Lustre in the Compute Canada Federation and the deployment of Beluga

LUG 2019

simon.guilbault@calculquebec.ca

compute canada | calcul canada

UNIVERSITÉ LAVAL

Calcul Québec

Financial Partners

POLYTECHNIQUE MONTRÉAL

UQTR

Savoir. Surprendre.

OURANOS

Université de Montréal

McGill

INNOVATION.CA

CANADA FOUNDATION FOR INNOVATION | FONDATION CANADIENNE POUR L'INNOVATION

UNIVERSITÉ Concordia UNIVERSITY

UNIVERSITÉ LAVAL

Québec

ÉTS

UNIVERSITÉ DE SHERBROOKE

risq

LE RÉSEAU AU SERVICE DU SAVOIR

UQÀM

compute canada | calcul canada

canarie

compute canada
calcul canada

What is Compute Canada ?

As you can see, Calcul means Compute

Compute Canada

- Similar to XSEDE
- More than 200 experts employed by 37 partner universities and research institutions across the country.
- 5 National systems
  - and various smaller legacy systems
- Funded by The Canada Foundation for Innovation (CFI), provincial partners and academic institutions.
- Free access to compute, storage, experts to researcher in Canada

User software environment

Calcul Québec

- Stored on CVMFS
  - Based on Nix, Easybuild and lmod
- Compiled for AVX512, AVX2, AVX, SSE3
- Support for Centos 6 and 7
  - Also used on legacy clusters
- Complete list of modules (> 620):
  - https://docs.computecanada.ca/wiki/Modules
- Pip wheelhouse with > 2065 packages
  - https://docs.computecanada.ca/wiki/Available_Python_wheels
- Restricted repos for commercial softwares
- Talk at PEARC19

Lustre in Compute Canada

- Various implementations
    - LDISKFS vs ZFS
    - MDT on HDD or SSD
    - Appliance, distributed RAID or JBODs
    - Omni-Path or Infiniband
    - Vendor support, Whamcloud support or self-support
    - $HOME on Lustre or NFS
    - DNE available or not

Nearline with HSM and TSM

Calcul Québec

- TSM is already used for backup of /project
- Inter-site replication of TSM archive
- Robinhood as a policy engine
- lhsmtool_cmd call a python script, “ct_tsm”
  - https://github.com/guilbaults/ct_tsm
  - Use UUID instead of FID on the backend
    - Could move the stubs to a new filesystem when retiring this one
    - Slight hack in Robinhood’s database for lhsm_remove

Deployment of Beluga
Storage design choice

Lustre for 3 different purposes

3 Filesystems

- /home
  - Small iops
  - Per user quota 50GB, 500k inodes
- /project and /nearline
  - Bulk of the storage, long term
  - Per group quota, depends on allocation
  - 1TB per default, 10TB with a ticket, > 10TB with a allocation
- /scratch
  - Per user quota, 10TB, could be increased depending on the purge policy

ZFS everywhere

- Bad experiences previously with proprietary distributed RAID
  - Frequent dual-disks failure during rebuild
- RAIDZ3
  - Peace of mind on 10TB disks
- LDISKFS corruption
  - By the RAID layer
    - Corruption of one of the 32 data folders on a OST
  - Upgrade from 2.1.6 to 2.5.3 and enabling dirdata
    - LU-2638 and LU-5626
  - fsck is scary to run (-f and cross fingers)

ZFS Compression

- As of end of April, FS are filled at around ~ 20%
- Ratio of 1.43x on /project
- Ratio of 1.34x on /scratch

Lustre DNE

- 8 MDS and 8 metadata targets
  - 4 for /project, 2 for /scratch, 2 for /home
- DNE 1 used
  - Directories are allocated randomly on MDT by cc-mkdir
    - Pull the users and projects from LDAP and create the directories
- DNE 2 supported
  - Available if needed by a user

PFL settings

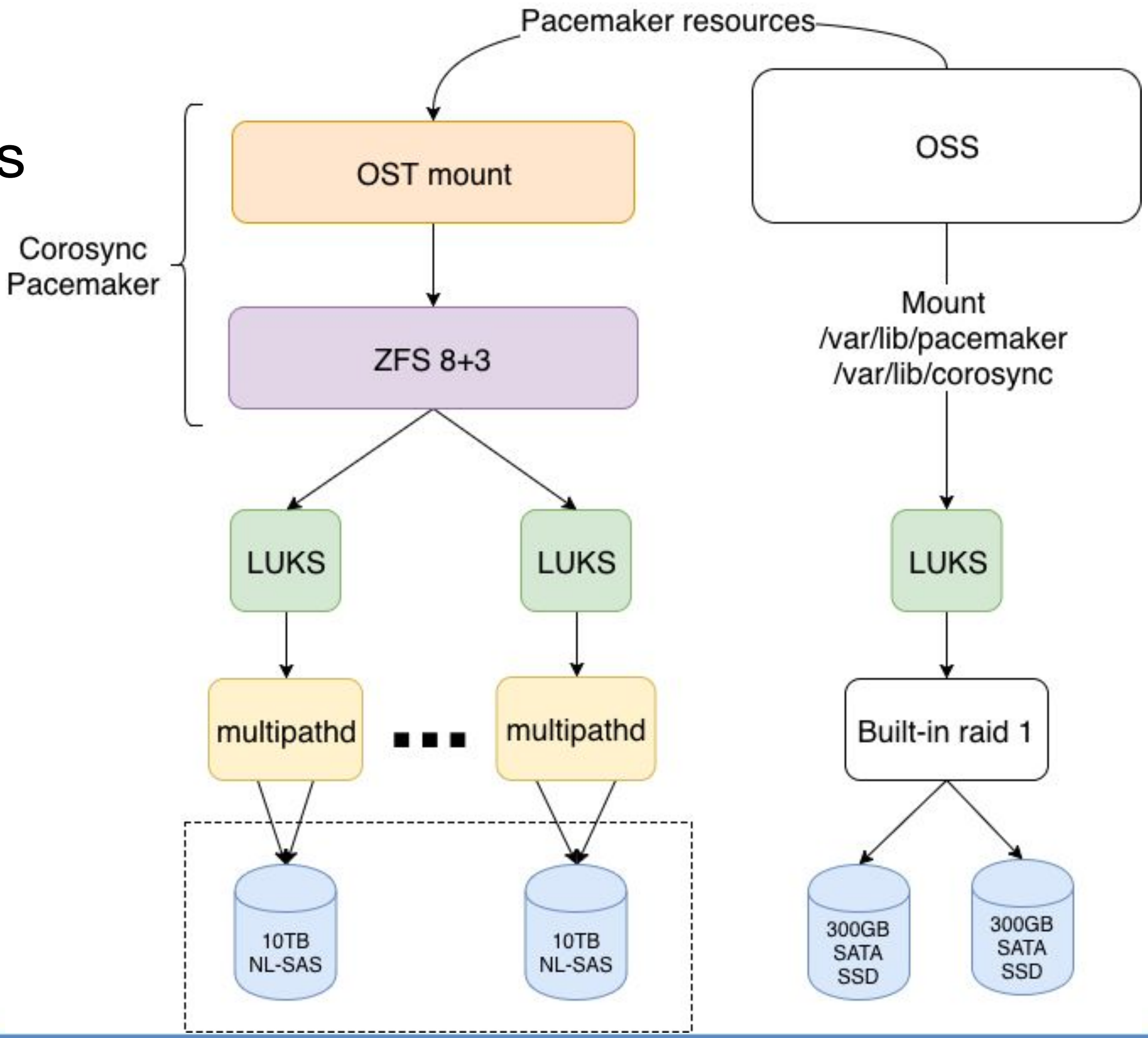
- Progressive file layout by default
  - 1 stripe for 0-128MB
  - 4 stripes for 128MB-4GB
  - 16 stripes for 4GB-16GB
  - 64 stripes above 16GB
- “Small targets” of 8+3, without stripping with ZFS
- Chosen arbitrarily, knowing that majority of the files are under 128MB.
- Will be adapted for DoM

