Experimental Results of Implementing NV Me-based Open Channel SSDs

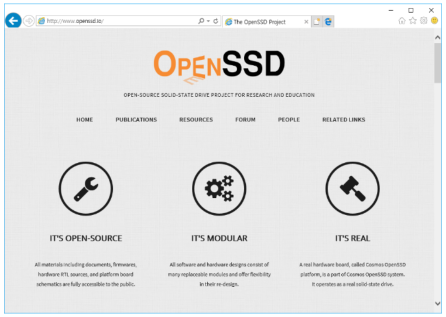
Sangjin Lee, Yong Ho Song

Hanyang University, Seoul, Korea

OpenSSD Project

- Open source SSD for search and education
  - Jasmine OpenSSD (2011)
  - Cosmos OpenSSD (2014)
  - Cosmos+ OpenSSD (2016)

- Cosmos+ OpenSSD (FMS 2016)
  - FPGA implementation of SSD ntroller hardware w/NVMe support
  - Can modify both SSD controller dware and firmware



OPEN-SOURCE SOLID-STATE DRIVE PROJECT FOR RESEARCH AND EDUCATION

IT'S OPEN-SOURCE

All materials including documents, firmwares, hardware RTL sources, and platform board schematics are fully accessible to the public.

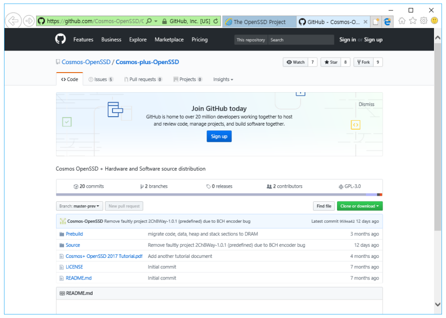
IT'S MODULAR

All software and hardware designs consist of many replaceable modules and offer flexibility in their re-design.

IT'S REAL

A real hardware board, called Cosmos OpenSSD platform, is a part of Cosmos OpenSSD system. It operates as a real solid-state drive.

www.openssd.io



Join GitHub today

GitHub is home to over 20 million developers working together to host and review code, manage projects, and build software together.

Cosmos OpenSSD + Hardware and Software source distribution

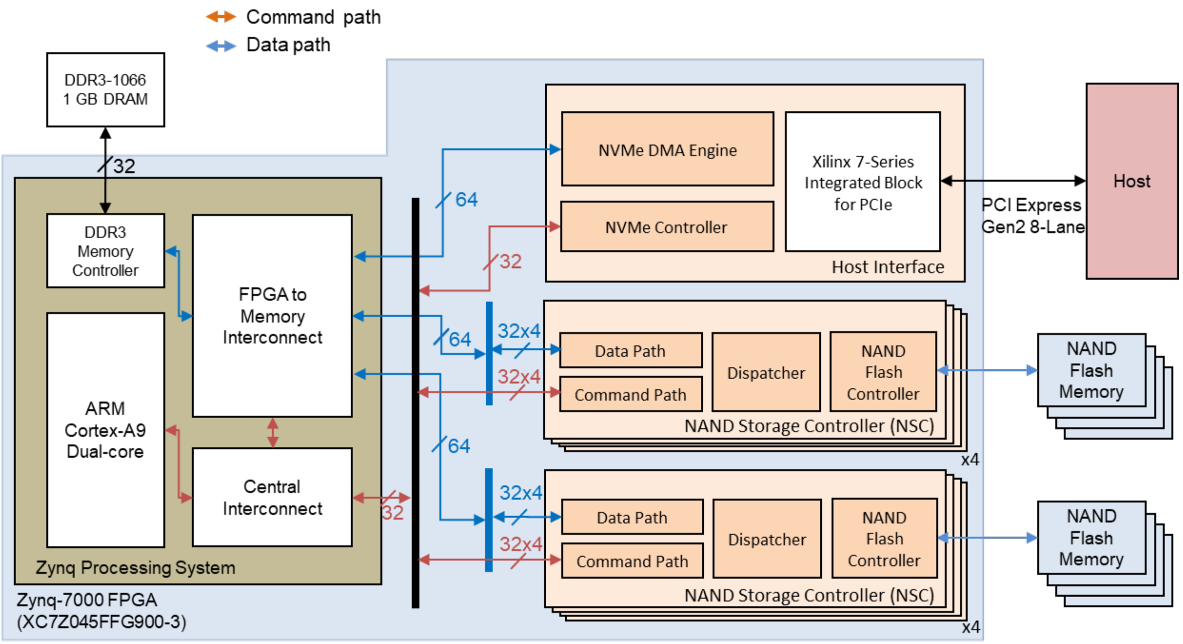
Cosmos-OpenSSD Remove faulty project 2Ch8Way-1.0.1 (predefined) due to BCH encoder bug		Latest commit 959ea42 12 days ago
Prebuild	migrate code, data, heap and stack sections to DRAM	3 months ago
Source	Remove faulty project 2Ch8Way-1.0.1 (predefined) due to BCH encoder bug	12 days ago
Cosmos+ OpenSSD 2017 Tutorial.pdf	Add another tutorial document	4 months ago
LICENSE	Initial commit	7 months ago
README.md	Initial commit	7 months ago

