

# 임베디드 플래시 파일 시스템을 위한 플레인 지움 정책

이태훈<sup>0</sup> 이상기 정기동  
부산대학교 정보컴퓨터공학과  
{withsoul<sup>0</sup>, bruce94, kdjung}@pusan.ac.kr

## A Plain Cleaning Policy for Embedded Flash File System

Taehoon Lee<sup>0</sup>, Sanggi Lee, Kidong Chung  
Dept. of Computer Engineering, Pusan National University

### 요 약

최근 디지털 융합(Digital Convergence)이 활발히 진행되면서 이동형 장치(Mobile Device)는 더욱 대용량, 고성능화 되고 비휘발성 메모리 요구가 커지고 있다. 이에 휴대가 용이하며, 접근시간이 빠르고, 전력소비가 적은 플래시 메모리가 많이 사용되고 있으나 상대적으로 느린 지움 시간과 지움 횟수의 한계 등 극복해야 할 문제점이 있다. 본 논문에서는 이러한 문제점을 해결하기 위한 플레인 지움 정책을 제안하고 성능평가를 실시한다. 제안하는 플레인 지움 정책은 기존의 지움 정책과 같이 플래시의 블록단위의 균등한 사용을 고려할 뿐만 아니라 임베디드 시스템의 제한된 성능을 고려하여 연산을 최소화한다. 제안된 방법은 Greedy, Cost-benefit 방법에 비해 Wear-leveling에서 성능을 향상시켰고, RCP(Ranked Cleaning Policy)에 비해 연산횟수를 감소시켰다.

### 1. 서 론

디지털 융합(Digital Convergence)이 활발히 진행되면서 이동 단말에서는 대용량, 고성능화된 비휘발성 메모리를 요구하고 있다. 이전의 휴대폰의 경우 전화기능과 PIMS(Personal Information Management System)을 위해서 소량의 비휘발성 공간만 있으면 충분했지만, 최근의 휴대폰은 MP3, 디지털 카메라, 게임 등을 위한 대용량의 저장 공간이 요구된다. 이에 부피가 클 뿐만 아니라 무겁고 소비전력이 큰 하드디스크를 대체한 플래시 메모리가 사용되고 있다. 플래시 메모리는 비휘발성이며 하드디스크에 비해 견고하다. 그리고, 저 전력으로 동작이 가능하고, 접근 시간이 RAM과 유사할 만큼 빠르다. 또한 크기가 작아서 임베디드 시스템에 적합하다.

플래시 메모리는 덮어쓰기(In-Place-Update)가 불가능하기 때문에 데이터의 갱신 및 수정을 위해서는 지우기 작업을 선행하여야 한다[2]. 또한, 읽기 속도는 매우 빠르지만 쓰기 속도와 지우는 속도가 상대적으로 느리고 지움 수 있는 횟수가 제한되어 있다. 그리고 삭제는 세그먼트 단위로 수행하는데 비해 쓰기는 페이지 단위로 수행한다.

지움 수 있는 횟수는 상온에서 약 10만 번으로 정해져 있기 때문에 일정 세그먼트를 연속해서 사용하게 되면 한계 수명에 도달하게 되고 더 이상 사용할 수 없게 된다. 이는 메모리가 줄어드는 결과를 가져오고 지움 과정이 빈번하게 발생하여 시스템에 부담을 준다. 그러므로 지움 과정이 일어나는 세그먼트를 고르게 선택해야만 임베디드 시스템 성능을 향상시킬 수 있다. 이렇게 세그먼트에 대해 지움 과정을 고르게 인배하는 과정을 지움 평준화(Wear-leveling)라 한다. 플래시 메모리의 일반적인 특성은 <표 1> 과 같다.

본 논문에서는 플래시 메모리의 균등한 사용을 위해 각 블록의 유효한 페이지의 수, 빈 페이지의 수, 삭제 횟수를 유지하며, 이를 이용하여 삭제할 세그먼트를 선택한다. 기존의 지움

정책과 같이 각 블록의 정보를 모아서 세그먼트 단위로 다시 계산하지 않으므로 연산을 최소화 한다.

<표 1> 플래시 메모리의 특성(intel 28F640J3A)

특징	값
읽기 접근 시간	100~150ns
버퍼 이용 쓰기 시간	218 $\mu$ s /32bytes
지움 블록 쓰기 시간	0.8 sec/block
블록 지움 시간	1.0sec/block
최대 지움 회수	100,000 times
지움 블록의 크기	128Kbyte
전력 소모량	대기 전력: 50~120 $\mu$ A 동작 전력: 15~70mA

2장에서 기존의 지움 정책에 대해 살펴보고, 3장에서는 제안하는 플래시 메모리 지움 정책인 플레인 지움 정책을 설명한다. 4장에서는 성능평가 결과를 보여주며, 5장에서는 결론을 제시한다.

