

# TLC 낸드 플래시 기반 스토리지를 위한 효율적인 SLC 버퍼 관리 기법

## (An Efficient SLC-buffer Management Scheme for TLC NAND Flash-based Storage)

권 기 록 <sup>†</sup>                      강 동 현 <sup>††</sup>                      엄 영 익 <sup>†††</sup>  
(Kirock Kwon)                      (Dong Hyun Kang)                      (Young Ik Eom)

**요 약** 최근, 스토리지의 가격 경쟁력 향상을 위해 TLC 낸드의 수요가 크게 증가하고 있다. TLC의 SLC/MLC 대비 낮은 성능의 한계를 극복하기 위하여, SLC 쓰기 버퍼를 채용한 TLC 낸드 기반의 스토리지가 제조사들에 의해 상용화되고 있다. 본 논문에서는 SLC 쓰기 버퍼를 채용하는 TLC 낸드 기반의 스토리지에서 파일시스템의 저널링 특성을 활용하여 성능을 향상하는 기법을 제안한다. 제안 기법은 실제 스토리지 플랫폼인 OpenSSD에 구현되어 기존 기술의 성능과 비교하였으며, 기존 기법 대비 최대 65%의 스토리지 처리량 성능 향상을 확인하였다.

**키워드:** 파일시스템 저널링, 낸드 플래시 스토리지, SLC-TLC 혼합 구조, SLC 버퍼, TLC 낸드, FTL

**Abstract** In recent years, almost all consumer devices have adopted NAND flash storage as their main storage, and their performance and capacity requirements are getting higher. To meet these requirements, many researchers have focused on combined SLC-TLC storage consisting of high-speed SLC and high-density TLC. In this paper, we redesign the internal structure of the combined SLC-TLC storage to efficiently manage the SLC region inside the storage and propose a scheme that improves the performance of the storage by employing the I/O characteristics of file system journaling. We implemented our scheme on the real storage platform, the OpenSSD jasmine board, and compared it with the conventional techniques. Our evaluation results show that our technique improves the storage performance by up to 65%, compared with the conventional techniques.

**Keywords:** file system journaling, NAND flash storage, combined SLC-TLC storage, SLC buffer, TLC NAND, FTL

- 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 SW컴퓨팅산업원천기술개발사업(SW스타랩)의 연구결과로 수행되었음(IITP-2015-0-00284)
- 이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 한국연구재단-차세대정보컴퓨팅기술개발사업의 지원을 받아 수행된 연구임(No. NRF-2015-M3C4A7065696)
- 이 논문은 2017 한국소프트웨어종합학술대회에서 '파일시스템 저널링의 입출력 특성을 이용한 TLC 낸드 기반 스토리지의 성능 개선 기법'의 제목으로 발표된 논문을 확장한 것임

<sup>†</sup> 학생회원 : 성균관대학교 반도체디스플레이공학과  
krkwon@skku.edu

<sup>††</sup> 학생회원 : 성균관대학교 전자전기컴퓨터공학과  
kkangsu@skku.edu

<sup>†††</sup> 종신회원 : 성균관대학교 소프트웨어대학 교수(Sungkyunkwan Univ.)  
yieom@skku.edu  
(Corresponding author임)

논문접수 : 2018년 2월 23일  
(Received 23 February 2018)  
논문수정 : 2018년 4월 3일  
(Revised 3 April 2018)  
심사완료 : 2018년 4월 4일  
(Accepted 4 April 2018)

Copyright©2018 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.  
정보과학회논문지 제45권 제7호(2018. 7)

## 1. 서론

1980년대 플래시 메모리가 발명된 이후 낸드 플래시 기술은 크게 발전하였다. 10nm까지의 공정 기술의 발전은 물론, 최근 3차원 구조(3D) 낸드의 개발에 따른 집적도 향상은 가격 경쟁력의 향상으로 이어졌다[1]. 이러한 플래시 메모리 기술의 발전에 따른 스토리지 용량의 증가와 가격 경쟁력의 향상을 발판으로 거의 모든 모바일 디바이스가 낸드 플래시 기반의 스토리지를 주 저장 장치로 채용하고 있으며, PC뿐만 아니라 대규모 저장 시스템에서도 Hard Disk Drive (HDD)에서 Solid State Drive (SSD)로의 전환이 진행되고 있다. 또한 사물인터넷(IoT), 웨어러블(Wearable) 및 차량용(Automotive) 스토리지가 낸드 플래시의 새로운 시장으로 부각되고 있다.

