Journal of The Institute of Electronics and Information Engineers Vol.52, NO.9, September 2015 http://dx.doi.org/10.5573/ieie.2015.52.9.054
ISSN 2287-5026(Print) / ISSN 2288-159X(Online)

논문 2015-52-9-8

SSD 수명 관점에서 리눅스 I/O 스택에 대한 실험적 분석

(An Empirical Study on Linux I/O stack for the Lifetime of SSD Perspective)

정 남 기*, 한 태 희**

(Nam Ki Jeong and Tae Hee Han©)

요 약

낸드 플래시 기반의 SSD (Solid-State Drive)는 HDD (Hard Disk Drive) 대비 월등한 성능에도 불구하고 쓰기 회수 제한이라는 태생적 단점을 가지고 있다. 이로 인해 SSD의 수명은 워크로드에 의해 결정되어 SSD의 기술 변화 추세인 SLC (Single Level Cell) 에서 MLC (Multi Level Cell) 로의 전환, MLC에서 TLC (Triple Level Cell) 로의 전환에 있어 큰 도전이 될 수 있다. 기존 연구들은 주로 wear-leveling 또는 하드웨어 아키텍처 측면에서 SSD의 수명 개선을 다루었으나, 본 논문에서는 호스트가 요청한 쓰기에 대해 SSD가 낸드플래시 메모리를 통해 처리하는 수명관점의 효율성을 대변하는 WAF (Write Amplification Factor) 관점에서 Host I/O 스택 중 파일 시스템, I/O 스케줄러, 링크 전력에 대해 JEDEC 엔터프라이즈 워크로드를 이용해 I/O 스택 최적 구성에 대해 실험적 분석을 수행하였다. WAF는 SSD의 FTL의 효율성을 측정하는 지표로 수명관점에서 가장 객관적으로 사용한다. I/O 스택에 대한 수명 관점의 최적 구성은 MinPower-Dead-XFS로 최대 성능 조합인 MaxPower-Cfq-Ext4에 비해 성능은 13% 감소하였지만 수명은 2.6 배 연장됨을 확인하였다. 이는 I/O 스택의 최적화 구성에 있어, SSD 성능 관점뿐만 아니라 수명 관점의 고려에 대한 유의미를 입증한다.

Abstract

Although NAND flash-based SSD (Solid-State Drive) provides superior performance in comparison to HDD (Hard Disk Drive), it has a major drawback in write endurance. As a result, the lifetime of SSD is determined by the workload and thus it becomes a big challenge in current technology trend of such as the shifting from SLC (Single Level Cell) to MLC (Multi Level cell) and even TLC (Triple Level Cell). Most previous studies have dealt with wear-leveling or improving SSD lifetime regarding hardware architecture. In this paper, we propose the optimal configuration of host I/O stack focusing on file system, I/O scheduler, and link power management using JEDEC enterprise workloads in terms of WAF (Write Amplification Factor) which represents the efficiency perspective of SSD life time especially for host write processing into flash memory. Experimental analysis shows that the optimum configuration of I/O stack for the perspective of SSD lifetime is MinPower-Dead-XFS which prolongs the lifetime of SSD approximately 2.6 times in comparison with MaxPower-Cfq-Ext4, the best performance combination. Though the performance was reduced by 13%, this contributions demonstrates a considerable aspect of SSD lifetime in relation to I/O stack optimization.

Keywords : SSD, Lifetime, WAF, I/O stack, File System, I/O Scheduler, Link Power

* 학생회원, 성균관대학교 반도체디스플레이공학과
(Sungkyunkwan University)
** 정회원, 성균관대학교 정보통신대학
(Sungkyunkwan University)
© Corresponding Author(E-mail: than@skku.edu)
※ 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 대학ICT연구센터 육성지원사업의 연구결과로 수행되었음 (IITP-2015-H8501-15-1005)

Received ; June 9, 2015 Revised ; July 30, 2015 Accepted ; September 3, 2015

Ⅰ. 서 론

최근 낸드 플래시 기반의 SSD는 HDD 대비 비트 당 가격 차이가 줄어들고 있으며, 고 성능, 저 전력 등의 우수성을 바탕으로 개인 플랫폼에서 기업용 서버에 이르기까지 다양한 분야에서 확대 적용 되고 있다. 그러나 SSD는 극복해야할 단점 또한 존재하는데, 대표적으로 플래시 메모리 셀 특성에 기인하여 쓰기 내구성에서 취약하다. 즉, SSD는 한정된 횟수의 쓰기만을 보장하며, 최신 공정의 스케일다운은 이를 더욱 제약한다. SLC는 30~20 nm 공정에서 100 K 수준의 P/E (Program /Erase) 사이클을 유지할 수 있고, MLC는 30 nm 공정에서는 5K 정도의 사이클을 보장하나 20 nm급에서는 3K로 감소되었고, 20 nm 공정의 TLC는 단지 수 백 사이클을 갖는다[1].

이렇듯 SSD 수명은 태생적 단점인 쓰기 횟수의 제한으로 워크로드 의존성을 갖는다. 이러한 워크로드의 수명 의존성은 서버환경의 SLC 제품군에서 MLC로의 전환, 개인 플랫폼 향의 MLC 제품에서 TLC로의 전환에 있어 큰 도전이 되고 있다. 따라서 쓰기 내구성이 더욱 취약해지는 초미세 공정의 다중 비트 셀 채용에 따른 SSD 수명 연장을 위한 다양한 연구가 필요하다.

