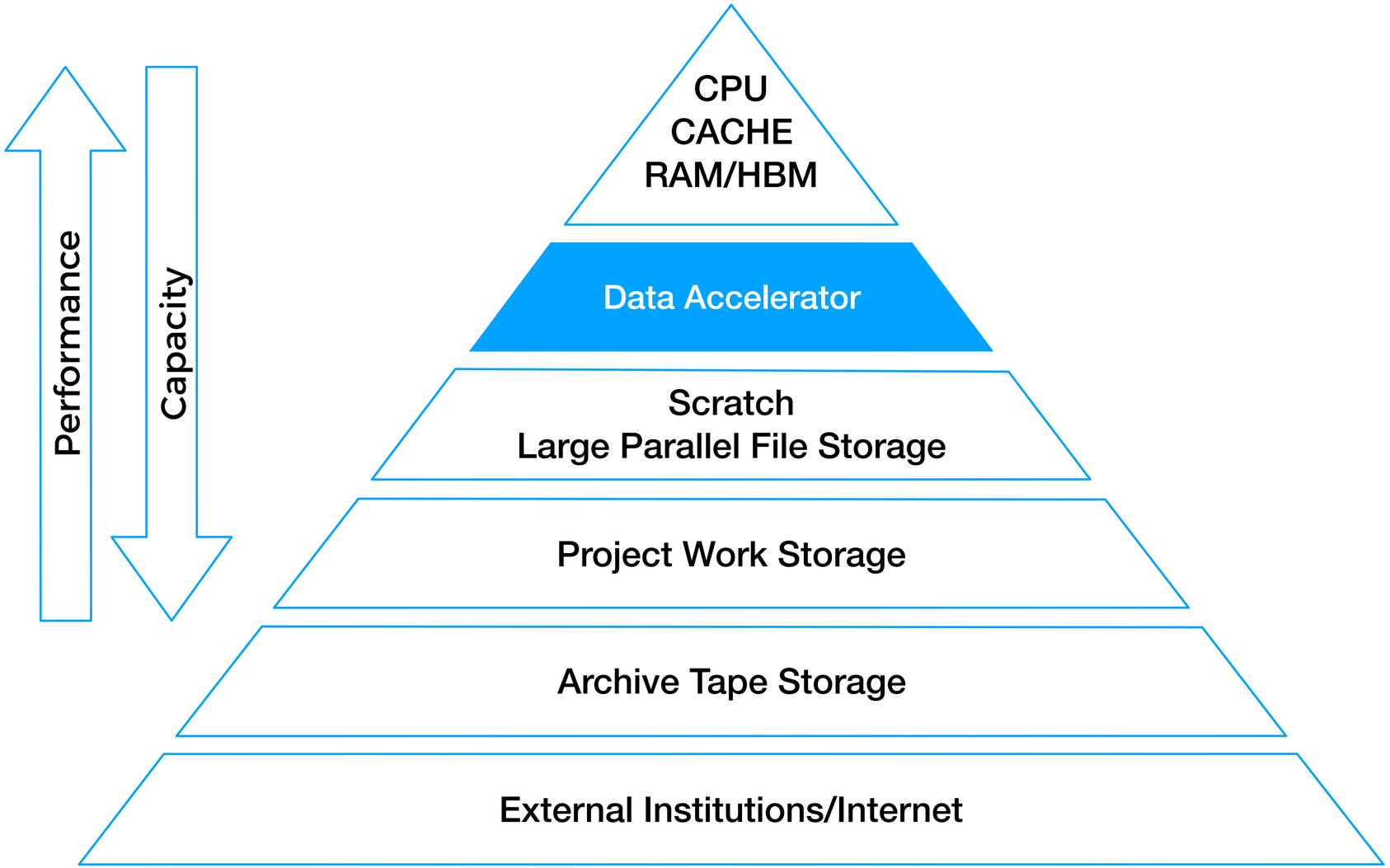
The Data Accelerator

Lustre Administrator and Developer Workshop LAD ‘18

Alasdair King ajk203@cam.ac.uk

UNIVERSITY OF CAMBRIDGE
Research Computing Services



UNIVERSITY OF CAMBRIDGE
Research Computing Services

Cambridge Lustre Estate

UNIVERSITY OF CAMBRIDGE
Research Computing Services

Lustre - Filesystem Hardware

Type	Quantity	Specs
MDS Server	2	Dell R630 - Dual E5-2667v3 3.2GHz 8 Core 128GB RAM FDR IB, 10GB Ethernet
MDT Storage	1	Dell PowerVault MD3420 20x 300GB SAS 15K HDDs Dual RAID controller with 8GB cache
OSS Server	6	Dell R630 - Dual E5-2623v3 3.0GHz 4 Core 64GB RAM FDR IB, 10GB Ethernet
OSS Storage	6	Dell PowerVault MD3460 60x 6TB NL-SAS HDDs Dual RAID controller with 8GB cache
Network	Infiniband	Mellanox SX6036 FDR IB switches 36x 56Gb FDR
	Ethernet	Mellanox SN2410 Spectrum switches 48x 10GbE + 8x 100GbE

*Newer Lustre systems use Intel Omni-Path

Dell MD3420

MDT - 20 disk Raid 10 with 4x SSD cache device

Dell MD3460

Each chassis supports 6x OSTs

Each OST: 10-disk Raid-6, ~58TB per OST

~350TB per chassis

UNIVERSITY OF CAMBRIDGE
Research Computing Services

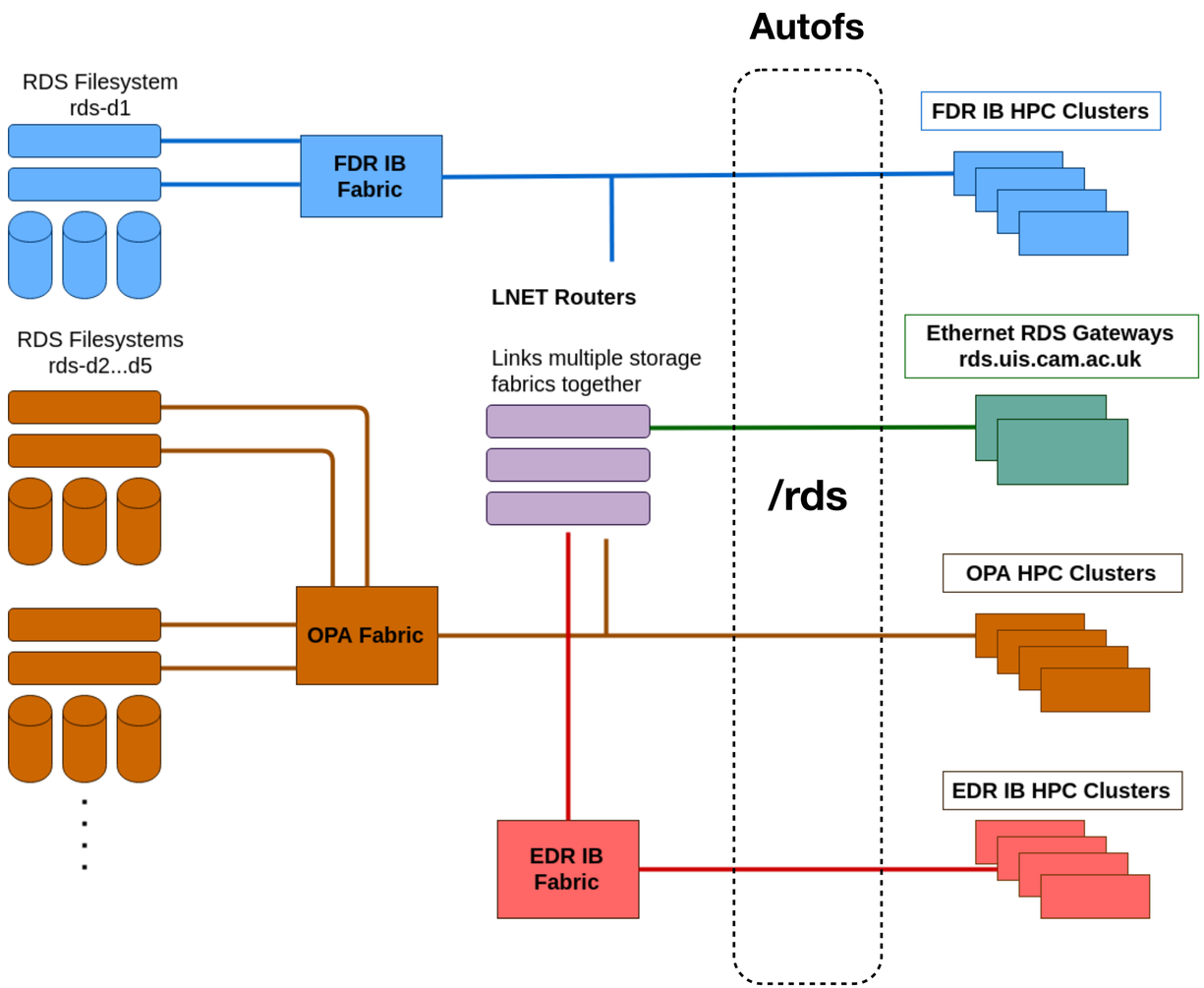
Cambridge Research Data Store

Multiple filesystems abstracted into one global namespace.

