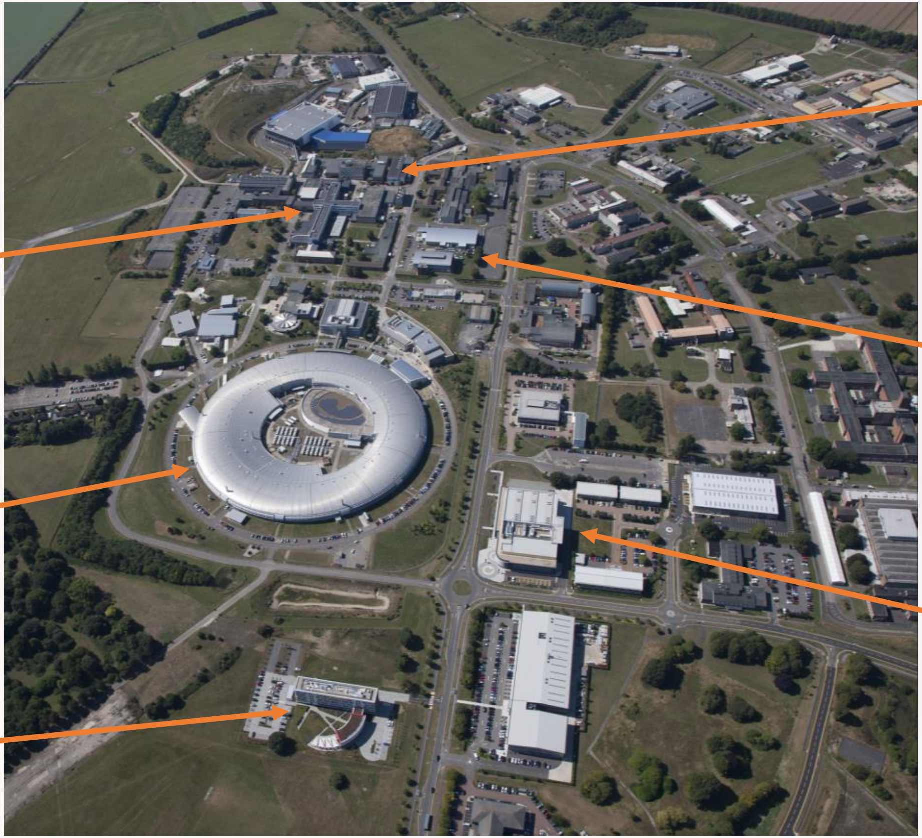
CEPHALOCON APAC 2018
THE FUTURE OF STORAGE
22-23 March 2018 | BEIJING

Erasure Code at Scale

Tom Byrne

Science and Technology Facilities Council

Who are we?



Central Laser Facility

Particle Physics Department

Scientific Computing Department

Diamond Light Source

RAL Space

European Space Agency

Science & Technology Facilities Council

Worldwide LHC Computing Grid

- Computing resources for the WLCG are provided by over 170 computing sites worldwide.
  - RAL is one of the 5 biggest sites providing 30PB of disk, 30PB of tape storage and 25,000 cores

- The raw data from the experiments is stored on tape at CERN and at another site. This means that while it may be time consuming is possible to regenerate data if it is lost.

