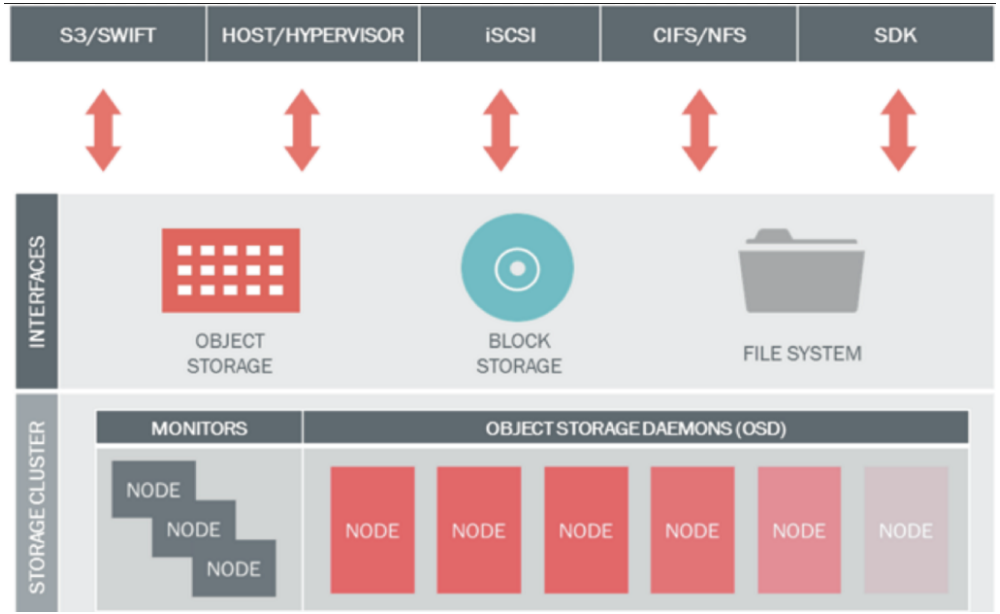
Improving Ceph Performance while Reducing Costs

Applications and Ecosystem Solutions Development
Rick Stehno

Flash Application Acceleration

- Three ways to accelerate application performance with flash:
  - Utilize flash caching features to accelerate critical data. Caching methods can be write-back for writes, write-thru for disk/cache transparency, read cache, etc..
  - Utilize storage tiering capabilities. Performance critical data resides on flash storage, colder data resides on HDD
  - Utilize all flash storage to accelerate performance when all application data is performance critical or when the application does not provide the features or capabilities to cache or to migrate the data

Ceph Software Defined Storage (SDS) Acceleration



Configurations:

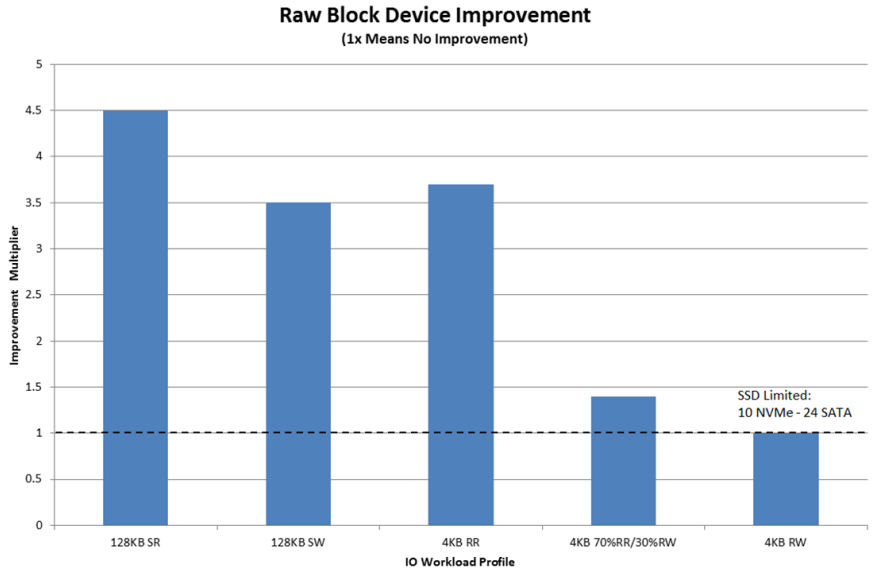
- All flash storage - Performance
  - Highest performance per node
  - Less maximum capacity per node
- Hybrid HDD and flash storage - Balanced
  - Balances performance, capacity and cost
  - Application and workload suitable for
    - Performance critical data on flash
    - Utilize host software caching or tiering on flash
- All HDD storage - Capacity
  - Maximum capacity per node, lowest cost
  - Lower performance per node

