Ceph SSD Performance Comparison

OSD Performance in Dumpling, Firefly, and Hammer

Mark Nelson

mnelson@redhat.com

2/17/2015

Table of Contents

INTRODUCTION

In the summer of 2013, Inktank released one of the first widely utilized LTS versions of Ceph called Dumpling. Since then Ceph Firefly was released and soon the first Red Hat backed LTS release of Ceph will be completed. When Ceph was originally created, most cluster deployments were based on spinning magnetic disks that can maintain relatively high throughput levels but also have relatively high seek times that translates into high synchronous op latency. Most spinning disks also are unable to read from multiple heads at the same time, limiting the performance of concurrent small random IO. Ceph is well optimized for spinning media, but as solid state disks have become more prevalent, users have requested optimization to take advantage of the new hardware. This paper will explore how Ceph performance has changed over the last couple of Ceph releases on several different models of SSD.

HARDWARE SETUP

To test the performance of Ceph OSDs with SSDs, a relatively simple system configuration was created. A single test node was employed with a single OSD backed by one SSD. No replication was used, and RADOS bench was configured to run on the same host as the OSD to limit the effect of network latency. A full description of the hardware follows:

Device	**Model**
Chassis	Supermicro SC847A
Motherboard	Supermicro X9DRH-7F
Disk Controller	Integrated + LSI SAS9207-8I
CPUS	2 X Intel XEON E5-2630L (2.0GHz, 6-core)
RAM	8 X 4GB Supermicro ECC Registered DDR 1333
NIC	Intel X520-DA2 10GbE (bonded configuration)

GNU parted was used to create a 10GB journal partition at the beginning of the drive with the remainder dedicated to a data storage partition. The “optimal” alignment setting was used to ensure that partition boundaries were well aligned. In the Firefly and Hammer releases of Ceph, the memstore OSD data store was also tested for comparison purposes. A description of the devices used in these tests follows:

Device	Description
Memstore	Alternate Ceph data store that uses in-memory STL containers. No journal overhead.
180GB Intel 520	Consumer grade SSD. Capable of up to 60K write and 50K read IOPS at a queue depth of 32. Appears to ignore ATA_CMD_FLUSH despite having no power failure protection.
200GB Intel DC S3700	Enterprise Grade SSD. Capable of up to 32K write and 75K read IOPS. Has power failure protection.
PCIe SSD #1	930GB Consumer/Enterprise grade PCIe SSD. Capable of 110K write and 155K read IOPS. Has power failure protection.
PCIe SSD #2	1.6TB Enterprise grade PCIe SSD. Capable of 120K write and 180K read IOPS. Has power failure protection.

