Ceph BlueStore Performance

- with Intel® 3D NAND and Intel® Optane™ Technologies

Yuan Zhou < yuan.zhou@intel.com >

Nov, 2016

Agenda

Team Introduction

Ceph BlueStore Introduction

Ceph performance with all flash configuration

Ceph with Intel Optane™ SSD technology

Summary

Acknowledgements

This is a joint team work

Thanks for the contributions of Haodong Tang, Jianpeng Ma and Ning Li

Team introduction

- Intel SSG/STO/cloud and big data technology
- Global team, local focus
- Open source leadership @Spark, Hadoop, OpenStack, Ceph etc.
- Working closely with community and end customers
- Bridging advanced research and real-world applications

This slides only covers Ceph part

Drive the community grow

2016 Aug Ceph Day @ Beijing

2016 APAC Ceph road Show

Shanghai Big Data Streaming Meetup

ceph days

Shanghai Apache Spark Meetup

ampcamp china 2015

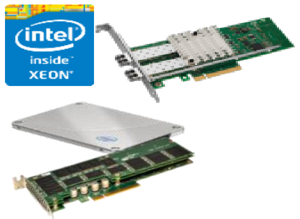
大数据金牌训练营

——AMPCamp来中国了!

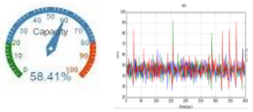




Ceph at Intel – Our 2016 Ceph Focus Areas



Optimize for Intel® platforms, flash and networking
- Compression, Encryption hardware offloads (QAT & SOCs)
- PMStore (for 3D XPoint DIMMs)
- RBD caching and Cache tiering with NVM
- IA optimized storage libraries to reduce latency (ISA-L, SPDK)



Performance profiling, analysis and community contributions
- All flash workload profiling and latency analysis
- Streaming, Database and Analytics workload driven optimizations

Ceph enterprise usages and hardening
- Manageability (Virtual Storage Manager)
- Multi Data Center clustering (e.g., async mirroring)

POCs

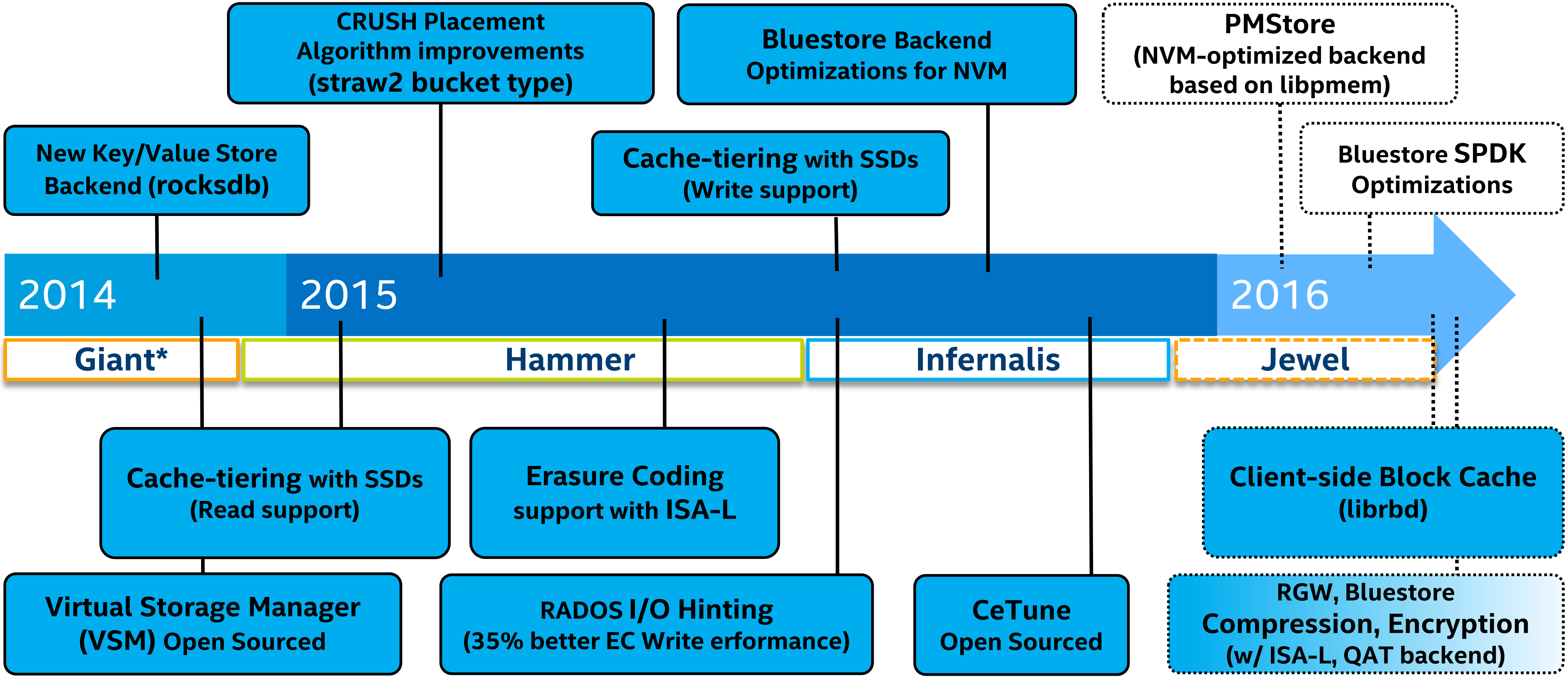
End Customer POCs with focus on broad industry influence
- CDN, Cloud DVR, Video Surveillance, Ceph Cloud Services, Analytics

Go to market

Ready to use IA, Intel NVM optimized systems & solutions from OEMs & ISVs
- Ready to use IA, Intel NVM optimized systems & solutions from OEMs & ISVs
- Intel system configurations, white papers, case studies
- Industry events coverage

Intel® Storage Acceleration Library (Intel® ISA-L)	Intel® Storage Performance Development Kit (Intel® SPDK)	Intel® Cache Acceleration Software (Intel® CAS)	Virtual Storage Manager	Ce-Tune Ceph Profiler