All Flash Storage
NVMe vs SATA SSD

- Why 1U server with 10 NVMe SSDs may be better choice vs 2U Server with 24 SATA SSDs
  - Higher performance in half the rack space
  - 28% less power and cooling
  - Lower MTBF inherent with reduced component count
  - Reduced OSD recovery time per Ceph node
  - Lower TCO

All Flash Storage
NVMe vs SATA SSD cont'd

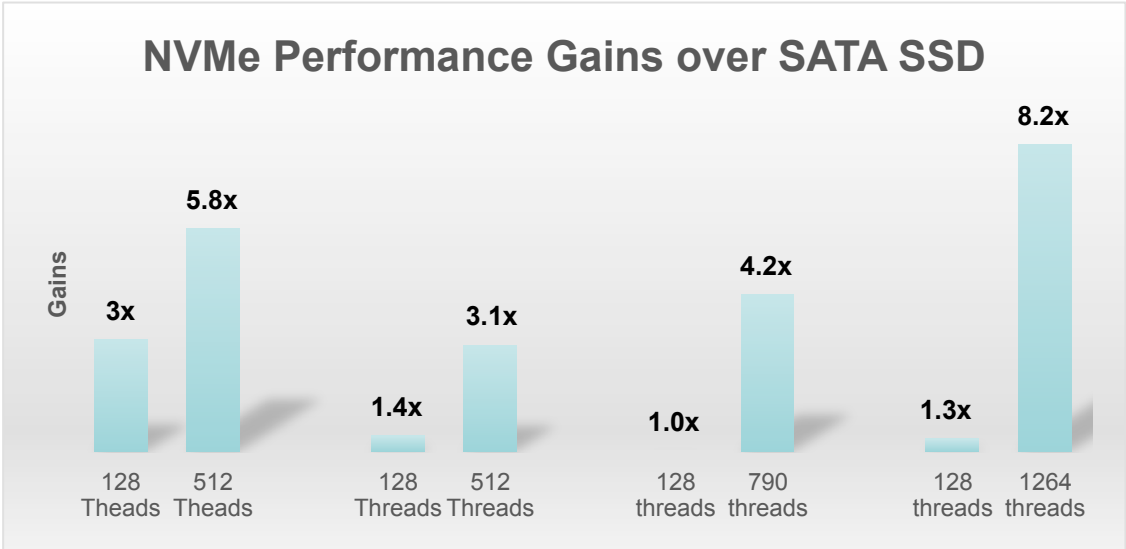
- Why 1U server with 10 NVMe SSD SSDs may be better choice vs 2U Server with 24 SATA SSDs



- 4.5x increase for 128k sequential reads
- 3.5x increase for 128k sequential writes
- 3.7x increase for 4k random reads
- 1.4x increase for 4k random 70/30 RR/RW
- Equal performance for 4k random writes

All Flash Storage
NVMe vs SATA SSD cont'd

- Why 1U server with 10 NVMe SSDs may be better choice vs 2U Server with 24 SATA SSDs



Increasing the load to stress NVMe capabilities over and above the 128 thread SATA SSD Test:

- 5.8x increase for Random Writes at 512 threads
- 3.1x increase for 70/30 RR/RW at 512 threads
- 4.2x increase for Random Reads at 790 threads
- 8.2x increase for Sequential Reads at 1264 threads

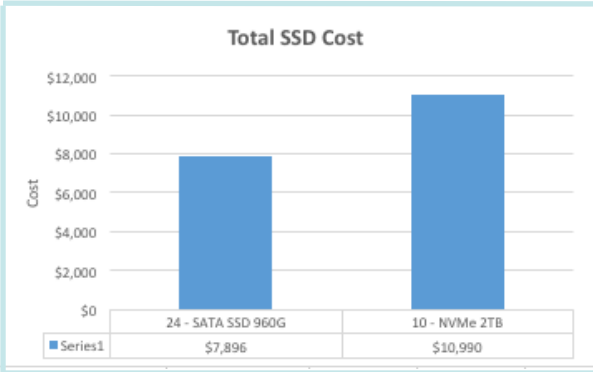
Ceph Storage Costs SATA SSD vs NVMe Equal FIO RBD Random Write - 128 Threads