README.md

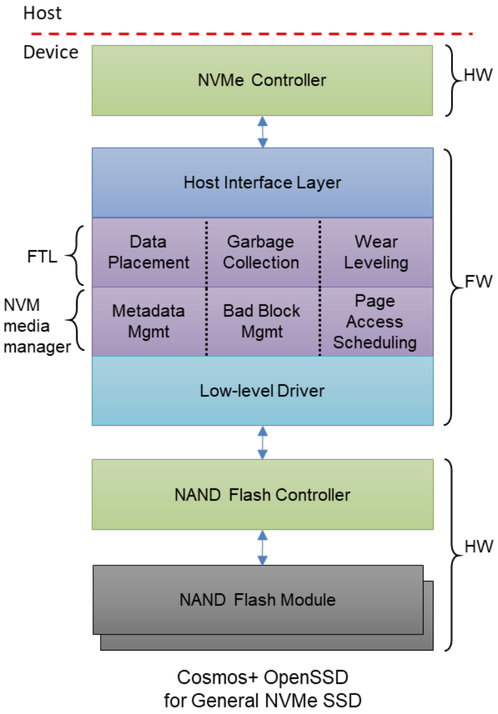
github.com/Cosmos-OpenSSD

Cosmos+ Storage Controller

Supporting up to 8 channels of NAND flash memories



Cosmos+ Component Layers



Cosmos+ OpenSSD
for General NVMe SSD

Fetches NVMe commands from host

Fetches NVMe commands from NVMe controller
Handles NVMe admin commands
Delivers NVMe NVM commands to FTL

Provides NAND flash controller API

Executes NAND commands to NAND flash module

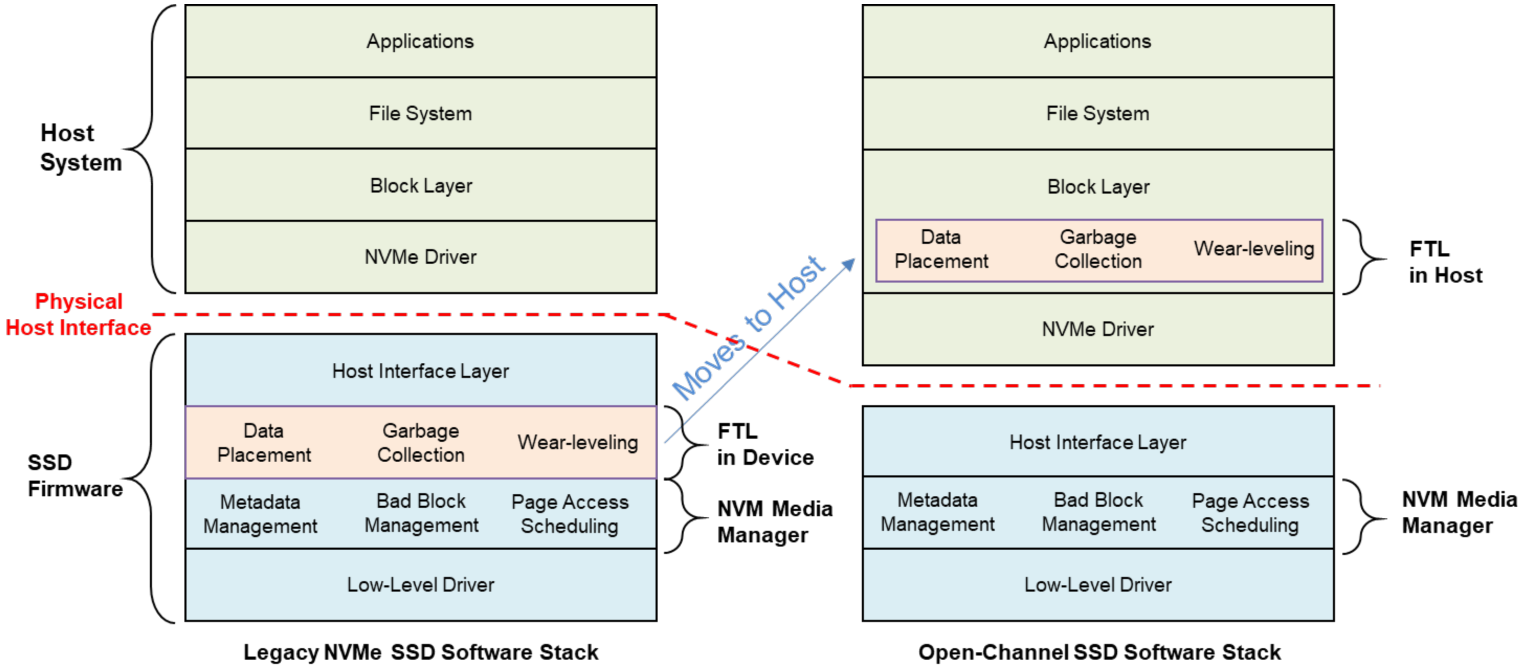
Open-Channel SSD (OCSSD)

WIKIPEDIA

a solid-state drive which does not have a firmware Flash Tr
anslation Layer implemented on the device,
but instead leaves the management of the physical solid-stat
e storage to the computer's operating system

- Moves FTL functions in storage device to host
  - Less operation loads on storage device
- Host-controlled I/O scheduling and data placement
  - Makes storage-specific policy for better performance