멀티 레벨링 기술은 낸드 플래시 기반의 스토리지의 용량 증가에 또 다른 기여를 하고 있다. 플래시 제조사의 공정 기술과 함께 발전한 멀티 레벨링 기술로 인하여 single-level cell (SLC)와 multi-level cell (MLC)를 거쳐 triple-level cell (TLC)까지 상용화되었고, 3D 낸드의 개발에 따른 성능과 신뢰성의 향상은 quadruple-level cell (QLC) 낸드의 상용화로 이어질 것으로 예상된다[2].

하지만 MLC와 TLC는 SLC 대비 상대적으로 낮은 성능과 수명의 한계를 가지고 있으며, 이를 극복하기 위하여 많은 연구들이 진행되어 왔다. 주로 SLC/MLC 혼합 구조의 FTL과 파일시스템 레벨의 여러 연구가 수행되었다[3-5]. Im 등[3]의 연구와 Murugan 등[4]의 연구에서는 SLC 영역에 선택적으로 Hot 데이터를 쓰는 FTL 레벨의 연구를 수행했다. Lee 등[5]은 SLC 영역을 쓰기 버퍼로 사용하는 파일 시스템을 제안했다. 이러한 기법은 인기도(hotness)에 기반한 데이터 분류에 중점을 두었으며 빈번하게 업데이트되는 데이터를 SLC 영역에 기록함으로써 전체 스토리지 성능을 효율적으로 향상 시켰다. 최근 MLC 낸드에 비해 성능과 내구성이 떨어지는 TLC 낸드에 적합한 다양한 기법들이 제안되었다[6,7]. Yao 등[6]의 연구와 Yang 등[7]의 연구에서는 성능과 수명을 향상시키기 위해 SLC/TLC 혼합 구조의 FTL을 제안했다. Yao 등은 SLC와 TLC 영역의 수명을 균형 있게 조정하여 스토리지의 수명을 향상시키는 워크로드 인식 FTL을 제안하였고, Yang 등은 SLC 및 TLC 영역의 크기를 동적으로 조정하여 성능을 향상시키는 기법을 제안하였다.

한편, 호스트의 입출력 특성에 대한 연구를 통하여 시스템 또는 스토리지 성능을 향상시킨 연구들도 활발히 진행되어 왔다[8-10]. 예를 들면, Multi-streamed SSD는 호스트의 입력 데이터를 각 데이터의 수명으로 구분하여 기록함으로써 성능을 향상시키는 연구이다[10].

본 논문에서는 파일시스템 저널링의 입출력 특성을 이용하여, SLC 버퍼의 용도를 쓰기 버퍼뿐만 아니라 파일시스템의 저널링 용도로도 사용하는 새로운 SLC 쓰기 버퍼 기술을 제안하였다. 또한, 저널링의 쓰기 패턴을 이용하여 호스트 입력 데이터로부터 저널 데이터를 분리하고, 고성능의 SLC 영역에 저널 데이터를 기록함으로써, 시스템 응답성에 큰 영향을 미치는 동기식(synchronous) 방식 저널링의 성능을 향상할 수 있음을 보였다. 본 기술은 Non-volatile Memory (NVM) 등의 추가적인 하드웨어를 채용하지 않고 기존의 SLC 쓰기 버퍼를 채용한 TLC 스토리지 만을 이용하여 가격 경쟁력을 유지하고, 호스트 레벨의 변경 없이 스토리지 레벨의 변경으로 성능을 향상시킬 수 있었다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 관련 연구들에 대해 살펴보고, 3장에서는 제안 기술에 대하여 상세하게 설명하였고, 4장에서는 실험 환경과 실험 결과에 대하여 논의한다. 마지막으로 5장에서 결론을 맺는다.

## 2. 관련 연구

본 장에서는 먼저 TLC 낸드 스토리지에서 사용되는 SLC 쓰기 버퍼에 대하여 살펴보고, 본 논문에서 주목하는 EXT4 파일시스템의 저널링 메커니즘의 특성을 이해한다.

### 2.1 SLC 쓰기 버퍼

현재 상용되고 있는 낸드 플래시 메모리는 SLC, MLC, TLC의 세 가지로 분류할 수 있다. 이와 같은 낸드 플래시 메모리의 분류는 단일 셀에 저장할 수 있는 비트의 수에 의한 분류이며, 일반적으로 알려진 바와 같이 단일 셀에 저장할 수 있는 비트 수가 증가하면 동일한 제조 비용으로 플래시 메모리의 용량을 증가시킬 수 있다. 하지만, 플래시의 성능과 수명을 현저하게 감소시킨다는 단점을 가지고 있다. 예를 들어, TLC의 성능과 수명은 SLC의 5 분의 1과 100 분의 1 수준으로 보고되고 있다[11,12].

