



내부 캐시를 장착한 SSD에서 얇은 쓰기의 성능 평가

신 일 훈*

Performance Evaluation of Applying Shallow Write in SSDs with Internal Cache

Il-Hoon Shin*

이 연구는 서울과학기술대학교 교내연구비의 지원으로 수행되었습니다.

요 약

SSD의 성능은 SSD의 병렬 처리 능력을 극대화하거나 SSD 내부 메모리를 낸드에 대한 캐시로 활용함으로써 빠르게 향상되었다. 근래에는 상당량의 데이터가 수명이 짧은 데이터임을 감안하여 이들을 속도가 빠른 얇은 쓰기로 기록하여 성능을 향상하는 방법이 제안되었다. 하지만 기존 연구는 SSD 내부 캐시를 사용하지 않는 상태에서 얇은 쓰기의 성능 향상 효과를 평가한 문제가 있다. 본 연구는 내부 병렬화를 활용하는 SSD에서 SSD 내부 캐시와 얇은 쓰기를 함께 사용할 때의 성능 향상 효과 검증은 목표로 한다. 성능 평가 결과, 캐시를 사용하는 SSD에 얇은 쓰기를 적용하면 성능이 오히려 최대 98.0%까지 저하되었으며 블록 삭제 회수 또한 최대 79.8%까지 증가했다. 즉 얇은 쓰기는 SSD 성능 및 수명에 부정적인 영향을 미치는 것으로 나타났다.

Abstract

SSD performance has been rapidly improved by maximally utilizing the parallel processing capability of SSDs and by using SSD internal memory as cache for NAND flash. In addition, a method of improving performance by applying a fast shallow write for short-lived data has been proposed recently, based on the observation that a considerable amount of data are short-lived. However, the existing studies have applied the shallow write to the SSDs with no internal cache. Therefore, this work aims at verifying the performance improvement when using both the internal cache and shallow write. The performance evaluation results show that the performance is degraded by up to 98.0% when the shallow write is applied. The total number of block erasure has also increased by up to 79.8%. That is, the shallow write has a negative effect on the SSD performance and lifetime.

Keywords

shallow write, cache, data lifetime, SSD, NAND flash memory

* 서울과학기술대학교 전자IT미디어공학과
- ORCID: <https://orcid.org/0000-0003-2819-0398>

· Received: Sep. 21, 2018, Revised: Oct. 18, 2018, Accepted: Oct. 21, 2018
· Corresponding Author: Ilhoon Shin
Dept. of Electronics & IT Media Engineering
SeoulTech University, Korea.
Tel.: +82-2-970-6420, Email: ilhoon.shin@snut.ac.kr

1. 서 론

낸드 플래시 메모리를 저장 미디어로 사용하는 SSD는 하드 디스크 대비 월등히 높은 입출력 성능을 제공하고 에너지 소모 또한 적다. 또한 반도체 공정 기술의 발달에 따라 저장 용량 당 가격이 급격히 하락하고 있으며 저장 용량 또한 하드 디스크에 필적한 수준으로 증가하고 있다. 이에 따라 SSD는 기존의 노트북이나 PC뿐만 아니라 서버와 데이터 센터 등 기업용 스토리지 시장에서도 하드 디스크를 대체하고 있다.

그동안 SSD의 성능을 향상시키기 위한 연구가 다양하게 진행되어 왔는데, 이들을 크게 세 가지로 분류할 수 있다. 첫째는 SSD의 내부 병렬성을 활용한 연구이다[1]-[4]. SSD는 다수의 낸드 플래시 칩을 여러 채널을 활용하여 컨트롤러와 연결한다. 각 칩은 여러 다이로 구성되며 각 다이는 다시 여러 플레인으로 구성된다. 동일 다이에 속하는 여러 플레인들은 같은 종류의 연산을 동시에 수행할 수 있으며 각 다이는 독립적으로 낸드 연산을 동시에 수행할 수 있다. 또한 채널들은 동시에 데이터를 전송한다. 따라서 입출력 요청들은 페이지 단위의 서브 요청으로 나누어서 이들을 여러 다이 및 플레인으로 분산시켜 동시에 처리함으로써 입출력 성능을 크게 향상시킬 수 있다.

