블록 쓰기 시간	0.4~0.6 sec
블록 소거 시간	0.6~0.8 sec
세그먼트 크기	64/128 Kbytes
세그먼트 소거 한도	10,000~1,000,000

Table 1에서 플래시 메모리의 특성을 간략하게 정리하였다[1].

플래시 메모리에 존재하는 데이터를 갱신하기 위해서는 두가지 방법이 가능하다: *in-place* 와 *non-in-place*. In-place 소거는 본래 데이터가 있던 블록에 갱신된 데이터를 쓰는 방법이다. 이와는 달리 non-in-place 갱신은 기존의 데이터를 무효화하고 새로운 블록에 갱신된 데이터을 기록한다. Non-in-place 갱신의 대표적인 예는 Log-Structured File System(LFS)이다[8]. 플래시 메모리는 소거 비용이 높은 관계로 non-in-place 갱신을 더 선호한다[3].

플래시 메모리에서 새로이 데이터를 포함하지 않는 세그먼트를 확보하기 위해서는, 부분적으로 무효화된 데이터(블록)를 포함한 세그먼트에서 유효한 자료를 다른 세그먼트로 옮긴 뒤, 해당 세그먼트를 소거한 뒤 사용한다. 이때 소거 대상 세그먼트를 결정하는 방법들을 클리닝(cleaning) 정책이라 하고, 클리닝 정책을 수행하기 위한 방법론은 클리닝 시기, 세그먼트의 선택, 유효 데이터의 다른 세그먼트(들)로의 분배와 관련된 사항을 고려해야 한다.

소거 대상 세그먼트의 선택에 있어 가장 기본적인 방법은 무작위 선택이다. 탐욕적 알고리즘은 탐색 대상 세그먼트들 중에서 제일 많은 무효 데이터와 사용되지 않는 블록을 가진 세그먼트를 소거 대상으로 선택한다[1]. 이와는 달리 비용/편익 기반 알고리즘은 획득할 수 있는 블록의 갯수와 함께 데이터의 생성 이후 시간을 고려하는 방법이다[1]. 무작위 선택을 제외한 나머지 방법론은 전체 세그먼트들을 검색할 필요가 있다.

이 논문에서는 먼저 탐색 절차에 필요한 비용을 최소화하는 동시에 위에서 언급한 방법론을 통해 획득 가능한 효율성의 상실을 최소화하는 방법론을 소개하고, 새로운 방법론에 사용되는 시스템 매개 변수값의 최적치에 대해 논의하고자 한다.

2. K-집합 플래시 메모리 관리

K-집합 플래시 메모리 관리는 소거 대상 세그먼트 탐색에 필요한 비용을 최적화하고, 그 결과로 발생할 지도 모르는 성능 악화를 방지하고자 하는 방법론이다. 이러한 특성은 차세대 저장매체로서 플래시 메모리의 크기가 기존의 크기보다 비교가 되지 않을 정도로 확대되는 상황을 고려한 것이다.

무작위 방법론을 제외하고, 이제까지 연구된 소거 대상 세그먼트 선택 알고리즘들은 최소 또는 최대 특정값을 탐색하기 위하여 모든 대상 세그먼트의 속성을 각각 계산하고 검색해야 하는 전체 검색을 필요로 한다. 주어진 플래시 메모리의 세그먼트 수를 N이라고 정의하고, 현재 적어도 한 개 이상의 유효한 데이터 블록을 포함하고 있는 세그먼트들의 전체 세그먼트 영역에 대한 비율을 u라고 정의하자. 알고리즘에 따라 목적 속성을 계산하고, 최소값/최대값을 탐색해야 하는 세그먼트의 수 S는 다음과 정의될 수 있다.

$$\text{탐색 영역 } S = \lceil N*u \rfloor \text{ 세그먼트} \quad (1)$$

플래시 메모리의 크기가 작거나 유효 데이터율이 작은 경우 S에 대한 계산과 탐색 비용은 간과될 수 있다. 하지만, 차세대 저장매체로서의 플래시 메모리의 크기가 기가 단위로 커지고 있는 점을 감안할 때, 그 비용은 저장매체 시스템의 성능에 심각한 성능 장애 요인이 될 수 있다. K-집합 플래시 메모리 관리는 이러한 계산/탐색 비용을 최소화하는 동시에 소요 비용에 비해 그 결과를 최대화하려는 방법이다.