이러한 TLC의 단점을 보완하기 위하여 낸드 플래시 제조업체는 그림 1과 같이 동일 낸드 chip에서 SLC방식과 TLC 방식을 함께 사용할 수 있는 기능을 제공하고 있다. 따라서, TLC 낸드 기반의 스토리지의 성능을 향상시키기 위하여 스토리지의 일부 영역을 SLC 방식으로 동작시키는 SLC 쓰기 버퍼 방식이 많은 제조사들로부터 상용화되고 있다[13,14].

따라서, 호스트로부터의 입력된 데이터는 먼저 고성능의 SLC 영역으로 입력되고 idle time동안 TLC 영역으로 이동하는 방식을 사용함으로써 사용자에게 SLC 수준의 성능을 보여준다. 그림 2는 상용되고 있는 TLC 기반의 SSD의 시간에 따른 성능을 보여준다. 초기 높은

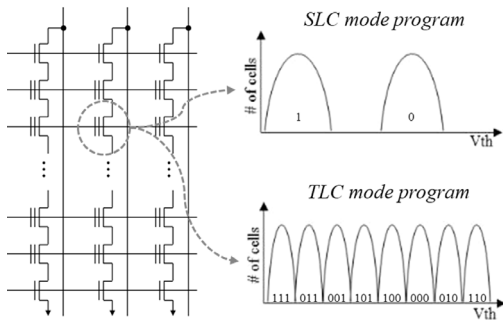


그림 1 SLC 모드 및 TLC 모드 프로그램  
Fig. 1 SLC mode and TLC mode program

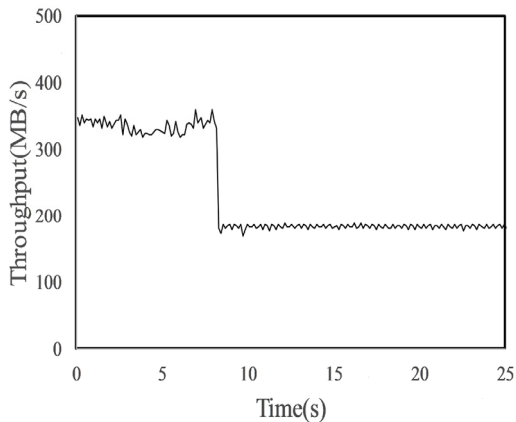


그림 2 TLC 낸드 기반 SSD의 쓰기 성능  
Fig. 2 Write throughput of TLC-based SSD

성능은 호스트로부터 입력된 데이터가 고성능의 SLC 쓰기 버퍼에 기록되는 성능이며, 이후 8초 지점부터는 SLC 쓰기 버퍼가 차서 SLC 쓰기 버퍼를 거치지 않고 TLC 영역으로 직접 기록되어 성능이 낮아진다. 이후 충분한 idle time이 확보되면 SLC 쓰기 버퍼의 데이터가 TLC로 이동하여 버퍼가 다시 확보되고 성능은 초기 SLC 성능으로 회복된다.

## 2.2 파일시스템 저널링

파일시스템 저널링은 파일시스템의 무결성을 유지하기 위하여 파일의 변경사항을 데이터 영역에 저장하기 전에 저널 영역에 로그로 유지하는 방식으로, 리눅스의 대표적인 저널링 파일시스템인 EXT4는 Ordered, Data, Writeback 세 개의 모드를 지원한다[15]. Data 모드는 세 가지 모드 중 안정성이 가장 높은 모드로 변경된 파일데이터와 메타데이터를 모두 저널 영역에 기록한다. Ordered 모드는 메타데이터만 저널 영역에 기록하며, 그 전에 파일데이터를 데이터 영역에 기록함으로써 일관성을 유지한다. Writeback 모드는 Ordered 모드와

같이 저널 영역에 메타데이터 저장하지만, 파일데이터와 메타데이터 기록의 순서는 보장하지 않는다.

파일시스템 저널링은 시스템 성능에 다음과 같은 영향을 준다. 첫째, 저널링은 파일시스템의 일관성을 유지하기 위하여 추가적인 쓰기 작업을 수행하게 되어 스토리지로의 쓰기 양을 증가시킨다. 예를 들어, Data 모드의 경우 파일 데이터를 저널영역과 데이터 영역에 두 번 쓴다. 둘째, 저널링 방식은 동기식 쓰기를 증가시킨다. 예를 들어, Ordered 모드는 메타데이터를 저널 영역에 기록하기 전에 파일데이터의 기록을 완료하여야 한다. 이로 인하여 뒤따르는 동작의 지연을 유발하여 시스템의 응답성에 직접적인 영향을 준다.