둘째로는 SSD 내부의 메모리를 낸드에 대한 캐시로 활용하는 것이다[5]-[8]. 낸드 플래시 메모리는 덮어쓰기 연산을 지원하지 않으므로 SSD와 같은 낸드 기반 저장 장치는 다른 자리 수정(Out-of-place update) 방식으로 쓰기 연산을 처리한다. 쓰기 요청시마다 아직 데이터가 쓰이지 않은 클린 페이지를 찾아서 데이터를 기록하는 이 방식을 사용하면 쓰기를 수행할 때마다 데이터의 위치가 변경되기 때문에 데이터의 논리 주소와 실제 주소를 매핑하는 테이블을 관리해야 한다. 데이터의 빠른 접근을 위해 일반적으로 매핑 테이블은 SSD 내부 메모리에 저장된다. 따라서 SSD는 내부에 대용량의 메모리를 장착하고 있으며 매핑 테이블 외에 남는 공간은 낸드 플래시 메모리에 대한 캐시 또는 쓰기 버퍼로 활용 가능하다.

셋째로는 얇은 쓰기(Shallow Write)를 활용하여 쓰기 성능을 향상시킬 수 있다. 일반적으로 낸드 플래시 메모리에서는 1년 이상의 데이터 유지를 보장하기 위해 속도가 느린 깊은 쓰기(Deep Write) 방식을 사용하여 쓰기 연산을 수행한다[9]. 하지만 서버 데이터 분석에 따르면 상당한 비율의 데이터는 그 수명이 짧다. 즉 업데이트 주기가 길지 않다. 따라서 이러한 데이터는 속도가 빠른 대신 데이터의 유지 시간이 짧은 얇은 쓰기를 활용하여 기록하면 SSD의 성능을 향상시킬 수 있다[9][10].

SSD 성능 향상을 목표로 한 기존 연구들을 보면 대부분 이들 세 가지 방식 중 한 가지 방식의 효과를 주로 검증했다. 하지만 근래 들어 이들을 함께 사용하여 성능 향상 정도를 분석하는 연구들이 수행되었다. 가령 연구 [8]은 SSD의 내부 병렬성을 활용하는 환경에서 SSD 내부 캐시를 위한 효율적인 관리 정책을 연구했으며, 연구 [9][10]은 역시 SSD의 내부 병렬성을 활용하는 환경에서 얇은 쓰기의 효과를 검증했다. 이들 연구들은 내부 병렬성과 SSD 내부 캐시를 함께 사용할 때 성능이 더욱 향상될 수 있음을 보여준다. 하지만 SSD 내부 캐시와 얇은 쓰기를 함께 적용하여 성능 향상 효과가 있는지를 검증하는 연구는 기존에 수행되지 않았다. 따라서 본 연구는 SSD 내부 병렬성 및 내부 메모리를 캐시로 활용하는 환경에서 얇은 쓰기를 적용하는 것이 성능 면에서 이점이 있는지를 분석하는 것을 목표로 한다.

본 논문은 다음과 같이 구성된다. 2장에서는 낸드 플래시 메모리 및 SSD 내부 구조, SSD의 성능을 향상시키기 위한 기존 연구들을 기술한다. 3장에서는 성능 평가를 위한 환경과 성능 평가 결과를 설명하고 4장에서 결론 및 향후 연구 이슈를 도출한다.

II. 배경 및 관련 연구

낸드 플래시 메모리는 EEPROM의 일종으로 블록과 페이지로 구성된다. 블록은 여러 페이지로 구성되며 삭제 연산의 기본 단위이다. 페이지는 보통 2-16KB의 크기이며 읽기 및 쓰기 연산의 기본 단위이다. 낸드 플래시 메모리는 덮어쓰기 연산을 지원

