JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 25 No. 2, pp. 21-29, February 2020
https://doi.org/10.9708/jksci.2020.25.02.021

CPWL : Clock and Page Weight based Disk Buffer Management Policy for Flash Memory Systems

Byung Kook Kang*, Jong Wook Kwak*

*Student, Dept. of Computer Engineering, Yeungnam University, Gyeongsan, Korea
*Professor, Dept. of Computer Engineering, Yeungnam University, Gyeongsan, Korea

[Abstract]

The use of NAND flash memory is continuously increased with the demand of mobile data in the IT industry environment. However, the erase operations in flash memory require longer latency and higher power consumption, resulting in the limited lifetime for each cell. Therefore, frequent write/erase operations reduce the performance and the lifetime of the flash memory. In order to solve this problem, management techniques for improving the performance of flash based storage by reducing write and erase operations of flash memory with using disk buffers have been studied. In this paper, we propose a CPWL to minimized the number of write operations. It is a disk buffer management that separates read and write pages according to the characteristics of the buffer memory access patterns. This technique increases the lifespan of the flash memory and decreases an energy consumption by reducing the number of writes by arranging pages according to the characteristics of buffer memory access mode of requested pages.

▸ **Key words**: Disk buffer, Flash memory, LRU algorithm, CLOCK algorithm, Page replacement, Page access patterns

[요 약]

IT 산업 환경에서 모바일 데이터의 수요 증가로 인해 NAND 플래시 메모리의 사용이 지속적으로 증가하고 있다. 하지만, 플래시 메모리의 소거 동작은 긴 대기 시간과 높은 소비 전력을 요구하여 각 셀의 수명을 제한한다. 따라서 쓰기와 삭제 작업을 자주 수행하면 플래시 메모리의 성능과 수명이 단축된다. 이런 문제를 해결하기 위해 디스크 버퍼를 이용, 플래시 메모리에 할당되는 쓰기 및 지우기 연산을 감소시켜 플래시 메모리의 성능을 향상시키는 기술이 연구되고 있다. 본 논문에서는 쓰기 횟수를 최소화하기 위한 CPWL 기법을 제안한다. CPWL 기법은 버퍼 메모리 액세스 패턴에 따라 읽기 및 쓰기 페이지를 나누어 관리한다. 이렇게 나뉜 페이지를 정렬하여 쓰기 횟수를 줄이고 결과적으로 플래시 메모리의 수명을 늘리고 에너지 소비를 감소시킨다.

▸ **주제어**: 디스크 버퍼, 플래시 메모리, LRU 알고리즘, CLOCK 알고리즘, 페이지 교체, 페이지 접근 패턴

• First Author: Byung Kook Kang, Corresponding Author: Jong Wook Kwak
*Byung Kook Kang (kangbk@yu.ac.kr), Dept. of Computer Engineering, Yeungnam University
*Jong Wook Kwak (kwak@yu.ac.kr), Dept. of Computer Engineering, Yeungnam University
• Received: 2019. 12. 24, Revised: 2020. 01. 15, Accepted: 2020. 01. 17.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

최근 5G망, 모바일 데이터 증가 등의 정보 산업의 트렌드 변화에 따라 빠른 연산 속도를 이용해 많은 데이터를 처리하고 저장하기 위해 NAND 플래시 메모리의 사용이 증가 되고 있다. 플래시 메모리는 DRAM과는 달리 전원이 꺼져도 저장된 내용물이 사라지지 않는 비휘발성 메모리로, 높은 직접도와 내구성으로 기존 저장 매체인 HDD를 대체하고 있다[1].

하지만 플래시 메모리는 읽기와 쓰기 연산에 있어 그 단위가 페이지로 한정되며, 제자리 덮어쓰기가 불가능하다는 특징이 있다. 따라서 플래시 메모리는 덮어쓰기를 하기 위하여 해당 페이지가 포함된 블록 전체를 삭제 한 후에 쓰기가 이루어진다[2]. 표 1은 플래시 메모리의 읽기, 쓰기 및 삭제 연산의 접근 시간과 단위를 보여준다[3-5]. 삭제 연산은 지연 시간과 전력 소모가 크고, 한 셀에 대한 제한된 수명이 있어 빈번한 쓰기/삭제 연산은 플래시 메모리의 성능과 수명을 저하시키는 요인이 된다. 이러한 문제점을 해결하기 위해 디스크 버퍼를 이용하여 플래시 메모리의 쓰기 및 삭제 연산을 줄이고, 이를 통한 스토리지의 성능 향상을 유도하는 관리 기법들이 연구 되었다.

Table 1. The characteristics of flash memory

Operation	Access Time	Access granularity
Read	80 μs	page (2KB)
Write	200 μs	page (2KB)
Erase	1.5 ms	block (128KB = 4pages)

플래시 메모리 기반의 디스크 버퍼 관리 기법은 플래시 메모리의 연산 효율을 높이기 위해 주기억 장치의 일부 혹은 별도의 DRAM 공간을 활용한다. 해당 공간을 최근에 사용된 내용을 기억하는 디스크 버퍼 영역으로 할당한다. 이를 사용하면 버퍼에 내용이 있을 경우 디스크 액세스 없이 바로 사용이 가능하므로 읽기 및 쓰기 효율을 높이게 되고, 그로 인한 플래시 메모리의 성능 및 태생적 한계인 수명에 대한 향상을 기대할 수 있다. 디스크 버퍼 관리 정책으로는 LRU, CLOCK과 같은 알고리즘과 이를 개선한 CF-LRU, CCF-LRU, LRU-WSR등 여러 DRAM 디스크 버퍼 기반의 기법들이 제안되고 있다[6-9].

