



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학석사 학위논문

고집적도 플래시 메모리에서의
Read Reclaim을 고려한
Wear Leveling 기법 제안

2013년 2월

서울대학교 대학원

전기컴퓨터공학부

박 경 진

고집적도 플래시 메모리에서의
read reclaim을 고려한
wear leveling 기법 제안

지도교수 김 지 흥

이 논문을 공학석사 학위논문으로 제출함
2013년 2월

서울대학교 대학원
전기컴퓨터공학부
박 경 진

박경진의 석사 학위논문을 인준함
2013년 2월

위 원 장 하 순 회 (인)

부위원장 김 지 흥 (인)

위 원 Mckay (인)

초 록

플래시 메모리는 하드 디스크에 비하여 읽기/쓰기 성능이 우수하며 충격에 강할 뿐 아니라, 저전력이라는 장점을 바탕으로 모바일 디바이스 뿐 아니라 서버 시스템에도 사용 되는 등 점차 사용 영역을 넓혀 가고 있다. 그러나 플래시 메모리는 하드 디스크와 달리 쓰기/삭제 횟수에 제한이 있어 쓰기 횟수를 줄이는 연구와 블록별 쓰기 횟수를 고르게 맞추는 wear leveling에 관한 연구가 계속 되어왔다.

그러나 최근 플래시 메모리의 직접도가 증가함에 따라 플래시 메모리의 최대 삭제 횟수도 급격하게 감소하였으며, 특정 블록에 읽기 연산이 누적되면 인접 셀이 프로그램되는 read disturb 문제가 심각해짐에 따라 read reclaim이라는 새로운 삭제 연산이 등장하게 되었다. 이러한 상황에서 저장장치의 수명과 용량을 보장하기 위하여 wear leveling의 역할이 어느 때보다도 중요해졌다.

따라서 본 논문에서는 read reclaim에 대한 정보를 이해하고 있는 read aware wear leveling을 도입함으로써, 기존의 wear leveling의 한계를 극복함으로써 wear leveling 으로 인한 부가 삭제 횟수를 줄이고, 기존의 wear leveling 성능은 유지하고자 하였다. 최적화 기법까지 도입하였을 때, wear leveling으로 인한 부가 삭제 횟수는 평균 69% 감소 시켰으며 전체 삭제 횟수는 평균 13% 감소시켰다.

주요어 : flash memory, wear leveling, read disturb

학 번 : 2011-20833

목 차

제 1 장 서 론	1
1.1 연구 배경	1
1.2 논문 구성	4
제 2 장 Read disturb 소개	5
2.1 read disturb의 소개	5
2.2 read reclaim의 소개	7
제 3 장 관련연구	9
3.1 기존 wear leveling 기법의 소개	9
3.2 절 기존 wear leveling 기법의 한계점	12
제 4 장 read reclaim과 wear leveling 비효율성 ...	17
4.1 read reclaim과 wear leveling 시기	17
제 5 장 읽기를 고려한 wear leveling 기법	20
5.1 데이터 특성을 고려한 wear leveling	20
5.2 wear leveler 최적화	26
제 6 장 실험 결과	26
6.1 실험 환경	26
6.2 실험 결과	28
제 7 장 결 론	36
7.1 결 론	36
7.2 향후 연구	37

참고 문헌 38

그림 목 차

[그림 1] wear leveling 여부에 따른 블록별 삭제 횟수	1
[그림 3] Read disturb	5
[그림 4] 블록 당 감내 할 수 있는 읽기량 추세	7
[그림 6] read reclaim 여부에 따른 wear leveling으로 인한 블록 삭제 횟수	12
[그림 6] read reclaim이 없을 때 wear leveling 후, 해당 블록이 삭제되는 시간	13
[그림 7] read reclaim이 있을 때 wear leveling 후, 해당 블록이 삭제되는 시간	13
[그림 8] read reclaim후 WL 희생블록 선택까지 거리	17
[그림 9] 각 요소별 복사 비용	18
[그림 10] 페이지 특성 별 wear leveling 1	18
[그림 11] 페이지 특성 별 wear leveling 2	22
[그림 12] 삭제 클러스터링	24
[그림 13] wear leveling 후, 삭제 될 때까지의 시간이 old 블록이 더 짧은 경우	28
[그림 14] multi1에서의 블록별 삭제 횟수	29
[그림 15] multi2에서의 블록별 삭제 횟수	29
[그림 16] 트레이스 별 wear leveling으로 인한 블록 삭제 수	38
[그림 19] read reclaim 후 WL 희생 블록 선택까지의 거리	39
[그림 18] 트레이스별 총 블록 삭제 횟수	39

[그림 19] read reclaim 후 WL 희생 블록 선택까지의 거리	… 32
[그림 20] 트레이스 별 wear leveling으로 인한 블록 삭제 수	… 41
[그림 20] 트레이스 별 wear leveling으로 인한 블록 삭제 수	… 41
[그림 22] 트레이스별 총 블록 삭제 횟수	…………… 42

표 목 차

[표 1] wear leveling 후, 삭제 될 때까지의 시간이 old 블록이 더 짧은 경우	14
[표 2] write cold 데이터 중 read hot인 비율	15
[표 3] 플래시 메모리 사양	15
[표 4] 트레이스 별 읽기/쓰기 량	27

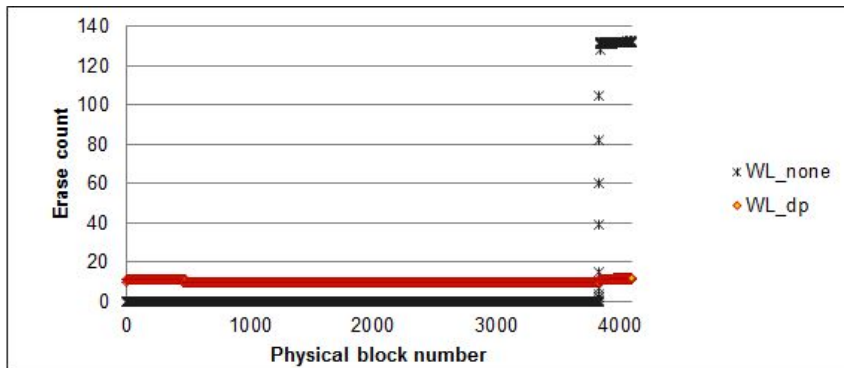
제 1 장 서 론

1. 1 연구 배경

빠른 성능, 내구성, 저전력 등의 장점이 있는 낸드 플래시 메모리 기반의 솔리드 스테이트 디스크(SSD: Solid state disk)는 하드디스크(HDD)를 대체하는 보조기억 장치로써 모바일 임베디드 시스템부터 서버까지 다양한 시스템에 사용되고 있다.

이러한 낸드 플래시 메모리는 기존의 HDD와 달리 cell당 삭제 횟수의 제한이 있어 이를 효과적으로 관리하는 것이 전통적으로 중요시 되어왔다. SSD 내의 cell들의 삭제 횟수가 고르지 못하여 특정 몇 블록의 삭제 횟수가 한계를 넘게 되면, bad block으로 관리하게 된다.

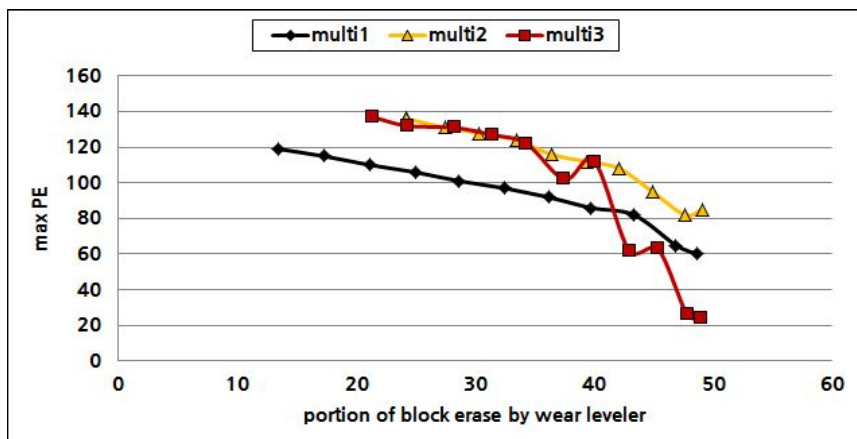
다음의 그림 1에서 보여지 듯 플래시 메모리 내에서 wear leveling을 적용 하지 않은 경우, 블록별 삭제 횟수의 차이가 크게 벌어지게 된다. 이때 두드러지게 삭제 횟수가 많은 특정 블록들의 수명이 다하게 되면 더 이상 사용할 수 없는 블록으로 간주 하여, 이들 블록을 bad block으로 관리하게 되는 것이다.



[그림 1] wear leveling 여부에 따른 블록별 삭제 횟수

그러나 bad block으로 관리 가능한 블록의 용량은 전체 SSD 용량의 3% 이내로 유지하기 때문에[2], bad block이 계속하여 발생하여 3%를 넘어갈 경우 저장장치의 가용 가능한 용량이 지속적으로 감소하는 문제가 발생하게 된다. 따라서 플래시 메모리에서 wear leveling을 이용하여 블록별 삭제 횟수가 균일하도록 관리하는 것이 저장장치의 전체 수명을 위해서 매우 중요하다.

