

Buffer Policy based on High-capacity Hybrid Memories for Latency Reduction of Read/Write Operations in High-performance SSD Systems

Sungho Kim*, Sang-Ho Hwang**, Myungsub Lee***, Jong Wook Kwak****, Chang-Hyeon Park****

Abstract

Recently, an SSD with hybrid buffer memories is actively researching to reduce the overall latency in server computing systems. However, existing hybrid buffer policies caused many swapping operations in pages because it did not consider the overall latency such as read/write operations of flash chips in the SSD. This paper proposes the clock with hybrid buffer memories (CLOCK-HBM) for a new hybrid buffer policy in the SSD with server computing systems. The CLOCK-HBM constructs new policies based on unique characteristics in both DRAM buffer and NVMs buffer for reducing the number of swapping operations in the SSD. In experimental results, the CLOCK-HBM reduced the number of swapping operations in the SSD by 43.5% on average, compared with LRU, CLOCK, and CLOCK-DNV.

▶ Keyword: NAND flash memory, Buffer policy, Swapping operation, Hybrid buffer systems, Non-volatile memories

I. Introduction

최근 낸드 플래시 메모리 기반의 SSD는 3D 스택 기술을 적용함으로써 기존 방식이 가진 구조적인 문제를 해결하여 대용량화와 동시에 단위 용량 당 비용이 크게 감소하고 있다[1]. 이러한 장점에도 불구하고 SSD에 적용하는 3D 스택 기술은 하나의 읽기/쓰기 연산에 대한 페이지 크기(page size)를 증가시켰다[2]. 또한 3D 스택 기술은 기존의 2D 기술과 비교하여 읽기/쓰기 연산을 수행하는데 걸리는 긴 지연시간(latency)을 크게 완화하였으나, 여전히 메인 메모리와 비교하여서는 성능 차이가 확연하다. 이러한 이유로 SSD 제조사들은 메인 메모리와 SSD의 성능 차이를 극복하기 위해서, DRAM을 버퍼(buffer)로 채택했다.

그러나 DRAM을 버퍼로 사용하기에는 단위 크기 당 직접도(density)에 대한 한계가 있으며, 이는 점진적으로 대용량화가 되고 있는 SSD의 성능 향상하기에는 부족한 측면이 존재한다.

이러한 문제점을 해결하기 위해, 최근에는 SSD에 하이브리드 메모리 구조 기반의 버퍼(이하 하이브리드 버퍼)를 활용하는 많은 연구가 진행 중에 있다[3]. 하이브리드 버퍼 구조는 DRAM 버퍼와 비휘발성 메모리(NVMs, non-volatile memories) 버퍼의 상호 장점들을 활용하는 정책으로, DRAM 버퍼의 짧은 연산 지연시간의 특성과 비휘발성 메모리 버퍼의 높은 직접도 특성으로 버퍼를 효율적으로 활용할 수 있다. 특히, 비휘발성 메모리 버퍼는 DRAM 버퍼와 유사한 읽기 지연시간을 가지고 있어, 하이브리드 버퍼의 활용성을 극대화할 수 있다.

기존의 대표적인 하이브리드 버퍼 정책은 CLOCK-DNV(CLOCK with DRAM and NVM hybrid write buffer)가 있다[4]. CLOCK-DNV는 하이브리드 버퍼의 특성을 기반으로 기존의 대표적인 DRAM 버퍼 기반의 버퍼 정책들 보다

• First Author: Sungho Kim, Corresponding Author: Chang-Hyeon Park
*Sungho Kim (shk@gitc.or.kr), Software Research Team, Gyeongbuk Institute of IT Convergence Industry Technology (GITC)
**Sang-Ho Hwang (shhwang@dgist.ac.kr), Wellness Convergence Research Center, Daegu Gyeongbuk Institute of Science and Technology (DGIST)
***Myungsub Lee (skydream@ync.ac.kr), Dept. of Computer Information, Yeungnam University College
****Jong Wook Kwak (kwak@yu.ac.kr), Dept. of Computer Engineering, Yeungnam University
****Chang-Hyeon Park (park@yu.ac.kr), Dept. of Computer Engineering, Yeungnam University
• Received: 2019. 04. 29, Revised: 2019. 06. 17, Accepted: 2019. 06. 17.

메인 메모리와 SSD 사이에 읽기/쓰기 연산의 지연시간을 감소시켰다. 그러나 CLOCK-DNV는 하이브리드 버퍼에 공간이 부족할 시 하이브리드 버퍼와 SSD 간의 상호 페이지들을 교체하는 스와핑(swapping) 연산에 대한 비용을 고려하지 않기 때문에, 쓰기 연산이 빈번하게 발생하는 페이지들을 오랜 기간 동안 DRAM 버퍼에 유지하지 않는다. 또한 CLOCK-DNV는 한 번 DRAM 버퍼에서 비휘발성 메모리 버퍼로 이주한 페이지들을 재차 DRAM 버퍼로 이주하는 정책이 존재하지 않기 때문에, DRAM 버퍼와 비휘발성 메모리 버퍼의 특성들을 제대로 활용하지 않고 하이브리드 버퍼 정책을 설계한 측면이 존재한다고 볼 수 있다.

