Open Channel SSD 플랫폼에서 쓰기 버퍼 및 스레드 구성에 따른 성능 분석

임희락
서울대학교 컴퓨터공학부
rockylim@snu.ac.kr

Performance Analysis Based on Write Buffer and Thread Configuration in Open Channel SSD Platforms

Heerak Lim
Seoul National University

요 약

Open Channel SSD는 스토리지 디바이스에 FTL(Flash Translation Layer)을 구현하지 않고, 운영체제에게 SSD의 관리를 맡기는 SSD이다. 따라서 리눅스에서는 LightNVM과 같은 추상화 계층을 제공한다. pblk(The Physical Block Device)은 LightNVM Layer에 위치하는 커널 모듈로서 기존의 SSD의 FTL에서 수행하는 기능들을 호스트에서 수행한다. 본 논문에서는 Open Channel SSD에서 쓰기 요청의 처리 과정을 보이고, pblk에 구현되어 있는 소프트웨어 버퍼인 쓰기 버퍼(Write Buffer) 및 입출력 요청의 스레드 구성에 따른 성능 분석 결과를 보인다.

1. Introduction

향후 수년 내에 Solid-State Drive (SSD)는 지배적인 보조기억장치가 될 것으로 예상된다. SSD는 기존의 전통적인 Hard Disk Drive (HDD)에 비해서 우수한 성능을 보이지만, 스토리지 디바이스에 최적화 부족으로 인한 자원의 비효율 적인 이용 문제[4], long tail-latency, unpredictable I/O latency와 같은 단점들을 갖는다 [1, 2, 3]. 이러한 문제점들은 대부분 Hard Disk Drive에 최적화 된 Block I/O Interface 때문이다 [5].

Open Channel SSD는 위와 같은 문제점들을 해결할 수 있는 새로운 형태의 SSD 플랫폼이다. Open Channel SSD는 그 내부 Geometry를 호스트 운영체제에 드러내고, 호스트가 스토리지 디바이스내부의 물리적인 데이터 배치나 I/O 스케줄링을 관리할 수 있게 한다. 이렇게 함으로서, 호스트와 SSD 컨트롤러는 SSD 디바이스 작동과 관련된 기능을 나누어 수행한다 [3]. 기존의 SSD의 FTL Layer에서 수행하던 address translation, garbage collection, error handling 과 같은 기능들이 호스트에서 수행될 수 있다. 따라서 시스템에 따라 스토리지 소프트웨어 스택을 Open Channel SSD를 사용하는 응용 프로그램에 알맞게 재 구성할 수 있다. 리눅스 커널 4.4 이후부터 호스트 기반의 SSD 관리 서브시스템인 LightNVM 계층이 사용되었고, 리눅스 커널 4.12 이후부터 Open Channel SSD를 위한 host-side FLT(Flash Translation Layer)인 pblk이 커널에 포함되었다. 따라서 Open Channel SSD를 하나의 block device 로 호스트에게 노출되어 보여지고, 사용자는 SSD를 특정 워크로드 패턴에 맞게 최적화 할 수 있다.

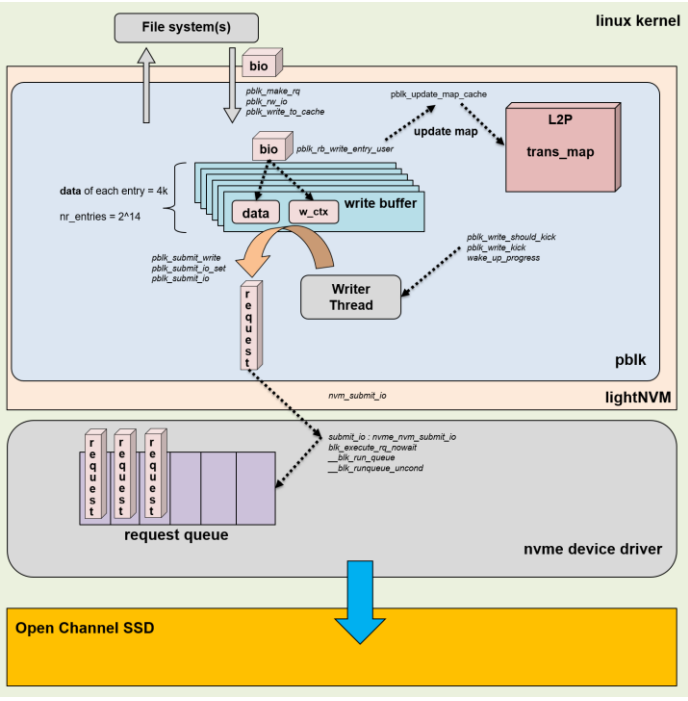


그림 1 – LightNVM 구조 및 쓰기 요청 진행 과정

2. Write path

어플리케이션으로부터 온 쓰기 요청은 <그림 1>과 같이 Lightnvm및 pblk의 abstraction layer를 거쳐 nvme device driver를 통해 Open Channel SSD 디바이스 에서 처리된다.