하지 않는다. 즉 특정 페이지에 한번 데이터가 기록 되면 해당 페이지에는 새로운 데이터를 쓸 수 없다. 새로운 데이터를 쓰기 위해서는 먼저 타겟 페이지가 속한 블록 전체를 삭제해야 한다. 따라서 낸드 기반 블록 장치들은 다른 자리 수정 방식으로 쓰기 연산을 처리한다. 즉 쓰기 요청이 들어오면 아직 데이터가 기록되지 않은 클린 페이지를 찾아 여기에 데이터를 기록한다. 따라서 데이터는 항상 기존의 위치와 상관없이 새로운 위치에 기록되며 플래시 사상 계층은 매핑 테이블을 이용하여 각 데이터의 현재 유효한 위치를 기억한다. 데이터의 빠른 접근을 위해 매핑 테이블은 통상 속도가 빠른 메모리에 유지된다. 플래시 사상 계층은 주소 매핑 단위에 따라 페이지 매핑[11][12], 블록 매핑[13], 하이브리드 매핑[14] 등으로 분류되는데 빠른 성능을 필요로 하는 저장 장치는 주로 페이지 매핑을 사용한다. SSD 또한 페이지 매핑을 사용하는 것으로 알려졌다[1]-[4].

반도체 공정 기술의 발전에 따라 단일 낸드 플래시 칩의 저장 용량이 점차 향상되고 있기는 하지만 여전히 하드 디스크를 대체하기에는 미흡하며, 낸드 플래시의 연산 속도는 지속적으로 저하되어 왔다. 그럼에도 불구하고 SSD는 다수의 낸드 플래시 칩을 병렬 구조로 연결함으로써 하드 디스크에 필적하는 저장 용량을 구현하는 동시에 하드 디스크를 월등하게 능가하는 입출력 속도를 제공한다. 다수의 칩은 동시에 데이터를 전송할 수 있는 여러 채널을 통해 컨트롤러와 연결된다. 또한 각 칩은 다수의 다리로 구성되며 이 다리는 독립적으로 낸드 연산을 수행할 수 있다. 즉 동시에 여러 낸드 연산을 수행하는 것이 가능하다. 또한 각 다리는 다수의 플레인으로 구성되며 동일 다이의 플레인은 동일한 종류의 낸드 연산을 동시에 수행할 수 있다.

따라서 각 입출력 요청들을 페이지 단위의 서브 입출력 요청으로 나누고 이들을 여러 다이 및 플레인에 분산하여 동시에 처리하면 입출력 성능을 크게 개선할 수 있다[1]-[4]. 이때 읽기 요청들은 타겟 다이 및 플레인이 정해져 있으므로 요청들이 특정 다이 및 플레인에 집중되면 SSD의 병렬성을 충분히 활용할 수 없다. 하지만 쓰기 요청은 타겟 다이 및 플레인이 정해져 있지 않다. 즉 임의의 다이와

플레인을 통해 쓰기 연산을 처리할 수 있다. 따라서 SSD의 병렬성을 극대화하는 동시에 특정 칩에 쓰기 연산이 집중되는 것을 방지하기 위한 다양한 타겟 다이 바인딩 정책들이 제안되었다[1]-[4]. 가령, 연구 [1]은 다이별 별도의 요청 큐를 두고 요청의 개수가 가장 적은 다이에 쓰기 요청을 보내는 정책을 제안했으며, 연구 [2]-[4]는 채널 및 다이의 현재 상태를 고려하여 타겟 다이를 결정한다. 연구 [4]는 추가적으로 멀티플레인 연산을 그 비용이 임계값보다 적을 때만 적용하도록 하였다. 최신 SSD들은 모두 이러한 내부 병렬성을 활용하고 있으므로 SSD 버퍼 관련 연구나 얇은 쓰기를 활용하는 연구들은 모두 SSD의 내부 병렬성을 지원하는 환경을 전제로 해야 한다.

한편, SSD의 내부 메모리 일부를 낸드 플래시 메모리의 캐시로 활용하여 성능을 향상시키는 연구가 활발히 진행되었다[5]-[8]. 이들은 주로 효율적인 캐시 교체 정책을 통해 캐시 적중률을 극대화하는 것을 목표로 했으며 낸드 플래시 메모리의 물리적 특성을 고려한다. FAB(Flash Aware Buffer)[5]와 BPLRU(Block Padding LRU)[6]는 블록 단위 버퍼 관리를 통해 가급적 순차 쓰기를 발생시키는데 초점을 맞추었으며 CFLRU(Clean First LRU)[7]는 클린 버퍼를 우선 교체함으로써 낸드 쓰기 연산을 최소화하는데 초점을 맞추었다. 하지만 이들은 주로 단일 낸드 칩을 가정한 환경에서 성능 평가를 수행했다는 한계가 있었다. 최근에는 SSD의 내부 병렬성을 구현한 시뮬레이션 환경에서 캐시 관리 정책을 평가하는 연구가 진행되었으며[8], 병렬성이 지원되는 환경에서도 캐시의 활용이 성능 향상에 크게 효과적이라는 사실을 보여주었다.

