

SSD 쓰기 감소를 위한 2WPR 캐시 페이지 교체 알고리즘 평가 및 개선

이 형 봉¹ · 권 기 현^{2*}

¹강릉원주대학교 컴퓨터공학과 교수

^{2*}강원대학교 전자정보통신공학부 교수

A Performance Evaluation and Improvement on the 2WPR Cache Page Replacement Algorithm for Reducing Writes in SSDs

Hyung-Bong Lee¹ · Ki-Hyeon Kwon^{2*}

¹Professor, Department of Computer Science & Engineering, Gangneung-Wonju National University, Wonju 25457, Korea

^{2*}Professor, Department of Electronics, Information & Communication Engineering, Kangwon National University, Samcheok 25913, Korea

[요 약]

플래시 메모리 기반 SSD는 기존의 HDD에 비해 접근 속도가 빠르고 가벼우며 물리적 크기가 작다는 장점이 있다. 반면에, 쓰기 전 지우기 연산에 따른 쓰기 팽창 및 지우기 연산 횟수 제한이라는 태생적 단점을 지니고 있다. 플래시 메모리 기반 SSD의 효과적 활용 연구들은 캐시 정책에 의한 쓰기 감소 방안부터 출발한다. 이런 연구 중의 하나로, 최근 2WPR 캐시 페이지 교체 알고리즘이 발표되었다. 이 연구에서는 FTL 환경에서 2WPR 알고리즘을 분석·평가하고 개선점을 제안하였다. 대규모 현장 트레이스 데이터 기반의 실험 결과, 쓰기 감소 관점에서 개선된 알고리즘의 희생 페이지 선택 기준 지표들이 기존 알고리즘의 지표들보다 약 2% 정도 더 우수하게 반영하는 것으로 나타났다.

[Abstract]

Flash memory-based SSDs have the advantage of faster access speed, lighter weight and smaller physical size than conventional HDDs. On the other hand, it has inherent disadvantages of amplifying writes and limiting the number of erase operation needed before writing. Therefore, researches for effective use of flash memory-based SSDs start with write reduction scheme based on cache policy. As one of such researches, the 2WPR cache page replacement algorithm was published recently. In a FTL environment, this paper analyzes and evaluates the 2WPR algorithm, and suggests improvements. As a result of the test based on large scale field trace data, it shows that the metrics for victim page selection of the improved algorithm reflect about 2% better than those of the original algorithm in terms of write reduction.

색인어 : 2WPR, 캐시 페이지 교체, 플래시 메모리, 플래시 변환 계층, LRU

Key word : 2WPR(write weight probability of reference), Cache Page Replacement, Flash Memory, FTL(flash translation layer), LRU(least recently used)

<http://dx.doi.org/10.9728/dcs.2021.22.2.317>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 26 November 2020; **Revised** 27 January 2021

Accepted 27 January 2021

***Corresponding Author; Ki-Hyeon Kwon**

Tel: +82-33-570-6400

E-mail: kweon@kangwon.ac.kr

I. 서론

비휘발성인 플래시 메모리나 휘발성인 DRAM 등 반도체 소자를 활용하여 대용량 주변 저장장치로 사용하는 드라이브를 SSD(solid state drive)라 하는데[1]. 최근에는 플래시 메모리 기반 SSD를 일반적인 SSD로 지칭한다[2]. 이 연구에서도 플래시 메모리 기반 SSD를 SSD로 약칭하기로 한다. 이러한 SSD는 최근 기존의 HDD를 대체하며 급속도로 일반화되고 있다. 플래시 메모리 제조 기술이 크게 진보함에 따라 용량이 증가하고 제조 원가가 낮아짐으로써 온라인 콘텐츠나 금융 등 대규모 DB 저장 매체로의 도입 사례가 늘어나고 있다[3], [4]. 플래시 메모리는 제자리 덮어쓰기가 불가능하고, 읽기와 쓰기 속도가 다르며, 쓰기 전 지우기 연산 횟수에 제한이 있다는 단점이 있다. 이러한 SSD 단점들의 극복은 쓰기 팽창 억제에서 출발한다. 쓰기 팽창이란 한 번의 쓰기가 거기서 멈추지 않고 2차, 3차 쓰기를 유발한다는 의미인데, 한 번 쓰인 페이지는 지우기가 필요하고, 지우기는 블록 단위로 가능하기 때문에 특정 가비지를 수집하기 위해서는 해당 블록에 포함된 유효 페이지들을 복사해야 한다[5], [6]. 그러므로, 단 1회라 하더라도 쓰기 감소의 파급 효과는 적지 않다.