SAS multipath and encryption

- Multipath
  - Previous RAID (temporary) failure
    - mdadm assemble --force…
  - Prevent failover / raid rebuild when looking at a cable the wrong way
  - JBOD SAS module crash
- Encryption
  - Security (of course…)
  - Prevent ZFS from detecting and importing /dev/sdb instead of /dev/mapper/jbod04_slot01

Layers of I/O

Prevent ZFS from importing the devices without multipath



Hardware and software provisioning

Lustre Hardware (OSS/MDS)

- OSS:
    - Dell R740
    - 2x Skylake Silver 4140 (8C, 2,1GHz)
    - 192 GB ram (all channels used)
    - 4 dual ports HBA SAS3 (6GB/s per card)
    - Infiniband EDR
- MDS:
    - Same as OSS, but Dell R640 (1U) and 1 SAS HBA

Lustre Hardware (OST/MDT)

Calcul Québec

- Seagate Exos E 4U106
  - 106x 10TB NL-SAS (ST10000NM0096)
  - 4U, 45 inches deep
  - 8x SAS3 ports

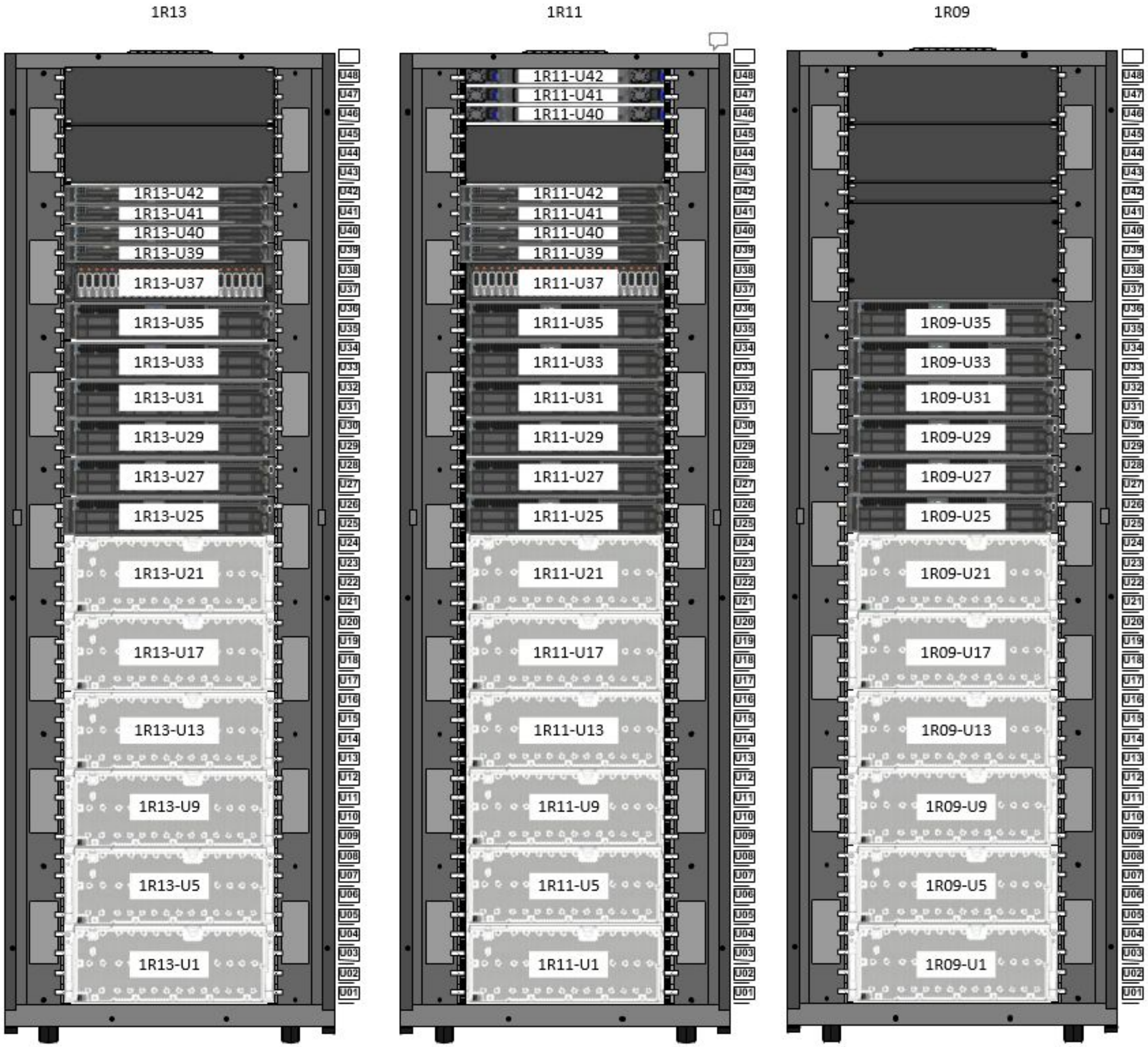
- Dell MD1420
  - 16x 1.92TB (Toshiba PX05SVB192Y, 3 DWPD)
  - 2U
  - 8x SAS3 ports

