CERN

CERN Ceph Ops

Dan van der Ster, CERN IT Storage Group

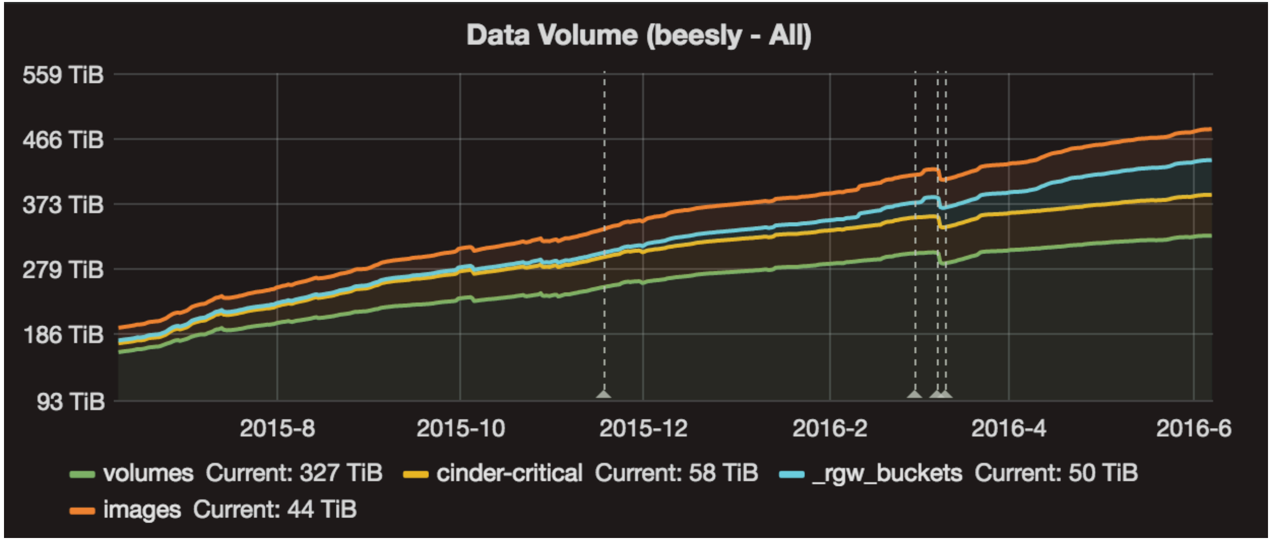
daniel.vanderster@cern.ch

Ceph HEP Day, 13 June 2016

Our Ceph Clusters

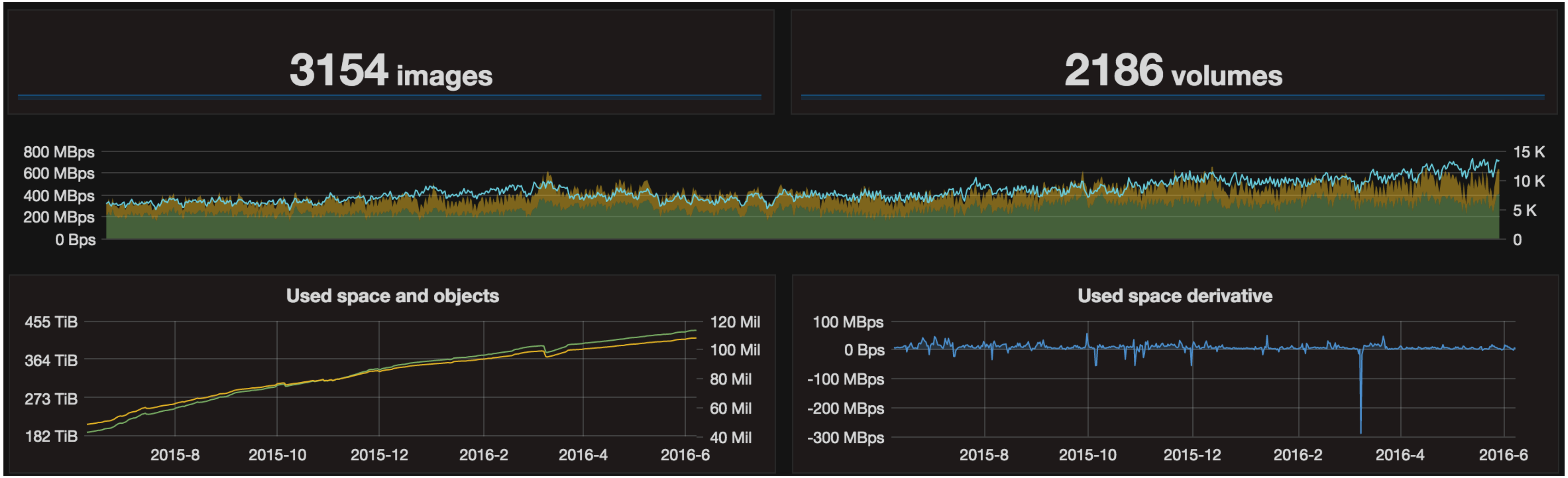
- *Beesly* + *Wigner (3.6 PB + 433 TB, v0.94.7):*
  - Cinder (various QoS types) + Glance + RadosGW
  - Isolated pools/disks for volumes, volumes++, RadosGW
  - Hardware reaching EOL this summer.
- *Dwight (0.5 PB, v0.94.7)*:
  - Pre-prod cluster for development (client side), testing upgrades / crazy ideas.
- *Erin (2.9 PB, v10.2.1++)*:
  - New cluster for CASTOR: disk buffer/cache in front of tape robots
- *Bigbang (~30 PB, master)*:
  - Playground for short term scale tests whenever CERN receives new hardware.