Development & Optimization – Intel Ceph Contributions

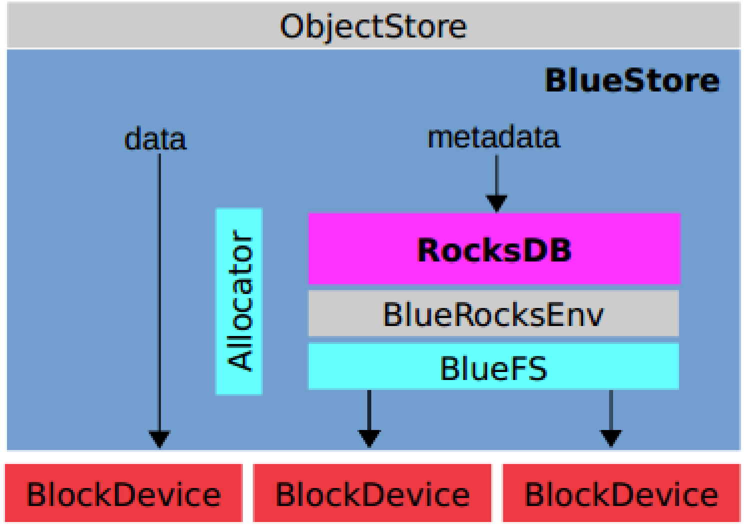


* Right Edge of box indicates approximate release date

Ceph BlueStore introduction

- BlueStore = **Block** + **NewStore**
  - consume raw block device(s)
  - key/value database (RocksDB) for metadata
  - data written directly to block device
  - pluggable block Allocator (policy)

- We must share the block device with RocksDB
  - implement our own rocksdb::Env
  - implement tiny “file system” BlueFS
  - make BlueStore and BlueFS share device(s)



http://www.slideshare.net/sageweil1/bluestore-a-new-faster-storage-backend-for-ceph

Suggested Configurations for Ceph* Storage Node

Standard/good (baseline):

Use cases/Applications: *that need high capacity storage with high throughput performance*

- NVMe*/PCIe* SSD for Journal + Caching, HDDs as OSD data drive
- Example: 1x 1.6TB Intel® SSD DC P3700 as Journal + Intel® Cache Acceleration Software (Intel® CAS) + 12 HDDs

Better IOPS

Use cases/Applications: *that need higher performance especially for throughput, IOPS and SLAs with medium storage capacity requirements*

- NVMe/PCIe SSD as Journal, no caching, High capacity SATA SSD for data drive
- Example: 1x 800GB Intel® SSD DC P3700 + 4 to 6x 1.6TB DC S3510

Best Performance

Use cases/Applications*: that need highest performance (throughput and IOPS) and low latency.*

- All NVMe/PCIe SSDs
- Example: 4 to 6 x 2TB Intel SSD DC P3700 Series

Ceph* storage node --Good	
CPU	Intel(R) Xeon(R) CPU E5-2650v3
Memory	64 GB
NIC	10GbE
Disks	1x 1.6TB P3700 + 12 x 4TB HDDs (1:12 ratio) P3700 as Journal and caching
Caching software	Intel(R) CAS 3.0, option: Intel(R) RSTe/MD4.3

Ceph* Storage node --Better	
CPU	Intel(R) Xeon(R) CPU E5-2690
Memory	128 GB
NIC	Duel 10GbE
Disks	1x Intel(R) DC P3700(800G) + 4x Intel(R) DC S3510 1.6TB

Ceph* Storage node --Best	
CPU	Intel(R) Xeon(R) CPU E5-2699v3
Memory	>= 128 GB
NIC	2x 40GbE, 4x Dual 10GbE
Disks	4 to 6 x Intel® DC P3700 2TB

More Information: https://intelassetlibrary.tagcmd.com/#assets/gallery/11492083/details

Ceph* on all-flash array

Storage providers are struggling to achieve the required high performance

- There is a growing trend for cloud providers to adopt SSD
  - CSP who wants to build EBS alike service for their OpenStack* based public/private cloud

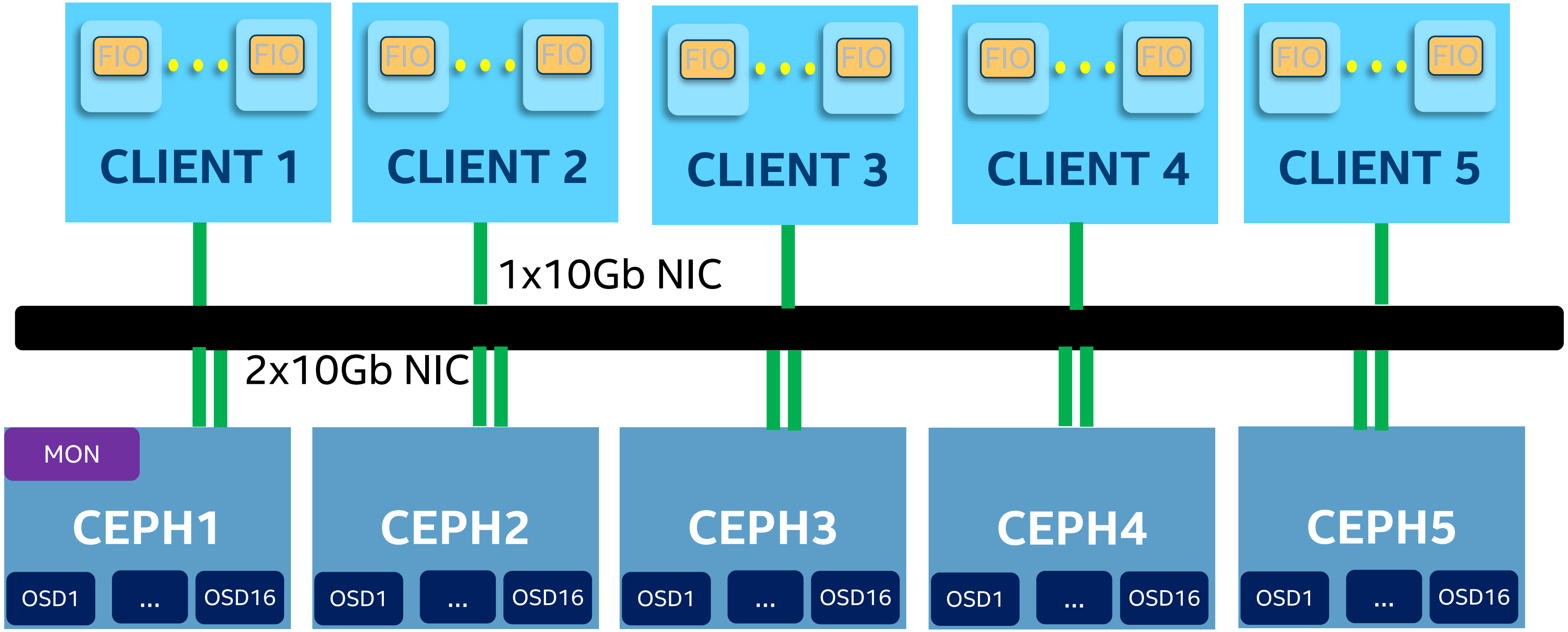
Strong demands to run enterprise applications