상위 Layer에서 VFS을 거쳐 block I/O 의 형태로 전달되는 I/O 요청은 write buffer에 data 및 metadata(write context)를 집어넣고 writer thread를 활성화시킨다. 하나의 writer thread I/O에 의해 스케줄링 되며, write buffer에 충분한 데이터가 모아졌을 경우, write request를 만들어 request queue에 집어넣는다. Request queue는 디바이스 드라이버에 의해서 처리된다.

3. Write Buffer

Write buffer는 호스트에서 정의된 sector 크기가 flash page 크기보다 작을 경우, flash page의 크기만큼 모아서 처리하기 위한 버퍼로서 작용한다[3]. 또한 write buffer에 데이터가 아직 있을 경우 Read시 write buffer에서 바로 캐시로서 접근 할 수 있다. 기존의 디바이스의 캐시가 호스트쪽에 위치하게 됨으로써 호스트-디바이스 간 path가 짧아지고, 캐시 히트 시, acknowledge를 바로 받을 수 있다는 장점이 있다.

write buffer는 pblk모듈에서 ring buffer의 형태로 구현된 소프트웨어 버퍼이다. I/O request를 담는 여러 개의 entry들로 구성되어 있으며, 각 entry는 I/O request를 구성하는 데이터와 메타데이터를 포함한다. Write buffer의 크기는 기본적으로 디바이스의 geometry에 의해서 결정되며, 그 크기를 결정하는 요소는, 페이지 당 섹터 개수, 디바이스의 plane 개수, LUN(=PU, Open Channel SSD에서 병렬적으로 동시에 처리할 수 있는 단위)의 개수 등에 의해서 결정된다.

본 논문에서는 이 write buffer의 크기와 입출력 요청의 병렬성에 따른 성능을 확인하기 위한 실험 및 분석 결과를 보인다.

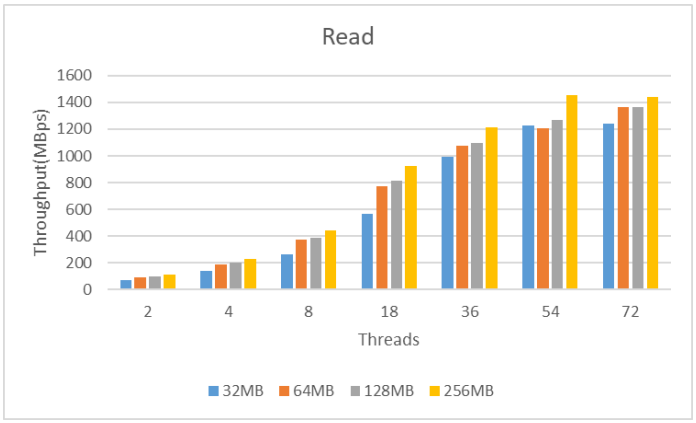
Controller	CNEX Labs Westlake ASIC
Interface	NVMe, PCI-e Gen3x8
Channels	8 (128 total)
PUs per Channel	16
Channel Data Bandwidth	280MB/s
Page Size	16K
Planes	4
Blocks	1067
Block Size	256 pages size
Type	MLC

표 1 - Open Channel SSD 특징

4. Experimental Evaluation

본 논문에서 보이는 실험의 목적은 두 가지 이다. 첫 번째로, LightNVM 스택의 pblk모듈에 구현된 write buffer의 크기에 따른 입, 출력의 성능을 분석하는 것이다. 두 번째로, 병렬적인 입, 출력의 정도에 따른 성능 변화를 분석한다. 즉 여러 다중 코어, 다중 스레드 환경에서의 Open Channel SSD의 성능을 분석하는 것이다.

본 실험을 위해 72코어의 Intel Xeon E7-8870 프로세서 서버를 사용하였으며, 16Gib DRAM, PIC 3.0 인터페이스 및 CNEX Labs Westlake SDK(2TB NAND MLC Flash) Open Channel SSD를 사용하였다. Open Channel SSD의 상세한 특징은 <표 1>에 나타내었다. 호스트는 Ubuntu 16.04.3 LTS server를 사용하였고, pblk 모듈을 사용한 리눅스 커널 4.14.0-rc2 버전을 사용하였다. 실험을 위해 fio[6]를 사용하였다.