SSD의 성능을 향상시키기 위한 또 다른 연구는 얇은 쓰기를 활용하는 것이다. 낸드 플래시 메모리는 플로팅 게이트에 전자를 채우고 비우는 동작을 통해 데이터를 표현한다. 가령 하나의 플로팅 게이트가 2개의 비트를 표현하는 MLC(Multi Level Cell) 낸드라면 플로팅 게이트의 충전 상태는 총 네 가지가 존재한다. 또한 한 플로팅 게이트가 3 비트를 표현하는 TLC(Triple Level Cell) 낸드에서는 충전 상태를 8가지로 세분화해야 한다. 따라서 플로팅 게이트의 충전 전하량을 정밀하게 제어하기 위해 낸드

쓰기 연산은 ISPP(Incremental Step Pulse Programming) 방식으로 수행된다[9]. ISPP 방식은 타겟 전하량이 충전될 때까지 전압을 단계적으로 높여 가며 플로팅 게이트를 충전하는 방식이다. 이 때 단계 전압이 낮을수록 충전 전하량을 보다 정밀하게 제어할 수 있어서 데이터 유지 에러율이 낮아지는 장점이 있으나 쓰기 연산의 속도가 느리다는 단점이 있다. 반대로 높은 전압을 사용하면 속도는 빠르지만 에러 없이 데이터를 유지할 수 있는 시간이 짧아지는 단점이 있다. 현재의 낸드 플래시 메모리는 1년 이상의 데이터 유지를 위해 낮은 전압을 사용하여 충전 전하량을 제어한다. 따라서 집적도가 향상될수록 쓰기 연산의 속도는 점차 감소되어 왔다.

하지만 실제 서버 워크로드 분석에 의하면 상당량의 데이터가 극히 짧은 수명을 갖는다. 즉 한번 기록되고 나서 짧은 시간 안에 업데이트된다. 따라서 이들 수명이 짧은 데이터를 낮은 전압을 사용하는 깊은 쓰기 방식으로 기록하는 것은 비효율적이다. 수명이 짧은 데이터는 높은 전압을 사용하여 얇은 쓰기 방식으로 기록하는 것이 효과적이다. 쓰기 연산의 속도가 그만큼 빨라지기 때문이다. 한편 데이터의 수명을 잘못 판단하는 경우, 즉 얇은 쓰기로 처리했는데 일정 시간 이상 데이터가 업데이트 되지 않으면 해당 데이터는 깊은 쓰기로 다시 재기록해야 한다. 데이터의 재기록은 백그라운드로 수행되지만 과도한 재기록 연산은 SSD의 성능을 저하시킬 수 있으며 또한 SSD의 전체 수명을 단축할 수 있다. 따라서 데이터의 수명을 정확하게 예측하는 것이 중요하다. 연구 [9]는 호스트의 모든 입출력 요청을 얇은 쓰기로 처리하는 정책을 제안했고, 연구 [10]은 요청의 크기가 작은 데이터만 얇은 쓰기로 처리하고 나머지 데이터 깊은 쓰기로 처리하는 정책을 제안했다.

이들 연구들은 모두 SSD의 병렬성을 지원하는 시뮬레이터에서 성능 평가를 수행했고 얇은 쓰기가 SSD의 성능을 향상시키는 것으로 나타났지만, SSD 내부 캐시를 가정하지 않았다는 한계가 있다. 만약 SSD 내부 캐시가 존재한다면 수명이 짧은 데이터에 대한 업데이트 연산이 상당 부분 캐시에 의해 흡수되고 낸드에 대한 쓰기 연산은 더티 버퍼가 캐시 교체 정책에 의해 교체될 때에만 발생하므로 이

들 데이터들은 수명이 긴 데이터가 될 확률이 크다. 따라서 캐시가 존재하는 환경에서도 얇은 쓰기가 SSD 성능 향상을 가져올 수 있는지 검증이 필요하며 본 연구는 이를 목표로 한다.

