Experiences with ceph object store at MSI

Graham T. Allan
Minnesota Supercomputing Institute
Ceph at MSI - timeline

Test cluster
Early 2014 - Emperor

Test cluster reborn
Early 2016 - Hammer - Jewel - Luminous - Mimic

Production object store "Tier 2"
Late 2014 - Firefly - Hammer - Jewel - Luminous
3.5PB - EC object store - S3 and globus

Cloud storage "Stratus"
Late 2016 - Jewel - Luminous
1.5PB block storage with some object store
Ceph at MSI

Primary storage, scratch space - Panasas

HPC clusters  Interactive HPC  External hosts  Globus

Object storage - "Tier 2"

Block storage for VMs

Openstack cloud
Evolution of Tier-2 storage

2014-2015
- 7 Supermicro osd servers with 60x 6TB HDD, 12x 480GB SSD

2016
- additional 2 HPE Apollo osd servers, 60x 8TB HDD, 8x 480GB SSD

3 mons (VMs)
4 load balanced rgw nodes

Early 2018...
- Approaching 80% capacity
- Hardware warranty end

Time for refresh - goals...
- larger number of smaller hosts
- filestore -> bluestore
- dual-tree crush map -> unified with device classes
- ubuntu 14.04 -> Centos 7
- fstab+init driven osds -> ceph-volume
Refreshed storage architecture

Build on experience with Stratus...

- 15x HPE Apollo 4200 storage nodes
- Also retain existing two Apollo 4510 nodes

HDD OSDs: 4.5PB raw
  - Bluestore with co-located WAL/DB
  - 4:2 erasure coded

SSD OSDs: 35TB raw
  - object store indexes

mons also migrated from VMs to physical hardware (HPE moonshot cartridges)

Server provisioning
- kickstart+puppet
- OSDs created using basic "ceph-volume"
Revamped crush map

Rebuild using device class-based rules
● Started with two separate trees for hdd and ssd
● convert so all beneath a common root

New CRUSH rules
3 main crush rulesets to convert to device classes
● replicated_hdd
● replicated_ssd
● ecprofile42_hdd
Starting data migration

Early August 2018…

- About half the new hardware delivered and installed…
  - Space crunch - now at 85% utilization
- Update crush_ruleset for replicated pools
  - Gradually increase weight of new OSDs
  - All backfilled within a few minutes to days.

What about the EC pool?
- most of our data is here ~2PB
EC pool device class change

- Process not really documented (explicitly, at least)
- Creation of new crush rule also required a new EC profile
  - But, "can't change the EC profile for a pool"?
  - Does this just apply to the k+m values? New and old both 4+2
  - So is this change safe? (spoiler: Yes)

Original profile (firefly)
- crush-failure-domain=host
- directory=/usr/lib/x86_64-linux-gnu/ceph/erasure-code
- k=4
- m=2
- plugin=jerasure
- technique=reed_sol_van

New profile (luminous)
- crush-device-class=hdd
- crush-failure-domain=host
- crush-root=default
- jerasure-per-chunk-alignment=false
- k=4
- m=2
- plugin=jerasure
- technique=reed_sol_van
- w=8
Another EC pool quirk

Noticed that "min_size" was set to 2 for our ec 4+2 pools...
What does min_size mean for an ec pool?

min_size
**Description:**
Sets the minimum number of replicas required for I/O. See [Set the Number of Object Replicas](#) for further details. Replicated pools only.

It does really have the same meaning for ec pools as replicated...
- min_size=2 perhaps inherited from early pool creation (firefly)?
The value made no sense to me, so I set it to 4.
Migrating the EC pool

Finally started 5th September 2018

- New osds already at their final crush weight - very heavy backfill load.
- Mitigated by reducing osd max backfills, osd recovery max active, osd max recovery threads

- osds on the older systems would often die
  - no supervisor like systemd to restart them - need extreme babysitting
  - Mostly suicide timeouts etc
  - Several times an entire node (30 osds) would fail at once
- mons ran out of disk space (db grew to ~40GB)
  - db never trimmed due to degraded pgs
- After 36 hours backfill, the cluster suffered a peering storm.
  - Finally resolved by stopping all osds clusterwide, then restarting.
Cleanup after the storm...

pg stuck incomplete after hdd failure

Peering flaps meant the set of active OSDs changed rapidly
  ● "min_size=4": pg became active with only 4 osds - then we lost one.
  ● Fortunately no writes to the pg during the peering storm, so past intervals were consistent...
  ● Declaring the dead osd as "lost" brought the pg back to life.

● pool min_size should be k+1 (5)
Final challenges

another pg stuck incomplete...

OSDs would crash when starting backfill
  - 3 different objects implicated across various osds
OSDs would stay running if "nobackfill" set
  - Map s3 objects to the suspect filestore objects, and download via s3.
  - On one file, calculated etag didn’t match - determined corrupt -
    deleted via s3.
Several hours later all filestore shards were still there; osd would still crash
on backfill.

Re-enabled backfill again next day - no crash!
  - Magic? rgw queues object deletion for later processing -
eventually the corrupt object was removed.
Calm waters ahead…?

Another 3 weeks from resolving these final problems, to complete the data migration and retire old osd nodes.
The EC pool migration process took ~2 months total.

Still have HEALTH_WARN because of large omap objects - Some buckets will need to be resharded to resolve this...
Bluestore tuning still to come…
Some Lessons

Migrating the EC pool separately after all new OSDs were active was probably a mistake.

- Incorrect min_size on pool compounded errors

Easy to trigger cluster flapping by making changes too quickly - eg when deleting old storage nodes and osds.

- Change OSD crush weights gradually with a small delay between each change
Thank You

Any Questions?
Challenges with s3 use in HPC - data sharing

Data storage at MSI is based around research groups with a P.I.

Challenges with S3
- No concept of groups
- No easy equivalent of a chown
- Data lifecycle - P.I. has to have ownership of group user data
- Object acls via s3cmd too hard to use - no inheritance

Migrate towards primary allocation to P.I. (120TB), small allocations to group members.
- P.I. shares group buckets to members using bucket policies (via helper scripts)
- Still testing this with a handful of groups.
- Issues with this (e.g., lack of visibility in bucket lists)