Ceph and XRootD Deployment and Monitoring at Lancaster

GridPP48, 31 August 2022

Gerard Hand, Steven Simpson, Matt Doidge
(and some tweaks from Sam)

The Story So Far

Lancaster decided to move to a CephFS + Xrootd for well documented reasons (see our talk at GridPP47)...

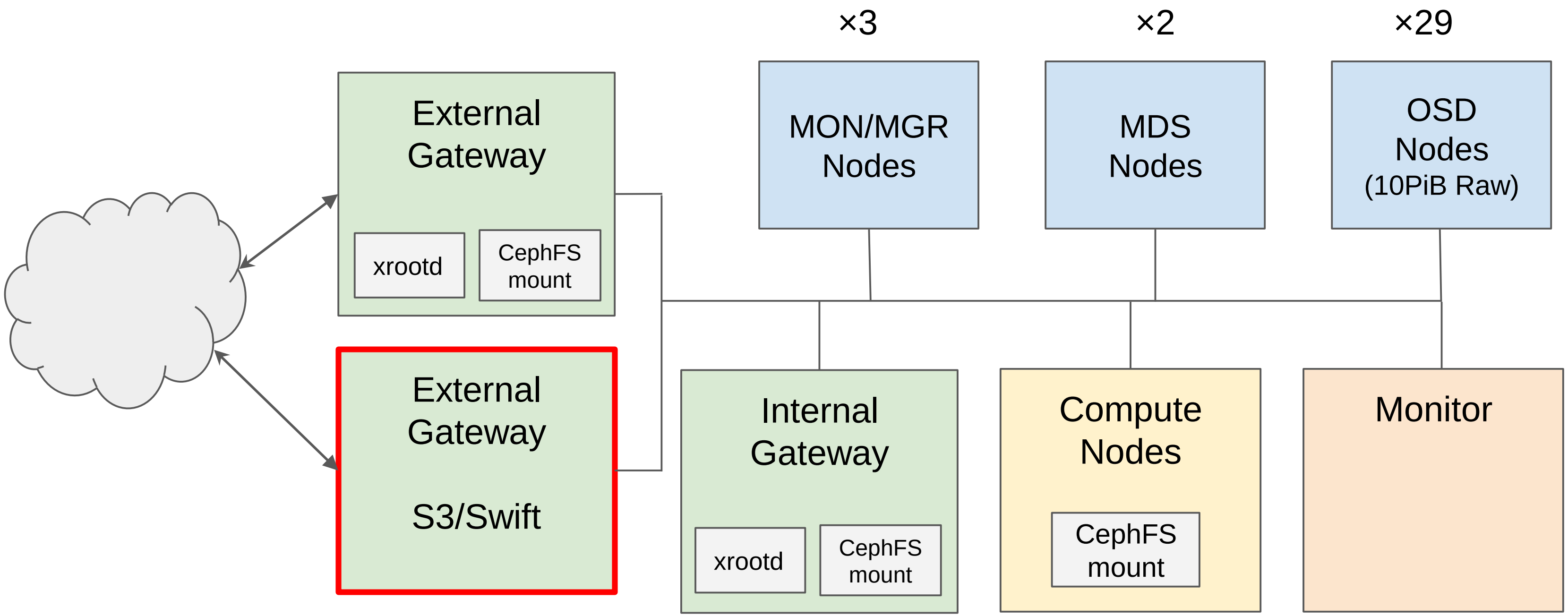
…and it’s gone quite well.

- No disasters (touch wood).
- Moved into production without a lot of people noticing.
- The placeholder setup (of a single standalone Xroot server fronting the whole thing) held up well.
  - And we’ve snuck a redirector in again without anyone really noticing.

But it’s not been a journey not without a few gotchas and lessons learnt.

Ceph architecture

- An S3/Swift gateway has been added.

