

Performance and Advanced Data Placement Techniques with Ceph's Distributed Storage System



by Michael Poat & Dr. Jerome Lauret

Outline

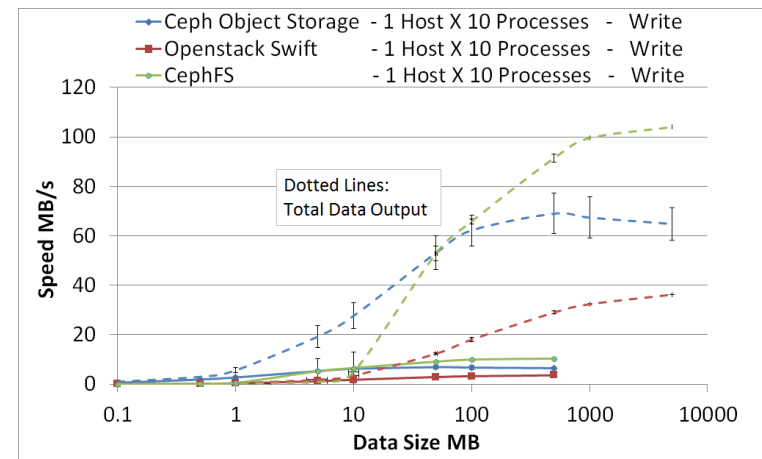
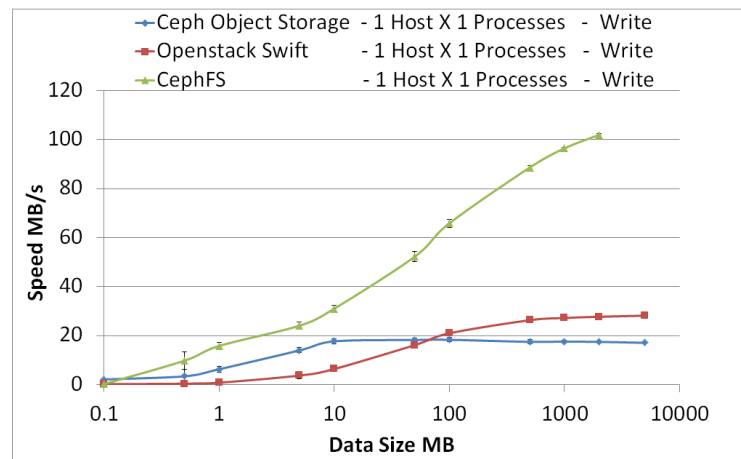
- Motivation
- Previous Work
- Architecture of Ceph
- Ceph features / capabilities
- Testing and applicability
- SSDs vs HDDs (w/o Ceph), alternative approach
- Conclusion

Motivation

- STAR has been interested in reusing our online cluster as a multipurpose storage system
 - Move some (fast) processing toward online domain requires storage (Online QA, real-time calibration, ...)
 - Users can store and recover data to a local storage while running jobs – “a” global space helps share results across processing nodes
 - Distributed file system & aggregators exists (Xrootd, Object storage, ...) – users most familiar with standard FS.
- CephFS offers
 - A familiar POSIX interface
 - Redundancy and shown to be crash tested safe [[doi:10.1088/1742-6596/664/4/042031](https://doi.org/10.1088/1742-6596/664/4/042031)]
 - Open source software that can be leverage to repurpose older hardware
 - An agile and simple architecture that ensures no single point of failure

Previous Work

- Presented at CHEP 2015
 - M. Poat, J. Lauret, W. Betts – “POSIX and Object Distributed Storage Systems - Performance Comparison Studies With Real-Life Scenarios in an Experimental Data Taking Context Leveraging OpenStack Swift & Ceph””, *J. Phys.: Conf. Ser.* **664** 042031 [doi:10.1088/1742-6596/664/4/042031](https://doi.org/10.1088/1742-6596/664/4/042031) (2015).
- Performance comparison study
 - Studied Openstack Swift object storage and Ceph Rados object storage
 - Investigated the performance and benefits of the CephFS distributed file system
- Ceph decreases the risk of single point of failure (no proxy server like Swift)
- CephFS outperformed both OpenStack Swift & Ceph object store due to its striping layer on top and breaking down files into 4MB chunks distributed to nearly all storage nodes simultaneously



Issues found (and fixed?) since

- **Stability**

- The STAR Ceph cluster has proven to be stable for beyond test use from 5/1/15 – present.
- Survived a “DNS meltdown” – 48/80 OSDs were down due to unstable DNS service. After DNS was restored, OSDs were able to come back in and recovered. No Data was lost.