이러한 블록별 삭제횟수의 편차를 조절하는 wear leveling은 데이터의 특성에 맞게 블록 내부에 존재하는 모든 데이터를 이동시키는 연산을 포함한다. 따라서 항상 블록 삭제와 페이지 복사에 대한 부가 연산으로 인한 비용이 존재한다. 다시 말해, 강력하게 wear leveling을 사용하여 블록별 삭제 횟수를 완벽하게 동일도록 조절하게 되면, wear leveling으로 인한 부가 연산으로 인한 비용이 급격하게 증가하게 된다. 이로 인하여, 플래시 메모리 내부의 삭제연산 수와 쓰기 연산 수가 급격하게 증가하여 오히려 플래시 메모리의 수명을 단축시키는데 기여할 수 있다. 따라서 wear leveling의 효과와 부가연산으로 인한 비용의 균형을 맞추는 것이 중요하다고 할 수 있다. 아래의 그림 2에서도 확인 할 수 있듯이, wear



[그림 2] wear leveling 정도에 따른 wear leveling overhead

leveling을 잘하게 되면 최대 삭제 블록의 삭제 횟수는 줄어드나 삭제 overhead가 급격하게 증가하는 현상을 확인할 수 있다. 따라서 블록 별 삭제 횟수를 고르게 만들면서도 부가 연산으로 인한 비용을 줄이는 것이 효과적인 wear leveler라 할 수 있다.

그런데, 최근 낸드 플래시 메모리의 집적도가 증가함에 따라 SLC (single-level cell), MLC (multi-level cell) 뿐만 아니라 하나의 셀에 3bit 이상의 정보를 저장하는 TLC(triple-level cell)도 등장하였다. 이로 인하여 SSD의 용량을 증가시키고 시장 가격 하락이 가능하게 되었다. 그러나, TLC는 내구성과 read disturb측면에서 상당히 취약하다는 단점이 있다. MLC는 최대 삭제 횟수를 3000회 정도 보장하였지만, TLC는 500번 정도 밖에 보장하지 못한다[1]. 뿐만 아니라, read disturb 문제가 점차 심각해지면서[3] read reclaim 이라는 새로운 삭제 연산이 등장하게 되었다.

위와 같이, 블록 당 최대 삭제 횟수가 감소하고 새로운 삭제 연산의 등장으로 인하여 SSD의 효과적인 사용을 위하여 플래시 메모리에서의 효과적인 wear leveling이 더욱 더 중요해졌다.

1. 2 논문 구성

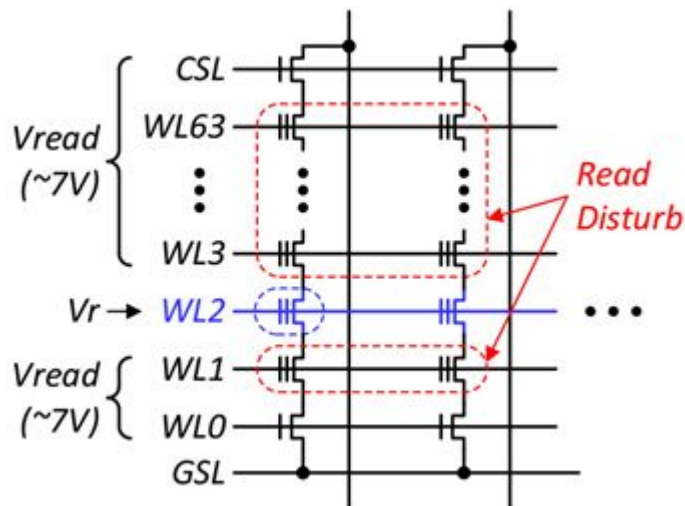
본 논문의 구성은 다음과 같다. 2장에서는 read disturb과 이를 해결하기 위한 read reclaim에 대한 소개를 한다. 3장에서는 기존에 연구되어왔던 wear leveling 기법과 그 한계점에 대해서 언급한다. 4장에서는 읽기 활용율과 read recalim으로 인한 wear leveling 시기를 통해 최적화 가능 여부를 탐색한다. 다음의 5장에서는 읽기를 고려한 wear leveling 기법을 제안한다. 6장에서는 실험 환경과 새로운 wear leveling 기법의 도입으로 인한 수명 향상 및 부하에 대한 결과를 소개한다. 마지막으로 장에서는 결론과 함께 향후 연구에 대해 언급할 것이다.

제 2 장 Read disturb 소개

2. 1 Read disturb

read disturb는 읽기 연산을 수행할 때 플래시 메모리의 구조적인 특성으로 인하여 불가피하게 발생하는 현상을 말한다. 다음의 그림 3은 플래시 메모리 셀의 표현한 것으로, 읽기 연산이 일어날 때의 변화를 도식화한 것이다.

플래시 메모리의 구조는 셀(cell) 배열이 일렬로 연결되어 있는데, 읽기 연산이 수행되면, 그림과 같이 선택된 word line(WL) 2에는 전압 V_r 을 걸고, 선택되지 않은 word-line에는 $V_{read}(= \sim 7V)$ 의 전압이 걸리게 된다. 이러한 전압은 프로그램(약 15V)에 비하여 상대적으로 약한 전압을 걸기 때문에 다른 셀에 직접적인 영향을 끼치지 않으나, 계속해서 읽기 연산이



[그림 3] Read disturb

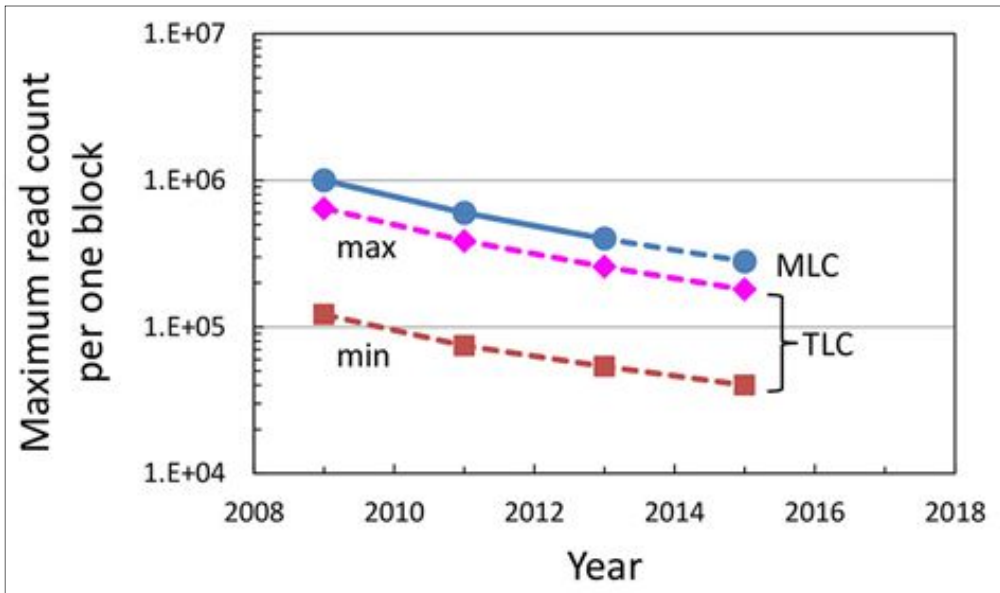
집중되는 경우에는 약한 전압이더라도 지속적으로 영향을 받게 되고, 블록의 삭제 횟수가 증가함에 따라 플래시 메모리의 물리적인 특성이 나빠지게 되면 인접 셀이 원하지 않게 프로그램 될 수 있다. 이러한 현상을 read disturb라고 한다[7].

이러한 블록의 삭제 횟수가 증가 할수록 플래시 메모리의 물리적 성질이 손상되기 때문에 read disturb 현상은 두드러지게 나타난다[10]. 게다가 더 높은 TLC로 갈수록 read disturb 문제가 심각해질 것으로 예측된다.

2. 2 Read reclaim

계속된 읽기 연산이 한 블록에 집중되게 되면, read disturb가 발생하게 되어 인접 셀의 데이터가 손상될 수 있기 때문에, read disturb가 발생하기 전에 플래시 메모리에서는 조취를 취하게 된다. 이러한 조취의 일환으로 read reclaim이 등장하였다. read reclaim은 특정 블록에 읽기 연산이 계속되어 인접 셀에 read disturb 현상이 일어날 수준이 되면, 해당 블록에 존재하는 모든 페이지의 모든 데이터를 다른 블록으로 이동시키고 해당 블록의 데이터를 삭제하는 연산을 수행한다.

30nm 이상의 플래시 메모리에서는 disturb의 영향이 미미했기 때문에 read reclaim이 플래시 메모리에서 중요한 문제로 떠오르지 않았다. 2009년도에 생산된 플래시 메모리도 600,000번의 읽기가 한 블록에 누적되어야 read reclaim이 발생되었다는 것을 보면 그 사실을 알 수 있다[2]. 그러나 점차 낸드 플래시 메모리의 집적도가 올라가 최근 20nm대로 떨어짐



[그림 4] 블록 당 감내 할 수 있는 읽기량 추세 [2]

에 따라 read disturb가 심각해지고 있으며, 한 블록이 감내 할 수 있는 읽기량이 줄어들고 있다.

실험을 통해 나타난 그림 4에서 확인 할 수 있듯이, 집적도가 상향됨에 따라 하나의 블록이 감내할 수 있는 읽기량이 점차적으로 감소하고 있으며, TLC의 경우에는 한 블록 당 5만 번의 읽기도 감내하지 못하는 것을 확인 할 수 있다.

제 3 장 관련 연구

3. 1 기존의 wear leveling 기법

Wear leveling의 가장 큰 목표는 플래시 메모리에 존재하는 블록들 간의 삭제 횟수의 격차를 좁힘으로써 플래시 메모리내의 특정 블록들의 삭제 횟수가 지나치게 상승하여 bad 블록이 되는 것을 억제함으로써 플래시 메모리의 용량을 유지하는 것이다. 기본적으로 wear leveling은 가장 삭제 횟수가 높은 블록 (oldest block)과 가장 삭제 횟수가 적은 블록 (youngest block)에 있는 데이터를 각각 hot 데이터와 cold 데이터로 간주하여 옮김으로써 oldest 블록에는 cold 데이터를 놓음으로써 삭제 횟수 증가를 억제하여 목적을 달성하고자 한다.