K-집합 플래시 메모리 관리는 계산/탐색 영역 S를 K개의 하부 검색 영역으로 분할한다. 이상적으로 각 하부 탐색 영역의 크기는 다음과 같이 정의된다.

$$\text{하부 탐색 영역 } = \lceil N*u/K \rfloor \text{ 세그먼트} \quad (2)$$

K-집합 플래시 메모리 관리를 사용할 경우 소거 대상 세그먼트를 결정하기 위한 탐색 영역의 크기는 식 (2)에서 주어진 크기를 갖는다. 여기서 각 하부 탐색 영역은 하나의 동일 속성을 공유하게 되며, 각 속성값에 따른 우선순위를 갖게 되어, 소거 대상 세그먼트를 결정할 때 하나의 최상 우선순위를 가지는 하부 영역만이 탐색의 대상이 된다. 결과적으로 대상 하부 영역에 무작위 선택이 수행된다면 별도의 검색이 없이, 적절한 세그먼트를 선택할 수 있게 된다.

하나의 하부 영역에 속하는 세그먼트들은 기본적인

각 세그먼트 속성값이라는 측면에서는 반드시 일치하는 값을 가지지 않으므로, 무작위 선택 방법은 그 성능 면에서 전체 탐색 기법보다 떨어질 수도 있다. 하지만, 한 하부 영역에 속하는 세그먼트들이 근사한 속성값을 가질 수 있도록 전 체 영역을 분할 할 수 있다면, 그 결과의 차이는 미세할 뿐만 아니라 비용 대비 효과라는 측면에서는 전체 탐색 방법론와 비교할 때 탐색 비용의 최소화라는 결과를 가져온다. 또 다른 방법은 대상 하부 영역에 대한 전체 탐색을 실시하는 것이다. 이 경우 하부 영역에 대한 탐색 비용이 간과할 수 있을 정도라면 이는 전체 탐색 방법론과 비교해 우수한 비용 대비 효과를 가져올 것이다.

전체 탐색 영역을 속하는 세그먼트들은 실제로 플래시 메모리의 수명 동안 계속하여 변화한다. 따라서, 하부 탐색 영역으로 대상 세그먼트들을 분할하기 위해서는 고정된 하부 영역 속성값을 설정하여야 한다. 이 목적을 위해 탐욕적 방법론에 사용되는 세그먼트 속성값을 K개의 하부 영역으로 분할하는 방법을 제안한다.

탐욕적 방법론은 대상 속성값을 한 세그먼트에 존재하는 무효 데이터 블록의 수와 데이터 블록으로 사용되지 않은 블록의 수의 합으로 한다. 따라서, B를 한 세그먼트에 속하는 블록수로 정의할 때, 다음과 같은 영역을 갖는다.

$$1 \le \text{탐욕적 속성값} \le (B-1) \quad (3)$$

이 영역을 K개로 분할하기 위해 다음과 같이 구간 간격 W를 정의한다.

$$W = \lceil (B-2)/K \rfloor \quad (4)$$

여기서 각 하부 영역은 대표 속성값 P를 가진다고 가정하면, 대표 속성값은 다음과 같은 영역을 가진다.

$$1 \le P \le K \quad (5)$$

따라서, 대표 속성값이 P인 하부 영역은 세그먼트 속성값에 대해 다음과 같은 하한값 low와 상한값 upper를 가지도록 정의한다.

$$\text{lower} = 1+(P-1)*W \quad (6)$$

$$\text{upper} = \text{lower}+(W-1) \quad (7)$$

하부 영역 대표 속성값이 K인 경우는 상한값을 (B – 1)로 정의한다.

한 세그먼트의 대상 속성값에 변화가 생길 경우, 위에 주어진 방법에 따라 대표 속성값이 계산되고, 해당 세그먼트는 대표 속성값에 따라 소속되는 하부 영역을 변경할 수 있다.