### 2. 관련연구

<표 1>에서 언급한 플래시 메모리의 한계 내에서 효율적으로 플래시 메모리를 사용하기 위한 다양한 플래시 파일시스템이 개발되었다. 특히 플래시 메모리의 가장 큰 제약인 느린 지움속도 문제를 극복하기 위한 지움 정책은 중요한 문제점 중의 하나이다.

Greedy 정책은 유효하지 않은 자료가 가장 많은 세그먼트를 선택하여 삭제하는 방법으로서, 클리닝 횟수는 가능한 한 최소화 하면서, 많은 공간을 확보 한다[3,4]. 삭제 효율은

"이 논문은 교육인적자원부 지방연구중심대학육성사업(차세대울류IT기술연구사업단)의 지원에 의하여 연구되었음."

중이나 삭제 시 특정 세그먼트에 집중해서 일어 날 수 있기 때문에 수명을 단축시킬 수 있는 단점이 있다.

$$\frac{u}{1-u} \quad (1)$$

세그먼트 내의 유효데이터의 비율  $u$ 만을 고려하여 (식 1)에서는 가장 작은 세그먼트를 선택한다.

Cost-Benefit 방법은 특정 세그먼트에 집중해서 삭제연산이 발생하여, 플래시 메모리의 수명을 단축시키는 문제점을 개선하기 위하여 고안된 정책이다[5]. (식 2)에서는 삭제의 효율성뿐만 아니라 특정 세그먼트에 대한 집중을 피하기 위해 세그먼트가 사용된 최근 시간을 고려하여, 클리닝할 세그먼트를 선택한다.

$$\frac{benefit}{cost} = \frac{age \times (1-u)}{2u} \quad (2)$$

(식 2)에서  $age$ 는 세그먼트가 가장 최근에 수정되었던 이후의 시간이며,  $u$ 는 세그먼트의 이용률,  $1-u$ 는 활용 가능한 세그먼트의 비율을 의미한다.

순위별 지움 정책(RCP)은 세그먼트들의 등급값을 계산하여 값이 큰 순서대로 세그먼트를 지우는 방법이다[1]. 지움 비용과 Wear-leveling 모두를 고려한 방법이다. (식 3)에서는 순위별 지움 정책(RCP)을 나타낸다.

$$R = A \times \frac{i}{2v \times f \times e} \quad (3)$$

(식 3)에서  $v$ 는 세그먼트 내에서 유효 블록에 대한 비율,  $f$ 는 비어 있는 블록에 대한 비율,  $i$ 는 무효 블록에 대한 비율,  $e$ 는 세그먼트의 전체 지움 가능 횟수에 대한 현재 지운 횟수의 비율,  $A$ 는  $e$  값에 대한 가중치를 의미한다.

### 3. 플레인 지움 정책 (Plain Cleaning Policy)

기존의 Greedy 방법은 Wear-leveling을 무시하고 있으며 Cost-benefit 방법은  $age$ 로 고려하고 있지만 삭제 횟수가 적다는 것을 의미하지는 않는다. 이러한 문제점을 해결하기 위해 RCP는 삭제 횟수를 사용하였다. 그러나 제한된 자원을 가진 임베디드 시스템에서는 Garbage Collection이 발생할 때마다 Invalid Data의 비율( $i$ ), Valid Data의 비율( $v$ ), Free data의 비율( $f$ ), 세그먼트의 전체 삭제 가능 횟수에 대한 현재 지운 횟수의 비율( $e$ )을 각 세그먼트에 대해 구하고, 순위를 결정하는 많은 연산으로 인해 전체 시스템의 성능을 저하시키게 된다.

본 논문에서는 임베디드 시스템의 제한된 연산능력을 고려하여 연산을 최대한 줄이며 Wear-leveling을 지원하는 플레인 지움 정책(Plain Cleaning Policy)을 제안한다.

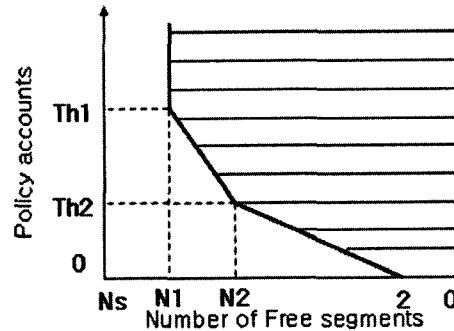
플래시 파일 시스템의 성능은 쓰기 연산에 의해 결정된다. 쓰기 연산은 데이터를 쓰는 연산뿐만 아니라 빈공간이 없을 경우 Invalid Data를 지우는 연산, 지우는 세그먼트에 Valid 데이터가 포함되어있을 경우 이 데이터를 빈 공간에 옮기는 연산도 포함한다. 그러므로 파일 시스템의 성능을 향상시키기 위해서는 지움 비용이 적은 세그먼트를 선택하여야 한다. (식 4)는 지움 비용을 나타낸다.