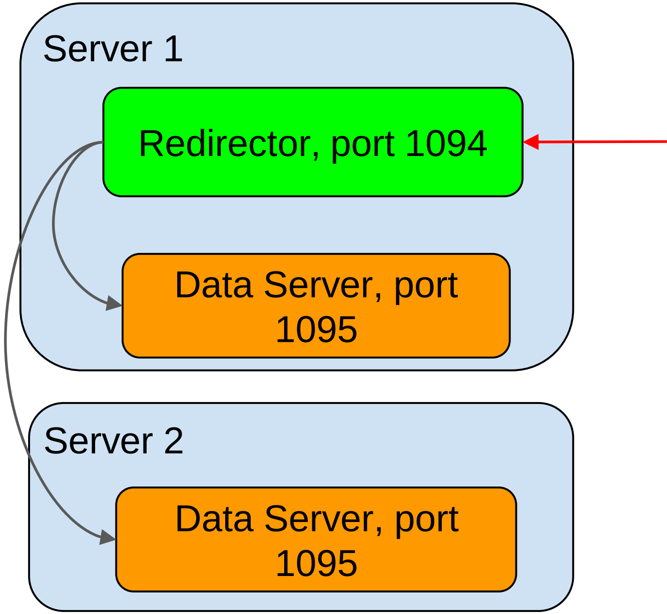


Lessons learned/Gotcha’s Experienced

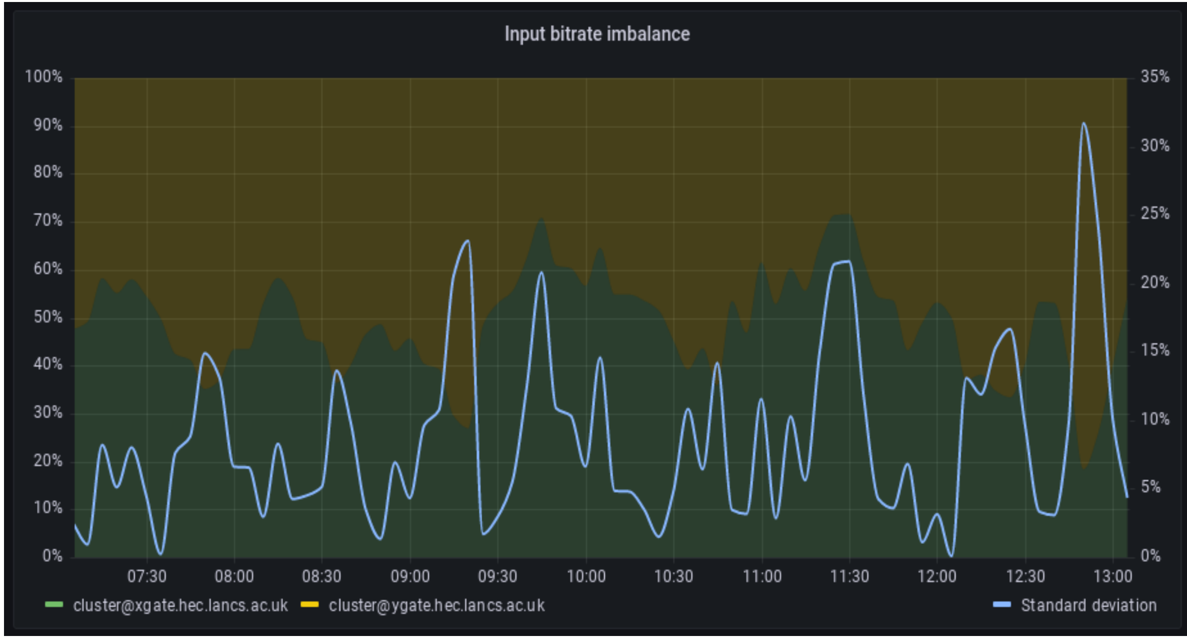
- Using Pacific instead of Octopus has proved a good choice as updates/patches are not getting back ported to Octopus.
- Creating an S3/Swift gateway was straightforward.
- Cephadm + Ceph Orchestrator have worked well. We have rebooted all the nodes in the cluster after system updates with no downtime.
- Recent problems with scrubbing not being performed in time required changes to default configuration settings.
- Running a Pacific cluster and using Octopus to mount CephFS on the client machines has worked without problems (No support for Pacific+ on Centos 7). There are currently 985 CephFS clients connected to the cluster.
    - All the odd jobs NFS used to do, like shared directories, CephFS now does.
- We have noticed that intermittently the Active MGR drops out of the cluster when running the Ceph Dashboard.

XRootD Xperiences

- Moved to a two-node redirector setup (see dodgy pic right)
  - Would rather have a 3 node setup (dedicated redirector + 2 dedicated servers), but this was easier to slot into place.
  - Discovery: Redirection very sensitive to IPv6 problems.
  - No problems noticed (so far).
  - A standalone server actually did quite well, the reasonably specced 25Gb-connected box coped with about 60% of our load.
- It was a race between rolling out the redirector or having the rucio "Symlink" plugin in place.
  - This won, as the rucio symlink hit a problem when it was attempted to be rolled out and had to be rolled back.
- Found that xrootd logging levels seem to be almost binary: "Too Much" or "Nothing Useful".
  - Almost ran out of disk space when we left the logging on "all -debug".
- Overlooked testing xrootd TPC…
  - We thought no one really used this, we were wrong.
  - Turned out you need an "-f" in the default xrdcp executed for TPCs.
- We were hit again by the rhel8 "assertion error" problem during our testing.
  - This should be fixed in the upcoming xroot 5.5.0
  - *But* Steven provided fixed code several minor versions ago (and some fixes were in some *earlier* 5.4.x releases…)