이 연구에서는 최근 발표된 SSD의 쓰기 감소를 위한 캐시 페이지 교체 알고리즘 2WPR[7]을 심층 분석하고 새로운 캐시 교체비용 측정지표 도입에 의한 개선방안을 제안한다. 이를 위하여 2 장에서 SSD 캐시 관련 연구를 살펴보고, 3 장에서 2WPR 알고리즘을 분석 및 평가하며, 4 장에서는 개선 점을 제안하고 실험한다. 그리고 마지막 5 장에서 결론으로 이 논문을 맺는다.

II. SSD 연구 배경 및 관련 연구

2-1 SSD 연구 배경

SSD의 쓰기 감소를 위한 기본적인 전략은 캐시 운영 전략에서 쓰기를 최대한 억제하는 방안을 모색하는 것이다[8]. 일반적인 캐시 전략의 최대 목표는 읽기와 쓰기를 포함하여 전체적인 적중률을 높이는 데 있지만, SSD의 경우에는 적중률 하락에 대한 보상으로써 쓰기 횟수를 줄이는 데 있다. SSD의 쓰기 감소를 위해서 쓰기 적중률은 유지하거나 높여 읽기 적중률의 하락을 감수하지는 의도이다. 좀 더 구체적으로 접근하면 쓰기 1회를 줄이는 대신 읽기 몇 회를 더 허용한다는 의미로 풀이된다. 연구[9]에서 사용된 표 1의 SSD 사양 예에 따르면 접근 시간 관점에서는 쓰기 1회가 읽기 10회와 동등하고, 에너지 소모관점에에서는 쓰기 1회가 읽기 15회와 맞먹는다. 여기에 쓰기 팽창 결과를 고려한다면 쓰기 대비 읽기의 대응 횟수는 더 많아질 것인데, 이에 대한 보다 정확한 산출을 시도하는 연구들이 있다[6], [10].

표 1. SSD 접근 시간과 에너지 소모의 예

Table 1. An Example of Access Time and Energy Consumption of SSDs

item	read	write	erase
time(us)	25	250	1500
energy(uJ)	0.5	7.5	40

2-2 SSD 쓰기 감소 관련 연구

SSD의 쓰기 감소를 위한 캐시 알고리즘 연구는 SSD가 본격적으로 보급되기 시작한 2000년대 중반에 시작되었다. 이 시기의 대표적인 연구로 CF-LRU(clear first LRU)[11]를 들 수 있다. CF-LRU는 희생 페이지가 필요할 때 그림 1과 같이 LRU 리스트의 상위 몇 개(윈도우 w)를 대상으로 클린(clean)과 더티(dirty) 그룹으로 분류한 뒤 클린 그룹이 없을 때 더티 그룹에서 희생 페이지를 선택한다. 더티 페이지를 방출할 때 발생하는 쓰기 연산을 최대한 미루자는 의미인데, 사용되지 않을 더티 페이지가 캐시를 오래 차지하여 캐시 전체의 용량을 감소시킴으로써 결국 다른 쓰기 연산을 유발할 가능성이 높다. 이러한 CF-LRU의 단점은 더티 그룹을 좀 더 세분화함으로써 완화시킬 수 있다. LRU-WSR(LRU and write sequence reordering) [12]는 첫 참조 페이지를 콜드(cold), 두 번 이상의 참조 페이지를 핫(hot)으로 정의하고, 클린과 콜드-더티 페이지를 동일 순위로 방출한다. 다만, 핫-더티 페이지에 대한 영원한 우대를 예방하기 위해 윈도우 내 핫-더티 페이지는 콜드-더티 상태로 변환하여 MRU로 이동시킨다. 이렇게 이동된 콜드-더티 페이지가 참조 없이 LRU 윈도우 범위에 진입하면 방출 대상이 된다.

AD-LRU(adaptive double LRU)[13]는 그림 2와 같이 리스트를 아예 콜드 페이지와 핫 페이지 등 두 개로 분리하고, 첫 참조시 콜드 페이지에 삽입한 다음, 재 참조 이후에는 핫 리스트로 이동시킨다. 희생 페이지의 선택은 콜드 리스트가 일정 크기 이상이면 콜드 리스트에서, 그렇지 않으면 핫 리스트에서 LRU-WSR 방식으로 이루어지는데, 다만 콜드-클린과 핫-클린 등 클린 그룹의 세분화가 더해진다.