42 countries

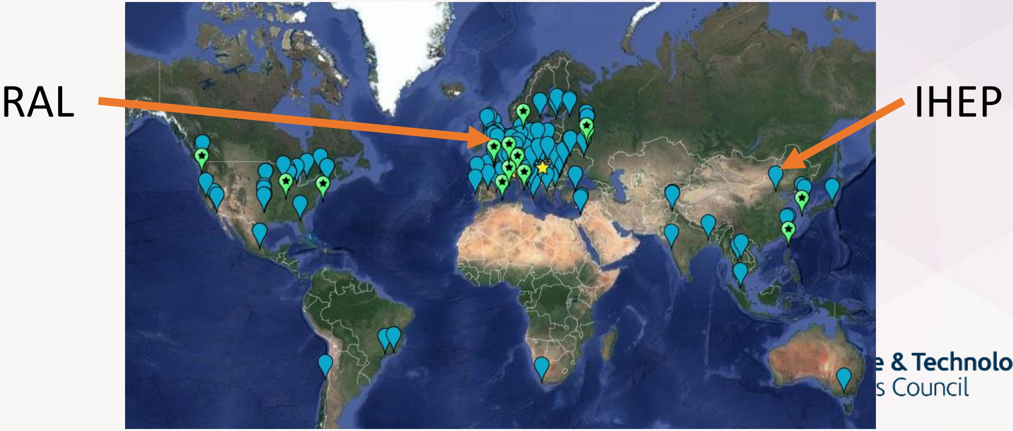
170 computing sites

2 million jobs run every day

750,000 computer cores

400 petabytes on disk

400 petabytes on tape

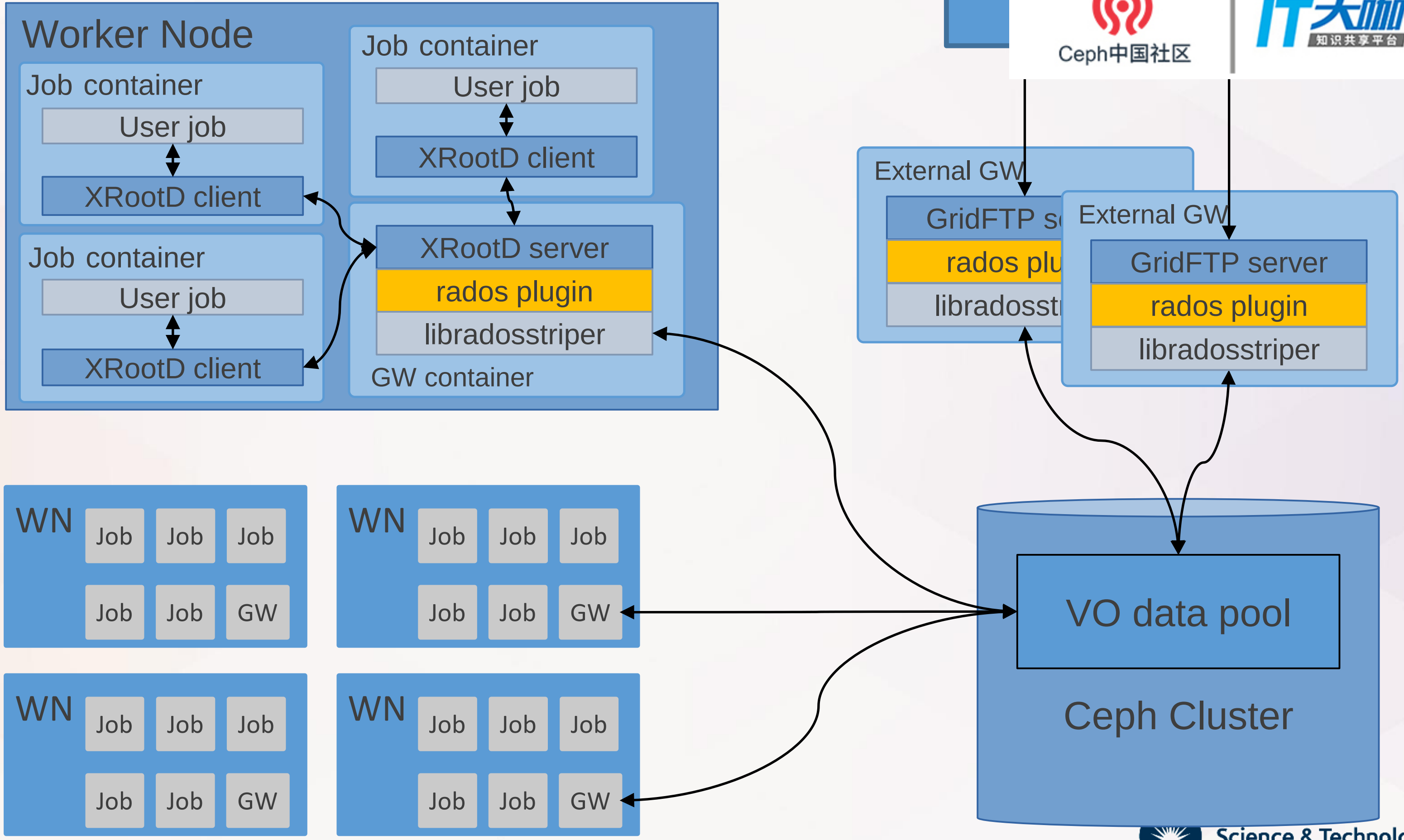


Grid storage requirements

- Storage at grid sites has traditionally been 'POSIX like'
  - No real need for this, each experiment keeps track of files, Tier-1 storage is already mostly used as an object store
- HTC, not HPC
  - No requirement for low latency
  - LHC experiments can easily parallelise their workflows, so overall throughput for data transfers is the key measurement
- Grid protocols are used for transfers
  - For transfers to other sites a protocol called GridFTP is used
  - For jobs analysing data a protocol called XRootD is used

