낸드 플래시 메모리의 가비지 컬렉션을 위한 적응적인 정책

한규태 ○ 김성조

중앙대학교 컴퓨터공학부

hellogt@konan.cse.cau.ac.kr○ sjkim@cau.ac.kr

An Adaptive Policy for Garbage Collection of NAND Flash Memory

GyuTae Han○ Sungjo Kim

School of Computer Science & Engineering Chung-ang University

요 약

낸드 플래시 메모리를 저장매체로 사용하기 위해서는 제자리 덮어쓰기가 불가능하고 블록의 지움 횟수가 제한된다는 문제점을 해결해야 한다. 이러한 문제점을 해결하기 위해 지움 횟수 평준화를 지원하는 다양한 가비지 컬렉션 정책들이 연구되고 있다. 하지만 기존 정책들은 지움 횟수 평준화를 지원하기 위해 가비지 컬렉션이 수행될 때마다 전체 블록에 대해 지움 대상 블록을 선정하기 위한 클리닉 지표를 구하는 연산을 수행하여야 하고 이 연산들은 시스템의 성능을 저하시킨다. 본 논문은 블록당 지움 횟수의 분산과 블록들의 최대 지움 횟수에 따라 변경되는 임계값을 이용하여 전체 블록에 대한 클리닉 지표를 구하는 연산을 수행하지 않으면서 지움 횟수 평준화를 제공한다. 가비지 컬렉션 시 분산이 임계값 보다 작을 때에는 Greedy 정책을 이용하여 지움 비용을 최소화하고, 분산이 임계값 보다 클 때에는 최대 지움 횟수를 가진 블록을 지움 대상에서 제외하여 지움 횟수를 평준화시킨다. 제안된 방법은 전체 블록에 대한 클리닉 지표를 연산하는 과정을 제거하여 가비지 컬렉션 속도를 향상시킨다.

1. 서 론[1][2]

최근 낸드 플래시 메모리는 전자 산업의 발전으로 가전, 통신기기, 휴대기기 등 다양한 기기의 저장매체로서 사용되고 있다. 낸드 플래시 메모리는 전기적으로 입출력을 수행하기 때문에 읽기 성능은 D램과 거의 같은 수준이며 쓰기 속도도 하드디스크에 비해 수백 배 이상 빠르다. 또한 가볍고 소비전력도 적게 든다.[1,2]

하지만 낸드 플래시 메모리를 저장 매체로 사용하기 위해서는 2가지 제약 조건이 있다. 첫 번째로, 낸드 플래시 메모리는 쓰기 연산을 하기 전에 반드시 해당 블록에 미리 지움 연산을 해주어야 한다. 낸드 플래시 메모리는 각 비트를 한쪽 방향으로만 토글링을 허락하기 때문이다. 이러한 이유로 낸드 플래시 메모리는 같은 주소에 덮어쓰기가 불가능하기 때문에 낸드 플래시 메모리에 데이터를 저장할 때 미리 그 영역을 확보하고 있어야 한다. 두 번째로, 낸드 플래시 메모리를 구성하고 있는 각각의 블록들은 지움 횟수에 제한을 가지고 있다. 허용되는 지움 횟수를 초과한 플래시메모리 블록에 대해서는 쓰기 오류가 발생하기 때문에, 전체 플래시 메모리 공간이 균등하게 활용되지 못하는 경우에는 사용 가능한 저장 공간이 급격히 줄어드는 현상이 발생할 수 있다.

낸드 플래시 메모리를 관리하는 파일 시스템들은 쓰기 공간을 확보하기 위해, 가비지 컬렉션(Garbage Collection)을 수행하게 된다. 그리고 이 과정 중 블록들의 지움 횟수를 균등하게 유지되게 하기 위해 지움 횟수 평준화 기법(wear-leveling)을 사용한다. 가비지 컬렉션이 수행 될 때 지움 대상 블록은 블록이 포함하고 있는 유효한 페이지와 블록의 지움 횟수를 판단하여 블록을 지웠을 때 지움 비용이 적게 들면서 지움 횟수 평준화를 지원하는 블록으로 결정된다.[3,4,5] 기존의 연구에서는 지움 대상 블록을 찾기 위하여 낸드 플래시 메모리에 존재하는 모든 블록들의 지움 횟수에 대한 클리닝 지표를 연산하여 지움 대상 블록을 선정하였다.