파일시스템의 저널링 영역은 스토리지가 포맷될 때 데이터 영역과 분리된 영역에 정해지며, 저널 데이터는 저널 영역에 라운드 로빈 방식의 순차적으로 기록된다. 본 논문에서는 이와 같은 파일시스템 저널링의 쓰기 패턴을 활용하여 SLC 쓰기 버퍼를 사용하는 스토리지의 성능을 향상시키는 기법을 제안한다.

## 3. 제안 기법

본 장에서는 본 논문에서 제안하는 SLC 쓰기 버퍼 방식을 적용한 낸드 플래시 기반 스토리지의 구조와 세부 구현 항목에 대해 설명한다.

### 3.1 SLC 버퍼 쓰기와 저널 쓰기

일반적인 SLC 쓰기 버퍼 방식과 달리 본 논문에서 제안하는 방식은 SLC 영역의 일부를 저널 데이터를 위한 별도의 영역으로 지정하고 호스트로부터 입력된 데이터에서 저널 데이터만 추출하여 별도로 지정한 저널 영역에 기록한다. 파일시스템 저널링은 저널 데이터를 라운드 로빈 방식으로 기록하기 때문에 저널 데이터는 저널 영역에 순차적으로 기록된다. 한편, 낸드 플래시는 In-place 업데이트를 지원하지 않기 때문에, 본 논문의 제안 기법은 호스트의 저널 데이터를 위하여 프리 블록(free block)을 할당 받아 Append 방식으로 저널 데이터를 기록하고 이전에 기록된 저널 데이터를 무효화하는 방식으로 처리한다. 따라서 이전에 기록된 저널 데이터는 새로운 저널 데이터에 의하여 순차적으로 무효화되며, 완전히 무효화된 블록을 프리 블록 전환하는 방식으로 garbage collection(GC)과 같은 추가적인 작업 없이 지속적으로 프리 블록을 만들어낼 수 있다. 이러한 방식으로 데이터 이동 철자(data migration)를 진행하지 않고도 저널 데이터를 저널 영역에 지속적으로 기록할 수 있다. 이로 인하여 동기식의 파일시스템 저널링을 고성능의 SLC 영역에 지속적으로 수행함으로써 응답속도의 향상을 가져올 수 있다. 그림 3의 예시는 SLC 영역의 4개 낸드 블록에 호스트의 쓰기 요청이 수행된 상태

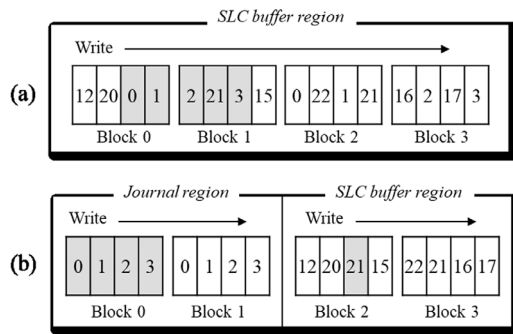


그림 3 쓰기 동작 예시. (a) 일반적인 방식, (b) 제안 기법  
Fig. 3 Example of write operations. (a) Conventional scheme, (b) Proposed scheme

를 보여준다. 각각의 낸드 블록에는 4개의 논리 블록 주소(LBA)에 해당하는 데이터가 기록되었으며, 파일시스템의 저널 영역은 스토리지가 포맷될 때 논리 블록 주소 0부터 3까지로 고정되어 만들어져 있음을 가정하였다. 그림 3(a)의 일반적인 방식과 달리 그림 3(b)의 제안 기법의 경우, SLC 영역을 저널 영역과 쓰기 버퍼 영역으로 구분한 것을 보여주며, 블록 1에 업데이트된 저널 데이터(LBA 0-3)에 의해서 블록 0에 기록되어 있던 이전 저널 데이터가 모두 무효화되었음을 보여준다. 따라서, 블록 0은 프리 블록이 되고, 이후 저널 데이터 기록을 위해 다시 활용될 수 있다. 이와 같이 본 논문의 제안 방식에 의하면 데이터 이동 절차를 진행하지 않고도 저널 데이터를 저널 영역(블록 0, 1)에 지속적으로 기록할 수 있다.

### 3.2 TLC 직접 쓰기와 데이터 이동

SLC 쓰기 버퍼를 사용하는 TLC 낸드 기반 스토리지는 SLC 쓰기 성능을 지속적으로 활용하기 위해서 SLC 쓰기 버퍼의 데이터를 idle time에 TLC영역으로 이동시켜야 한다. 하지만 충분한 idle time이 없는 과중한 워크로드에서는 SLC 쓰기 버퍼가 모두 소모되면 호스트에서 입력되는 데이터는 SLC 쓰기 버퍼를 거치지 않

고TLC 영역으로 바로 기록된다. 따라서 쓰기 성능은 SLC 성능에서 TLC 성능으로 급격히 감소하게 된다.