- HBA
  - LSI Logic / Symbios Logic SAS3008 PCI-Express Fusion-MPT SAS-3

Lustre building block

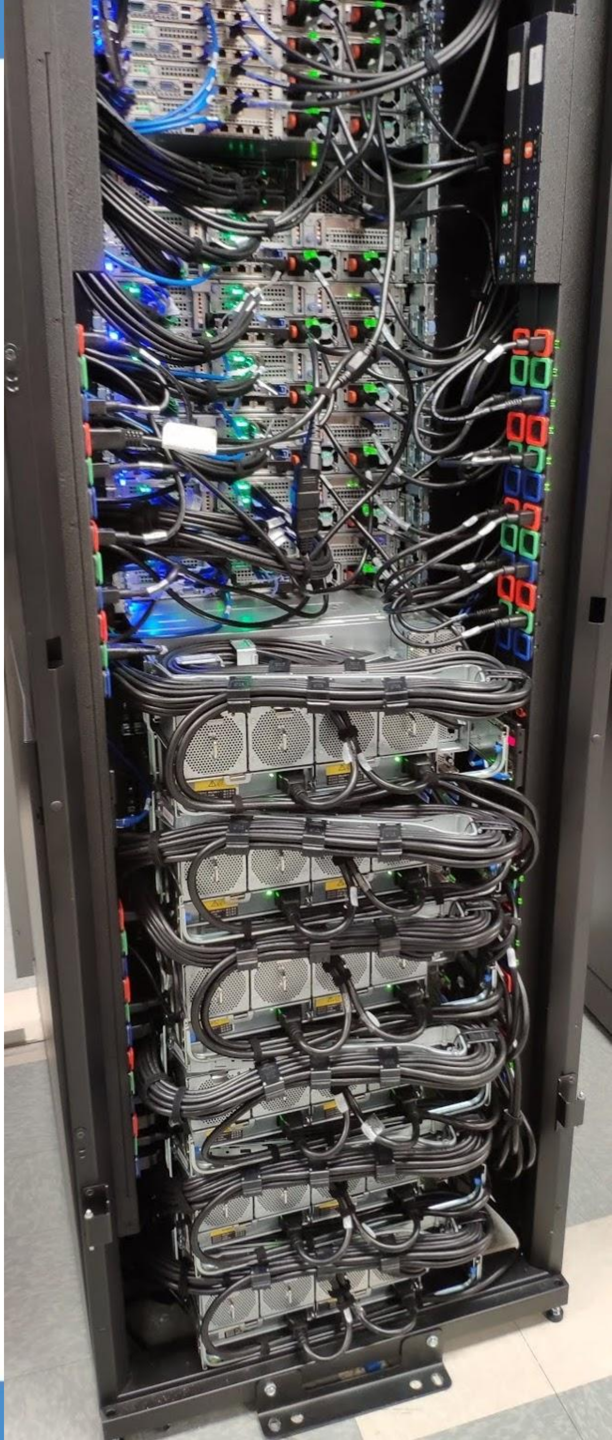
- OSS/OST pair
    - 2x R740
    - 2x JBODs
    - 212 disks
    - 16 SAS cables
    - 848 SAS links per server
    - 75TB per target
    - 1.4PB per BB
- MDS/MDT pair
    - 2x R640
    - 0.5x JBOD, shared by 4 MDS
    - 4 SSD per MDT (16/24 slots)
    - ~300M inodes per MDT (without DoM)

Rack layout


1R13
1R13-U42
1R13-U41
1R13-U40
1R13-U39
1R13-U37
1R13-U35
1R13-U33
1R13-U31
1R13-U29
1R13-U27
1R13-U25
1R13-U21
1R13-U17
1R13-U13
1R13-U9
1R13-U5
1R13-U1
1R11
1R11-U42
1R11-U41
1R11-U40
1R11-U42
1R11-U41
1R11-U40
1R11-U39
1R11-U37
1R11-U35
1R11-U33
1R11-U31
1R11-U29
1R11-U27
1R11-U25
1R11-U21
1R11-U17
1R11-U13
1R11-U9
1R11-U5
1R11-U1
1R09
1R09-U35
1R09-U33
1R09-U31
1R09-U29
1R09-U27
1R09-U25
1R09-U21
1R09-U17
1R09-U13
1R09-U9
1R09-U5
1R09-U1
U48
U47
U46
U45
U44
U43
U42
U41
U40
U39
U38
U37
U36
U35
U34
U33
U32
U31
U30
U29
U28
U27
U26
U25
U24
U23
U22
U21
U20
U19
U18
U17
U16
U15
U14
U13
U12
U11
U10
U09
U08
U07
U06
U05
U04
U03
U02
U01

Rack issues

- Extra deep, wide and tall racks
  - Rear door won't close because of length of the JBODs
- 3x 30A PDU
  - Plugs facing inwards