전형적인 SSD 스토리지 시스템은 그림 1과 같은 하드웨어/소프트웨어 구조를 가지고 있다. 호스트의 I/O 스택(파일 시스템, I/O 스케줄러)과 컨트롤러, FTL (Flash Translation Layer), 낸드플래시, DRAM 버퍼 등으로 이루어져있다. SSD의 제한된 수명을 개선, 연장하기 위한 기존 연구는 주로 FTL과 하드웨어 아키텍처 수준에서 진행되어왔다. 즉, FTL에서는 wear-leveling 알고리즘 개선[2], 매핑 방법 개선[3], 데이터 중복제거[4], 데이터 압축기술[5] 등을 통한 불필요한 데이터 이동 제거로 SSD의 쓰기 양을 감소시켰으며, 하드웨어 구조 면에서는 HDD를 SSD의 쓰기 캐시 (Write cache)로 사용하는 복합 구조를 통해 수명을 연장하였다[6]. 또한, Host의 I/O 스택에 대한 연구는 수명보다는 성능 개선[7], 성능 평가[8~9]가 주를 이루었다. 본 연구의 동기는 워크로드와 I/O 스택에 따라 스토리지 성능 및 전력이 상이하여 최적의 조합을 구성해야 한다는 점에서 출발하였다.[10] SSD 성능과 수명관점에서 비교 분석에 대해 I/O 스택의 영향에 대해서는 기존의 연구에서 다루지지 않았다. 이에 본 연구에서는 I/O 스택의 구성에 따른 SSD 수명 관점의 실험적 연구를 통해 성능과 수명을 분석하고, 수명 관점에서 I/O 스택의 최적 구성을 도출하고자 한다. 위와 같은 접근 방법은 세 가지 방향에서 기여할 수 있다. 첫째, 기존 연구 결과와 결합을 통해 SSD의 수명을 더욱 향상할 수 있고, 둘째, I/O 스택 구성에 따라 SSD 수명에 상당한 유의차가 발생한다면, I/O 스택의 최적화 구성에 있어 수명 고려의 당위성을 제시할 수 있다. 셋째, SSD의 수명관점에서 기존의 FTL 연구 범위를 I/O 스택의 구성에 따른 최적화 방법까지 확장할 수 있다.

본 논문에서는 SSD 수명과 이에 따른 신뢰성에 대한 영향이 가장 큰 분야인 데이터센터 (Datacenter) 환경에 대해 JEDEC 워크로드를 바탕으로 WAF(Write Amplification Factor)를 이용하여 수명관점에서 분석하고자 한다. 특히 데이터센터 환경은 SSD를 채용하는 규모와 유지 비용이 지속적으로 증가하고 있어, I/O 스택 구성에 따른 수명관점의 효과가 비중있게 반영될 수 있다. JEDEC 워크로드를 사용한 이유는 신뢰성에 대한 국제 표준이기에 SSD 제조사를 포함하여 데이터센터 산업계에서 보편적으로 사용하기 때문이다. 실제로 삼성전자에서는 SSD 제품 개발 시 Random 4K와 8K, JEDEC 워크로드 등을 사용하여 WAF에 대해 품질 및 기술 관점에서 평가와 연구를 진행하고 있으며, 인텔 및 샌디스크에서도 수명 보증을 위해 JEDEC 워크로드에 기반하고 있다.[17]

본 논문의 나머지 부분은 다음과 같이 구성된다. Ⅱ장에서는 I/O 스택에 대한 관련 연구를 살펴보고 Ⅲ장에서는 스토리지 I/O 스택에 대한 기술과 그 범위를 정한다. Ⅳ장에서는 실험에 사용되는 수명 워크로드를 기술하며, Ⅴ장에서는 실험 및 결과 분석을 통해 I/O 스택의 최적 구성을 도출한다. 마지막으로 Ⅵ장은 본 논문의 내용을 요약하고 정리한다.

Ⅱ. 관련 연구

I/O 스택에 대한 기존 연구는 주로 성능 개선 및 성능 평가 분야로 크게 나눌 수 있다.

성능 개선 연구는 I/O 스택의 최적화[7], 파일 시스템 개선[11] 등이 있으며, 대표적으로 [7]에서는 I/O 처리 지연시간 감소를 위해 종래의 I/O 구조에서 하위 계층에 대한 추상화 계층을 도입하여 인터럽트 등에 의해 야기

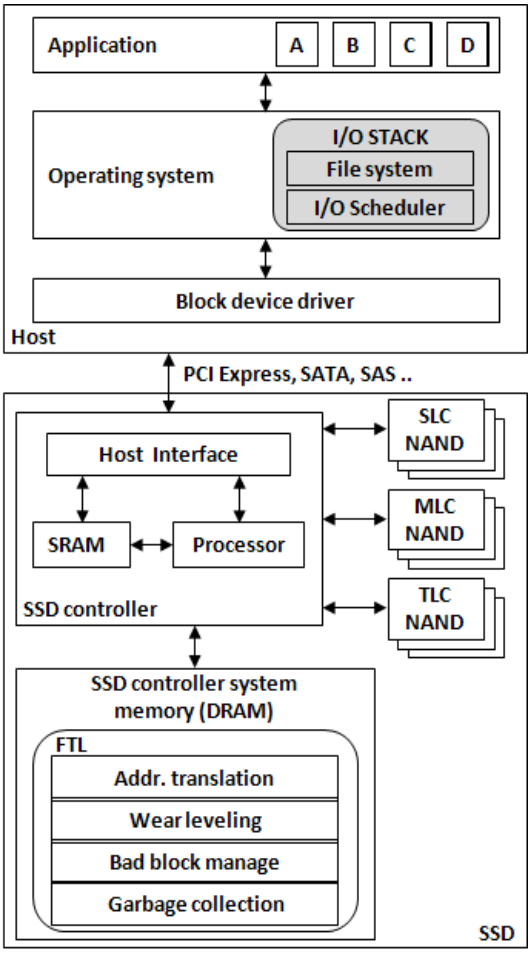


그림 1. SSD 스토리지 시스템의 구조
Fig. 1. Overall structure of whole SSD storage system.

된 부수적인 컨텍스트를 제거, 통합하는 I/O 스택의 최적화를 제안하였다.