- OLTP workloads running on Ceph
- high performance multi-purpose Ceph cluster is a key advantage
- Performance is still an important factor

SSD price continue to decrease

Ceph* All Flash 3D NAND configuration

Test Environment



5x Client Node

- Intel® Xeon™ processor E5-2699 v3 @ 2.3GHz, 64GB mem
- 10Gb NIC

5x Storage Node

- Intel Xeon processor E5-2699 v3 @ 2.3 GHz
- 128GB Memory
- 1x 400G SSD for OS
- 1x Intel® DC P3700 800G SSD for journal (U.2)
- 4x 2.0TB Intel® SSD DC P3520 as data drive
- 4 OSD instances one each P3520 SSD

Software Configuration

- Ceph* 10.2.1, 2 replica, 2048 pg per OSD
- Ubuntu 14.04

*Refer to P27 for Ceph configurations, P14 for Optimization Notes.

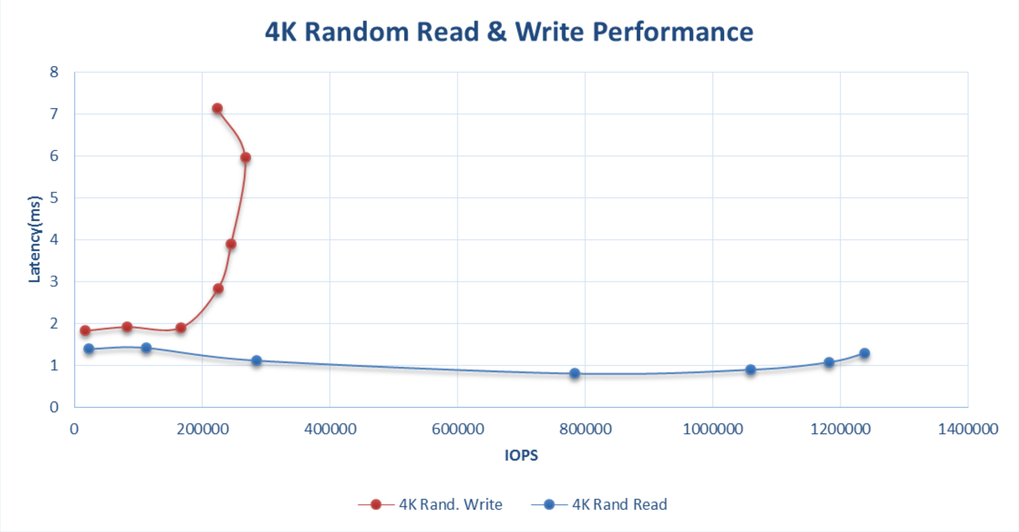
*Other names and brands may be claimed as the property of others.

Ceph 3D NAND Performance overview with bluestore

	Throughput	Latency	Comments
4K Random Read	1240K IOPS	1.29ms	Throttled by NIC BW
4K Random Write	270K IOPS	5.94ms	Throttled by CPU
64K Sequential Read	5207 MB/s	NA	Throttled by NIC BW
64K Sequential Write	5011 MB/s	NA	Throttled by NIC BW

- Excellent performance on 3D NAND cluster, performance was throttled by HW bottlenecks

The performance problems – Ceph* on all flash array – 4K and 64K performance scaling

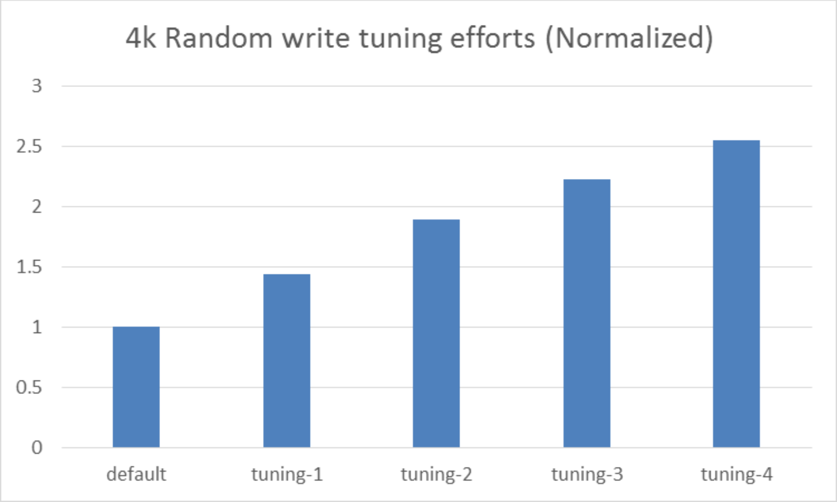


- 1.2M IOPS for 4K random read @ 1.3 ms latency, 270K IOPS for 4K random write with tunings and optimizations
- Sequential Read and Write performance throttled by NIC BW

Excellent random read performance and Acceptable random write performance

*Refer to P27 for Ceph configurations, P14 for Optimization Notes.