본 논문에서 제안하는 가비지 컬렉션 정책에서는 지움 횟수 평준화를 지원하기 위해 블록당 지움 횟수의 분산 사용한다. 그리고 낸드 플래시 메모리를 구성하고 있는 블록들 중 최대 지움 횟수, 블록당 제한된 지움 횟수, 블록들의 개수를 기준으로 분산의 임계값을

[1] 본 연구는 서울시 산학연 협력사업(CR070019) 지원으로 수행되었음.

[2] 본 연구는 지식경제부 및 정보통신 연구진흥원의 대학 IT연구센터(홈네트워크연구센터) 육성,지원사업의 연구결과로 수행되었음.

설정하여 분산이 임계값을 넘지 안도록 유지한다. 분산이 임계값보다 작을 때에는 지움 횟수 평준화를 고려하지 않고 최소의 지움 비용을 들여 빈 공간을 확보한다. 지움 횟수가 특정 블록에 집중되어 분산이 임계값을 넘어가게 되는 경우에는 최대 지움 횟수를 가지는 블록을 지움 대상에서 제외 시켜 지움 횟수를 평준화시킨다. 그리고 무효 페이지를 기준으로 데이터 블록들로 최대 힙(Data Block Heap: DBH)을 구성하여 지움 요청 시 지움 대상이 될 블록을 신속하게 제공한다. 비어 있는 블록들은 지움 횟수를 기준으로 하는 최소 힙(Clean Block Heap: CBH)을 구성하여 쓰기 연산 시 최소 지움 횟수를 가지는 블록을 할당하여 지움 횟수 평준화를 지원한다.

본 논문의 구성은 다음과 같다. 2장은 기존 NAND 플래시 메모리 전용 파일 시스템에서 사용하는 가비지 컬렉션의 문제점에 대해 알아보고, 3장은 본 논문에서 제안하는 가비지 컬렉션 정책에 대하여 설명한다. 마지막으로 3장은 결론과 함께 향후 연구과제에 대해 기술한다.

2. 관련 연구

Greedy 정책[3]은 무효 페이지가 많은 블록을 선택하여 삭제하는 방법이다. 유효한 페이지를 다른 비어 있는 블록에 저장하는 시간이 줄어 들지만 특정 블록에 지움 연산이 집중해서 일어 날 수 있기 때문에 낸드 플래시 메모리의 사용 공간이 급격히 줄어드는 현상이 발생할 수 있다.

Cost-Benefit 정책[3]은 낸드 플래시 메모리의 사용 공간이 급격히 줄어드는 문제점을 개선하기 위하여 고안된 정책이다. 지움의 효율성뿐만 아니라 특정 블록에 대한 집중을 피하기 위해 블록의 최근 지워진 시간을 고려하여, 지울 블록을 선택한다. 식(1)은 Cost-Benefit 정책의 계산 과정을 식으로 나타낸 것이다.

Benefit/cost = age * (1 − u) / 2u ▪▪▪▪▪▪ 식(1)
age : 블록이 가장 최근 지워진 시간
u : 블록의 이용율
1-u : 활용 가능한 페이지의 비율

Cost-Benefit 정책에서 사용하는 최근 지워진 시간자체가 지움 횟수가 적다는 것을 의미하지 않기 때문에 정확한 지움 횟수 평준화를 지원하기 어렵다.

이러한 문제점을 해결하기 위해 RCP(Ranking Cleaning Policy)[4]와 PCP(Plain Cleaning Policy)[5]는 지움 횟수를 사용한다. RCP는 세그먼트들의 등급을 계산하여 값이 큰 순서대로 세그먼트를 지우는 방법으로 지움 비용과 지움 횟수 평준화를 모두 고려한 방식이다. 식(2)는 RCP의 계산 과정을 식으로 나타낸 것이다.