소거 대상 세그먼트의 선택과 함께 새로운 자유 세그먼트가 필요할 경우 자유 세그먼트 집합으로부터의 세그먼트 선택도 신중히 고려해야 할 문제이다. 일반적으로 플래시 메모리 크기가 증가하면서 자유 세그먼트 집합의 크기도 비례하여 증가한다고 가정할 수 있다. K-집합 플래시 메모리 관리는 자유 세그먼트 집합에서 하나의 세그먼트를 선택할 때 전체 집합에서 최소 소거 횟수를 경험한 세그먼트를 우선적으로 선택한다. 이러한 방법은 비용/편익 기반 클리닝이 가지는 특성, 즉 모든 세그먼트에 대한 소거 횟수의 균등한 분포를 유지하기 위해 최소 소거 횟수에 기반한 대상 세그먼트를 선택한다 점에 착안을 한 것이다. 제안된 자유 세그먼트의 선택은 웨어 레벨링에 적용되는 방법의 효과를 한층 증폭시키는 데 목적을 두고 있다. 본 논문에서 제안하는 K-집합 플래시 메모리 관리의 유효성은 시뮬레이션 기반 실험을 통해 검증하였다.

3. 구현 사례 및 결과 분석

3.1 구현 사례

시뮬레이션 기법에 기반한 실험의 목적은 첫째로 K-집합 플래시 메모리 관리와 균등 레벨링을 위한 경험적 자유 세그먼트 선택 방법의 유효성을 실험 결과치를 통해 검증하고, 둘째로 K-집합 플래시 메모리 관리에서 K값의 최적값을 실험을 통해 밝혀보고자 하는 데 있다. 유효성 검사는 동일한 플래시 메모리 모델을 사용하여 기존의 클리닝 정책들과의 비교 분석 방법을 선택하였다. 시뮬레이션에 사용된 플래시 메모리 모델의 구성은 Fig. 1에서 볼 수 있다. 사용된 플래시 메모리 모델은 플

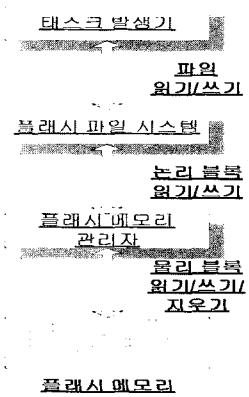

Fig. 1. Simulation System Configuration

래시 메모리의 전형적인 모델을 따라 구성하였다[1, 2].

테스크 발생기는 접근할 논리 블록 번호를 주어진 환경값에 따라 임의적으로 생성한다. 접근 논리 번호는 플래시 파일 시스템에 의해 플래시 블록 번호로 변환되어, 해당 블록의 읽기/쓰기를 한다.

플래시 메모리의 구조는 관장하는 메모리 공간을 정해진 수의 플래시 메모리 블록을 포함하는 세그먼트로 나누어진다. 각 세그먼트 헤더를 통해 관리되는 정보의 내용은 Table 2에 상세하게 나타나 있다. 전체 세그먼트들에 대한 정보인 총 세그먼트 수와 세그먼트 당 블록 갯수는 세그먼트 정리 헤더(Segment Summary Header)에 기록한다. 각 플래시 블록은 Table 3에 정리되어 있는 정보를 유지한다.

Table 2. Segment Header Structure

정보 이름	정보 내용
No of Erasures	세그먼트 소거 횟수
Timestamp	마지막 오퍼레이션 작동 시간
In Used Flag	세그먼트의 자유성 여부
Starting Block	세그먼트의 첫번째 블록 번호
Ending Block	세그먼트의 마지막 블록 번호
Valid Blocks	세그먼트의 유효 블록의 갯수
First Free Block	세그먼트의 첫번째 유효 블록 번호

Table 3. Block Information

정보 이름	정보 내용
LBlock No	상응 논리 블록 번호
Timestamp	마지막 오퍼레이션 작동 시간
Updates	쓰기를 위해 사용된 횟수
In Used Flag	사용 여부
Invalid Flag	자료의 무효화 여부