기본적인 데이터 특성을 이용한 아이디어를 가진 hot-cold swapping 기법은 oldest 블록에는 hot 데이터가 youngest 블록에는 cold 데이터가 있다는 전제하에, oldest 블록과 youngest 블록을 희생블록으로 고른 후 블록 내에 존재하는 데이터를 교환(swapping)한다[5]. 이러한 방법을 통해, oldest 블록에는 cold한 데이터를 옮김으로써 오래 동안 삭제가 일어나지 않게 되어 더 이상 삭제 횟수가 증가하지 않도록 방지한다. 또한 youngest 블록에는 hot한 데이터가 옮겨짐으로써 잦은 update로 인하여 삭제 횟수가 증가하게 되어 블록간의 wear leveling이 이루어지도록 한다. 그러나 hot-cold swapping은 한 번 oldest 했던 블록이, 다음 wear leveling이 진행 될 때에도 여전히 oldest하여, 계속해서 희생블록으로 선택되는 가능성이 높다. 따라서 소수의 old 블록의 삭제 횟수가 계속해서 증가하여 다른 블록들 보다 월등히 빠른 시간 내에 bad block이 될 수 있

다는 단점이 있다.

이러한 단점을 해결한 방법이 바로 dual-pool 알고리즘이다[6]. dual-pool 알고리즘은 hot-cold swapping 기법의 가장 큰 단점인, 특정 몇 블록이 계속해서 희생블록으로 선택되어 삭제 횟수가 급격하게 증가하는 것을 방지하기 위하여 고안된 기법으로 기본 아이디어는 hot-cold swapping과 유사하나 업데이트가 자주 일어나는 블록의 모음인 hot pool과 업데이트가 자주 일어나지 않는 블록의 모음인 cold pool을 두어 관리한다.

dual-pool은 3가지 방법인 cold-data migration, cold pool adjustment, hot pool adjustment을 통해 wear leveling한다. 먼저 cold-data migration은 cold pool의 youngest 블록과 hot pool의 oldest 블록의 삭제 횟수 차이가 일정 임계치보다 커졌을 때 수행된다. 이의 과정은 cold pool에 있는 youngest 블록의 cold data를 hot pool에 있는 oldest block으로 옮기고, hot pool의 oldest 블록 내에 존재하는 데이터는 새로운 블록을 할당받아 옮김으로써 oldest 블록은 더 이상 삭제 횟수가 증가하지 않도록 억제하도록 하고, youngest 블록은 데이터가 계속해서 들어올 수 있는 기회를 제공한다. 이때 hot-cold swapping의 단점이었던, 계속적으로 특정 블록이 희생블록으로 선정되는 것을 막기 위하여 블록의 pool을 바꾸어주는 데, hot pool에 있던 oldest 블록은 cold pool로 바꾸어 주고, cold pool에 있던 youngest 블록은 hot pool로 바꾸어 주어, 계속해서 희생블록으로 골라지지 않도록 한다.

또한 데이터의 특성 변화를 반영하여 wear leveling의 성능을 향상시키기 위하여, cold pool adjustment와 hot pool adjustment를 수행하는데, 이는 hot pool에 있는 블록 중 삭제 횟수가 가장 큰 블록과 cold pool에 있는 블록 중 가장 적은 삭제 횟수를 가진 블록의 풀을 교환하고, cold pool에 있는 블록의 최근 삭제 횟수가 hot pool에 있는 블록 중 최근 삭제 횟수가 가장 적은 블록보다 큰 경우 두 pool을 교환하는 방식으로 이루어진

다. 이러한 방법을 통해 hot-cold swapping의 부작용을 방지하고 상대적으로 적은 cost로 효과적인 wear leveling을 할 수 있다.

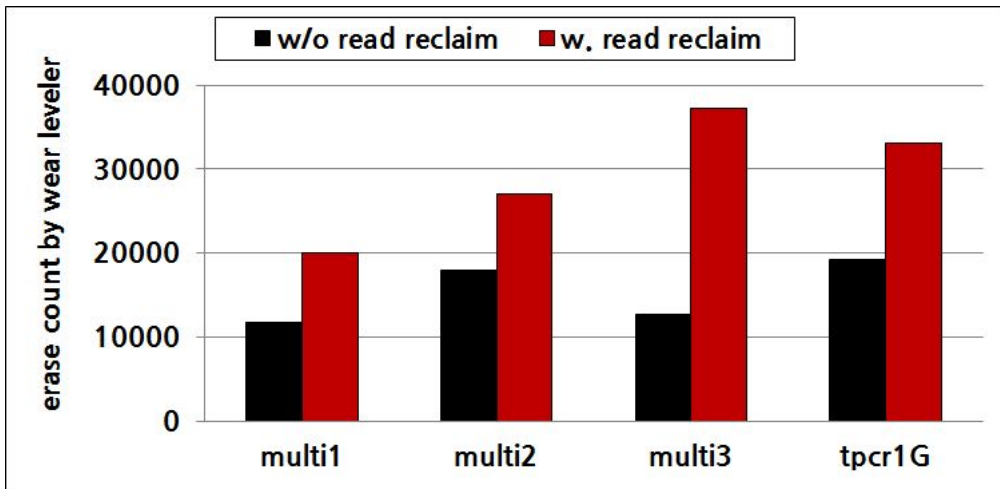
3. 2 기존의 wear leveling 기법의 한계점

앞서 언급한 dual-pool 알고리즘은 부가 연산의 수가 상대적으로 적으며 wear leveling의 성능도 상당히 효과적인 알고리즘으로 알려져 있다. 그러나 read reclaim이 적용된 FTL (Flash Translation Layer)에서 이를 고려하지 못하는 dual pool은 다소 비효율적인 면을 보인다. 이를 확인하기 위해, read reclaim이 존재하는 FTL 환경에서 앞서 언급하였던 wear leveling 기법인 dual pool 알고리즘을 단순히 적용해 보았다.

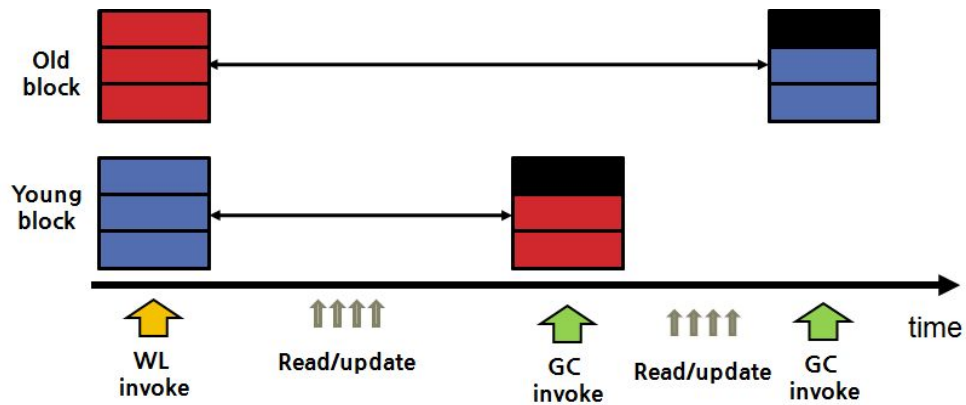
아래의 그림 6은 기존의 read reclaim이 존재하지 않았던 환경(w/o read reclaim)과 read reclaim이 유효한 환경(w/ read reclaim)에서의 wear leveling으로 인한 삭제 overhead를 나타낸다.

아래의 그림 6에서 보듯이, read reclaim이 유효한 환경에서의 wear leveling으로 인한 erase overhead가 급격하게 증가하는 것을 확인할 수 있다. 이러한 wear leveling으로 인한 삭제 overhead의 증가가 wear leveler가 read reclaim 고려하지 않아 발생한 부작용 인지 분석해 보았다.

기존의 wear leveling은 oldest 블록에 cold한 특성을 가진 데이터를 옮



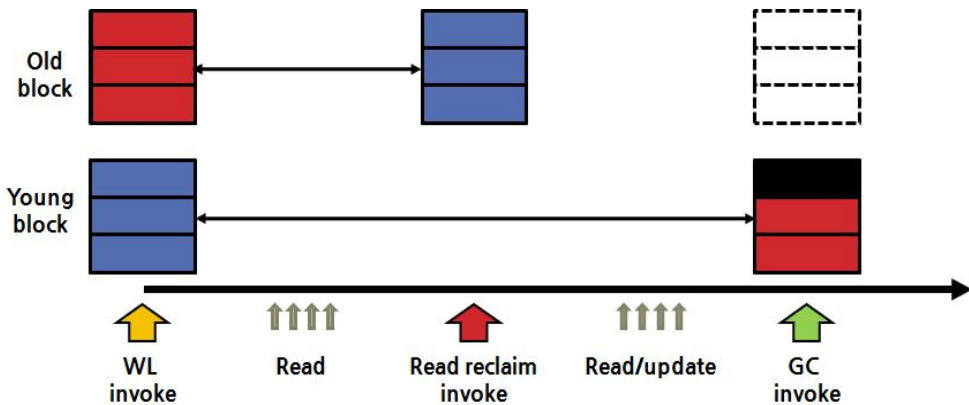
[그림 5] read reclaim 여부에 따른 wear leveling으로 인한 블록 삭제 횟수



[그림 6] read reclaim이 없을 때 wear leveling 후, 해당 블록이 삭제되는 시간

김으로써 oldest 블록의 삭제 횟수가 오랜 시간동안 증가하는 것을 억제하고 youngest 블록에 hot한 특성을 가진 데이터를 옮김으로써 youngest 블록에서 빈번한 삭제가 발생하게 되어 블록별 삭제 횟수의 정도가 유사해 지도록 유도한다.

