University of Zurich[UZH]

CephFS: looking for the Swiss Army knife of POSIX filesystems

Mattia Belluco
mattia.belluco@uzh.ch
S3IT - Services and Support for Science IT

What is S3IT?

A partner for data and compute-intensive science:

- **Enable** researchers and projects to run simulations and data analysis.
- **Develop** tools to integrate, automate and scale scientific use cases.
- **Provide** access to *innovative* infrastructures and technologies.

Infrastructure

We currently provide access to:

- An openstack deployment with 420 compute nodes.
- 2 Ceph clusters of 5 and 1.7 PB of raw capacity respectively.
- Several HPC-like resources.
- A time share on the CSCS Piz Daint super computer in Lugano.

More detailed information at:
https://www.zi.uzh.ch/en/teaching-and-research/science-it/infrastructure.html

Evolution of our HPC resources

2014-2016

- SGI UV 200
  - 96 cores
  - 4 TB of RAM
- HA NFS fileserver
  - 72 $\times$ 2TB disks
  - FC connection

Evolution of our HPC resources

2017-2018

- 6 nodes
  - 384 cores
  - 18 TB of RAM
- 5 nodes
  - 40 nVidia K80 GPUs

- HA NFS fileserver exporting `/home`
  - 72 $\times$ 2TB disks
  - FC connection
- ZFS fileserver w JBOD exporting `/scratch`
  - 192 $\times$ 8TB disks
  - SAS expander

Additional nodes were expected for the end of 2018:
it was increasingly clear that our Storage System needed an overhaul.

Requirements for a new data storage system

- POSIX compliance
- Reliability
  - at least comparable to RAID 6
- Expandability/scalability
  - the main pain point of the previous system
- Performance
  - HPC users have certain expectations
- Cheap
  - avoid expensive licenses

Nice-to-have

- Decoupled
  - a surge of traffic to a specific filesystem (or directory) should not cause a degradation in other parts of the system.
- Quota support
  - to avoid users filling up the whole filesystem.
- Have a mechanism to give users access to their shares from machines we don't directly control

The most important requirement: flexibility

Our requirements are
constantly changing:
we need a flexible tool to keep
up with them.

Why Ceph

We have been using Ceph since 2015: proven track record. Reliability and expandability/scalability are not a concern.

New Luminous release packed with features

- BlueStore backend:
  - Full data and metadata checksums of all data stored by Ceph
  - Erasure coded pools with full support for overwrites
- CephFS:
  - Multiple active MDS daemons dynamically adjustable at runtime.
  - CephFS directory fragmentation.
  - Directory subtrees pinning.

Why Ceph

We have been using Ceph since 2015: proven track record. Reliability and expandability/scalability are not a concern.

New Luminous release packed with features

- BlueStore backend:
  - Full data and metadata checksums of all data stored by Ceph
  - Erasure coded pools with full support for overwrites
- CephFS:
  - Multiple active MDS daemons dynamically adjustable at runtime.
  - CephFS directory fragmentation.
  - Directory subtrees pinning.

Why Ceph

We have been using Ceph since 2015: proven track record. Reliability and expandability/scalability are not a concern.

New Luminous release packed with features

- BlueStore backend:
  - Full data and metadata checksums of all data stored by Ceph
  - Erasure coded pools with full support for overwrites
- CephFS:
  - Multiple active MDS daemons dynamically adjustable at runtime.
  - CephFS directory fragmentation.
  - Directory subtrees pinning.

Testbed

We started small end of 2017:

- 3 Monitor nodes
- 3 nodes
  - SSDs for journals
  - 72 OSDs on spindles
  - Jewel release
    - CephFS finally stable.
    - Only one release away from our production cluster

Synthetic workloads to mimic the code run by our users

- lots of metadata operations
- moving big files
- concurrent access/modifications of files from multiple clients

The first issues began to surface...

Testbed

We started small end of 2017:

- 3 Monitor nodes
- 3 nodes
  - SSDs for journals
  - 72 OSDs on spindles
  - Jewel release
    - CephFS finally stable.
    - Only one release away from our production cluster

Synthetic workloads to mimic the code run by our users

- lots of metadata operations
- moving big files
- concurrent access/modifications of files from multiple clients

The first issues began to surface...

Testbed

We started small end of 2017:

- 3 Monitor nodes
- 3 nodes
  - SSDs for journals
  - 72 OSDs on spindles
  - Jewel release
    - CephFS finally stable.
    - Only one release away from our production cluster

Synthetic workloads to mimic the code run by our users

- lots of metadata operations
- moving big files
- concurrent access/modifications of files from multiple clients

The first issues began to surface...

Testbed

We started small end of 2017:

- 3 Monitor nodes
- 3 nodes
  - SSDs for journals
  - 72 OSDs on spindles
  - Jewel release
    - CephFS finally stable.
    - Only one release away from our production cluster

Synthetic workloads to mimic the code run by our users

- lots of metadata operations
- moving big files
- concurrent access/modifications of files from multiple clients

The first issues began to surface...