Redirector Balance distribution (default settings)



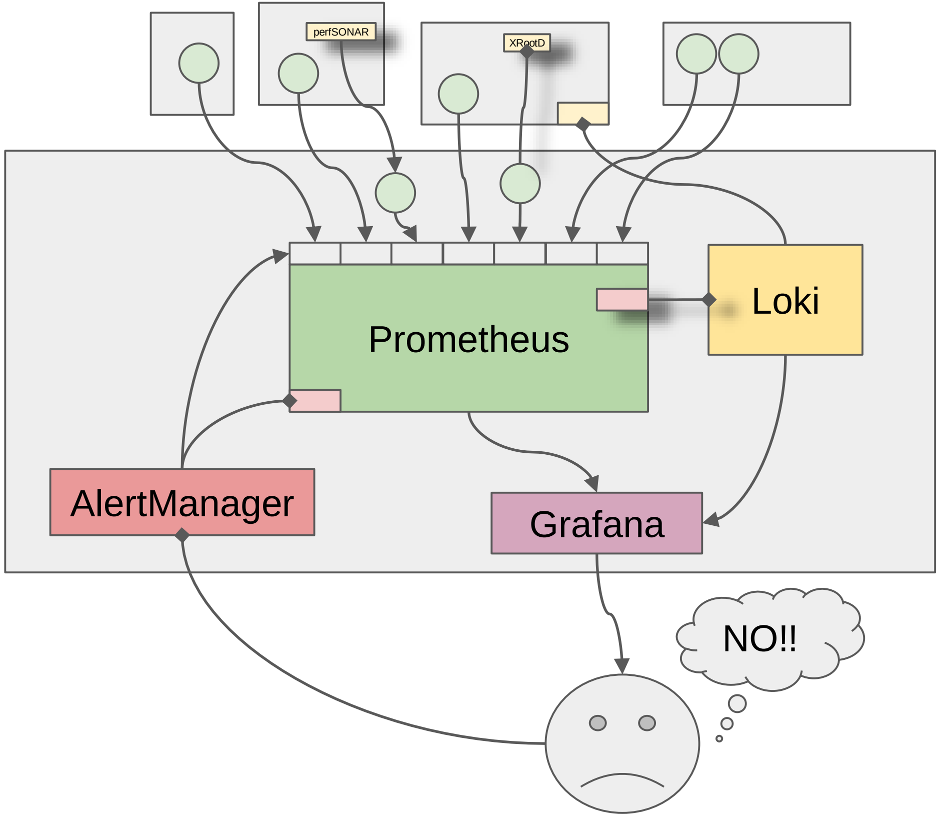
- I/O distribution
  - Short period oscillation (30min)
  - Flattens out over-all
- Request distribution
  - Even, even at small granularity
- Load distribution, based on getrusage
  - Even, even at small granularity
  - Slightly less load on shared host
    - (Would have expected the other way around.)

XRootD NeXt Steps

- How many 25Gb-connected xrootd boxes does it take to fill our 40Gb bandwidth? <insert lightbulb joke that doesn’t get us sent to HR>
    - I think the answer will be higher than 40/25
    - Whatever the number is, this is the minimum number of redirectors we will need.
- Tokens!
    - Scitokens have been enabled since day 1, but we’re not in the testbed (yet).
    - Enabling ESCAPE tokens for (e.g.) SKA looks like it should be “simple”.
    - The main blocker with rolling out tokens is testing - we’re well versed in grabbing a voms proxy and firing off gfal-*/xrd* commands. Not so with oidc-*/eldritch curl invocations.
- Using some old DPM nodes for an xrootd testbed - useful for trying out Rocky8/xroot 5.5.X/anything weird.
- Ultimate aim is an HA redirector in front of (enough + 1) xroot servers, with none pulling double duty as a redirector and data server.
- Our configs can be seen at: https://github.com/mdoidge/lancsxroot