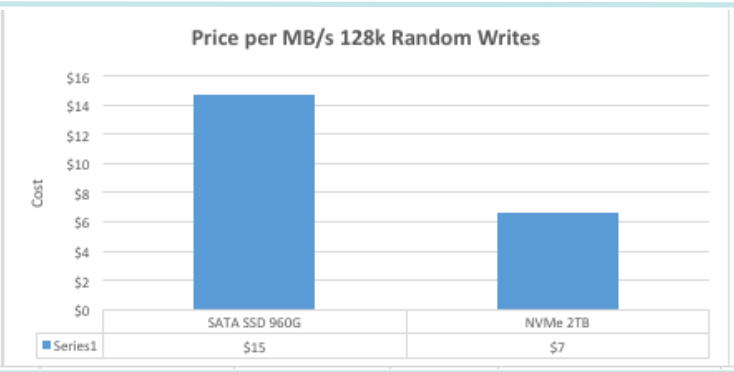
SSD	Total SSD Price		Price MB/s 128k Random Writes			Price MB/s 128k Sequential Reads	
24 - SATA SSD 960G	$7,896		24 - SATA SSD 960G	$15.00		24 - SATA SSD 960G	$8.00
10 - NVMe 2TB	$10,990		10 - NVMe 2TB	$7.00		10 - NVMe 2TB	$8.00

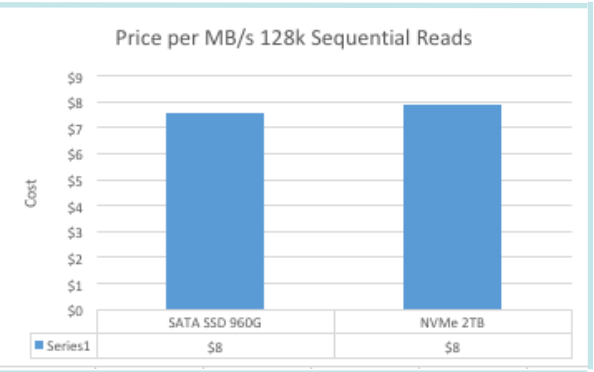
These tests were not done to show maximum performance for each set of devices, NVMe costs will be much lower when at maximum performance

Price per MB/s: Cost of ((Retail Cost of SSD) / MB/s for each test)

These prices do not include savings from electrical/cooling costs, reducing datacenter floor space, from the reduction of SATA SSD







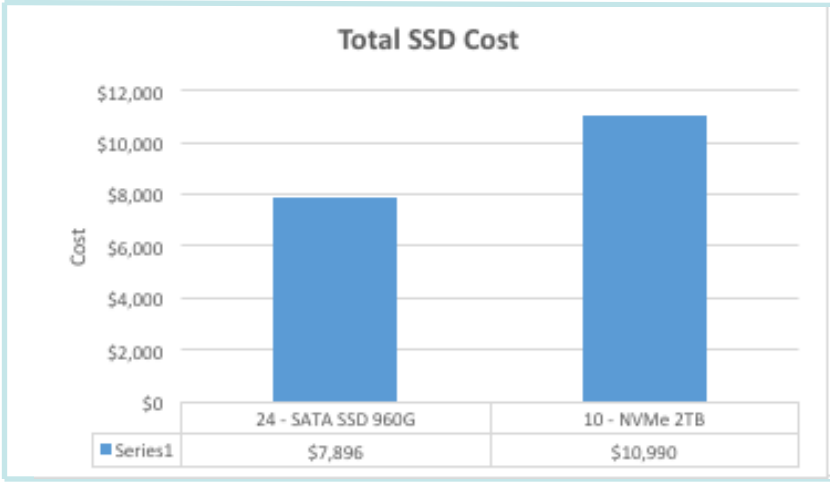
Note: 128k random write FIO RBD benchmark: SATA SSD averaged over 85% busy, NVMe averaged 35% busy