반면에 본 논문에서 제안하는 SLC 쓰기 버퍼 방식은 앞서 설명한 데로 SLC 쓰기 버퍼가 모두 소진된 상황에서도 저널 데이터를 위한 프리 블록을 지속적으로 확보할 수 있다. 따라서 저널 데이터를 고성능의 SLC 영역에 지속적으로 기록할 수 있어 저널링의 응답 속도를 향상시킬 수 있다. 또한, 저널 데이터에 대한 데이터 이동 절차를 생략할 수 있기 때문에 TLC 영역으로 기록되는 데이터의 양을 줄일 수 있어 성능 향상뿐만 아니라 스토리지의 수명 향상의 효과도 함께 얻을 수 있다.

### 3.3 OpenSSD를 활용한 SLC 버퍼 쓰기 구현

본 논문의 제안 기법은 성능 측정을 위하여 실제 하드웨어인 OpenSSD Jasmine Board Platform[16]를 사용하여 구현하였다. 그림 4는 호스트로부터의 쓰기 요청에 대한 처리 순서의 블록 다이어그램이다. 저널 감지기(journal detector)는 호스트의 쓰기 요청 데이터에서 저널 데이터를 구분한다. 저널 데이터를 구분하기 위해서 호스트가 쓰기 요청을 하면 저널 감지기는 저널링의 라운드 로빈 방식의 순차 쓰기 특성을 탐지하여 저널 데이터를 구분하고[17], 쓰기 요청이 저널 데이터인지를 나타내는 플래그를 이벤트 큐(event queue)에 설정한다.

SLC 영역 관리자(SLC region manager)는 SLC 영역의 주소 사상(address mapping) 및 SLC영역에서 TLC 영역으로의 데이터 이동 등을 관리한다. SLC/TLC 혼합 구조의 저장 장치를 에뮬레이션하기 위해 낸드 모드 변환기(NAND mode changer)를 추가하여 입력된 쓰기 데이터에 따라 낸드의 쓰기 방식을 SLC 방식 또는 TLC 방식으로 변경하였다. 그림 5는 베이스라인과 제안 기법의 SLC 영역 할당 방식을 보여준다. OpenSSD Jasmine Board는 MLC 낸드의 64GB 용량으로 구성되어 있으며, 전체 64GB 중에서 768MB를 SLC 영역으로 할당하여 성능 비교를 위한 베이스라인 기술의 SLC 쓰기 버퍼를 구현하였다. 본 논문의 제안 기법은 앞 절에서 설명한대로 SLC 영역의 일부를 저널 영역으로 할당

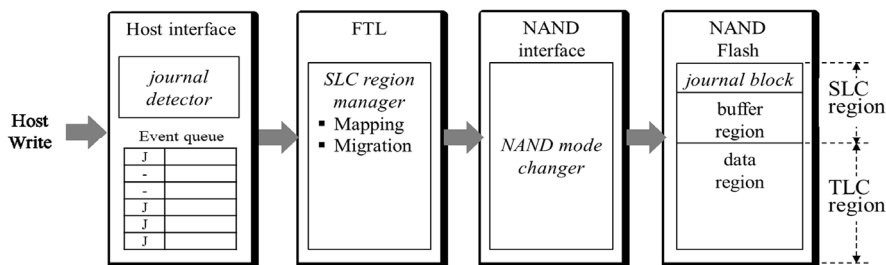


그림 4 쓰기 요청에 대한 처리 순서의 블록 다이어그램  
Fig. 4 Simplified write path of our combined SLC-TLC storage

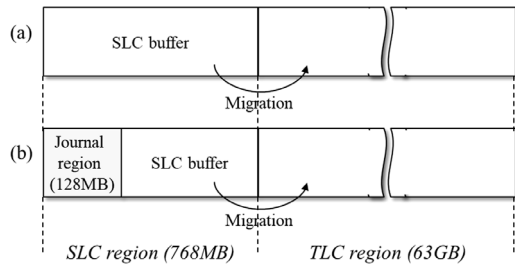


그림 5 SLC 영역 비교. (a) 베이스라인 방식, (b) 제안 기법 방식

Fig. 5 Comparison of SLC region. (a) Baseline scheme, (b) Proposed scheme

하였으며 저널 영역의 크기는 EXT4 파일시스템의 저널 영역과 동일하게 128MB로 설정하였다. 베이스라인 기술의 경우, 호스트로부터 입력되는 데이터는 모두 SLC 버퍼 영역에 기록되고 idle time 동안 TLC 영역으로 이동한다. 반면에 본 제안 기법의 경우에는 호스트로부터 입력된 데이터 중에서 저널 데이터는 저널 영역에 기록되고, 그 외 데이터만 SLC 버퍼 영역에 기록되며 idle time 동안 TLC 영역으로 이동한다.