Growth of the *beesly* cluster



From ~200TB total to **~450 TB of RBD + 50 TB RGW**

OpenStack Glance + Cinder



3154 images

2186 volumes

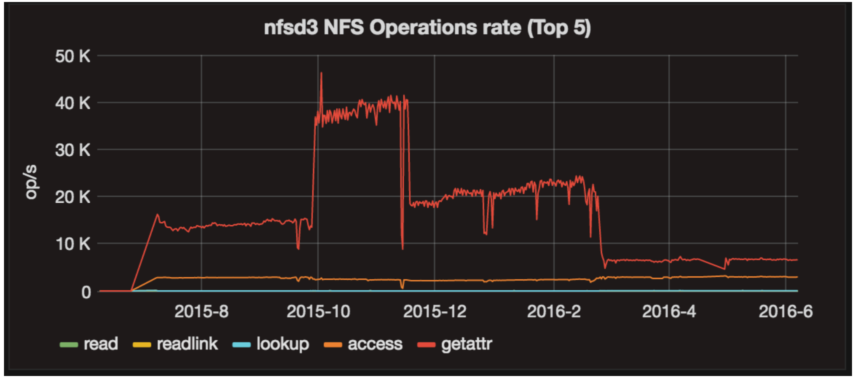
OpenStack is still Ceph’s killer app. We’ve doubled usage in the past year.

NFS on Ceph

- ~50TB of servers (28 in total):
  - OpenStack VM + RBD volume
  - CentOS 7.2 with ZFSonLinux 0.6.5.x

- Used for Puppet masters, Gitlab, Twiki, LSF, BOINC, ElasticSearch, MICroelectronics $HOME, …

- *Not highly-available, but…*
- cheap, thinly provisioned, resizable, easily add new filers
- disaster recovery via zrep to 2nd data centre
- (ZoL stability is pretty good, though it still locks up from time to time)



Example: ~25 puppet masters reading node configurations at up to 40kHz

Provisioning Large Clusters

- We still use puppet-ceph (originally from eNovance, but heavily modified)
  - Install software/configuration/tuning, copy in keys, but ***don't touch the disks***

- New: **ceph-disk-prepare-all**
  - Inspect the system to discover empty non-system drives/SSDs
  - Guess an optimal layout (with or w/o dedicated journals, then map journals to OSDs)
  - https://github.com/cernceph/ceph-scripts/blob/master/ceph-disk/ceph-disk-prepare-all

- ceph-disk prepare **--no-locking**
  - Currently ceph-disk prepares one disk at a time, so large servers take *hours* to prepare.
  - PR to remove this global prepare lock: https://github.com/ceph/ceph/pull/8829

- Deploying a large cluster takes one afternoon.