3.2 성능 비교를 위한 실험

플래시 메모리 사이즈가 큰 경우, 본 논문에서 비교를 위해 사용되는 세그먼트 선택 알고리즘들은 그 성능 측면에서 별 다른 차이를 보이지 않는 것으로 알려져 있다[1, 2]. 이 연구 결과를 따라, 본 논문에서는 플래시 메모리의 사이즈를 1기가 바이트로 설정하여 실질적 환경에서 응용할 수 있는 플래시 메모리 시스템을 검증하려고 노력하였다. 일반적인 플래시 메모리의 구성에 따라서, 세그먼트 크기는 64 KByte로 설정하고, 메모리 블록은 2 KByte로 설정하였다. 플래시 메모리 활용도 역시 플래시 메모리 시스템에 크게 영향을 미치는 것으로 알려져 있다[1, 2, 3]. 성능 측면에서 관찰할 때 병목 현상은 주로 플래시 메모리의 80%의 활용도에서 나타나기 때문에 본 논문도 활용도 80%를 설정하였다. 실험은 접근 집약성(locality)를 증가시키면서 수행되었다. 이는 접근 집약성에 따른 플래시 메모리의 성능 변화가 플래시 메모리 시스템 구현의 중요한 사항으로 인식되고 있기 때문이다[1]. 모든 실험은 5 개의 그룹을 사용하여 K-집합 플래시 메모리 관리 알고리즘을 적용하였다.

소거 대상 세그먼트 선택 알고리즘은 무작위 정책,

Table 4.. Selection of Cleaning/Free Segment and K-set

범례	소거 세그먼트 선택	자유세그먼트 선택	K-집합 중 하나에서 세그먼트 선택
Random-R	무작위	무작위	해당없음
Greedy-R	탐욕적	무작위	해당없음
CostBenefit-R	비용/편익 기반	무작위	해당없음
MultisetR-R	다중집합 기반	무작위	무작위
MultisetY-R	다중집합 기반	무작위	최소 소거
Random-Y	무작위	최소 소거	해당없음
Greedy-Y	탐욕적	최소 소거	해당없음
CostBenefit-Y	비용/편익 기반	최소 소거	해당없음
MultisetR-Y	다중집합 기반	최소 소거	무작위
MultisetY-Y	다중집합 기반	최소 소거	최소 소거

탐욕적 정책, 비용/편익 정책, 그리고 K-집합 기반 정책을 구현의 대상으로 삼았다. K-집합 기반 정책과 자유 세그먼트 선택 방법론은 세그먼트 선택 시 두 가지 방법을 사용하였다: 무작위 선택과 최소 소거에 의한 선택. 소거 대상 세그먼트 선택 방법과 자유 세그먼트 선택 알고리즘은 Table 4에 정리되어 있는 경우를 따라 수행하였다.

3.3 성능 비교 실험 결과 분석

플래시 메모리 시스템의 성능을 분석하기 위한 측정치로는 한 블록을 갱신했을 경우 지원 접근수를 수집하였으며, 레벨링 효과를 측정하기 위해서는 전체 세그먼트의 소거 횟수에 대한 표준편차를 수집하였다. Fig 2와 Fig 3은 한 개의 플래시 블록 갱신 당 지원 접근수를 보여 주고 있는데, 세그먼트 소거 당 지원 접근수와 유사하게 탐욕적 방법과 K-집합 기반 정책이 타 방법들보다 우수한 성능을 보이고 있다.

