Australian Characterisation Commons at Scale

Work Package 4: Big-Data Electron and Correlative Microscopy from Instrument to Publication

Deploying a national EM Data processing portal

Jay van Schyndel – Monash eResearch Centre

Australian Research Data Commons

This project is supported by the Australian Research Data Commons (ARDC) and the following partners. The ARDC is enabled by NCRIS.

The Plan

HPCasCode ✓

Strudelv2 ✓

CryoSPARC ✓

SLURM ✓

Minimal administration ?

LiberTEM ✓

Globus ✓

Deploying the Clusters

- https://github.com/Characterisation-Virtual-Laboratory/EM-Data-Processing-Portal
- 2 NecTAR locations – QRISCloud and Monash-02
- At each location:
  - 1x bastion node (admin access to the cluster)
  - 2x login nodes
  - 1x globus node
  - 12 NVIDIA A100 40 GB GPUS (12x compute nodes at QCIF, 6x compute nodes at Monash-02)
  - 1x SQL node (running the SLURM database)
  - 1x NFS node (Network File System server)
- Deployed using the HPCasCode repository Ansible scripts
- SLURM 21.08 to support NVIDIA MIG (multi-instance GPU)
- Both cluster deployments are the same, except for the compute nodes

Running CryoSPARC on a HPC cluster

CryoSPARC experience on MASSIVE

- 4 instances on MASSIVE. Each instance has many user accounts.
- Challenges:
  - CryoSPARC has its own user management system that does not interact with the HPC user management system.
  - CryoSPARC runs under its own user account, that user owns all the data. This make data management in a muti-user environment difficult.

- Solve data management challenges.
- Goal: reduced administration effort for this National Service.
  - Can CryoSPARC be installed seamlessly for an end user with no system administration help ?

Running CryoSPARC on a HPC cluster

Solution to data management:

- Each user runs their own CryoSPARC server that submits jobs to the HPC cluster.
- data access and ownership issues solved
- recommended approach for a HPC environment by CryoSPARC.
- each user supplies their own licence to run CryoSPARC.

Can it be done in practice ?

- configure cryosparc_worker to submit jobs to a SLURM cluster (standard CryoSPARC functionality)
- perform the CryoSPARC installation as a SLURM job
- run cryosparc_master as a SLURM job

Running CryoSPARC on a HPC cluster

Can it be done in practice ?

- configure cryosparc_worker to submit jobs to a SLURM cluster (standard CryoSPARC functionality) ✓
- perform the CryoSPARC installation as a SLURM job ✓
  - portChecker
  - install_cryosparc.sh (downloads using supplied licence, install master, creates user account, installs worker, configures worker for slurm job submission, stops cryosparc)
- Run cryosparc_master as a SLURM job ✓
  - portChecker
  - start_cryosparc.sh (updates hostname and port, starts master, connect worker to master)

Testing CryoSPARC

- Using newly developed scripts, performed a clean installation and start CryoSPARC.
- Queue the CryoSPARC job: T20s Extensive Workflow
  - automatically downloads a dataset
  - runs through many of the processing steps
  - good test for installation and setup

- On QRIScloud, T20s Extensive Workflow failed due to a timeout between the main and secondary jobs.
- Workaround: update the python code to increase the timeout from 120 to 360 seconds.
  - not a viable solution.
- Repeat the same test on the Monash-02 cluster, it ran successfully, however slow.
- Both clusters slow at installation and running T20S workflow compared to MASSIVE.

Deploying Globus

- Data movement tool for the service
- Ansible scripts previously developed under the ACCS WP4 project:

https://github.com/Characterisation-Virtual-Laboratory/Globus-Endpoint-deployment

  - recent updates to Globus v5, prevent their use, however still a good reference.
- Manually installed on both clusters.
- Collections known as:
  - EM Data Processing Portal at QCIF
  - EM Data Processing Portal at MeRC
- Installation was straight forward due to prior experience.

- QCIF collection: “The operation timed out”
- MeRC collection: “The operation timed out” intermittently

Troubleshooting Globus

- Globus support – really good.
- daily response to replies over more than 1 week.
- gained in-depth knowledge at troubleshooting Globus.
- conclusion, the Globus installation was fine.