본 논문에서는 하이브리드 버퍼에 적합한 정책을 수립하기 위해 기존의 AC-CLOCK(Adaptive Classification CLOCK)에서의 주요 정책을 하이브리드 버퍼의 특성에 맞게 수정한 정책을 제안하고자 한다[5]. 제안 정책은 기존의 하이브리드 버퍼 정책들과 달리 DRAM 버퍼에 쓰기 연산이 빈번한 페이지들을 상주하도록 유도한다. 또한 제안 정책은 비휘발성 메모리 버퍼에서 빈번하게 쓰기 연산이 발생하는 페이지들을 DRAM 버퍼로 이주하는 정책을 채택하여, 상대적으로 읽기 연산이 빈번한 페이지들을 비휘발성 메모리 버퍼에 유지함으로써 하이브리드 버퍼의 효율성을 향상시킨다. 이러한 특성으로 제안 정책은 하이브리드 버퍼의 스와핑 연산을 최소화하고, SSD의 전체 지연시간을 감소에 기여할 것이다.

이하 본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 활용하는 비휘발성 메모리의 배경지식을 서술하고, 버퍼에 대한 관련 연구를 기술한다. 3장은 본 논문에서 제안하는 하이브리드 버퍼 정책에 대해서 자세하게 언급하고, 4장은 상호 다른 버퍼 정책들과 비교하여 제안하는 정책의 성능 평가를 진행한다. 마지막 5장은 본 논문의 결론을 제시한다.

II. Background and Related Works

1. Background

SSD는 낸드 플래시 메모리 칩(이하 플래시 칩)들을 관리하는 플래시 컨트롤러(flash controller), 운영체제에서의 읽기/쓰기 연산을 수행하기 위한 플래시 전환 계층(FTL, Flash Translation Layer), 메인 메모리와 SSD 사이에 연산 지연시간을 감소시키기 위한 버퍼로 구성하며, 그림 1에서는 SSD의 구조를 보인다.

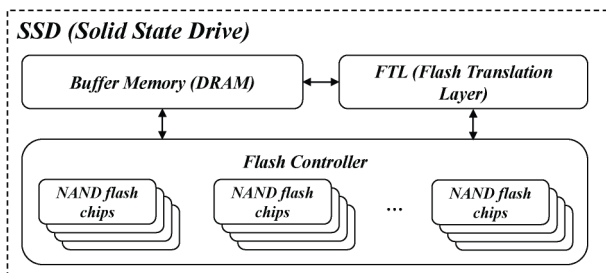


Fig. 1. Architecture of SSD

그림 1의 하단부에 보이는 플래시 컨트롤러는 다수의 플래시 칩들을 관리하기 위해 사용하며, 일반적으로 플래시 칩들을 병렬로 처리하기 위해 다수의 채널(channel)로 구성한다. 각 채널은 한 번에 하나의 플래시 칩으로 접근이 가능하며, SSD는 채널의 증가를 통해 처리량(throughput)을 향상할 수 있다. 하지만 채널의 증가는 많은 비용을 수반하므로, 제조사들은 일반적으로 하나의 채널 당 다수의 플래시 칩들로 구성하는 플래시 컨트롤러를 사용하고 있다[6-7].

플래시 전환 계층은 그림 1의 상단 오른쪽에 위치하고 있으며, 플래시 칩들의 고유의 성질들을 극복하기 위해 사용한다. 일반적으로 플래시 전환 계층은 주소 사상 테이블(address mapping table), 가비지 컬렉션(garbage collection), 마모도 평준화(wear leveling)로 구성한다[8-9]. 주소 사상 테이블은 플래시 칩에 제자리 덮어쓰기 문제를 해결하기 위해, 운영체제에서 요청한 논리 주소와 플래시 칩의 물리 주소를 상호 사상한다[10]. 가비지 컬렉션은 주소 사상 과정에서 과거에 사용한 주소 페이지들을 집약하여 새로운 여유 공간을 확보하는 작업을 수행한다[11]. 마지막으로 마모도 평준화는 가비지 컬렉션 과정에서 블록들을 고르게 선정할 수 있도록 유도한다[12].

그림 1의 상단 왼쪽의 보이는 버퍼는 SSD의 성능 향상을 위해 크게 세 가지의 상호 다른 정책을 사용할 수 있다. 첫 번째 정책은 플래시 칩들에게 읽기/쓰기 연산이 적게 발생할 수 있도록 버퍼에서 빈번하게 참조하는 페이지들을 상주하도록 유도한다. 두 번째 정책은 읽기 연산을 제외하고 쓰기 연산 전용의 버퍼를 사용하여 SSD를 구성하는 플래시 칩들의 수명을 연장한다. 세 번째 정책은 가비지 컬렉션의 여유 공간 확보 비용을 줄이기 위해, 동일한 블록에 해당하는 페이지들을 버퍼에 유지하여 SSD의 가비지 컬렉션 비용을 감소시킨다.

최근 SSD는 DRAM과 비휘발성 메모리를 결합한 하이브리드 버퍼 구조에 대한 연구가 활발히 진행 중이다[13]. 이하 본 논문에서의 비휘발성 메모리는 플래시 칩을 제외한 용어로 사용한다. 표 1은 DRAM, 비휘발성 메모리, 플래시 칩의 특성들을 보인다. 표 1에서 비휘발성 메모리의 직접도는 DRAM 보다는 높고 플래시 칩보다는 적으며, 비휘발성 메모리의 읽기/쓰기 연산 속도에서는 DRAM과 유사하거나 조금 느리다. 이러한 비휘발성 메모리의 특성은 SSD 버퍼에서 DRAM의 단점을 보완하기에 적합하여 하이브리드 버퍼 구조가 고안되었다[9, 14].