- **Identified problems & lessons learned**

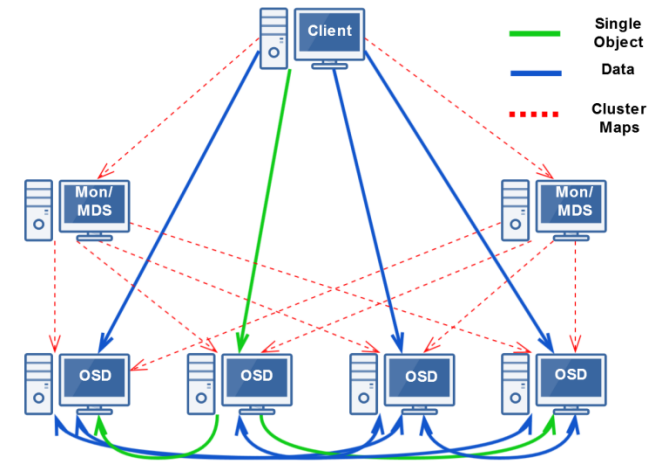
- Nightly condor jobs are run on the cluster reading OSG software from Ceph. Running Kernel 3.10.82 ~1x per month users reporting Jobs fail on one node due to missing OSG software. Kernel 3.18.24 contains latest modules (moral of story, run newest kernel).
- Do not remove 3 OSDs from 3 separate nodes at once from Ceph. Probability of data loss ~0.0012%

- **Experience**

- Once running Ceph requires little to no daily maintenance.
- If a drive fails, Ceph will reshuffle the data to maintain replication 3. The drive can be replaced at earliest convenience.

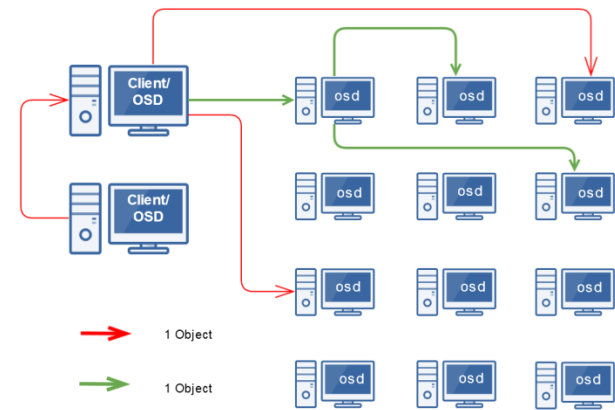
Architecture of Ceph

- Ceph is composed of three components:
 - Storage nodes (host individual OSDs), Monitor Nodes (Mon), & Metadata Nodes (MDS).
- Ceph clients write data directly to the storage nodes
 - Many storage systems use a proxy or middleman which can cause bottlenecks.
- The monitors distribute cluster maps to the clients and storage nodes.
 - Cluster map contains the cluster topology comprised five individual maps (Monitor, OSD, PG, CRUSH, and MDS)
- The metadata servers are required for CephFS



Ceph Communications

- Example: Total 20 Storage nodes. 2 nodes acting as both Ceph clients and storage nodes (OSD's)
- 2/20 (10%) chance "a" storage node will receive data from that client (but IO is spread)
- Replication 3 means 2 more stream of data will be sent => additional probability of $2/20 * 2 = +20\%$ to receive data
- Total data output increases by 20% in this example



Production system (β)

- 20 Dell PowerEdge 2950
- 6 – 2TB Seagate SAS drives
 - 4 Drives per node used for storage
 - XFS File System
- 1 HDD replaced with an SSD (per node)
- Intel Xeon QC E5440 – 2.83GHZ
- Scientific Linux 6.7 x86_64
- Kernel: 3.18.24 (needed to mount CephFS)
 - Kernel: 2.6.32 is the latest version released by SL6
- 1 Gb Public Network, 10Gb Private Network