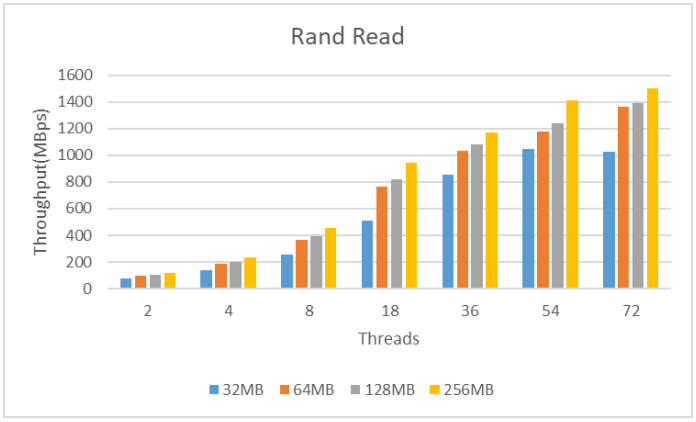


표 2 - write buffer 크기 및 스레드 수에 따른 읽기 처리량(Throughput, MB/s)

4.1 Read Request

<표 2>에 따르면 Open Channel SSD 읽기 요청 처리 성능은 스레드 수에 비례하여 점점 증가하다 어느 정도 병렬성의 정도가 증가하면 성능의 증가율이 감소하는

모습을 보인다. 특히, 읽기 요청 스레드가 54개에서 72개로 증가할 땐 성능의 변화가 거의 없었다.

Write buffer의 크기에 따른 성능은 스레드 개수에 상관 없이 모두 일정한 증가 비율을 나타냈는데, 이는 write buffer 크기가 증가함에 따라, 읽기 요청 시 버퍼 캐시 히트 비율이 증가했기 때문이다.

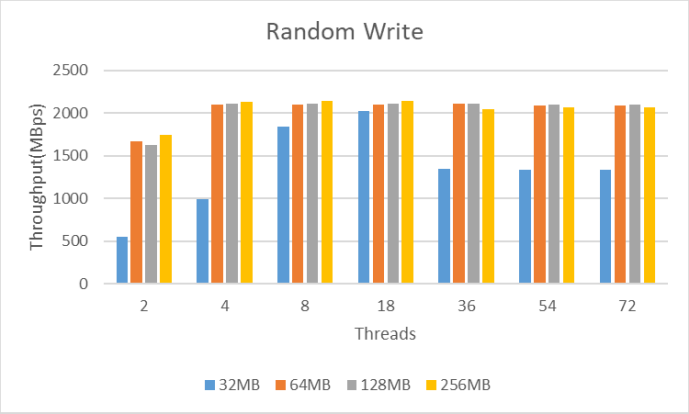


표 3 - write buffer 크기 및 스레드 수에 따른 쓰기 처리량(Throughput, MB/s)

4.2 Write

<표 3>은 스레드 및 write buffer크기에 따른 쓰기 요청 성능을 나타낸다. 64MB이상의 버퍼 크기에서 진행한 쓰기 요청에 대한 성능 실험결과는 스레드가 2개에서 4개로 증가할 때, 약간의 증가율을 보이지만 나머지 구간에서는 큰 증가율을 보이지 않는다. Write buffer의 크기가 32MB일 때에는 임의 쓰기 요청(random write request)시 스레드의 수에 따라 18개 스레드 수 까지는 비교적 큰 비율로 쓰기 성능이 증가한다.

Write buffer의 크기가 32MB일 때, 일반 쓰기와 임의 쓰기 모두 18쓰레드 이후 쓰기 성능이 18개의 스레드일 때와 비교하여 큰 폭으로 감소함을 보인다. 이는 작은 크기의 버퍼에 비해 많은 쓰기 스레드가 쓰기 요청을 하여, 항상 버퍼가 가득 차있는 상황을 나타내고, 더 이상 성능의 증가가 나타나지 않음을 나타내는 것으로 예상된다.

ACKNOWLEDGEMENT

본 연구는 2017년도 정부(과학기술정보통신부)의 재원으로 한국 연구재단 차세대정보·컴퓨팅기술 개발사업의 지원을 받아 수행됨 (No. 2015M3C4A7065646 & 2015R1A2A2A01005995)

5. 참고 문헌

[1] Hao, M., Soundararajan, G., Kenchammana Hosekote, D. R., Chien, A. A., & Gunawi, H. S. (2016, February). The Tail at Store: A Revelation from Millions of Hours of Disk and SSD Deployments. In FAST (pp. 263-276).

[2] Chen, F., Luo, T., & Zhang, X. (2011, February). CAFTL: A Content-Aware Flash Translation Layer Enhancing the Lifespan of Flash Memory based Solid State Drives. In FAST (Vol. 11, pp. 77-90).

[3] Bjørling, M., González, J., & Bonnet, P. (2017, February). LightNVM: The Linux Open-Channel SSD Subsystem. In FAST (pp. 359-374).

[4] Agrawal, N., Prabhakaran, V., Wobber, T., Davis, J. D., Manasse, M. S., & Panigrahy, R. (2008, June). Design Tradeoffs for SSD Performance. In USENIX Annual Technical Conference (Vol. 8, pp. 57-70).

[5] Swanson, S., & Caulfield, A. M. (2013). Refactor, reduce, recycle: Restructuring the i/o stack for the future of storage. Computer, 46(8), 52-59.

[6] AXBOE, J. Fio - Flexible I/O tester. URL http://freecode.com/projects/fio (2014).