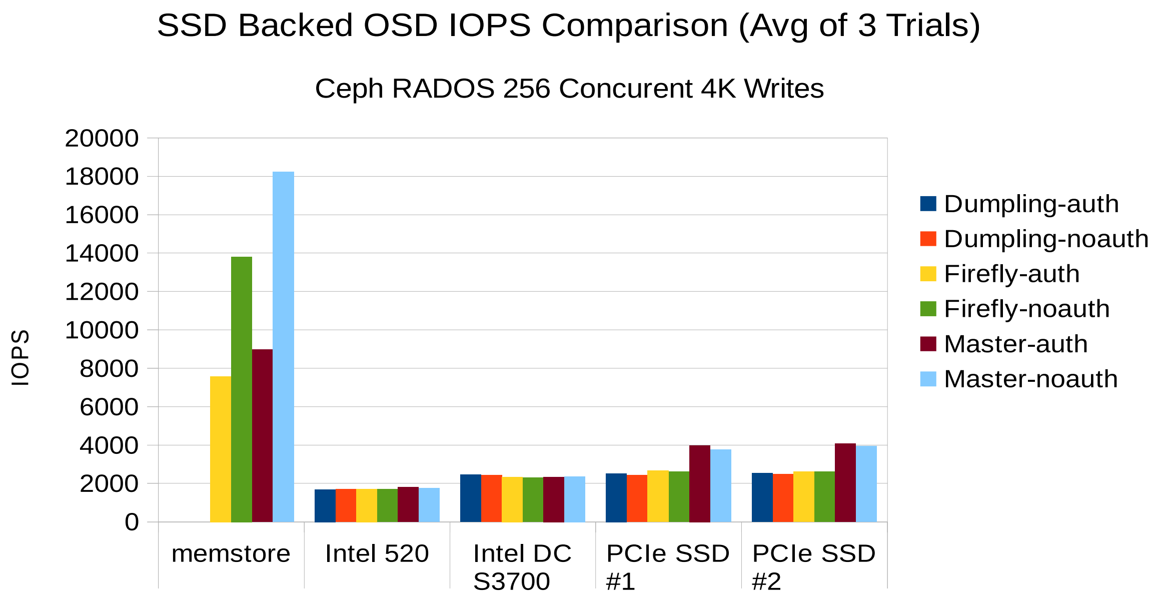
SOFTWARE SETUP

Three different versions of ceph were examined during these tests. Several settings, including disabling in-memory logging, were utilized to reduce overhead and improve performance. In January of 2015, Stephen Blinick from Intel noted that authentication appears to have a large effect on SSD performance in recent versions of Ceph. Each Ceph release in this study was tested with authentication enabled and disabled to observed the effect on performance.   A list of the software utilized for these tests follows:

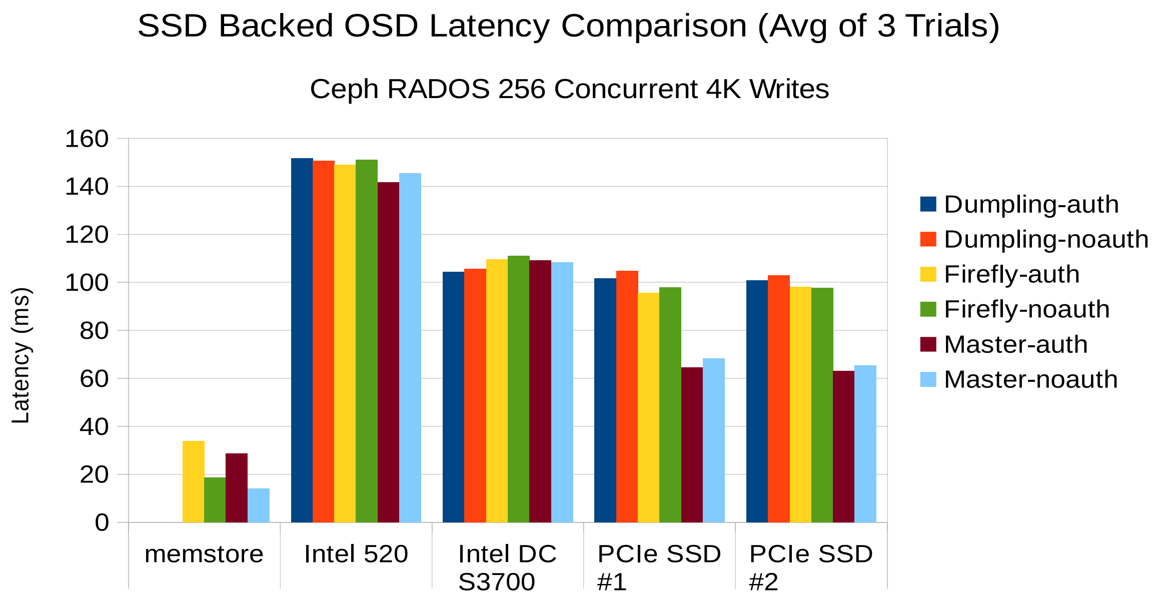
Software	Version
OS	Fedora Core 20
Kernel	3.17.4-200 from source
Ceph Dumpling	0.67.11-78-g657b1a2
Ceph Firefly	0.80.8-49-g9ef7743
Ceph Hammer (master)	0.89-465-gb2ca2e2
GNU parted	3.1
CBT	Latest Master

See Appendix A and Appendix B for details regarding the Ceph and CBT configuration files used during these tests.