Configuration

Ceph

- Ceph Hammer 0.94.5 (LTS)
- 80 OSD (20 nodes)
- 3 Monitor Server & 3 Metadata Servers (failover)
- Using Replication 3



Monitoring



- **Icinga** Infrastructure monitoring
- Ceph specific plugins have been built by the community
- Monitoring of Ceph OSDs, Metadata servers, & Monitor servers.

cephmon01.starp.bnl.gov

HEALTH WARNING: 4/76 in osds are down:

oni29.starp.bnl.gov

OSD WARN: Down OSDs osd.22 osd.23 osd.112 osd.114

Configuration Management

- **CFEngine** Configuration Management Tool
- Deploy Ceph Packages
- Set Configurations
- Deploy necessary Ceph files
- Detect and repair problems
 - Configuration drift, OSD service running, etc.



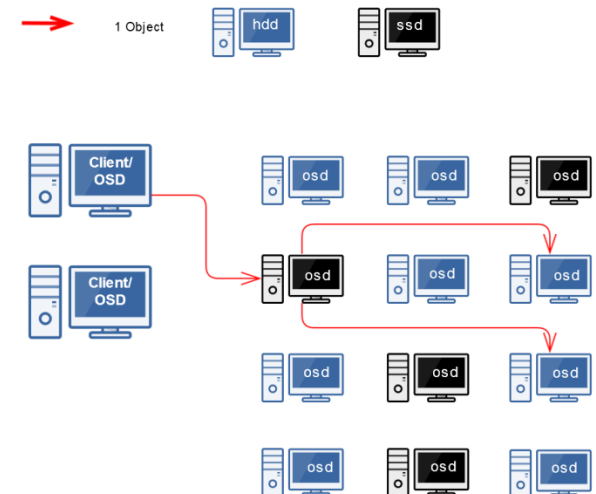
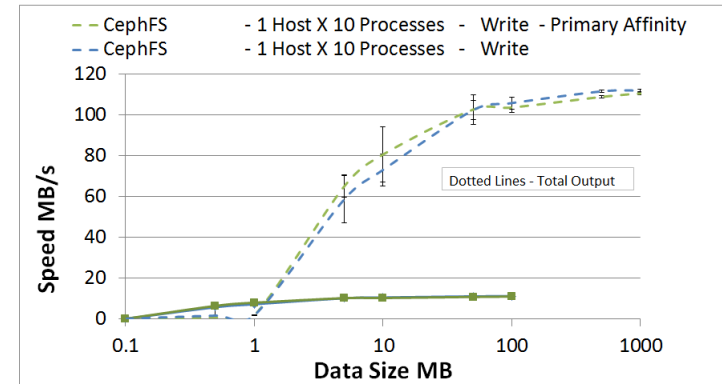
CEPH FEATURES

Data Placement Techniques

- **OSD Pool Mapping**
 - Ceph allows you to set specific OSDs to specific storage pools by customizing the CRUSH map.
 - You can modify the placement of data based on redundancy or performance, the combinations are endless.
- **Primary Affinity**
 - Specify which OSDs will be the primary OSDs when a client reads/writes data. This would ensure the client always contacts the primary OSD as the first OSD in the acting set.
- **Ceph OSD Journals on SSD**
 - When a client writes into Ceph, Ceph will write twice, first to the OSD journal and once to the OSD filesystem (with XFS).
- **Cache Tiering**
 - Cache tiering leverages OSD Pool Mapping by creating a pool of fast drives (SSD's) configured to act as the cache tier with a backing pool of slower drives (Currently does not work with CephFS)