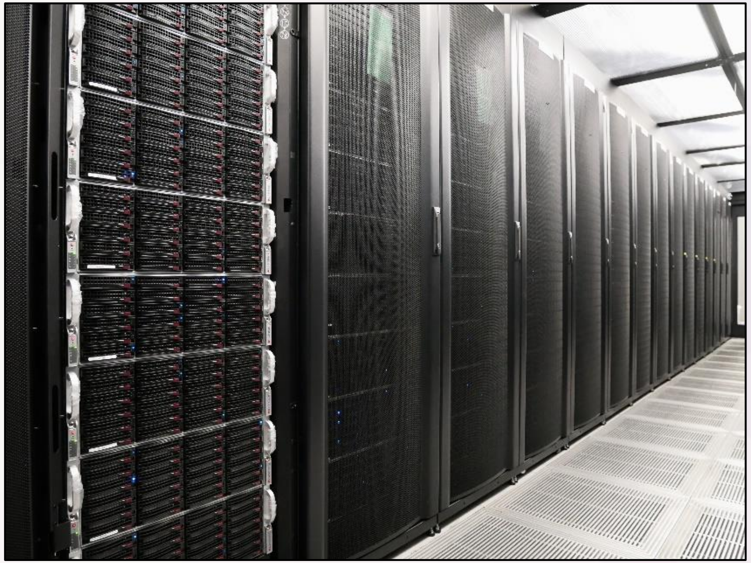
The beginnings of ‘Echo’

- In 2014, we started looking into Ceph for replacing our disk only storage.
- Very tight £/TB limit which necessitated EC and large storage nodes
  - No SSDs for journals
- We wrote plugins for the grid protocol servers on top of libradosstriper
  - We did try getting the experiments to use S3, but limited success so far
- **E**rasure Coding – **C**eph – **H**igh Throughput – **O**bject Store


Worker Node
Job container
User job
XRootD client
Job container
User job
XRootD client
Job container
User job
XRootD client
XRootD server
rados plugin
libradosstriper
GW container
External GW
GridFTP s
rados plu
libradosstr
External GW
GridFTP server
rados plugin
libradosstriper
WN
Job
Job
Job
Job
Job
GW
WN
Job
Job
Job
Job
Job
GW
WN
Job
Job
Job
Job
Job
GW
WN
Job
Job
Job
Job
Job
GW
VO data pool
Ceph Cluster

Meet Echo

- 60 storage nodes with ~13 PB total raw space
  - 36 6TB disks (216TB ‘lumps’)
  - 40 HT cores
  - 128GB ram
  - 4x10Gig networking
- 5 monitors
  - SSD for levelDBs
- 5 external gateways
  - 4x10Gig networking
- Luminous (started as Jewel), with mostly filestore OSDs



Erasure Coding settings

- k=8, m=3
  - Gives us a 38% overhead with security against 3 simultaneous failures
  - We initially went for k=16 but we couldn’t keep the cluster stable.
- plugin=jerasure
  - Default, most documentation and most widely used
- technique=reed_sol_van
  - Couldn’t use any of the m=2 optimised versions
  - cauchy required careful tuning
- crush-failure-domain=host

PGs and peering

- 1024/2048 placement groups per pool
  - Low numbers of large (2TB!) PGs, makes single PG operations slow.
  - Aiming to go up to 4096/8192 PGs for the data pools soon, 1TB per PG seems reasonable.
  - The recommendations for PGs counts aren’t correct for Erasure Coding, especially at scale.

- We’re not seeing any performance issues with these large PGs
  - Peering after a node reboot takes ~1 minute

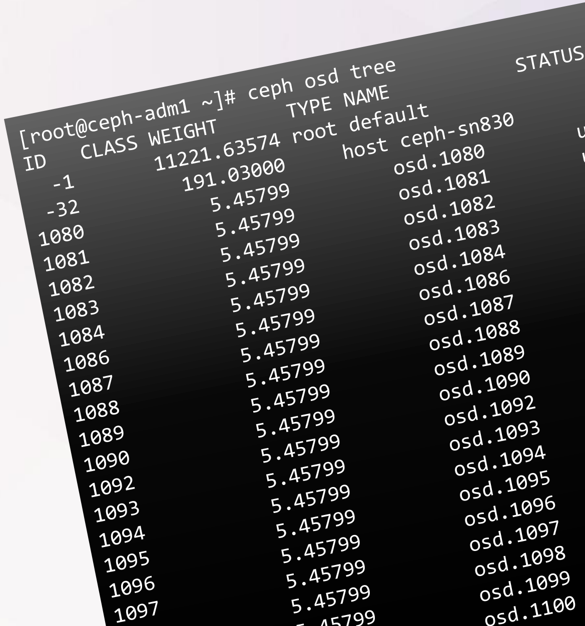
- Tuning to stop OSDs being marked out so aggressively
  - mon_osd_min_down_reporters=6 (from 1)
  - osd_heartbeat_grace=90 (from 20)
  - osd_heartbeat_interval=10 (from 6)