본 논문에서는 버퍼 메모리를 두 개의 영역으로 나누어 읽기 / 쓰기 페이지를 분리하여 배치하는 디스크 버퍼 관리 기법을 제시한다. 제안된 기법은 요청되는 페이지에 접근하는 연산의 종류에 따라 버퍼 메모리의 특성에 맞게 페이지를 배치하여 쓰기 횟수를 줄인다. 이를 통해 플래시 메모리의 수명은 증가시키고 에너지의 소모는 감소시킨다.

본 논문의 구성은 다음과 같다. 2장은 플래시 메모리의 특성과 기존에 연구된 디스크 버퍼 관리 정책에 대하여 설명한다. 3장에서는 제안된 시스템 전체 구조와 알고리즘에 관해 서술한다. 4장에서는 기존에 제안된 기법들과의 비교로 본 논문에서 제안하는 기법의 차별성을 확인할 것이다. 5장에서는 결론을 제시한다.

II. Background and Related works

본 장에서는 플래시 메모리의 특징에 대해 살펴보고 디스크 버퍼를 관리하는 기존의 정책을 소개한다.

1. Background

플래시 메모리는 전력 공급이 없더라도 데이터가 보존되는 비휘발성 저장 장치이며, 전기적으로 데이터를 저장하고 지우므로 하드디스크보다 필요 전력이 적고 데이터 접근 속도 역시 빠르다. 플래시 메모리는 최소 저장 단위인 셀(Cell)의 배열에 따라 노어(NOR) 및 낸드(NAND) 플래시 메모리로 나뉜다.

노어 플래시 메모리는 병렬로 연결된 셀의 구조로 인하여 고속의 임의 접근 읽기가 가능하지만 쓰기 연산의 수행 속도가 느리며, 낮은 직접도로 인해 대용량화하기 어려운 단점이 있다. 그에 비하여 낸드 플래시 메모리는 직렬로 연결된 셀의 구조로 인해 노어 플래시 메모리에 비해 읽기 연산 속도는 느리지만, 쓰기 연산 속도가 상대적으로 빠르며 특히 직접도가 높아 대용량화하기 용이하여 데이터를 저장하기 위한 용도로 적합하다[10]. 이와 같은 플래시 메모리의 특성을 바탕으로 데이터의 휘발성이 없으며 기존의 하드디스크보다 향상된 입출력 속도를 갖는 저장 장치인 SSD(Solid State Drive)가 개발되었다.

그러나 플래시 메모리는 연산 수행 시간의 비대칭 특성으로 쓰기 연산에 발생하는 비용이 읽기 연산의 그것보다 크며, 데이터가 존재하는 페이지에 대한 덮어쓰기 연산 발생 시 지우기 연산을 먼저 수행한 후에 쓰기 연산을 수행해야 한다. 이러한 제자리 덮어쓰기가 자주 발생할 경우 수행속도가 가장 느린 지우기 연산이 자주 발생하여 시스템의 성능이 저하되며, 셀 당 쓰기 및 삭제 회수 제한으로 인해 메모리의 수명이 단축된다[11]. 이러한 한계점으로 인해 플래시 메모리를

기반으로 하는 모바일이나 서버 환경의 데이터베이스 시스템이 기존의 하드 디스크 기반의 디스크 버퍼 정책을 적용할 때 기대에 미치지 못하는 성능을 보일 수 있다. 따라서 모바일이나 데이터베이스 시스템의 성능 향상을 위해 하드디스크에 최적화된 기법들을 플래시 메모리의 특성에 맞추려는 시도가 지속적으로 이루어지는 중이다.

2. Related works

플래시 메모리의 개발 이전에는 하드디스크를 위한 디스크 버퍼 기법들이 주로 연구되었다. 그러나 하드디스크 기반의 디스크 버퍼 관리 기법들은 그림 1과 같이 플래시 저장장치를 보조기억장치로 사용하는 시스템에 적용했을 때 비대칭적인 연산 수행 시간과 제자리 덮어쓰기가 불가능하다는 특성으로 인하여 시스템 성능이 저하될 것이다. 이런 플래시 메모리의 문제를 개선하기 위한 여러 시도가 이루어지고 있다. 특히, 플래시 메모리의 성능 향상을 위해 기존의 LRU(Least Recently Used)를 개선하는 연구들이 진행되고 있다. CF-LRU(Clean First Least Recently Used) 기법은 플래시 메모리의 쓰기 연산 비용을 줄이기 위한 기법이다. 해당 기법은 버퍼를 용도에 맞게 두 가지로 나누는 것을 핵심으로 한다[7]. 그 중 워킹 영역은 최근 참조된 페이지들이 저장하고, 클린 퍼스트 영역은 그렇지 않은 페이지들이 저장된다. 버퍼에서 페이지 교체가 필요할 때 클린 퍼스트 영역에서 LRU에 가까운 클린 페이지(Clean Page)를 우선적으로 버퍼에서 퇴거시켜 쓰기 연산 비용을 절감한다. 그러나 콜드 더티 페이지(Cold Dirty page)는 유지하고 핫 클린 페이지(Hot Clean page)는 내보내는 경우가 발생할 수 있는데, 이는 페이지