그러나 read reclaim이 존재하는 FTL 환경에서, 기존의 wear leveling 알고리즘(hot-cold swapping, dual-pool)은 모두 read reclaim을 예측할 수 없다. 따라서 기존의 wear leveler가 cold한 특성을 가졌다고 판단한 데이터를 oldest 블록에 옮김으로써 oldest 블록의 삭제 횟수 상승을 저지하려 한다. 그러나 기존의 wear leveling 알고리즘은 쓰기만을 고려하기 때문에 읽기 특성을 고려하지 않아, 예측하지 못한 read reclaim이 발생하게 된다. 이로 인하여 wear leveling은 oldest 블록으로 간 데이터가 오랜 시간 동안 oldest 블록에 머물기를 기대하지만, read reclaim으로 인하여 예상보다 더 빠르게 블록이 삭제된다. 이러한 경우, oldest 블록이 youngest 블록 보다 빠르게 빈 블록(free block)이 되게 되는데, 이로 인하여 빈 블록이 된 oldest 블록에 새로운 데이터가 유입되는 데, 이때 hot 데이터의 유입 가능성이 상당히 높아지게 되어 블록의 삭제 횟수가 급격하게 증가할 수 있다는 문제점이 생긴다. 이는 블록들 간의 삭제 횟



[그림 7] read reclaim이 있을 때 wear leveling 후, 해당 블록이 삭제되는 시간

수 차이를 더 증가시킴으로써 wear leveling이 계속해서 발생하게 되는 원인을 제공하게 되고 이로 인하여 wear leveling으로 인한 삭제 overhead가 증가하게 되는 것이다.

앞서, oldest 블록으로 옮겨간 cold한 특성을 가진 데이터가 read reclaim으로 인하여 wear leveler의 예상 보다 빨리 삭제되는 현상이 실제로 wear leveling의 삭제 overhead를 발생시킨다는 것을 설명하였다. 그러므로 이러한 현상이 발생하는 정도를 분석하여 실제 FTL에서 영향을 알아보았다.

이 현상의 결과가 아래의 표1 과 같다. multi1, multi2, multi3에서는 wear leveling 후, oldest 블록으로 옮겨간 데이터가 상대적으로 빨리 지워지는 현상이 두드러지게 나타나는 것을 확인할 수 있으며, 전체 평균 55%의 경우에 이러한 현상이 발생하여 wear leveling의 효과를 미미하게

trace	multi1	multi2	multi3	tpcr
w. Read reclaim	0%	0%	0%	0%
w/o Read reclaim	71%	68%	49%	13%

[표 1] wear leveling 후, 삭제 될 때까지의 시간이 old 블록이 더 짧은 경우

만들고 있음을 확인 할 수 있다. 이러한 결과는 계속해서 블록간의 삭제 격차를 벌이는데 기여를 하게 되고 이로 인하여, wear leveling을 계속적으로 발생시키는 것으로 예측할 수 있다.

이러한 데이터를 기반으로 구체적으로 살펴보면, youngest 블록에서 oldest 블록으로 옮겨간 데이터 중에는 read reclaim을 야기하는, 다시 말하여 read hot한 특성을 가진 데이터가 wear leveling에 의해 oldest 블록으로 이동한 데이터 중에 상당량 존재할 것이라는 것을 유추 할 수 있다. 이에, 본 논문에서는 youngest 블록에서 oldest 블록으로 옮겨가는 write cold한 특성을 가진 데이터라고 간주되어 옮겨지는 데이터 중에서 읽기의 특성 측면에서는 hot하다고 판단되는 데이터의 비중을 알아보았다. 다음의 표 21는 실제 시뮬레이션 당시 youngest 블록에 있던 데이터 중 read hot한 특성을 가진 데이터의 비율을 나타내며, 이때의 데이터의 hot/cold 분류 기준은 해당 데이터의 읽기 속도가 전체 트레이스의 평균 읽기 속도 보다 빠른 경우를 hot한 특성을 가진 데이터라고 분류한 결과이다.

아래의 표 2의 결과를 기반으로 보았을 때, write cold 하다고 판단된 데이터 중 35%이상의 데이터가 read hot한 데이터라는 것을 확인 할 수 있다. 이를 통하여, 기존의 쓰기만을 고려한 wear leveler에는 한계가 있음을 예상할 수 있고, 또한 read reclaim이 발생하는 환경에서는 wear leveler가 데이터의 read hot/cold에 대한 특성 정보와 read reclaim에 대한 정보를 알게 됨으로써, read reclaim을 예상하게 되어 앞서 언급하였던

Trace	mutl1	multi2	multi3	T-pcr
write cold data 중 read hot data 비율	63.0%	79.7%	56.3%	26.5%

[표 2] write cold 데이터 중 read hot인 비율

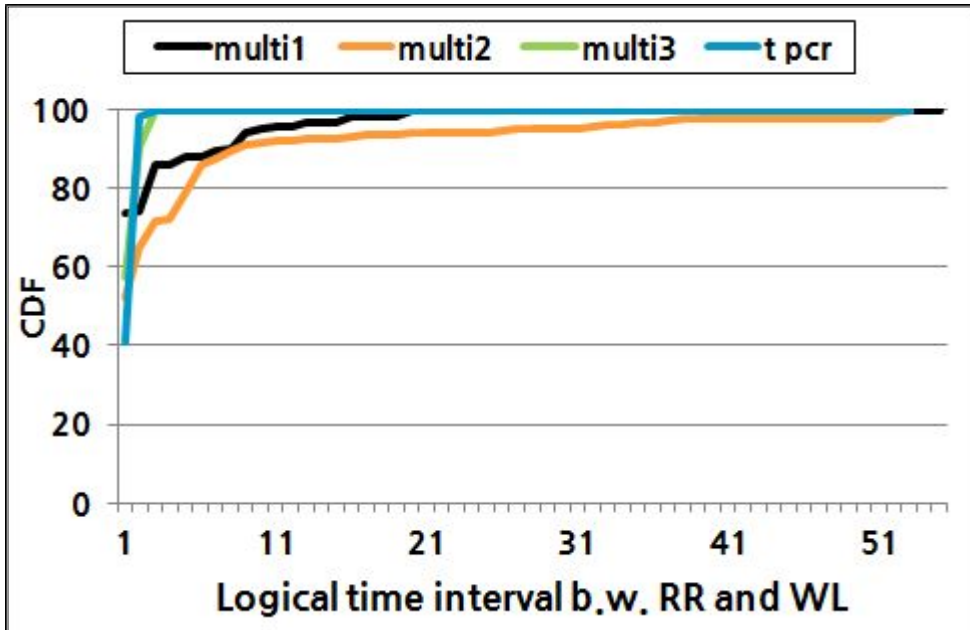
old 블록에서의 삭제가 wear leveler의 예상보다 빠랐던 문제를 해결함으로써 wear leveler가 지속적으로 야기되는 현상을 막음으로써, wear leveler로 인한 부가 삭제 연산을 줄일 수 있을 것이다.

제 4 장 read reclaim과 wear leveling 비효율성

4. 1 read reclaim 후 wear leveling 시기

한 블록에 읽기가 누적되면 인근 셀(cell)에 꾸준히 전압이 가해여 주변 셀이 프로그램 되는 현상인 read disturb가 발생하지 않도록 하기 위하여 한 블록에 읽기 연산이 계속되면, read reclaim을 하여 해당 블록 내부에 존재하는 모든 데이터를 다른 블록으로 이동시키고 해당 블록을 삭제하는 연산을 수행한다. 이와 같이 read disturb 문제가 심각해지고, 집적도가 높아짐에 따라 read reclaim의 빈도가 높아짐에 따라, garbage collection 이외에도 새로운 삭제 연산의 추가로 wear leveling을 야기할 수 있는 요인이 한 가지 더 추가 된 것이다. 다시 말해, garbage collection으로 인한 블록의 삭제 횟수 증가 뿐 아니라, read reclaim으로 인한 삭제 횟수 증가로 인하여 블록 간 삭제 횟수 격차가 벌어지게 되고, 이로 인하여 wear leveling의 발생 조건을 만족시키게 되어 wear leveling이 invoke 될 수 있다.

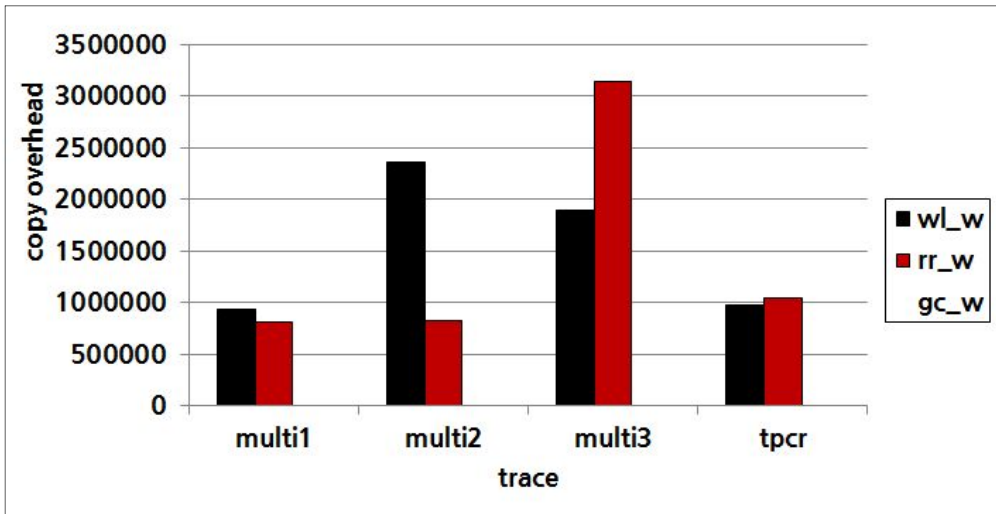
따라서 본 논문에서는 read reclaim이 발생한 후 해당 블록이 앞으로 발생하는 N번째 wear leveling에 의하여 희생 블록으로 선택되어 지는지를 알아보았다. 다음의 그림 8은 이를 나타내고 있는데, 전체 27%의 블록이 read reclaim 후, 5번째 이내에 발생하는 wear leveler에 의하여 희생 블록으로 채택된다. 또한 read reclaim과 wear leveling이 모두 일어나는 블록 중에서는 평균 85%의 블록이 read reclaim 후, 10번째 이내에 발생하는 wear leveler에 의하여 희생 블록으로 채택되는 것을 확인할 수



[그림 8] read reclaim 후, WL 희생 블록 선택까지의 거리

있었다.