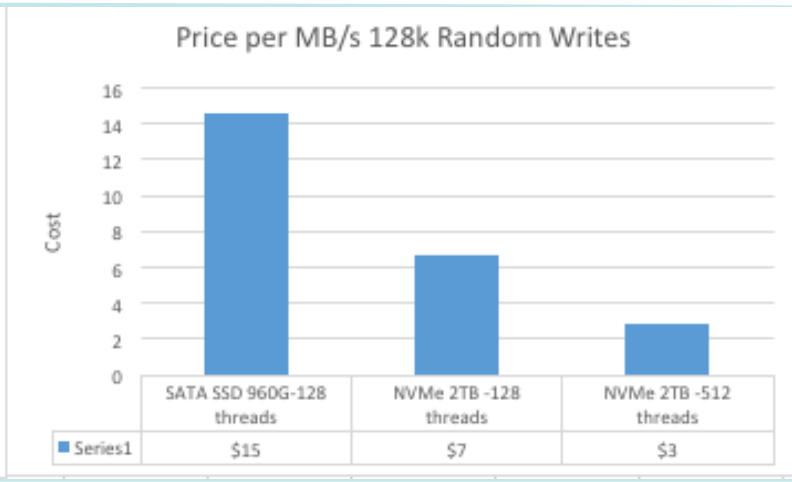
Ceph Storage Costs SATA SSD vs NVMe - FIO RBD 512 Threads Random Write Maximum Performance for NVMe

SSD	Total SSD Price		Price MB/s 128k Random Writes 128 threads			Price MB/s 128k Random Writes 512 threads	
24 - SATA SSD 960G	$7,896		24 - SATA SSD 960G	$15.00			
10 - NVMe 2TB	$10,990		10 - NVMe 2TB	$7.00		10 – NVMe 2TB	$3.00

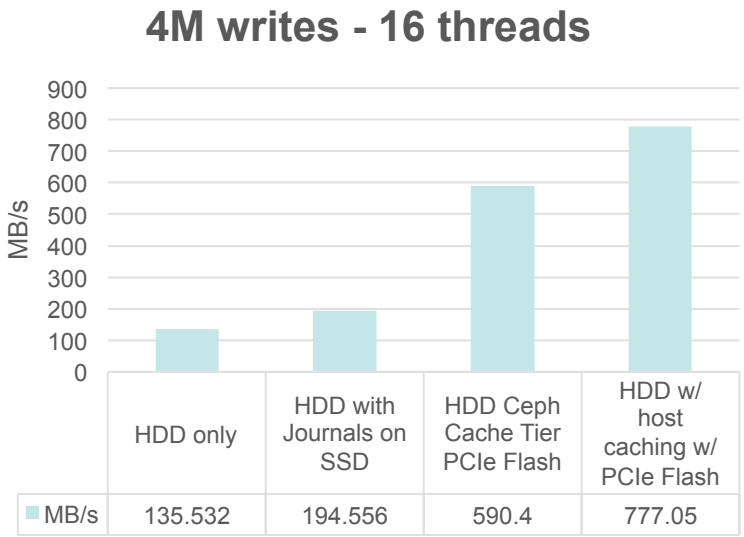
Price per MB/s: Cost of ((Retail Cost of SSD) / MB/s for each test)

These prices do not include savings from electrical/cooling costs, reducing datacenter floor space, from the reduction of SATA SSD

