ETH zürich

Scientific IT Services

CephFS: from synthetic benchmarks to real-world user workloads

Mattia Belluco

1.2.2025

SIS: Scientific IT Services

- A section of ETH Zürich IT Services
- Composed of about 40 experts in various areas of scientific computing
- Background in different areas of science

Outline

1. Context
2. Following best practices
3. Benchmarking
4. Going in production
5. What we would do differently

Outline

1. Context
2. Following best practices
3. Benchmarking
4. Going in production
5. What we would do differently

LeonhardMed Trusted Research Environment

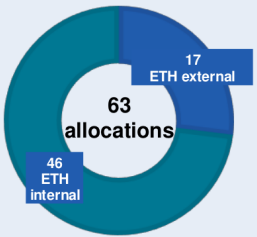
> 507 researchers

> 3.2 PB secure data storage

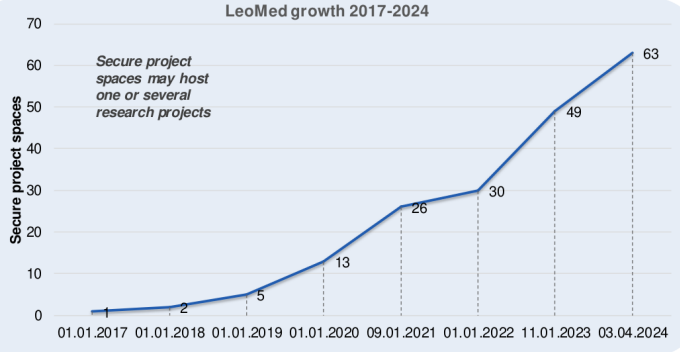
> 10 Data Providers (DPs)
Swiss (e.g. USZ, EOC) and international (UCSF)

> 1500 Data Transfer Requests from DPs in active reseach projects

Leomed Customers Affiliation (Eth-internal Or External)



LeoMed growth 2017-2024



The starting point

Our requirements:

- A usable capacity of ~2 PB
- POSIX compliant
- Reliable
- Scalable
- Affordable
- Reasonably performant
- Accessed by virtualized workers (VMs) of various size and capabilities

Reasonably performant?

- Predictable performance.
- Smooth metadata operations.
- Circumscribe as much as possible poor performance caused by bad behaviors to specific projects.

Outline

1. Context
2. Following best practices
3. Benchmarking
4. Going in production
5. What we would do differently

Blueprint

We settled on an initial configuration of:

- 16 dual socket "osd" servers distributed in 4 distinct racks, each equipped with:
  - 24 HDD of 18 TB each
  - 4 NVME of 2 TB each
  - 1 LACP bond composed of 2 links at 25 Gbit
- 3 "mon" servers (hosting MON and MGR services)
- 3 "mds" servers (hosting the MDS services) each hosting 4 MDS daemons

Deployment guidelines

- When using spindles, provision WAL+DB on fast devices
- Provision 4 OSDs per NVME device
- Metadata pool on fast devices
- Higher clocked CPU model on the servers hosting MDSes.

Outline

1. Context
2. Following best practices
3. Benchmarking
4. Going in production
5. What we would do differently

Benchmarking requires time

Benchmarking data are:

- gathered and stored but never properly analyzed
- gathered but only used for coarse grain estimations
- not gathered at all

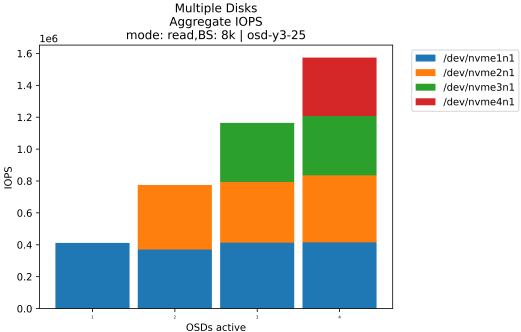
With the wisdom gathered in past deployment experiences, we decided to bring in some external help in the person of Matt Anson from StackHPC.