Tuning results dashboard

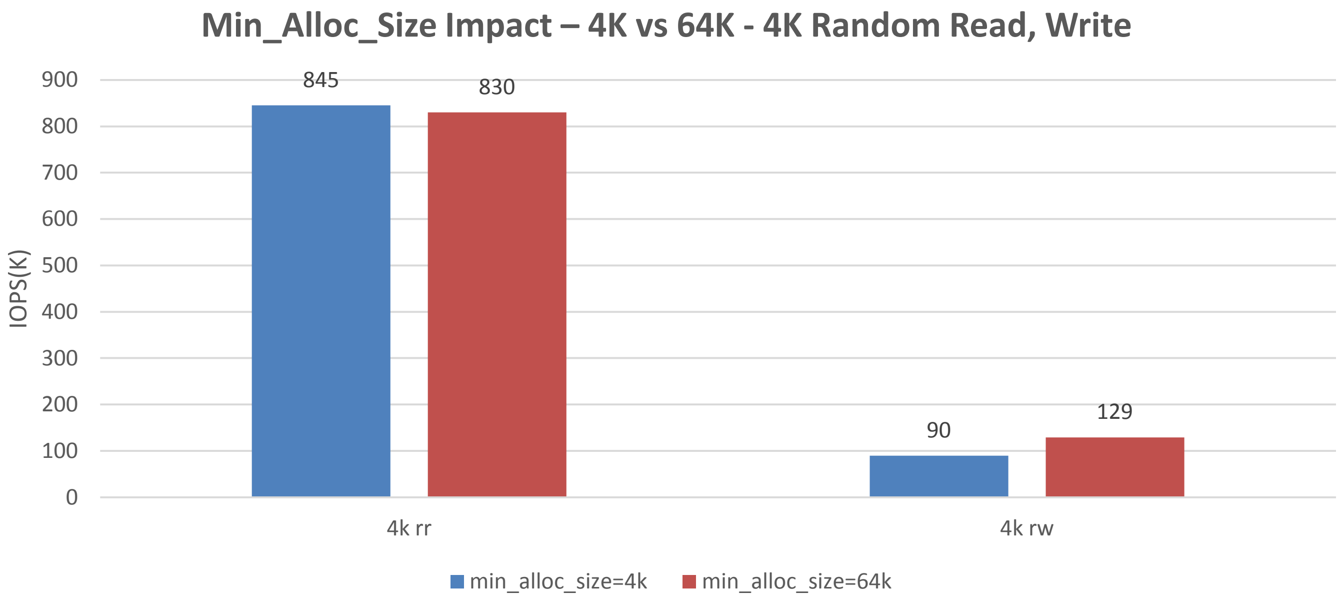


	Tunings
Default	/
Tuning-1	Default + set min_alloc_size to 64K
Tuning-2	Tuning-1 + RockDB compaction thread tuning
Tuning-3	Tuning-2 + RocksDB&WAL on NVME
Tuning-4	Tuning-3 + disable bdev-flush*

- 1.5x performance improvement with various tunings and optimizations
- **Metadata plane**(Rocksdb) has significant performance impact!

*Notes: disable fdatasync() in KernelDevice.cc and relies on pdflush

Metadata Tuning: min_alloc_size impact

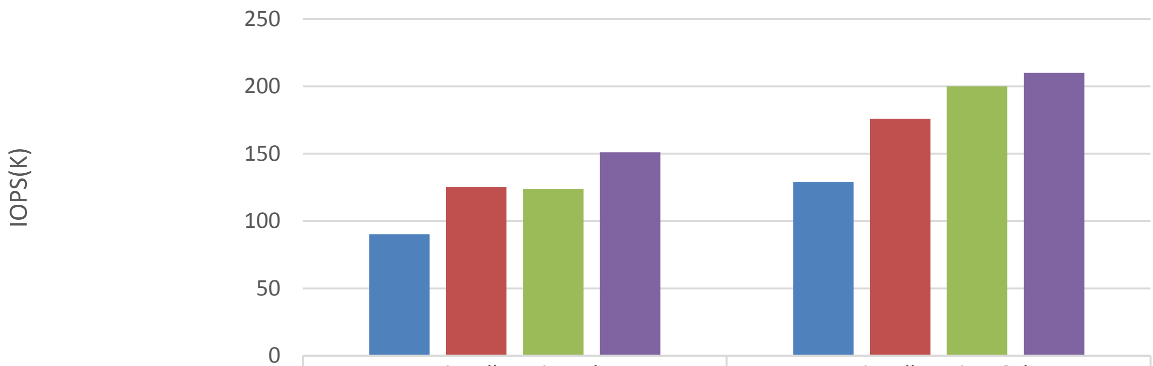


- min_alloc_size = 64k
  - Less meta data stored in rocksdb
  - Less device fdatasync operation
  - More write amplification

Note: Performance measured on another cluster

Metadata Tuning: RocksDB compaction tuning

Delay RocksDB Compaction – Min Alloc Size 4K vs 64K – 4K Random Write



	min_alloc_size=4k	min_alloc_size=64k
without rock tuning	90	129
with rocksdb tuning	125	176
moving to nvme device	124	200
delay rocksdb compaction	151	210

without rock tuning   with rocksdb tuning   moving to nvme device   delay rocksdb compaction

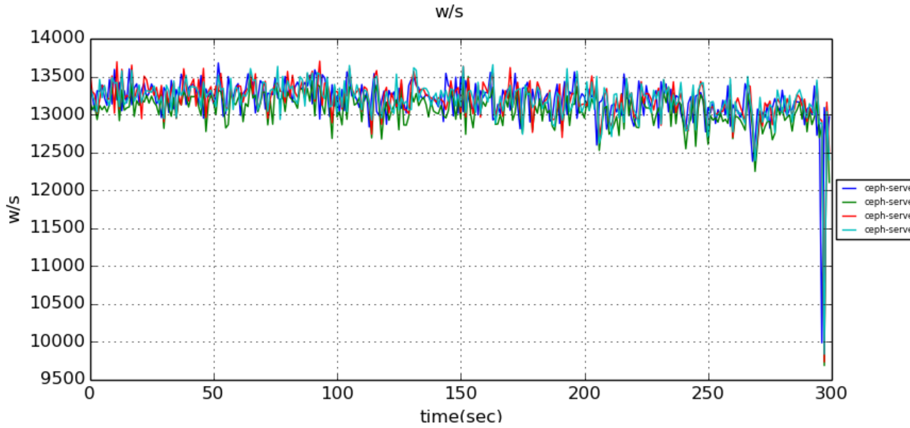
- Tuning
  - Increase level0 compaction trigger
  - Increase write buffer size
  - Use NVME(p3700) as RocksDB based device
  - Delay RocksDB compaction