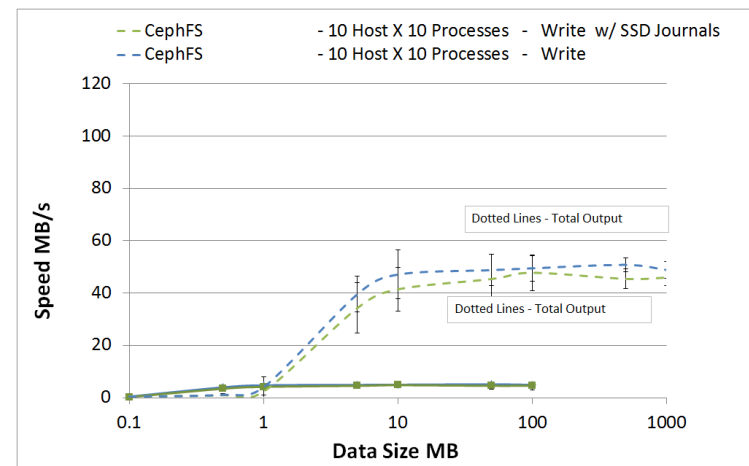
Primary Affinity with SSD OSDs

- Replace 1 HDD OSD per node with 1 SSD drive.
- Set the primary affinity on the SSDs to 1.0 while all other OSDs are set to 0.0
 - Theory is that the SSD OSDs will be accessed 100% for initial writes and HDD OSDs will be accessed 0%
 - If Primary Affinity is set to 1.0 on SSD OSDs & 0.5 on HDD OSDs: SSD should be accessed 75% and HDD will be access 25%
- After testing multiple PA configurations the performance was unchanged
- Each OSD has individual weights per drive (determines the fill ratio)
 - OSD weights are taken into account when using PA and may cause a skew in the expected performance
 - STAR's Ceph cluster (2TB HDD & 1TB SSD – 2:1 fill ratio)
- Possible explanation
 - The SSDs may be finished but still waiting for the HDD to sync causing no performance gain.



Mount Ceph OSD Journals on SSD

- Ceph OSDs writes twice to the disks (not simultaneously), first to the journal and second to the filesystem (with the XFS filesystem format).
- Lots of I/O would cause lots of journal operations (and small IO, possibly becoming random with process concurrency)
- Test: All nodes have their journal operation set to occur on SSDs, all other IOs to HDD

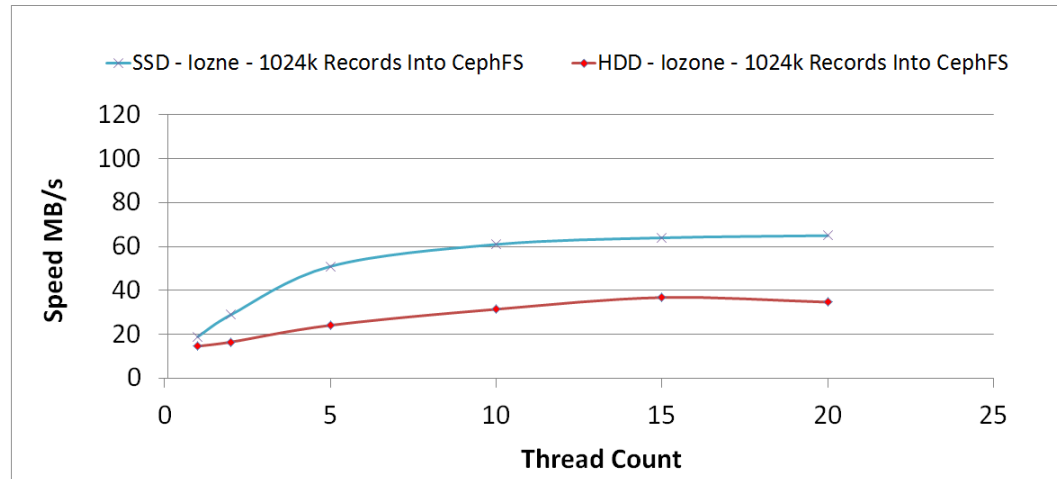
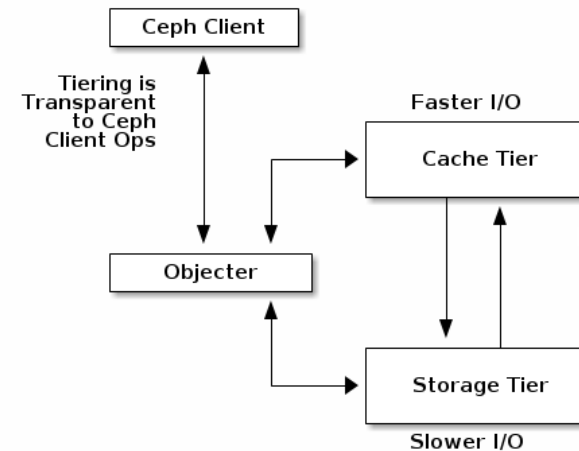