Hardware Replacement

- Starting next month we will begin replacing our 48 *beesly* servers (each with 20x3TB drives) with 48 new servers (each 24x6TB drives)

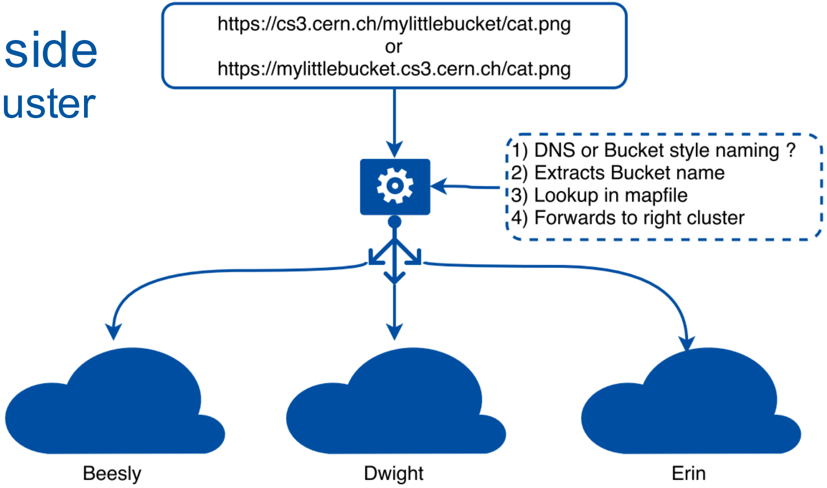
- **How not to do it…** add new OSDs and remove old OSDs all at once
  - Would lead to massive re-peering, re-balancing, unacceptable IO latency.

- **Our plan:** gradually add new & remove old OSDs
  - How quickly we can do this: OSD-by-OSD, server-by-server, rack-by-rack?
- Considerations:
  - We want to reuse the low OSD id's (implies add/remove/add/remove/… loop)
  - We don't want to have to babysit (need to automate the process)
  - **We want to move rgw pools to another cluster!**

- This is an area where high level cluster management tools could help.
  - Watching Apache Mesos work with interest.

RadosGW: One endpoint, many clusters

- `*.cs3.cern.ch` is a DNS load balanced alias
- HaProxy (>=1.6) listens on public side
  - Mapping file from bucket name to cluster
- RadosGW listens on loopback


https://cs3.cern.ch/mylittlebucket/cat.png
or
https://mylittlebucket.cs3.cern.ch/cat.png
1) DNS or Bucket style naming ?
2) Extracts Bucket name
3) Lookup in mapfile
4) Forwards to right cluster
Beesly
Dwight
Erin

https://gist.github.com/cernceph/4a03316a31ce7abe49167c392fc827da

Bigbang Part II

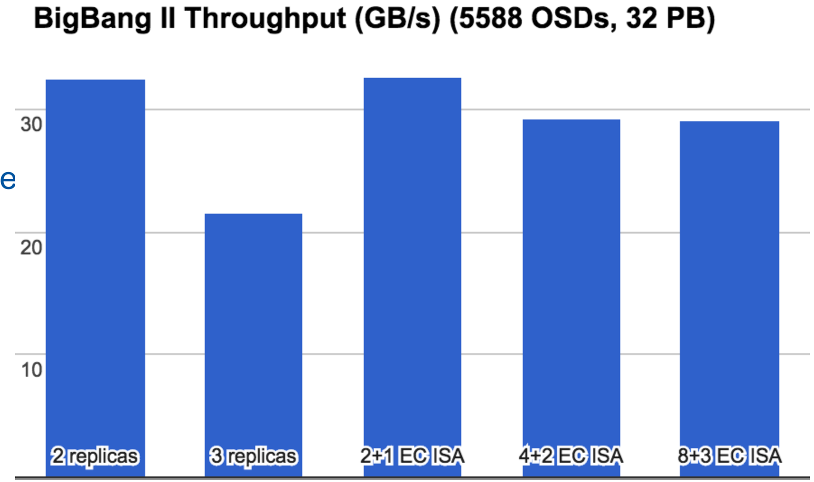
- Bigbang II is a second 30PB test during May 2016
  - Previous scaling issues seem all solved. Create/delete 128k PG pools. Three mons w/ 5588 OSDs is working well.
- Benchmarking: ~30GB/s is doable (internally)
  - On these large clusters we replicate/EC across core routers. (broke a line card, taking out 6 racks)
  - Network limits the throughput.. Should investigate more clever replication (shingled…?)
- New jewel features:

```
ms type = async
```

  - reduces threads, eliminates tcmalloc thrashing, and noticeably decreases OSD memory usage
  - Rare peering glitches, hopefully fixed in next jewel.

```
op queue = wpq
```

  - better recovery transparency
  - Seems to be working.