FTL Function Migration



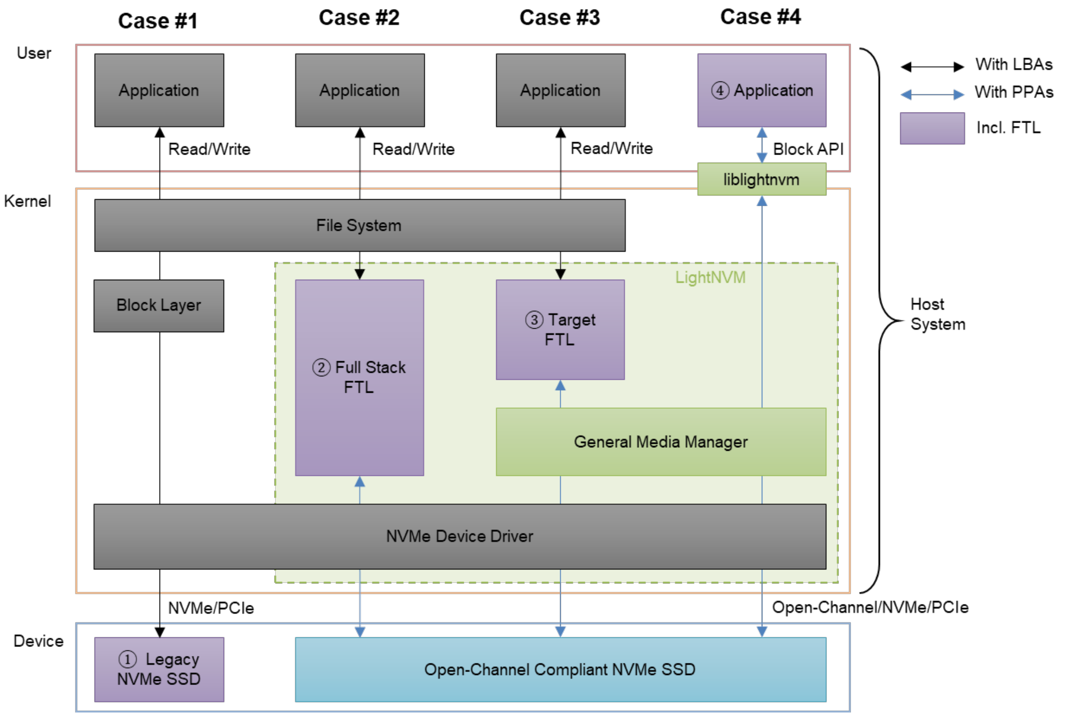
8 OCSSD Commands

4 mandatory commands and 4 optional commands

Open-Channel Command	Mandatory / Optional	Cmd Set	Description	Imple-mented
Device identification	M	Admin	Gets device and media information	Yes
Physical block erase	M	NVM	Erases target PPAs	Yes
Physical page address write	M	NVM	Writes data to target PPAs w/ device ECC engine	Yes
Physical page address read	M	NVM	Reads data from target PPAs w/ device ECC engine	Yes
Set bad blocks table	O	Admin	Sets bad block information	Yes
Get bad blocks table	O	Admin	Gets bad block information	Yes
Physical page address raw write	O	NVM	Writes data to target PPAs w/ host ECC engine	No
Physical page address raw read	O	NVM	Reads data from target PPAs w/ host ECC engine	No