Crush map

- We decided to leave racks out of the crush map
  - Reduces complexity
  - Much simpler crush map means less chance of errors

- Much better than existing system
  - Being able to take down a machine without affecting availability is great from an operational standpoint.

- Given our SLA (98% availability), losing some data availability due to the loss of power or network to entire racks is acceptable
  - Hasn’t happened yet!



```
[root@ceph-adm1 ~]# ceph osd tree
ID   CLASS WEIGHT      TYPE NAME                    STATUS
  -1       11221.63574 root default
 -32         191.03000     host ceph-sn830
1080           5.45799         osd.1080
1081           5.45799         osd.1081
1082           5.45799         osd.1082
1083           5.45799         osd.1083
1084           5.45799         osd.1084
1086           5.45799         osd.1086
1087           5.45799         osd.1087
1088           5.45799         osd.1088
1089           5.45799         osd.1089
1090           5.45799         osd.1090
1092           5.45799         osd.1092
1093           5.45799         osd.1093
1094           5.45799         osd.1094
1095           5.45799         osd.1095
1096           5.45799         osd.1096
1097           5.45799         osd.1097
```

Striping

- Objects range from 10s of MBs to 10s of GBs in size

- Libradosstriper stripes objects into 64MB chunks
  - Leads to 8MB shards on disk
  - Lots of work has been done to optimise file transfers speed based on stripe size.

- In testing, large k+m pools and default stripe size lead to hilariously small/distributed objects
  - A 1GB object ends up as 3K shards. Object might be present on every OSD!
  - Losing a single PG turns into a complete disaster

Living with Echo

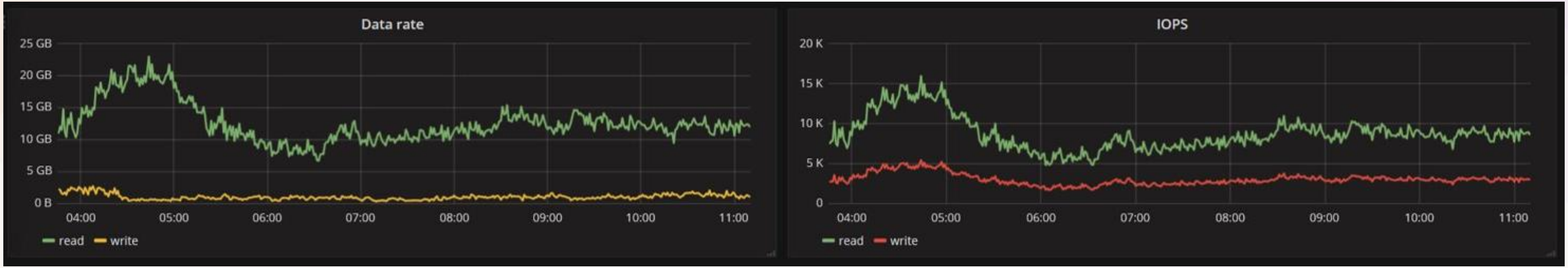
How has Echo been to maintain operationally?

ceph Performance

- In general, Echo's performance has been good
  - Sustains >10 GB/s, >10k IOPS happily
  - Ceph has handled everything we have thrown at it, no sign of a bottleneck yet.