위와 같이 read reclaim과 wear leveling 모두 복사 비용이 상당히 높은 두 연산이 짧은 시간 내에 연이어 발생한다는 것을 의미한다. 따라서, 이 두 연산을 클러스터링 하는 것만으로도 상당한 삭제 횟수 절감에 의의가 있을 것이다. 다만, garbage collection과는 함께 클러스터링 하지 못하는 원인은 두 연산의 복사 비용에 있다. garbage collection은 연산 특성 상 한 블록에 유효한 페이지가 적은 페이지를 주로 선택하게 되어 복사 비용을 최대한 낮추는 쪽으로 연구되어 왔었다. 그러나 read reclaim은 복사 비용을 낮추는 쪽으로 블록을 선택할 수 없으며, 실제로 복사 비용이 상당히 높은 연산이다. 다음의 그림 9에서 보듯이 read reclaim과 wear leveling으로 인한 추가 복사 비용은 두 연산 모두 상당히 높은 것을 확인할 수 있으며, garbage collection은 복사 비용이 그래프에서 거의 나타나지 않는 것을 확인할 수 있다. 따라서 garbage collection은 제외하고 복사 비용이 모두 높은 read reclaim과 wear leveling이 연이어 발생하는 것



[그림 9] 각 요소별 복사 비용을 저지함으로써 전체 플래시 메모리의 삭제 횟수를 감소시킬 수 있을 것이라 기대한다.

제 5 장 읽기를 고려한 wear leveling 기법

본 장에서는 앞서 언급한 기존 wear leveling 기법의 문제점을 극복하는 읽기를 고려한 wear leveling 기법에 대해 이야기 하고자 한다.

5. 1 데이터 특성을 고려한 wear leveling 기법

5. 1. 1 데이터의 특성 파악

기존에는 데이터 분류를 쓰기 연산을 기준으로 하였기 때문에, write hot과 write cold로 나누었으나 read disturb 문제가 심각해짐에 따라, read reclaim가 빈번하게 등장하여, read에 대한 데이터 특성도 고려해야 할 필요가 생겼다. 다시 말하여, 쓰기 특성의 관점에서는 cold한 특성을 가지지만 읽기의 측면에서는 hot한 특성을 가진 데이터는 read reclaim으로 인한 삭제를 야기하기 때문에 플래시 메모리를 관리하는 데 있어 영향을 끼치기 때문이다. 따라서 기존과 달리 본 논문에서는 멀티 함수에 기반을 둔 bloomfilter를 이용하여 기존에 write cold라고 분류하였던 데이터를 읽기 연산의 측면에서 분석하여[4], read very hot, read hot, read write cold로 4가지로 데이터의 특성을 분류한다.

따라서 본 논문에서 제안하는 wear leveling 기법은 선택된 희생 블록 내의 데이터를 write hot (WH), read very hot (RVH), read hot (RH), write read cold (WRC)로 분류하여 특성에 맞게 데이터를 분류한다.

write hot을 기존과 같이 우선적으로 분류하는 데, 이는 write의 연속성을 유지함으로써 garbage collection으로 인한 부가 복사 비용이 증가하지

않도록 방지하기 위함이다. 다시 말하여, 본 논문에서는 write cold한 데이터를 더 세분화 시켜 분류함으로써, write의 hotness를 헤치지 않고 read의 특성을 고려하고자 한다.

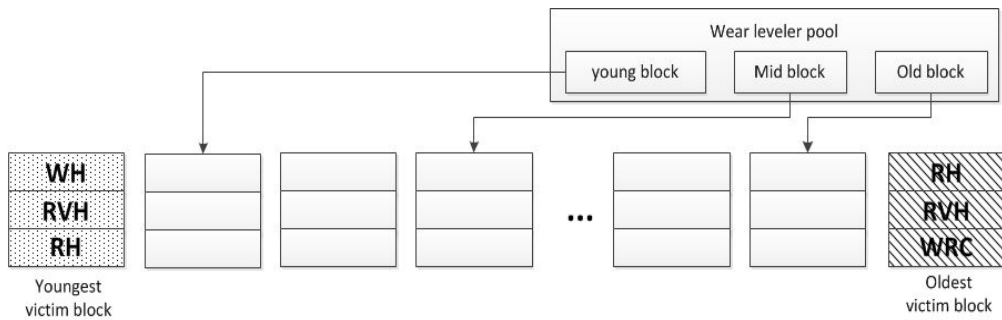
또한 데이터의 분류 시 멀티 해쉬 함수 기반의 multiple bloomfilter를 사용함으로써 데이터 특성의 빠른 탐색과 공간 효율성을 확보 하여 read에 대한 데이터 특성을 파악하는 overhead를 없앨 수 있다[4].

write cold한 데이터 중에서 read hot과 cold는 write와 같은 기준으로 분류하며, read very hot는 read hot한 데이터를 read hot과 cold로 나눈 기준으로 분류한다.

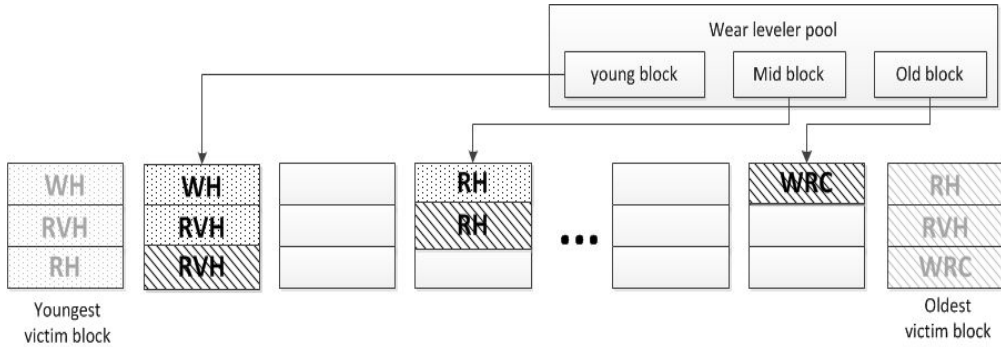
5. 1. 2 페이지 단위의 wear leveling 기법

페이지 단위의 wear leveling은 우선 희생 블록 선정 방법에 따라 선택된 희생 블록을 선정한다. 구체적인 선정 방법은 wear leveling의 다른 기법들과 마찬가지로 oldest 블록과 youngest 블록을 택하되, 계속해서 소수의 블록이 희생 블록이 선택되지 않도록, 최근에 wear leveling을 수행한 블록은 희생블록 후보에서 제외한다.

희생 블록을 선정할 후, 해당 희생 블록내에 존재하는 데이터의 특성을 앞서 언급한 multiple bloomfilter를 통해 데이터의 특성을 분류하는데, 이



[그림 10] 페이지 특성 별 wear leveling 1



[그림 11] 페이지 특성 별 wear leveling 2

를 그림 10와 같이 write hot (WH) /read very hot (RVH) /read hot (RH) /write read cold (WRC)로 분류한다.

희생 블록 내에 존재하는 유효 데이터의 데이터 특성을 분류한 후, youngest과 oldest 블록 내의 데이터는 데이터 특성 분류에 따라 그림 11과 같이, write hot과 read very hot으로 판단되는 데이터는 wear leveling pool에 존재하는 young 블록으로 옮기고, read hot은 평균치 정도의 삭제 횟수를 지닌 wear leveler pool 내의 mid 블록으로 옮긴다. 마지막으로 write와 read 측면 모두에서 cold인 데이터에 한해서만 wear leveler pool내의 old 블록으로 옮긴다.

5. 1. 3 wear leveler pool 교체 정책

wear leveler pool은 데이터의 연속성을 최대한 유지하면서도 wear leveling을 위한 검색 추가비용을 최대한 줄이기 위해 유지한다.

앞서 언급했듯, wear leveling 희생 블록의 데이터는 wear leveler pool 내에 유지하는 블록으로 옮겨지게 된다. 이때, wear leveler pool 내의 블록이 꽉 찬 경우에는 해당 블록은 wear leveler pool을 나가게 되고, wear

leveler pool 진입 후보 블록이 wear leveler pool내로 들어오게 된다. 그 후, 해당 블록으로 희생블록의 데이터가 옮겨지게 된다.

wear leveler pool 진입 후보 블록은 최근 wear leveling을 수행한 희생 블록으로써, 현재 wear leveler pool 내의 블록으로 데이터를 모두 옮긴 희생 블록은 wear leveler pool 진입 후보 블록으로 채택된다.

단, mid 블록의 경우에는 평균치에 근사한 삭제 횟수를 가지는 블록을 랜덤하게 선택하여 wear leveling pool에 삽입한다.

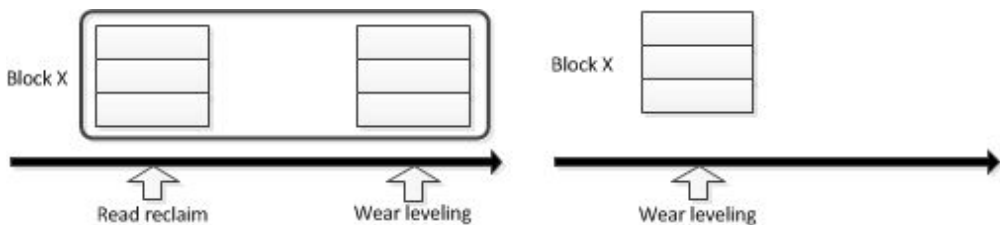
5. 2 wear leveler 최적화

플래시 메모리의 고집적화로 인하여, 특정 블록에 읽기 연산이 누적되면, read reclaim으로 인하여 블록의 삭제 횟수가 증가하게 되고 이를 감지한 wear leveler가 해당 블록을 wear lever의 희생 블록으로 선정하게 된다. 이렇게 되면 부가 연산 비용이 높은 read reclaim과 wear leveling이 연이어서 발생하게 되는데, 이는 성능에도 영향을 줄 수 있으며 삭제 횟수 측면에서도 최적화의 여지가 있음을 알 수 있다.