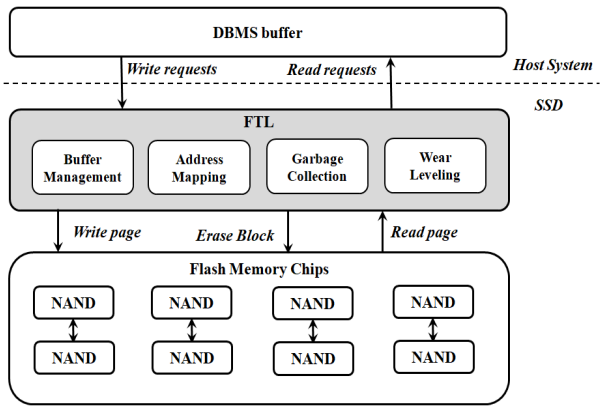


Fig. 1. The structure of flash-based DBMS

의 접근 빈도를 고려하지 않는 형태로 시스템의 성능 저하를 초래할 수 있다. 한편, LRU-WSR(LRU-Write Sequence Reorderging) 기법은 버퍼 교체 시 LRU의 퇴거 후보 페이지가 클린 페이지일 경우 버퍼에서 퇴거시키고 더티 페이지일 경우 콜드 페이지이면 희생자 페이지로 간주한다[12]. 하지만 이 기법은 클린 페이지의 액세스 빈도와 더티 페이지의 최신성을 고려하지 않아 콜드 더티 페이지와 핫 클린 페이지가 더 빨리 퇴거되는 문제가 발생할 수 있다. 마지막으로 DPW-LRU 기법은 버퍼를 두 개의 영역으로 나누어 쓰기 페이지와 읽기 페이지를 별도의 영역에 저장하도록 하고, 쓰기 페이지에 대해 가중치를 두어 희생자 페이지로 선정되는 것을 줄임으로써 쓰기 연산을 감소시킬 수 있다[13].

이와 같이 플래시 메모리 기반의 버퍼 관리 기법들은 페이지 교체 과정에서 더티 페이지의 디스크 쓰기 연산을 최소화함으로써 비용을 줄이는데 중점을 두고 있다. 이를 위해 연구된 플래시 메모리 대상의 버퍼 관리 기법들은 크게 두 가지 형태로 분류될 수 있다. 첫째는 쓰기 연산을 줄이기 위해 클린 페이지를 우선적으로 버퍼에서 내보내는 방법, 둘째는 각 페이지의 시간적 지역성을 고려하는 방법이다. 이를 통해 캐시에 대한 적중률을 높이고 페이지 교체의 최소화를 꾀한다. 이러한 형태를 고려하여 현재 제안된 기법들보다 좀 더 개선된 방안을 찾아 이를 플래시 메모리에 적용하는 연구가 필요하다.

III. CPWL policy

1. Motivation

본 논문에서는 쓰기 연산과 시간적 지역성을 고려한 디스크 버퍼 관리 기법을 적용함으로써, 에너지 소모를 감소시키고 쓰기 횟수를 줄임으로써 플래시 메모리의 수명감소를 최소화하는 디스크 버퍼 관리 기법을 제안한다. 본 논문에서 제안하는 방법은 버퍼를 두 개의 영역(LSR: Long Stay Region, SSR: Short Stay Region)으로 나누어, 이를 페이지 가중치 방식의 LRU와 Clock으로 적용한다. LSR은 페이지 가중치 방식의 LRU를 적용하여 쓰기 페이지에 대한 히트율을 높이고, SSR은 Clock을 적용하여 읽기 페이지에 대한 히트율을 높이고 지연 시간을 줄인다. 제안된 기법은 기존에 제안된 디스크 버퍼 정책과 비교할 때 에너지 소모 및 쓰기 횟수를 감소시킬 수 있는데, 이는 플래시 메모리의 수명 증대라는 결과로 돌아올 수 있다.

2. CPWL operation

본 논문에서 제안하는 CPWL(Clock and Page Weight-LRU) 기법은 쓰기 페이지에 대한 요청은 Page Weight LRU 방식으로, 읽기 페이지에 대한 요청은 Clock 방식으로 구현하여, 지연시간 및 쓰기 연산의 횟수를 감소시킴으로써 플래시 메모리의 수명 문제를 개선할 수 있는 디스크 버퍼 관리 기법이다. 제안하는 전체 시스템의 구조는 그림 2와 같이 나타낼 수 있다.

호스트 시스템에서의 플래시 메모리에 대한 읽기나 쓰기 요청은 디스크 버퍼를 통해서 이루어진다. 디스크 버퍼는 호스트의 읽기/쓰기 요청에 따라 LSR 또는 SSR에 페이지를 쓰거나 읽는다. LSR은 쓰기 페이지에 대해 MRU로 배치하고, LSR의 공간 부족으로 인해 퇴거 발생 시 페이지 가중치 계산을 통해 가중치 값(w)이 작은 페이지를 SSR 영역으로 이주시킴으로써 쓰기 비용을 줄이고 페이지의 참조율을 높일 수 있다. SSR은 읽기 페이지에 대해 Clock 기법을 이용하여 배치하고 공간 부족 시 참조 카운터가 0인 페이지를 퇴거시킴으로써 지연시간을 줄일 수 있다. 페이지 참조(Page hit)가 발생할 경우, LSR에서 페이지 참조 발생시 해당 페이지를 LSR의 MRU로 이동시키고, SSR에서 페이지 참조 발생시 페이지 접근 모드가 쓰기일 경우에는 해당 페이지를 LSR의 MRU로 이동시킴으로써 쓰기 연산을 줄일 수 있다.