그림 2의 LRU 리스트 분리 개념은 클린/더티, 핫/콜드 등 단순한 두 가지 이분법적 조합 외에 더 유연하고 다양한 지표의 적용을 가능케 하는 틀로 자리 잡고 있다. 이러한 LRU 리스트 분리 개념이 쓰기 감소 알고리즘을 한 단계 업그레이드할 수 있는 계기가 되었고, 그 연장선 상에서 가장 최근에 발표된 연구 결과 중의 하나가 2WPR 알고리즘이다.

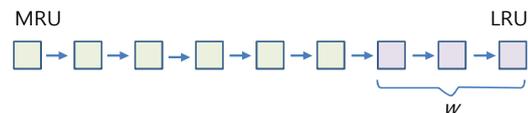


그림 1. CF-LRU와 LRU-WSR의 LRU 리스트
Fig. 1. LRU List of CF-LRU and LRU-WSR

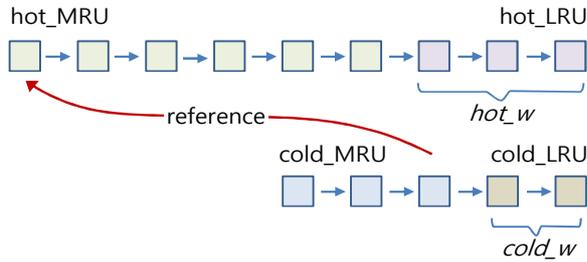


그림 2. AD-LRU의 LRU 리스트
Fig. 2. LRU List of AD-LRU

III. 2WPR 알고리즘의 분석 및 평가

3-1 2WPR 알고리즘의 LRU 리스트

2WPR 알고리즘은 그림 3과 같이 LRU 리스트를 2WPR_List와 Victim_List로 분리하여 아래와 같이 운영한다.

- ① 첫 참조 페이지는 2WPR_List의 MRU에 삽입한다.
- ② 2WPR_List의 크기가 주어진 한계를 초과하면 윈도우(w) 내에서 희생 페이지를 선택하여 Victim_List의 MRU로 이동삽입한다.
- ③ 과정 ②에서 Victim_List의 크기가 주어진 한계를 초과하면 CF-LRU 방식으로 희생 페이지를 선택하여 방출한다.
- ④ 2WPR_List에서 재 참조된 페이지는 해당 리스트의 MRU로 이동시킨다.
- ⑤ Victim_List에서 재 참조된 페이지는 2WPR_List의 MRU로 이동삽입한다.

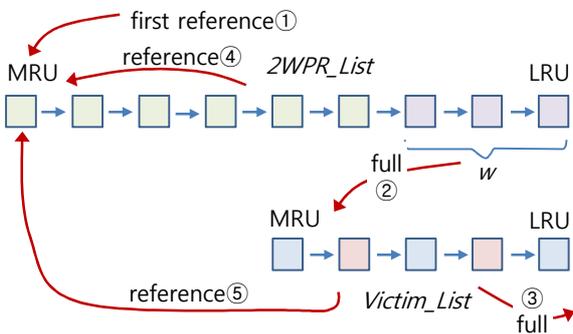


그림 3. 2WPR의 LRU 리스트
Fig. 3. LRU List of 2WPR

3-2 2WPR_list의 희생 페이지 선택

2WPR 알고리즘의 핵심은 그림 3의 과정 ②에서 쓰기를 최대한 줄일 수 있는 방출 후보 페이지 선정 방법에 있는데, 쓰기로 재 참조될 가능성을 의미한 쓰기 비중 혹은 쓰기 가중치

(WW: write weight)를 측정하여 그 비중이 가장 가벼운 페이지를 선택하도록 한다. 쓰기 비중을 산출하기 위해 아래의 3가지 지표를 도입한다.