Table 1. Characteristics of memories

description	DRAM	NVMs	flash chip
read latency	50ns	50ns	100us
write latency	50ns	350ns	2.4ms
erase latency	-	-	3.0ms
density	very low	low	high
endurance	-	10 ⁸	10 ³

2. Related Works

가장 대표적인 버퍼 정책들은 LRU(Least Recently Used)와 CLOCK이 있다[15]. LRU는 버퍼에 상주하는 페이지들 중 사용한 가장 오래된 페이지들을 SSD로 쫓아낸다. 이 정책은 페이지들의 사용 빈도에 대한 정보를 유지하기 위한 리스트를 활용하며, 리스트에 사용된지 오래된 페이지들을 유지 및 관리하기 위해서 많은 시간이 소요하므로 캐시나 적은 크기의 버퍼에 주로 사용한다. CLOCK은 페이지들의 접근한 정보를 알리는 참조 비트(reference bit)를 활용하여, 시계 방향으로 회전하면서 참조 비트가 설정되어 있지 않은 페이지들을 SSD로 쫓아낸다. CLOCK은 LRU와 비교하여 연산의 적중률(hit ratio)이 유사하거나 조금 낮으나, LRU 페이지를 유지 및 관리하는 오버헤드가 없기 때문에 메인 메모리나 대용량 버퍼에서 주로 활용한다.

AC-CLOCK은 하이브리드 메모리 구조를 메인 메모리에 적용한 정책이다[5]. AC-CLOCK은 DRAM과 비휘발성 메모리를 효율적으로 활용하기 위해 각기 메모리의 특성에 적합한 CLOCK 정책을 설계했다. DRAM 정책은 메인 메모리의 특성에 적합하게 쓰기/읽기 연산의 패턴을 추적하고, 이를 기반으로 쓰기 연산이 빈번한 페이지들을 DRAM에 상주시키고 읽기 연산이 빈번한 페이지들을 비휘발성 메모리로 이주했다. 비휘발성 메모리 정책은 연산 패턴을 추적하여 DRAM으로 이주했으며, 그렇지 않은 페이지들은 저장소(storage)로 쫓아냈다. 그러나 AC-CLOCK은 하이브리드 메인 메모리의 적합하도록 설계되었기 때문에, 하이브리드 버퍼에서 발생하는 스와핑 연산에 대한 고려가 없다.

BPLRU(Block Padding Least Recently Used)는 블록 단위의 쓰기 버퍼를 이용하여 가비지 컬렉션 비용을 줄이는 정책이다[16]. BPLRU는 버퍼에 가장 많은 페이지들을 상주시킨 블록들을 우선적으로 쫓아낸다. 이때 BPLRU는 쫓아내는 블록 중 버퍼 내에 존재하지 않는 페이지들을 SSD에서 읽기 연산을 통해 완전히 새로운 블록을 만들며, 이를 블록 패딩(block padding)이라 부른다. 하지만 BPLRU는 가비지 컬렉션에 대한 비용을 크게 감소시켰지만, 블록 패딩으로 인하여 추가적인 SSD의 읽기 연산에 대한 오버헤드가 발생한다. 이는 SSD 읽기 연산이 긴 지연시간을 동반하기 때문에 SSD 전반의 지연시간을 증가할 위험성이 있다.

TS-CLOCK(Temporal and Spatial locality aware CLOCK)은 각 페이지에 2비트 참조 카운트(reference counter)와 더티 비트(dirty bit)를 이용하여, 참조 빈도가 낮은 페이지들을 쫓아낸다[17]. 참조 카운터는 읽기/쓰기 연산의 횟수를 측정하고, 더티 비트는 쓰기 연산에 대한 발생 여부의 기록이다. TS-CLOCK은 버퍼에 공간이 부족할 경우 참조 카운트가 0이면서 더티 비트가 기록되어 있지 않은 페이지들을 SSD로 쫓아낸다. 만약 이를 만족하지 못하면, TS-CLOCK은 더티 비트가 기록된 페이지들을 순차로 검색하여 참조 카운트가 0인 페이지들을 SSD로 쫓아낸다. 하지만 TS-CLOCK은 DRAM 기반의 정책이기 때문에 하이브리드 버퍼에 특성들을 고려하지 않은 단점이 있다.

CLOCK-DNV는 참조 비트와 더티 비트로 모든 페이지들을 구분하고, 하이브리드 버퍼의 특성을 고려하여 DRAM 버퍼와

비휘발성 메모리 버퍼를 각기 다른 CLOCK으로 설계했다[4]. CLOCK-DNV에서 DRAM 버퍼 정책은 스와핑 연산이 발생하고 DRAM 버퍼에 여유 공간이 부족할 경우 더티 비트가 설정된 페이지들을 비휘발성 메모리 버퍼로 이주한 후, 새로운 페이지를 할당한다. 또한 CLOCK-DNV는 기존의 BPLRU에서 블록 패딩 정책을 비휘발성 메모리 버퍼 정책에 적용하되, 하이브리드 버퍼에 한해서 수행했다. 하지만 CLOCK-DNV는 DRAM 버퍼의 여유 공간을 확보하는 과정에서 더티 비트가 설정되지 않은 페이지들을 폐기하기 때문에, 그로 인해 많은 스와핑 연산이 발생하는 단점이 있다.

III. The Proposed Policy

본 논문에서 제안하는 정책인 CLOCK with Hybrid Buffer Memories(CLOCK-HBM)는 기존의 메인 메모리 기반인 AC-CLOCK의 주요 정책들을 차용하여 설계한다. AC-CLOCK은 메인 메모리의 특성을 기초로 설계하였으므로, 하이브리드 버퍼에 그대로 적용하기 힘들다. 다음은 하이브리드 버퍼에서의 주요 특성들이다.