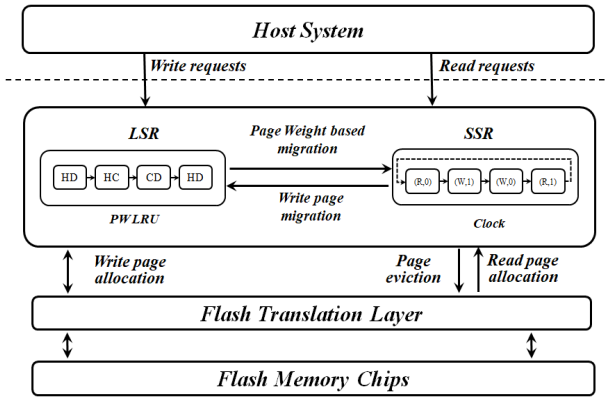


Fig. 2. The structure of CPWL

2.1. Page migration within and between regions

그림 3은 페이지 참조 발생시 LSR과 SSR 간의 페이지 이동 과정을 보여준다. LSR에서 페이지 참조가 발생하면 LSR의 MRU로 페이지를 이동 시킨다 (그림3 ⓐ). SSR에서 페이지 참조가 발생할 경우는 페이지 접근 모드에 따라 달라진다. 페이지의 접근 모드가 읽기일 경우, Clock 방식에 따라 페이지의 참조 카운터(Reference Count)를 1로 설정한다 (그림3 ⓑ). 반면 페이지 접근 모드가 쓰기일 경우 LSR의 MRU로 이주시킨다. 만약 LSR이 Full인 경우, LSR에서 페이지 가중치(w)가 가장 적은 페이지를 선정해서 (그림3 ⓒ) 이를 SSR로 이주시킨 후 (그림3 ⓓ) SSR의 참조된 페이지를 LSR의 MRU에 배치한다 (그림3 ⓔ).

2.2. Page eviction and insertion when page miss

그림 4는 디스크 버퍼의 페이지 배치 및 이주에 대한 과정을 보여준다. 페이지의 접근 모드가 읽기(read)인 경우 SSR 영역에서 CLOCK 알고리즘을 통한 페이지 배치가 이루어진다(그림4 ⓐ). 하지만 페이지의 접근 모드가 쓰기(write)인 경우 LSR 영역에 페이지가 배치된다. 이때 LSR에 쓰기 가능한 여유 공간이 있으면 LSR의 MRU에 페이지를 배치한다.

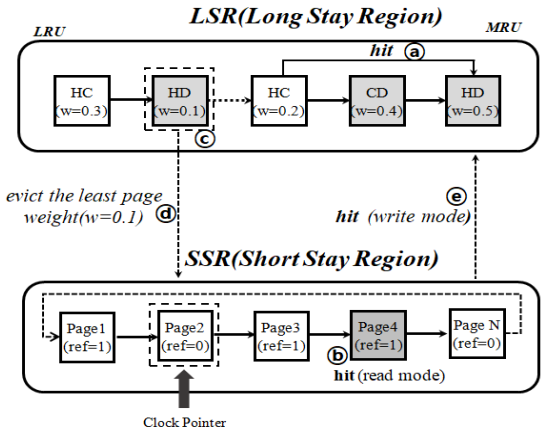


Fig. 3. The example of a page migration

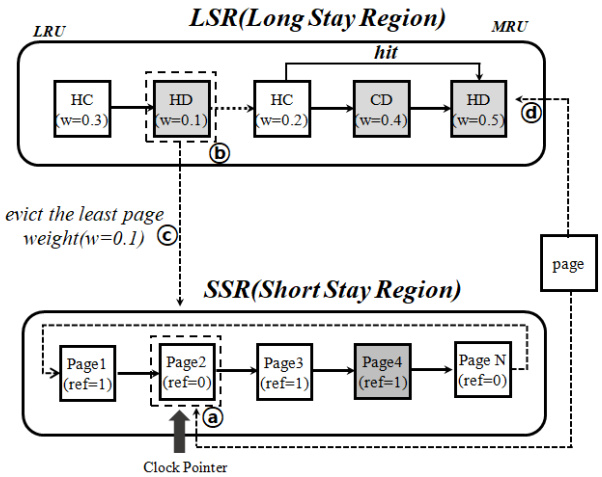


Fig. 4. The example of a page allocation

그러나 LSR이 Full인 경우 페이지 가중치 계산을 통해 희생 페이지를 선정하고 (그림4 ⓑ) 이를 SSR로 이주 시킨 후 (그림4 ⓒ) LSR의 MRU로 페이지를 배치시킨다 (그림4 ⓓ).

2.3. CPWL Algorithm

LSR에서의 희생자 페이지 선택에는 페이지 가중치를 이용한 PWL 알고리즘을 이용하고, SSR에서는 Clock 알고리즘을 이용해 희생자 페이지를 선택한다. PWL 알고리즘의 주요 구성요소는 시간 지역성의 합(TLS)과 최신성(RE)이다.