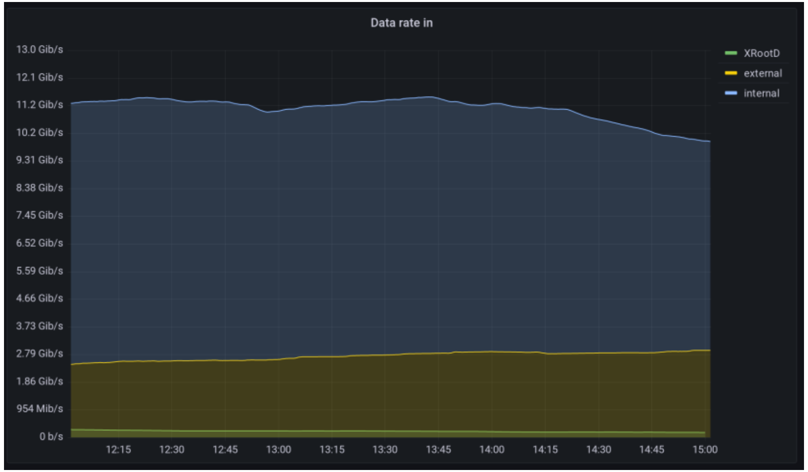
Monitoring updates

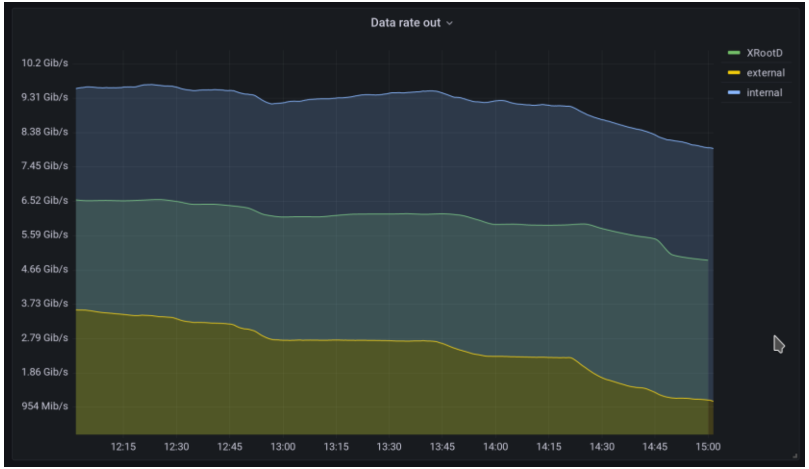
- Use Loki in anger
  - Easy to knock over
    - Too much logging from XRootD during debug
    - Slow queries in Grafana; many fail
    - Mitigated by recordings pushed into Prometheus
- Overhaul of metric labelling
  - Distinguish between host and interface
  - Eases correlation of metrics from diverse sources
- Custom exporters
  - XRootD
    - xrd.report push XML over UDP
    - pulled as time-stamped OpenMetrics
  - perfSONAR
    - periodic pull JSON from esmond
    - pulled as time-stamped OpenMetrics



Balancing the metric sources

An ongoing task is figuring out the cause for differences between the xroot and system metrics, such as in the data in/out plots below*. The differences seem too large to just accounted for by CEPHFS traffic - unless there’s unexpected activity.





*external/internal rates are stacked.

Ceph and XRootD Deployment at Glasgow

GridPP48, 31 August 2022

Sam Skipsey

Summary

- Xrootd upgrades + patches from RAL
- Significant improvements from
  - CephSUM "external checksum" python script
  - Buffered IO in XrdCeph
  - Namelibs in XrdCeph (so, one less reason to have a proxy xrootd)
- Weird stuff
  - Xrootd 5.4.3
    - Pgreads everywhere
    - Apparently you're only supposed to turn xrootd.async on if you're a proxy
    - (Even the DPM xrootd config files make this "mistake", so it's interesting to ask how the "wrong approach" became so commonplace…)
- Xrootd 5.5.0rc2