R = A * l / (2v * f * e) ▪▪▪▪▪▪ 식(2)
v : 세그먼트 내에서 유효 블록에 대한 비율
f : 비어 있는 블록에 대한 비율
l : 무효 블록에 대한 비율
e : 세그먼트의 전체 지움 가능 횟수에 대한 현재 지운 횟수의 비율
A : e에 대한 가중치

RCP는 플래시 메모리의 수명이 연장될 수 있지만, 상수 A의 값이 상황에 맞게 결정되어야 하며, 가비지 컬렉션이 발생할 때마다 지워야 하는 블록을 찾기 위해 모든 블록에 대한 R값을 구해야 하며 이 연산은 전체 시스템의 성능을 저하시키게 된다.

PCP는 RCP의 연산 과정을 줄이며, 지움 횟수 평준화를 지원하도록 하였다. 식(3)은 PCP의 계산 과정을 식으로 나타낸 것이다.

C = (2 * V) + F
R = EA / C ▪▪▪▪▪▪ 식(3)
C : 세그먼트 삭제 비용
(2 * V) : 유효한 데이터를 다른 세그먼트로 옮기는 비용과 지우는 비용
F : 지워질 세그먼트의 빈 공간을 지워 낭비되는 비용
EA : 플래시 메모리의 지움 횟수 한계치와 현재 지움 횟수의 차이 값

PCP는 RCP에 비하여 연산 시간을 줄였지만, 두 개의 임계값을 사용자가 상황에 맞게 설정해주어야 한다. 그리고 이 정책도 가비지 컬렉션이 발생할 때마다 지워야 하는 블록을 찾기 위해 모든 블록에 대한 R값을 구해야 하기 때문에 전체 시스템의 성능을 저하시키게 된다.

3. 가비지 컬렉션 정책

본 논문에서는 분산과 분산의 임계값을 이용하여 가비지 컬렉션 시 지움 횟수 평준화를 지원하고, DBH와 CBH 두 개의 힙을 구성하여 쓰기 대상이 되는 블록과 지움 연산의 대상이 되는 블록을 빠르게 할당한다. 그리고 쓰기 시간뿐만 아니라 Idle한 시간에 가비지 컬렉션을 수행하여 쓰기시간에 수행될 수 있는 가비지 컬렉션 시간을 줄인다.

3.1 분산과 분산의 임계값

블록들의 지움 횟수를 기준으로 한 표준편차 값은 블록들의 지움 횟수가 얼마나 평준화 되었는지를 보여주는 지표가 된다. 실제로 연산 과정에서는 표준 편차

대신 분산을 사용한다. 표준 편차를 구하기 위해서는 분산에 제곱근 연산을 해야 하고, 이 제곱근 연산은 반복해서 수행할 경우 큰 오버헤드를 발생 시킬 수 있기 때문이다. 본 논문에서는 분산이 낸드 플래시 메모리의 블록들의 최대 지움 횟수에 따라 변경되는 임계값보다 커지면 지움 횟수 평준화를 위한 작업을 수행한다.

분산을 구하는 공식은 식(4)와 같다.

$$variance = \frac{\sum_{i=1}^{n}(E_i - ave)^2}{n-1} = \frac{\sum_{i=1}^{n}E_i^2}{n-1} - ave^2 \quad \cdots\cdots \text{식(4)}$$

n : 낸드 플래시 메모리를 구성하는 전체 블록의 수
E_i : i번째 블록의 지움 횟수
ave : 블록들의 지움 횟수 평균

하지만 이 공식을 직접 적용할 경우 지움 횟수가 증가 할 때 마다 모든 블록들의 지움 횟수를 평균에서 빼주는 연산을 수행해야 한다. 이 연산은 블록의 개수가 n개일 때, O(n)의 연산 시간이 필요하다. 하지만 한번에 하나의 블록이 지워진다는 점을 이용하면 평균과 분산을 식(5), 식(6)과 같은 방법으로 구할 수 있다.

$$ave_{new} = ave_{old} + 1/n \quad \cdots\cdots \text{식(5)}$$

$$variance_{new} = variance_{old} + (-E_k^2/(n-1) + (E_k+1)^2/(n-1)) + (ave_{old}^2 - ave_{new}^2)$$
$$= variance_{old} + ((2E_k+1)/(n-1)) - ((2ave_{old}+1/n)/n) \quad \cdots\cdots \text{식(6)}$$