전체 페이지 요청 R은 수식 (1)과 같다. j는 페이지 접근의 순차 번호이고, n은 전체 페이지 요청 개수이다.

$$R = \{R_1, R_2, ..., R_j, ..., R_n\}, \ 1 \le j \le n \quad (1)$$

페이지 요청에서 특정 페이지를 저장하기 위한 집합 U는 수식 (2)와 같다. Pi는 전체 페이지 요청에서 유일한 페이지이고 m은 유일한 페이지 요청의 개수이다.

$$U = \{P_1, P_2, ..., P_i, ..., P_m\}, \ 1 \le i \le m \le n \quad (2)$$

U에서 특정 페이지의 접근 순서 집합은 수식 (3)과 같이 표현할 수 있다. ASk는 Pi의 k번째 접근 정보이다.

$$AS^i = \{AS_1, AS_2, ..., AS_k, ..., AS_o\}, \ 1 \le k \le o \le n \quad (3)$$

위의 정보를 이용한 특정 페이지 Pi의 시간 지역성의 합(TLS)은 수식 (4)와 같다.

$$P_i^{TLS} = \sum_{k=1}^{o} AS_k^i \quad (4)$$

최신성(RE)은 수식 (5)와 같다.

$$P_i^{RE} = \frac{AS_{latest} - AS_o}{AS_{latest}}, \ AS_o \le AS_{latest} \quad (5)$$

TLS와 RE를 요소로 한 PWL 알고리즘은 수식 (6)과 같이 계산된다.

$$P_i^w = \frac{P_i^{TLS} * o}{P_i^{RE}} \quad (6)$$

알고리즘 1은 CPWL를 수도 코드(Pseudo Code)로 나타낸 것이다. 요청 페이지가 LSR에서 참조가 일어날 경우 LSR의 MRU로 이동하게 된다(1-2행). 페이지 참조가 SSR에서 일어나고 접근 모드가 쓰기일 경우 해당 페이지를 LSR로 이동시킨다. 이때 LSR에 빈공간이 있을 경우 페이지를 LSR로 이동시키고, LSR에 빈공간이 없을 경우 PWL 알고리즘을 이용해 희생자 페이지를 먼저 선택한다. 그리고 요청 페이지는 LSR의 MRU로 이동 시키고 희생자 페이지는 SSR로 이동 시킨다(3-12행). 페이지 참조가 SSR에 일어나고 접근 모드가 읽기인 경우 SSR에서 Clock 알고리즘을 수행한다(13-15행). 페이지 폴트(Page Fault)의 경우에는 접근 모드가 읽기이면 SSR에서 Clock 알고리즘을 수행하고(17-18행), 접근모드가 쓰기이며 LSR에 빈공간이 있을 경우 페이지를 LSR의 MRU로 삽입한다 (20-21행). LSR에 빈공간이 없을 경우 PWL 알고리즘을 이용해 희생자 페이지를 먼저 선택하고 요청 페이지는 LSR의 MRU에 넣고 희생자 페이지는 SSR로 이동시킨다. SSR의 희생자 페이지는 플래시 메모리로 퇴거시킨다 (22-27행)

Algorithm 1. CPWL algorithm

```
input   page : a requested page;
        Pam : the access mode of a page;
        LSR : long stay region
        SSR : short stay region
 1  if page is in LSR then
 2     move page to MRU of LSR;
 3  else if page is in SSR then
 4     if Pam == Write then
 5        if LSR is not full then
 6           insert page into MRU of LSR;
 7           remove page in SSR;
 8        else
 9           victim = PWL(page);
10           insert page into MRU of LSR;
11           insert victim into SSR;
12        end if
13     else  // Pam == Read
14           run clock algorithm in SSR;
15     end if
16  else      // page miss.
17     if Pam == Read then
18        run clock algorithm in SSR;
19     else  // Pam == Write
20        if LSR is not full then
21           insert page into MRU of LSR;
22        else
23           victim = PWL(page);
24           insert page into MRU of LSR;
25           insert victim into SSR;
26           evict victim in SSR to the flash;
27        end if
28     end if
29  end if
```

IV. Performance Evaluation

1. Experimental setup

제안하는 시스템의 구조와 알고리즘의 성능 평가를 위해 워크로드 기반의 버퍼 시뮬레이터를 구현하여 실험을 하였다. 시스템은 64GB의 크기를 갖는 플래시 메모리에 각각 4MB, 8MB, 16MB, 32MB, 64MB의 사이즈로 버퍼를

구성하였고, 한 페이지 크기는 4KB, 하나의 블록은 64개의 페이지로 구성되어 있다.

표 2는 성능 평가 요소에서 사용된 낸드 플래시 속성 값이다. 비교 대상으로는 LRU와 DPW-LRU 기법을 선택하였으며, 버퍼 영역은 페이지 단위로 관리된다. DPW-LRU와 제안 방식인 CPWL은 두 개의 영역을 5:5로 분할해서 실험을 하였다.

Table 2. The parameter in experiment

Parameter	Description
Page size (bytes)	4096
Block size (pages)	64
Read latency (μs)	25
Write latency (μs)	200
Erase latency (μs)	1,500

Table 3. The characteristics of workloads

Trace name	Read/Write ratio (%)	Total read counts	Total write counts
ascii	78 : 22	806397	210281
home	4 : 96	39055	908863
wdev	28 : 72	65148	171448

본 논문에서는 플래시 메모리의 에너지 소모 감소 및 수명 증가를 검증하고자 쓰기 연산의 비율이 큰 워크로드를 다수 선택하여 사용하였다. 표 3은 실험에 사용된 워크로드의 특성을 읽기와 쓰기 비율 및 그 횟수로 구분하여 나타내었다.

2. Experimental result

본 논문에서 제안된 기법인 CPWL은 각각의 워크로드에서 쓰기 연산을 줄임으로써 지연시간 감소와 플래시 메모리의 수명을 증가 시키는 효과를 기대한다. 실험에 사용된 성능 평가 요소는 히트율(hit ratio), 플래시 메모리에 대한 쓰기 횟수(write count), 그리고 지연시간(latency)이다.