1. 하이브리드 버퍼에서 DRAM 버퍼는 비휘발성 메모리 버퍼와 비교하여 쓰기 지연시간 짧다. 이는 읽기/쓰기 연산이 빈번한 랜덤적인 특성을 가지는 페이지들을 DRAM 버퍼에 상주시키는 것이 유리하다.
2. 하이브리드 버퍼는 순차 연산에 대한 고려가 필요하다. 순차 연산은 일반적으로 읽기 연산을 중심으로 이루어진다. 즉, 비휘발성 메모리 버퍼는 DRAM 버퍼보다 상대적으로 큰 용량을 가지고 있기에 순차적인 특성을 가지는 페이지들을 상주시키는데 큰 장점을 가지고 있다.
3. 하이브리드 버퍼와 SSD 사이에 페이지들의 스와핑 연산은 많은 비용을 요구한다. 표 1에서 플래시 칩은 다른 메모리들과 비교하여 긴 연산 지연시간의 특성을 가지고 있음을 확인할 수 있다. 이는 플래시 칩에 연산을 최소한으로 수행할 필요성을 의미한다.

앞서 제시한 하이브리드 버퍼의 세 가지 특성들을 고려하여 제안하는 CLOCK-HBM은 다음의 세 가지 해결책을 제시한다.

1. CLOCK-HBM은 랜덤 특성을 가지는 페이지들을 구분하기 위해 스와핑 연산이 발생한 페이지들을 오직 DRAM 버퍼에만 할당한다. 이는 랜덤 특성을 가지는 페이지들을 적은 비용으로 구분할 수 있기 때문이다.
2. CLOCK-HBM은 DRAM 버퍼에 여유 공간이 부족한 경우 쓰기 연산이 빈번하지 않은 페이지들을 비휘발성 메모리 버퍼로 이주한다. 이는 비휘발성 메모리 버퍼가 DRAM 버퍼와 유사

한 읽기 지연시간을 가지고 있어 읽기 연산에 대한 이익을 얻을 수 있고, 또한 DRAM 버퍼에 쓰기 연산으로 얻는 이득을 극대화할 수 있기 때문이다.

3. CLOCK-HBM은 하이브리드 버퍼에 읽기/쓰기 연산이 빈번하게 발생하는 페이지들을 상주시키기 위해 비휘발성 메모리 버퍼의 페이지들만 SSD로 쫓아낸다. 이는 메인 메모리와 달리 버퍼가 스와핑 연산에 대한 비용이 크기 때문이다.

Algorithm 1: CLOCK with Hybrid Buffer Memories

```

input:  $p$ , // page according to a requested operation
          $op$ , // read or write operation
          $d-clock$ , // current clock pointer in DRAM buffer
          $n-clock$ , // current clock pointer in NVMs buffer
          $fs$  // space of a free page

output:  $d-clock$ ,  $n-clock$ 

1 if there is no  $p$  in hybrid buffer do
2   if there is no free space in DRAM buffer then
3      $fs = \text{MigrateToNVMSBuffer}(d-clock, n-clock, p)$ ;
4   end
5    $\text{AllcateToDRAMBuffer}(fs, p)$ ;
6 end
7  $\text{SetPageBitFlags}(p, op)$ ;
8 if there is  $p$  in NVMs buffer and reference bit is set do
9   if  $op$  is the write operation then
10     $\text{MigrateToDRAMBuffer}(d-clock, n-clock, p)$ ;
11  end
12 end
    
```

위에서 언급한 하이브리드 버퍼의 세 가지 특성들과 해결책들을 기반으로 CLOCK-HBM은 페이지들의 레벨에 따라 DRAM 버퍼에 상주시키고 비휘발성 메모리 버퍼로 이주한다. 그림 2는 CLOCK-HBM의 구조이다.

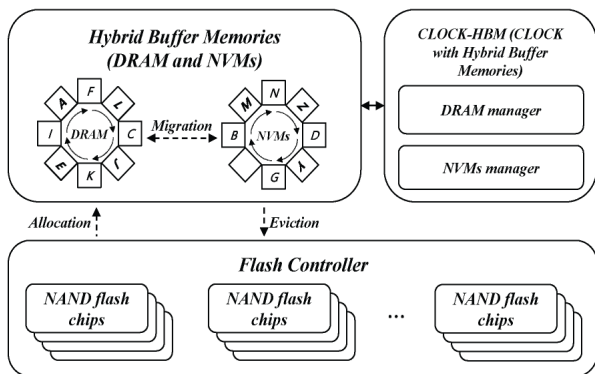


Fig. 2. Architecture of CLOCK-HBM

제안하는 정책은 하이브리드 버퍼를 효율적으로 관리하기 위해, DRAM 버퍼와 비휘발성 메모리 버퍼에 각기 다른 CLOCK 정책으로 관리한다. 또한 CLOCK-HBM은 DRAM 버퍼와 비휘발성 메모리 버퍼의 상호 다른 특성을 고려하여 정책을 수립한다.

알고리즘 1은 CLOCK-HBM에서 DRAM 버퍼와 비휘발성 메모리 버퍼의 정책이다.

CLOCK-HBM은 SSD에 읽기/쓰기 연산이 발생했을 경우 하이브리드 버퍼에 페이지(p)가 존재하는지 확인한다. 만약 하이브리드 버퍼에 페이지가 존재하지 않은 경우, 제안하는 정책은 DRAM 버퍼에 여유 공간이 있는지 확인한다. 만약 여유 공간이 존재하지 않는다면, CLOCK-HBM은 DRAM 버퍼에 존재하는 페이지들 중 하나의 페이지를 선정하여 비휘발성 메모리 버퍼로 이주하고, 해당하는 공간에 새로운 페이지를 할당한다. 이후 CLOCK-HBM은 해당 페이지가 비휘발성 메모리 버퍼 안에 존재하고 참조 비트가 설정되어 있으며, 그에 해당하는 페이지가 쓰기 연산인 경우 DRAM 버퍼로 이주한다. 이러한 일련의 과정과 페이지들이 할당되고 레벨을 구분하며 DRAM 버퍼와 비휘발성 메모리 버퍼 상호 이주하는 자세한 과정은 그림 3과 표 2를 통해서 상세하게 설명한다.