MLC로 구성되어 있는 OpenSSD는 TLC 낸드를 지원하지 않기 때문에 SLC와 TLC 영역에 대하여 낸드 프로그램 동작 시 적절한 Delay를 추가하여 SLC 영역과 TLC 영역의 성능 차이를 에뮬레이션 하였다.

#### 4. 실험 및 분석

본 장에서는 성능 평가를 위한 실험 환경에 대하여 설명하고, OpenSSD를 활용한 제안 기법과 베이스라인 기법의 성능을 비교하였다.

##### 4.1 실험 환경

본 논문의 제안 기법에 대한 성능 평가는 3.40GHz Intel Core i7-2600 CPU와 8GB RAM을 채용한 PC에서 진행되었으며, OS는 리눅스 커널 버전 4.5.4의 Ubuntu 14.04이다. EXT4 파일시스템의 Data 모드와 Ordered 모드의 두 가지 저널링 방식을 사용하였으며, 성능 평가를 위하여 FIO[18]를 이용한 합성 워크로드와 Filebench [19]를 이용한 다양한 실제 워크로드를 사용하였다.

##### 4.2 FIO 워크로드

본 논문의 제안 기법의 효과를 확인하기 위하여 저널 데이터의 비율에 따른 성능을 비교하였다. 실험에 사용한 워크로드는 FIO를 사용한 8KB 데이터 단위의 임의 쓰기이며 fsync 시스템 콜(system call)의 주기를 2~128까지 변화시켜 저널 데이터의 양을 변화시켰다. 또한, 스토리지에 기록되는 데이터는 SLC 영역의 두 배로 설정하여 SLC 성능과 TLC 성능을 모두 포함하도록 하였다.

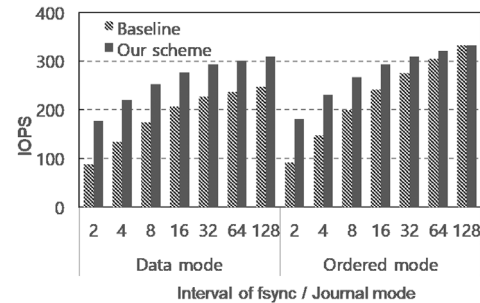


그림 6 fsync 호출 주기에 따른 쓰기 성능 비교  
Fig. 6 Performance comparison by varying the interval of fsync() calls

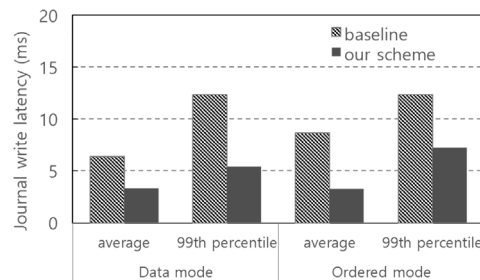


그림 7 쓰기 응답 시간 비교  
Fig. 7 Comparison of journal write latency

그림 6은 fsync 주기에 따른 성능을 보여주며, X 축은 fsync의 주기를 나타내고 Y 축은 임의 쓰기 성능을 나타낸다. 실험 결과와 같이 fsync 호출 간격이 짧아짐에 따라 두 가지 모드에서 베이스라인 대비 각각 최대 102%와 97%까지 성능이 향상되었다. 이는 빈번한 fsync 호출이 저널 데이터의 양을 증가시키고, 이로 인하여 고성능의 SLC 영역에 기록되는 데이터의 양을 증가시키기 때문이다. 다음으로 FIO 임의 쓰기의 저널 쓰기 응답 시간을 비교하였다. 그림 7은 평균 응답 시간 및 99번째 백분위 기준의 응답 시간을 보여주며, Data 모드에서 각각 48% 및 56%의 응답 시간을 감소시켰으며, Ordered모드에서는 각각 62% 및 41%의 응답 시간을 감소시킬 수 있었다. 베이스라인의 경우 저널 데이터는 먼저 SLC 쓰기 버퍼에 기록되고 SLC 쓰기 버퍼가 가득 차면 TLC 영역으로 직접 쓰여지는 반면에 본 논문의 제안 기법은 저널 데이터를 위한 SLC 쓰기 버퍼가 모두 소진된 상황에서도 저널 데이터를 위한 프리블록을 지속적으로 확보할 수 있다. 따라서 저널 데이터를 고성능의 SLC 영역에 지속적으로 기록할 수 있어 저널링의 응답 속도를 향상시킬 수 있다.