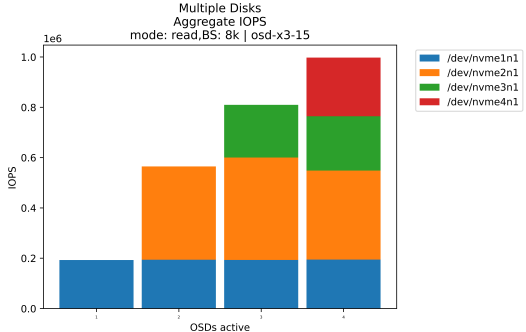
Our plan

- Characterize the hardware
- Use 16 virtualized worker nodes as clients
  - each virtual node uses an entire hypervisor, which mimicks the final layout while dispensing from the effects of having colocated virtual machines.
- Compare replica with EC 2+2 and EC 4+2 schemes

Hardware issues

On one node a NVME device seems to be outperforming the other:





Looking closer at the Y axis it turns out **the exact opposite is true!**

Initial data

EC 4+2 is out

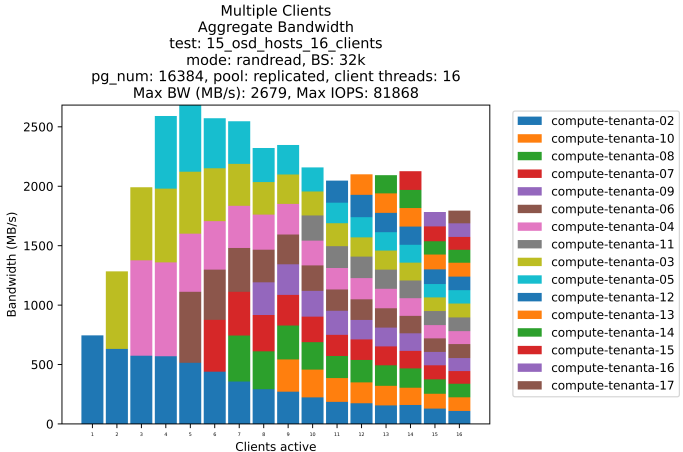
An initial rough evaluation of Replica vs EC 2+2 and EC 4+2 excludes the latter because of the sizable performance gap.

First data comes in:

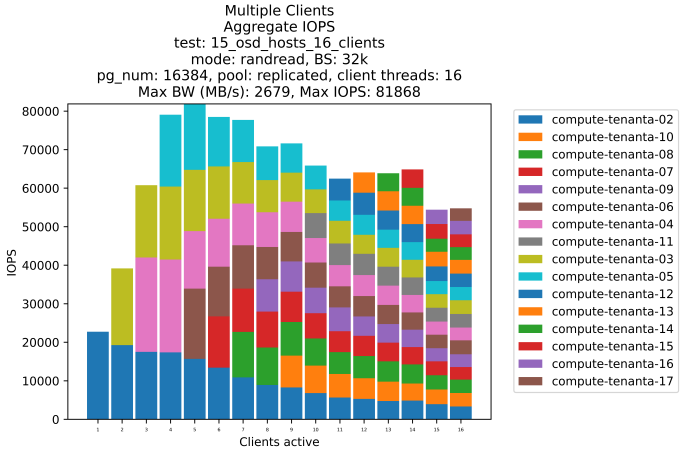
client_type	client_n	test	osd_hosts	io_size	io_type	pool_pg_num	client_threads	pool_name	max_bw_client_n	max_bw	max_iops
baremetal	8	erasure.pg_num8192.15osd_hosts-8_clients	15osd_hosts	32k	randread	8192	16	erasure	1	1155	35249
baremetal	8	erasure.pg_num8192.15osd_hosts-8_clients	15osd_hosts	32k	read	8192	16	erasure	8	9822	299901
baremetal	8	erasure.pg_num8192.15osd_hosts-8_clients	15osd_hosts	4M	randread	8192	16	erasure	7	27674	6596
baremetal	8	erasure.pg_num8192.15osd_hosts-8_clients	15osd_hosts	4M	read	8192	16	erasure	7	29206	6960
baremetal	8	replicated.pg_num16384.15osd_hosts-8_clients	15osd_hosts	32k	randread	16384	16	replicated	3	2167	66186
baremetal	8	replicated.pg_num16384.15osd_hosts-8_clients	15osd_hosts	32k	read	16384	16	replicated	7	13107	400082
baremetal	8	replicated.pg_num16384.15osd_hosts-8_clients	15osd_hosts	4M	randread	16384	16	replicated	4	34683	8268
baremetal	8	replicated.pg_num16384.15osd_hosts-8_clients	15osd_hosts	4M	read	16384	16	replicated	7	43702	10416

Usually at this point the benchmarking effort would be overrun by other priorities but not this time!