Fig 4는 레벨링 효과를 보여주고 있는데, 사용된 방법 중에서 비용/편익 기반 세그먼트 선택 알고리즘은 무작위 자유 세그먼트 우선 선택을 사용할 경우에도 우수한 성능을 보여주고 있다. 이는 비용/편익 기반 세그먼트 선택 알고리즘이 세그먼트 선택시 균등한 소거를 고려하기 때문이다. 같은 이유로 자유 세그먼트 선택에

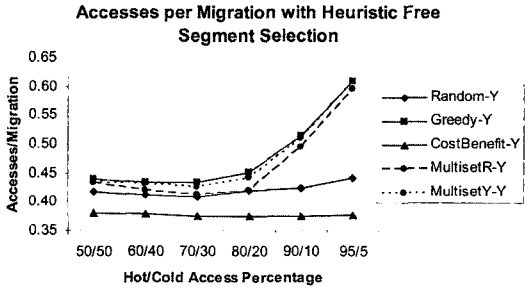

Fig. 2. Accesses per Migration with Random Free Segment Selection

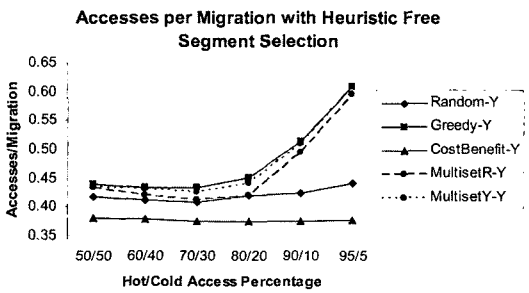

Fig. 3. Accesses per Migration with Heuristic Free Segment Selection

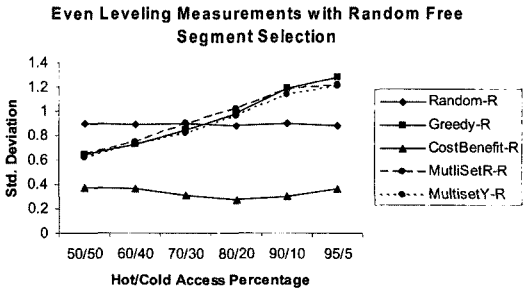

Fig. 4. Even Leveling Measurement with Random Free Segment Selection

서도 유사한 방법을 사용할 때 현저하게 레벨링 효과가 개선되는 것으로 나타났다.

3.4 최적 K 값 실험

K-집합 플래시 메로리 관리는 K 값에 의해 그 성능에 영향을 미칠 수 있다. 따라서, 최적 K 값을 결정하기 위하여 앞의 실험에서 사용한 시스템을 이용하여 K 값을 변화시키면서 그 성능을 플래시 블록 갱신 당 접근수와 레벨링을 측정하였다. 접근 집약성은 95%로 설정하여 대부분의 플래시 관리 알고리즘이 높은 성능을 보이는 환경을 채택하였다. 아울러, 소거 대상 세그먼트의 선택과 자유 세그먼트의 선택 방법에 대하여, 무작위 선택과 최소 소거 세그먼트 선택에 대한 모든 조합을 실험하였다. 사용된 조합은 Table 5에서 종합해서 정리를 하였다.

3.5 최적 K 값 실험 결과 분석

K-집합 플래시 메모리 관리에서 성능에 영향을 미치는 K값을 3에서 30까지의 범위에서 증가치를 3으로 사용하였다. 실험에서 한 세그먼트의 블록수는 32개 였다.

Table 5. Legends for K-set Optimization

범례	소거 세그먼트 선택	자유 세그먼트 선택
MS-RR	무작위 선택	무작위 선택
MS-YR	최소 소거에 의한 선택	무작위 선택
MS-RY	무작위 선택	최소 소거에 의한 선택
MS-YY	최소 소거에 의한 선택	최소 소거에 의한 선택