Multiple network technologies across various access points.

LNET routing bridges networks to ensure access from all user gateways

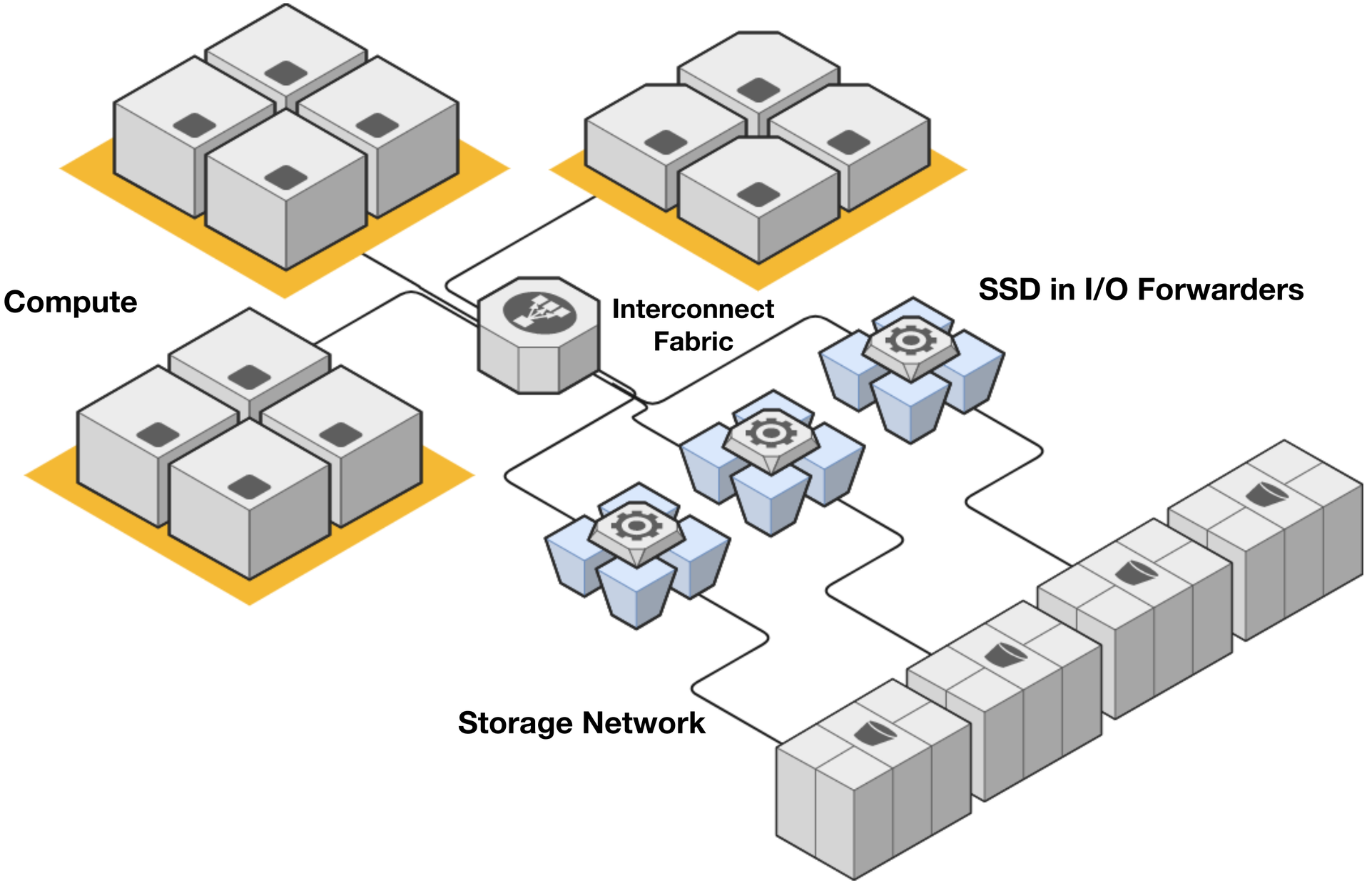
HSM Integrated with Robin-hood.



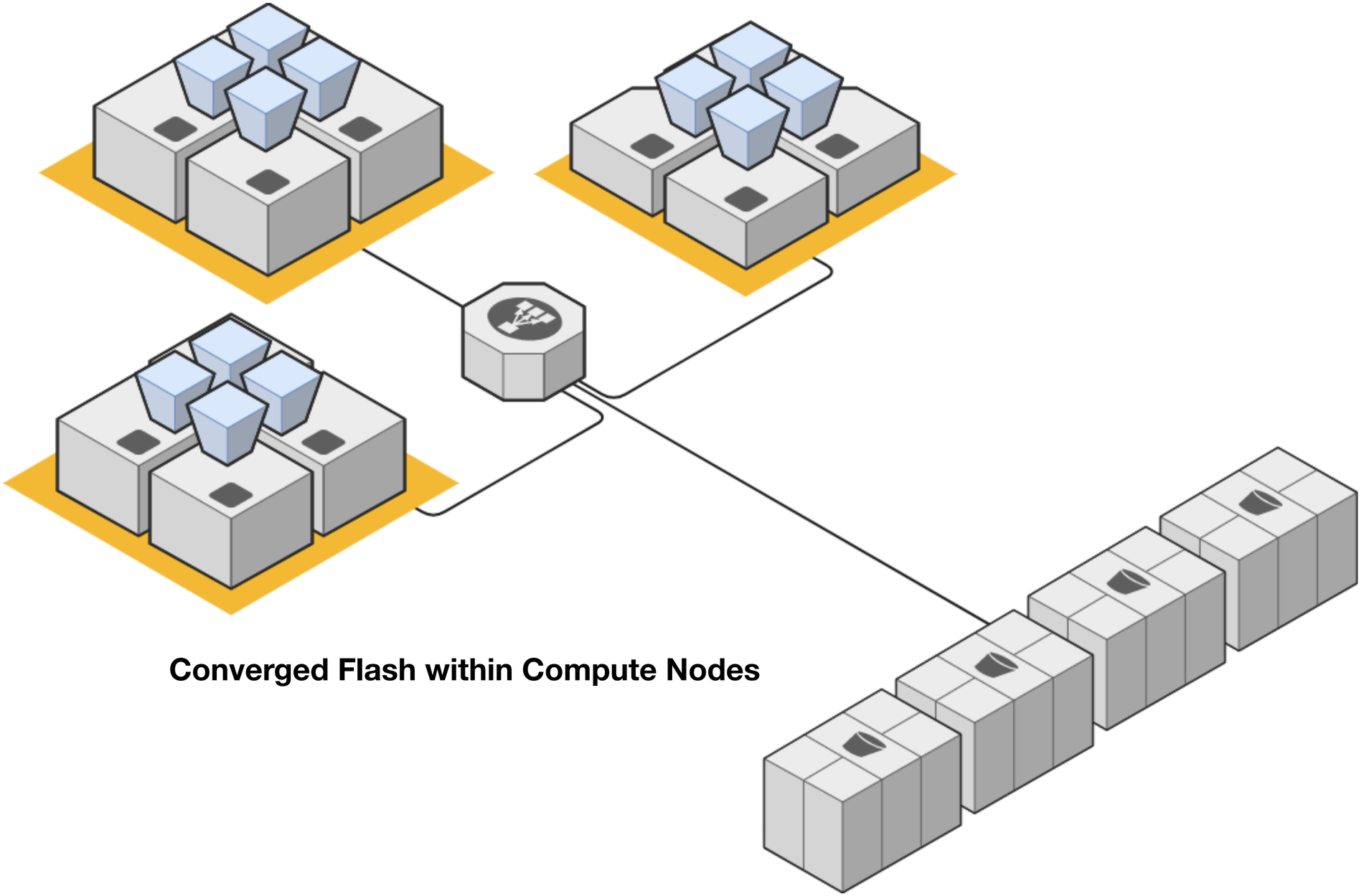
UNIVERSITY OF CAMBRIDGE
Research Computing Services