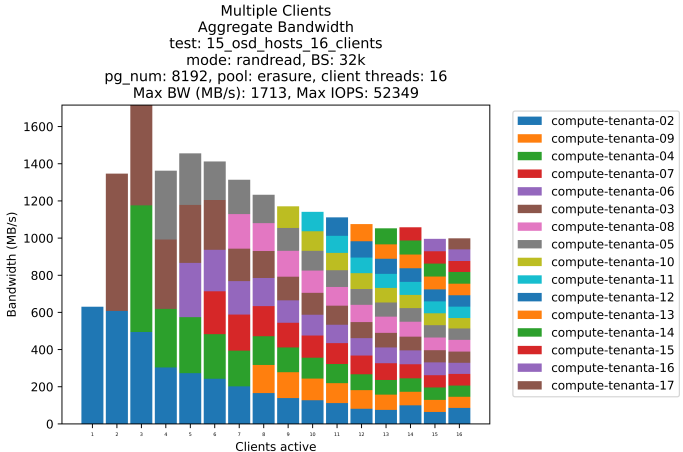
Replica
Bandwidth



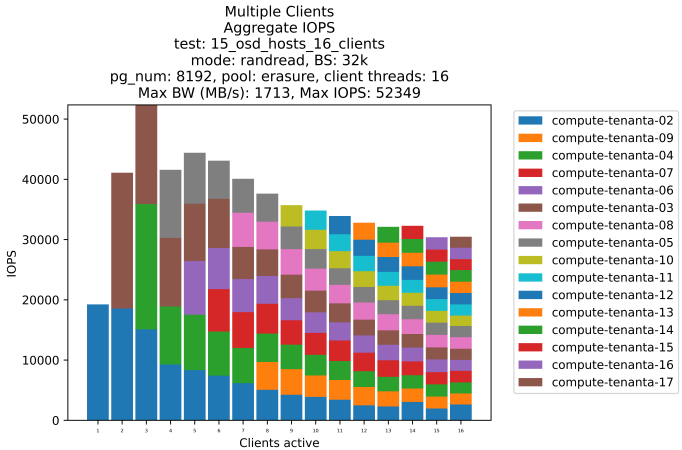
Replica
IOPS



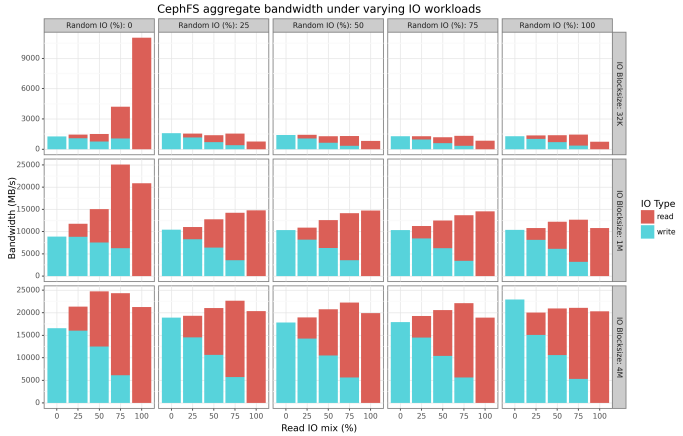
Erasure Coding 2+2
Bandwidth



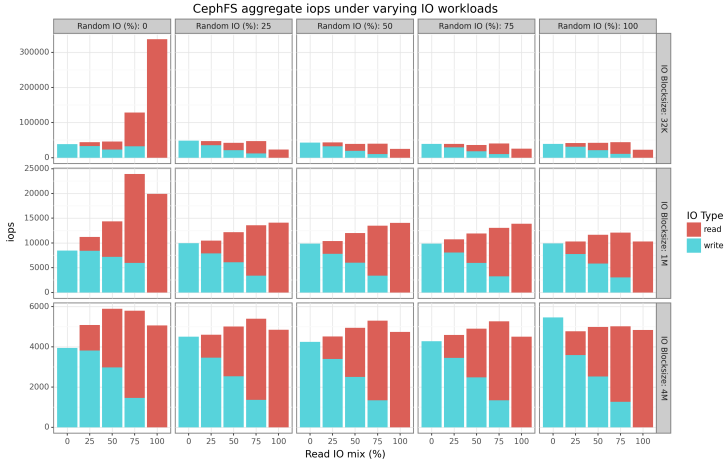
Erasure Coding 2+2
IOPS



Erasure Coding 2+2
Mixed workload - Bandwidth



Erasure Coding 2+2
Mixed workload - IOPS



Ceph Pool: erasure | Pool pg_num: 8192 | Clients: 16 | Client Threads: 16

Outline

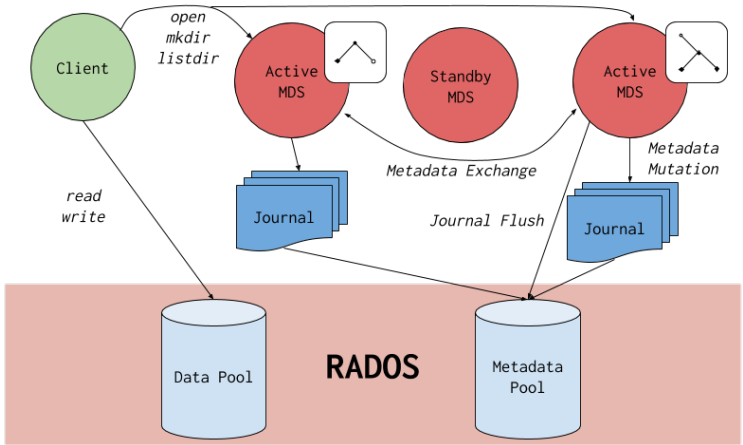
1. Context
2. Following best practices
3. Benchmarking
4. Going in production
5. What we would do differently

Production
It starts with a bang

- System benchmarked with the default Pacific releases settings and 1 active MDS
- More than 1 PB of data migrated from the previous storage system
- Gradual ramp up by migrating individual projects to the new system
- It became soon clear that a single MDS daemon could not cope with the amount of requests originating from the clients.

CephFS architecture

MDS are the key to coordinate clients data access.



Corrective measures

- Incrementally increase `max_mds` to help the metadata service to cope with the requests, up to the current value of `7`
- After the first increase to 3 we noticed a sudden increase of the `Req/s` counters that we later realized was due to the mds balancer exporting subtrees.
- We pinned the most active project directories to dedicated MDS demons (later we pinned them all).

Cache configuration

Doc page: https://docs.ceph.com/en/pacific/cephfs/cache-configuration/

- Lowered `mds_max_caps_per_client` to 524288 from 1000000
- Incrementally increase `mds_cache_memory_limit` up to the current value of `110 GB`
- Increase the mds `mds_cache_reservation` from `5%` to `10%` (i.e. `0.10`)
- Lowered the `mds_cache_threshold` value from `1.5` to `1.1`
- Lowered `mds_cache_trim_decay_rate` to 0.8 from 1.0
- Increased `mds_cache_trim_threshold` to 393216 from 256K for Quincy and 64K for Pacific

Throttle deletion operation

Issue: deletion of millions of files caused the MDS to report being behind on trimming, eventually causing slow metadata ops.