$$C = (2 \times V) + F \quad (4)$$

(식 4)에서  $V$ 는 Valid 데이터를 다른 세그먼트로 옮기는 비용과 지우는 비용을 포함한다.  $F$ 는 삭제될 세그먼트의 Free 공간을 지움으로써 낭비되는 비용이다. 삭제할 세그먼트는 (식 5)를 이용하여 결정된다.

$$R = \frac{EA}{C} \quad (5)$$

(식 5)에서  $EA$  (ErasAble)는 플래시 메모리의 삭제 한계치와 현재 삭제 횟수의 차이 값으로 균등한 삭제를 위해 사용한다. 블록 내의 페이지는 Valid, Invalid, Free 세 가지 상태 중 하나에 해당된다. 그러므로 두 가지의 상태를 가진 페이지의 비율을 알 수 있다면 나머지 한 가지 상태의 페이지의 비율을 알 수 있게 된다. 그러므로 제안하는 지움 정책에는  $V$ 와  $F$ 의 비율을 만들어 이용하여 계산 복잡도를 낮추었다.



<그림 3> PCP의 임계값 결정 그래프

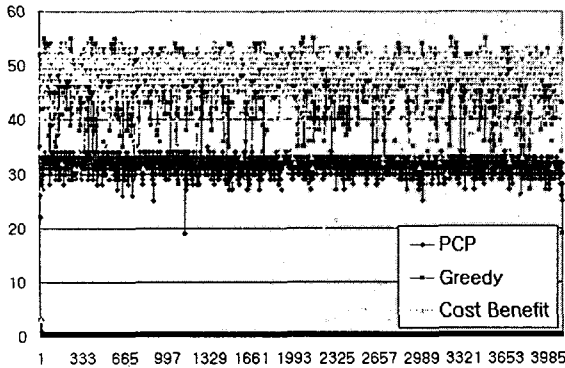
RCP 방법은 한 개의 임계값을 가지고 Free 세그먼트의 수가 임계값 이하로 떨어지면 지움 정책이 수행된다. 그리고 임계값 이상을 확보할 때까지 세그먼트를 지워준다[1]. 하지만 임계값이 하나이기때 임계값 이하로 떨어진 후에만 지움 정책이 동작하게 되어 시스템의 부하가 집중될 수 있다. 본 논문에서는 지움 연산의 집중을 해결하기 위해 <그림 1>과 같이 2개의 임계값을 사용하였다. Free 세그먼트의 수가  $Th1$ 과  $Th2$ 의 사이에 있을 때는 세그먼트 내의 모든 블록이 Invalid이며 삭제 횟수가 최소인 세그먼트를 찾아서 삭제한다. Valid data가 하나라도 있는 세그먼트는 지움 대상에서 제외한다. 이러한 이유로 본 논문에서 제안한 기법은 Valid data의 복사 비용이 없는 이상적인 지움 연산을 수행 한다. Free 세그먼트의 수가  $Th2$ 이하인 경우는 RCP 방법과 같이  $Th2$  개의 Free 세그먼트를 유지하기 위한 차이 만큼의 세그먼트를 지워준다. 이렇게 두 개의 임계값을 사용함으로써 지움 연산은 문산적으로 수행한다.

### 4. 실험

본 논문에서 제안한 플레인 지움 정책에 대하여 시뮬레이션으로 성능을 평가하였다. 시뮬레이션 환경을 64MB(Small Block)Block 단위를 RAM 상에 구현하였다. 초기에 플래시 메모리 공간에 40%의 데이터를 저장시킨 후, 새로운 데이터를 갱신하는 방법으로 테스트를 수행하였다. Greedy와 Cost-benefit 방법에서 임계값  $Th$ 는 한 세그먼트에 저장 가능한 128개의 페이지 중 115개를 기준으로 하였으며,  $N = 12$ 로 설정하였고, RCP 방법은  $N1 = 2$ ,  $N2 = 12$ 로 두었다. PCP 방법에서도 다른 방법과의 동일한 실험 환경설정을 위해  $Th1 =$

15, Th2 = 12 로 설정 하였다. 데이터 갱신은 난수를 발생시켜 갱신하였다[4,5].