Based on Open-Channel SSD 1.2 specification

4 FTL Implementation Options



Case #1

Case #2

Case #3

Case #4

User

Application

Application

Application

④ Application

Read/Write

Read/Write

Read/Write

Block API

liblightnvm

Kernel

File System

Block Layer

LightNVM

② Full Stack FTL

③ Target FTL

General Media Manager

NVMe Device Driver

NVMe/PCIe

Open-Channel/NVMe/PCIe

Device

① Legacy NVMe SSD

Open-Channel Compliant NVMe SSD

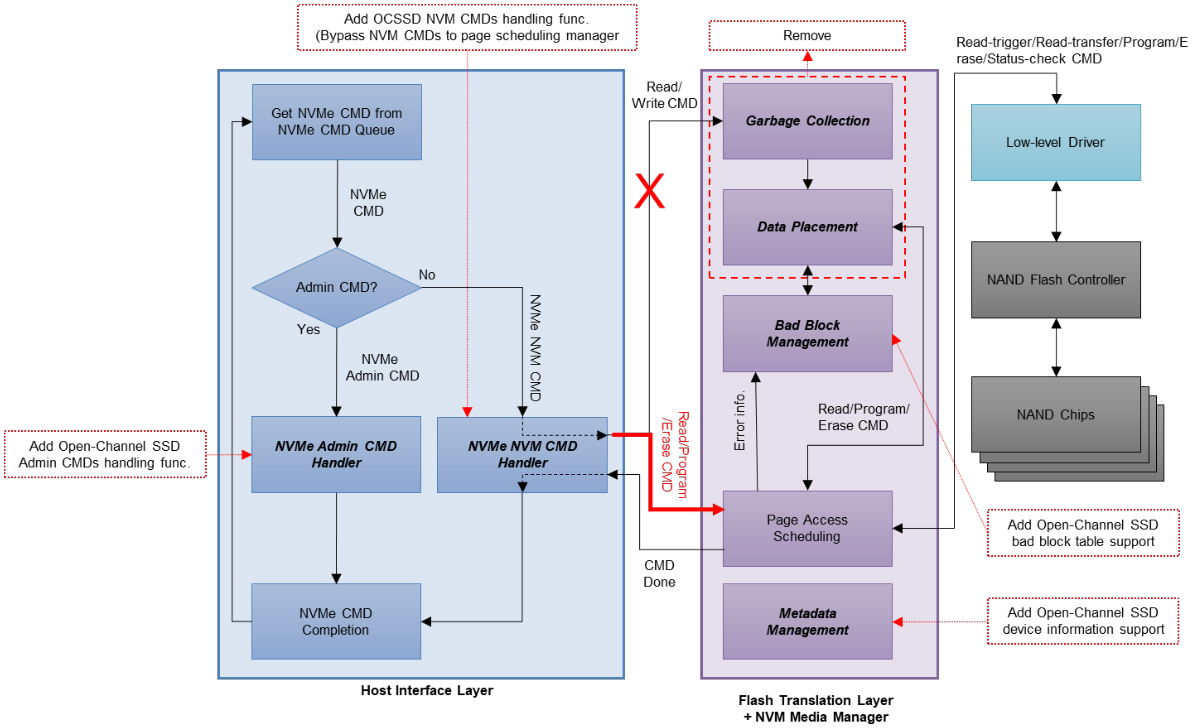
Host System

⟷ With LBAs

⟷ With PPAs

▨ Incl. FTL

Modification in Processing Flow



Evaluation Environment

Items		Descriptions
OCSSD Target FTL		pblk
Benchmark		ioping* (I/O latency measurement) (4K random, 128K sequential)
SSDs		1. Legacy NVMe SSD (Native Cosmos+) 2. OCSSD (Implemented on Cosmos+)
SSD platform board		Cosmos+ OpenSSD rev. 2.1**
FPGA bitstream		Prebuild 3.0.0*** (8-channel 8-way)
Device firmware	Legacy NVMe	GreedyFTL 2.7.0.c****
	OCSSD	Modified from GreedyFTL 2.7.0.c

* https://github.com/koct9i/ioping

** http://openssd.io/index.html

*** https://github.com/Cosmos-OpenSSD/Cosmos-plus-OpenSSD/tree/master-prev/Prebuild/Prebuild-3.0.0

**** https://github.com/Cosmos-OpenSSD/Cosmos-plus-OpenSSD/tree/master/Source/Firmware/GreedyFTL-2.7.0.c

Flash Memory Summit

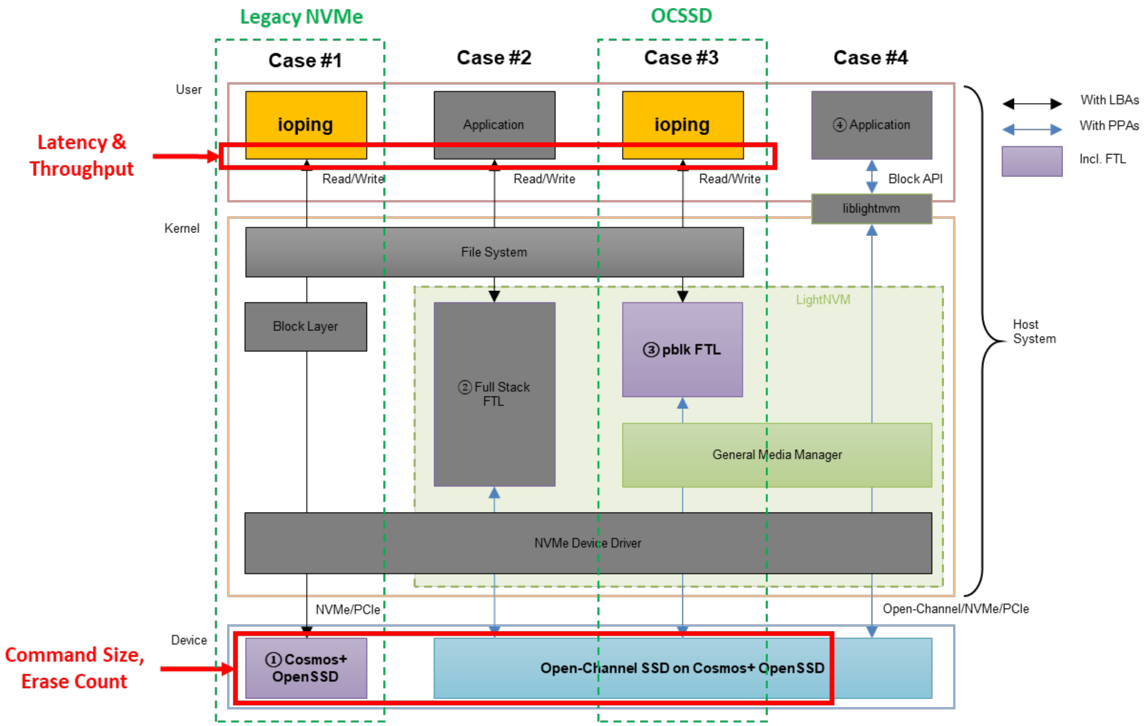
Information for Device Identification

About Open-Channel SSD	
Bad blocks table management	O
Hybrid command support	X
L2P map location	Host
ECC support	Storage
Multi-plane operation	O
Command suspension	X
Scramble on/off	X
Encryption	X

About NAND flash	
Flash media type	SLC
Number of channels	8
Number of ways per channel	8
Number of planes per way	2
Number of blocks per plane	Max. 4096*
Number of pages per block	128
Number of bytes in a page	16384
Sector size	4096