- Most of the problems have been on the external gateways
  - memory usage, port exhaustion due to misconfiguration

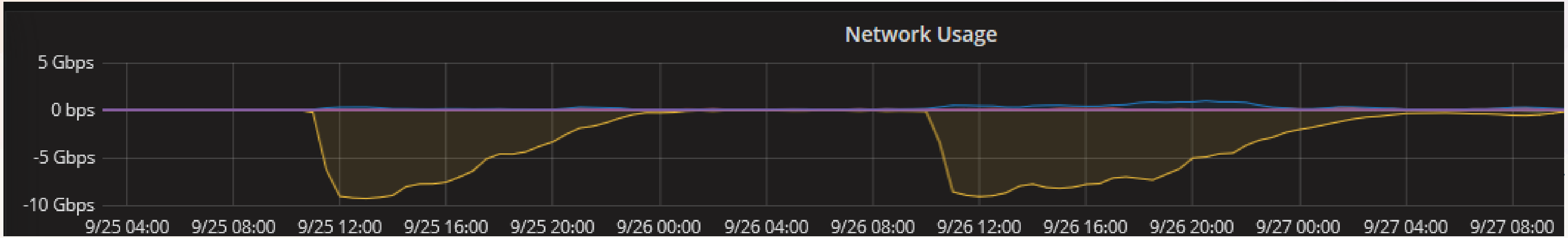
- Gateways on worker nodes has worked spectacularly well



Backfill performance

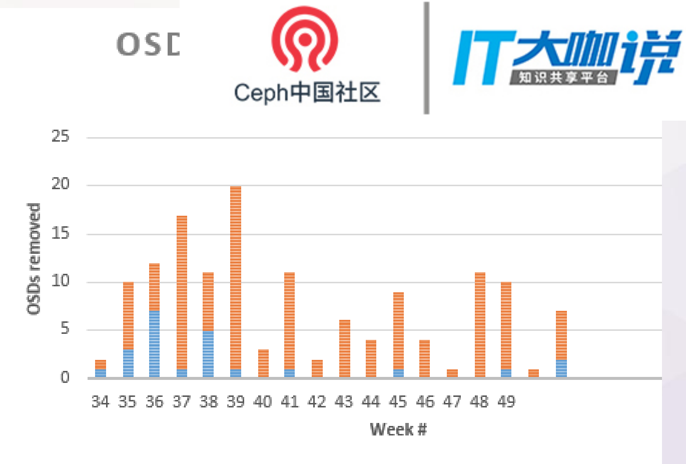
- Issues impacting cluster performance were seen with default backfill settings, reducing number of objects per scan helped
  - osd_backfill_scan_max=64 (from 512)
  - osd_backfill_scan_min=16 (from 64)

- The bottleneck on backfilling speed when adding nodes to Echo has been the new nodes networking
  - Will happily saturate the 10gig cluster network interface



Speaking of backfilling...

- In August we encountered an EC backfill bug when adding 30 new nodes to the cluster.
  - http://tracker.ceph.com/issues/18162



- A read error on an object shard on any existing OSD in backfilling PG will:
  - crash the primary OSD, and the next acting primary, and so on, until the PG goes down
  - Misdiagnosis of the issue lead to the loss of an Atlas PG, 23,000 files lost

- Any backfilling in the last 6 months has required babysitting
  - High disk failure rate coinciding with this bug caused an operational nightmare
  - Leaving disks with any pending sectors in is risky
  - Bug now fixed in 12.2.3 ☺

Science & Technology
Facilities Council

Inconsistent PGs

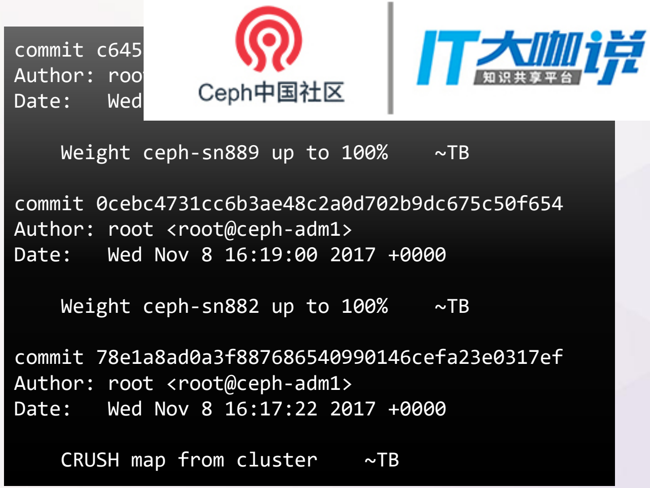
- We get a lot of inconsistent PGs
  - 5-10 a week
    - operational pain!
  - 80% are due to bad sectors on otherwise healthy disks

- These inconsistencies are inevitable and harmless on high *k+m* EC pools
  - Fixed with a simple ‘ceph pg repair’
  - HEALTH_ERR seems like overkill
    - Our healthy cluster is in this state for 10+ hours a week, monitoring ‘ceph health’ for callouts is becoming tiring.