■ 시간적 지역성(TL: temporal locality)

총 참조 횟수가 낮은 페이지는 시간적 지역성이 낮을 것으로 높은 방출 우선순위를 부여한다. 이를 위해 식 1과 같이 윈도우 내 모든 페이지의 평균 참조 횟수 대비 각 페이지의 참조 횟수를 시간적 지역성 지표로 정의한다.

$$TL(page_i) = \frac{Rcount_{page_i} + Wcount_{page_i}}{\sum_{j=1}^w (Rcount_{page_j} + Wcount_{page_j}) * \frac{1}{w}} \quad (1)$$

■ 공간적 지역성(SL: spatial locality)

파일 시스템을 구성하는 LBA(logical block address) 순서 상 특정 페이지의 참조 시간과 바로 전 페이지 참조 시간의 차가 길면 공간적 지역성이 낮으므로 높은 방출 우선순위를 부여한다. 이를 위해 식 2와 같이 각 페이지들 사이의 참조 시간의 차를 공간적 지역성 지표로 정의한다. 이 식에서 $page_{prev(i)}$ 는 LBA 순서 상 $page_i$ 의 바로 전 페이지를 의미한다.

$$SL(page_i) = abs(Reftime_{page_{prev(i)}} - Reftime_{page_i}) \quad (2)$$

■ 쓰기 비율(WR: write ratio)

참조 횟수 중, 쓰기 참조 횟수가 적은 페이지는 쓰기 부담이 낮으므로 높은 방출 우선순위를 부여한다. 이를 위해 식 3과 같이 윈도우 내 모든 페이지의 평균 참조 횟수 대비 각 페이지의 쓰기 참조 횟수를 쓰기 비중 지표로 정의한다.

$$WR(page_i) = \frac{Wcount_{page_i}}{\sum_{j=1}^w (Rcount_{page_j} + Wcount_{page_j}) * \frac{1}{w}} \quad (3)$$

위에서 식 1과 식 2는 쓰기 비중과 반비례 관계에 있으므로 일차적인 쓰기 비중 WW는 아래의 식 4와 같이 구할 수 있다.

$$WW(page_i) = 1/TL(page_i) + SL(page_i) + 1/WR(page_i) \quad (4)$$

그런데, 식 4는 어느 한 항목이 전체 결과 값을 지배할 수 있으므로, 각 항을 0~1 범위로 변환하기 위해 각 지표 및 최종 쓰기 비중을 식 5~식 8과 같이 재정의한다.

$$TL(page_i)' = e^{-1 * \left(\frac{1 + \sum_{j=1}^w (Rcount_{page_j} + Wcount_{page_j}) * \frac{1}{w}}{1 + Rcount_{page_i} + Wcount_{page_i}} \right)} \quad (5)$$

$$SL(page_i)' = e^{-1 * (1 + abs(Reftime_{page_{prev(i)}} - Reftime_{page_i}) * \frac{1}{2})} \quad (6)$$

$$WR(page_i)' = e^{-1 * (\frac{1 + \sum_{j=1}^w (Rcount_{page_j} + Wcount_{page_j}) * \frac{1}{w}}{1 + Wcount_{page_i}})} \quad (7)$$

$$WW(page_i)^{\delta} = TL(page_i)' + SL(page_i)' + WR(page_i)' \quad (8)$$

위의 기초 지표 식 5 ~ 식 7을 기반으로 시뮬레이션을 통해 2WPR 알고리즘의 성능을 분석하고 개선한다.

3-3 실험 환경

■ 실험 플랫폼

분석 및 평가가 이루어진 하드웨어 플랫폼 환경은 표 2와 같다.

표 2. 실험 환경 (하드웨어 플랫폼)

Table 2. Experimental Environment (Hardware Platform)

Items	Specification
CPU	Intel E7500(Duo Core), 2.93GHz
Memory	2.0GB
OS	LINUX(Ubuntu 16.04)
Language	gcc / POSIX thread

■ FTL 시뮬레이터

FTL 시뮬레이션을 위해 DiskSim을 그림 4의 모형으로 축약하여 연구[8]에서 소개한 리눅스 FUSE 기반 시뮬레이터를 사용한다.

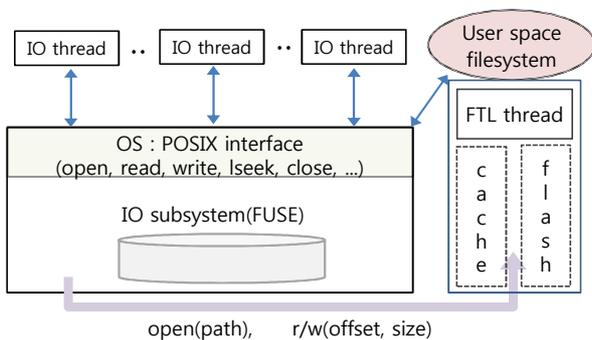


그림 4. FTL 시뮬레이터 구조

Fig. 4. Architecture of the FTL Simulator

■ 캐시 구성

페이지 사이즈 8K, RAM 사이즈 256MB인 SSD를 모델로 하여 이중 128MB는 매핑 테이블로, 나머지 128MB는 캐시로 사용한다. 캐시는 총 16K개인데 이를 128개 그룹으로 구성하고, "LSN(logical sector number) mod 128"의 해시 함수로 캐시 할당