* Varies depending on a request size for faster evaluation

Some Performance Measurement

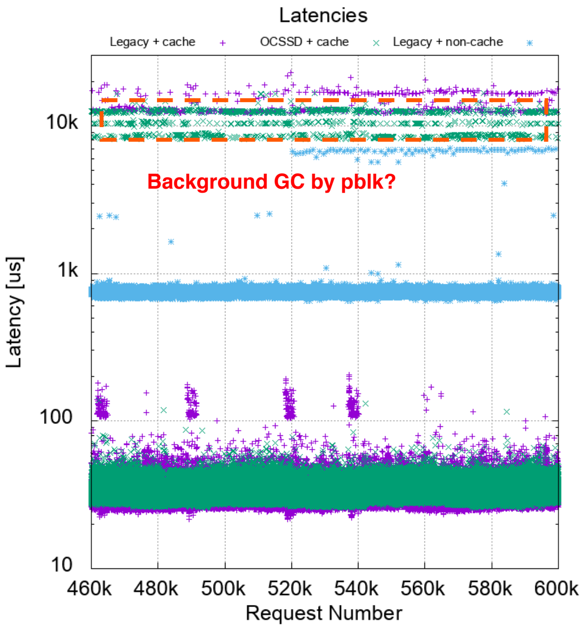


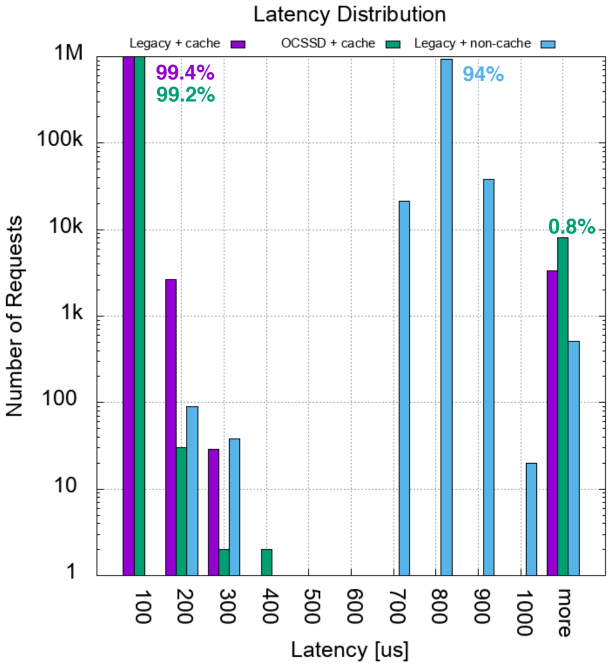
Latency & Throughput

Command Size, Erase Count

Latency Analysis - Legacy NVMe vs OCSSD 12 8K Seq. Write

- The latency of most commands with cache is less than 100 us



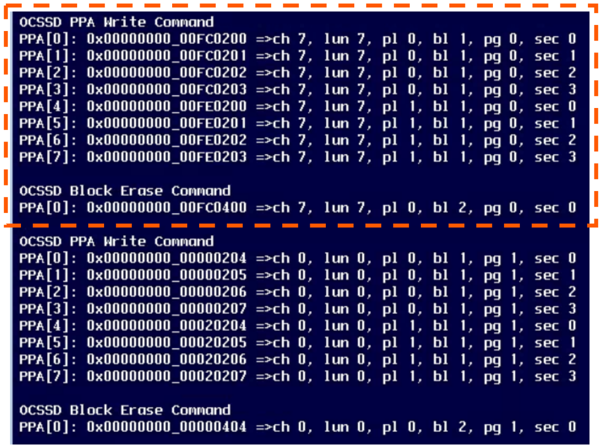


QD1, 1 M REQs, 64 GB NAND, 8-channel 8-way (64 LUNs)
There is no 'pblk + non-cache' data, because pblk with direct I/O occurs pblk's corruption on our environment.

Cmd. Size Distribution - Legacy NVMe vs OCSSD 128K Seq. Write

- OCSSD divides the write request to page write commands
- OCSSD issues an erase command after write command to next block of the same LUN