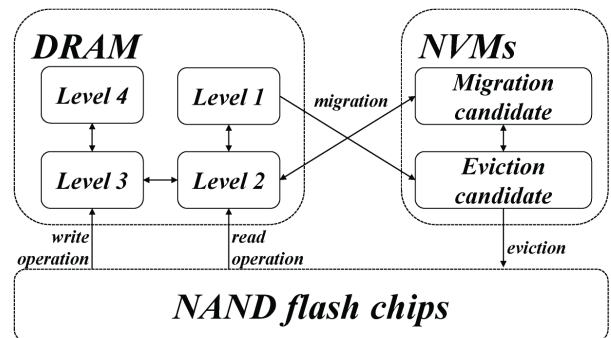


Fig. 3. The detailed diagram for the state of pages in CLOCK-HBM

Table 2. Switching conditions for the number of write operations of each level in DRAM buffer page

Level	Increment	Maintenance	Decrease
1	1-3	0	-
2	2 or 3	1	0
3	3	1 or 2	0
4	-	2	0 or 1

그림 3은 페이지들의 레벨을 구분하는 일련의 과정을 나타낸 것이다. 스와핑 연산으로 할당되는 페이지들은 읽기/쓰기 연산에 따라 DRAM 버퍼에 차등 할당한다. 이는 쓰기 연산을 읽기 연산보다 오래 DRAM 버퍼에 상주시키기 위함이다. CLOCK-HBM은 DRAM 버퍼에 상주시키는 페이지들을 네 단계 레벨로 구분하며, 쓰기 확률이 높은 페이지들을 높은 단계로 분류한다. 단계의 변화는 CLOCK이 한 사이클이 회전하는 동안에 쓰기 연산의 횟수에 따라 평가되며, 자세한 평가 조건은 표 2와 같다.

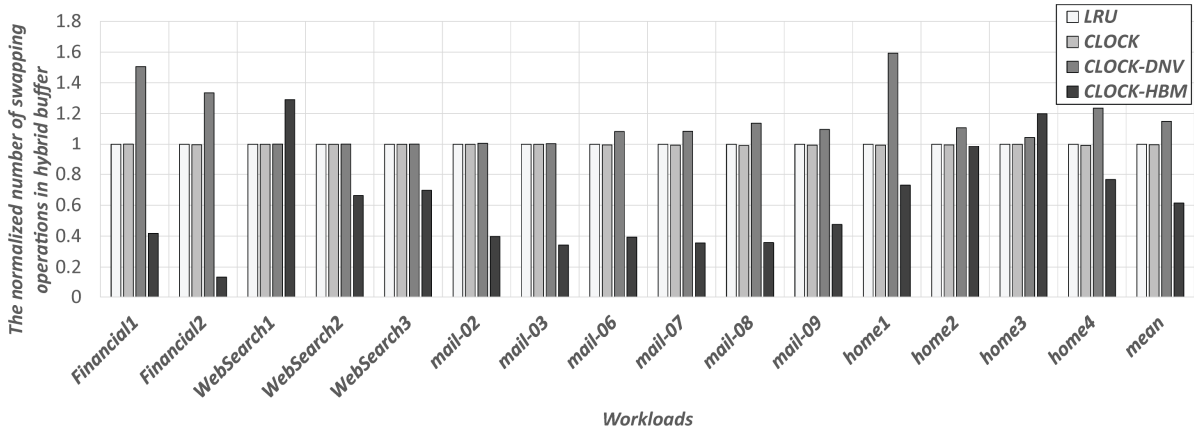


Fig. 4. The normalized number of swapping operations in hybrid buffer

표 2에서 CLOCK-HBM은 각 레벨에 해당하는 쓰기 연산의 횟수를 차등 조건으로 적용했다. 이는 낮은 레벨이 순간 작업부하의 변동성으로 쓰기 연산이 적게 발생함을 고려한 것이다. 즉 제안하는 정책은 과거에 빈번히 쓰기 연산을 수행한 페이지들을 선호한다. 이러한 과정을 거쳐 DRAM 버퍼 정책은 페이지들의 레벨을 구분하며, DRAM 버퍼에 여유 공간이 부족할 시 알고리즘 1과 그림 3과 같이 CLOCK을 시계 방향으로 회전하면서 레벨 1과 레벨 2에 해당하는 페이지들을 비휘발성 메모리로 이주함으로써 여유 공간을 확보한다.

마지막으로 CLOCK-HBM의 비휘발성 메모리 버퍼 정책은 DRAM 버퍼에서 이주한 페이지들 중 레벨 1과 레벨 2에 해당하는 페이지들을 할당한다. 여기서 레벨 1과 레벨 2의 차이점은 비휘발성 메모리 버퍼에서 차등 분류된다는 것이다. 이는 비휘발성 메모리 버퍼에 지속적으로 상주하길 원하는 페이지인 이주 후보자(migration candidate)와 조기에 쫓아내고 싶은 페이지인 퇴출 후보자(eviction candidate)를 분류하여 할당하기 위함이다. 이러한 페이지들은 DRAM 버퍼 정책과 마찬가지로 할당된 이후 CLOCK이 한 사이클이 회전하는 동안에 읽기 연산이 발생한 여부에 따라 재분류한다. 최종적으로 CLOCK-HBM은 하이브리드 버퍼에서 읽기/쓰기 연산에 대한 접근 빈도가 낮은 페이지들을 스와핑 연산의 대상으로 선정한다.