성능 평가에 대한 기존 연구는 I/O 스택의 구성요소인 파일 시스템과 IO 스케줄러 등에 대한 비교 분석으로, [8]에서는 성능 관점에서 최적 파일 시스템의 선정, [9]는 성능과 전력에 효율적인 파일시스템과 I/O 스케줄러의 구성에 대해 제안하였다.

또한, 수명 평가에 대한 연구의 일환으로, 기존 연구의 분석 인자(파일 시스템, I/O 스케줄러)에 호스트 환경에서 성능에 직접적으로 영향을 줄 수 있는 링크전력을 추가하여 이를 성능 및 수명 관점에서 비교, 분석하였다.

Ⅲ. 스토리지 I/O 스택

스토리지 I/O 스택에 대한 일반적인 구조는 그림 2와

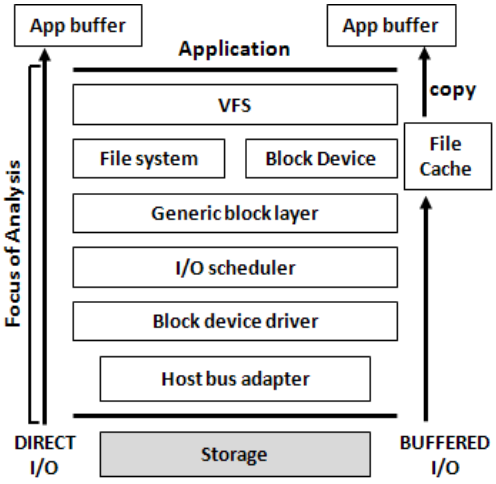


그림 2. 스토리지 I/O 스택 구조
Fig. 2. Structure of Storage I/O stack.

같다. 파일 시스템은 데이터를 파일과 디렉토리로 조직하여 데이터 생성, 갱신, 삭제를 위한 관리 방법을 제공한다. VFS (Virtual File System)는 파일 시스템 위의 추상 계층으로 응용 프로그램과 파일 시스템의 인터페이스를 제공하며 디스크를 파일처럼(예:/dev/sda) 접근 가능하게 만든다. Generic block layer는 스토리지의 물리적 주소와 논리적 주소의 매핑을 위해 LVM (Logical Volume Manager)를 호출하고, 상위 계층의 I/O 요청을 I/O 스케줄러에 전달한다. I/O 스케줄러는 I/O 요청을 재 정렬 및 통합하고, 블럭 디바이스 드라이버는 I/O 스케줄러에서 가져온 I/O 요청을 스토리지에 전달한다.

본 논문에서는 수명에 영향을 줄 수 있는 성능인자인 파일 시스템, I/O 스케줄러, 링크 전력 정책에 집중하고자 한다. 호스트로부터 전송된 대량의 데이터는 플래시 메모리에 저장을 위해 쓰기를 해야 하므로 성능과 식 (1)의 플래시 메모리 쓰기 총량은 비례하게 되며, 성능은 데이터를 전송할 수 있는 능력이므로 식 (2)의 호스트 쓰기 총량 관계 역시 비례한다. 결국 수명은 성능 인자를 포함한 식 (2)의 WAF로 대변된다. 하지만, 파일 시스템, I/O 스케줄러와 같은 성능인자의 조합에 의해 식(1), 식(2)의 결과는 가변될 가능성이 있다. 하기의 수식은 ATA 표준으로 정의된 스토리지의 속성을 포함한 스마트 데이터[12] ID 177 (P/E 사이클), ID 241(호스트 쓰기 총량)을 이용하였다.

$$\text{플래시 메모리 쓰기 총량} = \text{P/E 사이클} \times \text{SSD 용량(GB)} \quad (1)$$

WAF = 플래시 메모리 쓰기 총량 / 호스트 쓰기 총량 (2)

파일 시스템은 기업용으로 사용되는 RHEL의 기본 파일 시스템인 Ext3, Ext4, XFS를 선정하였다. RHEL 5.0에서는 Ext3가 기본 파일 시스템으로 사용되며, RHEL 6.0과 7.0은 각각 Ext4, XFS가 기본 파일 시스템으로 사용되고 있다.

파일 시스템의 성능은 파일시스템의 종류에 따라 다르고, 성능과 저널링 기능은 식 (1)의 P/E 사이클 소모에 직접적인 영향을 준다. 특히 저널링은 데이터 또는 메타 데이터를 기록하는 커밋(Commit) 작업 후 디스크 영역에 데이터를 쓰는 체크포인트(Checkpoint) 과정으로 이루어져 디스크를 중복적으로 접근한다. I/O 스케줄러[13]는 효율적인 처리를 위해 I/O 이슈 순서를 바꾸며, 그 결과 성능이 가변되어 식 (1), (2)에 간접적으로 영향을 준다.

뿐만 아니라 파일 시스템과 I/O 스케줄러는 워크로드에 따라 성능 차이가 발생한다. 예를 들어, 임의 읽기와 순차 쓰기 중심의 워크로드에서는 Ext2가 가장 우수한 성능을 보이지만, 웹 서버와 같은 읽기 중심의 워크로드에서는 XFS 파일 시스템이 가장 우수한 성능을 보인다. 파일 서버, 메일 서버 워크로드에서는 각각 Deadline, Noop 스케줄러 사용 시 최고 성능을 가지게 된다[8~9].이러한 특징은 기존 연구에서 워크로드 특징에 따라 가장 우수한 성능을 내는 파일시스템과 I/O 스케줄러의 조합에 대해 다양한 실험을 통해 도출되었으며, 본 논문에서는 식 (2)의 관점에 이를 확대 적용하였다.

링크 전력 정책[14]은 SATA 등의 링크에 유휴시간이 존재할 때 링크 전력을 줄이기 위해 링크를 저 전력모드로 진입시키기 위한 정책이다. Min_power는 Max_performance에 비해 데이터 전송 시 유휴시간이 발생하면 저 전력모드로 진입하여 전력소모를 감소시키는 장점이 있지만, 이에 대한 오버헤드로 호스트 쓰기 총량을 감소시키는 영향을 준다.