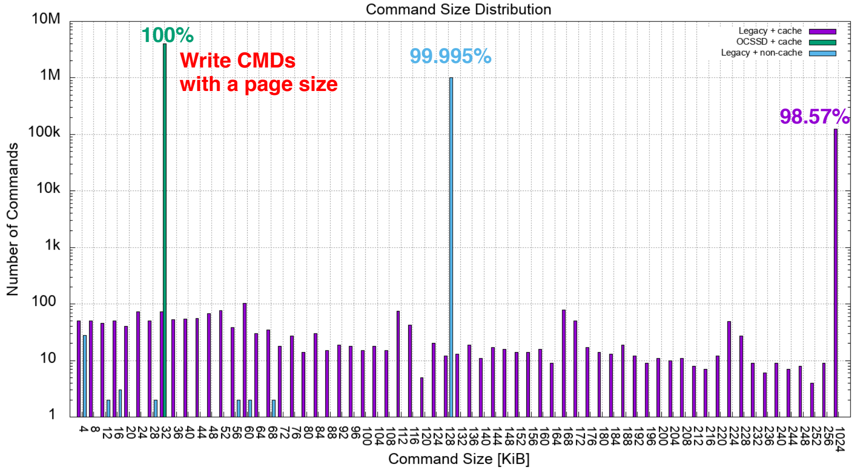
Same LUN



```
OCSSD PPA Write Command
PPA[0]: 0x00000000_00FC0200 =>ch 7, lun 7, pl 0, bl 1, pg 0, sec 0
PPA[1]: 0x00000000_00FC0201 =>ch 7, lun 7, pl 0, bl 1, pg 0, sec 1
PPA[2]: 0x00000000_00FC0202 =>ch 7, lun 7, pl 0, bl 1, pg 0, sec 2
PPA[3]: 0x00000000_00FC0203 =>ch 7, lun 7, pl 0, bl 1, pg 0, sec 3
PPA[4]: 0x00000000_00FE0200 =>ch 7, lun 7, pl 1, bl 1, pg 0, sec 0
PPA[5]: 0x00000000_00FE0201 =>ch 7, lun 7, pl 1, bl 1, pg 0, sec 1
PPA[6]: 0x00000000_00FE0202 =>ch 7, lun 7, pl 1, bl 1, pg 0, sec 2
PPA[7]: 0x00000000_00FE0203 =>ch 7, lun 7, pl 1, bl 1, pg 0, sec 3

OCSSD Block Erase Command
PPA[0]: 0x00000000_00FC0400 =>ch 7, lun 7, pl 0, bl 2, pg 0, sec 0
OCSSD PPA Write Command
PPA[0]: 0x00000000_00000204 =>ch 0, lun 0, pl 0, bl 1, pg 1, sec 0
PPA[1]: 0x00000000_00000205 =>ch 0, lun 0, pl 0, bl 1, pg 1, sec 1
PPA[2]: 0x00000000_00000206 =>ch 0, lun 0, pl 0, bl 1, pg 1, sec 2
PPA[3]: 0x00000000_00000207 =>ch 0, lun 0, pl 0, bl 1, pg 1, sec 3
PPA[4]: 0x00000000_00020204 =>ch 0, lun 0, pl 1, bl 1, pg 1, sec 0
PPA[5]: 0x00000000_00020205 =>ch 0, lun 0, pl 1, bl 1, pg 1, sec 1
PPA[6]: 0x00000000_00020206 =>ch 0, lun 0, pl 1, bl 1, pg 1, sec 2
PPA[7]: 0x00000000_00020207 =>ch 0, lun 0, pl 1, bl 1, pg 1, sec 3

OCSSD Block Erase Command
PPA[0]: 0x00000000_00000404 =>ch 0, lun 0, pl 0, bl 2, pg 1, sec 0
```


Command Size Distribution

100%

Write CMDs with a page size

99.995%

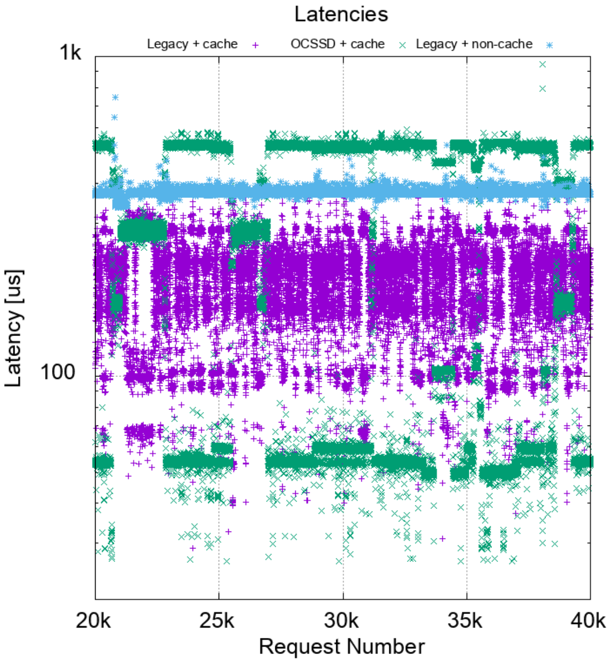
98.57%

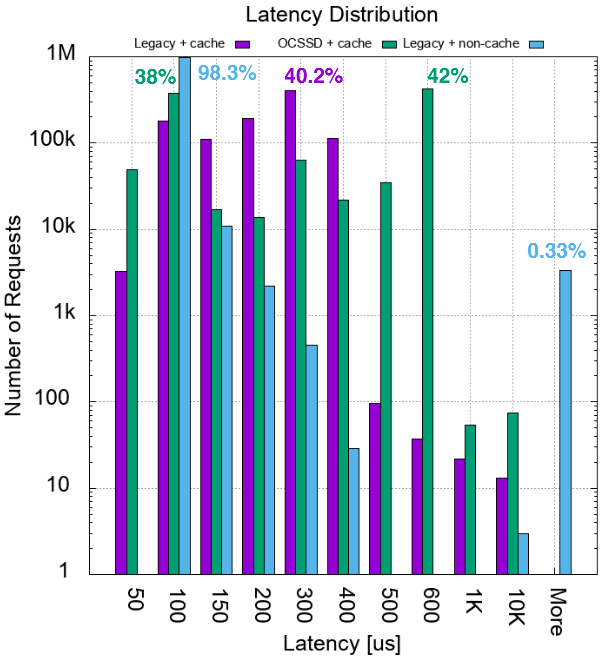
	Legacy + cache	OCSSD + cache	Legacy + non-cache
Block Erase Count	29952	31358	30016

QD1, 1 M REQs, 64 GB NAND, 8-channel 8-way (64 LUNs)