III. 성능 평가

성능 평가를 위해 SSD의 내부 병렬성을 지원하는 SSDSim[2] 시뮬레이터를 사용했으며, 타겟 SSD는 다음과 같이 설정했다. SSD는 4개의 낸드 플래시 칩으로 구성되며 이들은 2개의 병렬 채널을 통해 낸드 컨트롤러와 연결된다. 각 칩은 4개의 다이로 구성되며, 각 다이는 4개의 플레인으로 구성된다. 각 플레인을 구성하는 낸드 블록 크기는 512KB이며, 페이지 크기는 4KB이다. 페이지 읽기 연산, 쓰기 연산, 블록 삭제 연산의 속도는 각각 500 μ s, 900 μ s, 3.5ms이다. 채널을 통해 1바이트의 데이터를 보내는 데는 25ns이 소요되며 메모리 접근 시간은 워드 당 15ns이다. SSD는 페이지 매핑 플래시 사상 정책을 사용하며 초과 저장 용량 비율은 20%로 설정했다. 각 플레인의 클린 블록의 비율이 10% 미만으로 떨어지면 클린 블록을 재생하기 위한 가비지 컬렉션이 백그라운드로 수행된다. 시뮬레이터의 입력 트레이스로는 대표적인 서버 트레이스인 마이크로소프트 워크로드들을 사용했다[15]. 사용한 서버 워크로드의 상세한 특성은 표 1과 같다.

표 1. 트레이스 정보
Table 1. Trace information

Trace	Read req. ratio (%)	Avg. read req. size(sectors)	Avg. write req. size(sectors)
hm0	32.73	14.74	16.67
mds0	30.65	47.42	14.48
proj0	5.85	35.68	81.83
prn0	22.21	45.67	23.31

그림 1(a) ~ (d)는 평균 반응 시간을 측정된 결과를 보여준다. X축은 캐시의 크기로서 16MB부터 128MB까지로 설정했다. Y축은 반응 시간으로 μ s 단위로 도시했다. 캐시를 사용하는 SSD에서 얇은 쓰기의 효과를 검증하기 위해 총 6가지의 정책을 시뮬레이터에 구현하여 성능을 비교했다.