본 논문에서는 상기 논의된 I/O 스택의 성능인자에 대한 수명관점 평가를 위해 그림 3과 같이 데이터 경로(Data Path)[15]를 구성하였다. 블록 경로(Block Path)는 파일 시스템을 배제하여 SSD의 펌웨어 관점에서 평가 가능하다. 이에 반해 파일 경로(File Path)는 스토리지의 데이터를 관리하는 파일 시스템으로 데이터 접근 시

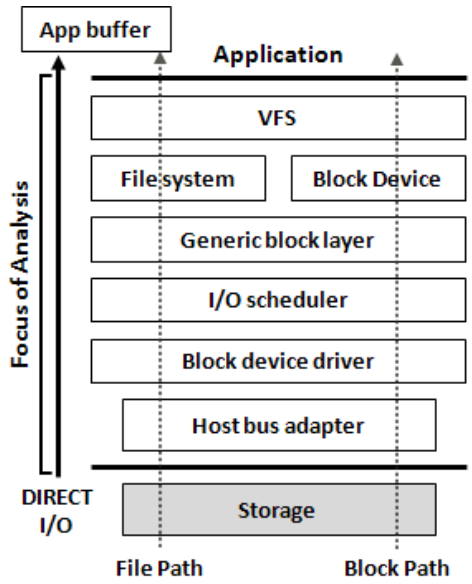


그림 3. 데이터 경로 - 블록, 파일
Fig. 3. Data path - Block, File.

파일 위치 정보에 대한 메타 데이터를 포함하여 평가할 수 있다. 블록 경로와 파일경로 모두 Direct I/O 방식을 사용하여 메타 데이터를 항상 버퍼 캐시에 상주시켜 캐시의 영향성을 배제하였다. 이와 같은 구성을 통해 블록 경로와 파일 경로 간의 수명 영향성에 대해 비교, 분석을 하고자 한다.

Ⅳ. 수명 시험 워크로드

본 논문에서는 SSD 수명 시간 평가를 위해 JEDEC 엔터프라이즈(Enterprise) 워크로드를 활용하였다. 엔터프라이즈 워크로드에는 파일서버, Varmail, OLTP 등 다양한 워크로드가 존재하지만, JEDEC 워크로드는 이를 공통적으로 포괄할 수 있어 삼성, 인텔, 샌디스크를 포함한 산업계에서 수명 시험에 필수적으로 사용된다[16~17]. 그 이유로 워크로드 관점에서 Database, OLTP, Email의 공통점을 추출하여 보편성을 확보하였고, 동시에 TRIM, UNMAP 명령어를 미사용하여 Random data pattern을 사용하기 때문에 WAF에 혹독한 스트레스를 주기 때문이다. 또한 블록의 크기가 FTL 매핑 정보와 정렬이 안 맞기 때문에 FTL의 매핑 평가에 사용하는 Random 4K, 8K 수명 평가를 모두 포함할 수 있는 대표성을 지닌다.

수명 시험 워크로드 시퀀스는 그림 4와 같으며, 디스

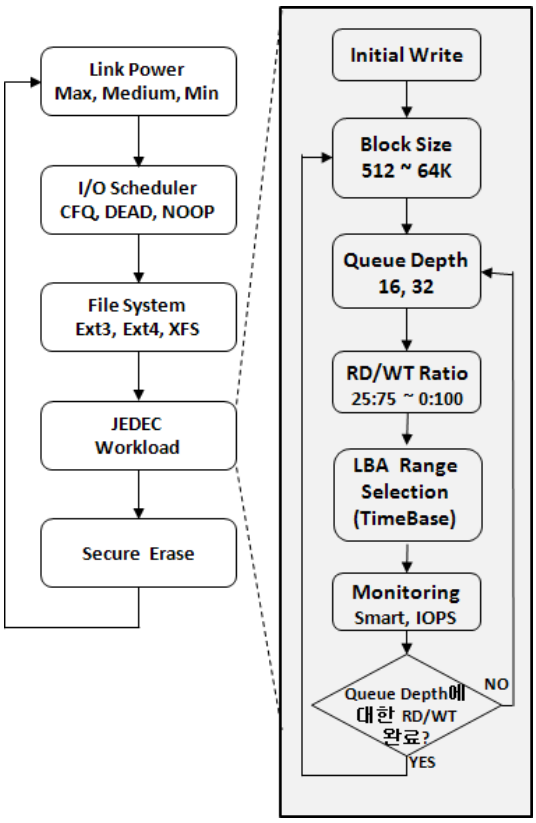


그림 4. SSD 수명 시험 워크로드 시퀀스
Fig. 4. SSD Lifetime workload sequence.

크 전체영역을 순차 쓰기를 하여 SSD 상태를 지속(Sustain) 상태로 만들었다. 이를 통해 SSD의 FTL 동작을 유발시켜 필드 조건의 사용자 환경을 반영하였다. 만일 SSD가 포맷 상태의 초기화 된 깨끗한 상태이면 FTL의 복잡한 로직 동작을 모두 거치지 않을 수 있어 최대 성능의 평가로는 의미가 있지만 수명 관점의 평가로는 적합하지 않다. SSD의 수명 평가를 위해 디스크에 전송되는 블록 크기와 비율을 다양하게 포함하고 있다. 이외 다양한 워크로드 특성을 반영하고 FTL의 특성을 평가하기 위해 블록 크기 이외, 읽기, 쓰기 비율을 다변화하였다. 512 B 블록 크기는 총 4%를 차지하며, 1 K~3.5 KB의 블록 크기는 0.5 KB의 크기 마다 각각 1%로 구성하고 있다. 4 KB의 블록 크기는 총 67% 차지하며, 8 KB의 블록 크기는 10%를 점유하며, 16 KB의 크기는 7%로 구성하고, 32 KB와 64 KB 블록 크기는 각각 3%를 이루고 있다. 이 중 4 KB 크기의 비중이 가장 큰 이유는 OS 환경에서 데이터가 페이지 사이즈인 4 KB 단위로 전송되는 확률이 가장 많기 때문이고, SSD 또한 물리적 블록과 논리적 블록의 매핑의 단위를 4 KB에 기반하여 사용하고 있다. 본 논문에서는 블록 크기별 비율은 전체 처리 시간에 대한 각 블록별 실행 시간의 비로 처리하였고, LBA 영역을 1~5%, 6%~20%, 21~100% 3개로 나누어 각각 50%, 30%, 20% 비율로 수명시험 워크로드가 SSD를 접근하도록 스케줄링 하였다. 다양한 워크로드의 환경을 모사하기 위해 임의 읽기와 쓰기의 비율은 25:75, 50:50, 75:25, 0:100 으로 구성하여 데이터센터 환경에서의 SSD 내부 FTL 처리 방식을 반영하고자 하였다.