그룹을 결정한다. 페이지 교체는 각 그룹별 128개의 캐시에서 이루어진다. 각 그룹별 128개의 캐시 중 2WPR_List에 90개(약 70%)를, Victim_List에 나머지 38개를 배분한다. 그리고 2WPR_List에서 쓰기 가중치를 적용할 윈도우는 6(약 7%)으로 설정한다.

■ 워크로드 데이터

실험용 데이터로는 실제 현장 트레이스 데이터로 알려진 표 3의 금융 데이터[14]를 활용한다. 이들 금융 데이터는 512 바이트 블록 기반 파일 시스템에서 약 12시간 동안 운영체제 압축력 요청을 모니터링한 결과로 <LBA, size, R/W, time_stamp> 형태로 구성되어 있다.

표 3. 실험 데이터를 위한 워크 로드

Table 3. Experimental Data for Workload

Data	Max LBA	N. of Rec.	I/O Ratio(%)
Financial1	1,215,633	5,334,984	23/77
Financial2	1,215,222	3,699,194	82/18

3-4 평가 및 분석

■ 각 지표들의 유효성 분석

식 5 ~ 식 7에서 도출된 TL', SL', WR' 각 지표의 유효성을 분석하기 위해 이들 지표들의 조합이 아닌 개별적 적용에 따른 쓰기 횟수 추이를 관찰한다. 이를 위해 각 지표에 대하여 서로 상반되는 반대 부호(+, -)의 지표 값을 두 데이터에 적용한 결과를 비교한다. 그림 5와 그림 6이 그 결과인데, 예상과 다르게 반대 부호로 적용했을 때 오히려 더 우수한 결과를 낳는 경우가 많다. 이 결과로부터, 각 지표들에 대한 전체 내용에 내부적인 변수가 있으며, 이들 지표들을 모두 합산하여 최종 쓰기 비중으로 활용하는 방안이 최선이 아닐 수도 있음을 예상할 수 있다. 이들 지표들의 합산 외에 다른 조합들에 대한 실험 필요성이 제기된다.

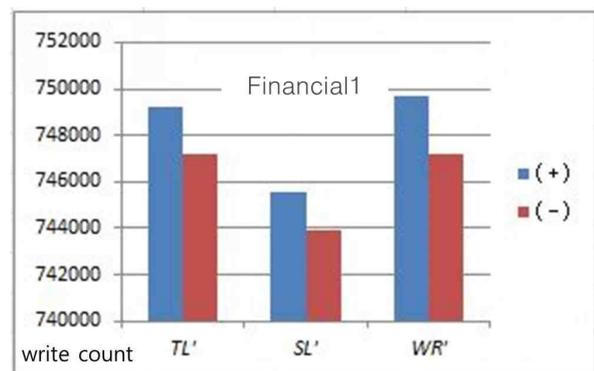


그림 5. Financial1의 개별 지표별 쓰기 횟수 추이

Fig. 5. Write Count Trend of Financial1 with Respective Metric



그림 6. Financial2의 개별 지표별 쓰기 횟수 추이
Fig. 6. Write Count Trend of Financial2 with Respective Metric

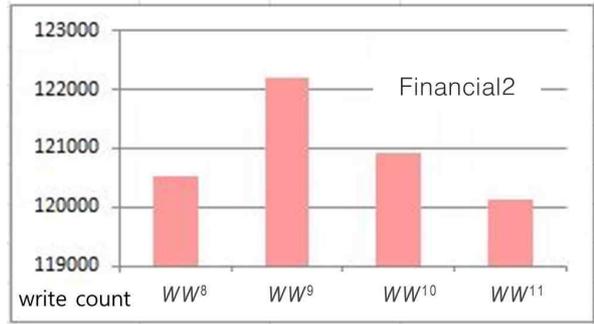


그림 8. 재조합된 쓰기 비중들의 쓰기 횟수 추이(Financial2)
Fig. 8. Write Count Trend with Recombined Write Weight(Financial2)