IO Acceleration Architectures

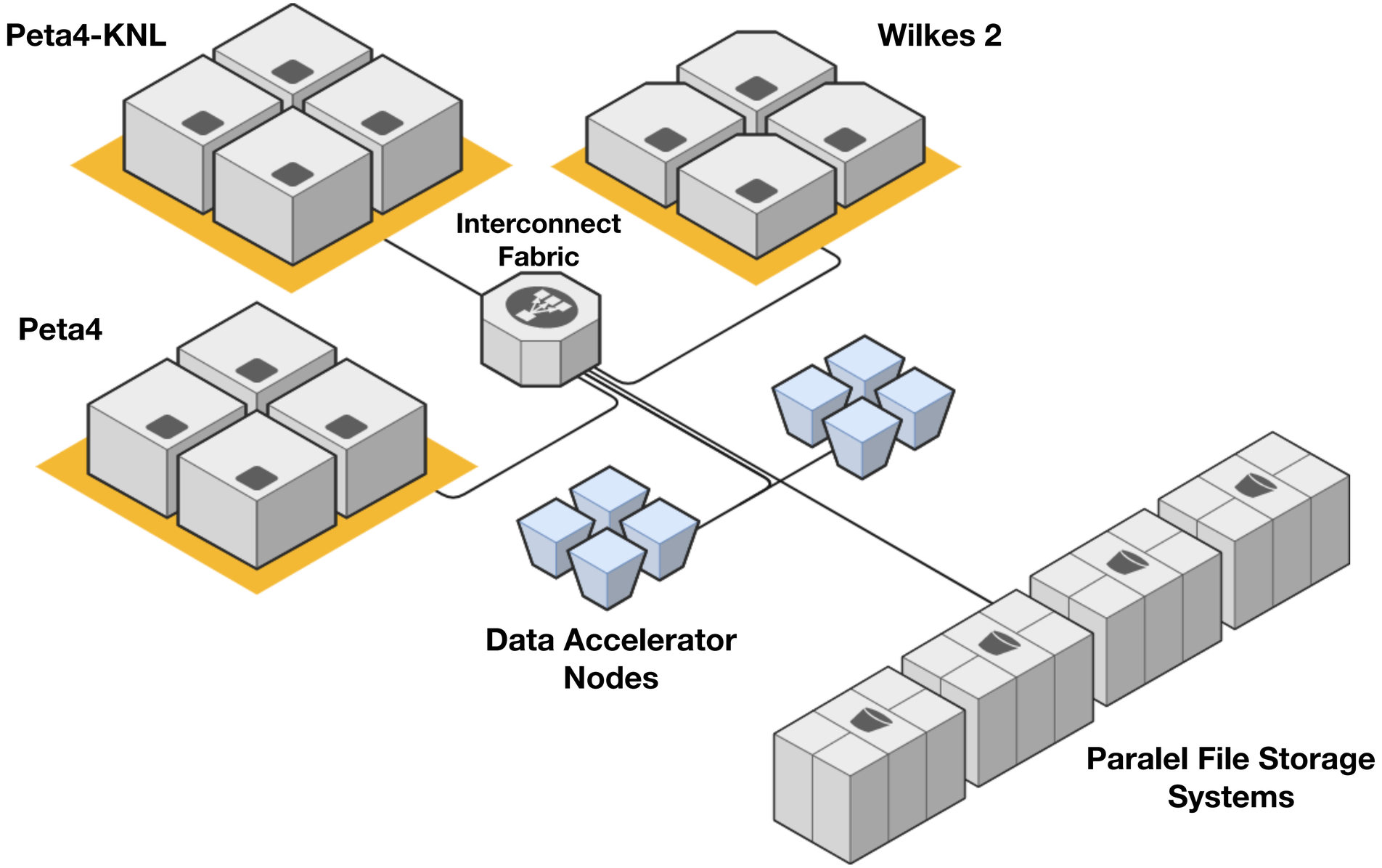
UNIVERSITY OF CAMBRIDGE
Research Computing Services



UNIVERSITY OF CAMBRIDGE
Research Computing Services



Converged Flash within Compute Nodes



UNIVERSITY OF CAMBRIDGE
Research Computing Services

Data Accelerators
Workflows and Features

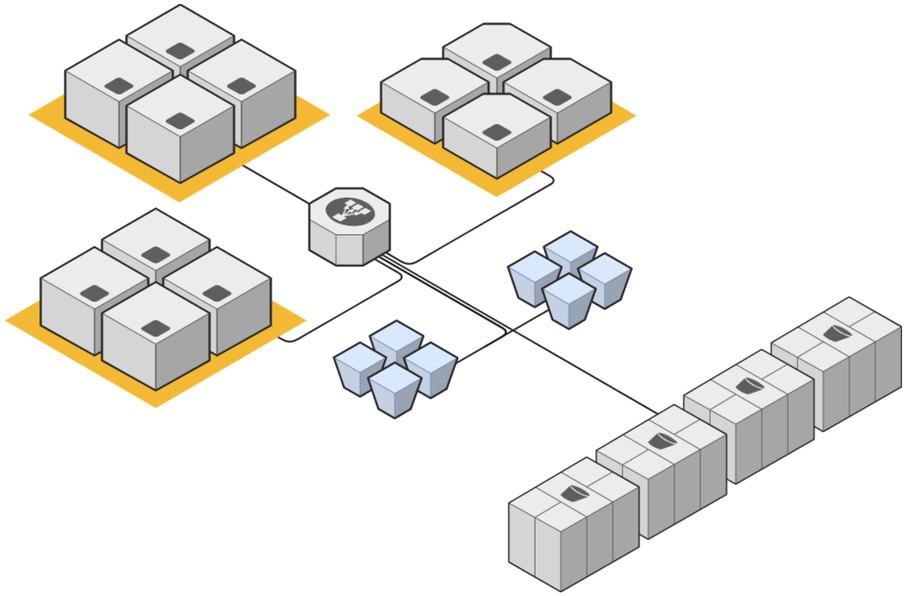
- Stage in/Stage out
- Transparent Cashing
- Checkpoint
- Background data movement
- Journaling
- Swap memory

Storage volumes - namespaces - can persist longer than the jobs and shared with multiple users, or private and ephemeral.

POSIX or Object (this can also be at a flash block load/store interface)

UNIVERSITY OF CAMBRIDGE
Research Computing Services

The Data Accelerator Platform



24 Dell EMC PowerEdge R740xd

2 Intel Xeon Scalable Processors

2 Intel Omni-Path Adaptors

Each with 12 Intel SSD P4600

½PB of Total Available Space

Integration with SLURM and flexible storage orchestrator to provide maximum performance

UNIVERSITY OF CAMBRIDGE
Research Computing Services

Integrating Lustre for the Data Accelerator

UNIVERSITY OF CAMBRIDGE
Research Computing Services

Ansible Enabled Lustre Install

```
ansible-playbook test-dac-lustre.yml -i test-inventory-lustre --tag format --tag reformat_mgs
ansible-playbook test-dac-lustre.yml -i test-inventory-lustre --tag mount,create_mdt,create_mgs,create_osts,client_mount
ansible-playbook test-dac-lustre.yml -i test-inventory-lustre --tag stop_all,unmount,client_unmount
ansible-playbook test-dac-lustre.yml -i test-inventory-lustre --tag format

ansible-playbook test-dac-lustre.yml -i test-inventory-lustre --tag stop_mgs
ansible-playbook test-dac-lustre.yml -i test-inventory-lustre --tag reformat_mgs
```

*test-inventory-lustre

```
dac:
  children:
    fs1:
      hosts:
        dac1:
          fs1_mgs: nvme0n1
          fs1_mdt: nvme1n1
          fs1_osts: {nvme2n1: 2}
        dac2:
          fs1_osts: {nvme3n1: 1}
      vars:
        fs1_mgsnode: dac1
```

*test-dac-lustre.yml

```
---
- name: Setup buffer for fs1
  hosts: fs1
  become: yes
  roles:
    - role: lustre
      vars:
        fs_name: fs1
```

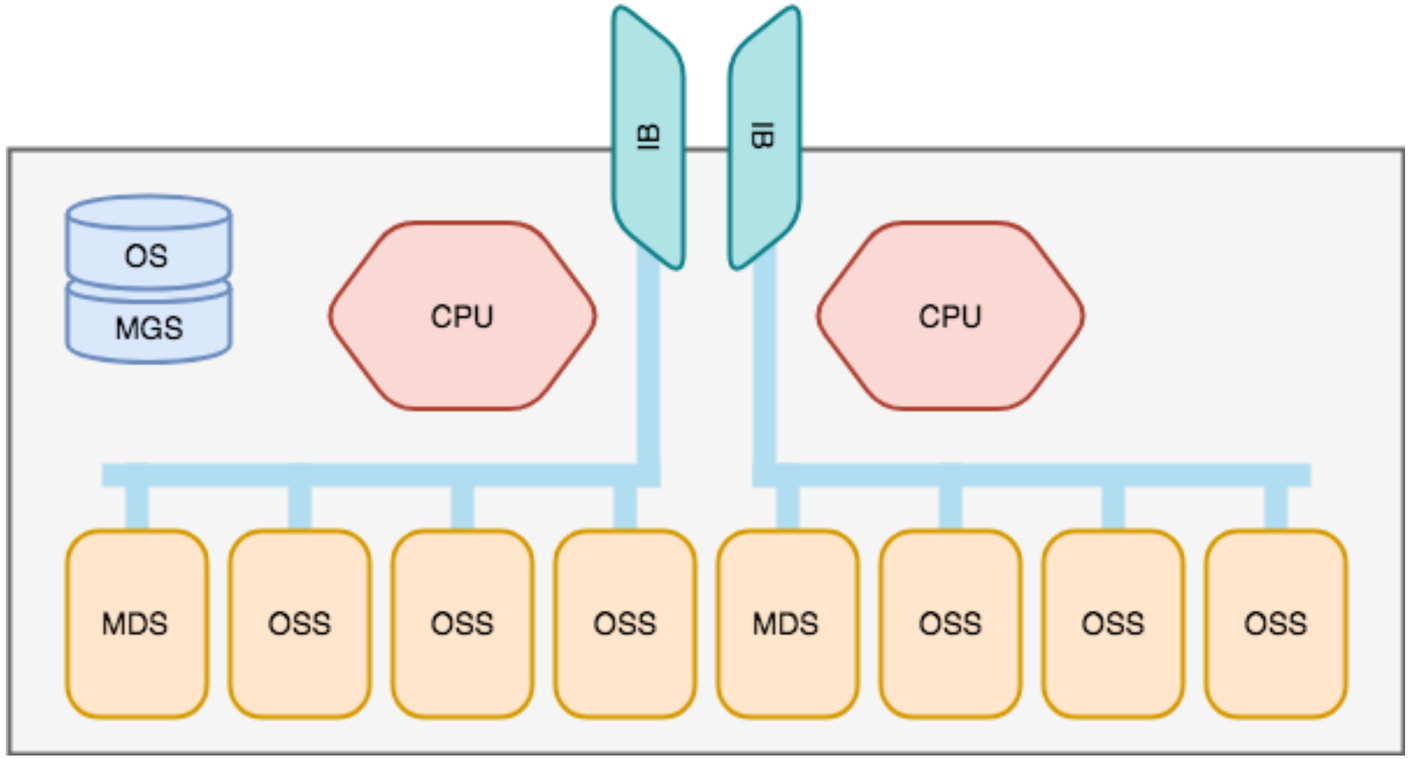
UNIVERSITY OF CAMBRIDGE
Research Computing Services