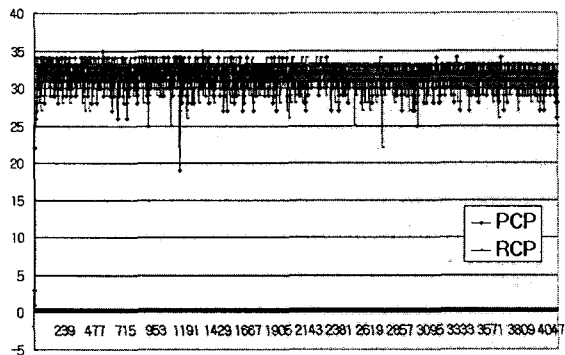
<그림 2>에서 각 블록의 지움 횟수에 대한 성능을 보여주고 있다. Greedy 방법은 지움 횟수에 대하여 Cost Benefit 방식과 유사한 성능을 보이고 있으나, Wear-leveling에 있어서 좋은 성능을 보이고 있다. <그림 2>와 <그림 3>에서 PCP는 본 논문에서 제안한 임베디드 시스템의 제한된 연산능력을 고려하여 연산을 최대한 줄이며 Wear-leveling을 고려한 Plain Cleaning Policy 이다. PCP 방법은 전체적인 범위에서 좋은 성능을 보이고 있다.



<그림 4> Greedy, Cost-Benefit과의 성능 비교

그 이유는, 데이터가 작을 때 세그먼트의 지움 횟수가 작으면 지움 평균화를 거의 고려하지 않지만, 지움 횟수가 한계 값에 가까워지면, 지움 비용이 고려되기 때문이다.

<그림 3>는 40% 초기 데이터에 대한 RCP 방법과 PCP 방법에 대한 지움 회수의 성능평가를 보이고 있다.



<그림 5> RCP와의 성능 비교

PCP 방법은 각 블록의 지움 횟수와 Wear-leveling 정도에 있어서 RCP와 매우 유사한 성능을 보이고 있으나, 계산 복잡도를 고려한 비용에 있어서 본 논문에서 제안한 기법이 우수한 성능을 보인다.

<그림 4>는 플래시 메모리의 쓰기 작업인 write(), cleaning()과정을 나타낸다.

PCP 방법의 계산 복잡도는  $\theta(N) = 6N + 1$ , RCP방법의 경우는  $\theta(N) = 12N$  으로 cleaning() 수행 시 약 2배의 차이를 나타낸다. 복잡도의 차이는 각 세그먼트의 비율에 대한 연산으로

인한 차이이다. PCP에서는 Valid Page의 비율과 사용하지 않은 빈 공간의 비율을 이용하여, 최소의 지움 비용을 가지는 세그먼트를 선택한다. 이러한 지움 비용을 고려한 계산 복잡도의 단순화는 플래시 메모리의 수명을 연장시킨다.

```

write(){
    Allocate a free block;
    If new write
        Write data into the free block
    else{
        Mark the obsolete data as invalid;
        Write data into the free block;
    }
}
cleaning(){
    Select a victim segment for cleaning;
    Identify valid data in the victim segment;
    Copy out valid data to another clean flash memory
    Erase the victim segment;
}
    
```

<그림 4> write(), cleaning() 과정

5. 결론

본 논문에서는 플래시 메모리 관리를 위해 계산 복잡도를 고려한 새로운 지움 정책을 제안했다.

Plain Cleaning Policy 방법은 NAND 플래시 메모리를 기준으로 설계 하였고 Valid page와 빈 공간을 고려 하였다. 플래시 메모리의 수명은 특정 세그먼트의 편중화 된 클리닝 횟수와 좋은 지움 정책을 설정하느냐에 따라 결정된다. 균등한 세그먼트의 사용을 위해 지움 비용을 최소화하는 세그먼트 선택이 중요하다. 적절한 시기의 G.C 수행은 지움 비용을 최소화 시킨다. 향후에 이 방법을 고려한 기법을 연구할 것이다.

NAND 플래시 메모리와 같은 비휘발성 메모리는 다양한 형태와 다양한 용량으로 개발되고 있다. Embedded 시스템의 유형 또한 다양화 되고 있다. 이에 맞는 새로운 플래시 파일 시스템의 개발이 필수적이다.

참고 문헌

- [1] 김정기, 박승민, 김채규 "임베디드 플래시 파일 시스템을위한 순위별 지움 정책", 정보처리학회논문지 제9-A권 제4호 2002년 12월.
- [2] 김한준, 이상구 "신뢰성있는 플래시메모리 저장시스템 구축을위한 플래시메모리 저장 공간 관리방법" 정보과학회논문지: 시스템 및 이론 제27권 제6호 2000년 6월.
- [3] M. L. Chang, P. C. H. Lee, R. C. Chang, "Managing Flash Memory in Personal Communication Devices", Proc. of IEEE Symp. on Consumer Electronics pp177-182, 1997.
- [4] Mei-Ling Chiang, Paul C. H. Lee Rwei-Chuan Chang, "Cleaning Policies in Mobile Computers Using Flash Memory", journal of System and Software.
- [5] Atsuo Kawaguchi, Shingo Nishioka, Hiroshi Motoda "A Flash-Memory based File System", Proceedings of 1995 USENIX Technical Conference, pp.155-164, 1995.