IV. 2WPR 알고리즘 개선 및 평가

4-1 쓰기 비중 지표들의 재조합

종합 지표인 쓰기 비중을 구성하는 세 가지 지표들의 다양한 조합을 시행착오를 거쳐 실험한 결과, 원래 제안된 식 8 외에 식 9~식 11이 비교해볼 가치가 있는 것으로 판단된다. 식 9는, 그림 5와 그림 6에서 각 지표가 쓰기 횟수 감소에 부정적 영향을 미친 것으로 나타나 각 지표의 부호를 반대로 설정한 것이고, 식 10은 식 8의 각 지표 간 가중치 불균형에 의한 영향을 줄이기 위해 합산 대신 비례식(곱셈)으로 변환한 것이며, 식 11은 그림 5의 쓰기 상위 데이터에서 SL의 부정적 영향이 크게 나타나 식 10을 변형하여 이를 반비례 지표로 반영한 것이다.

$$WW(\text{page}_i)^9 = -TL(\text{page}_i)' - SL(\text{page}_i)' - WR(\text{page}_i)' \quad (9)$$

$$WW(\text{page}_i)^{10} = TL(\text{page}_i)' * SL(\text{page}_i)' * WR(\text{page}_i)' \quad (10)$$

$$WW(\text{page}_i)^{11} = TL(\text{page}_i)' / SL(\text{page}_i)' * WR(\text{page}_i)' \quad (11)$$

그림 7과 그림 8에 새로 조합된 쓰기 비중들의 비교를 위한 식 8(WW8) ~ 식 11(WW11)에 대한 쓰기 감소 효과를 보였다. 이 두 그림에서 보는 바와 같이 공간 지역성 SL'을 반비례 지표로 적용한 식 11이 가장 우수함을 알 수 있다.

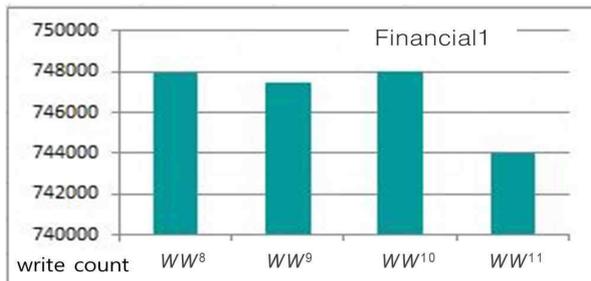


그림 7. 재조합된 쓰기 비중들의 쓰기 횟수 추이(Financial1)
Fig. 7. Write Count Trend with Recombined Write Weight(Financial1)

4-2 새로운 쓰기 비중 도출

식 11의 특징 중의 하나는 쓰기 비중이 지표들의 합산이 아닌 비례식으로 구성되었다는 점이다. 지표들을 비례식으로 구성할 경우 어느 한 지표가 전체를 지배하는 지표 간 가중치 불균형의 위험성을 완화시킬 수 있다. 이 추론의 범위를 좀 더 확장하면 가중치 균등화를 위해 지수 함수로 변환하기 이전의 원천 지표인 식 1 ~ 식 3을 비례식에 곧바로 적용하는 방안을 검토해볼 필요가 있다. 식 11의 쓰기 비중을 변환 이전의 지표들인 식 1 ~ 식 3으로 대체하여 재구성하면 식 12와 같다.

$$WW(\text{page}_i)^{12} = TL(\text{page}_i) * SL(\text{page}_i) * WR(\text{page}_i) \quad (12)$$

4-3 제안 지표의 평가

표 4와 그림 9, 그리고 그림 10은 원래의 쓰기 비중인 식 8(WW8) 대비 지표들을 재조합한 쓰기 비중인 식 11(WW11)과 새로운 쓰기 비중인 식 12(WW12)의 쓰기 감소 효과를 비교해 보이고 있다. 새로 도출된 쓰기 비중인 식 12가 쓰기 비율이 높은 Financial1 워크로드에서 원래의 식 8 대비 쓰기 및 읽기 감소 비율이 약 2%로 나타나 그 효과가 높다. 읽기 비율이 높은 Financial2에서는 쓰기 감소 개선이 약 0.1%로 나타나 효과가 높지 않다. 개선된 쓰기 비중은 쓰기 비율이 높은 워크로드에서 높은 효과를 보인다.

