

Hot-and-Cold 플래그를 이용한 플래시 메모리 균등 사용을 위한 관리 정책

정필영*, 이성준, 안광선
경북대학교 컴퓨터공학과
e-mail : withpyj@msn.com

Flash Memory Management Policy To Improve Wear-Leveling using Hot-and-Cold Flag

Phil Young Jung*, Sung Joon Lee, Kwang Seon Ahn
Dept. of Computer Engineering, Kyungpook National University

요 약

비휘발성 메모리인 플래시 메모리는 가볍고, 전력 소비가 적으며, 충격에 대한 저항이 강한 특징이 있다. 그러나 플래시 메모리는 세그먼트당 삭제 횟수가 최악의 경우 10 만 번 정도로 제한된다는 단점이 있어서, 만약 삭제가 특정 세그먼트에 집중된다면 전체 수명이 단축되게 된다. 하지만, 플래시 메모리의 모든 세그먼트가 골고루 사용된다면 수명을 연장시킬 수 있다. 이를 위해 저장되는 자료의 유형을 파악하여 Hot, Cold 그리고 Lukewarm 이라는 그룹별로 분리해서 관리한다. 단순한 분리는 사용 횟수의 양극화를 가져오지만, 양극화된 세그먼트에 반대 유형의 자료를 할당하고 관리하면 전체 세그먼트의 사용률을 균등하게 할 수 있다. 이를 위해, 쓰기와 클리닝 작업 외에 양극화의 패턴을 통해 유형을 결정해주는 루틴을 포함하였다.

1. 서론

플래시 메모리는 가볍고, 전력 소비가 적으며 충격에 대한 저항이 강하고 비휘발성 메모리라는 특징을 가진다. 또한, EEPROM 처럼 내용을 갱신하기 위해 모든 내용을 지워야 하는 방식이 아니라, 선택적 삭제가 가능하다는 장점 때문에 PDA 나 휴대폰과 같은 임베디드 시스템의 보조 기억장치에 주로 사용된다.

그러나 플래시 메모리는 두 가지 단점을 가진다. 첫째, 플래시 메모리는 특정 영역의 덮어쓰기(update-in-place)가 불가능하다는 점을 들 수 있다. 이로 인해 새로운 자료를 특정 영역에 덮어쓰기 위해서는 쓰기 단계 이전에 해당 영역을 지우는 과정이 필요하다. 게다가 세그먼트라는 한번에 지울 수 있는 단위 영역이 64Kbytes~128Kbytes 인데 비해 쓰기 단위 영역인 블록은 4Kbytes 이다. 또한, 보통 한 세그먼트를 지우는 데 1 초 정도의 긴 시간이 걸린다. 이런 문제를 보완하기 위한 방법으로 쓰기 작업과 삭제 작업을 분리해서 수행하는 Log-Structured File System 이라는 방식이 고안되었다[1].

둘째, 세그먼트당 삭제 횟수가 최악의 경우 10 만 번 정도로 제한된다는 점이다. 이 문제점은 플래시 메모리 영역을 전반적으로 골고루 사용함으로써 그 수명을 연장시킬 수 있다. 이를 위해 Hot-and-Cold 라고 불리는 플래그를 사용한다. 빈번한 수정 작업이 일어나는 영역을 Hot, 이동이 주로 이루어지는 영역을 Cold 라고 한다. 또한, Hot 과 Cold 의 경계 부근에 위치하고 있으면서 그 성격이 모호한 Lukewarm 이라는 영역을 두었다. 이 영역은 추이를 좀더 지켜볼 수 있도록 완충 지대 역할을 하므로, 유형 결정의 정확성을 좀더 높일 수 있게 한다.

2 장에서는 이와 관련된 연구들을 살펴보고, 3 장에서는 기존의 Hot-and-Cold 방식의 보완 방법을 언급한다. 4 장에서는 구성 방법과 그 알고리즘을 밝히며, 결론은 5 장에서 기술한다.