Ansible Enabled Lustre Install

```
- name: Ensure MGS has been formatted
  command: /usr/sbin/mkfs.lustre --mgs /dev/{{ mgs }}
  register: command_result
  failed_when: "command_result.rc != 0 and ('was previously formatted for lustre' not in command_result.stderr)"
  changed_when: "command_result.rc == 0"
  when:
    - mgs is defined
  tags: [ 'never', 'format_mgs', 'format']

- name: Reformat MGS
  command: /usr/sbin/mkfs.lustre --mgs --reformat /dev/{{ mgs }}
  when:
    - mgs is defined
  tags: [ 'never', 'reformat_mgs']

- name: Reformat MDT
  command: /usr/sbin/mkfs.lustre --mdt --reformat --fsname={{ fs_name }} --index=0 --mgsnode={{ mgsnode }} /dev/{{ mdt }}
  when:
    - mdt is defined
  tags: [ 'never', 'reformat_mdts', 'format']

- name: Reformat OSTs
  command: /usr/sbin/mkfs.lustre --ost --reformat --fsname={{ fs_name }} --index={{ item.value }} --mgsnode={{ mgsnode }} /dev/{{ item.key }}
  loop: "{{ osts|dict2items }}"
  when:
    - osts is defined
  tags: [ 'never', 'reformat_osts', 'format']
```

UNIVERSITY OF CAMBRIDGE
Research Computing Services

Lustre Server Composition



- Each DAC uses an internal SSD for the MGS should it be elected to run a file system.
- NVMeS then have an MDS or OSS applied. This arrangement can be changed as required.

Multirail Lustre

- Set up the ARP and Linux Kernel Routing before enabling multirail

#Setting ARP so it doesn't broadcast
(Do this for every IB interface)

```
sysctl -w net.ipv4.conf.all.rp_filter=0
sysctl -w net.ipv4.conf.ib0.arp_ignore=1
sysctl -w net.ipv4.conf.ib0.arp_filter=0
sysctl -w net.ipv4.conf.ib0.arp_announce=2
sysctl -w net.ipv4.conf.ib0.rp_filter=0
```

Multirail Lustre

- Set up the ARP and Linux Kernel Routing before enabling multirail

```
ip neigh flush dev ib0
ip neigh flush dev ib1

echo 200 ib0 >> /etc/iproute2/rt_tables
echo 201 ib1 >> /etc/iproute2/rt_tables

ip route add 192.168.0.0/16 dev ib0 proto kernel scope link src 192.168.1.1 table ib0
ip route add 192.168.0.0/16 dev ib1 proto kernel scope link src 192.168.2.1 table ib1

ip rule add from 192.168.1.1 table ib0
ip rule add from 192.168.2.1 table ib1

ip route flush cache
```

Multirail Lustre

- Now set up your yaml or lnet.conf

```
#Example lnet.conf in /etc/modprobe.d

options lnet networks=“o2ib1(ib0,ib1)"


#Example command to create yaml conf

lnetctl net add --net o2ib --if ib0,ib1
```

System Performance

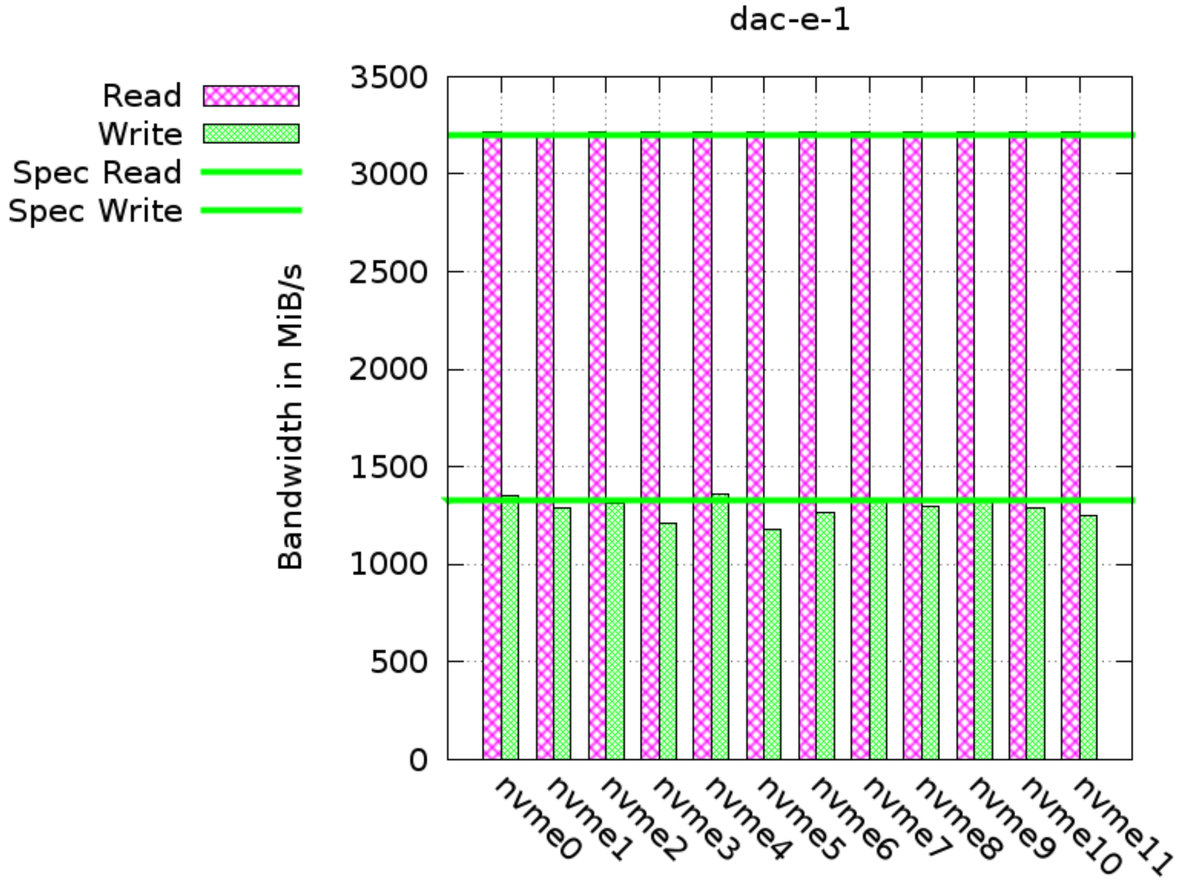
UNIVERSITY OF
CAMBRIDGE
Research Computing Services

Intel P4600 Performance

Single FIO performance tests

```
fio --name=nvme${i} --readwrite=write –bs=128k
--invalidate=1 --group_reporting --direct=1 –time_based
--ramp_time=300 --runtime=300 –ioengine=libaio –numjobs=32
--iodepth=256 –norandommap --randrepeat=0 –stonewall
--size=100% --exitall --filename=/dev/nvme${i}n1p1

fio --name=nvme${i} --readwrite=randwrite –bs=4k
–invalidate=1 --group_reporting --direct=1 –time_based
--ramp_time=300 --runtime=300 --ioengine=libaio –numjobs=32
--iodepth=256 --norandommap –randrepeat=0 –stonewall
--size=100% --exitall --filename=/dev/nvme${i}n1p1
```

***for brevity similar settings for read and randread**



UNIVERSITY OF CAMBRIDGE
Research Computing Services

Early Lustre Performance Results

Single Rail IOR Performance

IOR READ 264 GiB/s
IOR WRITE 225 GiB/s

DAC has 74x better efficiency of performance per TB than our scratch lustre!

To achieve the same performance we would be spending in the region of 10x more on traditional storage.

MultiRail Results should show over 250GiB/s and 600GiB/s for write and read respectively.

