QoS@JSI

Andrej Filipcic

JSI & other SI Infrastructure used by ATLAS, Belle 2 ...

- SiGNET Tier-2
  - 7k cores
  - 4.2 PB in NDGF-T1, Infortrend, Lenovo Raid6 Boxes
  - 750 TB ceph: CephFS HDD cache, rbd, permanent user storage
- JSI-NSC - general purpose JSI cluster, partially HPC
  - 2k cores, 30TB cache
  - Ceph under deployment these days
- ARNES - general purpose SI cluster, partially HPC
  - 4.5k cores
  - 300 TB CephFS

JSI & other SI Infrastructure used by ATLAS, Belle 2 ...

- HPC-RIVR-UM: general purpose (prototype) , since 11. 2019
  - 5k core Epyc1, 150TB SSD CephFS, 100Gb/s eth + infiniband
  - 24 GPU cards
- Vega (HPC-RIVR-IZUM): peta scale EuroHPC, coming end of 2020. Very rough preliminary specs:
  - ~200k core Epyc2(3?) or Intel Cascade/Cooper Lake
  - ~500 GPU Cards
  - ~30PB HDD Ceph, ~4PB NVMe or NVMeOF Ceph/SpectrumScale/Lustre
  - 100 Gb/s HDR infiniband, 500Gb/s WAN, GEANT/LHCONE, IPoX and external connectivity

Most of the cluster in Slovenia starting to use Ceph + CephFS

Storage benchmarks

- IO500 comparision on HPC-RIVR-UM
  - SAS LSI SAS3008 SSD - 4GB/s throughput

beegfs: rhel7 np=168, nodes=42
[RESULT] BW phase 1 ior_easy_write
7.406 GB/s : time 535.61 seconds
[RESULT] BW phase 3 ior_easy_read
10.395 GB/s : time 381.60 seconds
[RESULT] IOPS phase 4 mdtest_easy_stat
71.995 kiops : time 90.84 seconds
[[SCORE] Bandwidth 2.68325 GB/s : IOPS 24.1339 kiops : TOTAL 8.04719

ceph size=1: fc30 np=184, nodes=46
[RESULT] BW phase 1 ior_easy_write
6.305 GB/s : time 1009.66 seconds
[[RESULT] BW phase 3 ior_easy_read
10.867 GB/s : time 585.75 seconds
[RESULT] IOPS phase 4 mdtest_easy_stat
15.791 kiops : time 178.40 seconds
[SCORE] Bandwidth 4.30218 GB/s : IOPS 9.06217 kiops : TOTAL 6.24396

gpfs: ec 2+1 rhel7 np=96, nodes=4
[RESULT] BW phase 1 ior_easy_write
2.234 GB/s : time 422.24 seconds
[[RESULT] BW phase 3 ior_easy_read
7.557 GB/s : time 124.83 seconds
[RESULT] IOPS phase 4 mdtest_easy_stat
57.845 kiops : time 109.49 seconds
[[SCORE] Bandwidth 1.44139 GB/s : IOPS 9.52048 kiops : TOTAL 3.70442

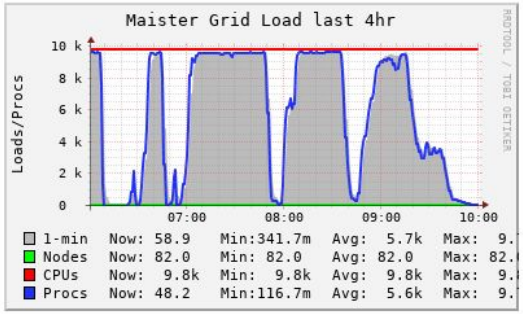
ceph size=2: fc30 np=184, nodes=46
[RESULT] BW phase 1 ior_easy_write
2.557 GB/s : time 2441.12 seconds
[RESULT] BW phase 3 ior_easy_read
10.971 GB/s : time 568.87 seconds
[RESULT] IOPS phase 4 mdtest_easy_stat
15.903 kiops : time 183.74 seconds
[[SCORE] Bandwidth 2.94947 GB/s : IOPS 9.06193 kiops : TOTAL 5.1699