V. 실험 및 결과 분석

1. 실험 방법

수명 평가 실험 환경으로 운영체제는 Ubuntu 14.04와 커널 3.13.0-24-generic을 사용하였으며, 24 Core 와 16 GB RAM을 갖춘 시스템에서 MLC 기반 SSD를 대상으로 평가하였다. 평가방법은 블록 경로와 파일 경로에 대해 수명시험 워크로드를 수행하여 성능과 WAF를 분석하였으며, 실험과 실험 사이에는 Secure Erase를 통해 스토리지를 초기화하여 동일한 조건으로 평가하고자 하였다. 수명 평가 시험 시간은 1개 조합 평가 당 9시간 소요되며, 10번의 평균치로 정합성을 확보하였으고, 블록경로는 410 시간, 파일 경로는 820 시간 평가를 하였다.

실험은 벤치마크 Fio, Smartctrl, Iostat을 이용하여 쉘 기반 스크립트를 그림 4의 워크로드로 작성하여 평가하였다. 대표적으로 Fio는 Direct I/O 위해 표 1과 같이 사용하였으며, Smartctrl은 WAF 측정을 위한 스마트 데이터를 기록하기 위해 사용하였으며, Iostat은 IOPS (input/output operation per second)를 분석하기 위해 사용하였다.

표 1. 벤치마크 Fio 주요 파라미터 설정 값
Table 1. Parameter configuration for Benchmark Fio.

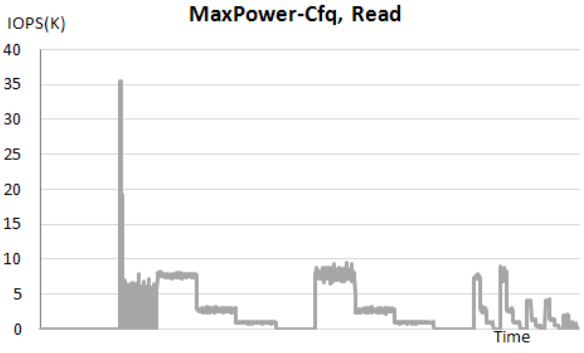
ioengine	libaio
filename	/sdb, /sdc/, mount point
randrepeat	0
rw	randwrite
rwmixwrite	25, 50, 75
iodepth	16, 32
overwrite	1

2. 실험 결과 분석

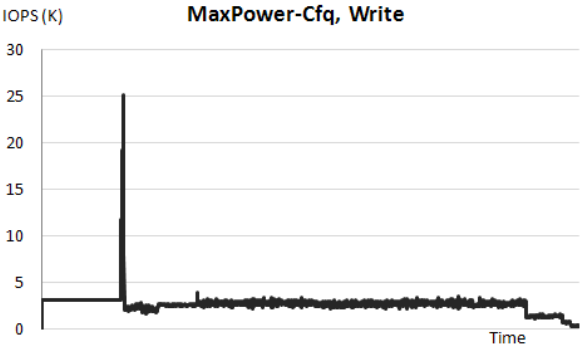
1) Block Path(I/O 스케줄러 + 링크 전력)

블록 경로에 대한 성능은 MaxPower-Noop이 가장 우수하였으나 수명관점에서는 거의 동등하였다. 이는 호스트 쓰기 총량이 I/O 스케줄러와 링크 전력의 상호 작용에 의해 성능과 비례 관계를 보이지 않았기 때문이다. 예를 들어, 그림 6의 결과와 같이 MaxPower-Cfq의 성능은 Noop보다 작고, Dead보다 크지만, WAF는 두 방식에 비해 높다.

그림 5에서, 블록 경로의 IOPS 경향성은 읽기의 경우 데이터 전송 크기와 SSD 내부 블록 크기와 동일하게 4 KB로 정렬되었을 때 월등하였고, 임의 읽기 비율에 따라 선형적으로 비례한 반면, 쓰기 IOPS는 읽기와 다르게 지속(Sustain) 성능을 보여 블록 크기, Queue의 깊이, 임의 쓰기의 비율에 따른 변화가 미미하였다. 이는 워크로드의 데이터 전송 크기가 SSD 내부의 블록



(a)



(b)

그림 5. MaxPower-Cfq 블록 경로의 IOPS 경향성
(a) 읽기 IOPS, (b) 쓰기 IOPS
Fig. 5. IOPS Result with respect to MaxPower-Cfq in block path, (a) Read IOPS, (b) Write IOPS.

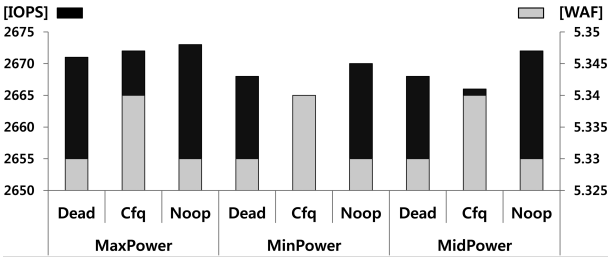


그림 6. 블록 경로 WAF 결과
Fig. 6. WAF Result with respect to block path.