Fix:

- Decrease `mds_max_purge_files` to `64`
- Decrease `mds_max_purge_ops_per_pg` to `0.500000`

Deletes would take longer but without ill effects on the MDS demon.

Tune MDS Recall

- `mds_recall_max_caps` to 20000 (from 5k in Pacific and 30k in Quincy)
- `mds_recall_max_decay_rate` to 2.000000 (from 2.5 Pacific and 1.5 in Quincy)
- `mds_recall_max_decay_threshold` to 65536 (from 16K in Pacific and 128K in Quincy)
- `mds_recall_global_max_decay_threshold` to 262144 (from 64K in Pacific and 128K in Quincy)
- `mds_recall_warning_threshold` to 131072 (from 32K in Pacific and 256K in Quincy)

MDS failover

When an MDS demon fails in a busy system like ours, the newly MDS assigned to the failed rank needs more time then the default value of 60 seconds to replay the journal and join the cluster.
After several attempts `mds_beacon_grace` is set to `300` seconds

Outline

1. Context
2. Following best practices
3. Benchmarking
4. Going in production
5. What we would do differently

CephFS data pool on a replicated flash pool

Replication required since Nautilus because of the tiny objects that are created on the root of the filesystem with backtrace information (used in Disaster Recovery) and hardlinks references.

Problem:

- Currently around 560 M inodes in our 480 OSDs filesystem
- 3.5M tiny objects per disk that needs to be rebalanced if a drive breaks.

In our system it takes **as much time** to rebalance the default "empty" cephfs datapool as it tkes to rebalance the actual data.

Carefully evaluate the selection of the EC algorithm

Presentations at Cephalocon 2024 by Jamie Pryde from IBM shows how performance is highly dependent on the chosen Erasure Coding algorithm and debunk the belief that ISA-L only works only on Intel CPUs.

Reference:
Erasure Coding: 5 Ways to Split a Squid
https://www.youtube.com/watch?v=aM8sJgDD-x4

Questions?

ETH zürich

Mattia Belluco
Cloud Architect
mattia.belluco@id.ethz.ch

ETH Zurich
IT Services
OCT G 35
Binzmühlestrasse 130
8092 Zürich, Switzerland
https://sis.id.ethz.ch