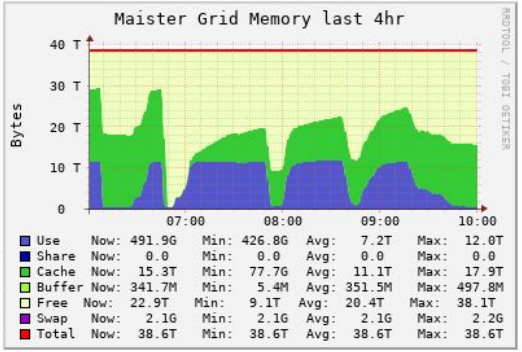
- 3 servers
  - 100 Gb/s ethernet
  - 24 2TB SSDs

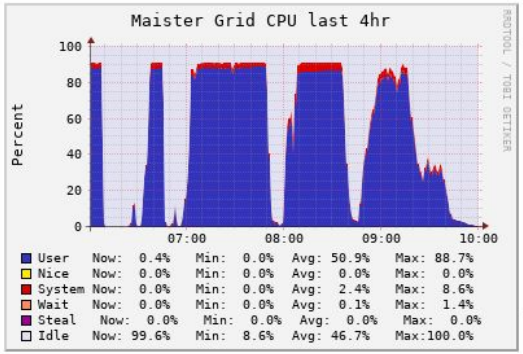
- CephFS quite comparable to others, slower on metadata
- Throughput limited by SAS

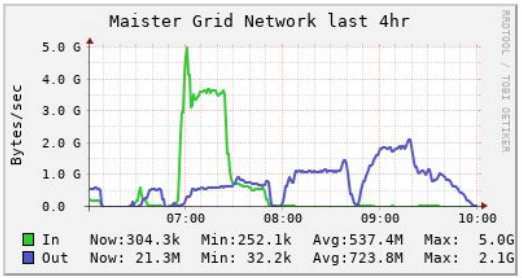
HPC Test run 32-core ATLAS digi+reco job

- Start 6:50 - End 10:00
- All jobs different inputs, cloned from the same 30GB input file of a single job
- CephFS:
  - Up to 26k read iops
  - Up to 4k write iops.
- 9.4TB workdir size
  - local storage not used