OS and provisioning

- Stateless OS with xCAT
  - corosync/pacemaker state on local disks
  - Puppet with a shared certificate and hiera per hostnames
    - hiera-eyaml with GPG for disk passphrases
    - Configuration in Git, modules handled by R10k
    - Same base config as a compute node
    - Update by rebooting with failover
      - Done live from 2.10.6 to 2.10.7
- Lustre recompiled without ldiskfs
  - Latest Centos 7 stock kernel

Puppet module for Lustre

- https://github.com/guilbaults/puppet-lustre
  - Formating (with an interlock, require a file in /tmp and disks need to be zeroed)
  - HA stack config with corosync and pacemaker
  - Backport some scripts from newer Lustre versions
    - LU-8384, patched in 2.12.1 (systemd)
  - Add some monitoring hook for NRPE
    - check_targets
    - check_lustre_healthy
    - check_zfs

Forked sasutils

- From Stephane Thiell (stanford-rc/sasutils)
  - Xyratex branch in https://github.com/guilbaults/sasutils
- Can handle Xyratex 7 segments display naming scheme
  - `/dev/mapper/jbod09-bay46`
    - For disk I/O
  - `/dev/mapper/single_jbod09-bay46`
    - For smartctl commands
  - Also handle SP-34106

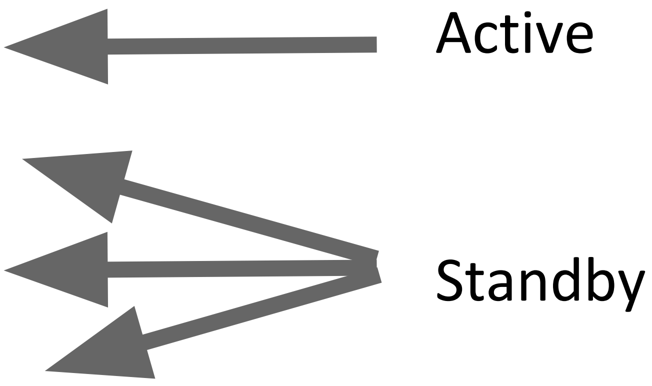
Script to find disks in a JBOD

- https://github.com/guilbaults/blinkenlights
  - Loosely based on the commands of our old SUN J4400
  - Can toggle LED and power off/on a disk
    - `blinkenlights --rtr /dev/mapper/jbod00-bay00`
    - `blinkenlights --insert /dev/mapper/jbod00-bay00`
  - Can resolve slot position or disk name
    - `blinkenlights --rtr /dev/sdx`
    - `blinkenlights --locate-on /dev/mapper/jbod00-bay00`

Patched multipathd

- Using random priority to spread the IO on multiple cards/cables without round-robin on each request
  - By default, random is only 1 to 10

```
multipath -ll
35000c500a62000b3 dm-89 SEAGATE ,ST10000NM0096
size=9.1T features='1 queue_if_no_path' hwhandler='0' wp=rw
|-+- policy='round-robin 0' prio= 1119 status=active
| `- 17:0:109:0 sdmr  70:304  active ready running
|-+- policy='round-robin 0' prio= 6037 status=enabled
| `- 15:0:196:0 sdaca 135:544 active ready running
|-+- policy='round-robin 0' prio= 2291 status=enabled
| `- 16:0:221:0 sdafh 68:880  active ready running
`-+- policy='round-robin 0' prio= 4277 status=enabled
  `- 18:0:221:0 sdafj 68:912  active ready running
```



Zpool status (device name)

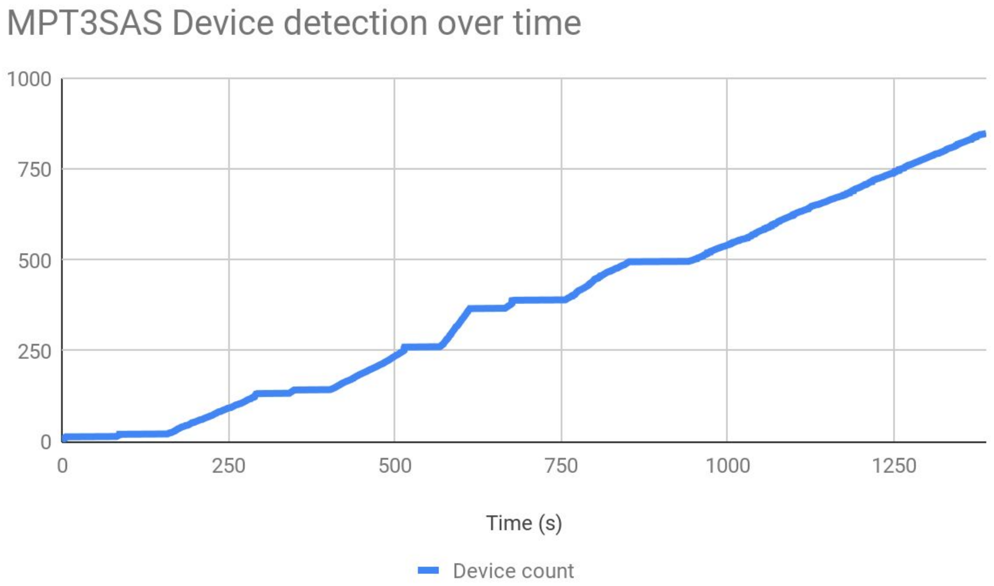
- jbod02-bay100 is a encrypted device
- open_jbod02-bay100 is the decrypted device made available by LUKS