IV. Performance Evaluation

Table 3. Parameters for SSD model in the simulator

description		parameter
buffer size	DRAM	6.4MB (20%)
	NVMs	25.6MB (80%)
SSD capacity		20GB
page size		4KB
pages per block		64
garbage collection trigger		# of free blocks < 20%

이 장에서는 제안하는 CLOCK-HBM의 성능 평가를 진행하기 위해 disksim 4.0 기반의 하이브리드 버퍼 시뮬레이션을 구축했다[18-19]. 표 3은 하이브리드 버퍼 시뮬레이션에서 사용한 자세한 매개변수들이다.

시뮬레이션의 신뢰성을 보장하기 위해, 본 논문에서는 UMass와 FIU(Florida International University) 작업부하를 사용했다[20-21].

표 4는 시뮬레이션에서 사용한 작업부하들의 요약이다. UMass는 OLTP(Online Transaction Processing)에서 수행하는 앱의 특성을 가지고 있으며, FIU는 단말 사용자, 웹 서비스 등의 동작 특성을 가지고 있다. CLOCK-HBM의 성능 평가는 LRU, CLOCK, CLOCK-DNV 정책들을 선정했으며, 성능 평가 지표는 하이브리드 버퍼에서 발생하는 스와핑 연산의 횟수와 읽기/쓰기 연산의 횟수로 비교 분석을 진행했다.

Table 4. Detailed characteristics of workloads

types	characteristics	ratio of request (%)		avg. request (KB)
		read	write	
UMass	Financial1	23.0	76.0	3.28
	Financial2	83.0	17.0	2.39
	WearSearch1	0	100	15.14
	WearSearch2	0	100	15.07
	WearSearch3	0	100	15.4
FIU	mail-02	17.0	83.0	39.8
	mail-03	13.0	87.0	41.53
	mail-06	10.0	90.0	29.97
	mail-07	8.0	92.0	27.96
	mail-08	16.0	84.0	44.73
	mail-09	13.0	87.0	29.61
	home1	2.0	98.0	6.3
	home2	13.0	87.0	12.77
	home3	3.0	96.0	23.75
	home4	0	100	12.58

그림 4는 하이브리드 버퍼에서 스와핑 연산의 횟수에 대한 결과이다. 스와핑 연산은 SSD에서 페이지 읽기 연산을 수행해야 하기 때문에 긴 지연시간이 요구한다. 따라서 SSD 전반의 성

능 향상을 위해, 하이브리드 버퍼 정책은 스와핑 연산의 횟수를 줄여야한다. 제안하는 정책인 CLOCK-HBM은 기존의 다른 정책들과 비교하여 평균적으로 43.5%의 스와핑 연산의 횟수를 줄였다. 이는 쓰기 연산이 빈번한 페이지들을 DRAM 버퍼로, 반대로 읽기 연산이 빈번한 페이지들을 비휘발성 메모리 버퍼로 상주시켜 이러한 결과가 보였을 것이라고 추측된다. 결과적으로 CLOCK-HBM은 기존의 CLOCK, LRU, CLOCK-DNV와 비교하여 하이브리드 버퍼에 많은 읽기/쓰기 연산들이 발생해야 할 것이다. 다음은 하이브리드 버퍼에서 읽기/쓰기 연산의 횟수를 파악하여, CLOCK-HBM에서 스와핑 연산을 감소시킨 근거를 찾기 위한 결과들이다.

그림 5는 하이브리드 버퍼에서 DRAM 버퍼와 비휘발성 메모리 버퍼에 쓰기 연산이 발생한 횟수에 대한 결과이다. 쓰기 연산에 대한 결과는 작업부하에 특성에 따라 UMass와 FIU를 분리했다. CLOCK-HBM은 기존의 다른 정책들과 비교하여 UMass 작업부하는 평균적으로 49.96%의 쓰기 연산 횟수가 증가했고, FIU 작업부하는 평균적으로 70.54%의 쓰기 연산 횟수가 증가했다. 제안하는 정책은 UMass와 FIU에 해당하는 모든 작업부하에서 우수한 성능을 보였다.

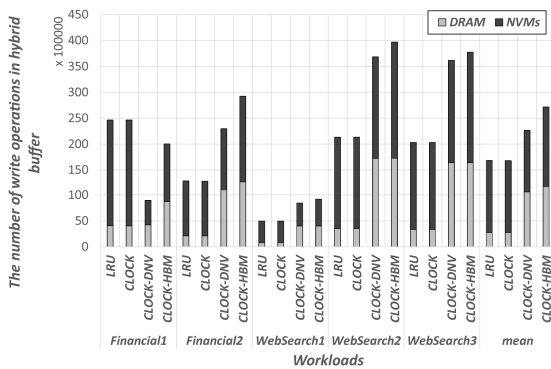
마지막으로 그림 6은 하이브리드 버퍼에서 읽기 연산이 발생한 횟수에 대한 결과를 보여주고 있다. 쓰기 연산에 대한 결

과와 마찬가지로 읽기 연산도 작업부하의 특성에 따라 분리했다. CLOCK-HBM은 기존의 다른 정책들과 비교하여 UMass 작업부하는 평균적으로 106.9%의 읽기 연산 횟수가 증가했고, FIU 작업부하는 평균적으로 911.58%의 읽기 연산 횟수가 증가했다. 이는 특정한 작업부하들로 인해 성능이 상대적으로 증가한 것을 확인할 수 있다. 하지만 이러한 특정한 작업부하인 mail-02와 mail-03을 제외하고도 상대적으로 많은 읽기 연산이 발생한 것을 확인할 수 있다.