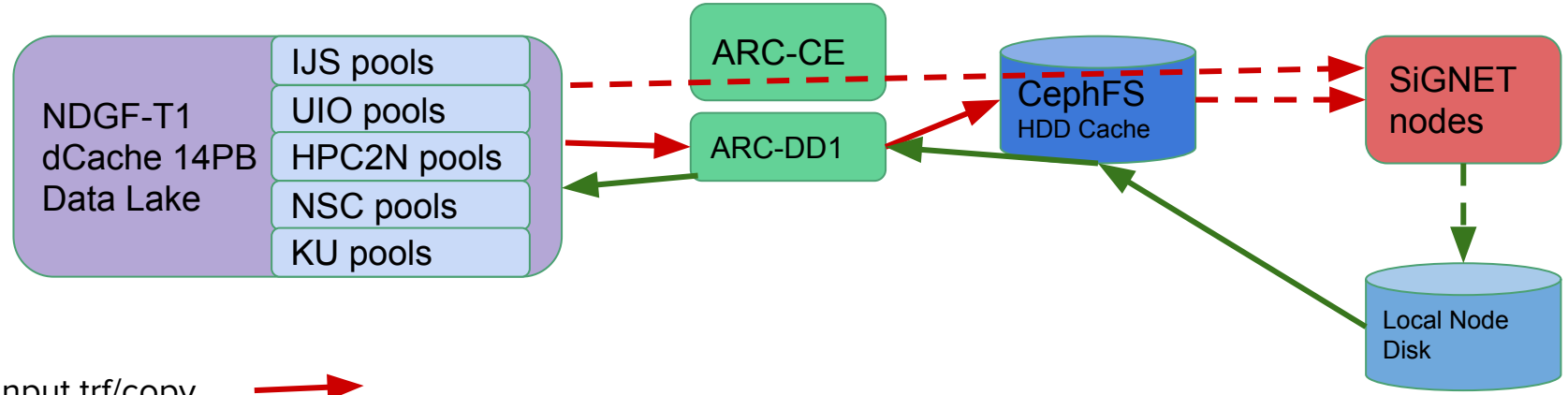








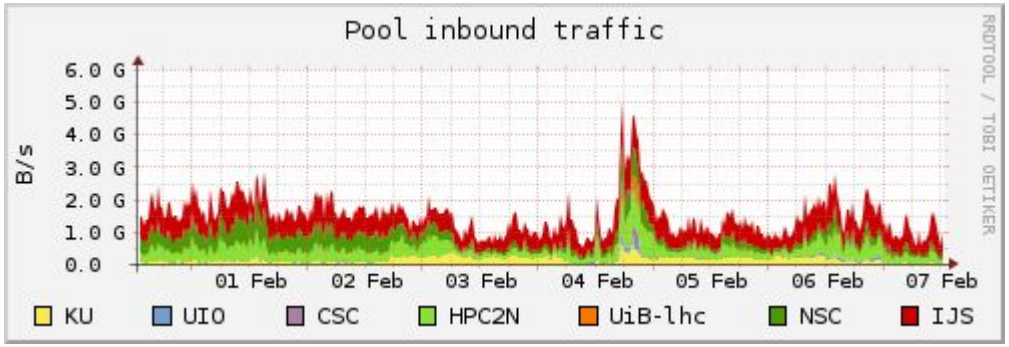
Current ATLAS job data flow (push)

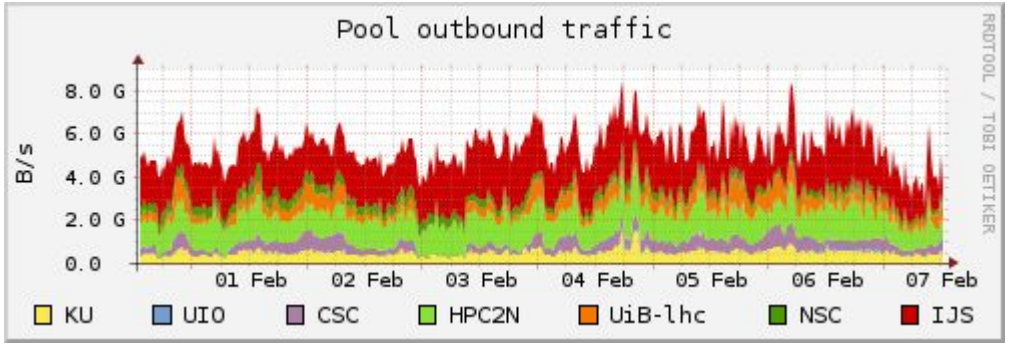


Input trf/copy
Input direct I/O
Output trf/copy
Output direct I/O

- Similar for all other SI clusters
  - non-ATLAS users use JSI and ARNES dCache
- Works rather well, but requires big WAN pipes
- 20Gb/s dedicated LHCONE link saturated when all jobs are I/O heavy