앞선 절에서 언급했듯이 read reclaim을 수행하는 블록은, 블록내 128개의 페이지 중 평균 120개 이상이 유효한 데이터로써 복사 비용이 매우 높은 연산이다. 따라서 마찬가지로 부가비용이 높은 wear leveling과 함께 삭제 클러스터링을 통해서 2번의 삭제를 1번의 삭제로 줄임으로써 wear leveling을 통하여 블록 삭제를 줄일 수 있다.

다음의 그림 12에서 확인 할 수 있듯이, 하나의 블록이 read reclaim 후 짧은 시간 내에 wear leveling의 희생 블록으로 선정 된다면, wear leveling의 발생 시기를 앞당김으로써 read reclaim과 wear leveling의 삭제 클러스터링 할 수 있다.

다시 말하자면, read reclaim이 발생하였을 때, 해당 블록이 wear leveling이 가능한 정도의 블록 삭제 횟수를 지닌 경우, 해당 블록을 read reclaim 하는 대신 wear leveler의 희생 블록으로 선정한 후, wear



[그림 12] 삭제 클러스터링

leveling을 수행함으로써 블록 내 데이터를 옮기고 블록을 삭제하는 read reclaim의 효과와 wear leveling의 효과를 모두 얻을 수 있다.

제 6 장 실험 결과

6. 1 실험 환경

읽기를 고려한 wear leveling 기법에 대한 성능 평가를 위해 본 논문에서는 read reclaim이 유효한 환경의 FTL(Flash Translation Layer) 시뮬레이터를 개발하다. 다른 기법과의 성능 비교를 위하여 dual-pool 알고리즘을 추가적으로 구현 하였으며, 시뮬레이터에 사용한 플래시 메모리의 사양을 다음의 표 2과 같다.

또한 사용한 트레이스는 읽기가 많은 일반 PC 사용 트레이스 mult1, multi2, mult3과 서버용 트레이스 t pcr이며, 실험은 읽기 연산의 누적을 위하여 각 트레이스를 반복적으로 수행하는 방식으로 수행하였다. 각 트레이스별 특징은 다음 표3과 같다.

또한 데이터의 특성을 파악하기 위해 수행한 데이터 분류는 멀티 해쉬

블록 개수	4096
블록 당 페이지 수	128
페이지 크기	2K
읽기 속도	100us
쓰기 속도	1600us
삭제 속도	5000us

[표 3] 플래시 메모리 사양

trace	multi1	multi2	multi3	t pcr
read	895107	1025099	1464724	277786
write	16132	23001	6076	4619

[표 4] 트레이스 별 읽기/쓰기 량

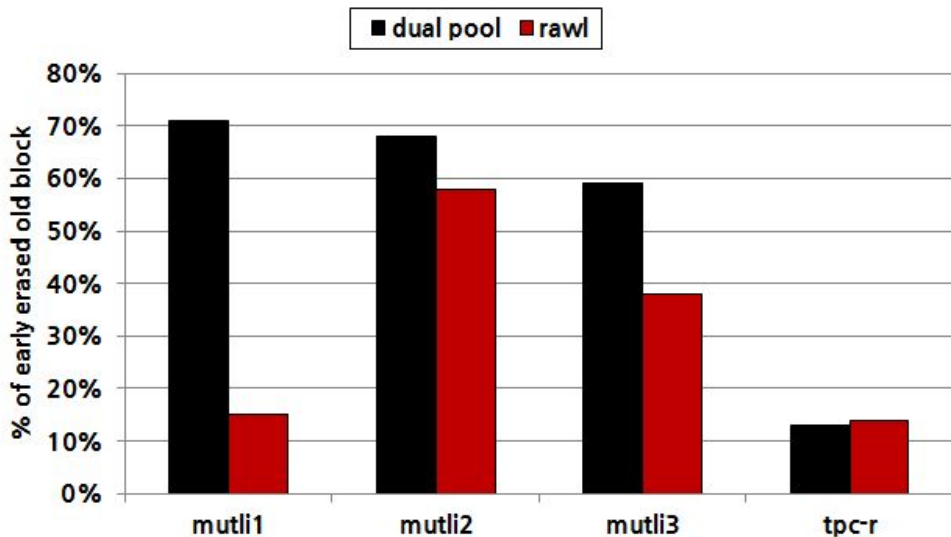
함수에 기반을 둔 bloomfilter를 사용하였으며, 분류 기준은 해당 논문을 따랐다[4].

6. 2 실험 결과

6. 2. 1 페이지 단위의 wear leveling

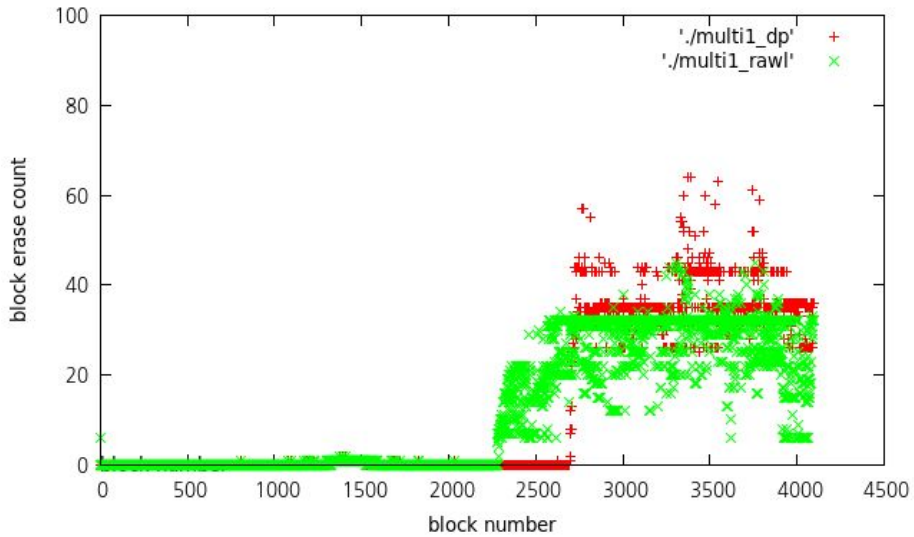
앞선 5절에서 설명한 데이터 특성 분석을 통한 페이지 단위의 wear leveling 기법을 수행한 결과에 대해서 설명하고자 한다.

본 논문에서 제안한 page 단위의 wear leveling 기법을 적용하였을 때, 기존의 wear leveling 기법이 가지고 있던 한계점을 극복했는지에 대해 분석해보았다. 아래의 그림 13은 read reclaim을 고려하지 못했던 wear leveling 기법인 dual pool 알고리즘과 read reclaim-aware한 wear leveling 기법에서 wear leveling 후, oldest 블록으로 옮겨간 cold한 특성을 가진 데이터가 삭제될 때까지의 시간이 youngest 블록으로 옮겨간 블록보다 짧은 경우를 나타낸다. dual pool 알고리즘을 적용하였을 때는 평

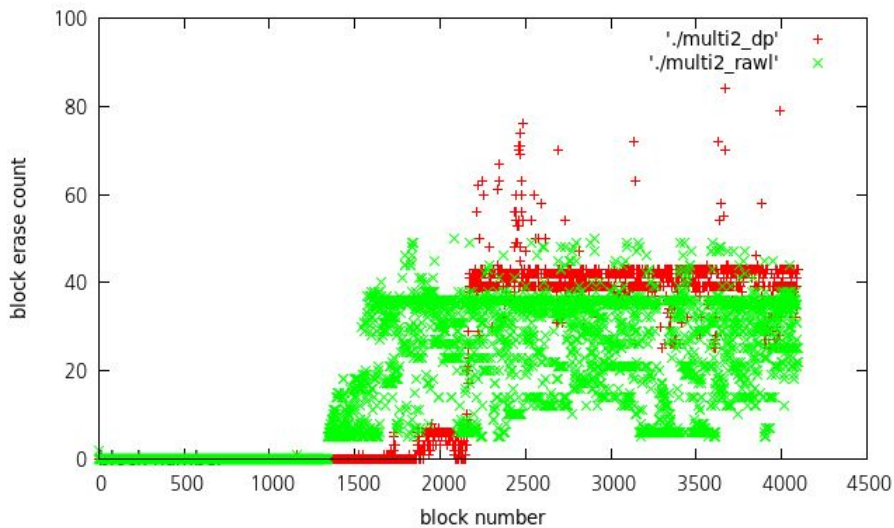


[그림 13] wear leveling 후, 삭제 될 때까지의 시간이 old 블록이 더 짧은 경우

균 55%정도의 경우에서 oldest 블록으로 옮겨간 데이터가 상대적으로 빨리 지워졌으나, read reclaim-aware한 wear leveling 기법에서는 평균 22%의 경우에서만 oldest 블록으로 옮겨간 데이터가 상대적으로 빨리 지워진 것을 확인하였다. 이러한 결과는 wear leveling의 부가연산을 크게