Ceph and XRootD Deployment at Brunel

GridPP48, 31 August 2022

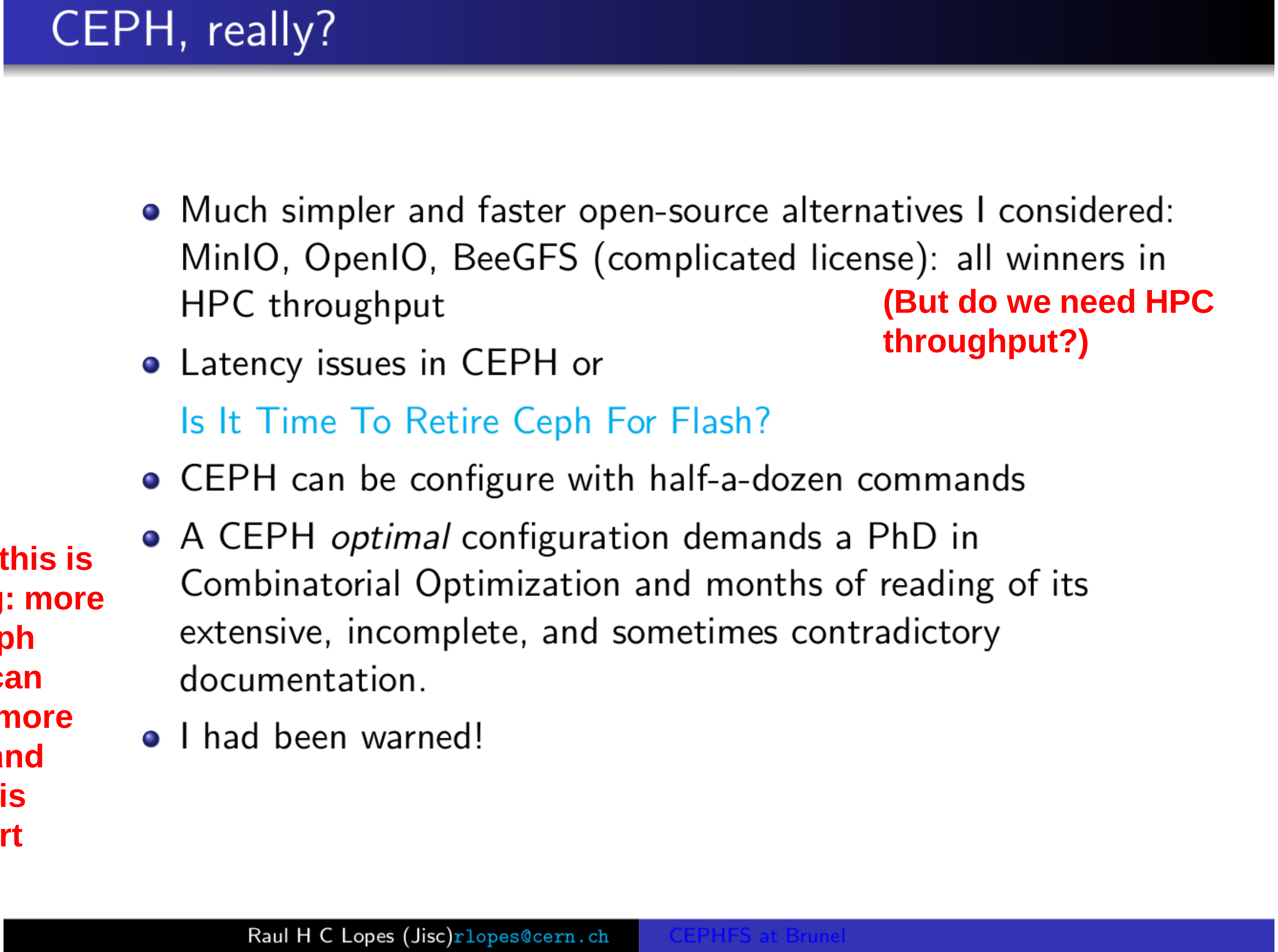
Raul Lopes

(with some notes in red from Sam)

The Briefing

(DPM)

- DOME is doomed
    - End of support for new development coming
    - Good enough in 2018
    - 2022: still too brittle in of face server failure
    - Another piece of HEP-only software artifact
- Why CEPH?
    - Open source with development/support at Red Hat, Canonical, SUSE (some quite good docs), Hetzner, others...
    - distributed storage with resilience to server failure (or disk or rack or data centre...)
    - Scalability
    - Storage solution for WLCG work, Brunel HPC groups, Brunel private cloud
    - Community growing in GridPP: RAL T1, Glasgow, Lancaster.
    - And... sorry to repeat: it's not a HEP-only artifact.



CEPH, really?

- Much simpler and faster open-source alternatives I considered: MinIO, OpenIO, BeeGFS (complicated license): all winners in HPC throughput

(But do we need HPC throughput?)