Problem 1: Where to store the metadata

- Issue: Slow metadata access was compromising the performance of the whole cluster:
- Fix: We added one NVME to each node for the metadata pool and the situation got dramatically better

Problem 2: Failing to respond to cache pressure

- Issue: ”failing to respond to cache pressure” showed up from time to time with several different workloads.
- Fix: upgrading to Luminous solved the issue.

Key decisions

- We did not want to integrate CephFS into the existing Ceph cluster.
  - New features roll out impacts only on the HPC infrastructure.
  - No need to wait for the upgrade of the main cluster.
- We wanted to use the kernel driver for performance reasons.
  - We decided to deploy Mimic $\Rightarrow$ it had quota support for it!

Production deployment

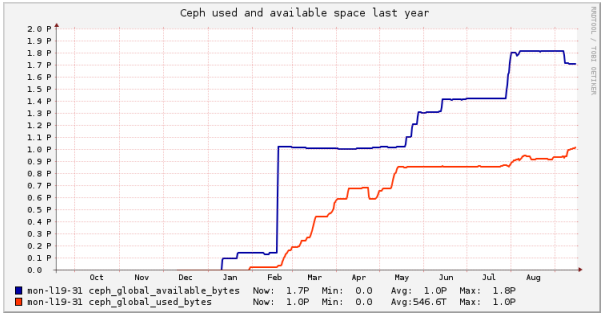
- 3 MONs (single socket 64 GB of RAM)
- 120 OSDs (10 servers, 12 x 8TB spindles, a 1,6 TB NVME, 256 GB of RAM)

From previous purchases:

- 2 MDSes (existing nodes with 64 GB of RAM $\Rightarrow$ big mistake)
- 40 solid state OSDs (5 servers, 5 x 1,6 TB enterprise SSDs, 32GB RAM)

We tried to get the new nodes single socket but the vendor came up with some excuses and gave us the second processor for free.

Deployed in stages



1. Jan 2019:
   SSDs backed FS (Replica 3) for `/home` and `/data`
2. Feb 2019:
   HDDs backed FS (EC 4+2) for `/scratch`
3. May 2019:
   added more nodes as usage increased

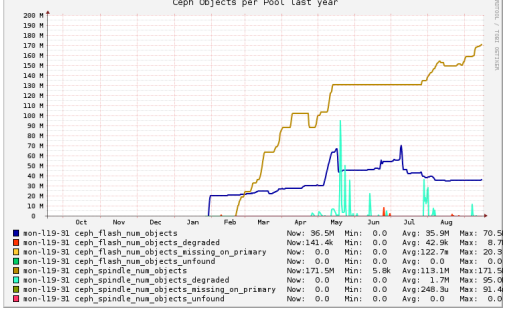
Pain point: quota

We deployed Mimic because of the quota support however:

- only from kernel 4.15 ⇒ we are still working on upgrading the infrastructure

Our solution:

- leverage `getfattr` to script out excuses from our users.
- add a wrapper to show the information at login and **delay** login with a message if over quota.



Lessons learned: Design

- provision the fastest storage you can afford for your metadata
- MDS should have enough RAM and computing power
  - they become single point of failure for the whole cluster (multi MDSes helps but does not do miracles)
- multi active MDSes work better if you can foresee the hotspots and pin directory accordingly
- using more than one filesystem is a bad idea:
  - still experimental, at least in Mimic.
  - does not provide any clear advantage (at least in our case)
- RocksDB partition size: 4% of the available space is not always needed but you should not be too cheap otherwise you risk spilling over to slower storage and degrade performances.

Lessons learned: Operations

- Ceph deals with disk failures quite gracefully but you are better off with decent drives as swapping drives is no fun.
- you should not use `du` unless you really need to: it's painfully slow and `getfattr -d -m ceph.dir.rbytes` works much better.
- the flexibility of controlling filesystem related parameters like quotas and directory pinning via extended attributes is amazingly convenient.
- you should be extra careful about your `ceph.conf` as some directives can have confusing names: mds cache size and mds cache memory limit comes to mind.
- kernel clients are not properly identified by the `ceph features` command:
  - to use pg-upmap on your cluster (that needs at least Luminous clients) it is enough to have a kernel > 4.13 despite what `ceph features` reports.

Current status

- 6 nodes
  - 384 cores
  - 18 TB of RAM
- 5 nodes
  - 40 nVidia K80
- 2 nodes (4 by the end of this year)
  - 48 cores
  - 16 nVidia V100 GPUs
  - Infiniband
- 16 nodes
  - 384 cores
  - 6 TB of RAM
  - Infiniband
- 30 cloud workers

- CephFS to serve both `/home` and `/scratch`

Future plans

- Leverage the multiple data pools support in CephFS to:
  - Make it cheaper: more aggressive EC profile 6+2 or 8+2 with compression for warm data, no NVME (we would rely on the already existing metadata pool).
  - Make it simpler: deploy the all-flash tier as an additional datapool to get rid of the second filesystem.
- Make it more tolerant to users' mistakes: periodic user accessible snapshots to retrieve data previously deleted/modified.