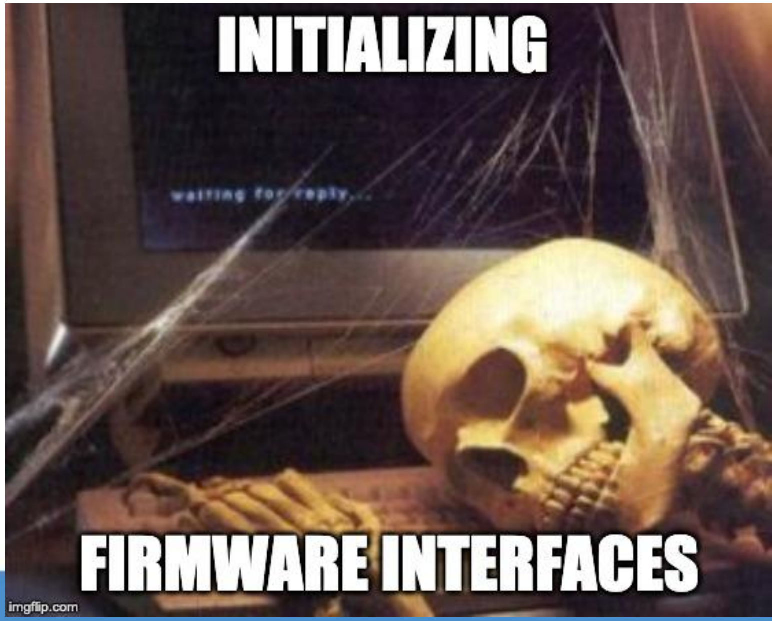
```
        NAME                    STATE     READ WRITE CKSUM
        lustre04-ost8           ONLINE       0     0     0
          raidz3-0              ONLINE       0     0     0
            open_jbod02-bay100  ONLINE       0     0     0
            open_jbod03-bay100  ONLINE       0     0     0
[...]
```

- Easy to open a hardware ticket and turn the disk off with blinkenlights

Slow mpt3sas detection speed

- BIOS = 14 minutes
- modprobe mpt3sas = 23 minutes
- udev probing = 15 minutes (could be optimised)
- udev takes 30 seconds on a dual path 84 slots JBOD



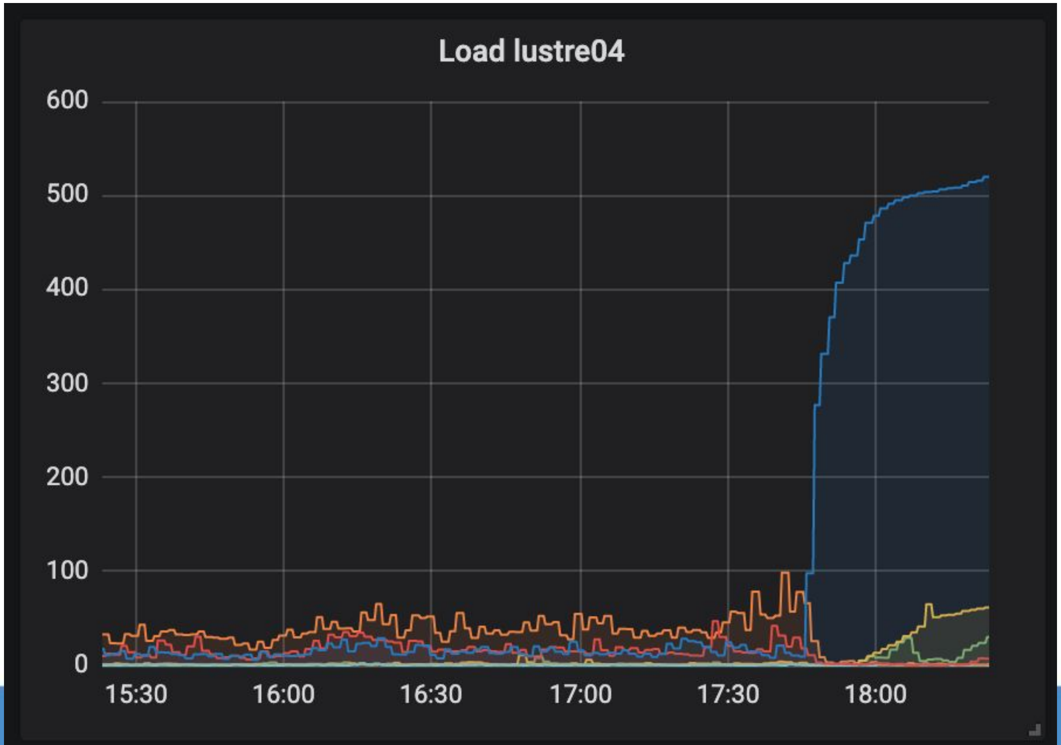


Stable ?

Mostly

Deadlock on scratch

- Also trigger the ZFS MMP timeout
  - Still using default ZFS timeout values
- A few patch in 2.12.1



A few timeout with some drives

- Multipath is currently using “Test Unit Ready” command to verify the paths
  - 1 HDD had timeout on 4 paths at the same time
    - Firmware bugs ?
    - Happen again after a month on the same HDD

```
14:28:16 remaining active paths: 3
14:28:17 remaining active paths: 2
14:28:17 remaining active paths: 1
14:28:17 remaining active paths: 0
14:28:28 remaining active paths: 1
14:28:28 remaining active paths: 2
14:28:29 remaining active paths: 3
14:28:29 remaining active paths: 4
```