[그림 14] multi1에서의 블록별 삭제 횟수

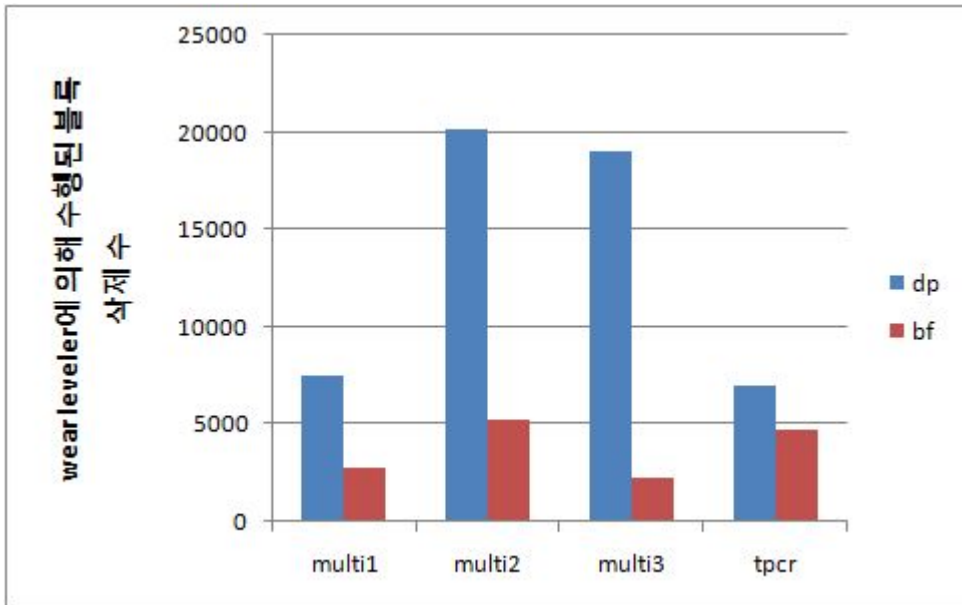


[그림 15] multi2에서의 블록별 삭제 횟수

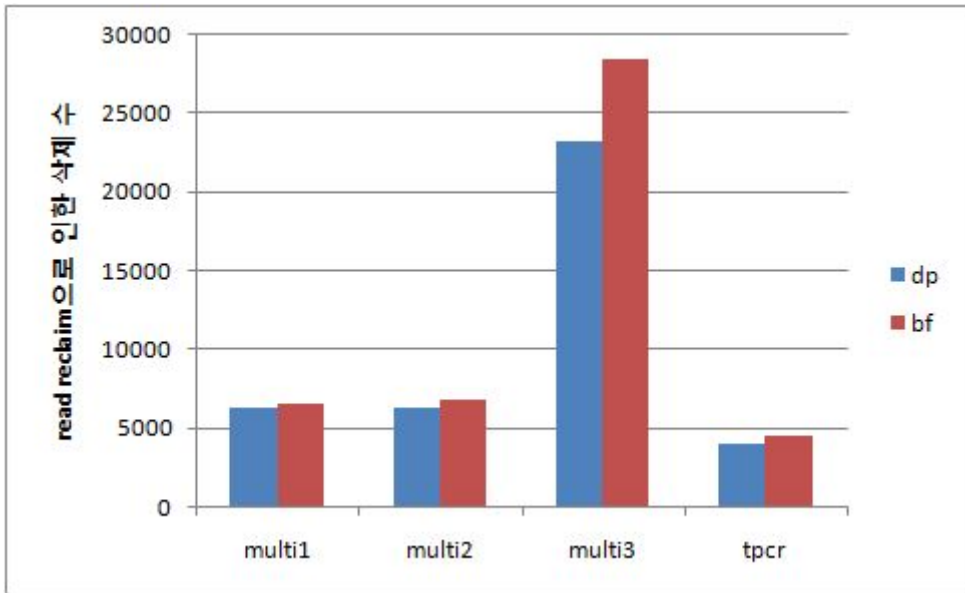
줄임으로써 전체 PE cycle의 감소를 유도할 수 있는데 이는 다음에 나타난 블록별 삭제 횟수에서 확인 할 수 있다.

앞의 그래프는 가장 효과가 뛰어났던 multi1과 multi2에서의 블록 별 삭제횟수를 나타낸다. X자로 나타난 녹색은 논문에서 제안한 read reclaim-aware wear leveling 기법을 적용하였을 때이며, 붉은색으로 나타난 +자는 dual-pool 알고리즘을 적용하였을 때의 블록별 삭제 횟수를 나타낸다. 이러한 결과를 통해 알 수 있듯 논문에서 제안한 기법은 읽기와 write의 비중이 적절히 섞여있을 때, 효과를 뛰어나게 발휘함을 알 수 있다.

또한 본 논문의 최종 목표였던 wear leveling을 통한 부가연산 수의 변화는 아래와 같다. 각 트레이스별 좌측 막대는 read reclaim이 있는 환경에 dual pool 알고리즘을 적용한 것이고, 가운데는 데이터 특성 분석을 통한 페이지 단위의 wear leveling 기법을 수행한 것이다. 또한, 가장 우측은 data separation을 oracle하게 수행하였을 때를 나타내 해당 기법의



[그림 16] 트레이스 별 wear leveling으로 인한 블록 삭제 수

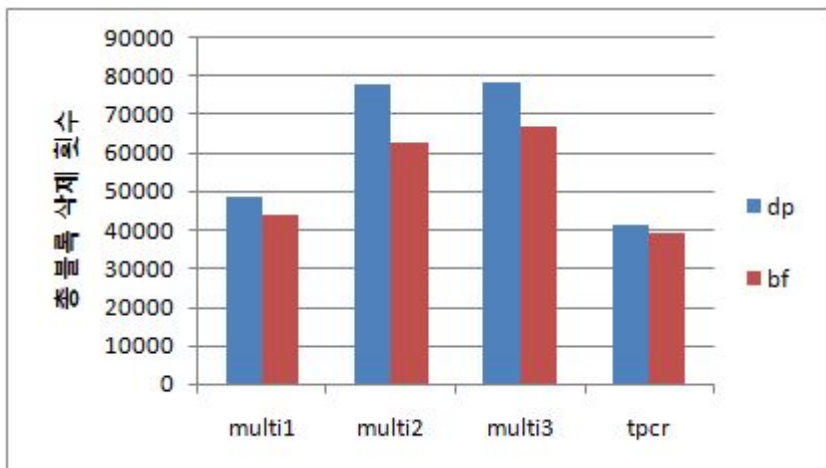


[그림 17] 트레이스별 read reclaim으로 인한 삭제 수

lower bound를 알 수 있다.

wear leveling으로 인한 삭제 부가 연산의 수는 아래에서 확인할 수 있듯이 read reclaim-aware 한 기법에서 72%감소하였다.

한편, wear leveling으로 인한 삭제 연산의 수가 줄어들어 따라 read



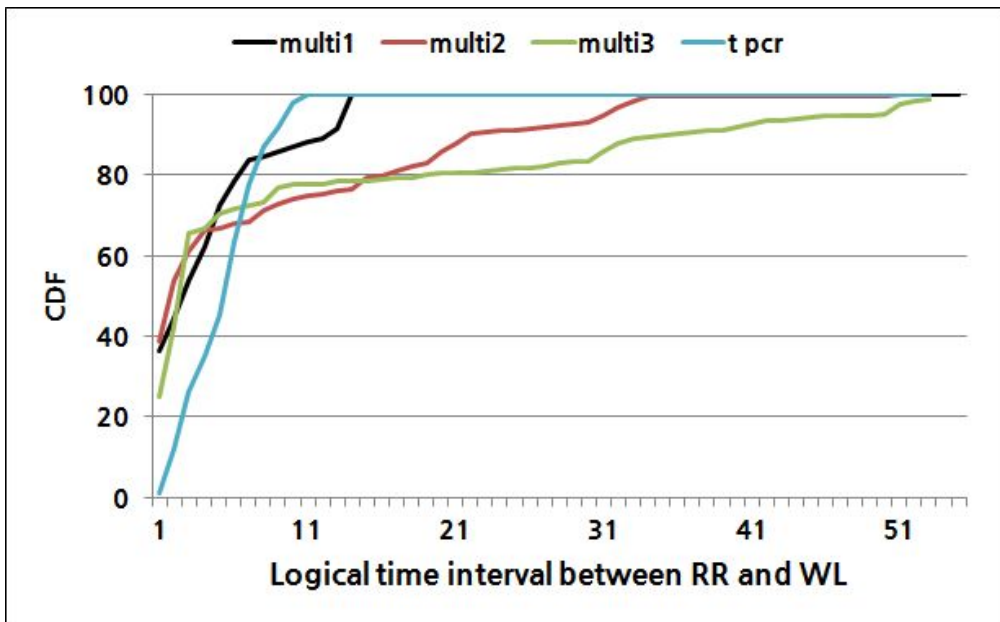
[그림 18] 트레이스별 총 블록 삭제 횟수

reclaim으로 인한 삭제 연산의 수가 증가하는 현상을 나타내었다. 이는 아래의 그래프 18에서 확인 할 수 있다. 본 논문에서 제안한 기법을 사용하였을 때는 16%정도 증가하였다.