- Latency issues in CEPH or

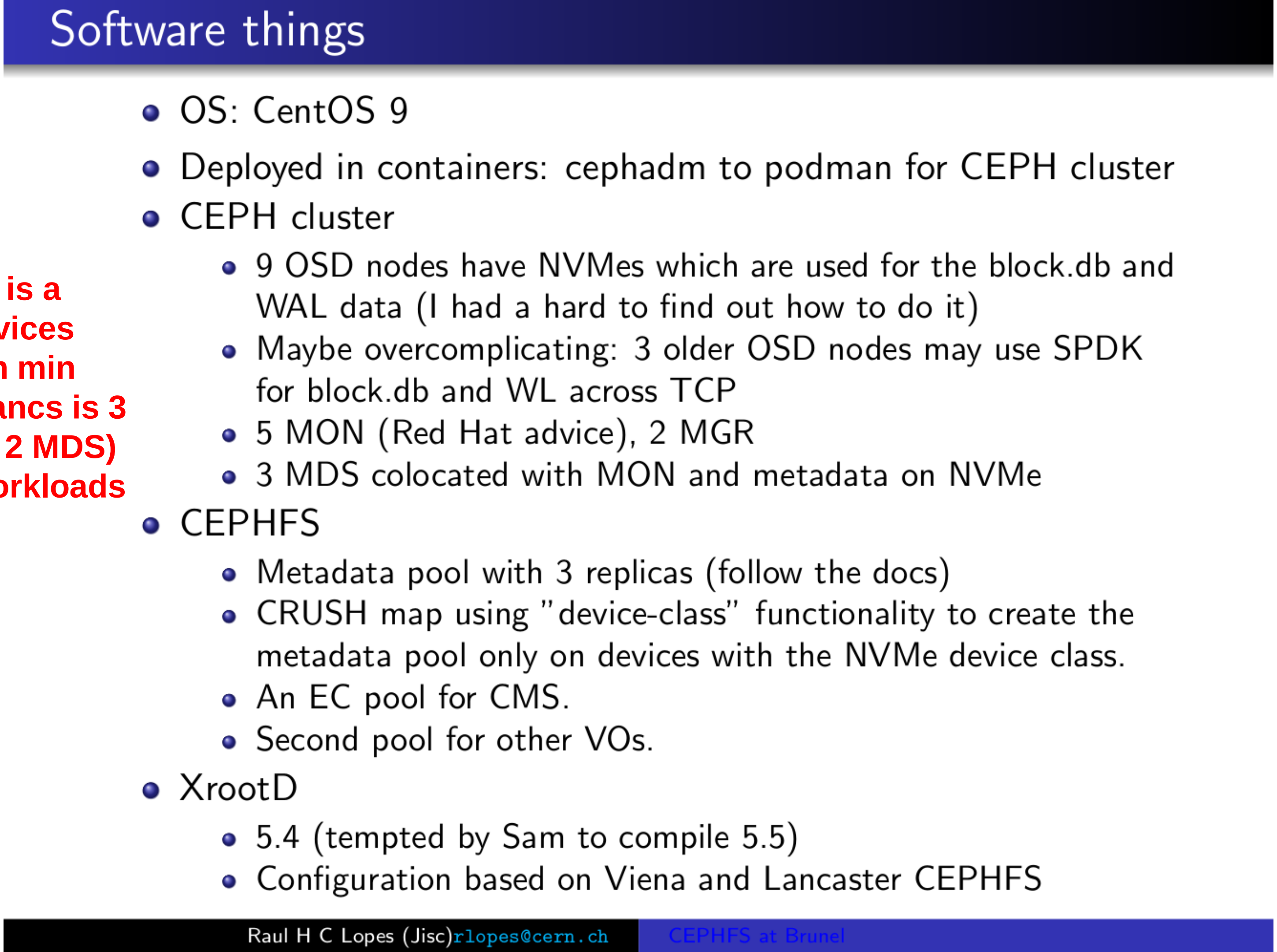
  Is It Time To Retire Ceph For Flash?

- CEPH can be configure with half-a-dozen commands
- A CEPH *optimal* configuration demands a PhD in Combinatorial Optimization and months of reading of its extensive, incomplete, and sometimes contradictory documentation.
- I had been warned!

Although this is improving: more recent Ceph releases can autotune more things… and cephadm is fairly smart