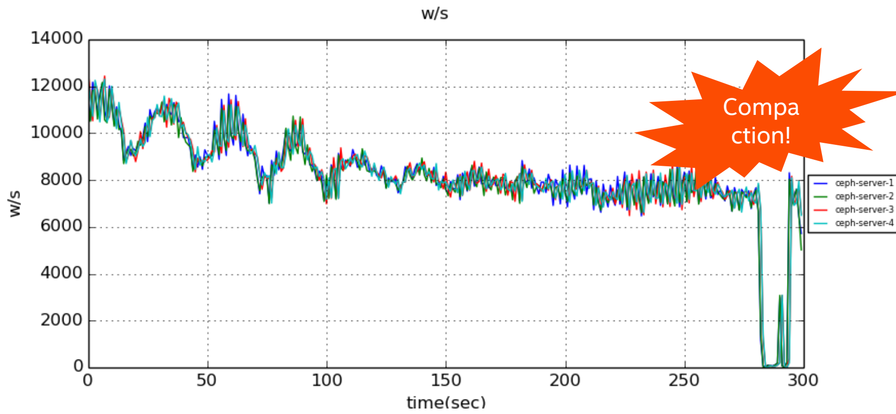
- Benefit
  - 4k random write performance improve by 55.0%
  - More stable IO

Metadata Tuning: RocksDB compaction tuning

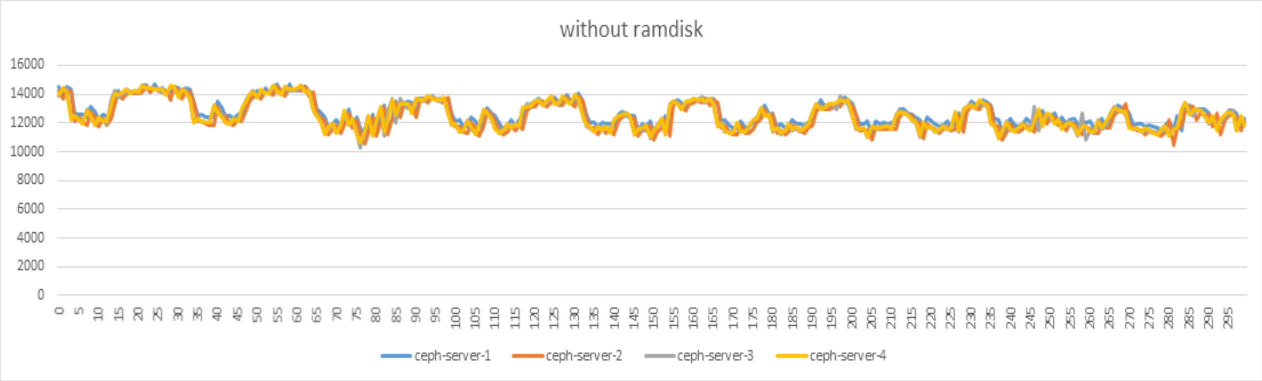
- RocksDB Problem
  - Can’t find one general RocksDB configuration for all use case
  - Tradeoff between write-amp, read-amp and space-amp
  - We can delay RocksDB compaction, but can’t avoid

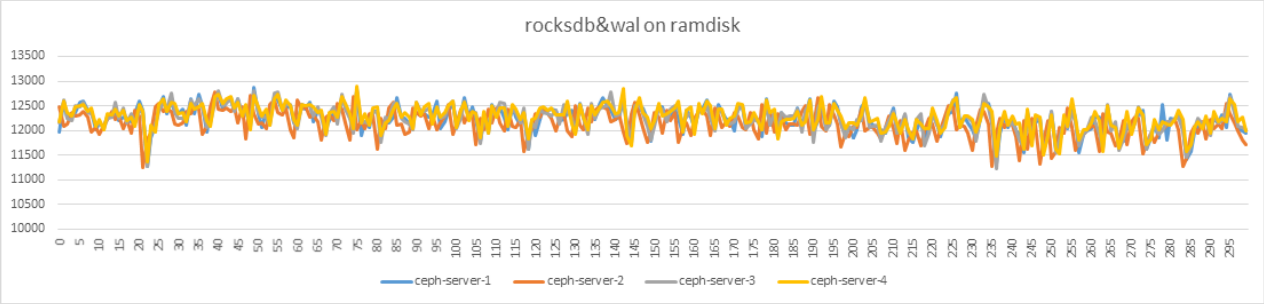
- Next steps
  - Identify RocksDB overhead





Metadata Tuning: RocksDB on Ramdisk



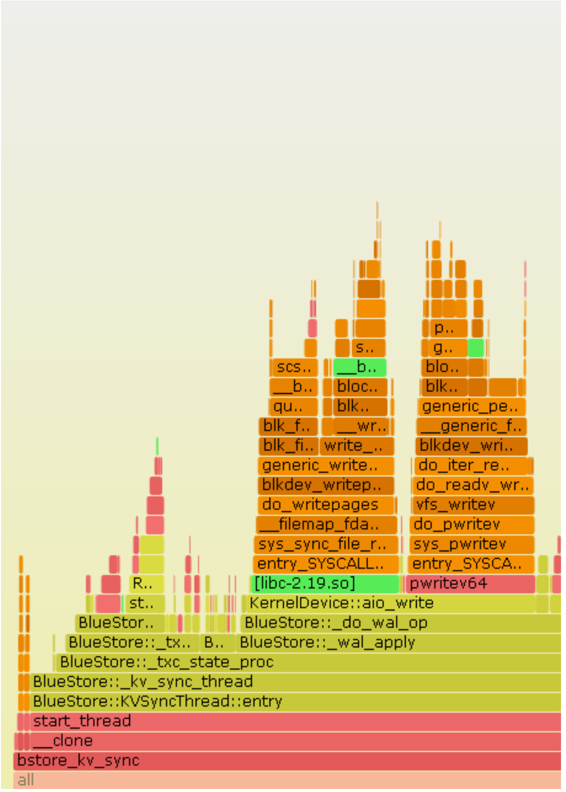


	IOPS	CPU Idle
Baseline	200K	40%
RocksDB on Ramdisk	194K	37%
RocksDB&WAL on Ramdisk	194K	36%