Latency Analysis - Legacy NVMe vs OCSSD 12 8K Seq. Read

- OCSSD shows high average latency in this experiment

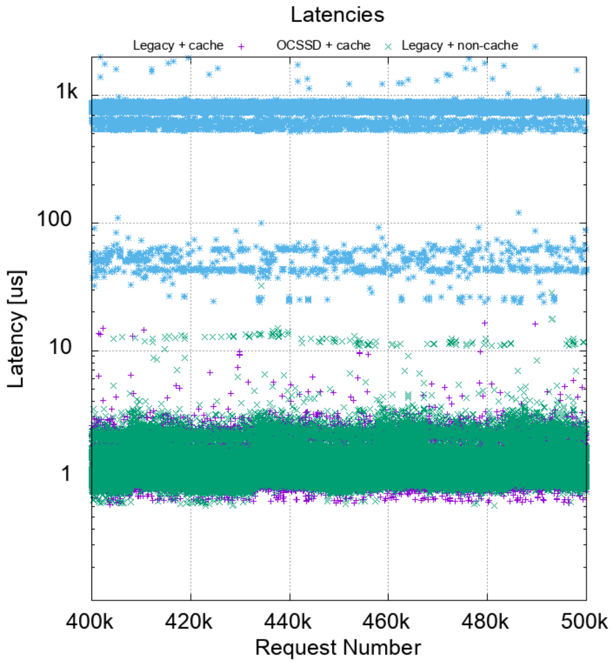


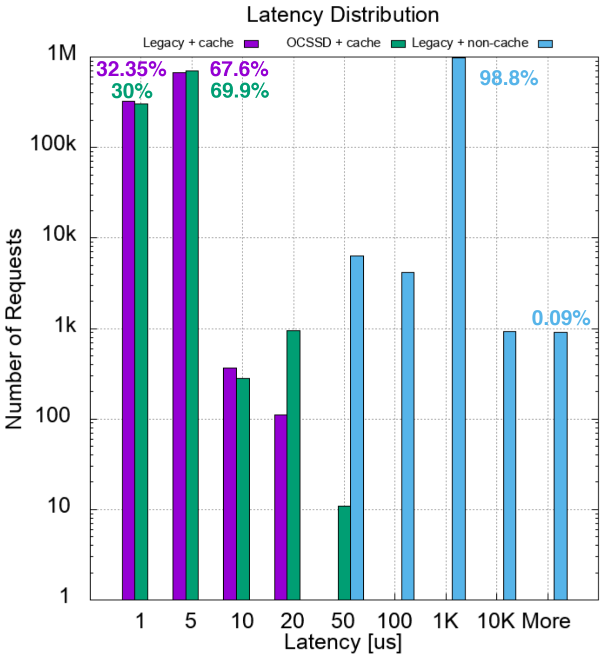


QD1, 1 M REQs, 64 GB NAND, 8-channel 8-way (64 LUNs)

Latency Analysis - Legacy NVMe vs OCSSD 4 K Rnd. Write

- OCSSD and legacy (with cache) shows similar result

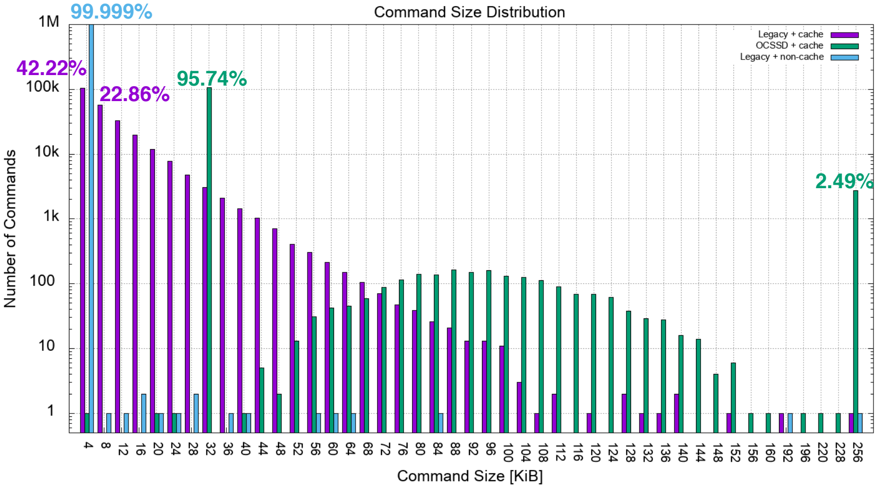




QD1, 1 M REQs, 2 GB NAND, 8-channel 8-way (64 LUNs)

Cmd. Size Distribution - Legacy NVMe vs OCSSD 4K Rnd. Write

- OCSSD combines the write request to page write commands
- OCSSD shows a lower block erase count compared to others