2. 관련 연구

Log-Structured File System 은 쓰기와 삭제 작업을 분리해서 수행하는 방식이다[1]. 플래시 메모리에 이

미 기록되어 있는 자료를 수정할 때, 갱신(update)되는 자료는 첨부의 형태로 빈 공간에 기록을 하고, 저장 위치를 최근의 것으로 변경한다. 이것은 유효하지 않은 자료의 삭제는 추후에 수행할 수 있기 때문에 수정 작업의 속도를 향상시키는 장점을 가진다. 이전에 기록되었던, 현재 사용 중이 아닌 자료를 삭제함으로써 빈 공간을 확보하는 것을 클리닝(cleaning)이라 한다.

Greedy 방식은 유효하지 않은 자료가 가장 많은 세그먼트를 선택하여 삭제하는 방법으로, 클리닝 횟수는 가능한 한 최소화 하면서, 가능한 한 많은 공간을 확보하고자 한다[1]. 이것은 삭제 효율은 좋으나 삭제 시 특정 세그먼트에 집중해서 일어날 수 있어서 수명을 단축시킬 수 있다는 단점이 있다.

이점을 보완하여 Cost-Benefit 방식이 고안되었다. 이 방식은 삭제의 효율성뿐 아니라 특정 세그먼트에 대한 집중을 피하기 위해 다음의 식을 최대화 만족시키는 세그먼트를 선택한다[2]:

$$age * (1-u) / 2u$$

age 는 세그먼트가 사용된 최근 시간이며, u 는 세그먼트의 활용도, (1-u)는 활용 가능한 빈 공간의 양을 의미한다. 한 세그먼트를 클리닝 하는데 필요한 비용은 2u 라고 표현되어지는데, 하나의 u 는 유효 블록을 읽는데, 다른 u 는 그것을 다른 곳에 옮겨 쓰는데 필요하다. 여기서 유효 자료를 클리닝을 위해 다른 곳에 옮겨 써주는 것을 이동(migration)이라 한다.

Cost-Age-Times 방식은 클리닝을 위한 대상 세그먼트를 선택할 때 다음의 식을 고려한다[3]:

$$Cleaning\ Cost * 1/age * Number\ of\ Cleaning,$$

여기서 클리닝 비용(Cleaning Cost)은 u / 1-u 로 정의된다. 그리고 age 는 세그먼트가 사용된 최초의 시간을 의미하며, 반비례 시키는 것은 갱신 빈도가 높은 세그먼트가 너무 일찍 삭제되는 것을 방지함으로써, 한 세그먼트가 가능한 한 오래 사용되게 해주는 역할을 한다. 이렇게 하여, 유효하지 않을 데이터를 집약시킨다면, 유효 데이터를 이동 시키는데 소요되는 클리닝 비용을 줄일 수 있어, 빈번한 데이터의 이동과 세그먼트 삭제 횟수를 줄일 수 있다. 갱신이 곧 발생되어 유효하지 않게 될 데이터를 구별해내기 위해 Hot-and-Cold 개념이 사용되었다.

Hot-and-cold 는 보조 기억 장치에 저장되는 파일을 두 그룹으로 분류하기 위해 사용 된다[1,3]. 첫 번째 그룹은 전체 파일 용량에서는 10%를 차지 하지만, 사용 시간에 있어서는 90%를 차지하는 Hot 이라고 불리는 그룹이다. 이를테면, 메타 정보를 기록하여 수정이 빈번하게 발생하는 파일이 이에 해당된다. 두 번째 그룹은 파일 용량의 90%를 차지하지만, 사용 시간의 10%만을 차지하는 Cold 라고 불리는 그룹이다. 버전업그레이드가 가끔씩 이루어 지는 응용프로그램 같은 경우가 여기 해당된다.