~~FileStore~~ BlueStore

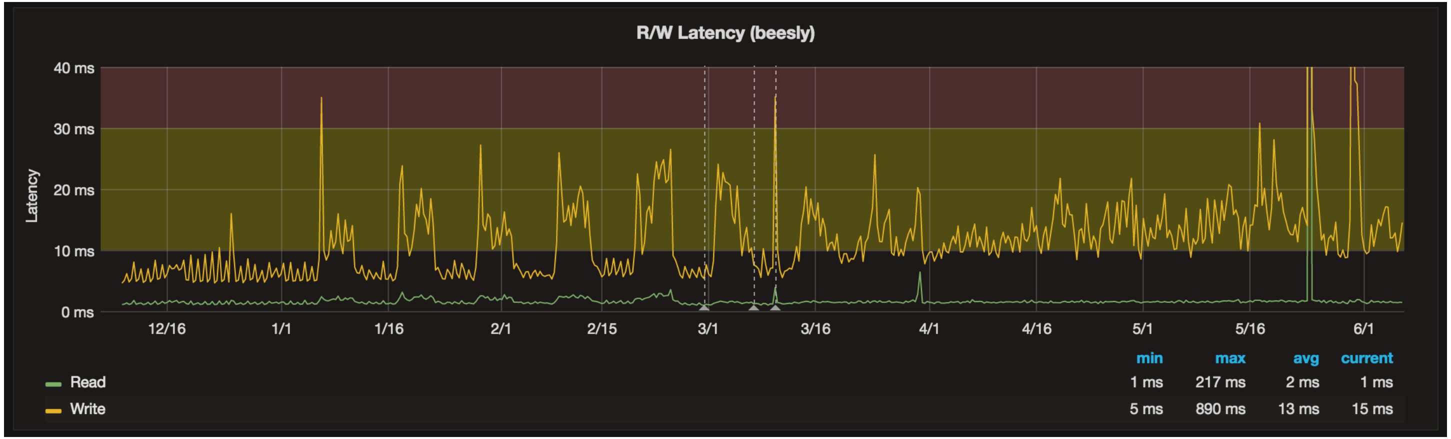
- FileStore is the stable supported OSD backend.
- **XFS-only**. Don't bother with btrfs/ext4.
- Double write penalty: first write synchronous, 2nd async.
- Trivial to tune for max write bandwidth: filestore max sync interval = 60
- RHEL 6->7 XFS upgrade issue (64k directories)
  - We *cannot* upgrade our OSDs to EL7

- Bluestore is the promised solution to the double write penalty (and all the other XFS-induced seeks)
- Tested with one host out of 18: loadavg on the bluestore machine was lower. Seems.. pretty ok ☺
- But after a short test I saw inconsistent objects, so aborted the test.
- We really need BlueStore to work!!

Ceph exposes the real performance of your hardware (data must be written durably)

Mixed read/write workloads will always be a challenge. (Because we cannot cheat and buffer writes)