256 CONCURRENT 4K WRITE RESULTS

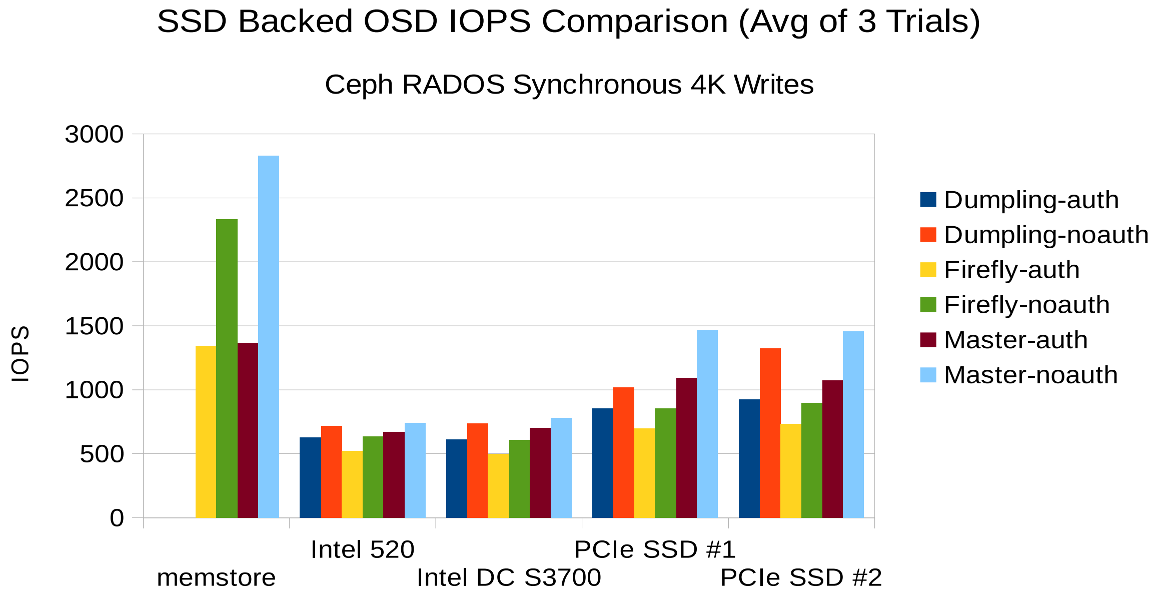


Hammer is doing very well compared to Firefly and Dumpling when the back-end is fast such as using the memstore or filestore with PCIe SSDs. This is also giving us a first glimpse into the effect authentication has on performance. When IOPS are low (say below 4K), the overhead caused by authentication is negligible. In situations where there is very low back-end latency (say using the memstore), disabling authentication can double performance!