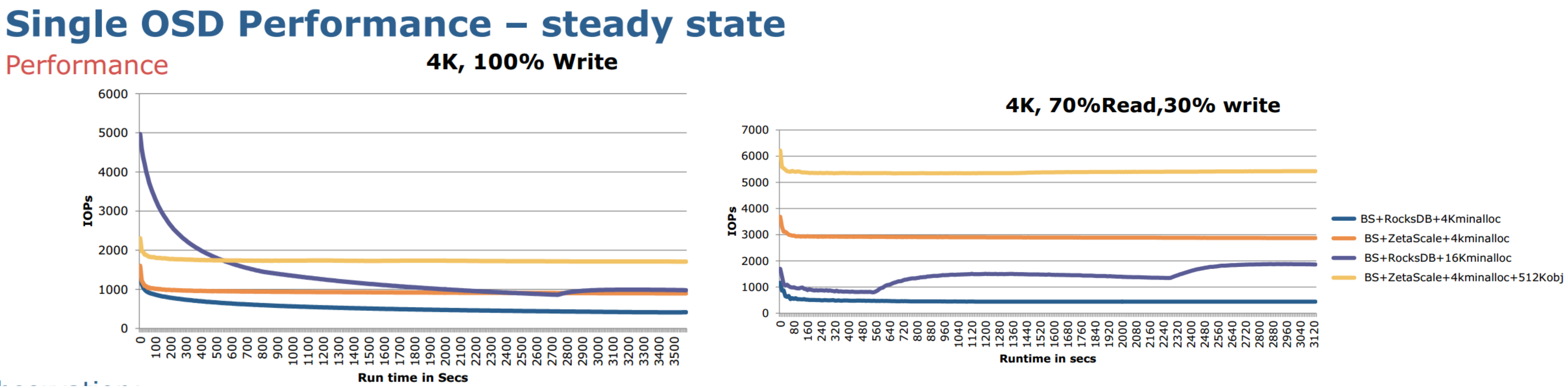
- Ramdisk can speed up compaction of rocksdb, but there is still some other rocksdb internal overheads, even though running on ramdisk

Metadata Tuning: reduce races on KV Sync transaction

- Rocksdb transaction is heavy …
  - ~10% of OSD write latency
- Cleanup kv_sync_thread and move transaction unrelated code out
  - So we could speed up on submit transaction
  - Pending PR #11189

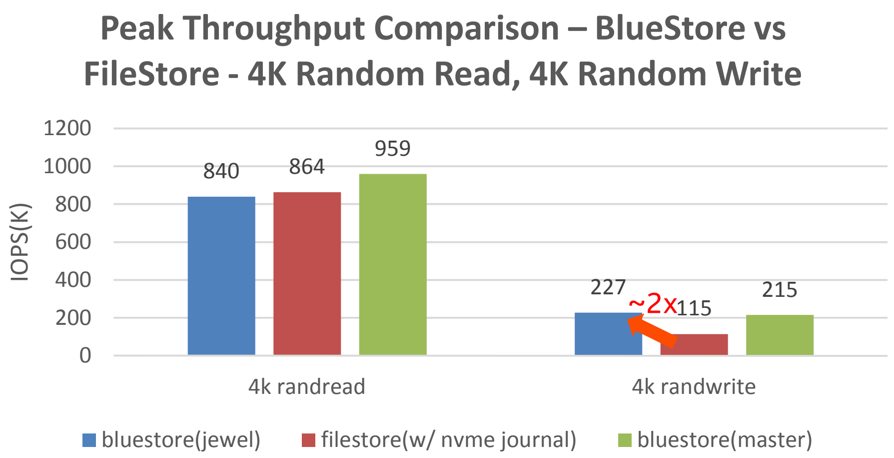


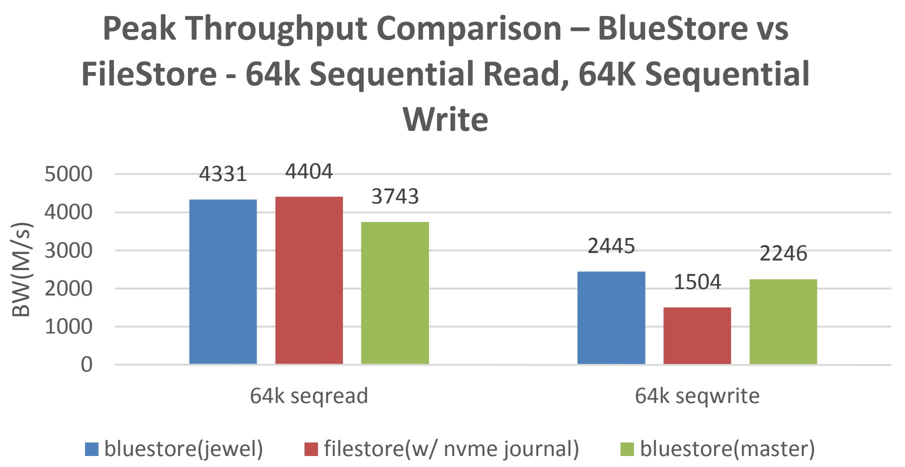
Metadata tuning: replacing Rocksdb?