E_k : 지워지는 블록의 지움 횟수

낸드 플래시 메모리의 각 블록은 지움 횟수가 제한된 지움 횟수를 넘어가게 되면 더 이상 사용이 불가능해진다. 따라서 낸드 플래시 메모리를 구성하는 블록들의 최대 지움 횟수가 제한된 지움 횟수가 증가할수록 수명이 줄어든다고 정의할 수 있다. 긴 수명을 보장하기 위해서는 블록들의 지움 횟수가 특정 블록에 집중되지 않고, 모든 블록들에 고르게 분포 되어 있어야 한다. 분산이 작아질수록 낸드 플래시 메모리를 구성하고 있는 블록들의 지움 횟수의 평준화가 잘 이루어지고 있다는 것을 의미하기 때문에, 낸드 플래시 메모리의 수명이 다해 갈 때에는 분산이 0에 가까워져야 수명을 최대한 보장했다고 할 수 있다. 이를 지원하기 위해 분산을 조절하는 임계값은 낸드 플래시 메모리의 수명이 다해 갈수록 0에 가깝게 설정 되어야 한다. 그리고 분산이 전체 블록의 개수와 블록당 최대로 지울 수 있는 횟수에 따라 달라 지기 때문에 이 점들을 반영하여 분산의 임계값을 설정해야 한다, 낸드 플래시 메모리의 남은 수명은 블록당 최대 지울 수 있는 횟수를 LN, 현재 블록들 중 최대 지움 횟수를 MN이라 했을 때, (LN-MN)/LN으로 연산하여, 0과 1사이의 숫자로 표현 할 수 있다. 그리고 낸드 플래시 메모리를 구성하고 있는 블록들의 수를 NB라 했을 때, 분산은 최대 $NB*(LN/2)^2/(NB-1)$이 까지 나타날 수 있다. 이 두 식을 가지고 수명이 최대 일 때, 최대의 분산을 허용하고 수명이 다해 갈수록 최소의 분산이 되도록 하기 위해 $(LN-MN)/LN*NB*(LN/2)^2/(NB-1)$이라는 식으로 표현 할 수 있다. 분산은 대상 값들의 개수가 많아질수록 최대 값과 최소 값의 차이를 민감하게 반영하지 못한다. 이는 분산을 구하는 과정에 (NB-1)로 나누어 주는 부분이 있기 때문이다. 위에서 구한 임계값을 낸드 플래시 메모리를 구성하는 블록의 개수에 따라 민감하게 반영하도록 하기 위해서는 임계값 또한 (NB-1)로 나누어주어야 한다. 위의 식들을 조합하면 $(LN-MN)/LN*NB*(LN/2)^2/(NB-1)/(NB-1)$와 같고, 이 식을 정리하면 식 (7)와 같다.

$$Th_v = (LN-MN)/LN*NB*(LN/2)^2/(NB-1)^2 \quad \cdots\cdots \text{식(7)}$$

식(7)에서 NB, LN은 변하지 않는 상수 값이므로 낸드 플래시 메모리를 구성하고 있는 블록들의 최대 지움 횟수(MN)가 변경될 때에만 임계값이 변경된다.

블록들의 최대 지움 횟수가 0인 처음에 임계값은 가장 큰 수를 나타내게 되고, 이 때는 분산을 최대한 허용하여 지움 횟수 평준화를 고려하지 않는다. 하지만 블록들의 최대 지움 횟수가 블록당 최대 지울 수 있는 횟수에 가까워지게 되면 임계값은 0에 가깝게 설정되어, 지움 횟수 평준화를 엄격하게 고려하게 된다.