크기와 다르고, 임의 쓰기로 전송되어, 가비지 콜렉션, 데이터 컴팩션 같은 FTL 동작이 지속 발생되었기 때문이다. 따라서 블록 경로는 제조사에서 FTL의 효율성을 평가할 수 있기 때문에 SSD의 WAF를 측정하는 한 방법으로 삼성에서도 WAF 관점의 평가에 적용하고 있다. 블록 경로에서는 CFQ 스케줄링 방식이 가장 스트레스를 주는 인자이기에, 리눅스 환경에서 WAF 평가 시 CFQ 방식의 사용은 수명관점의 가혹 조건 평가로 활용도가 크다.

2) File Path (F.S + I/O 스케줄러 + 링크 전력)

파일 경로에 대한 성능은 표 2와 같이 MaxPower-Cfq-Ext4이 가장 우수하였으며 수명은 MinPower-Dead-Xfs 조합이 우수하였다. 파일 시스템 관점에서는 Xfs-Ext3-Ext4 순으로 수명이 우수하였다. Ext3는 Ext4와 달리 블록 매핑에 따른 단편화와 메타데이터 쓰기가 많아져 성능에 비해 호스트 쓰기 총량이 증가하므로 WAF 관점에서는 열세가 되어야 하나, SSD P/E 사이클 소모는 적어 Ext4에 비해 WAF는 우수하였다. 특히 Cfq-Ext4 조합은 P/E 사이클을 2 배 이상 소모하는데, Cfq 스케줄링은 Ext4의 데이터 처리 방식과 SSD의 매핑 방식에 있어 수명 관점에서 중복되는 원인을 제공하는 것으로 추정된다. 이러한 특징은 호스트 I/O 스택을 고려하여 SSD의 내부 FTL 최적화 포인트를 개선할 수 있도록 추가 실험 및 관련 연구가 요구된다.

스케줄러 관점에서 Dead의 호스트 쓰기 총량은 타 스케줄러의 경향과 달리 MinPower가 MaxPower보다 많았다. MaxPower-Dead는 MinPower-Dead보다 성능이 우세하였으나, 호스트 쓰기 총량이 적어 결과적으로 WAF가 열세하였다. 이는 흥미로운 실험 결과로 수명 관점에서는 Dead 사용 시 Minpower를 사용해야한다.

표 2. 성능, WAF 종합 비교
Table 2. Comparison result in terms of Performance and WAF.

		Host GB	P/E cycle	성능(MB/s)	WAF
Min-Dead	Ext3	3767	12	117.29	1.630
	Ext4	3748	14	118.87	1.912
	XFS	3735	11	118.59	1.507
Min-Cfq	Ext3	3758	12	119.27	1.634
	Ext4	3749	29	119.59	3.9603
	XFS	3727	11	114.93	1.510
Min-Noop	Ext3	3762	12	118.04	1.632
	Ext4	3739	13	118.12	1.779
	XFS	3731	11	115.71	1.509

		Host GB	P/E cycle	성능(MB/s)	WAF
Max-Dead	Ext3	3766	12	121.51	1.631
	Ext4	3746	13	119.46	1.776
	XFS	3731	11	126.94	1.509
Max-Cfq	Ext3	3781	12	120.15	1.634
	Ext4	3749	29	137.41	3.9602
	XFS	3716	11	122.86	1.508
Max-Noop	Ext3	3769	12	118.48	1.630
	Ext4	3744	14	120.63	1.914
	XFS	3734	11	116.80	1.508