UNIVERSITY OF CAMBRIDGE
Research Computing Services

Network Performance

LNET_selftest Lustre 2.11.0 Clients Write

```
[LNet Bandwidth of servers]
[R] Avg: 22050.16 MiB/s Min: 22050.16 MiB/s Max: 22050.16 MiB/s
[W] Avg: 3.36     MiB/s Min: 3.36     MiB/s Max: 3.36     MiB/s
[LNet Rates of servers]
[R] Avg: 42669    RPC/s Min: 42669    RPC/s Max: 42669    RPC/s
[W] Avg: 42670    RPC/s Min: 42670    RPC/s Max: 42670    RPC/s
```

LNET_selftest Lustre 2.11.0 Clients Read

```
[LNet Bandwidth of servers]
[R] Avg: 3.62     MiB/s Min: 3.62     MiB/s Max: 3.62     MiB/s
[W] Avg: 23693.94 MiB/s Min: 23693.94 MiB/s Max: 23693.94 MiB/s
[LNet Rates of servers]
[R] Avg: 23697    RPC/s Min: 23697    RPC/s Max: 23697    RPC/s
[W] Avg: 47388    RPC/s Min: 47388    RPC/s Max: 47388    RPC/s
```

LNET_selftest.sh

```
#!/bin/bash
export LST_SESSION=$$
lst new_session read_write
lst add_group servers 10.47.18.1@o2ib1
lst add_group clients 10.47.0.[1-8]@o2ib1
lst add_batch bulk_rw
lst add_test --batch bulk_rw --concurrency 12 --from clients --to servers \
brw read check=simple size=1M
# start running
lst run bulk_rw
# display server stats for 30 seconds
lst stat servers & sleep 30; kill $!
# tear down
lst end_session
```

UNIVERSITY OF CAMBRIDGE
Research Computing Services

Disk Performance

obdfilter test 8NVMe Lustre 2.11.0

```
ost  8 sz 1702887424K rsz 1024K obj 4096 thr 8192
write 10670.58 [1237.28, 1529.71]
rewrite 10575.61 [1153.67, 1542.69]
read 17581.66 [2178.53, 3084.63]
```

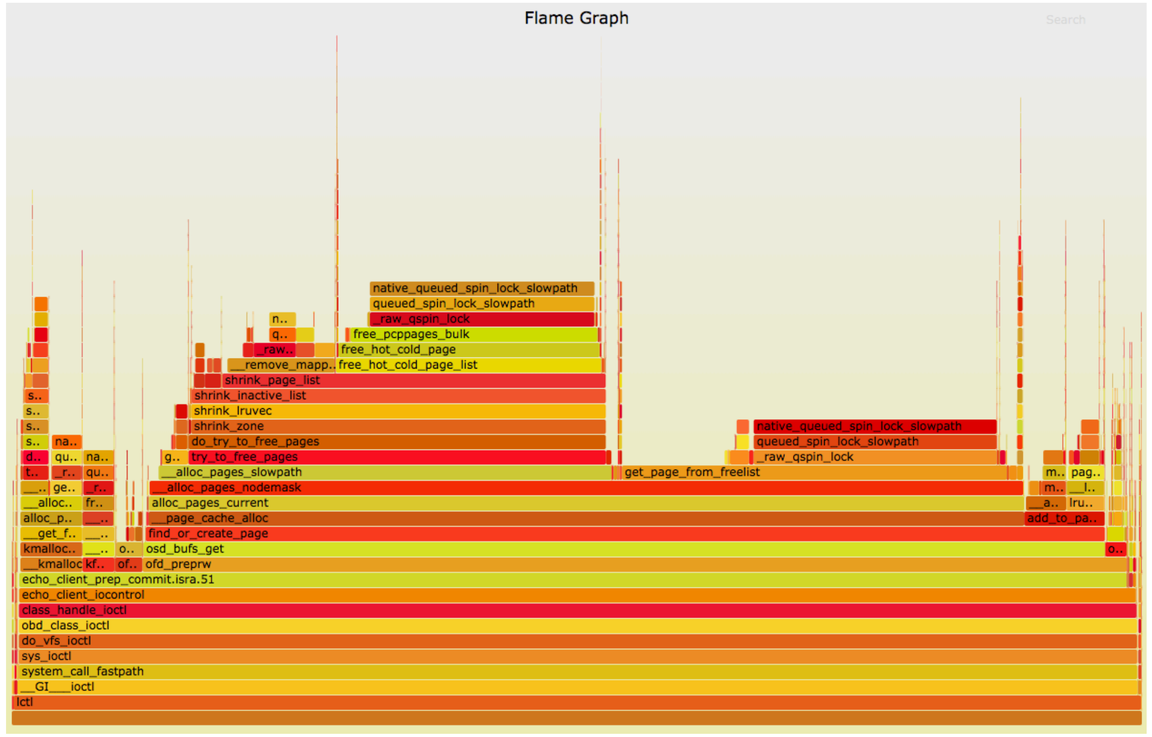
```
ost  8 sz 838860800K rsz 1024K obj   64 thr  512
write 10671.99 [1276.94, 1534.92]
rewrite 10692.21 [1268.93, 1542.90]
read 17020.97 [1485.89, 2460.10]
```

OSS not translating read performance correctly requires further investigation

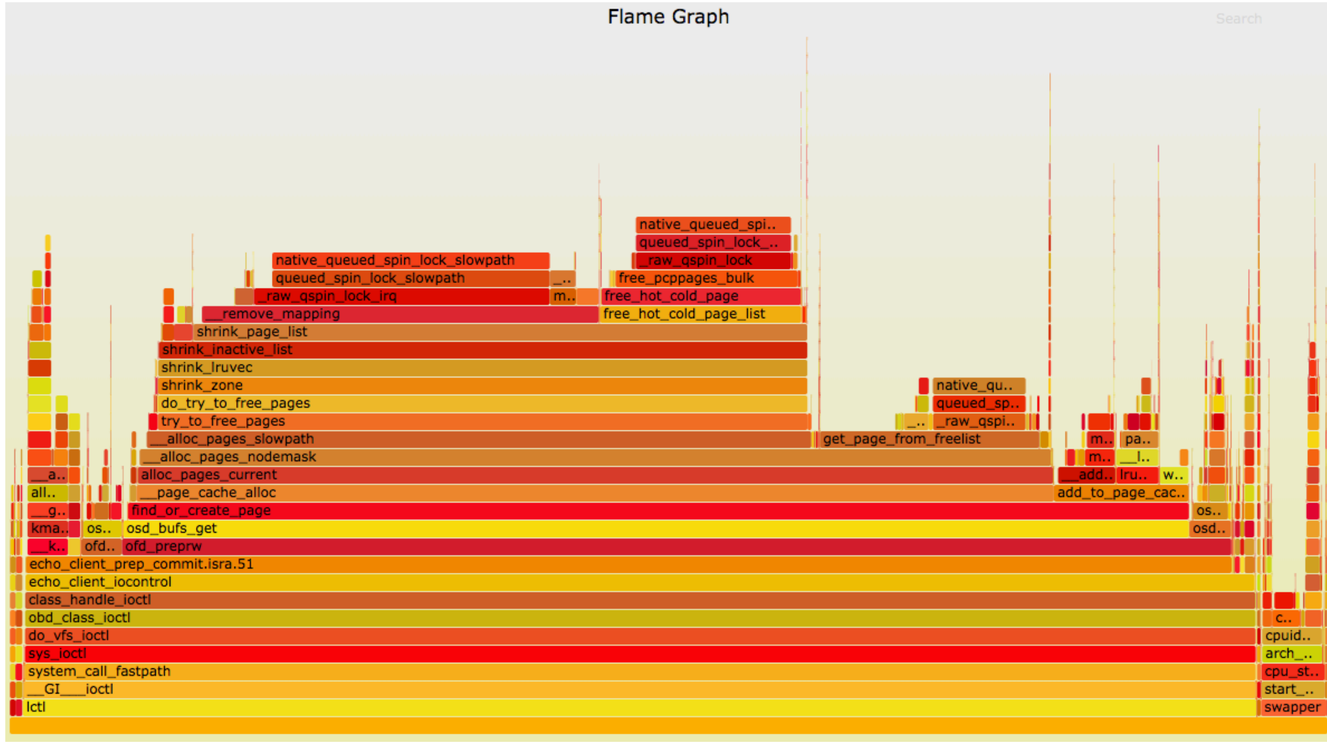
Can be made to reach ~19GiB/s but this is still lower than it should be (22-23GiB/s)