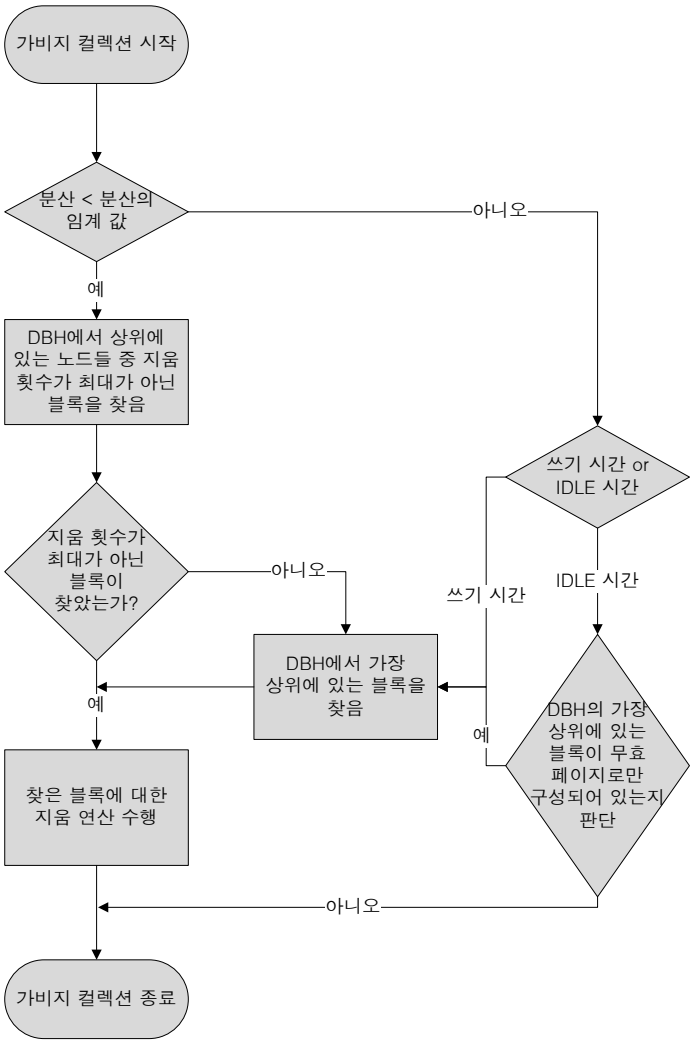
3.2 힙 구성

낸드 플래시 메모리에 빠른 쓰기를 지원하기 위해서는 비어 있는 블록을 신속하게 제공해야 한다. 본 논문에서는 비어있는 블록들을 대상으로 지움 횟수에 따른 최소 힙을 구성한다. 새로운 데이터를 낸드 플래시 메모리에 저장할 때 CBH의 루트에 있는 블록을 특별한 연산 없이 신속하게 할당하고, 최소 지움 횟수를 가지는 블록을 할당함으로써 지움 횟수 평준화를 지원한다.

블록을 지우기 위해서는 블록들이 가지고 있는 유효 페이지를 다른 블록에 복사하는 작업이 필요하다. 이 작업은 페이지를 읽고, 쓰는 작업을 해야 하므로 많은 시간이 소비된다. 따라서 가비지 컬렉션 시 무효 페이지를 최대로 가지는 블록을 지워야 한다. 전체적인

블록의 정보를 검색하지 않고 지움 대상이 되는 블록을 찾기 위하여 무효 페이지를 기준으로 최대 힙을 구성한다. CBH에 존재하는 블록에 데이터를 저장하게 되면 해당 블록은 DBH로 이동하게 된다. DBH에 존재하는 블록들은 무효 페이지가 발생하면 부모 노드에 있는 블록의 무효 페이지 수와 비교하여 많을 경우 부모 노드와 교체된다.

3.3 가비지 컬렉션 수행 과정



(그림 1) 가비지 컬렉션 수행 과정

가비지 컬렉션은 보통 쓰기 연산을 할 때, 비어 있는 블록이 부족할 경우 발생한다. 본 논문에서는 쓰기 연산을 할 때 뿐만 아니라 낸드 플래시 메모리가 Idle할 때도 가비지 컬렉션을 수행하여 쓰기 시간에 수행될 가비지 컬렉션에 대한 부담을 덜어준다. 전체적인 가비지 컬렉션 수행 과정은 (그림 1)과 같다.