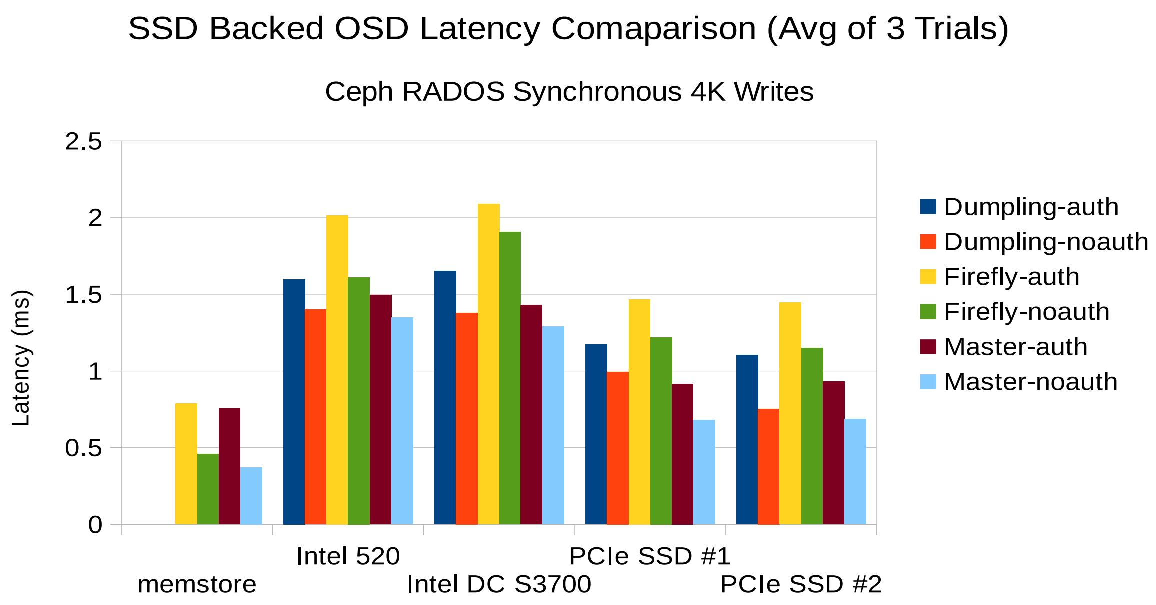


The latency in Hammer is correspondingly lower when fast storage back-ends are used. In this case, when the memstore is used a single OSD can handle 256 concurrent writes with latency below 20ms. There has been work on several improved messenger implementations that will be available in Hammer, which may help lower latency even further. In future releases of Ceph, a new data store that utilizes the best aspects of key/value stores and traditional POSIX based file storage may help improve latency of SSD based solutions as well.

SYNCHRONOUS 4K WRITE RESULTS

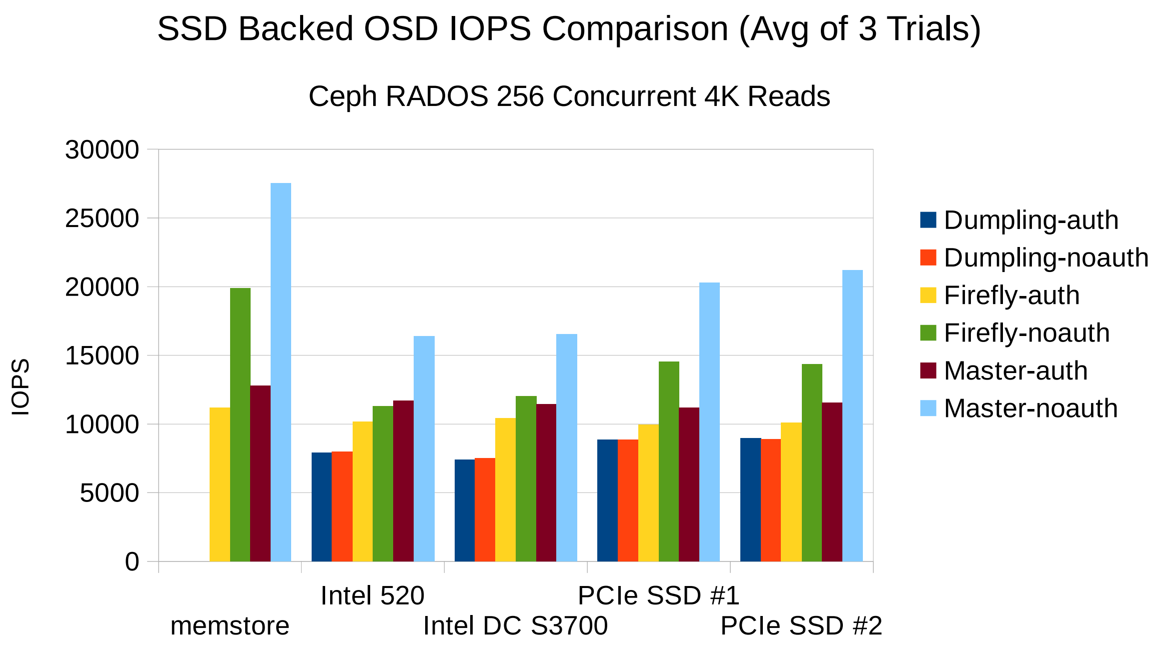


Interestingly, when doing synchronous 4k writes, Dumpling is actually faster that Firefly. Unfortunately the memstore back-end did not exist in Dumpling so it is difficult to determine if this is due to changes in the filestore, client, or something else. An initial theory was that this was due to the sharded thread pool work which improves parallelism at the expense of additional IO path complexity. However, that was not implemented until after Firefly was released. In any event, the most recent development version of Hammer matches or exceeds the performance of Dumpling in all cases.