NDGF-T1 dCache traffic

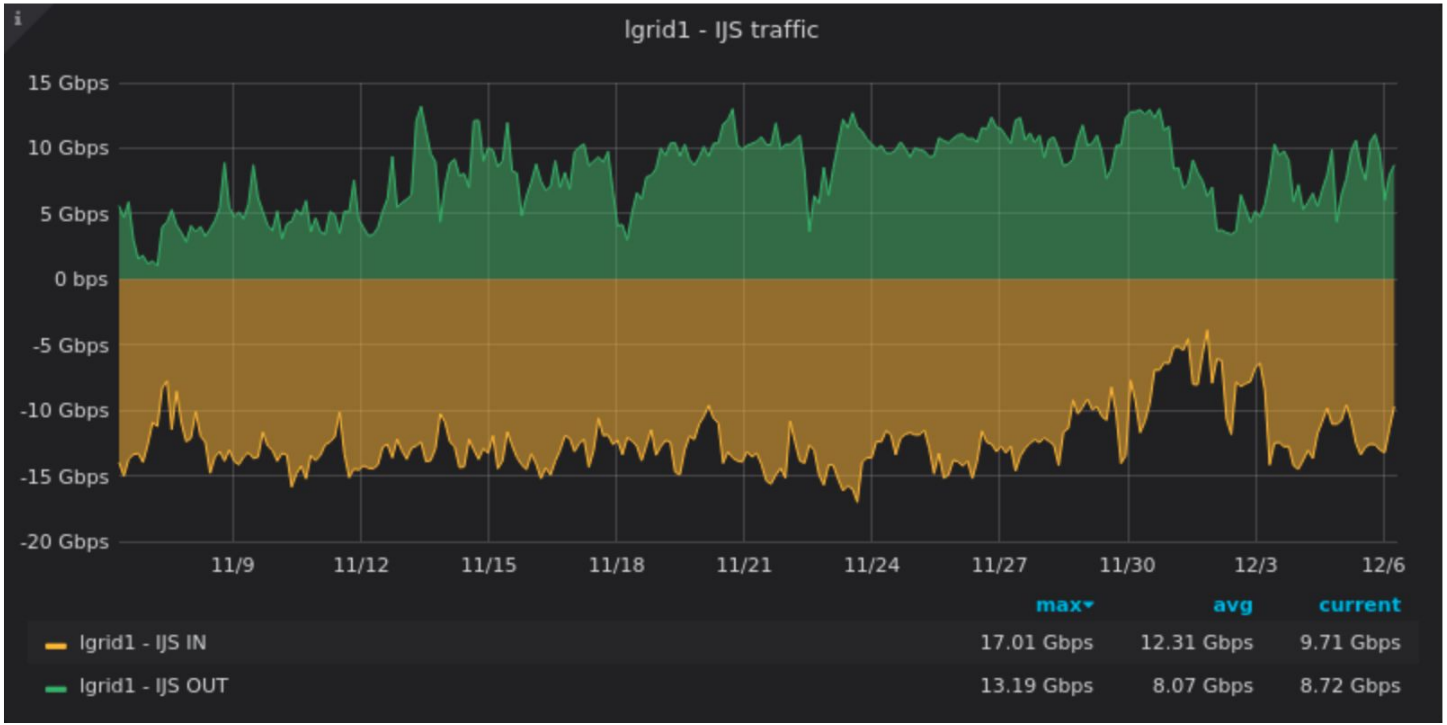




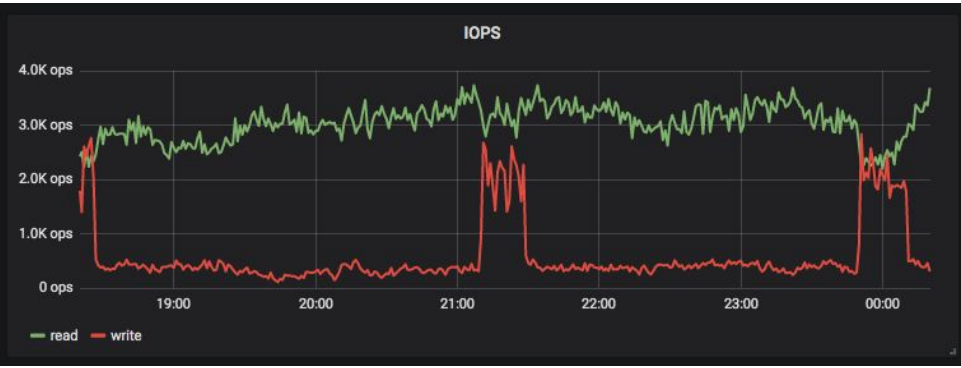
LHCONE - JSI traffic

lgrid1 - IJS traffic

Traffic between lgrid1 and to IJS LHCone router (gridgw).