Lost the disk for 10 seconds

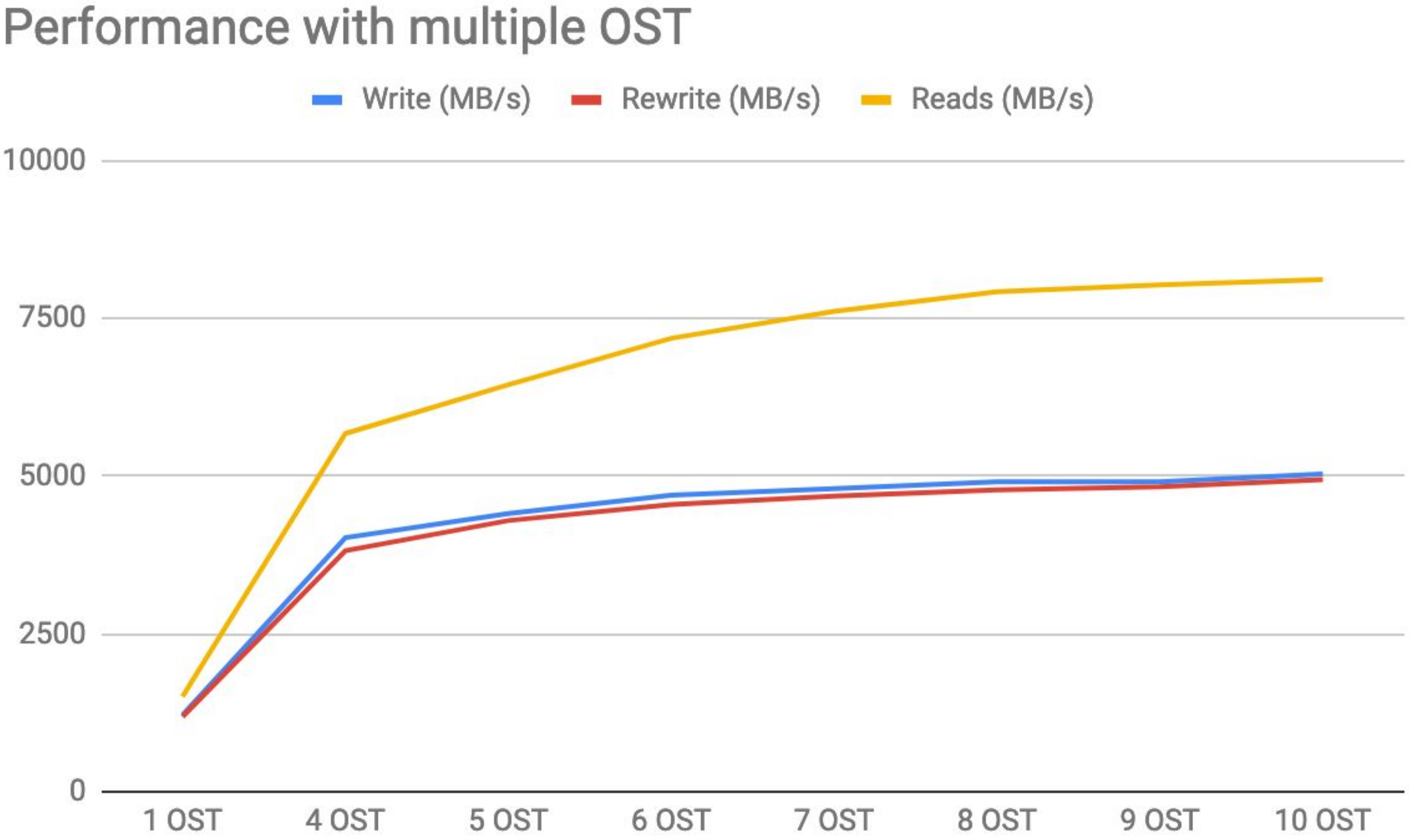
Benchmarks

And trying to find bottlenecks

Raw IO and OBDfilter benchmarks

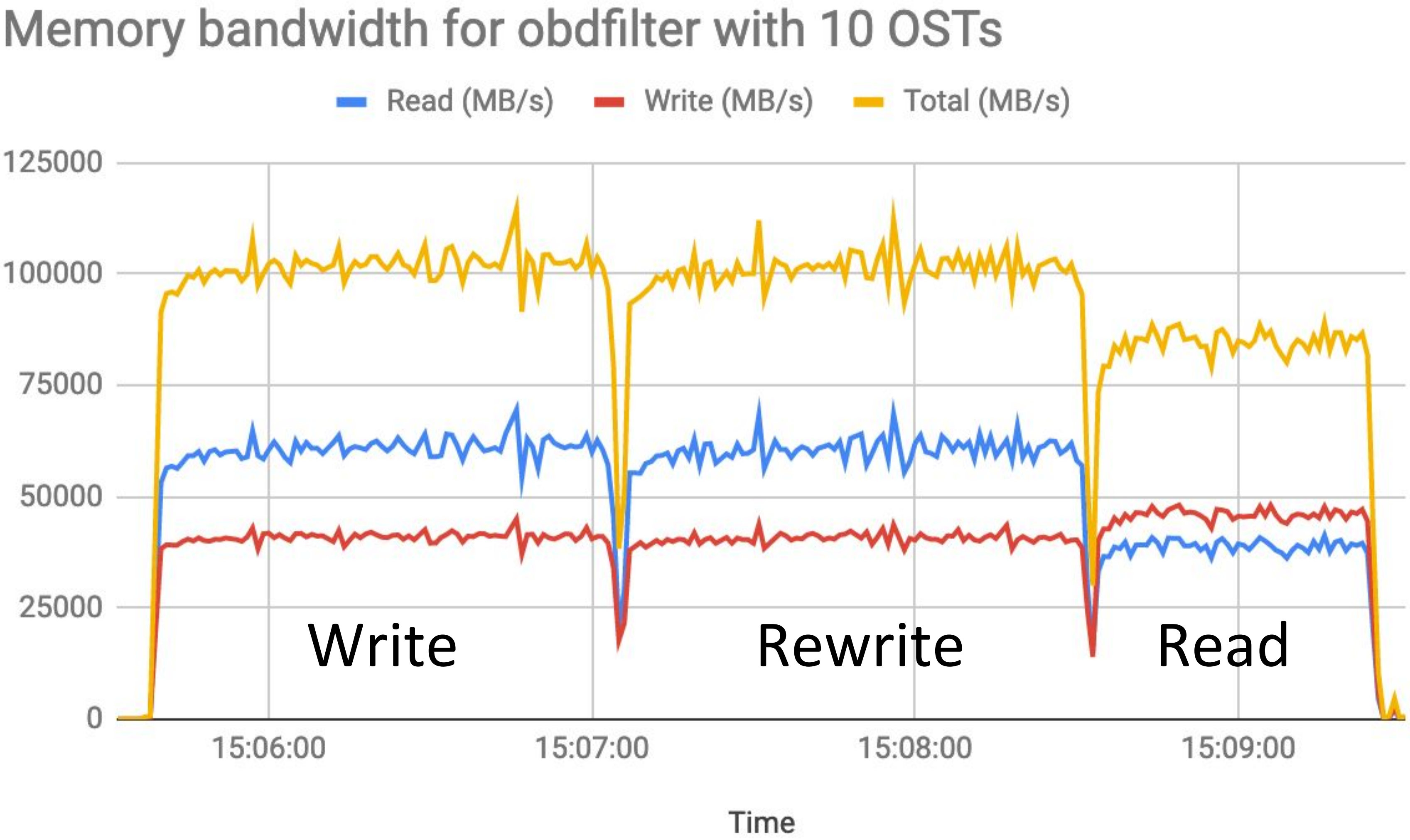
- Each SAS card is advertised as 6GB/s
- VDbench can reach 22GB/s over 4 cards
  - 19 GB/s with encryption
- OBDFilter results with 10 OSTs and encryption (½ Building block):
  - Write 6.2GB/s
  - Rewrite 6.2GB/s
  - Read 9.5GB/s
  - ~20% of idle cpu cycle

Performance limits with obdfilter



Memory bandwidth limitation ?