3. Lukewarm

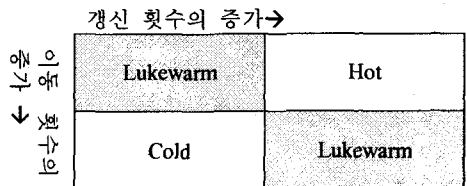
이러한 Hot-and-Cold 의 개념을 실제 플래시 메모리 관리 정책에 적용하였다. 쓰기 작업의 빈도수에 따라 파일의 성질을 구분하여 그룹별로 나누고 서로 다른 세그먼트를 할당한다. 쓰기 작업이 빈번하게 발생하는 Hot 그룹이 상주하는 세그먼트는 그렇지 않은 Cold 그룹의 세그먼트에 비해, 삭제 연산의 빈도가 많아져서 플래시 메모리의 사용에 있어서 양극화 현상을 유도할 수 있다. 이렇게 양극화된 세그먼트에 쓰여질 대상 데이터를 적절한 시점에 상대방 그룹의 것으로 바꾸어 기록 한다면 전체적으로 세그먼트를 균등하게 사용할 수 있게 된다. 즉, Hot 그룹이 사용하던 세그먼트에는 Cold 그룹의 파일을 기록하고, Cold 그룹이 사용하던 세그먼트에는 Hot 그룹의 파일을 기록한다. 관련연구에서 지적인 바와 같이 그룹으로 나누어 주는 것은 불필요한 이동을 줄여 클리닝 비용도 줄일 수 있다[3].

Hot 과 Cold 의 구분은 갱신 작업의 빈도수와 이동 횟수에 의해 결정된다. 이 값들은 파일이 플래시 메모리에 처음 기록되었을 때 판별하기란 쉽지 않다. 즉, 새로 기록되는 파일은 그 어느 편에도 속하지 않는 중립적인 성격으로 두고, 파일의 사용 양태를 관찰하여 그룹을 결정하므로 플래시 메모리를 좀더 효과적으로 사용할 수 있게 한다. (그림 1)은 미지근함의 의미하는 Lukewarm 이라는 그룹을 제안하고 있다. 이 그룹에는 모든 파일의 사용 방식이 이분법적 논리로만 국한시킬 수 없는, 즉 Hot 그룹에도 속할 만큼의 빈번한 쓰기가 발생하지는 않는다고 해서 항상 Cold 그룹의 쓰기 특성을 보인다고 할 수 없는 특성을 가진 파일도 포함된다.

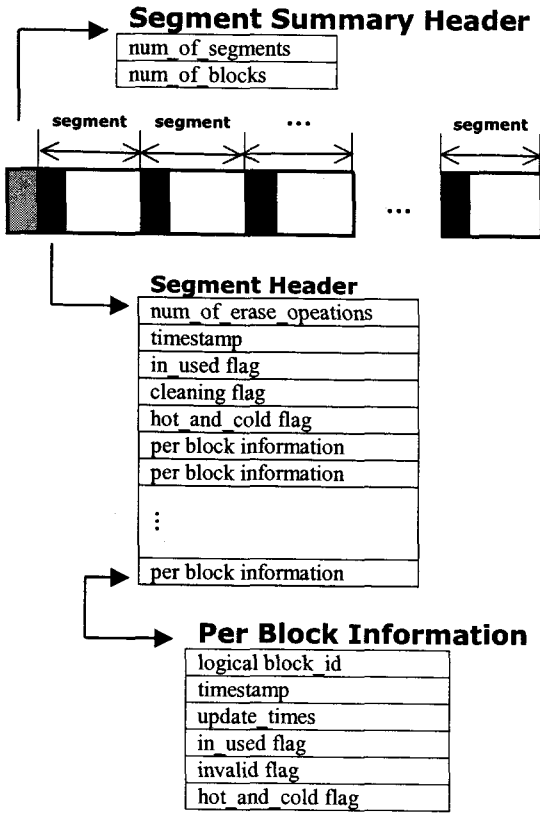
4. 구성

자료의 유형에 따라 Hot, Cold 그리고 Lukewarm 으로 그룹을 나누어 세그먼트의 활용 빈도에 따라 적절하게 할당해주기 위해서는 블록의 갱신 횟수와 이동 횟수, 그리고 세그먼트의 클리닝 횟수 정보를 잘 관리하여야 한다.

이를 위해 블록에는 그 블록에 저장된 자료의 특성을 표현하는 즉, 그 자료가 빈번한 갱신으로 쓰기가 이루어진 Hot 인지, 이동으로 인해 쓰기가 이루어진 Cold 인지 혹은 Lukewarm 인지를 나타내는 Hot-and-Cold 라는 플래그를 (그림 2)와 같이 둔다. 또한, 세그



(그림 1) 플래그 종류



(그림 2) Flash Memory 레이아웃

먼트에 삭제 횟수에 따라 결정되는 또 다른 Hot-and-Cold 플래그를 둔다. (이하, 특별한 설명이 없는 한, 세그먼트의 Hot-and-Cold 플래그는 segment 플래그로, 블록의 Hot-and-Cold 플래그는 block 플래그로 명명한다.)