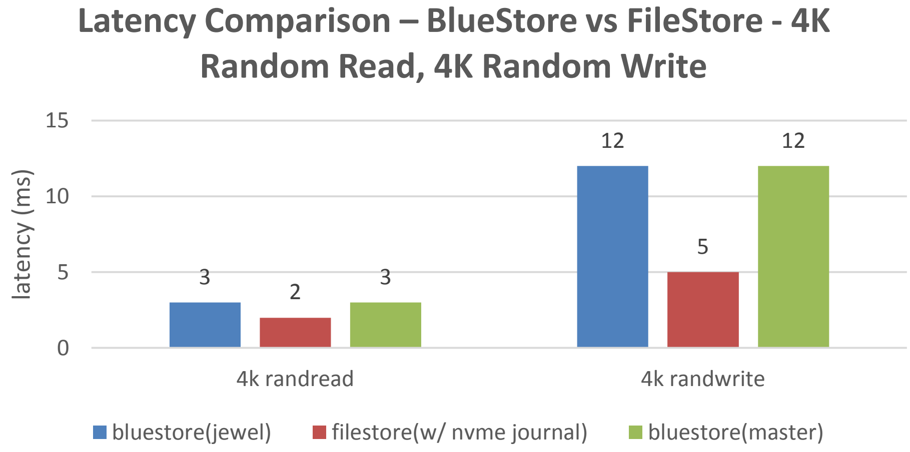


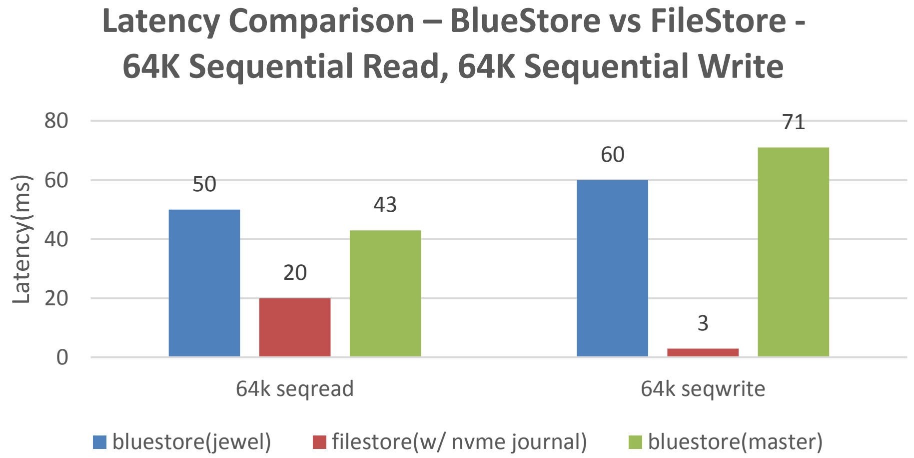
Source: Westen Digital: bluestore performance – RocksDB, ZetaScale
https://drive.google.com/file/d/0B7W-S0z_ymMJZXI3bkZLX3Z2U0E/view?usp=sharing

BlueStore vs FileStore*









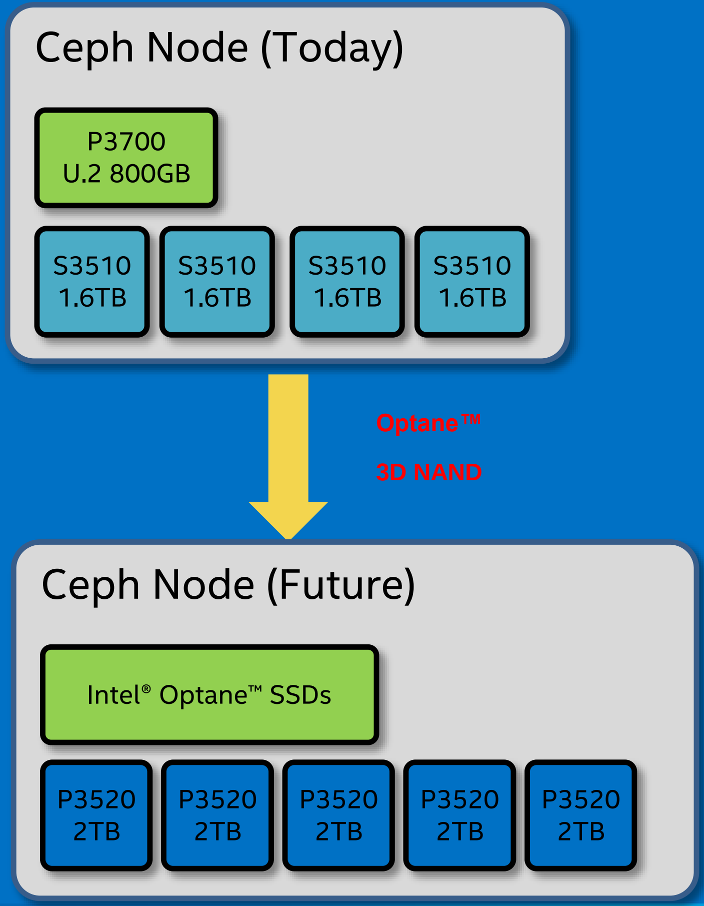
- FileStore throughput is normalized throughput
- QD = 64 for BlueStore

Intel® Optane™ & Intel® 3D NAND SSDs
High performance & cost effective solutions

- Enables high performance & cost effective solutions
- Enterprise class, highly reliable, feature rich, and cost effective All Flash Array (AFA) solution:

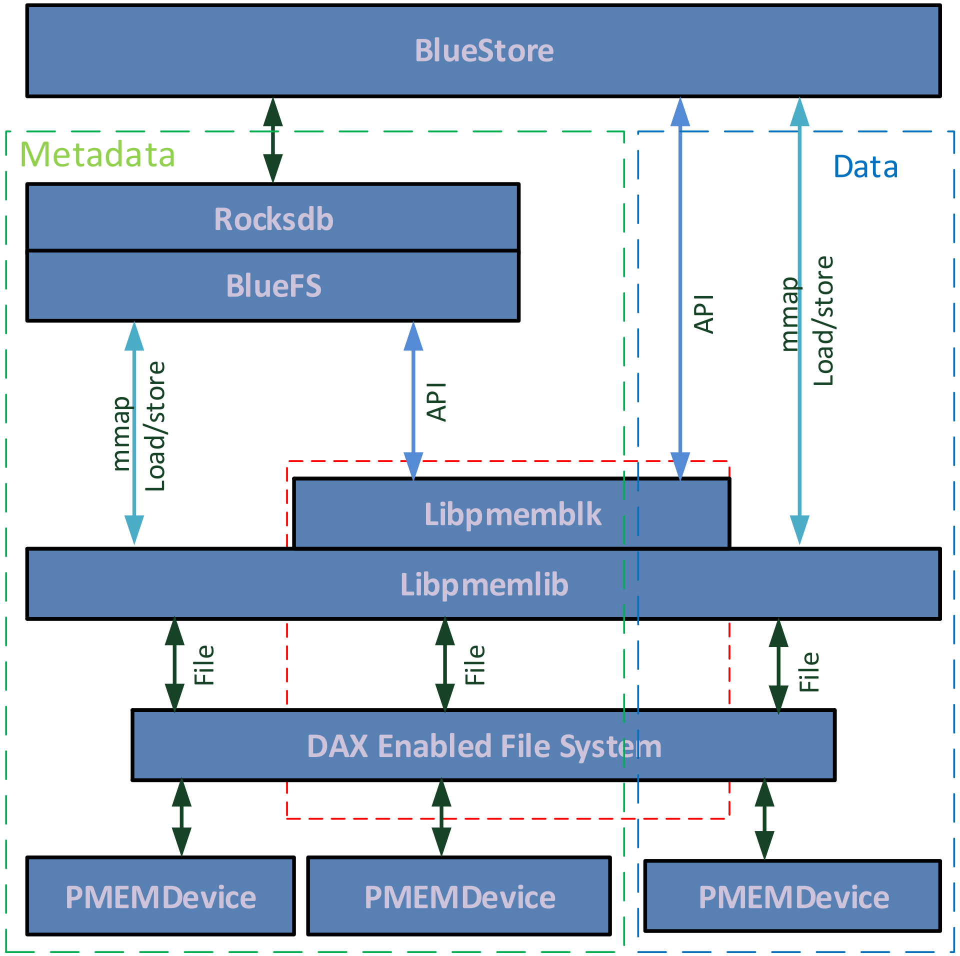
  Intel Optane™ (performance) + 3D NAND TLC SSD as data store (capacity)

- Enhance value through special software optimization on filestore and bluestore backend



INTEL® 3D Xpoint™ opportunities: Bluestore backend

- Three usages for PMEM device
  - Backend of bluestore: raw PMEM block device or file of dax-enabled FS
  - Backend of rocksdb: raw PMEM block device or file of dax-enabled FS
  - Backend of rocksdb's WAL: raw PMEM block device or file of DAX-enabled FS
- Two methods for accessing PMEM devices
  - libpmemblk
  - mmap + libpmemlib
  - https://github.com/Ceph*/Ceph*/pull/8761



Summary

- Ceph* is awesome!
- Strong demands for all-flash array Ceph* solutions
- SATA all-flash array Ceph* cluster is capable of delivering over 1M IOPS with very low latency!
- Bluestore shows significant performance increase compared with filestore, but still needs to be improved
- Let's work together to make Ceph* more efficient with all-flash array!

LEGAL NOTICES

FTC Optimization Notice
Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice revision #20110804

The cost reduction scenarios described in this document are intended to enable you to get a better understanding of how the purchase of a given Intel product, combined with a number of situation-specific variables, might affect your future cost and savings. Nothing in this document should be interpreted as either a promise of or contract for a given level of costs.

intel®

BACKUP

Software Configuration

```
"global":
    "debug objectcacher": "0/0",
    "debug paxos": "0/0",
    "debug journal": "0/0",
    "mutex_perf_counter": true,
    "cephx require signatures": false,
    "debug mds": "0/0",
    "mon_pg_warn_max_per_osd": 10000,
    "debug lockdep": "0/0",
    "debug auth": "0/0",
    "debug mds_log": "0/0",
    "debug mon": "0/0",
    "debug perfcounter": "0/0",
    "perf": true,
    "debug monc": "0/0",
    "debug throttle": "0/0",
    "debug mds_migrator": "0/0",
    "debug mds_locker": "0/0",
    "debug rgw": "0/0",
    "debug finisher": "0/0",
    "debug journaler": "0/0",
    "debug bdev": "0/0",
    "debug mds_balancer": "0/0",
    "debug ms": "0/0",
    "debug hadoop": "0/0",
    "debug client": "0/0",
    "debug context": "0/0",
    "debug osd": "0/0",
    "debug bluestore": "0/0",
    "debug memdb": "0/0",
    "debug bluefs": "0/0",
    "debug objclass": "0/0",
```

```
“global”:
    "debug objecter": "0/0",
    "debug log": "0/0",
    "debug filer": "0/0",
    "debug rocksdb": "0/0",
    "osd_pool_default_pgp_num": 32768,
    "debug mds_log_expire": "0/0",
    "debug crush": "0/0",
    "debug optracker": "0/0",
    "osd_pool_default_size": 2,
    "debug tp": "0/0",
    "cephx sign messages": false,
    "debug rados": "0/0",
    "osd_pool_default_pg_num": 32768,
    "debug heartbeatmap": "0/0",
    "ms_nocrc": true,
    "debug buffer": "0/0",
    "debug asok": "0/0",
    "debug rbd": "0/0",
    "debug filestore": "0/0",
    "debug timer": "0/0",
    "rbd_cache": false,
    "throttler_perf_counter": true,
```

```
"client":
  "admin socket": "/var/run/ceph/rbd-$pid.asok",
  "log file": "/var/log/ceph/$name.log"
"mon":
  "mon_max_pool_pg_num": 166496,
  "mon_osd_max_split_count": 10000,
  "mon_pg_warn_max_per_osd": 10000
"disk":
  "read_ahead_kb": "16"
"osd":
  "osd_client_message_size_cap": 0,
  "objecter_infilght_op_bytes": 1048576000,
  "ms_dispatch_throttle_bytes": 1048576000,
"  "osd_op_num_threads_per_shard": 2,
  "osd_op_num_shards": 8,
  "osd_op_threads": 1,
  "objecter_inflight_ops": 102400,
  "osd_enable_op_tracker": false,
  "osd_client_message_cap": 0,
  "bluestore_wal_threads": 3
```