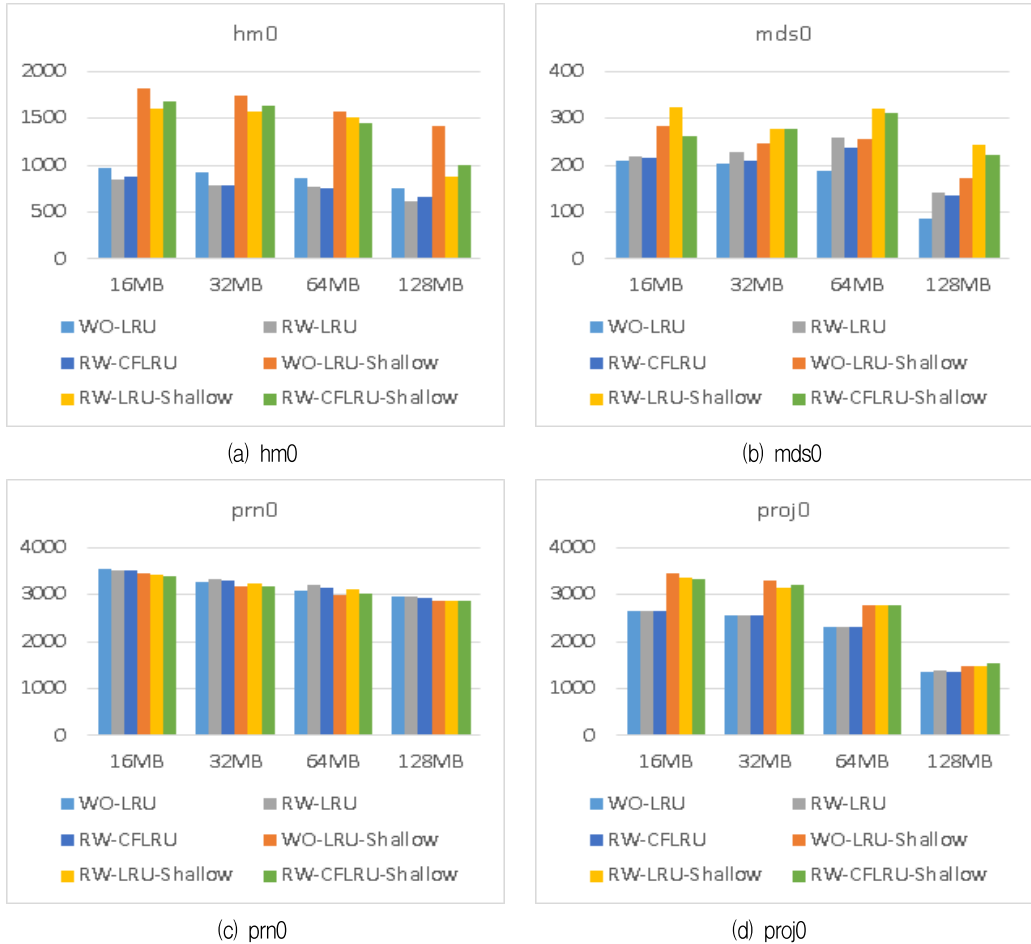


그림 1. 평균 반응 시간
Fig. 1. Average response time

먼저 WO-LRU는 LRU 교체 정책을 사용하며 쓰기 요청만을 캐시로 수용하는 정책을 의미한다. 즉 캐시를 쓰기 버퍼로 활용되는 정책이다. RW-LRU는 LRU 교체 정책을 사용하며 읽기와 쓰기 요청 모두를 캐시에 수용하는 정책이다. RW-CFLRU는 CFLRU 교체 정책을 사용하며 읽기와 쓰기 요청 모두를 캐시에 수용한다. 각 정책의 이름에 Shallow가 추가된 정책들은 각 캐시 관리 정책을 운영하는 환경에서 만약 더티 버퍼의 교체에 의해 낸드 플래시에 대한 쓰기 요청이 발생하면 해당 쓰기를 깊은 쓰기 대신 얇은 쓰기로 수행하는 정책을 의미한다. 여기서는 데이터의 수명에 대한 별도의 예측을 하지 않고 모든 더티 버퍼 교체를 얇은 쓰기로 처리했다. 즉 모든 호스트 요청을 얇은 쓰기로 처리한 연구 [9]와

유사한 방식이다.

결과를 보면 prn0를 제외한 모든 트레이스에서, 캐시만을 사용하고 얇은 쓰기를 활용하지 않을 때의 평균 반응 시간이 얇은 쓰기를 사용하는 것보다 훨씬 짧음을 알 수 있다. 얇은 쓰기를 캐시와 함께 활용하게 되면 16MB 캐시에서는 반응 시간이 21.6 ~ 91.2% 길어졌으며 128MB 캐시에서는 8.0 ~ 98.0% 길어졌다. 즉 캐시를 사용하지 않을 때는 상당 비율의 데이터가 짧은 수명을 갖는 데이터여서 얇은 쓰기를 적용하는 것이 효과적이지만 캐시를 사용하게 되면 짧은 수명을 갖는 데이터에 대한 쓰기 요청이 캐시에 의해 흡수되므로 캐시에서 교체되는 데이터는 대부분 수명이 긴 데이터여서 얇은 쓰기를 적용하는 것이 효과적이지 않음을 알 수 있다.