플래시 메모리의 쓰기 작업은 (그림 3)의 write() 루틴과 같이 수행된다. 새로운 자료는 유형 결정 없이 segment 플래그가 'lukewarm'인 곳에 써주며, 갱신되는 자료는 먼저 유형을 결정한 후에 유형에 맞는 세그먼트의 빈 블록에 쓰기를 해준다. 자료의 유형 결정 과정을 (그림 4)에 기술 하였다. Hot 과 Cold 각각에 대한 threshold 를 미리 정의 하여 유형 결정의 기준으로 삼는다. ① 자료가 쓰여지는 block 플래그가 'hot'으로 바뀌는 경우는 블록의 update_times 플래그가 Hot 의 threshold 를 넘을 때이다. ② 'cold'로 바뀌는 경우는 블록의 timestamp 가 Cold 의 threshold 를 넘을 때 발생한다. 그러나 유형이 결정된 자료라도 시간에 따라 그것이 변할 수 있기 때문에, 계속 그 유형을 관찰하고 만일 필요하다면 'lukewarm'으로 변경해 주어야 한다. 다시 말해, Hot 인 자료가 어느 순간부터는 Cold 의 성질을 띌 수 있고, 그 반대도 될 수 있기 때문에 그런 경우 일단 Lukewarm 으로 처리한 후 그 유형을 관찰해서 다시 결정해 주도록 한다. ③ 'lukewarm'으로 들

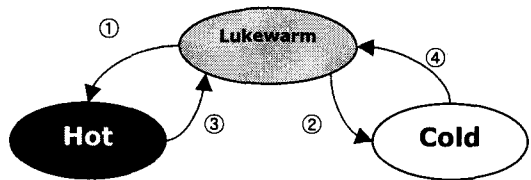
```

write() {
    Allocate a free block;
    If new write
        Write data into the free block of lukewarm segment;
        Set block_flag as 'lukewarm';
    else {
        Mark the obsolete data as invalid;
        Decide block_flag type;
        Write data into the free block according to its type;
    }
}

cleaning() {
    Select a victim segment for cleaning;
    Identify valid data in the victim segment;
    Decide block_flag type;
    Copy out valid data to another clean flash memory spaces according to its type;
    Erase the victim segment;
    Decide segment_type;
}
    
```

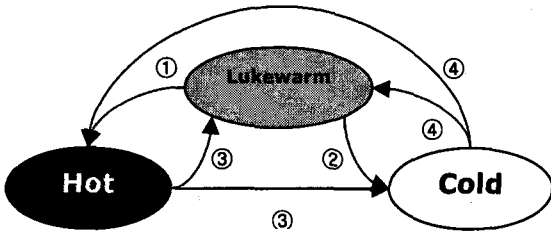
(그림 3) Non-in-Place 갱신과 클리닝 작업

아오는 경우는, block 플래그가 'hot' 일때 블록의 timestamp 가 Cold 의 threshold 를 넘을 경우 이다. ④ block 플래그가 'cold'이고, 블록의 update_times 가 Hot 의 threshold 를 초과할 때에도 발생한다. 클리닝은 (그림 3)의 cleaning() 루틴과 같이 수행된다. 유효한 자료를 아직 보유하고 있는 세그먼트를 클리닝 해야할 수도 있기 때문에, 자료의 보존을 위한 식별작업이 먼저 이루어진다. 유효한 자료가 있다면 그것을 다른 곳에 이동시켜 주어야 하는데, 이동시켜 줄 자료의 유형을 결정하고 쓰는 방식은 쓰기 작업에서의 자료 결정 방식과 쓰기 방식을 따른다. 이렇게 클리닝 작업을 하다 보면 세그먼트마다 클리닝 횟수의 차이가 발생하게 된다. 이 클리닝 횟수의 정도를 판별하여 세그먼트의 유형을 결정해 주는 과정을 (그림 5)에 나타내었다. 세그먼트의 상태를 바꾸어 주는 기준으로 (그림 2)의 num_of_erase_operations 플래그들의 평균값을 기준으로 하는 일정 범위를 정의 해준다. ① segment 플래그를 'hot'으로 바뀌는 경우는 num_of_erase_operations 플래그가 그 기준 범위를 상회 할 때이다. ② 'cold'로 바뀌는 경우는 기준 범위에 미치지 못할 때이다. 이렇게