쓰기 시간에는 현재 블록 지움 횟수들의 분산이 임계값을 넘었는지, 넘지 않았는지 파악한다. 넘었을 경우는 블록들의 지움 횟수를 평준화 하기 위하여 최대 지움 횟수를 가지는 블록을 지움 대상블록에서 제외한다. 만약 분산이 임계값을 넘지 않았을 경우는 지움 횟수의 평준화를 고려하지 않아도 되므로 무효페이지가 가장 많은 페이지를 삭제하여 빠르게 가비지 컬렉션을 수행한다.

낸드 플래시 메모리가 2초 동안 Idle할 경우, 다시 4초간 Idle할 확률은 95%이상이다.[6] 낸드 플래시 메모리가 Idle할 경우 가비지 컬렉션을 수행 하게 되면 쓰기 시간에 가비지 컬렉션 발생을 줄일 수 있어, 쓰기 시간을 줄이는 효과를 가진다. Idle시간에도 블록 지움 횟수들의 분산이 임계값을 넘었는지, 넘지 않았는지 파악하며, 넘었을 경우 쓰기 시간과 같이 최대 지움 횟수를 가지는 블록을 지움 대상블록에서 제외하고, 블록 지움 연산을 수행하여 블록들의 지움 횟수를 평준화한다. 만약 넘지 않았을 경우는 무효 페이지들로만 구성된 블록들을 지우고, 무효 페이들로만 구성된 블록이 존재하지 않을 경우 가비지 컬렉션을 종료하게 된다.

4. 결론 및 향후 연구 과제

기존의 지움 횟수 평준화 기법들은 블록을 지우기 전 모든 블록들의 클리닉 지표를 구해 지워야 할 블록을 찾기 때문에 연산 시간이 많이 필요하다. 또한 낸드 플래시 메모리의 수명이 많이 남아있음에도 불구하고, 처음부터 지움 횟수 평준화를 위한 연산을 수행하므로 가비지 컬렉션의 수행 속도를 느리게 한다.

본 논문에서는 전체적인 분산과 분산의 임계값을 사용하여 낸드 플래시 메모리의 수명이 줄어들수록 지움 횟수 평준화를 엄격하게 지원하였으며, 이 과정에서 전체 블록에 대한 클리닉 지표를 연산하는 과정을 제거하여 가비지 컬렉션 속도를 향상시켰다. 그리고 지움 대상 블록과 비어 있는 블록을 빠르게 찾기 위해 두 개의 힙을 제안하였고, 낸드 플래시 메모리를 사용하지 않는 시간인 Idle 시간을 활용하여 쓰기 시간에 발생할 수 있는 가비지 컬렉션을 수행함으로써 쓰기 시간에 발생 할 수 있는 가비지 컬렉션의 오버헤드를 줄였다.

향후 과제로 본 논문에서 제시한 가비지 컬렉션 정책을 플래시 메모리 파일 시스템에 직접 구현하여 성능을 측정하여 보는 연구가 진행되어야 한다.

5. 참고 문헌

[1] 민용기, 박승규, “이동컴퓨터를 위한 플래시메모리 클리닝 정책,” 한국통신학회논문지, 제24권 5호, pp. 657~666, 1999

[2] M. L. Chang, P. C. H. Lee, R. C. Chang, “Managing Flash Memory in Personal Communication Devices,” Proceedings of IEEE Symp. On Consumer Electronics, pp.177-182, 1997

[3] Atsuo Kawaguchi, Shingo Nishioka, Hiroshi Motoda "A Flash-Memory based File System," Proceedings of USENIX Technical Conference, pp.155-164,1995.
[4] 김정기, 박승민, 김채규, "임베디드 플래시 파일 시스템을 위한 순위별 지움 정책," 정보처리학회논문지, 제9-A권, 제4호, pp.399-404, 2002년
[5] 이태훈, 이상기, 정기동 , "임베디드 플래시 파일 시스템을 위한 플레인 지움 정책," 한국정보과학회 2005 추계학술 발표 논문집(1), pp. 778-780, 2005
[6] T. Blackwell, J.Harris and M. Seltzer, "Heuristic Cleaning Algorithms in Log-Structured File Systems," Proceedings of the 1995 Winter Usenix, January 1995