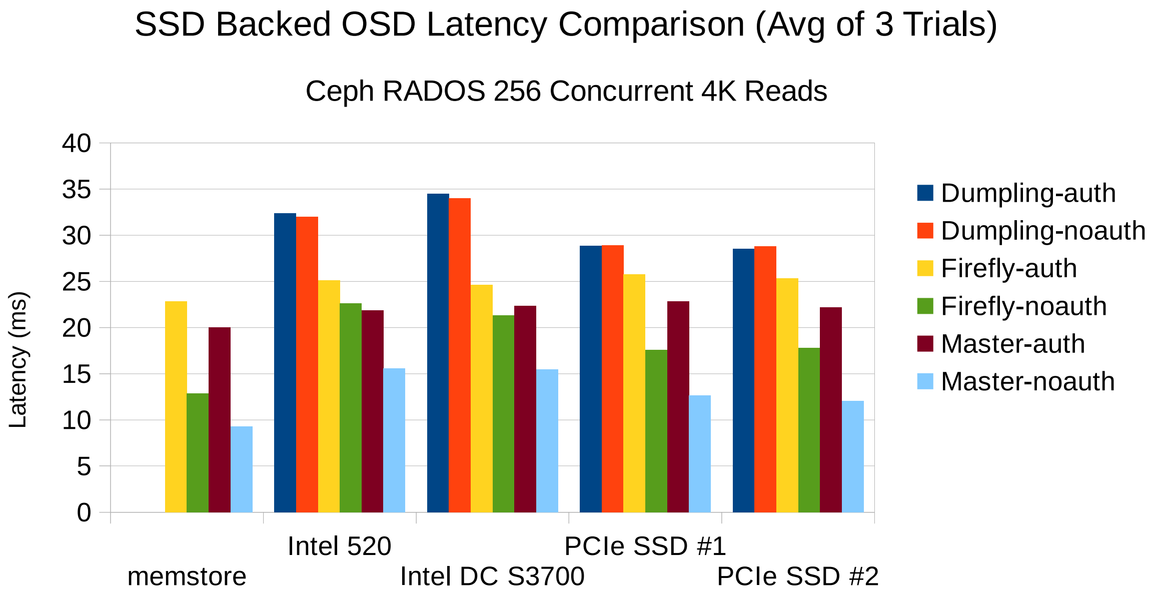


Across the board, disabling authentication reduces synchronous write op latency. Firefly again seems to be lagging both Dumpling and Hammer. Hammer again is showing the best results in these tests both with and without authentication. There is active work happening to continue to improve the write path in Ceph. These results may improve further in subsequent releases.