표 4. WW¹과 WW²의 쓰기 감소 효과

Table 4. Write Reduction Effects of WW¹ and WW²

data \ weight	Financial1		Financial2	
	Read	Write	Read	Write
WW ⁸	638,486	747,926	338,928	120,527
WW ¹	677,030	744,001	339,854	120,139
WW ²	627,764	734,656	342,546	120,107

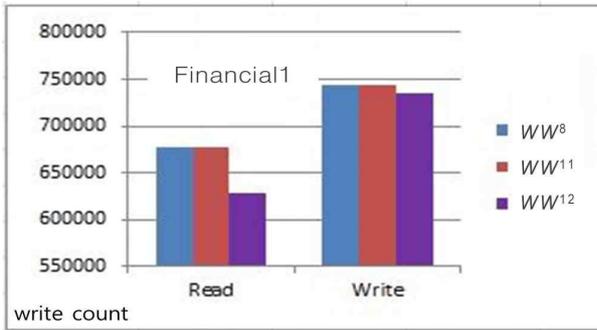


그림 9. 개선된 쓰기 비중의 쓰기 감소 비교(Financial1)
 Fig. 9. Write Reduction Comparison of Improved Write Weight(Financial1)

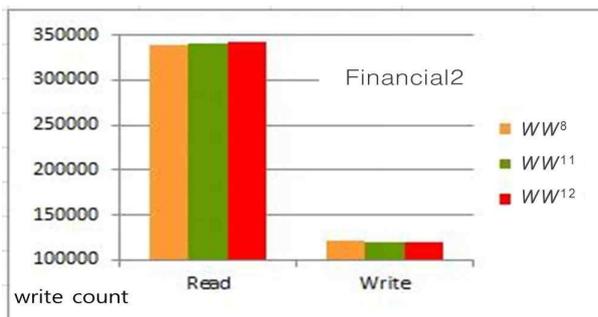


그림 10. 개선된 쓰기 비중의 쓰기 감소 비교(Financial2)
 Fig. 10. Write Reduction Comparison of Improved Write Weight(Financial2)

새로운 쓰기 비중 WW12에 대한 개선 효과의 안정성을 검증하기 위해 표 3의 워크 로드 데이터를 표 5와 같이 1,000,000 여 건 단위로 분라-재구성하여 실험한 결과를 표 6에 보였다. 이 표로부터 새롭게 제안된 쓰기 비중 WW12가 기존의 쓰기 비중인 WW8보다 낮은 경우는 발견되지 않음을 확인할 수 있다.

표 5. 실험 데이터의 재구성

Table 5. Reconfiguration of Experimental Data

Data	Name	Max LBA	N. of Rec.	I/O Ratio(%)
Financial1	Fin11	1,215,633	1,050,000	16/84
	Fin12	1,139,449		13/87
	Fin13	1,215,633		18/82
Financial2	Fin21	1,213,863	1,200,000	76/24
	Fin22	1,213,863		79/21
	Fin23	1,215,222		80/20

표 6. 재구성 데이터에서 WW⁸과 WW¹²의 쓰기 횟수

Table 6. Write Count of WW8 and WW12 in Reconfigured Data

data weight	Fin11	Fin12	Fin13	Fin21	Fin22	Fin23
WW ⁸	218,036	126,513	137,780	46,723	38,930	42,609
WW ¹²	216,420	125,339	136,780	46,471	38,742	42,300

V. 결 론

플래시 메모리 기반 SSD는 여러 가지 장점이 있는 반면 쓰기 팽창 및 마모 현상이라는 단점이 있다. 이 연구에서는 SSD의 쓰기 횟수를 감소시키기 위한 2WPR SSD 캐시 페이지 교체 알고리즘을 분석·평가하고 개선 과정을 제시하였다. 쓰기 감소 영향 인자들은 복합적인 상호 작용 관계로 인해 희생 페이지 선택의 기준이 되는 쓰기 비중을 결정론적으로 도출하는 일이 쉽지 않다. 따라서, 쓰기 비중은 수많은 시행착오를 거치는 휴리스틱 방법으로 도출되어야 하고, 시간적 성능과는 달리 단 1회의 쓰기 감소라도 쓰기 팽창에 의한 누적 과급효과가 적지 않다는 점에서, 안정적인 개선 비율 2%가 비록 수치상으로는 크지 않지만 의미가 크다고 볼 수 있다. 더불어, 제시된 개선 과정을 바탕으로 향후 더 나은 지표 도출을 기대할 수 있다.