Results

- No performance gain seen with the OSD journals mounted on SSD
- We know Ceph performs journal operations using `D_SYNC` and `O_DIRECT` – this should however not influence IO over multiple threads (and the random access caused by concurrency)
- 10 processes for 10 clients (100 simultaneous writes) may not be enough to see a difference

Cache Tiering

- Cache tiering is configured where a small pool of fast drives (SSD) is overlaid on top of a larger slower pool (HDD)
- A cache tiering storage pool is represented as one storage pool to the user
- The client will write into the fast pool, as the pool begins to fill to a specified percentage the fast pool will release cold data (old data/ data not accessed often)
- Cache tiering is not implemented in CephFS yet, however the speed gain should be seen in a “only SSD” / “only HDD” CephFS storage comparison



Use of SSD

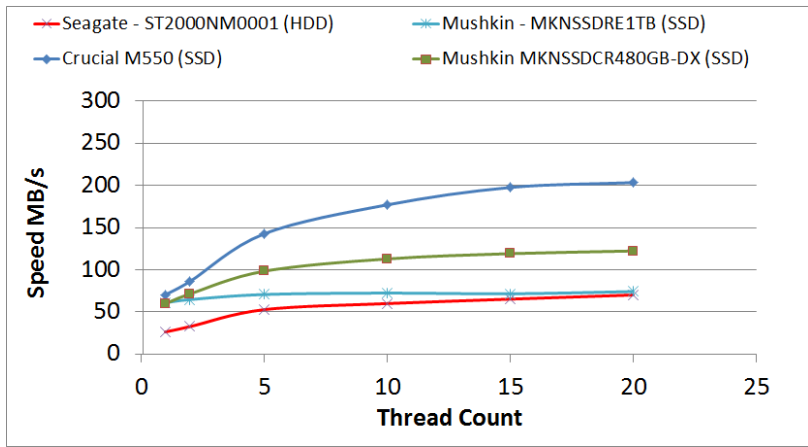
Manufacture Spec

Drive	Write MB/s	Read MB/s	IOPS Write	IOPS Read
Crucial M550	500	550	80,000	90,000
Mushkin MKNSSDRE1TB	460	560	76,000	74,000
Mushkin MKNSSDCR480GB-D	455	540	42,000	42,000

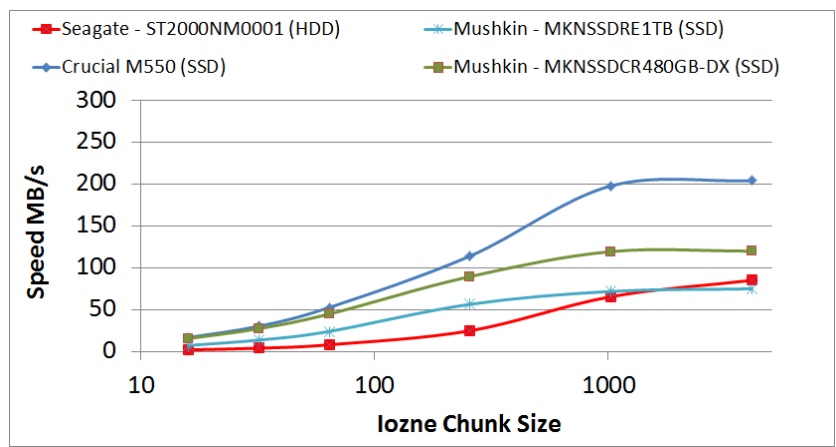
- We have obtained a few different models of SSD drives (4 x 1TB Crucial M550, 20 x 1TB Mushkin MKNSSDRE1TB, & 4 x 480GB Mushkin MKNSSDCR480GB-DX)
- Iozone test were performed to define raw performance of drives
- With large block size (1024KB) the SSD outperforms the HDD as the number of threads increase

- With small IO, even at 15 threads the performance between the drives are the same.
- As the chunk size increase the SSD outperforms the HDD