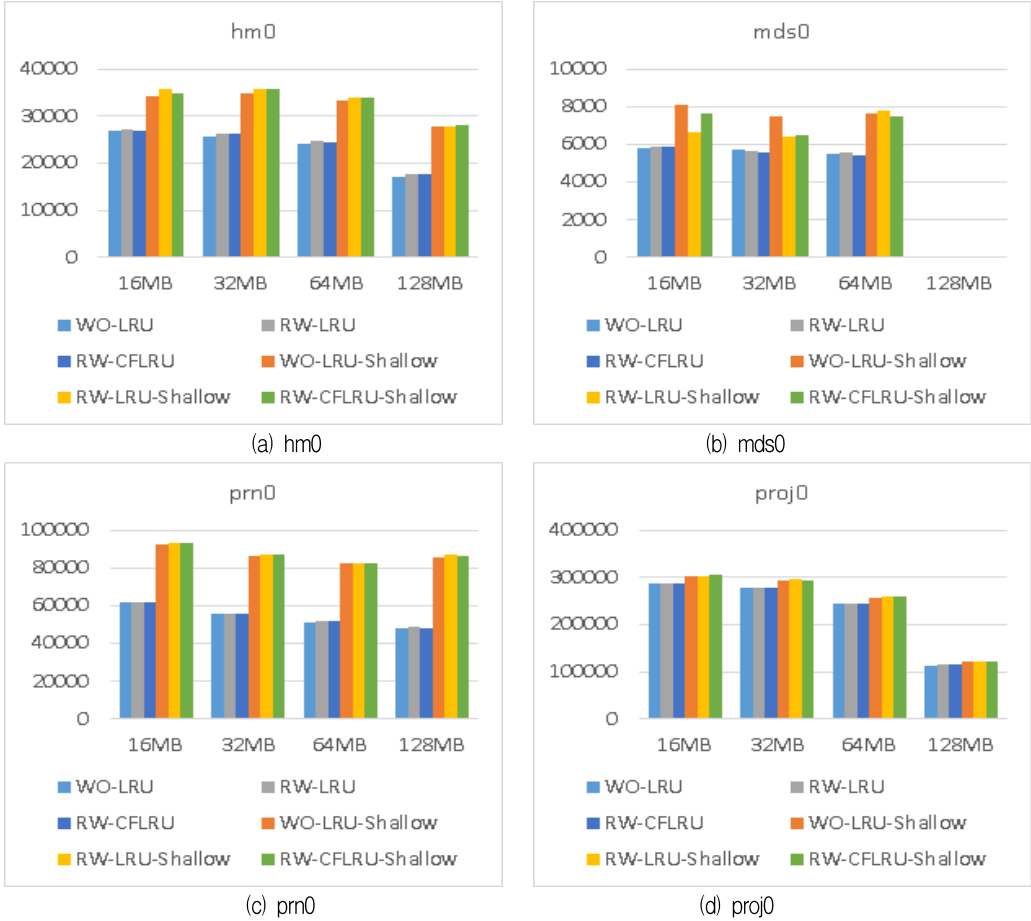


그림 2. 블록의 총 삭제 회수
Fig. 2. Total block erasure count

그림 2(a) ~ (d)는 총 블록 삭제 회수를 보여준다. X축은 캐시의 크기로서 16MB부터 128MB까지로 설정했다. Y축은 블록 삭제 회수이다.

결과를 보면 얇은 쓰기를 사용하면 모든 트레이스에서 블록 삭제 회수가 크게 증가함을 알 수 있다. 16MB 캐시에서는 블록 삭제 회수가 13.3 ~ 62.7% 증가했으며 128MB 캐시에서는 0.0 ~ 79.8% 증가했다. 즉 캐시에서 교체되는 수명이 긴 데이터를 모두 얇은 쓰기로 기록했기 때문에, 데이터 수명을 잘못 판단한 데이터들을 깊은 쓰기로 다시 재기록하는 오버헤드가 매우 컸음을 알 수 있다. 한편 mds0 트레이스에서 캐시 크기가 128MB일 때는 가비지 컬렉션이 전혀 발생하지 않아 블록 삭제 회수가 모두 0이었다.

결과적으로 SSD 내부 메모리의 일부를 캐시로 사용하는 환경에서는 얇은 쓰기를 활용하여 성능을 향상시키는 정책은 성능 면에서 또한 SSD의 수명 면에서 부정적인 영향을 미침을 알 수 있다.