##### 4.3 Filebench 워크로드

그림 8은 Filebench의 워크로드별 성능 비교를 보여

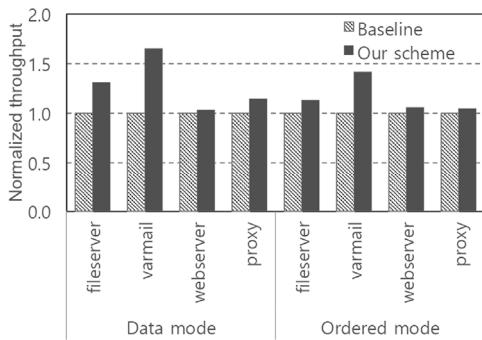


그림 8 Filebench 워크로드 성능 비교

Fig. 8 Performance comparison under the Filebench workloads

표 1 Filebench 워크로드 특성  
Table 1 Characteristics of Filebench workloads

Workload	R : W	Portion of journal data [%]	
		Data mode	Ordered mode
Fileserver	1 : 2	54.5	2.9
Varmail	1 : 1	85.3	45.5
Webserver	10 : 1	52.4	0.2
Proxyserver	5 : 1	78.3	15.9

준다. X축은 저널링 모드와 워크로드를, Y축은 베이스라인 기준의 노멀라이즈 성능을 나타낸다. 읽기 대비 쓰기의 비중이 높은 Fileserver와 Varmail 워크로드에서 베이스라인 대비 제안 기법의 성능이 크게 향상되는 것을 확인할 수 있으며, Data 모드에서 베이스라인 대비 30%와 65%의 성능 향상을 보이고 Ordered 모드에서는 각각 13%와 10%의 성능 개선을 보인다. Data 모드에서 Varmail 워크로드의 성능 향상이 가장 큰 이유는 표 1과 같이 4개 워크로드 중 저널 데이터의 비중이 가장 높기 때문이다. 이는 Data 모드에서 파일데이터와 메타데이터가 모두 저널링되기 때문에 Ordered 모드 대비 저널 데이터가 증가하고, Varmail 워크로드 포함된 fsync 시스템 콜이 메모리의 더티 페이지(dirty page)를 즉시 스토리지에 기록하여 저널링을 발생시키기 때문에 다른 워크로드와 비교하여 저널 데이터가 증가한다.

마지막으로 수명에 직접적인 영향을 미치는 TLC 영역으로 기록되는 데이터의 양을 비교하였다. 그림 9의 X축은 저널링 모드와 워크로드를, Y축은 TLC 영역에 쓰기 요청된 데이터의 양을 나타낸다. 그림과 같이 본 논문의 제안 기법은 TLC 영역으로 기록되는 데이터 양을 크게 줄일 수 있었다. TLC 영역에 기록되는 데이터의 양은 SLC 쓰기 버퍼로부터 이동되는 데이터의 양과 TLC 영역으로 직접 기록되는 데이터 양의 합이다. 따라서, 본 논문의 제안 기법에서 저널 데이터는 SLC 영

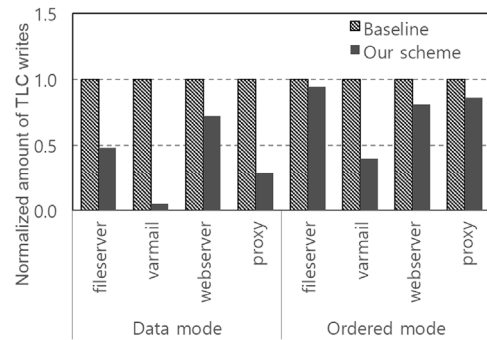


그림 9 TLC 영역의 쓰기 양 비교

Fig. 9 Comparison of the amount of data transferred to the TLC region under the Filebench

역의 저널 영역에 모두 흡수되어 TLC 영역으로 이동되지 않기 때문에 베이스라인 대비 TLC 영역으로 이동하는 데이터의 양을 크게 줄일 수 있다.

## 5. 결론

본 논문은 TLC 낸드 기반 스토리지의 성능 향상을 위하여 파일시스템의 저널링 특성을 이용한 SLC 버퍼 쓰기 방식을 제안하였다. 제안 기법은 파일시스템 저널링의 응답 속도를 향상시키기 위하여 저널 데이터를 위한 영역을 고성능의 SLC 영역에 분리하여 할당하였다. 성능 평가를 위하여 제안 기법을 OpenSSD Jasmine Board Platform에 구현하였고 다양한 워크로드를 사용하여 상용화 기법과 성능을 비교하였다. 그 결과 65%까지의 성능 향상을 확인하였다.