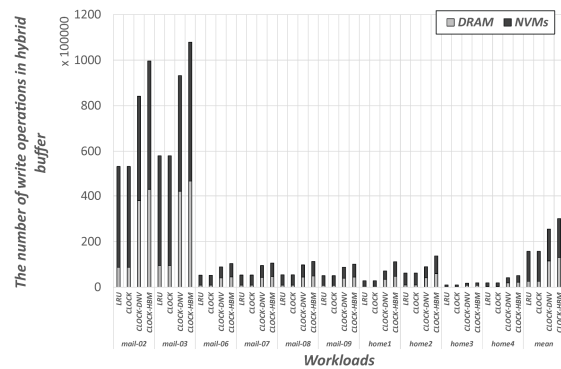
결과적으로 CLOCK-HBM은 기존의 정책들과 비교하여 읽기와 쓰기 연산이 빈번하게 발생하는 페이지들을 하이브리드 버퍼에 상주시켜, 스와핑 연산의 횟수를 감소 시켰다. 또한 제안하는 정책은 랜덤/순차 작업부하와 상관없이 우수한 성능을 실험 결과를 통해 입증했다.

V. Conclusions

본 논문에서는 고성능 SSD 시스템에서 하이브리드 버퍼 기반의 연산 지연시간 감소를 위한 새로운 버퍼 정책인 CLOCK-HBM을 제안했다. 제안하는 정책은 SSD의 연산 지연

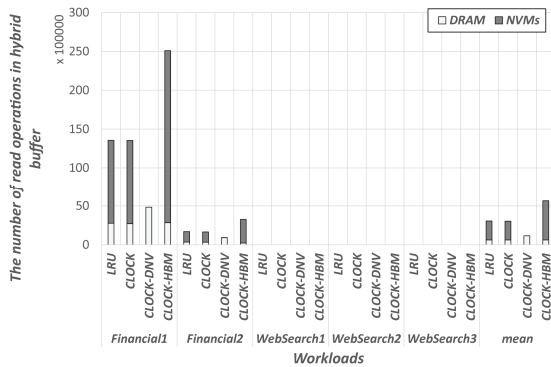


(a) UMass

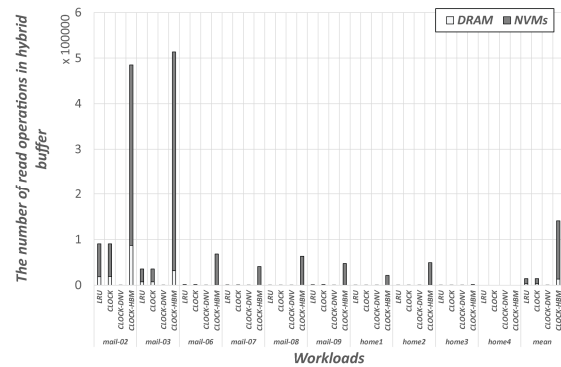


(b) FIU

Fig. 5. The number of write operations in hybrid buffer



(a) UMass



(b) FIU

Fig. 6. The number of read operations in hybrid buffer

시간을 감소시키기 위해 DRAM 버퍼와 비휘발성 메모리 버퍼 간 상호 특성에 적합한 정책을 고안했다. 이를 통해 CLOCK-HBM은 하이브리드 버퍼에서 쓰기 연산이 빈번한 페이지들을 DRAM 버퍼로 읽기 연산이 빈번한 페이지들을 비휘발성 메모리 버퍼로 상호 상주시킬 수 있도록 유도했다. 성능 평가를 통해, CLOCK-HBM은 기존의 LRU, CLOCK, CLOCK-DVN와 비교하여 평균적으로 43.5%의 스와핑 연산 횟수를 줄였다. 이는 CLOCK-HBM가 다른 정책들과 비교하여 하이브리드 버퍼에 쓰기 연산을 평균적으로 60.25%, 읽기 연산을 평균적으로 509.24%의 증가를 통해 하이브리드 버퍼의 스와핑 연산의 횟수를 감소시킬 수 있었다.