IV. 결론 및 향후 과제

본 연구는 SSD 내부 메모리의 일부를 낸드 플래시 메모리에 대한 캐시로 활용하는 환경에서 얇은 쓰기의 성능 향상 효과를 검증했다. 성능 평가 결과 캐시를 사용하는 SSD에 얇은 쓰기를 적용하면 성능이 오히려 최대 98.0%까지 저하되었으며 블록 삭제 회수 또한 최대 79.8%까지 증가했다. 즉 얇은 쓰기는 SSD 성능 및 수명에 부정적인 영향을 미쳤

다. 이는 수명에 짧은 데이터에 대한 쓰기 연산의 상당수가 캐시에 의해 흡수되며 캐시에서 교체되는 데이터 버퍼의 대부분은 수명이 긴 데이터임을 시사한다. 결과적으로 캐시를 사용하는 SSD 환경에서 얇은 쓰기를 효과적으로 활용하기 위해서는 교체되는 데이터 버퍼의 수명을 보다 더 정확하게 예측하여 얇은 쓰기를 선별적으로 활용해야 한다. 이는 향후 연구 과제이다.

References

- [1] C. Park, E. Seo, J. Shin, S. Maeng, and J. Lee "Exploiting internal parallelism of flash-based SSDs", IEEE Computer Architecture Letters, Vol. 9, No. 1, pp. 9-12, Mar. 2010.
- [2] Y. Hu, H. Jiang, D. Feng, L. Tian, H. Luo, and S. Zhang, "Performance impact and interplay of SSD parallelism through advanced commands, allocation strategy and data granularity", Proc. 25th ACM International Conf. on Supercomputing, Tucson, Arizona, pp. 96-107, May 2011.
- [3] Y. A. Winata, K. Sanghoon, and I. Shin "Enhancing internal parallelism of solid-state drives while balancing write loads across dies", Electronics Letters, Vol. 51, No. 24, pp. 1978-1980, Nov. 2015.
- [4] I. Shin, "Improving internal parallelism of solid state drives with selective multi-plane operation", Electronics Letters, Vol. 54, No. 2, pp. 64-66, Jan. 2018.
- [5] H. Jo, J. Kang, S. Park, J. Kim, and J. Lee, "FAB: Flash-aware buffer management policy for portable media players", IEEE Trans. Consumer Electron., Vol. 52, No. 2, pp. 485-493, May 2006.
- [6] H. Kim and S. Ahn, "BPLRU: A buffer management scheme for improving random writes in flash storage", USENIX FAST Conference, Article. 16, pp. 239-252, Feb. 2008.
- [7] S. Park, D. Jung, J. Kang, J. Kim, and J. Lee, "CFLRU: A replacement algorithm for flash memory", International Conference of Compilers, Architecture, and Synthesis for Embedded Systems, pp. 234-241, Oct. 2006.
- [8] I. Shin and J. Kim, "Performance analysis of buffer management policy considering internal parallelism of solid state drives", IEICE Electronics Express, Vol. 15, No. 15, pp. 1-8, Aug. 2018.
- [9] R. Liu, C. Yang, and W. Wu "Optimizing NAND flash-based SSDs via retention relaxation", USENIX FAST Conference, Article. 14, Feb. 2012.
- [10] I. Shin, "Applying fast shallow write to short-lived data in solid-state drives", IEICE Electronics Express, Vol. 15, No. 13, pp. 1-9, Jul. 2018.
- [11] A. Ban, "Flash file system", United States Patent. No. 5,404,485, Apr. 1995.
- [12] D. Yoo and I. Shin, "Implementing Greedy Replacement Scheme using Multiple List for Page Mapping Scheme", Journal of KIIT, Vol. 9, No. 6, pp. 17-23, Jun. 2011.
- [13] A. Ban, "Flash file system optimized for page-mode flash technologies", United States Patent. No. 5,937,425, 1999.
- [14] J. Kim, J. M. Kim, S. Noh, S. L. Min, and Y. Cho, "A space-efficient flash translation layer for compactflash systems", IEEE Trans. Consumer Electron., Vol. 48, No. 2, pp. 366-375, May 2002.
- [15] D. Narayanan, A. Donnelly, and A. Rowstron, "Write off-loading: practical power management for enterprise storage", USENIX FAST Conference, Article. 10, pp. 253-267, Feb. 2008.

저자소개

신 일 훈 (Il-Hoon Shin)



1998년 2월 : 서울대학교

계산통계학과(이학사)

2000년 2월 : 서울대학교

전산과학과(이학석사)

2005년 2월 : 서울대학교

컴퓨터공학부(공학박사)

2008년 3월 ~ 현재 :

서울과학기술대학교 교수

관심분야 : 스토리지 시스템, 운영체제, 임베디드 시스템