Inconsistent PGs (cont.)

- This is one of the issues that is new (for us) with Ceph.
  - RAID cards have been very good at background scrubbing, repairing errors
  - We have had to remove hundreds of disks that were not fit for replacement (by the vendors metrics)
    - This was exacerbated by the EC backfill bug
- Dealing with health disks with a few media errors has been a unexpected workload on us
  - We have had to spend time on a disk recycling scheme as a short term solution but that is not sustainable long term – Ceph has to deal with the media errors.
- Ceph has been very fragile in this regard
  - Lots of layers between spinning rust and the storage logic, so it doesn't have the same ability to try and fix issues
    - Can this be improved with BlueStore?
  - Healthy disks do occasionally pick up media errors. Are we really going to have to pull them out, reformat them and then put them back into the cluster?

CRUSH map management



```
commit c645
Author: roo
Date:   Wed

    Weight ceph-sn889 up to 100%    ~TB

commit 0cebc4731cc6b3ae48c2a0d702b9dc675c50f654
Author: root <root@ceph-adm1>
Date:   Wed Nov 8 16:19:00 2017 +0000

    Weight ceph-sn882 up to 100%    ~TB

commit 78e1a8ad0a3f887686540990146cefa23e0317ef
Author: root <root@ceph-adm1>
Date:   Wed Nov 8 16:17:22 2017 +0000

    CRUSH map from cluster    ~TB
```

- We made the decision to manually manage Ceph
  - configuration management stops at ceph.conf
  - ceph-deploy is fine for bulk OSD creation

- Bulk changes to cluster layout done via manual edits
  - OSD addition, removal and reweighting for whole hosts
  - A few helper scripts (awk) to do host weight calculations, bulk reweights, and diffs between versions
  - crush map changes tracked in git

- The second half of the Echo was deployed this way last year
  - Planning to do the same for the next generation (and the following generation) this year
    - 120 more storage nodes to go in this year

CRUSH map management (cont.)

- Bulk addition/removal of 1000s of OSDs using ‘ceph osd crush add/rm/reweight’ was not so great
  - Prolonged peering, 1000s of new crush maps being pushed out, no ability to roll back.
  - manual crush map edits has a much lower impact on the cluster
    - error prone however

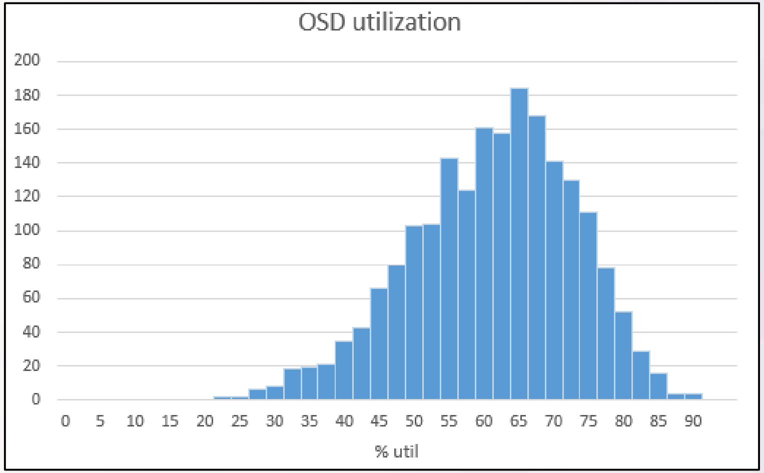
- Would be good to have a transactional, session-y tool to do this
  - Start a new ‘session’
  - make changes (crush add/rm/reweight) offline
  - and then execute as one crushmap change (and rollback if needed)

Data distribution

- Echo has a large spread in OSD utilization

- Reweight by utilization is effective in reweighting the ‘full’ OSDs down
  - Still left with a long tail of empty OSDs
  - Currently run once a week

- Balancer mgr module seems promising
  - Using the crush-compat mode on our development cluster with good results



Conclusion

- Echo’s first year in production has been encouraging
  - Ceph has handled everything we have thrown at it, and exceeded expectations while doing it
- Echo is growing!
  - Dealing with disk errors as the cluster scales and the hardware ages is going to be interesting
  - Better tools for crush map management would be helpful
- Erasure coding on large storage nodes is proving to be an effective ‘cheap’ storage solution for our use case
  - EC was definitely a gamble when we started, but we have a great deal more confidence now

ceph

Ceph中国社区

IT大咖说
知识共享平台

Thank you

Science & Technology
Facilities Council