REFERENCES

- [1] Samsung. V, and SSD. N, "860 EVO," 2018.
- [2] C. Kim, D. H. Kim, W. Jeong, H. J. Kim, I. H. Park, H. W. Park, and S. B. Kim, "A 512-Gb 3-b/Cell 64-Stacked WL 3-D-NAND Flash Memory," *IEEE Journal of Solid-State Circuits*, Vol. 53, No. 1, pp. 124-133, Aug, 2018.
- [3] C. Matsui, C. Sun, and K. Takeuchi, "Design of hybrid SSDs with storage class memory and NAND flash memory," *Proceedings of the IEEE*, Vol. 105, No. 9, pp. 1812-1821, Sep, 2017.
- [4] D. H. Kang, S. J. Han, Y. C. Kim, and Y. I. Eom, "CLOCK-DNV: a write buffer algorithm for flash storage devices of consumer electronics," *IEEE Transactions on Consumer Electronics*, Vol. 63, No. 1, pp. 85-91, Feb, 2017.
- [5] S. Kim, S. H. Hwang, and J. W. Kwak, "Adaptive-Classification CLOCK: Page replacement policy based on read/write access pattern for hybrid DRAM and PCM main memory," *Microprocessors and Microsystems*, Vol. 57, pp. 65-75. Mar, 2018.
- [6] Micron, M. L. C. "SSD Specification," 2013.
- [7] T. Coughlin, "A Timeline for Flash Memory History [The Art of Storage]," *IEEE Consumer Electronics Magazine*, Vol. 6, No. 1, pp. 126-133, Jan, 2017.
- [8] T. S. Chung, D. J. Park, S. Park, D. H. Lee, S. W. Lee, and H. J. Song, "A survey of flash translation layer," *Journal of Systems Architecture*, Vol. 55, No. 5-6, pp. 332-343, May-Jun, 2009.
- [9] F. Chen, T. Zhang, and X. Zhang, "Software Support Inside and Outside Solid-State Devices for High Performance and High Efficiency," *Proceedings of the IEEE*, Vol. 105, No. 9, pp. 1650-1665. Sep, 2017.
- [10] D. Ma, J. Feng, and G. Li, "A survey of address translation technologies for flash memories," *ACM Computing Surveys (CSUR)*, Vol. 46, No. 3, pp. 36-75, 2014.
- [11] M. C. Yang, Y. M. Chang, C. W. Tsao, P. C. Huang, Y. H. Chang, and T. W. Kuo, "Garbage collection and wear leveling for flash memory: Past and future." In *Smart Computing (SMARTCOMP)*, 2014 International Conference on, IEEE, pp. 66-73, Nov, 2014.
- [12] S. H. Kim, and J. W. Kwak, "EPET-WL: Enhanced Prediction and Elapsed Time-based Wear Leveling Technique for NAND Flash Memory in Portable Devices," *Journal of the Korea Society of Computer and Information*, Vol. 21, No. 5, pp. 1-10, May, 2016. (in Korean)
- [13] J. J. Yang, D. B. Strukov, and D. R. Stewart, "Memristive devices for computing," *Nature nanotechnology*, Vol. 8, No. 1, pp. 13-24, Dec, 2013.
- [14] S. H. Hwang, J. H. Choi, and J. W. Kwak, "Adaptive Writeback-aware Cache Management Policy for Lifetime Extension of Non-volatile Memory," *JOURNAL OF SEMICONDUCTOR TECHNOLOGY AND SCIENCE*, Vol 17, No. 4, pp. 514-523, Aug, 2017.
- [15] S. Jiang, F. Chen, and X. Zhang, "CLOCK-Pro: An Effective Improvement of the CLOCK Replacement," In *USENIX Annual Technical Conference, General Track*, pp. 323-336, Apr, 2005.
- [16] H. Kim, and S. Ahn, "BPLRU: A Buffer Management Scheme for Improving Random Writes in Flash Storage," In *FAST*, Vol. 8, No. 16, pp. 1-14, Feb, 2008.
- [17] D. H. Kang, C. Min, and Y. I. Eom, "TS-CLOCK: temporal and spatial locality aware buffer replacement algorithm for NAND flash storages," In *ACM SIGMETRICS Performance Evaluation Review*, Vol. 42, No. 1, pp. 581-582, Jun, 2014.
- [18] G. Ganger, B. Worthington, and Y. Patt, *The DiskSim simulation environment (v4. 0)* [Online], Available: <http://www.pdl.cmu.edu/DiskSim/Online-document>, Parallel Data Lab, 2009.
- [19] V. Prabhakaran, and T. Wobber, "SSD extension for DiskSim simulation environment," *Microsoft Research*, 2009.
- [20] *OLTP and Websearch Traces form UMass Trace Repository* [Online]. Available: <http://traces.cs.umass.edu> (downloaded 2013, Jan. 16).
- [21] R. Koller, and R. Rangaswami, "I/O deduplication: Utilizing content similarity to improve I/O performance," *ACM Transactions on Storage (TOS)*, Vol. 6, No. 3, pp. 13-39. Sep, 2010.

Authors



Sungho Kim received a B.S. degree in the Department of Computer Engineering from Yeungnam University College, Daegu, Korea in 2012 and a Ph.D. degree in Department of Computer Engineering from Yeungnam University, Gyeongsan, Gyeongsangbuk, Korea in 2019

He is currently as a senior researcher in the software research team at Gyeongbuk Institute of IT Convergence Industry Technology (GITC). His current research interests include a big data, a deep learning, embedded systems and non-volatile memory systems.



Sang-Ho Hwang received the B.S., M.S. and Ph.D degrees in Computer Engineering from Yeungnam University, Korea, in 2009, 2013 and 2017 respectively. He is currently a researcher in the Wellness Convergence Research Center, Daegu Gyeongbuk

Institute of Science and Technology (DGIST). His current research interests include non-volatile memory systems and machine learning.



Myungsub Lee received his Ph.D. degree in Computer Science from Yeungnam University, Gyungsan, Korea, in 2003. He also worked as a full-time instructor from 2002 to 2003 at Gyungdong College, Gyungsan, Korea. He was a Research

Professor at Yeungnam University from 2004 to 2007. Subsequently, he was a Senior Researcher with the Regional Innovation Center at Yeungnam University. Since 2008, he has been Professor at the Yeungnam University College of Science and Technology. His research interests include wireless and sensor networks, location-based systems.



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. He is currently a professor in the Department of Computer Engineering, Yeungnam University. His research interests include advanced processor architecture, low-power mobile embedded system, and high performance parallel computing.



Chang-Hyeon Park received the B.S. degree in Electronics Engineering from Kyungpook University, Korea, in 1986 and M.S. and Ph.D degrees in Computer Science from Seoul University, Korea, in 1988 and 1992, respectively.

Dr. Park joined the faculty of the Department of Computer Engineering at Yeungnam University, Gyeongsan, Korea, in 1993. He is currently a Professor in the Department of Computer Engineering at Yeungnam University. He is interested in artificial intelligence, data mining, and embedded system.