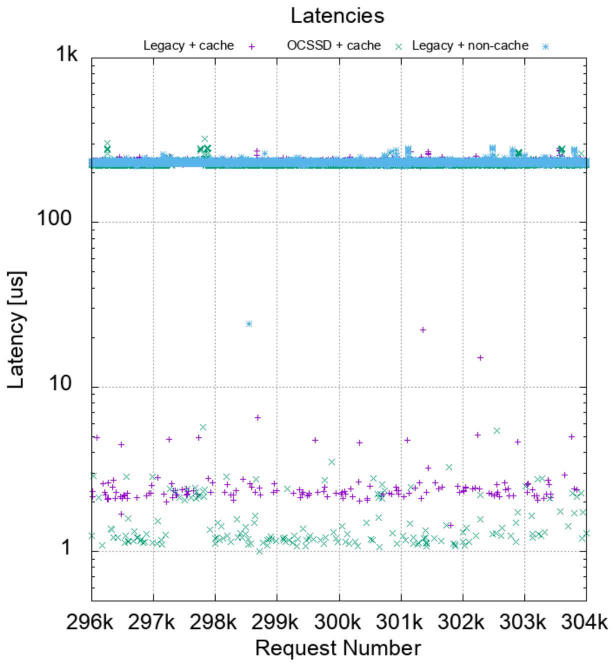


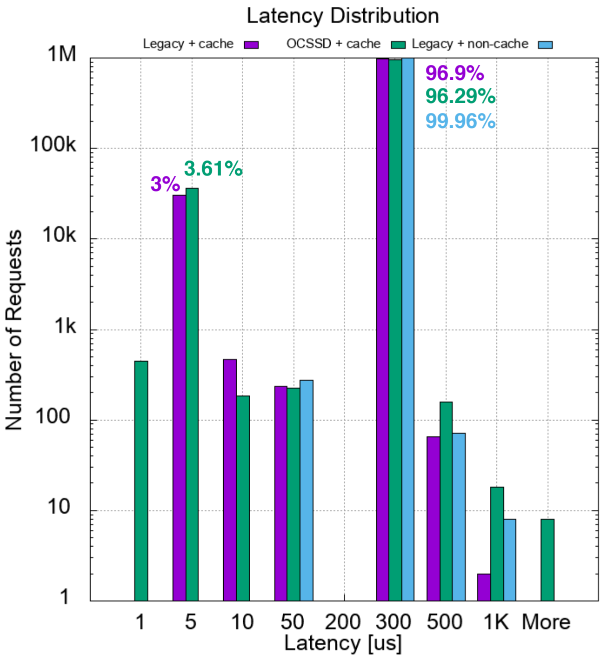
	Block Erase Count
Legacy + cache	4480
OCSSD + cache	**832**
Legacy + non-cache	57536

QD1, 1 M REQs, 2 GB NAND, 8-channel 8-way (64 LUNs)

Latency Analysis - Legacy NVMe vs OCSSD 4 K Rnd. Read

- OCSSD and legacy (with cache) shows similar result

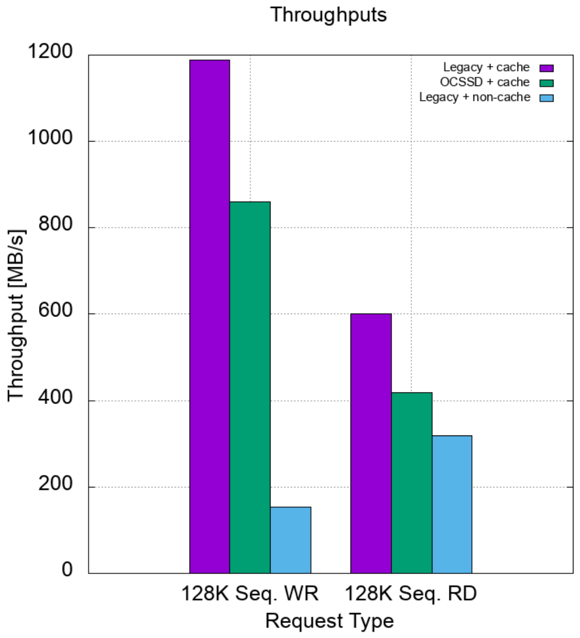


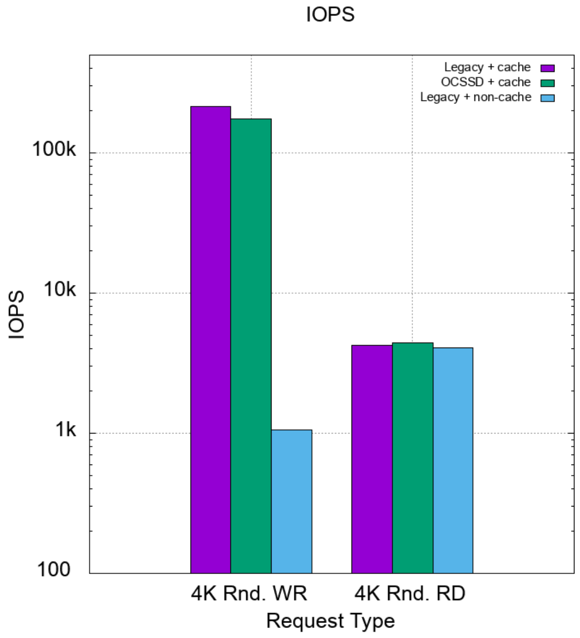


QD1, 1 M REQs, 64 GB NAND, 8-channel 8-way (64 LUNs)

Performance Benchmark - Legacy NVMe vs OCSSD

- Note that this is the result with QD1





QD1, 1 M REQs, 64 GB NAND, 8-channel 8-way (64 LUNs)

Summary

- OCSSD implementation on Cosmos+ OpenSSD system
  - Makes it possible to investigate internal operations and analyze performance
- From experiments, we observe that OCSSD
  - divides/combines write request(s) to page write command(s)
  - shows significantly low block erase count at 4K random write
  - shows high average latency at 128K sequential read
- Any collaboration is welcome

Thank you

For further information,

visit http://www.openssd.io!