UNIVERSITY OF CAMBRIDGE
Research Computing Services

Disk Performance



obdfilter result where threads=8192

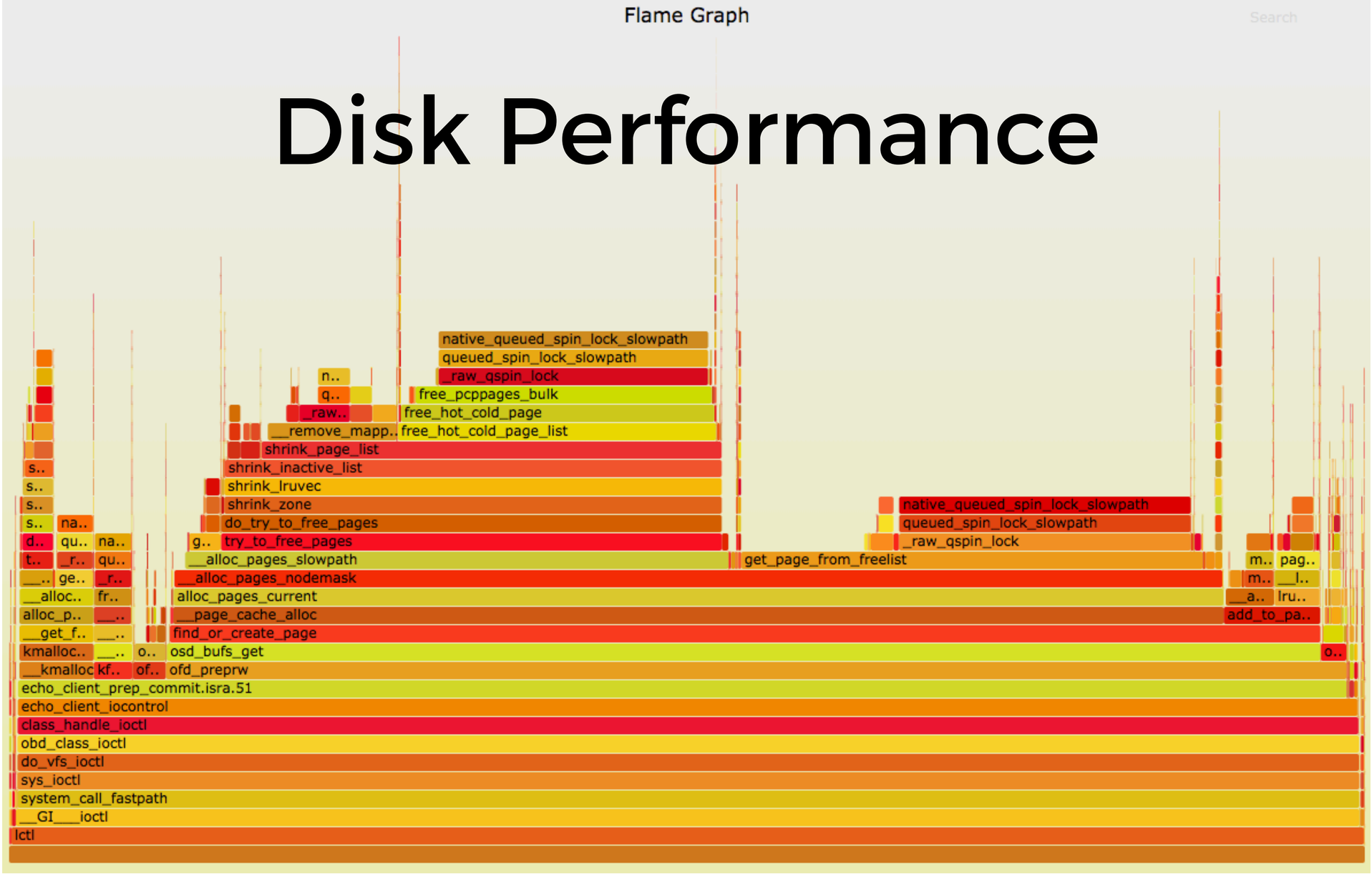


obdfilter result where threads=512

Flame Graphs show the OSS is stuck in in memory calls.
/proc/buddyinfo shows the memory fragmentation is high
This requires further investigation and submitted to jira.

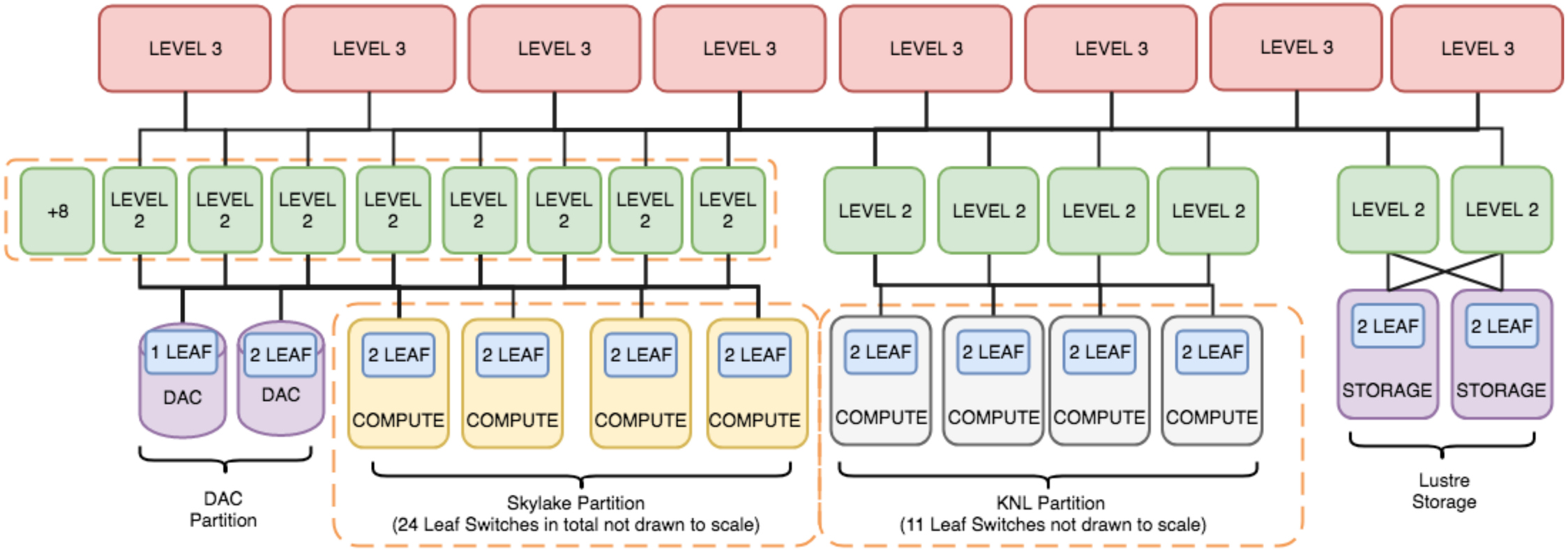
UNIVERSITY OF CAMBRIDGE
Research Computing Services