1024KB Records Iozone Writes



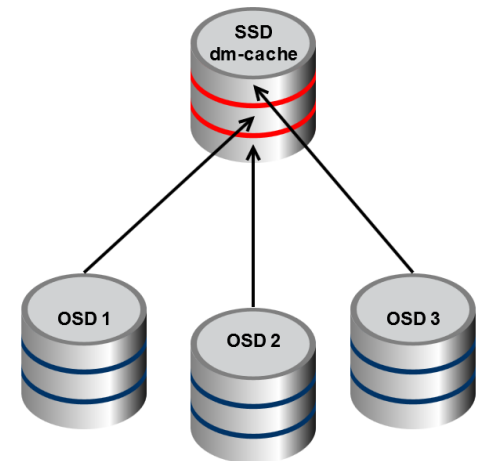
15 Thread Iozone Writes



dm-cache – an alternative to recover performance

- dm-cache is one example of a disk caching framework which is not part of the Ceph features
 - dm-cache allows a (fast) SSD drive to act as a cache for regular (slower) HDDs.
 - The two drives are paired to act as one volume and will have faster I/O due to the SSD acting as cache
 - dm-cache deploys their own policy on how/when to write data to the backing HDD
- Would circumvent the “affinity” and “weight” mechanism as Ceph would have no ideas of the presence of SSD
- Pro: dm-cache handles at low level the caching and migration, Ceph IO always goes first to SSDs
- Cons: if the SSD dies, we lose 3 OSDs

dm-cache has not yet been tested in our cluster. If our IO test results (previous slide) are right, applying this technique should create the performance impact / benefit we seek.

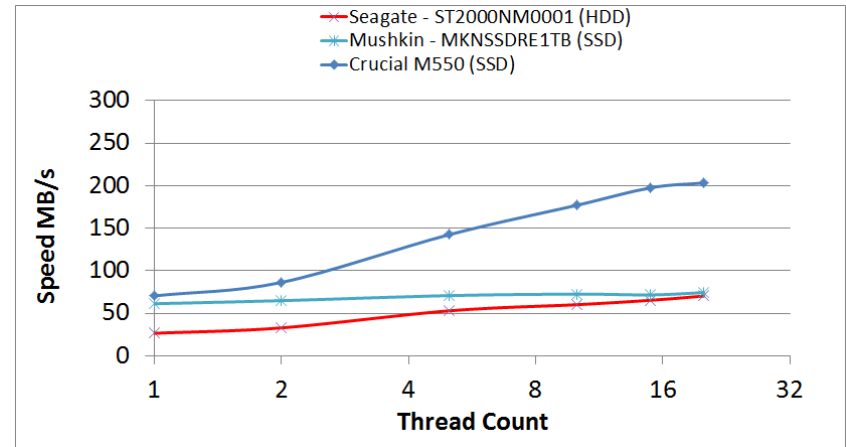


Cost Comparisons

- Crucial M550 1TB SSD: \$500 per drive
- Mushkin 1TB SSD: \$270 per drive
- Seagate SAS 2TB HDD: \$180 per drive

CRUCIAL: Cost impact ~1:2.1
performance increase ~1:2.5

MUSHKIN: Cost impact ~1:1.5
performance increase ~1:2



Drive	Size	Cost	Cost per TB
Seagate SAS	2TB	\$180	\$90
Crucial M550	1TB	\$500	\$500
Mushkin MKNSSDRE1TB	1TB	\$270	\$270

Drive	Cumm. Size	Cumm. Cost	Cost per TB	Cost % over base	Perf % over base
4 X Seagate SAS Per node	8TB	\$720	\$90	—	—
3 X Seagate SAS & 1 x Crucial SSD Per node	7TB	\$1040	\$192	115%	160%
3 X Seagate SAS + 1 x Mushkin SSD Per node	7TB	\$810	\$135	50%	100%

Conclusion

What we have learned

- The process in which Ceph writes to the OSDs brings difficulty when only replacing a portion of slow hardware with faster hardware – asymmetric size increase confusion (battle between weight and affinity)
- Journaling could help but in very specific (and large scale concurrency) cases
- Not all SSDs fall from the same tree – test carefully first
- Cache teiring (composed of SSD pools) may be the only Ceph option to see a performance increase – not available in CephFS
- dm-cache appears to be a promising solution, combining “cache tier-like” features while providing a POSIX FS to users (CephFS on top)

Outcomes & Future work

- dm-cache may be our only solution for performance increase with current hardware allocations. Testing on the way (unfortunately, was not ready for this conference)
- Our cluster has been running for ~ 1 year and has served as a successful distributed storage system for our users and has thus far delivered consistent data