256 CONCURRENT 4K READ RESULTS

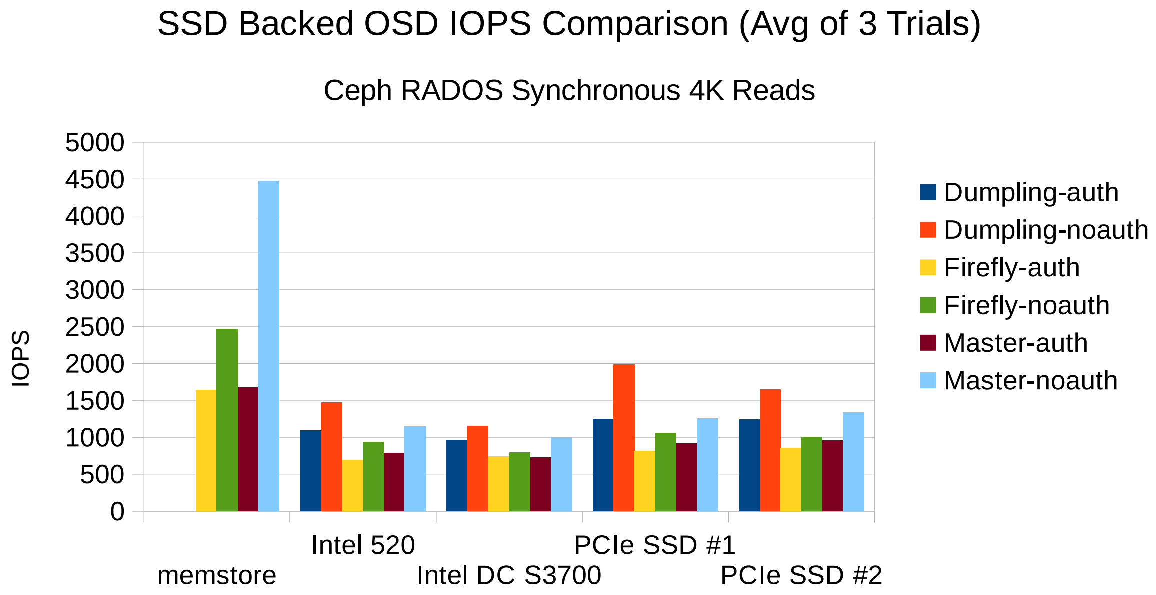


The fastest throughput that Dumpling can achieve on any of these SSDs is roughly 9K IOPS. Firefly improved on those numbers, but Hammer blows both Dumpling and Firefly out of the water. Hammer can achieve nearly 21K read IOPS on a single PCIe SSD backed OSD using the traditional filestore back-end. The effects of authentication are especially interesting. Disabling authentication has virtually no effect in Dumpling, but in later Ceph releases where other bottlenecks have been resolved, the overhead caused by authentication becomes a bigger and bigger performance limitation.

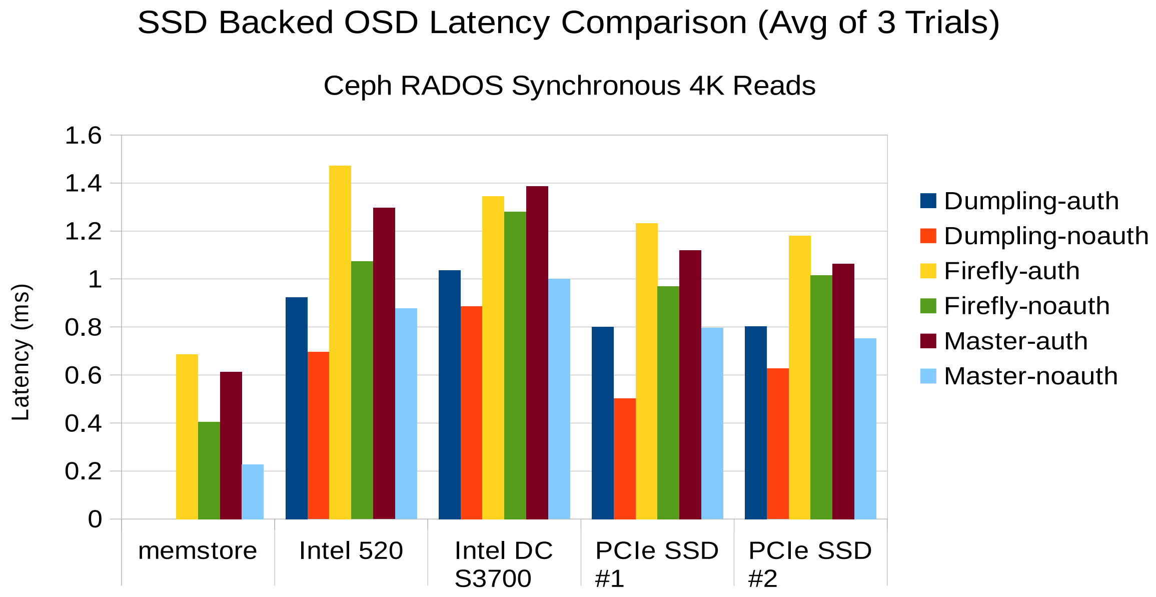


With authentication disabled and 256 concurrent 4K reads, Hammer is able to keep the average op latency to roughly 15ms or less on all SSDs. The memstore back-end is faster, but it appears that in this case improvements to the messenger layer and network layer may be the next lowest hanging fruit.

SYNCHRONOUS 4K READ RESULTS



In a final twist, Dumpling is the fastest release for synchronous 4K reads while Firefly is the slowest. Without extensive analysis it is difficult to know why. The improvements in Hammer help, but are not sufficient to offset the higher latency introduced in Firefly. Despite this, both new releases are significantly faster at processing parallel 4K IOs so this is not necessarily a clear regression. Finally, authentication is again hampering performance, especially with the memstore back-end.



Latency of read operations from the memstore with authentication disabled are an impressive 0.2ms. With the SSD backed OSDs, read operations can be kept to 1ms or less in many cases. The challenge in the future will be to decrease this latency even further while simultaneously improving concurrent IO performance.