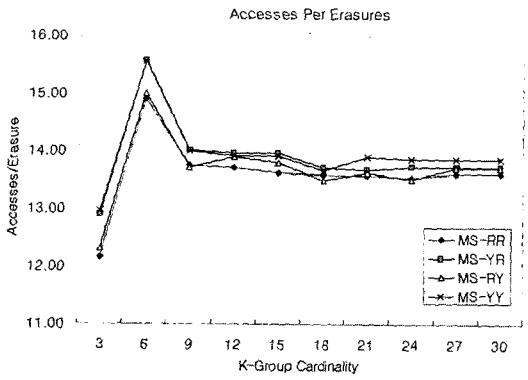

Fig. 5. Accesses per Erasure

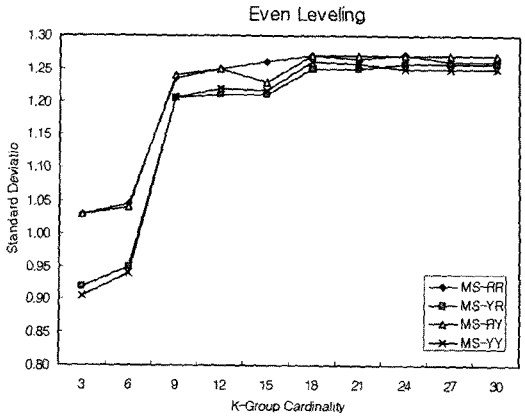

Fig. 6. Even Leveling with K-set

플래시 블록 당 갱신 당 접근수는 Fig 5에서 예시되었다. 그림에서 알 수 있듯 이 6개의 집합을 사용할 때 가장 최적의 성능을 보이고 있음을 알 수 있다. Fig 6를 통하여 레벨링 또한 6개의 집합을 사용하였을 때에 최적 성능을 보이고 있다.

4. 결 론

이 논문은 K 개의 집합을 이용하여 비용과 효율면에서 우수한 플래시 메모리 관리 알고리즘을 소거 대상 세그먼트 선택과 자유 세그먼트 선택이라는 측면에서 고찰하였다. 제안된 K-집합 플래시 메모리 관리는 소거 대상 세그먼트 선택에 있어서 그 비용이 낮은 반면에 성능은 기존의 방법과 유사하거나 보다 우수하다는 것을 실험을 통해 검증했다. 또한 자유 세그먼트를 선택함에 있어, 균등 레벨링을 고려한 방법은 소거 대상 세그먼트 선택을 할 때 균등 레벨링과 함께 사용되었을 때 현저히 성능을 개선 시킬 수 있음을 실험 결과 알 수 있었다.

K-집합 플래시 메모리 관리는 시스템 매개 변수로서 사용되는 집합의 수를 변화시킬 수 있다. 따라서, 최적 집합의 개수를 결정하는 것은 전체 플래시 메모리 관리에 중요한 과제이다. 이 최적값을 실험을 통해 추정한 결과 한 세그먼트 당 블록수의 15%에서 19% 사이의 값을 K 값으로 설정하는 것이 최적의 시스템을 구성하는 것으로 나타났다.

참고문헌

[1] Mei-Ling Chiang, Paul C. H. Lee Ruei-Chuan Chang, "Cleaning policies in mobile computers using flash memory", Journal of Systems and Software, 48(3):213-231, 1999.

[2] Mei-Ling Chiang, Paul C.H. Lee Ruei-Chuan Chang. "Using Data Clustering to Improve Cleaning Performance for Flash Memory", Software-Practice and Experience, 29(3):267-290, 1999.

[3] Fred Douglis et al. "Storage Alternatives for Mobile Computers", In Proceedings of the first USENIX Symposium on Operating Systems Design and Implementation (OSDI), November 14-17, Monterey, California, USA.

[4] Han-joon Kim, Sang-goo Lee, "An Effective Flash Memory Manager for Reliable Flash Memory Space Management", IEICE Transaction on Information and Systems, E85-D(6):950-964, June 2002.

[5] Atsuo Kawaguchi, Shingo Nishioka and Hiroshi Motoda, "A Flash-Memory Based File System", In Proceedings of the 1995 USENIX Technical Conference, New Orleans, Louisiana, USA, January 1995.

[6] Li-Pin Chang and Tei-Wei Kuo, "A Real-Time Garbage Collection Mechanism for Flash-Memory Storage Systems in Embedded Systems", In Proceedings of the 8th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2002), 24-27 September 2002, San Jose, CA, USA.

[7] Joshua B Frymann et al. "Energy Efficient Network Memory for Ubiquitous Devices", IEEE Micro Special Edition Sept./Oct. 2003.

[8] M. Rosenblum and J. K . Ousterhout, "The Design and Implementation of a Log-Structured File System", ACM Trans. Computer Systems, 10(1): 26-52, 1992.