- Trying to find the bottleneck
- Intel Processor Counter Monitor (PCM) for additional metrics
  - https://github.com/opcm/pcm
  - Metrics: Cache miss, IPC, memory bandwidth per channel, UPI bandwidth ...
- UPI bandwidth is not full according to PCM



Memory bandwidth for obdfilter with 10 OSTs

Maximum of 10GB/s on a memory channel

Average at 8.8GB/s, fluctuating between memory channels

Lets reduce the memory bandwidth

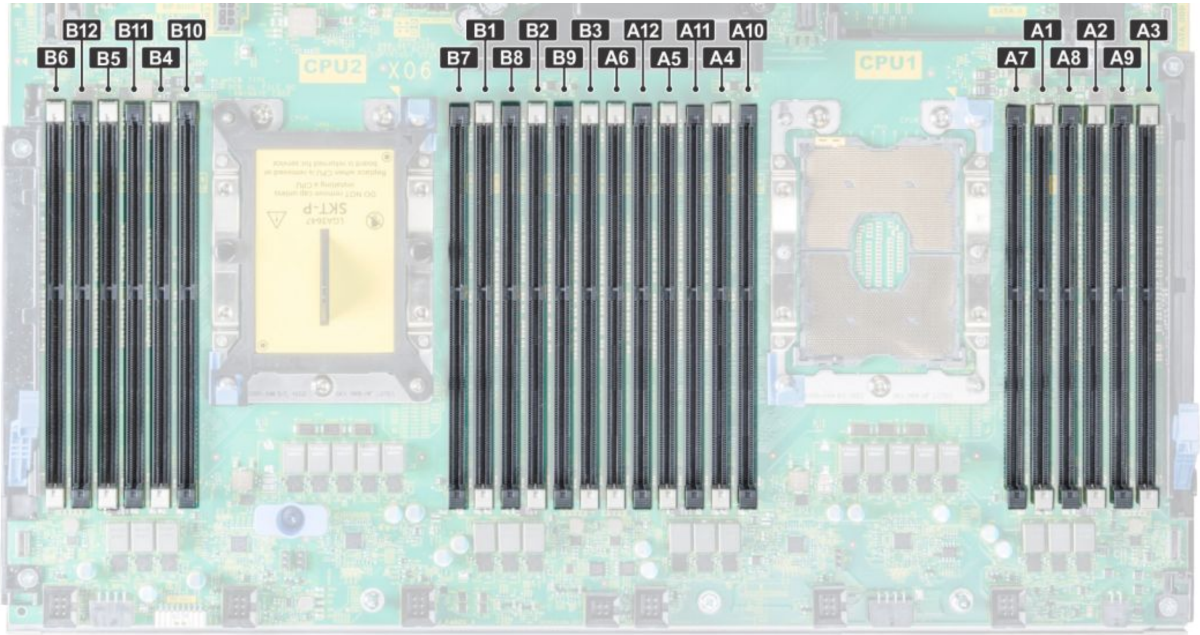
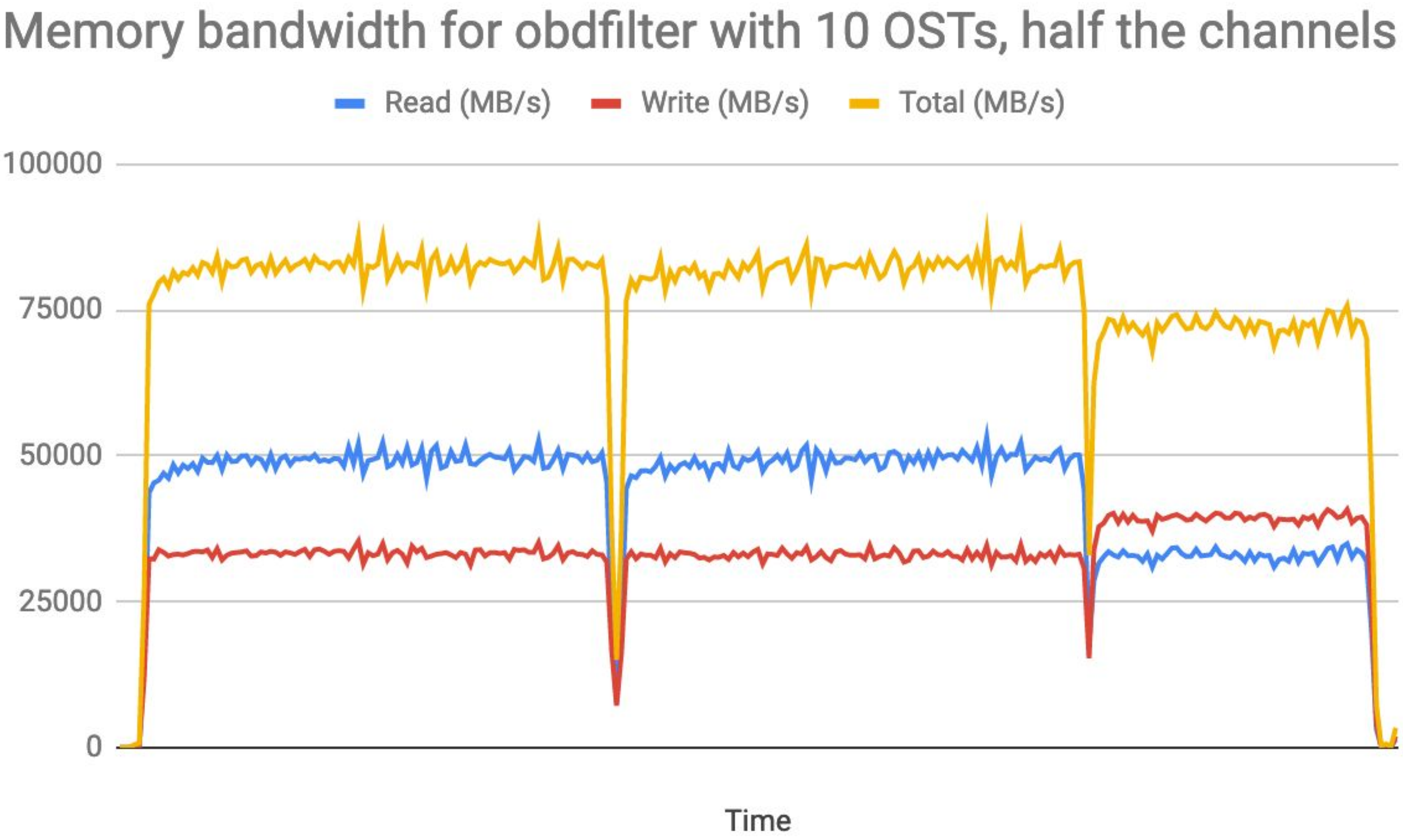