CONCLUSION

There are a couple of general conclusions that can be made after looking at this data:

- OSD parallel small IO performance has consistently been improving throughout the last several Ceph LTS releases. In some cases performance is over twice as fast in Hammer as it was in Dumpling.
- While authentication was generally a minor bottleneck in Dumpling, it has become a more major bottleneck on fast SSDs in Hammer. An initial prototype to mitigate some of the performance penalties associated with authentication has been introduced and tested in the Ceph wip-auth git branch.
- Synchronous Read/Write latency generally has only improved slightly or even regressed since the Dumpling release. More investigation will need to take place to determine why this is happening.

Given these initial results, it may be worthwhile to perform additional testing to examine how the performance of the higher level Ceph interfaces have changed over the last several releases. Primarily the Librbd, kernel RBD, and RGW interfaces have all seen changes that could affect how they perform. There are also several additions to Ceph that would be worth testing such as new messenger implementations (an asynchronous TCP implementation that improves on SimpleMessenger and a Libxio based messenger capable of efficiently using either RDMA or TCP). As always, studying performance in complex distributed systems is rarely easy, but with careful analysis improvements can be made.

APPENDIX A. CEPH CONFIGURATION

```
[global]
        osd crush update on start = false
        osd crush chooseleaf type = 0
        osd pg bits = 10
        osd pgp bits = 10
        osd pool default size = 1
#       These are set when authentication is disabled
#        auth client required = none
#        auth cluster required = none
#        auth service required = none
        keyring = /tmp/cbt/ceph/keyring
        log to syslog = false
        log file = /tmp/cbt/ceph/log/$name.log
        rbd cache = true
        filestore merge threshold = 40
        filestore split multiple = 8
        osd op threads = 8
        mon pg warn max object skew = 100000
        mon pg warn min per osd = 0
        mon pg warn max per osd = 32768
#       These are set when testing the memstore interface
#        osd objectstore = memstore
#        memstore_device_bytes = 17179869184
        debug_lockdep = 0/0
        debug_context = 0/0
        debug_crush = 0/0
        debug_buffer = 0/0
        debug_timer = 0/0
        debug_filer = 0/0
        debug_objecter = 0/0
        debug_rados = 0/0
        debug_rbd = 0/0
        debug_journaler = 0/0
        debug_objectcatcher = 0/0
        debug_client = 0/0
        debug_osd = 0/0
        debug_optracker = 0/0
        debug_objclass = 0/0
        debug_filestore = 0/0
        debug_journal = 0/0
        debug_ms = 0/0
        debug_monc = 0/0
        debug_tp = 0/0
        debug_auth = 0/0
        debug_finisher = 0/0
        debug_heartbeatmap = 0/0
        debug_perfcounter = 0/0
        debug_asok = 0/0
        debug_throttle = 0/0
        debug_mon = 0/0
        debug_paxos = 0/0
        debug_rgw = 0/0

[mon.a]
        mon addr = 192.168.10.1:6789
        host = burnupiX
        mon data = /tmp/cbt/ceph/mon.$id

[osd.0]
        host = burnupiX
        keyring = /tmp/cbt/mnt/osd-device-0-data/keyring
        osd data = /tmp/cbt/mnt/osd-device-0-data
        osd journal = /dev/disk/by-partlabel/osd-device-0-journal
```

APPENDIX B. CBT CONFIGURATION

```
cluster:
  user: 'nhm'
  head: "burnupiX"
  clients: ["burnupiX"]
  osds: ["burnupiX"]
  mons:
    burnupiX:
      a: "192.168.10.1:6789"
  osds_per_node: 1
  fs: 'xfs'
  mkfs_opts: '-f -i size=2048 -n size=64k -K'
  mount_opts: '-o inode64,noatime,logbsize=256k'
  conf_file: '/home/nhm/src/ceph-tools/regression/test/memstore/ceph.conf'
  iterations: 3
  use_existing: False
  clusterid: "ceph"
  pool_profiles:
    radosbench:
      pg_size: 1024
      pgp_size: 1024
      replication: 1

benchmarks:
  radosbench:
    op_size: [4096]
    write_only: False
    time: 30
    concurrent_ops: [1, 256]
    concurrent_procs: 1
    osd_ra: [4096]
    pool_profile: 'radosbench'
```