Disk Performance



UNIVERSITY OF CAMBRIDGE
Research Computing Services

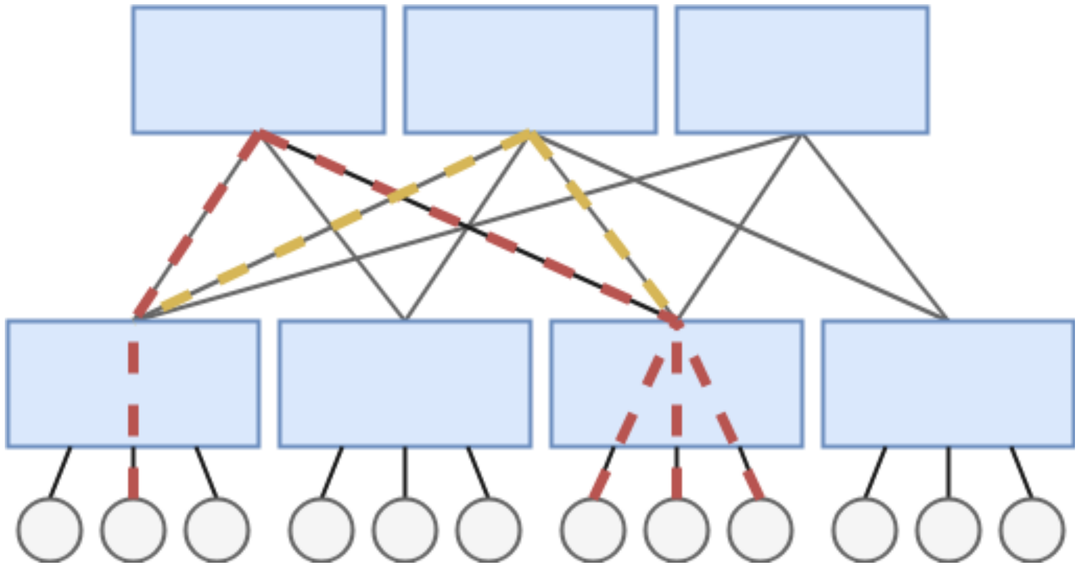
Peta 4 Interconnect Topology



UNIVERSITY OF CAMBRIDGE
Research Computing Services

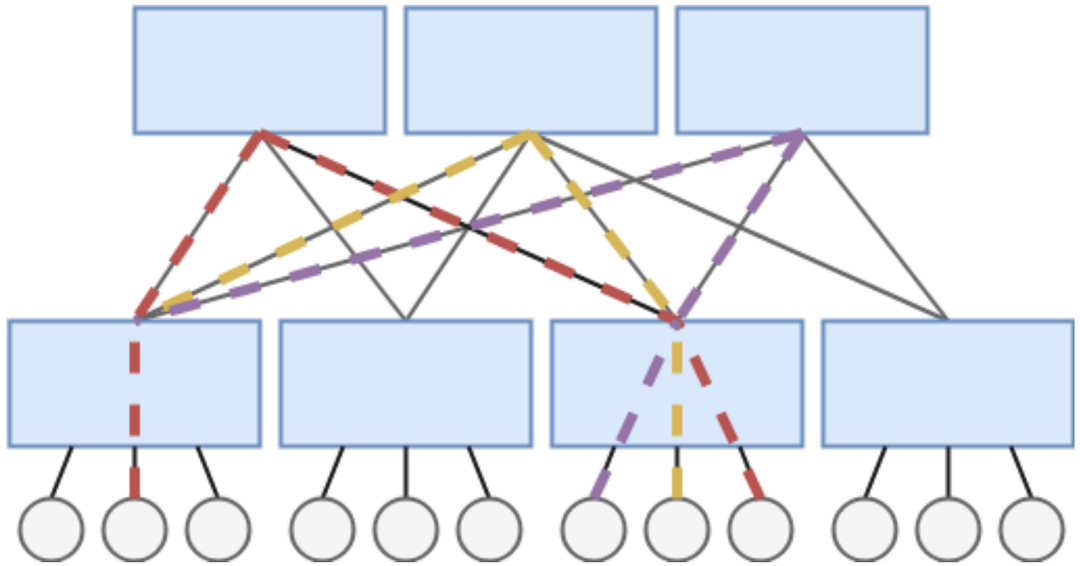
* **Wilkes II (Not shown)**
Connects via LNET routers 4 in each storage rack

Fat Tree Static Routing



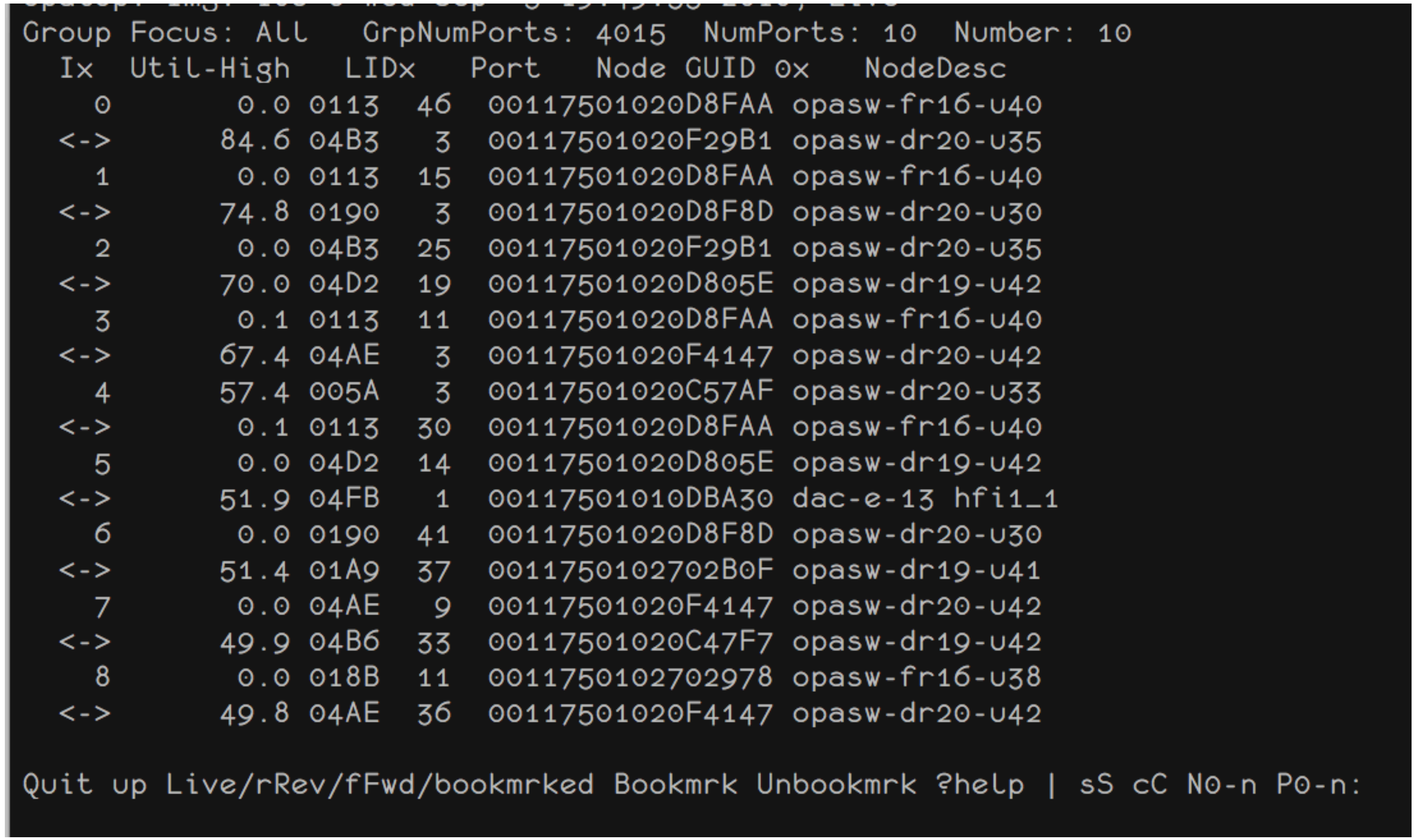
- All nodes take the same Inter Switch Links(Red)
- Other Links are Posible(Gold)

Adaptive Routing



- Nodes can now take alternate routes (Gold,Purple)
- Utilisation of Inter switch links improved

Diagnosing in Intel Omni-Path



- Example of *opatop* during a test. Can highlight oversubscribed links based on the percentage utilised.

UNIVERSITY OF CAMBRIDGE
Research Computing Services

Intel Omni-Path Tuning

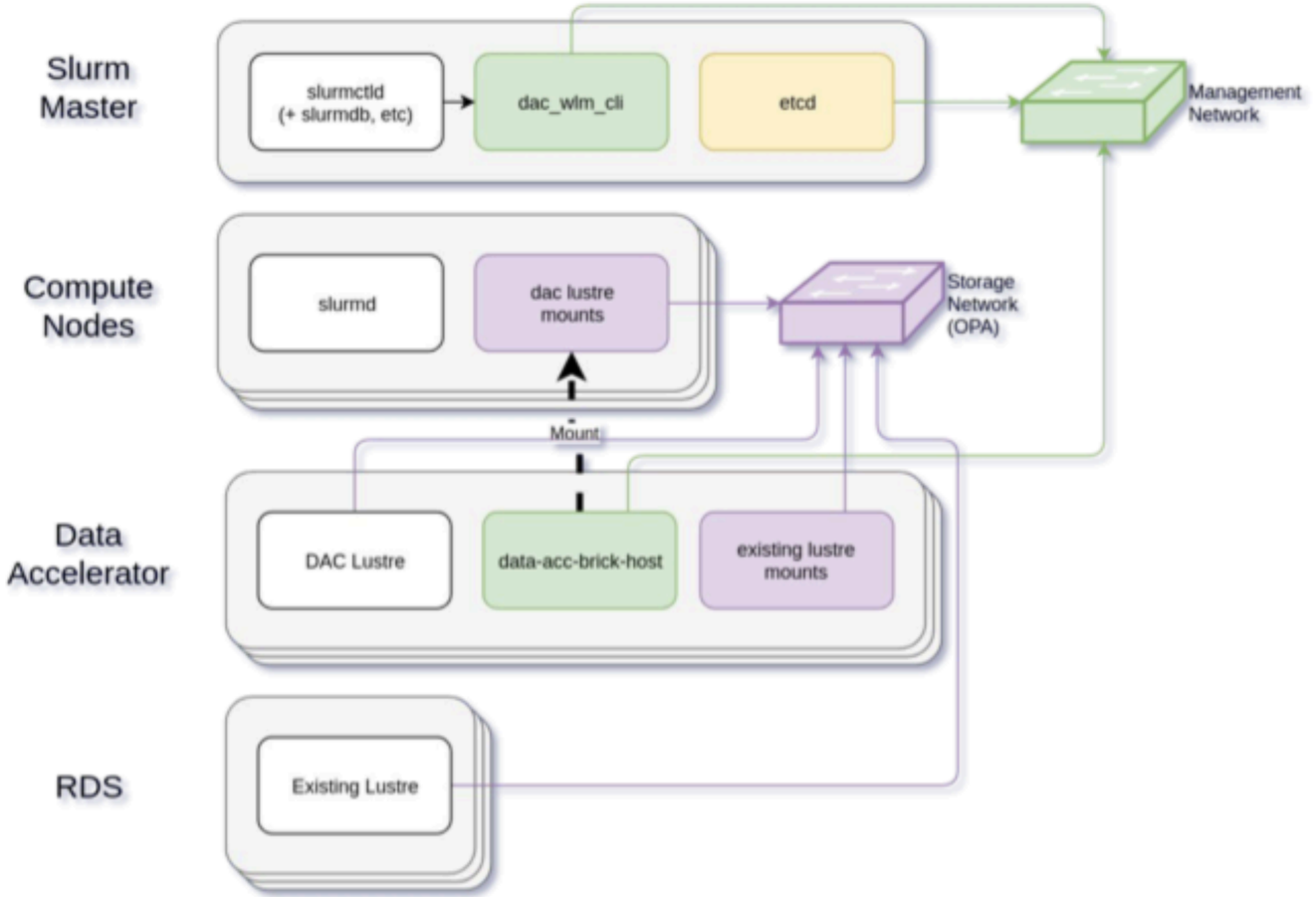
- /etc/modprob.d/hfi1.conf
- krcvqs=8, pcie_caps=0x51,rcvhdrcnt=8192
- *krcvqs can be varied based on node role