참고문헌

- [1] H. C. Song, H. J. Ahn, "Development of Efficient Data Distribution Storage Algorithm for High Speed Data Backup in DRAM based SSD," The Journal of The Institute of Internet, Broadcasting and Communication (IIBC), Vol. 15, No. 6, pp. 11-15, 2015.
- [2] H. B. Lee, T. Y. Chung, "A Comparative Analysis on Page Caching Strategies Affecting Energy Consumption in the NAND Flash Translation Layer," IEMEK Journal of Embedded Systems and Applications, Vol. 13, No. 3, pp. 109-116, 2018.
- [3] I. H. Shin, "Performance Evaluation of Applying Shallow Write in SSDs with Internal Cache," Journal of KIIT, Vol. 17, No. 1, pp. 31-38, 2019.
- [4] definition of SSD [Internet]. Available: <https://namu.wiki/w/SSD>.
- [5] definition of write amplification [Internet]. Available: https://en.wikipedia.org/wiki/Write_amplification.
- [6] X. Y. Hu, E. Eleftheriou, R. Haas, I. Iliadis and R. Pletka, "Write amplification analysis in flash-based solid state drives," in Proceedings of SYSTOR 2009: The Israeli Experimental Systems Conference, pp. 1-9, 2009.
- [7] W. H. Lee, J. H. Kwak, "2WPR: Disk Buffer Replacement Algorithm Based on the Probability of Reference to Reduce the Number of Writes in Flash Memory," Journal of The Korea Society of Computer and Information, Vol. 25, No. 2, pp. 1-10, 2020.
- [8] H. B. Lee, K. H. Kwon, "Implementation of a Light Weight SSD Performance Evaluation Simulator based on Linux FUSE," Journal of Digital Contents Society, Vol. 20, No. 12, pp. 2545-2553, 2019.

[9] V. Mohan, T. Bunker, L. Grupp, S. Gurumurthi, M. R. Stan, S. Swanson, "Modeling Power Consumption of NAND Flash Memories Using FlashPower," IEEE Transactions on Computer-Aided Design of Integrated Circuit and Systems, Vol. 32, No. 7, pp. 1031-1044, 2013.

[10] R. Agrarwal, M. Marrow, "A closed-form expression for Write Amplification in NAND Flash," in Proceedings of IEEE Globecom 2010 Workshop on Application of Communication Theory to Emerging Memory Technologies, pp. 1846-1850, 2010.

[11] S. Y. Park, D. W. Jung, J. U. Kang, J. S. Kim, J. W. Lee, "CFLRU: A Replacement Algorithm for Flash Memory," in Proceedings of the 2006 international conference on Compilers, architecture and synthesis for embedded systems(CASES'06), pp. 234-241, 2006.

[12] H. Y. Jung, H. K. Shim, S. M. Park, S. Y. Kang, and J. H. Cha, "LRU-WSR: Integration of LRU and writes sequence reordering for flash memory," IEEE Transactions on Consumer Electronics, Vol. 54, No. 3, pp. 1215-1223, 2008.

[13] P. Jin, Y. Ou, T. Harder, and Z. Li, "AD-LRU: An efficient buffer replacement algorithm for ash-based databases," Data Knowledge Engineering, Vol. 72, pp. 83-102, 2012.

[14] UMass I/O Trace Repository [Internet]. Available: <http://traces.cs.umass.edu/index.php/Storage/Storage>.



이형봉(Hyung-Bong Lee)

1984년 : 서울대학교 계산통계학과(학사)
 1986년 : 서울대학교 대학원 계산통계학과(석사)
 2000년 : 강원대학교 대학원 컴퓨터과학과(박사)

1986년~1994년: LG전자 컴퓨터연구소

1994년~1999년: 한국디지털(주)

2004년~현 재: 강릉원주대학교 컴퓨터공학과 교수

※관심분야 : 무선 통신 (Wireless Networks), 센서 네트워크 (Sensor Networks), 임베디드 시스템 (Embedded Systems), 사물 인터넷 (IoT)



권기현(Ki-Hyeon Kwon)

1993년 : 강원대학교 컴퓨터과학과(학사)
 1995년 : 강원대학교 대학원 컴퓨터과학과(석사)
 2000년 : 강원대학교 대학원 컴퓨터과학과(박사)

1998년~2002년: 동원대학 인터넷정보과 교수

2002년~현 재: 강원대학교 정보통신공학과 교수

※관심분야 : 사물 인터넷 (IoT), 인공지능 응용