Hardware

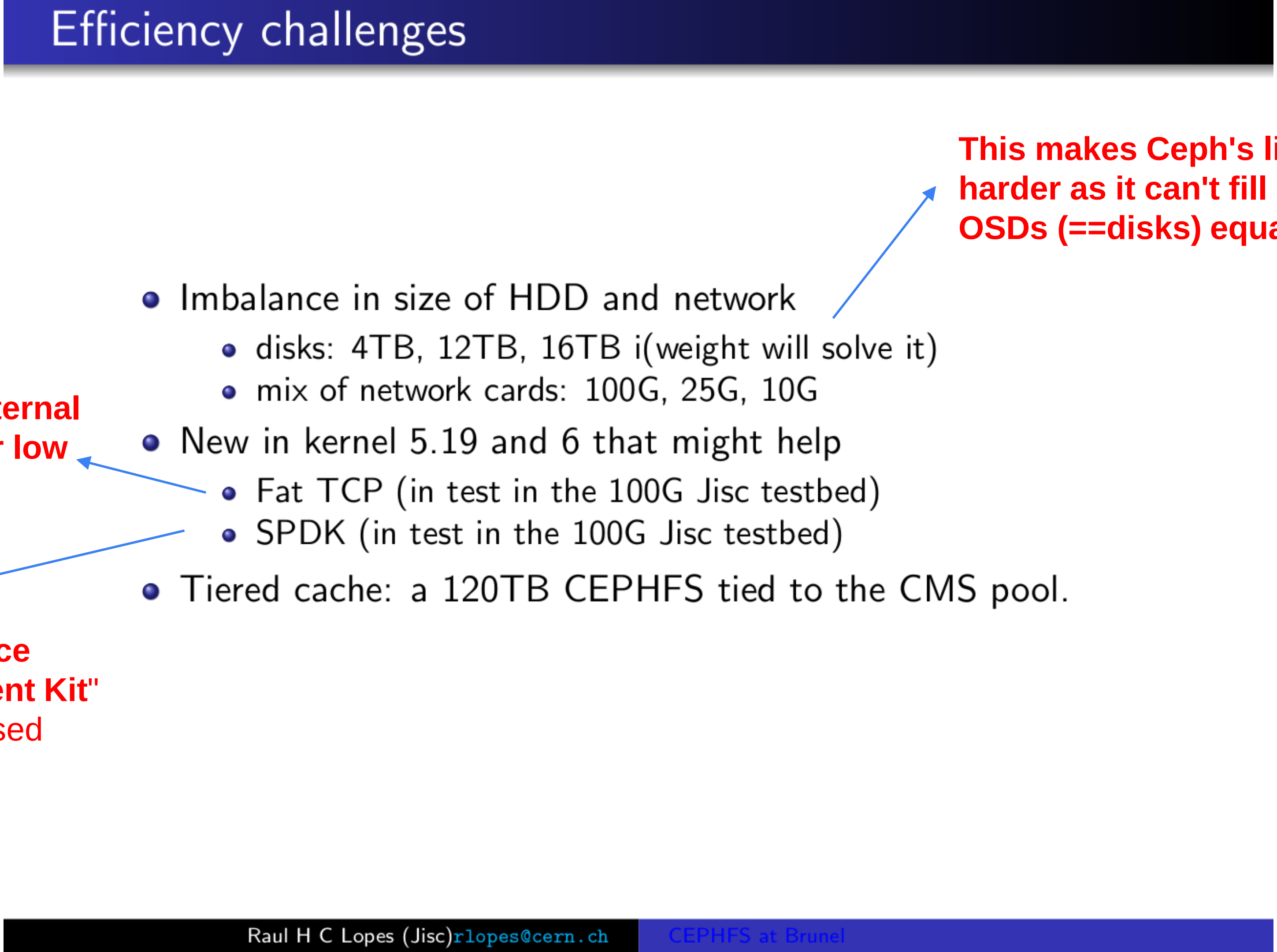
- six new storage servers
  - dual 100G NICs
  - 24 x 16TB HDD
  - 2xSSD (for OS)
  - 2x8TB NVMe
  - 2x16 cores CPU
  - 192 GB RAM
- six older servers (perfect imbalance)
  - disks from 4TB to 12 TB
  - NICS: dual 10G or 25 G
  - 192 GB RAM
  - 12 cores or 24 cores
- two XrootD gateways
  - Internal: dual 100G
  - External: dual 25G
- Cache service: 120 TB on NVMe



Software things

- OS: CentOS 9
- Deployed in containers: cephadm to podman for CEPH cluster
- CEPH cluster
  - 9 OSD nodes have NVMes which are used for the block.db and WAL data (I had a hard to find out how to do it)
  - Maybe overcomplicating: 3 older OSD nodes may use SPDK for block.db and WL across TCP
  - 5 MON (Red Hat advice), 2 MGR
  - 3 MDS colocated with MON and metadata on NVMe
- CEPHFS
  - Metadata pool with 3 replicas (follow the docs)
  - CRUSH map using ”device-class” functionality to create the metadata pool only on devices with the NVMe device class.
  - An EC pool for CMS.
  - Second pool for other VOs.
- XrootD
  - 5.4 (tempted by Sam to compile 5.5)
  - Configuration based on Viena and Lancaster CEPHFS

I think this is a bigger services layout than min needed (Lancs is 3 MONs and 2 MDS) for Grid workloads