히트율은 쓰기 참조율과 읽기 참조율의 합을 쓰기 횟수와 읽기 횟수의 합으로 나눠 계산하였다. 수식 (7)은 이를 나타낸 것이다.

$$hit\ ratio = \frac{T_{read\,hit} + T_{write\,hit}}{T_{read} + T_{write}} \times 100 \qquad (7)$$

지연 시간은 참조 횟수, 플래시 쓰기/읽기 횟수에 대한 지연 시간과 블록 삭제 횟수에 대한 지연 시간의 합으로 계산하였다. 수식 (8)은 이를 나타낸 것이다. 수식에서 BECgc는 가비지 컬렉션으로 인해 발생하는 블록 소거 횟수이다.

$$\begin{aligned} total\ latency = {} & (Buffer_{hit\,count} \times Dram_{latency}) + \\ & (Flash_{write\,count} \times Flash_{write\,latency}) + \\ & (Flash_{read\,count} \times Flash_{read\,latency}) + \\ & (BEC_{gc} \times Block_{erase\,latency}) \end{aligned} \qquad (8)$$

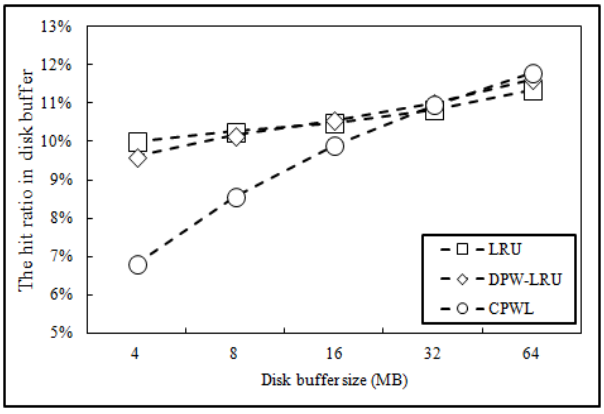
그림 5는 각 기법의 히트율을 나타낸 것이다. 쓰기 연산이 많은 워크로드 (b) home의 경우 CPWL 알고리즘이 가장 높은 히트율을 보이며, 읽기 연산이 많은 (a) ascii의 경우 버퍼 크기가 커질수록 CPWL의 히트율이 높아진다. 그 외의 경우는 전반적으로 LRU의 히트율이 높다.

그림 6은 디스크 버퍼로부터 페이지 퇴거로 인해 스토리지에 쓰기가 발생하는 횟수를 나타낸 것이다. 쓰기 연산에 대한 페이지 참조율을 높이기 위한 PWL 알고리즘 적용으로 CPWL 기법이 쓰기 횟수가 가장 적음을 확인 할 수 있다. 다만 읽기 연산이 큰 (a) 워크로드의 경우 버퍼크기가 8MB보다 작은 경우 DPW-LRU의 쓰기 횟수가 더 적지만, 버퍼크기가 16MB보다 커질 경우 제안된 기법의 쓰기 횟수가 가장 적다.

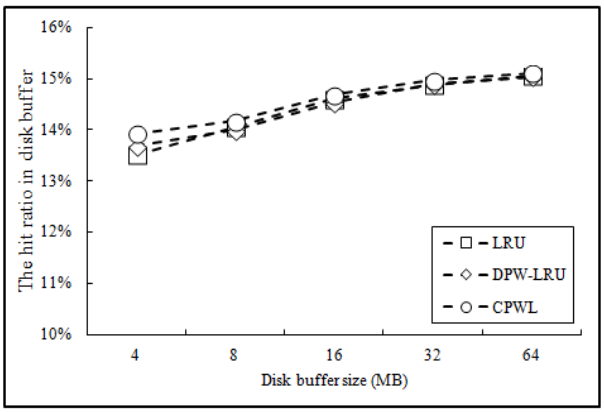
그림 7은 쓰기 / 읽기 / 지우기 연산 횟수를 반영한 전체 지연시간을 나타낸 것이다. 쓰기 연산이 많은 워크로드 (b)와 (c)의 경우 CPWL의 지연 시간이 가장 작으며, 읽기 연산이 많은 (a) ascii의 경우는 쓰기 회수와 동일한 패턴을 보인다. 이를 통해 플래시 메모리의 쓰기 연산이 지연시간과 밀접한 관련이 있음을 알 수 있다.

V. Conclusions

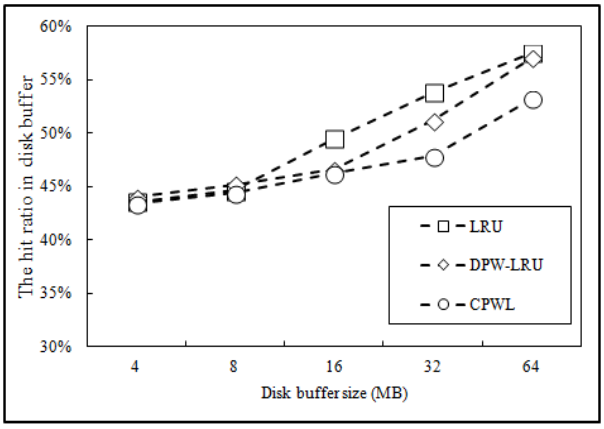
본 연구에서는 쓰기 연산과 시간 지역성을 고려한 버퍼 관리 기법을 적용함으로써 에너지 소모를 감소시키고, 쓰기 횟수를 줄여 플래시 메모리의 수명 감소를 최소화 하는 디스크 관리 기법을 제안하였다. 이를 위해 버퍼를 LSR(Long Stay Region)과 SSR(Short Stay Region)의 2개의 영역으로 나누고 페이지 가중치 기법을 사용하여 쓰기 페이지들이 버퍼에 오래 머물러 있도록 하면서 읽기 페이지들에 대한 참조율도 높이도록 하였다.