CephFS rate/iops



- Larger write IOPS - cache cleanup
- With heavy jobs:
  - Read goes up to 4GB/s
  - IOPS up to 20k

Things to improve

- Data placement in NDGF-T1 pools is random
  - Job brokering based on input file dCache pool locality - TODO
- Outputs to random pool
  - Could go to close pool
  - Easy to implement in dCache, but could cause large imbalance in pool occupancy, when local cluster size/pool size varies a lot between sites - in general, Output ~ 1/10 Input
- Remote direct I/O
  - Most of analysis reads a fraction (<10%) of inputs - queue already implemented at SiGNET
  - Direct I/O vs full input transfer: no of jobs in 1st queue is 5 times higher (though jobs are also different)
  - To experiment with XCache, but limited community interest apart from LHC

Related to dCache, Rucio QoS implementation

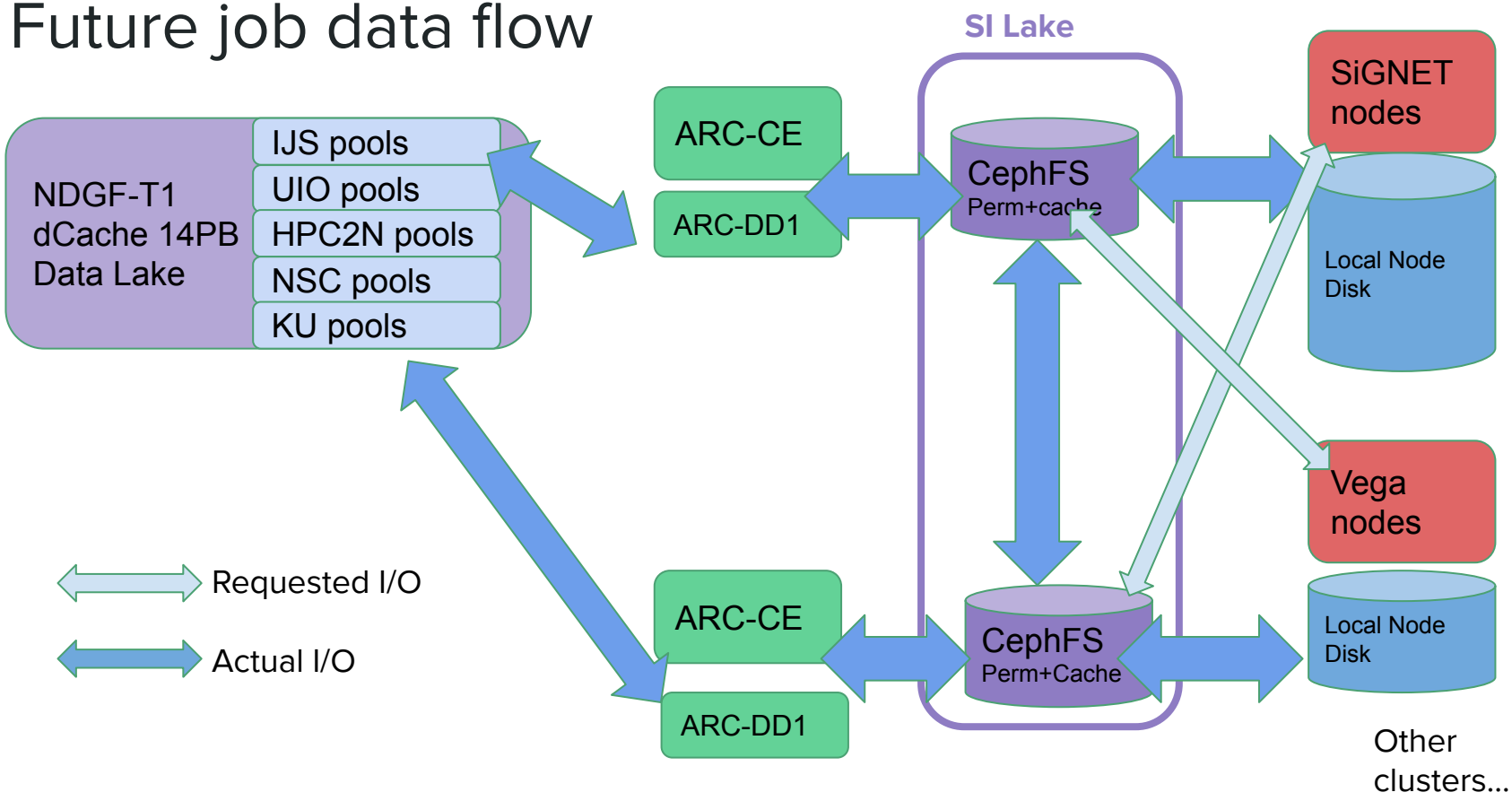
CephFS, local disk throughput

- Current CephFS: 280 HDD, 750 TB, 7 servers
  - Metadata on 20 small SSDs
- Bottlenecks:
  - Can reach up to 20k IOPS, ~4GB/s (current LAN on nodes is the limit)
  - Before ceph wpq, frequent problem with slow requests
  - Currently: 2OSD HDD/batch node - faster for input than single local HDD
  - Too slow for workdir (large mds stress, frequent small iops)
- Node size “problem” with upcoming hw
  - 128C/256HT Rome, more in the future - ~4000 hs06/node
  - Local HDDs out of question
  - ATLAS heavy jobs use 2-3Gb/s LAN
  - Local disk: 5TB with the WLCG recommendations , expensive for SSD/NVMe - fast shared FS might be cheaper and more performant

Speeding up Ceph

- Ordered 42 4TB SSDs for fast CephFS cache
  - 140eur/TB vs ~35 for HDD
- To evaluate:
  - Ceph Tiering between HDD and SSD, though there are concerns on performance
  - Copy2ssd before execution, clean after - need for QoS
  - Use SSD only for cache, with size=1, though for SiGNET cluster, turnaround is 75TB/day
  - Experiment with BeeGFS on Demand (for job scratch) - private FS (shared for parallel jobs)
  - Experiment with multi-site (SiGNET, NSC
  - To report on one of the next meetings
- Similar will be used on Vega HPC as well

Future job data flow



Disk Storage Cost Considerations

- Permanent reliable storage: size 3 or more, Price factor 3
- More risky(?) EC: eg 8+3, Price factor (k+m)/k - 1.4