Figure 12. Memory socket locations

Memory channels are organized as follows:

Table 8. Memory channels

Processor	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
Processor 1	Slots A1 and A7	Slots A2 and A8	Slots A3 and A9	Slots A4 and A10	Slots A5 and A11	Slots A6 and A12
Processor 2	Slots B1 and B7	Slots B2 and B8	Slots B3 and B9	Slots B4 and B10	Slots B5 and B11	Slots B6 and B12



Lower observed bandwidth and performance, but not 50% less (6.2GB/s -> 5.0GB/s)

Impact on OBDFilter

Test type	12 channels	6 channels
Write	6.2GB/s	5.0 GB/s
Rewrite	6.2GB/s	4.9 GB/s
Read	9.5GB/s	8.1 GB/s

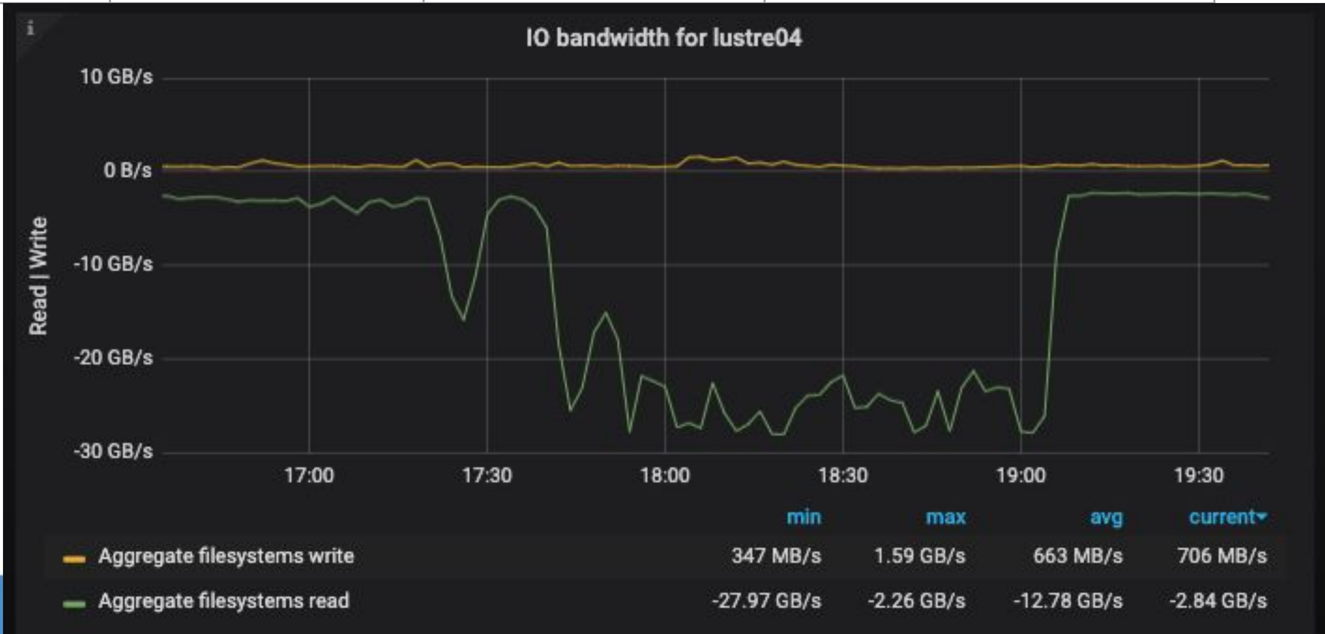
Memory bandwidth does have an impact, but does not seem to be the main limitation on our server.

Might be NUMA related even if UPI bandwidth is not fully used

Filesystem benchmarks

Filesystems	IOR Write	IOR Read	mdtest write (with SELinux)	mdtest stat (with SELinux)	Comments
Project	64 GB/s	44 GB/s	71k iops (DNE2)	217k iops (DNE2)	DNE2 not used in production
Scratch	21 GB/s	24 GB/s	20k iops	20k iops	
Home	7GB/s	26 GB/s	22k iops	20k iops	3 copies

Real users on /scratch :



	min	max	avg	current
Aggregate filesystems write	347 MB/s	1.59 GB/s	663 MB/s	706 MB/s
Aggregate filesystems read	-27.97 GB/s	-2.26 GB/s	-12.78 GB/s	-2.84 GB/s

Questions ?

I have one for the attendee, is anybody using Lustre in Kubernetes / Openshift ?