(a) ascii

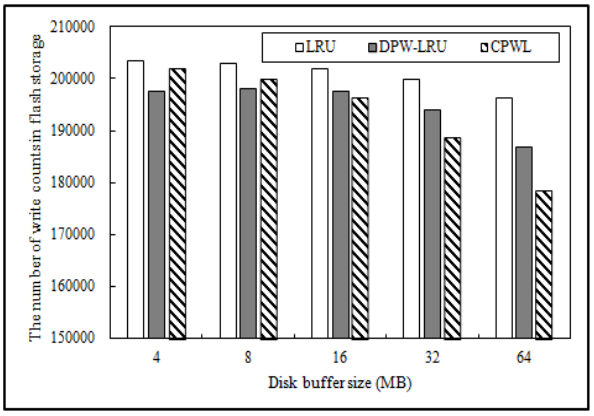


(b) home

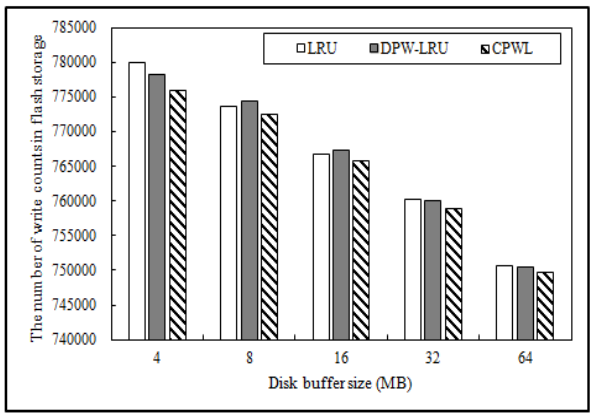


(c) wdev

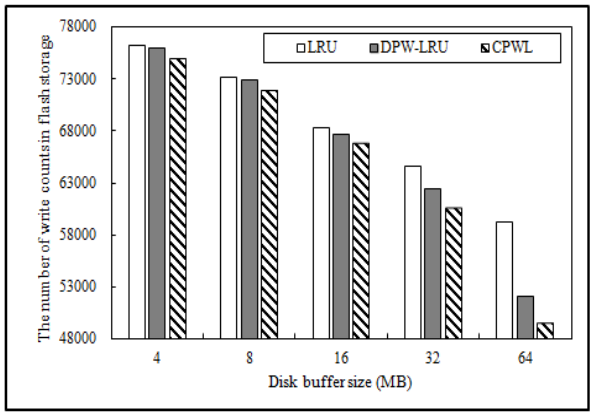
Fig. 5. The hit ratio in disk buffer



(a) ascii



(b) home

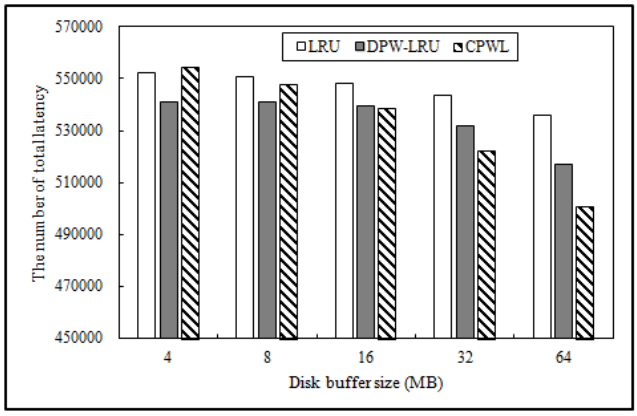


(c) wdev

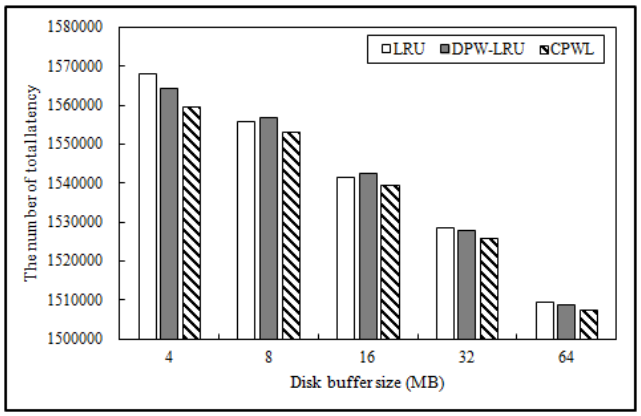
Fig. 6. The number of write counts in flash storage

실험 결과 본 논문에서 제안된 기법의 히트율은 평균 1.2%, 최대 6.0% 증가하였고, 쓰기 연산 횟수는 평균 1.3%, 최대 16.6%가 감소되었다. 지연 시간의 경우 평균 1.1%, 최대 13.7%가 감소됨을 확인하였다. 이렇게 향상된 성능들은 플래시 메모리가 이용되는 장비를 더 효율적으로 이용할 수 있게 돕는 결과를 가져온다. 특히, 저전력을 요구하고 상대적으로 적은 공간의 플래시 메모리를 이용하는 임베디드 장비에서 그 효율이 극대화될 것이다.