Efficiency challenges

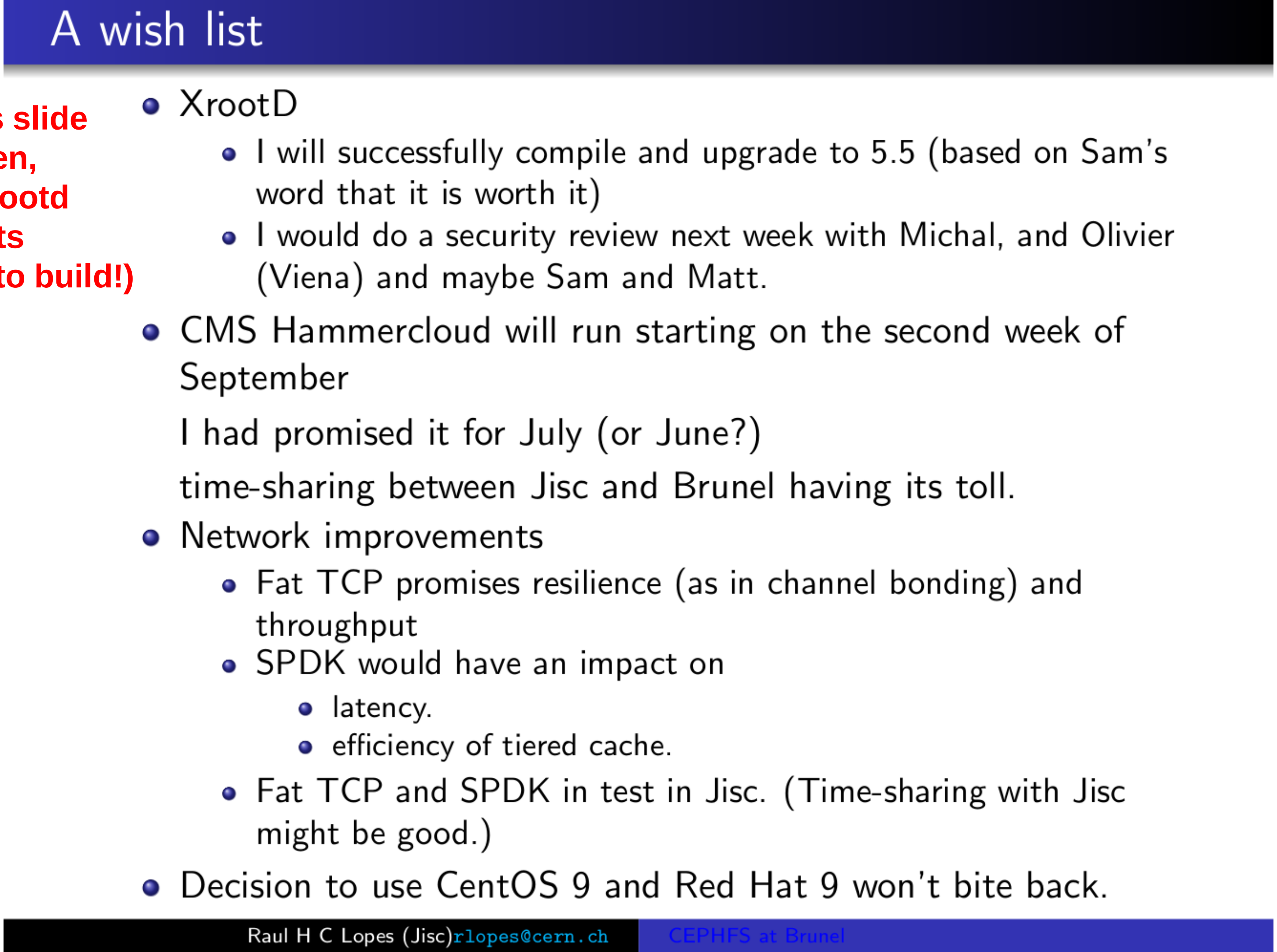
- Imbalance in size of HDD and network
  - disks: 4TB, 12TB, 16TB i(weight will solve it)
  - mix of network cards: 100G, 25G, 10G
- New in kernel 5.19 and 6 that might help
  - Fat TCP (in test in the 100G Jisc testbed)
  - SPDK (in test in the 100G Jisc testbed)
- Tiered cache: a 120TB CEPHFS tied to the CMS pool.

This makes Ceph's life harder as it can't fill OSDs (==disks) equally

"BIG TCP"
- Bigger internal packets for low latency

"Storage Performance Development Kit"
nVME focused
lockless io

Since this slide was written, official Xrootd 5.5.0 exists (no need to build!)



A wish list

- XrootD
  - I will successfully compile and upgrade to 5.5 (based on Sam's word that it is worth it)
  - I would do a security review next week with Michal, and Olivier (Viena) and maybe Sam and Matt.
- CMS Hammercloud will run starting on the second week of September

  I had promised it for July (or June?)

  time-sharing between Jisc and Brunel having its toll.
- Network improvements
  - Fat TCP promises resilience (as in channel bonding) and throughput
  - SPDK would have an impact on
    - latency.
    - efficiency of tiered cache.
  - Fat TCP and SPDK in test in Jisc. (Time-sharing with Jisc might be good.)
- Decision to use CentOS 9 and Red Hat 9 won't bite back.