Slurm Integration

UNIVERSITY OF
CAMBRIDGE
Research Computing Services

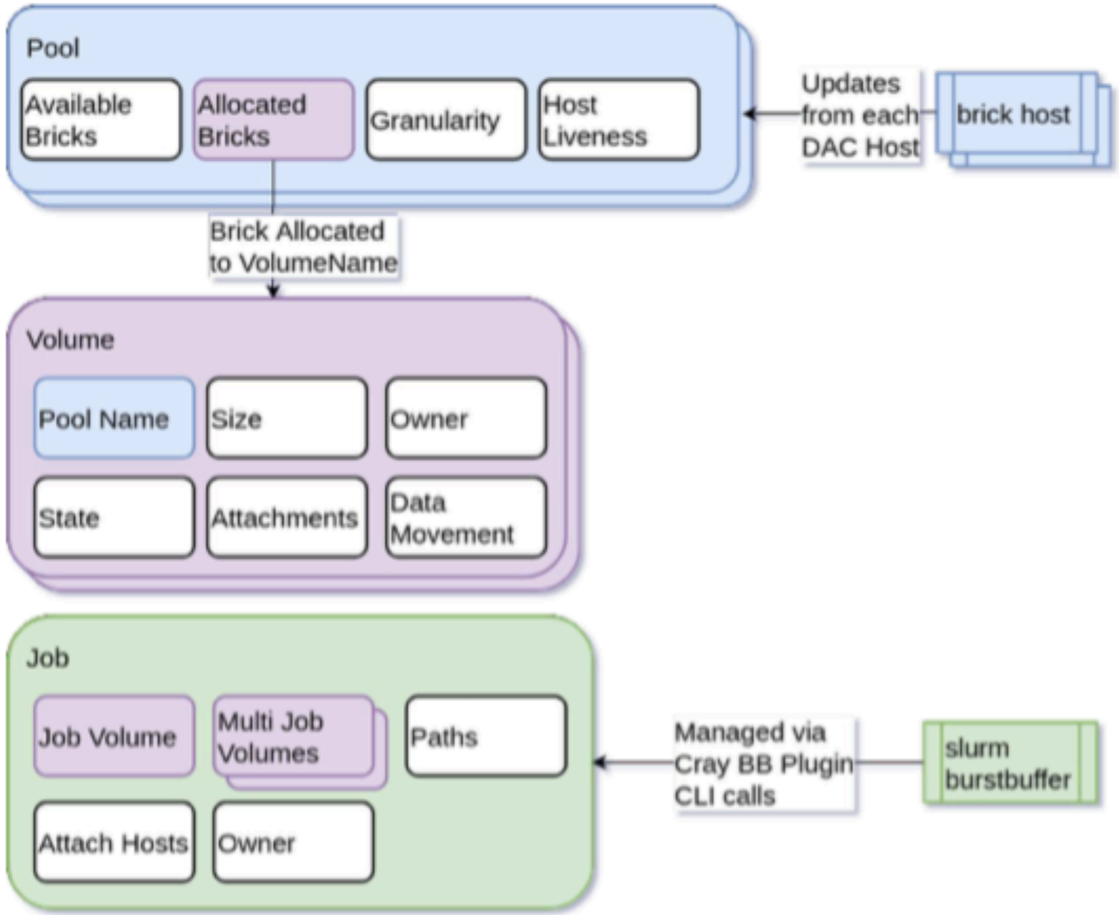
DAC Plugin

- Reuses Existing Cray plugin.
- Cambridge has implemented a orchestrator to manage the DAC nodes.
- GO-Lang project utilising ETCd and Ansible for dynamic automation
- To be released as an OpenSource project.

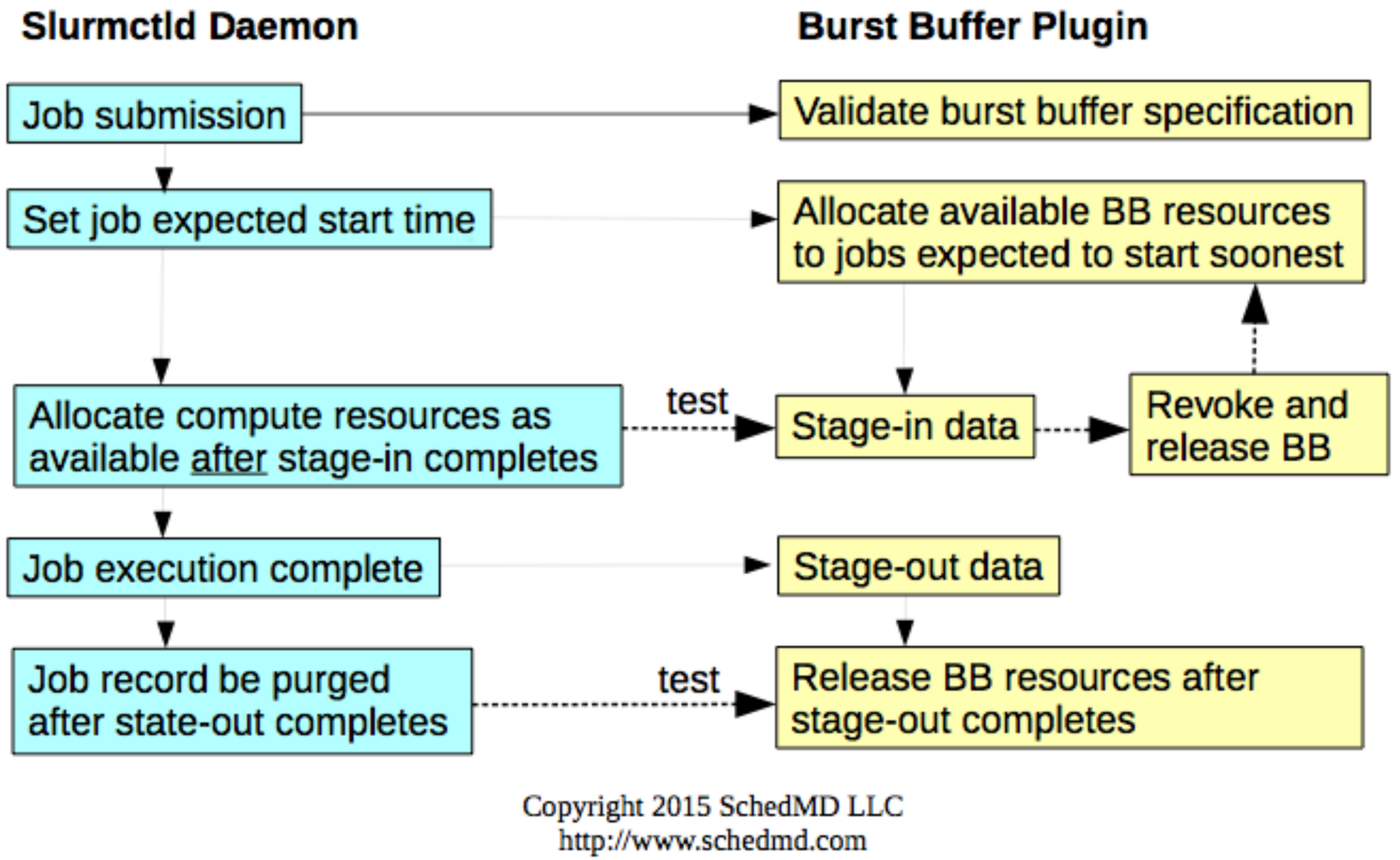


UNIVERSITY OF CAMBRIDGE
Research Computing Services

SLURM DAC Plugin



SLURM DAC Plugin



UNIVERSITY OF CAMBRIDGE
Research Computing Services

From https://slurm.schedmd.com/burst_buffer.html

SLURM DAC Plugin

```
#!/bin/bash
#BB create_persistent name=alpha capacity=32GB access=striped type=scratch
#DW jobdw type=scratch capacity=1GB access_mode=striped
#DW stage_in  type=file source=/home/alan/data.in  destination=$DW_JOB_STRIPED/data
#DW stage_out type=file destination=/home/alan/data.out source=$DW_JOB_STRIPED/data
/home/alan/a.out
```

User specified burst buffer options within a slurm batch file.
Example from https://slurm.schedmd.com/burst_buffer.html

Special Thanks to...

- RCS Platforms Team
- whamcloud.io
- Intel Omni-Path Team
- Intel SSD Division
- Dell EMC Inovation Lab

UNIVERSITY OF CAMBRIDGE
Research Computing Services

Questions and Comments?

UNIVERSITY OF CAMBRIDGE
Research Computing Services

Alasdair King ajk203@cam.ac.uk

Thanks for the Continued Support of :