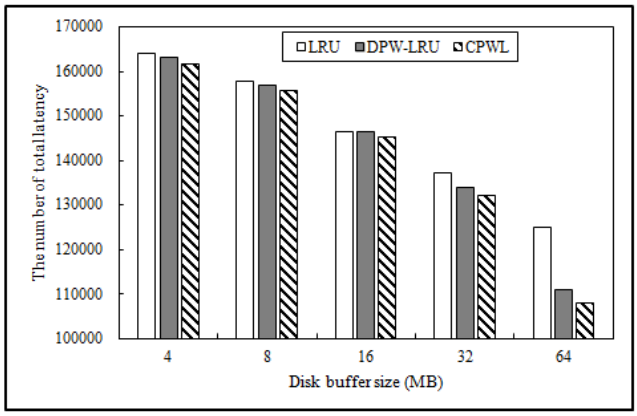
그러나 제안된 CPWL 기법에서 각 영역의 고정된 비율은 다양한 환경에 대한 적응력을 갖추지 않는다. 그러므로 향후에는 두 영역에 대한 비율이 성능에 영향을 미치는 정도를 확인하고, 이를 동적으로 반영할 수 있는 새로운 알고리즘을 연구할 것이다. 이를 통해 플래시 메모리의 수명을 개선하고 에너지 소비를 줄일 수 있게 될 것이다.



(a) ascii



(b) home



(c) wdev

Fig. 7. The number of total latency

ACKNOWLEDGEMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (2017R1D1A1A09000654).

REFERENCES

[1] Lawton, George, “Improved flash memory grows in popularity”, Computer, Vol. 39, No. 1, pp. 16-18, 2006.

[2] M. Lee, J. Kim, and S. Park, "A Recent Trend of Buffer Management based on Database using Next-generation Memory Module," Database Research, KIISE, Vol. 32, No. 1, pp. 31-45, 2016.

[3] K. Zheng, and J. Wang, "Page Weight-Based Buffer Replacement Algorithm for Flash-Based Databases," 2017 International Conference on Computer Technology, Electronics and Communication, pp. 466-470, 2017.

[4] Dong, Xiangyu and Xu, Cong and Xie, Yuan and Jouppi, Norman P, “Nvsim: A circuit-level performance, energy, and area model for emerging nonvolatile memory”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 31, No. 7, pp. 994-1007, 2012.

[5] Saxena, Mohit and Swift, Michael M, “FlashVM: Virtual Memory Management on Flash.”, USENIX Annual Technical Conference, 2010.

[6] C. Kavar, S, and S. Parmar, "Performance Analysis of LRU Page Replacement Algorithm with Reference to Different Data Structure," International Journal of Engineering Research and Applications, Vol. 3, No. 1, pp. 2070-2076, 2013.

[7] S. Park, D. Jung, J. Kang, J. Kim, and J. Lee, "CFLRU: A Replacement Algorithm for Flash Memory," Proccedings of the 2006 international conference on Compilers, architecture and synthesis for embedded systems, pp 234-241, 2006.

[8] G. Xu, F. Lin, and Y. Xiao, "CLRU: A New Page Replacement Algorithm for NAND Flash-based Consumer Electronics," IEEE Transactions on Consumer Electronics, Vol. 60, No. 1, pp. 38-44, 2014.

[9] U. Anwar, J. Paik, R. Jin, and T. Chung, "Log-Buffer Aware Cache Replacement Policy for Flash Storage Devices," IEEE Transactions on Consumer Electronics, Vol. 63, No. 1, pp. 77-84, 2017.

[10] R. Bez, E. Camerlenghi, A. Modelli, and A. Visconti, "Introduction to flash memory," Proceedings of the IEEE, Vol. 91, No. 4, pp. 489-502, 2003.

[11] H. Kim, and S. Lee, "A New Flash Memory Management for Flash Storage System," Proceedings of Computer Software and Applications Conference, pp.284-289, 1999.

[12] H. Jung, H. Shim, S. Park, S. Kang, and J. 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] Y. Yuan, J. Zhang, G. Han, G. Jia, L. Yan and W. Li, "DPW_LRU: An efficient buffer management policy based on dynamic page weight for flash memory in cyber-physical systems," IEEE Access, Vol. 7, pp. 58810-58821, 2019.

Authors

Byung Kook Kang received a B.S. degree in the Department of Computer Engineering from Kyungil University, Daegu, Korea in 2000. He is currently M.S. student in Department of Computer Engineering at Yeungnam University and works as a CEO at MBex Co., Ltd. His curer research interests include embedded systems and non-volatile memory.

Jong Wook Kwak received a B.S. degree in Computer Engineering from Kyungpook National University, Daegu, Korea in 1998, a M.S. degree in Computer Engineering from Seoul National University, Seoul, Korea in 2001, and a Ph.D. degree in Electrical Engineering and Computer Science from Seoul National University, Seoul, Korea in 2006. From 2006 to 2007, he worked as a Senior Engineer in the SoC R&D Center, at Samsung Electronics Co., Ltd. During 2011~2012, he was a Guest Researcher at the Research Institute of Advanced Computer Technology, Seoul National University. During 2012~2013, he was a Visiting Scholar at the Georgia Institute of Technology, Atlanta, GA, USA. As a Head Director, he led DREAM Software Human Resource Training Center from 2014~2015. During 2018~2019, he was a Visiting Scholar at Arizona State University, Tempe, AZ, USA. He is currently a professor in the Department of Computer Engineering, Yeungnam University. His research interests include advanced processor architecture, low-power mobile embedded systems, and high performance parallel computing.