- Cache storage: Price factor 1 or 2 for HDD, 4 or 8 using NVMe (to get lower)
- Raid 6: Price factor typically 1.15, 1.3-1.4 with dedicated external RAID box
- HDD vs NVMe throughput:
  - HDD max 90 * 0.15GB/s - ~13GB/s
  - NVMe limited by network, 4x100Gb/s - ~50GB/s with 24 SSDs and PCI4/5
  - Factor of 4 in cost, factor ¼ in throughput - roughly equal in terms of performance for sequential read/write, HDD much worse for random
- Optimizing cost vs performance is non trivial, best configuration heavily depends on usage patterns

Plans on Large (Euro)HPCs

- Several site storage hierarchies:
  - Tape - though typically only for archival
  - Large (Distributed) Capacity (OS) - data lake in LUMI CSC
  - HDD shared FS - for > EByte not shared any more, input migration to Fast is needed
  - Fast shared FS - fast vs cheap only recently
  - Shared memory across nodes - already used by large parallel apps
  - Local NVMe or attached through NVMeOF (burst buffers)
  - Local Memory (eg persistent Optane DC DIMM)
- Large data jobs should be aware and use all those for best performance
- Even smaller centers might have 4 or 5 of those
- There are some tools/sw to do migration automatically, but not sufficient and universal - more intelligent QoS and DDM needed
- Big challenge how to address it in a coherent automated way
  - Top level orchestration (eg Rucio), automated by access, optimized based on application behaviour. SLURM already supports data-aware plugins

What needs to be addressed?

- Multiple clusters:
  - Share the Ceph cache (eg ARC-CE data service, or Ceph multi-site)
  - Minimize WAN to GEANT and WAN between the clusters
- Topology:
  - For data lakes, other large storages, QoS with data locality is a must
  - Potential side effects need to be addressed (eg placement, occupancy imbalance)
- Cost vs Performance:
  - SSD/NVMe are now affordable for caches, not yet for large permanent storage
  - With CephFS cache, even size=1 could be used (does not hurt too much if it breaks once a year)
- Ceph for permanent storage:
  - Replication 3 is expensive, EC might be risky, not sure if much cheaper than Raid6
  - But Raid6: days for full recovery, risky if raid controllers break (happened at JSI)
  - Ceph: hardware agnostic