## References

- [1] K.-T. Park et al., "Three-dimensional 128Gb MLC Vertical NAND Flash-memory with 24-WL Stacked Layers and 50MB/s High-speed Programming," *IEEE Journal of Solid-State Circuits*, Vol. 50, No. 1, pp. 204-213, 2015.
- [2] S.-H. Shin et al., "A New 3-bit Programming Algorithm using SLC-to-TLC Migration for 8 MB/s High Performance TLC NAND Flash Memory," *Proc. of the IEEE Symposium on VLSI Circuits*, pp. 132-133, Jun. 2012.
- [3] S. Im and D. Shin, "ComboFTL: Improving Performance and Lifespan of MLC Flash Memory using SLC Flash Buffer," *Journal of Systems Architecture*, Vol. 56, No. 12, pp. 641-653, Dec. 2010.
- [4] M. Murugan and D. Du, "Hybrot: Towards Improved Performance in Hybrid SLC-MLC Devices," *Proc. of the IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems*, pp. 481-484, Aug. 2012.

- [5] S. Lee, K. Ha, K. Zhang, J. Kim, and J. Kim, "FlexFS: A Flexible Flash File System for MLC NAND Flash Memory," *Proc. of the USENIX Annual Technical Conference*, pp. 115-128, Jun. 2009.
- [6] D. Liu, L. Yao, L. Long, Z. Shao, and Y. Guan, "A Workload-aware Flash Translation Layer Enhancing Performance and Lifespan of TLC/SLC Dual-mode Flash Memory in Embedded Systems," *Micro-processor and Microsystems*, Vol. 52, pp. 343-354, Jul. 2017.
- [7] M.-C. Yang, Y.-H. Chang, C.-W. Tsao and C.-Y. Liu, "Utilization-aware Self-tuning Design for TLC Flash Storage Devices," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 24, No. 10, pp. 3132-3144, Oct. 2016.
- [8] D. Jeong, Y. Lee, and J.-S. Kim, "Boosting Quasi-asynchronous I/O for Better Responsiveness in Mobile Devices," *Proc. of the USENIX Conference on File and Storage Technologies*, pp. 191-202, Feb. 2015.
- [9] S. Kim, H. Km, J. Lee, and J. Jeong, "Enlightening the I/O Path: A Holistic Approach for Application Performance," *Proc. of the USENIX Conference on File and Storage Technologies*, pp. 345-358, Feb. 2017.
- [10] J.-U. Kang, J. Hyun, H. Maeng, and S. Cho, "The Multi-streamed Solid-state Drive," *Proc. of the USENIX HotStorage*, pp. 13-17, Jun. 2014.
- [11] J. Jeong, S. S. Hahn, S. Lee, J. Kim, J. Jeong, S. S. Hahn, S. Lee, and J. Kim, "Lifetime Improvement of NAND Flash-based Storage Systems using Dynamic Program and Erase Scaling," *Proc. of the USENIX Conference on File and Storage Technologies*, pp. 61-74, Feb. 2014.
- [12] A. Maislos, "A New Era in Embedded Flash Memory," *Flash Memory Summit 2011*, Aug. 2011.
- [13] Samsung 850 EVO SSD. [Online]. Available: <http://www.samsung.com/semiconductor/minisite/ssd/product/consumer/850evo.html>, 2015.
- [14] SanDisk X300 SSD. [Online] Available: <http://downloads.sandisk.com/downloads/datasheet/x300-datash eet.pdf>.
- [15] EXT4 Documentation [Online]. Available: <https://www.kernel.org/doc/Documentation/filesystems /ext4.txt>.
- [16] The OpenSSD Project [Online]. Available: [http://www.openssd-project.org/wiki/The\\_OpenSSD\\_Project/](http://www.openssd-project.org/wiki/The_OpenSSD_Project/).
- [17] E. Lee, H. Bahn, M. Jeong, S. Kim, J. Yeom, S. Yoo, S. H. Noh, and K.G. Shin, "Reducing Journaling Harm on Virtualized I/O Systems," *Proc. of the ACM International Systems and Storage Conference*, pp. 15-20, Jun. 2016.
- [18] Flexible I/O Tester. [Online]. Available: <https://github.com/axboe/fio>.
- [19] Filebench [Online]. Available: <http://filebench.sourceforge.net/>.



#### 권 기 록

2000년 경북대학교 전자전기공학부 학사  
2002년 경북대학교 전자공학과 석사. 2015  
년~현재 성균관대학교 반도체 디스플레이  
이공학과 박사과정. 관심분야는 운영체제,  
스토리지 시스템



#### 강 동 현

2007년 한국산업기술대학교 컴퓨터공학  
과 학사. 2010년 성균관대학교 전자전기  
컴퓨터공학과 석사. 2018년 성균관대학교  
전자전기컴퓨터공학과 박사. 관심분야는  
스토리지 시스템, 운영체제

#### 엄 영 익

정보과학회논문지  
제 45 권 제 6 호 참조