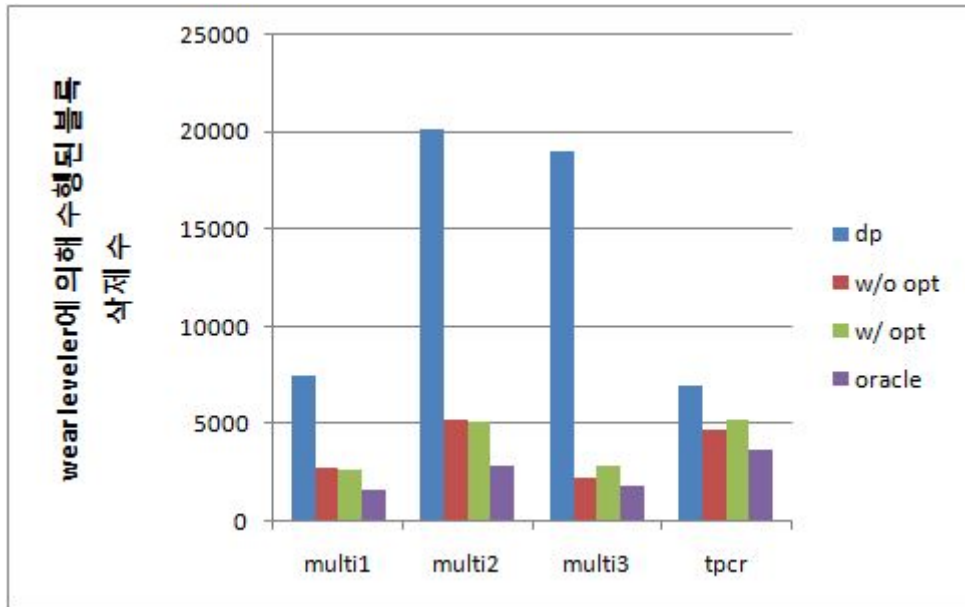
그러나, wear leveling으로 인한 삭제 연산 수의 감소의 정도가 크기 때문에 전체 PE는 13%만큼 감소하여 전체 PE cycle을 감소시키는 데 read reclaim-aware wear leveling 기법이 기여를 한 것을 확인 할 수 있다.

6. 2. 2 wear leveling 최적화

마지막으로, wear leveling의 최적화기법이다. read reclaim 후 짧은 시간 내에 연이어서 같은 블록이 wear leveling의 희생 블록으로 선정되는 경우, 이러한 삭제 연산을 클러스터링 함으로써 전체 삭제 연산의 수를



[그림 19] read reclaim 후 WL 희생 블록 선택까지의 거리

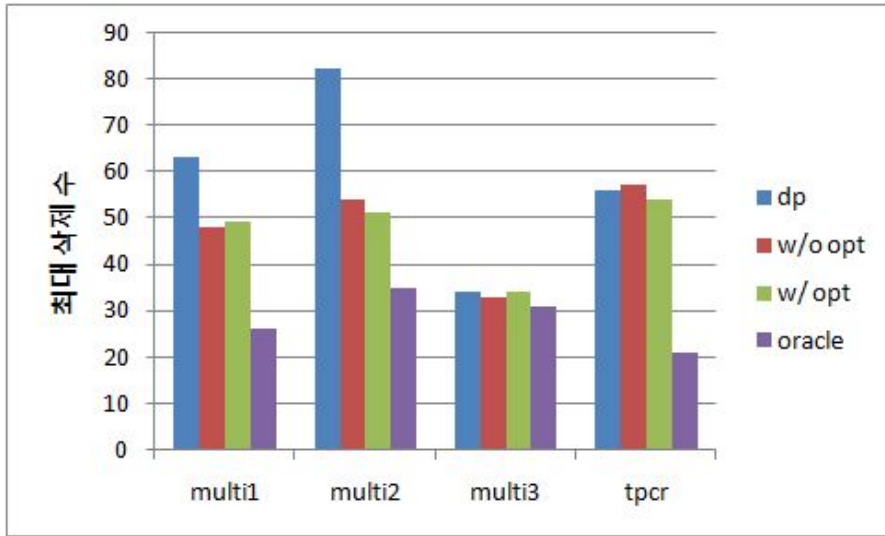


[그림 20] 트레이스 별 wear leveling으로 인한 블록 삭제 수

줄이는 효과를 얻고자 하는 기법이다.

이러한 기법을 적용하였을 때, read reclaim 후 해당 블록이 wear leveling의 희생블록으로 선택 될 때 거리의 변화를 나타낸 그래프가 위의 그림 19이다. 기법을 적용하기 전에는 전체 27%의 블록이 read reclaim 후, 5번째 이내에 발생하는 wear leveler에 의하여 희생 블록으로 채택되는 것을 확인할 수 있었다. 그러나, 최적화 기법을 적용한 후에는 전체 20% 미만의 블록이 read reclaim 후, 5번째 이내에 발생하는 wear leveler에 의하여 희생 블록으로 채택 되었다. 다시 말하여, 최적화 기법의 적용을 통하여 약 7%정도의 블록이 삭제 연산의 중복성을 회피할 수 있었다.

이러한 결과는 다음 그림 20과 같은 결과로 나타났다. 각 트레이스 별로 좌측의 막대는 read reclaim이 있는 환경에 dual-pool 알고리즘을 수행한 것이고 두 번째 막대는 읽기를 고려한 wear leveling 기법 중 최적화 기법을 수행하지 않은 결과이며, 3번째 막대가 최적화 기법을 수행한 결



[그림 21] 트레이스별 블록 최대 삭제 수

과이다. 또한 마지막은 데이터의 특성을 모두 알고 있다 가정했을 때의 oracle한 기법으로써 해당 기법의 lower bound라고 할 수 있다. 최적화 기법을 통하면, wear leveling을 공격적으로 수행하기 때문에 wear leveling으로 인한 삭제 횟수가 더 증가한 것을 확인 할 수 있다. 그러나, 중복된 삭제 횟수를 줄이고자하는 기법이기 때문에, 최대 삭제 횟수는 다소 감소한 경향을 보였다. 대부분의 트레이스에서 큰 변화를 보이진 않았지만, wear leveling의 효과가 상대적으로 적었던 서버향 트레이스인 tpcr에서 최대 삭제 수가 다소 감소하는 경향을 보였다. 또한, wear leveling으로 인한 삭제 수는 dual pool 알고리즘에 비하여 69% 감소하였으며 전체 트레이스 평균 13%정도의 삭제 횟수가 감소한 것을 알 수 있다. 이를 통하여, 전체적인 wear leveling의 성능을 다소 개선하면서 효과적으로 wear leveling으로 인한 부가 삭제 비용을 줄일 수 있음을 확인 할 수 있다.

마지막으로, 이러한 3가지 기법을 모두 적용하였을 때, 부가적인 garbage collection으로 인한 부가 복사 비용이 증가하지 않은 것을 확인

하였을 때, write hot을 유지함으로써 데이터의 연속성을 해치지 않는다는 것을 알 수 있었다.

제 7 장 결 론

7. 1 결 론

본 논문에서는 처음으로 읽기를 고려한 wear leveling을 제안하였다. 즉, read reclaim에 대한 문제를 FTL에서 이해함으로써, 지금까지의 wear leveling 기법들이 read reclaim이 있는 환경에서 부적응하였던 원인을 파악하였다. 또한 읽기 활용률을 늘림으로써 read reclaim을 줄이고자 하는 시도와, read reclaim과 wear leveling 삭제 연산을 클러스터링 함으로써 블록의 삭제 횟수를 최적화 시키는 기법을 제안하였다. 이를 통하여 wear leveling의 성능을 기존 수준과 유사하게 유지하면서, 전체 블록 삭제 횟수가 평균 13% 감소하였다.

7. 2 향후 연구

본 논문에서 읽기 활용률을 증가시키기 위하여 희생 블록 선택 시, 읽기 연산이 적은 블록보다는 많은 블록을 우선 선택하게 하였는 데, 이러한 방법으로는 읽기 활용률의 증가가 미미하게 나타났다. 따라서, 단순히 시간을 주는 방법보다는 보다 더 적극적인 방법을 통하여 읽기 활용률을 높임으로써 read reclaim을 감소시키는 방향으로 향후 연구를 진행할 계획이다.

참 고 문 헌

- [1] Laura M. Grupp et al, “The Bleak Future of NAND Flash Memory”, 10th USENIX Conference on File and Storage Technologies, 2012
- [2] JEDEC Standard, “Solid-State Drive(SSD) Requirements and Endurance Test Method”, 2010 9.
- [3] A. A. Chien and V. Karamcheti. “Moore’s Law: The First Ending and A New Beginning”, Technical report, Aug. 2012.
- [4] D. Park et al., “Hot Data Identification for Flash-based Storage Systems Using Multiple Bloom Filters”, MSST 2011.
- [5] Li-Pin Chang, “On Efficient Wear Leveling for Large-Scale Flash Memory Storage Systems, ” SAC, 2007.
- [6] Li Pin Chang et. al. “A Low Cost Wear Leveling Algorithm for Block Mapping Solid State Disks, ” LCTES, 2011.
- [7] Micron. “Design and Use Considerations for NAND Flash Memory”, Technical report, 2006.
- [8] S. Lee. Scaling Challenges in NAND Flash Device toward 10nm Technology. In Memory Workshop (IMW), 2012 4th IEEE International, pages 1 - 4. IEEE, 2012.
- [9] D. Post and H. Thio. READ DISTURB SCORECARD, Feb. 2 2012. US Patent 20,120,030,506.
- [10] M. Kang, K. Park, Y. Song, S. Hwang, B. Choi, Y. Song, Y. Lee, and C. Kim. Improving Read Disturb Characteristics by Self-Boosting Read Scheme for Multilevel NAND Flash Memories. Jpn J Appl Phys, 48(4):04C062 - 04C062, 2009.

Abstract

Flash memory has gradually broaden its user area not only in mobile devices but also in server system, based on the merits such as superior performance in reading and writing compared to hard disk, resistance to shock, and a low power feature.

Unlike the hard disk, flash memory has a limit on numbers of programming/erase (P/E) cycle. This resulted in successive studies on making less P/E cycle and wear leveling which makes the programming number for each block even.

However, along with the recent increase in density of flash memory, maximum numbers of erase has drastically decreased. Also, a new erase operation called the read reclaim came to light, as the read disturb problem worsened.

In this circumstance, the role of wear leveling became more important than ever, in order to assure the life time and capacity of storage.

Therefore, this thesis introduces the read aware wear leveling which is aware of the information on read reclaim, so as to overcome the limitations of wear leveling; Thereby; decreasing the erase overhead numbers caused by wear leveling, with sustaining the existing performance of wear leveling.

When an optimization technique was introduced, numbers of additional erase count by the wear leveling decreased by 69% on average, the entire erase count decreased by 13%.

keywords : flash memory, wear leveling, read disturb

Student Number : 2011-20833