Block Storage: Latency



	min	max	avg	current
Read	1 ms	217 ms	2 ms	1 ms
Write	5 ms	890 ms	13 ms	15 ms

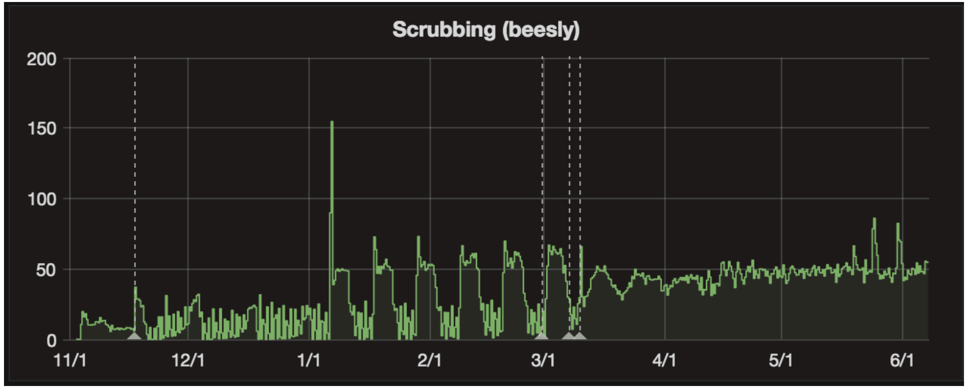
Single threaded users are latency bound.

KPI: 10ms 4kB write latency
SSD journals still essential.

Leverage extra SSD capacity?
Trying different flash caching options.

Scrubbing

- Scrubbing has been a problem historically
  - Too many concurrent scrubs *kills* latency
- Hammer / jewel randomize the scrub schedule. See plot →
- Jewel scrub IOs go via the OSD op queue
  - Better fair sharing of disk time ☺
  - But still needs tuning ☹
  - Need to throttle on high-BW clusters →

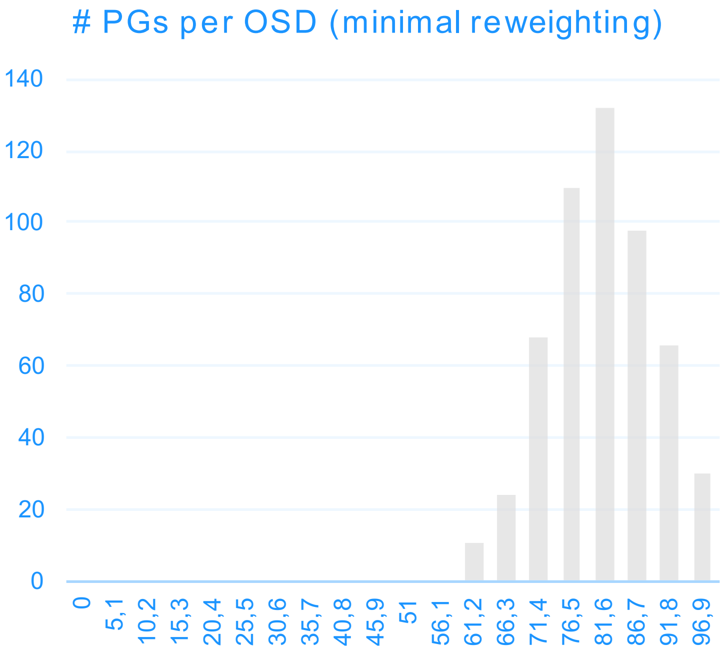


Minimal scrubbing:

```
osd scrub chunk max = 1
osd scrub chunk min = 1
osd scrub priority = 1
osd scrub sleep = 0.1
```

Balancing OSD data

- We need to fill our clusters: Imagine not being able to use 10% of a 10PB cluster !!
- *Best practises* suggests 100-300 PGs per OSD
  - But more PGs == increased RAM, so we're cautious
- Hammer 0.94.7 and Jewel have a new (test-) reweight-by-utilization feature
  - Test reweight before making changes
  - Change only 4 OSDs at a time, only +- 0.05
  - This is a good workaround, but it decreases the flexibility of the OSD tree
- Q: Current reweight is between [0,1]. Why don't we allow reweight > 1? It should help boost underutilized OSDs



Papercuts

- OpenStack-related:
  - Thanks to libnss / cephx crashes (< 0.94.7) our IT colleagues now know that Ceph exists
  - ceph-mon IPs hard-coded in each VM's libvirt xml
  - No live-migration for VMs with attached Ceph volumes. (wb worries)
  - Tracking connected client versions is hard. We **don't know if we can enable firefly tunables :-/**
  - Large clusters need increased ulimits; causes endless confusion
- PG repair gymnastics:
  - Repair often doesn't start because of osd_max_scrubs limits
  - Workaround: Disable scrubbing, increase max scrubs, ceph pg repair, reset max scrubs, enable scrubbing

CERN