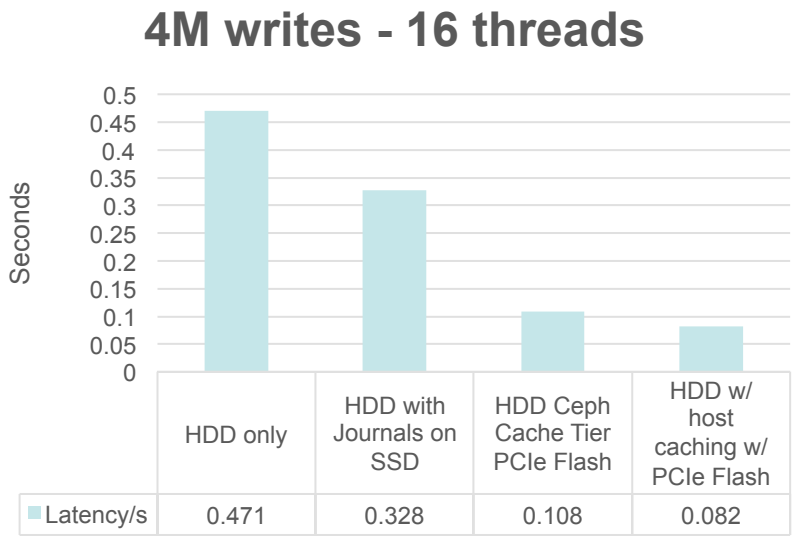




HDD Augmented with SSD/PCIe Flash Ceph Benchmarks – 4M Random Writes

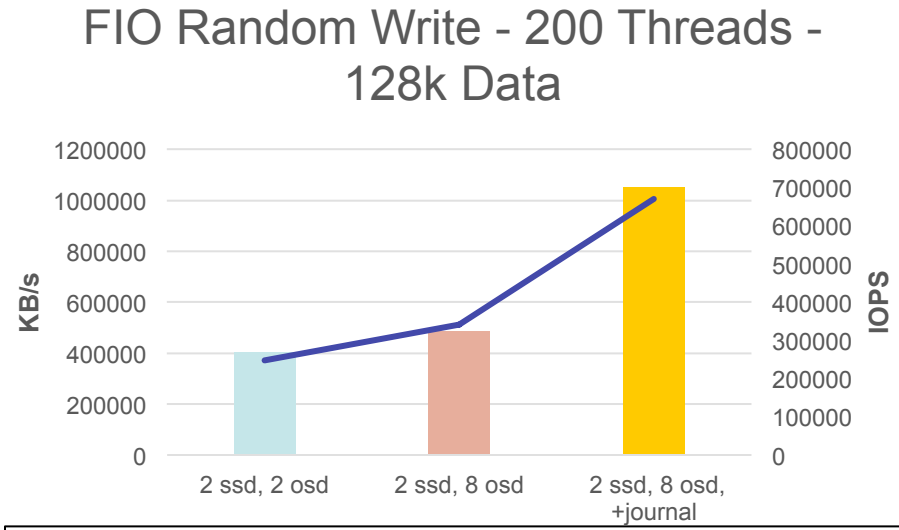


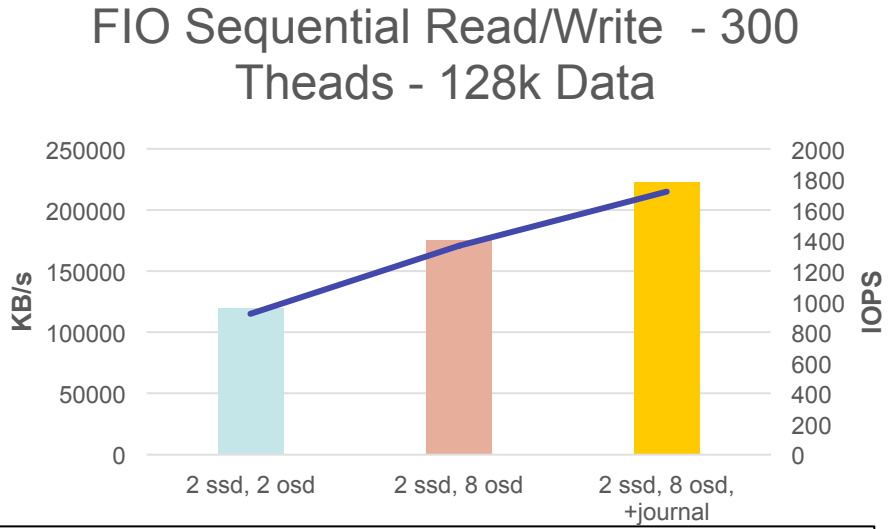
7x gain in MB/s



83% decrease in Latency

Ceph All Flash Storage Acceleration





All SSD

Case-1:
- 2 SSDs
- 1 OSD/SSD

Case-2:
- 2 SSDs
- 4 OSDs/SSD

Case-3:
- 1 PCIe flash
- 4 OSDs/SSD
- 8 OSD journals on PCIe flash

Flash Storage Tuning

Linux tuning is still a requirement to get optimum performance out of an SSD

- Use RAW device or create 1st partition SSD on 1M boundary (sector 2048)
  - Ceph-deploy uses an optimal alignment when creating an OSD
- Use blk-mq/scsi-mq if kernel supports it
- rq_affinity = 1 for NVMe, rq_affinity = 2 for non-NVMe
- rotational = 0
- blockdev --setra 4096

Flash Storage Tuning cont'd

Linux tuning is still a requirement to get optimum performance out of an SSD

- Using an older kernel, use:
  - "deadline" IO-Scheduler with supporting variables:
    - fifo-batch
    - front-merges
    - writes-starved

- Mount options:
  - noatime,inode64,logbsize=256k,noquota

- If using a smaller number of SSD, test with creating multiple OSD's per SSD. Have seen good performance increases using 4 OSD per SSD

Thank You! Questions?

Visit Seagate Booth #505

Learn about Seagate's ever-expanding portfolio of SSDs, Flash solutions and system level products for every segment

www.seagate.com/flash