(그림 4) 블록 유형 전이도

참고문헌



(그림 5) 세그먼트 유형 천이도

세그먼트의 사용이 같은 유형의 자료 별로 이루어진 후, 어느 정도 시간이 지나면 세그먼트 사용에 있어서 양극화가 발생하게 된다. 즉, 'hot'인 세그먼트는 전체 평균 이상의 사용률을 가지고, 'cold'인 세그먼트는 평균에 미치지 못하는 사용률을 가지게 된다. 이 상태가 아무 조치 없이 계속 유지 된다면, 양극화가 지나쳐서 플래시 메모리의 수명이 단축된다. 이것을 방지하기 위해 적절한 시점에 반대의 유형을 지닌 자료가 쓰여지도록 한다. 0 segment 플래그가 'hot'이었던 세그먼트는 클리닝 후 'cold'나 'lukewarm'으로 표시를 해서, 'cold'나 'lukewarm'인 자료의 쓰기를 받도록 한다. 0 마찬가지로, segment 플래그가 'cold'이었던 세그먼트는 'hot'이나 'lukewarm'으로 표시를 해서, 'hot'이나 'lukewarm'인 자료의 쓰기를 받도록 한다. 이렇게 해주는 것은 세그먼트를 전체적으로 균등하게 사용하게 해주기 때문에, 플래시 메모리의 수명을 늘리는 효과를 갖게 한다.

5. 결론

Hot, Cold 그리고 Lukewarm 으로 자료의 성격을 파악하기 위해 갱신과 이동 횟수를 사용 하였고, 이렇게 판별된 그룹별로 자료를 관리하는 것은 플래시 메모리를 좀더 효과적으로 사용 할 수 있다는 것을 보았다. 여기서 블록의 유형을 정확하게 판단하는 것은 Cold 와 Hot 각각의 threshold 를 얼마나 잘 선택하느냐에 달려 있다.

플래시 메모리를 이렇게 관리해주는 것은 균등 사용 이상의 의미를 갖는다. 만약 특정 부분이 전체 평균 보다 더 많은 클리닝 횟수를 지니게 된다면, 다른 부분을 더 사용할 수 있을 지라도 결국 그 특정 부분 때문에 플래시 메모리를 더 이상 사용할 수 없게 될 수 있다. 그렇게 하기 위해 세그먼트의 유형을 변경해 주는 시기를 잘 결정해 주어야 한다. 이렇게 세그먼트를 클리닝 횟수에 따라 적절하게 관리해주는 것은 플래시 메모리의 수명에도 영향을 끼치게 된다.

물론, 플래시 메모리의 물리적인 특성상 획기적인 수명 연장을 기대하기는 어려울지 모르지만, 세그먼트와 쓰여지는 자료의 특성에 대한 연구 조사는 어느 정도의 수명 향상을 가능케 할 수 있다.

[1] Mendel Rosenblum and John K. Ousterhout, "The Design and Implementation of a Log-Structured File System," ACM Transactions on Computer Systems, Vol.10, No.1, pp. 26-52, 1992.

[2] Atsuo Kawaguchi, Shingo Nishioka, and Hiroshi Motoda, "A Flash Memory Based File System," Proceedings of USENIX Technical Conference, New Orleans, LA, pp. 155-164, 1995.

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

[4] H.J. Kim and S.G. Lee, "A New Flash Memory Management for Flash Storage System," Proceedings of the 23rd International Computer Software and Application Conference (COMSAC), 1999.

[5] 김도형, "NAND 형 플래시 메모리의 균등한 사용을 위한 블록 관리 방법," 서울대 석사학위 논문, 2002.

[6] 임대영, "데이터 접근율을 이용한 플래쉬 메모리용 클리닝 방법," 부산대 석사학위 논문, 2002.