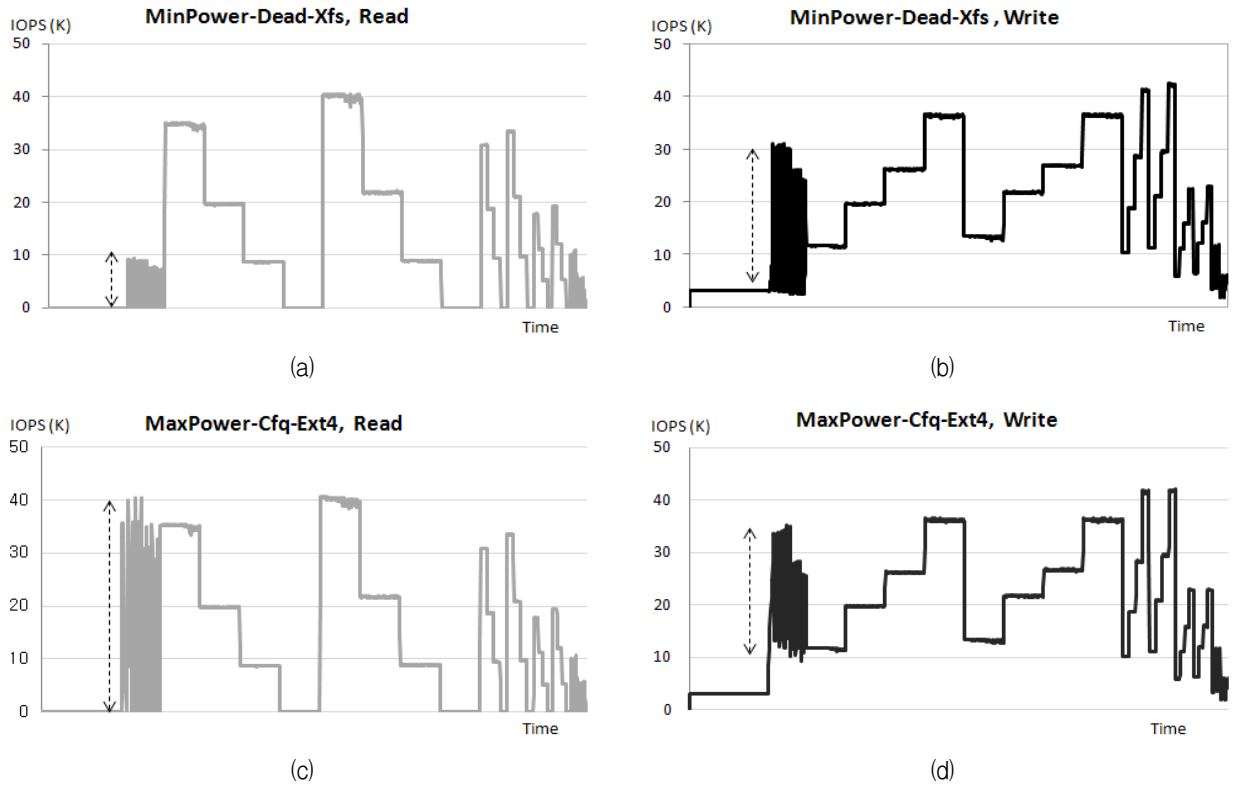


그림 7. 파일 경로의 IOPS 경향성, (a) MinPower-Dead-Xfs의 읽기 IOPS, (b) MinPower-Dead-Xfs의 쓰기 IOPS, (c) MaxPower-Cfq-Ext4의 읽기 IOPS, (d) MaxPower-Cfq-Ext4의 쓰기 IOPS
Fig. 7. IOPS Result with respect to file path, (a) Read IOPS of MinPower-Dead-Xfs, (b) Write IOPS of MinPower-Dead-Xfs, (c) Read IOPS of MaxPower-Cfq-Ext4, (d) Write IOPS of MaxPower-Cfq-Ext4.

그럼에도 불구하고 그림 8의 결과처럼, MaxPower-Noop-Ext4 조합이 MinPower-Noop-Ext4 보다 WAF 값이 큰 이유는 호스트 쓰기 총량이 P/E 사이클의 경계 범위를 근소하게 넘어 P/E 사이클이 증가되었기 때문이다. 즉, 호스트 쓰기 총량의 4 GB 차이가 P/E 사이클 증가를 1 만큼 발생시켜, WAF가 크게 계산되었다. 따라서 WAF 판단 시 P/E 사이클에 대한 경계 값을 초과했는지 이를 고려해야 한다.

파일 경로의 IOPS의 경향성은 블록 경로의 결과에 비해 약 4 배의 IOPS가 증가 되었는데, 이는 파일 시스

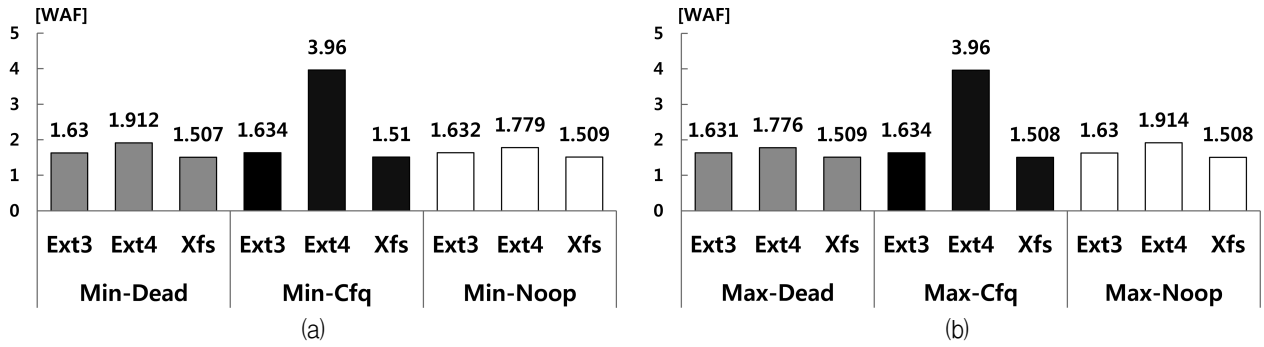


그림 8. 파일 경로 WAF 결과, (a) MinPower 조합, (b) MaxPower 조합
Fig. 8. WAF Result with respect to file path, (a) MinPower combination, (b) MaxPower combination.

템이 데이터 전송을 위해 파일을 조직하고 관리하는데 관여했기 때문이다. 따라서 호스트 I/O 스택에서 성능과 관련되어 파일시스템이 차자하는 비중이 높다. 특히, Cfq-Ext4 조합은 블록 크기가 4 KB 미만인 워크로드에서 그림 7의 화살표 크기 차이로 알 수 있듯이 타 조합 보다 IOPS가 월등하여 4KB 미만의 워크로드에서는 Cfq-Ext4 조합을 사용하는 것이 성능관점에서는 가장 유리하다. 파일 경로는 SSD 제조사가 TBW를 보증하기위한 항목인 UBER(Uncorrectable Bit Error Rate), FFR(Functional Failure Requirement) 테스트 시 블록 경로보다 4 배 이상의 시간을 단축할 수 있어 리눅스 환경에서 수명 평가 방법으로 활용가치가 높다.

I/O 스택의 구성에 따라 성능과 수명은 차이가 있으며, 최대 성능조합 MaxPower-Cfq-Ext4는 최소 성능조합 MinPower-Cfq-Xfs에 비해 16% 우수한 성능을 보였다. 최장 수명 조합 MinPower-Dead-Xfs는 최단 수명 조합 MinPower-Cfq-Ext4 보다 2.6 배 수명 연장이 가능하였다. 본 논문에서 제안하는 수명 관점의 I/O 스택 최적 구성은 MinPower-Dead-Xfs이며, 최고 성능 조합 MaxPower-Cfq-Ext4에 비해 13 %의 성능 하락을 감수하면, 최대 2.6 배의 수명 시간을 보장할 수 있다. 이는 I/O 스택 최적화 구성에 있어, SSD 성능뿐만 아니라 수명 관점의 고려에 대한 유의미를 입증한다.

Ⅵ. 결 론

본 논문에서는 낸드 플래시 셀의 미세화와 다중 비트화로 기본적으로 취약한 쓰기 내구성이 점점 더 악화됨에 따라 낸드플래시 기반 SSD 수명을 리눅스 기반 I/O 스택 관점에서 고찰하였다. SSD 수명과 신뢰성에 대한 영향이 가장 큰 분야인 데이터센터 환경에 대해 수명시험 워크로드를 바탕으로 블록경로와 파일 경로를 구성하여 분석과 모의 실험을 통한 검증을 수행하였고, 이를 통해 리눅스 I/O 스택을 수명관점에서 효율적으로 사용하기 위한 최적 구성으로 MinPower-Dead-Xfs를 도출하였다. 이는 최고 성능 조합 MaxPower-Cfq-Ext4에 비해 13% 성능 저하가 있지만, 2.6 배의 수명 시간 연장됨을 확인할 수 있었다. I/O 스택은 그 구성에 따라 SSD의 성능과 수명의 결과가 상이하여 I/O 스택구성 시 성능뿐만 아니라 수명에 대한 고려가 필요하며, 본 논문을 통해 수명관점의 고려에 대한 유의미를 입증하였다.

향 후 연구 방안으로 데이터 센터향으로 PCIe 기반의 NVMe 프로토콜 SSD가 개발되고 있어 이에 대한 I/O 스택의 추가 연구와 I/O 스택을 SSD Endurance 평가에 접목한 수명 Testing 방안을 모색하고자 한다.

REFERENCES

[1] M. Goldman, K. Pangal, G. Naso, and A. Goda, “25nm 64Gb 130mm2 3bpc NAND Flash Memory,” in Proceedings of the International Memory Workshop, pp. 1-4, May 2011.
[2] Y. J. Woo and J. S. Kim, “Diversifying Wear Index for MLC NAND Flash Memory to Extend the Lifetime of SSDs,” in Proceedings of the International Conference on Embedded Software, pp. 1-10, October 2013.
[3] S. W. Lee, D. J. Park, T. S. Chung, D. H. Lee, S. W. Park, and H. J. Song, “A Log Buffer-based Flash Translation Layer Using Fully Associative Sector Translation,” ACM

Transactions on Embedded Computing Systems, vol. 6, no. 3, July 2007.
[4] F. Chen, T. Luo, and X. D. Zhang, “CAFTL: A Content-aware Flash Translation Layer Enhancing the Lifespan of Flash Memory Based Solid State Drives,” In Proceedings of the 9th USENIX Conference on File and Storage Technologies, p. 6, February 2011.
[5] Y. J. Park and J. S. Kim, “Compression Support for Flash Translation Layer,” in Proceedings of the International Workshop on Software Support for Portable Storage, pp. 19-24, August 2010.
[6] G. Soundararajan, V. Prabhakaran, M. Balakrishnan, and T. Wobber, “Extending SSD Lifetimes with Disk-Based Write Caches,” In Proceedings of the 8th USENIX Conference on File and Storage Technologies, p. 8, February 2010.
[7] W. Shin, Q. C. Chen, M. W. Oh, H. S. Eom, and H. Y. Yeom, “OS IO Path Optimizations for Flash Solid-state Drives”, In Proceedings of USENIX Annual Technical Conference, pp. 483-488, June 2014.
[8] K. Zhou, P. Huang, C. H. Li, and H. Wang, “An Empirical Study on the Interplay Between Filesystems and SSD,” in Proceedings of IEEE 7th International Conference on Networking, Architecture and Storage, pp. 124-133, June 2012.
[9] H. Sun, X. Qin, and C. S. Xie, “Exploring Optimal Combination of a File System and an I/O Scheduler for Underlying Solid State Disks,” Journal of Zhejiang University Science C, vol. 15, issue. 8. pp. 607-621, August, 2014.
[10] P. Sehgal, V. Tarasov, and E. Zadok, “Evaluating Performance and Energy in File System Server Workload,” In Proceedings of 8th USENIX Conference on File and Storage Technologies, pp. 19-19, February 2010.
[11] H. Cook, J. Ellithorpe, L. Keys, and A. Waterman “IotaFS: Exploring File System Optimizations for SSDs,” University of California at Berkeley, 2008.
[12] Technical Committee T13 AT Attachment, http://www.t13.org.
[13] http://en.wikipedia.org/wiki/I/O_scheduling
[14] Active-State Power Management, http://access.redhat.com.
[15] A. Foong, B. Veal, and F. Hady. “Towards SSD-Ready Enterprise Platforms,” In Proceedings of 36th the International Conference on Very Large Data Bases, September 2010.
[16] http://www.jedec.org/sites/default/files/docs/JESD218A.pdf
[17] Vasudevan, Venkatesh, Hanmant Belgal, and Neal Mielke. “Verification and Management of Endurance in NAND SSDs.” Flash Memory Summit, August, 2012.

저 자 소 개

정 남 기(학생회원)

2006년 동국대학교 정보통신공학 학사 졸업.
2006년 3월～현재 삼성전자 반도체사업부 책임연구원.
2014년 3월～현재 성균관대학교 반도체디스플레이공학과 석사과정.

<주관심분야 : 메모리/스토리지 시스템 구조>

한 태 희(평생회원)

1992년 KAIST 전기 및 전자공학과 학사 졸업.
1994년 KAIST 전기 및 전자공학과 석사 졸업.
1999년 KAIST 전기 및 전자공학과 박사 졸업.
1999년 3월～2006년 8월 삼성 전자 통신연구소 책임 연구원.
2006년 9월～2008년 2월 한국산업기술대학교 전자공학과 조교수.
2008년 3월～현재 성균관대학교 정보통신대학 반도체시스템공학과 부교수.
2011년 5월～2013년 4월 지식경제부 시스템반도체 PD.

<주관심분야 : SoC 아키텍처 및 설계 방법론,3D IC, 메모리/스토리지 시스템 구조, 임베디드 SW, IT 융합기술>