- Where to next ?
- Suspects: storage and/or network performance

Performance Testing

- Recap: CryoSPARC performance slow on both clusters, Globus times out.
- modified install script to gain metrics

Availability zone, storage tested.	Time to install and configure CryoSPARC.
QRISCLOUD volume storage (nfs mounted) on a compute node.	3258 seconds
Monash-02 volume storage (nfs mounted) on a compute node	2176 seconds
MASSIVE - /scratch (Lustre), run on m3p001 node	533 seconds

Performance Testing

Availability zone, storage tested.	Time to install and configure CryoSPARC.
QRISCLOUD volume storage (nfs mounted) on a compute node.	3258 seconds

- NFS and slurm database sharing the same machine – m3.small with 2 CPUs and 4 GB
- Redeployed to m3.xlarge with 16 CPUs and 32 GB
- re-install and check installation time

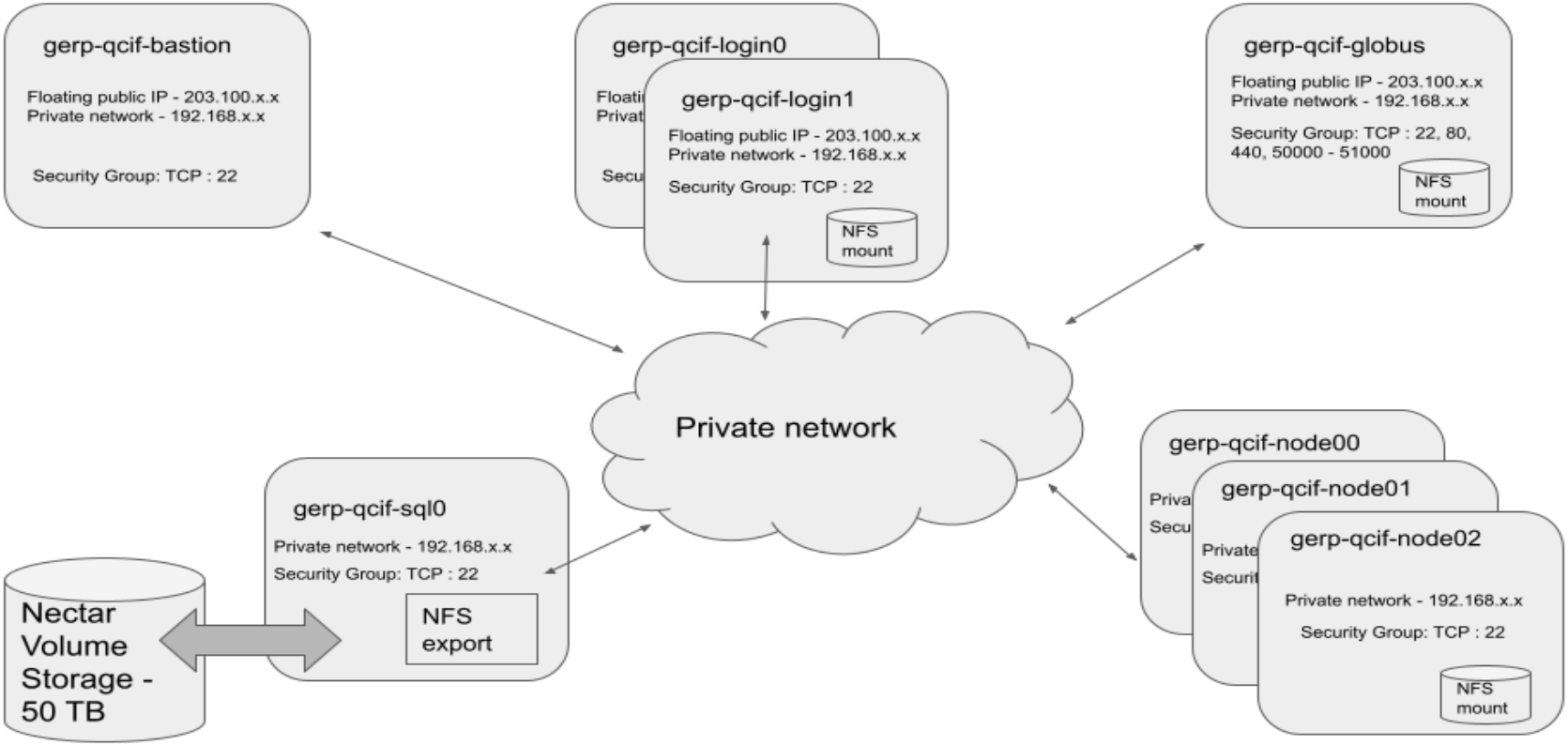
Availability zone, storage tested.	Time to install and configure CryoSPARC.
QRISCLOUD volume storage (nfs mounted) on a compute node.	2942 seconds

Performance Testing – data movement

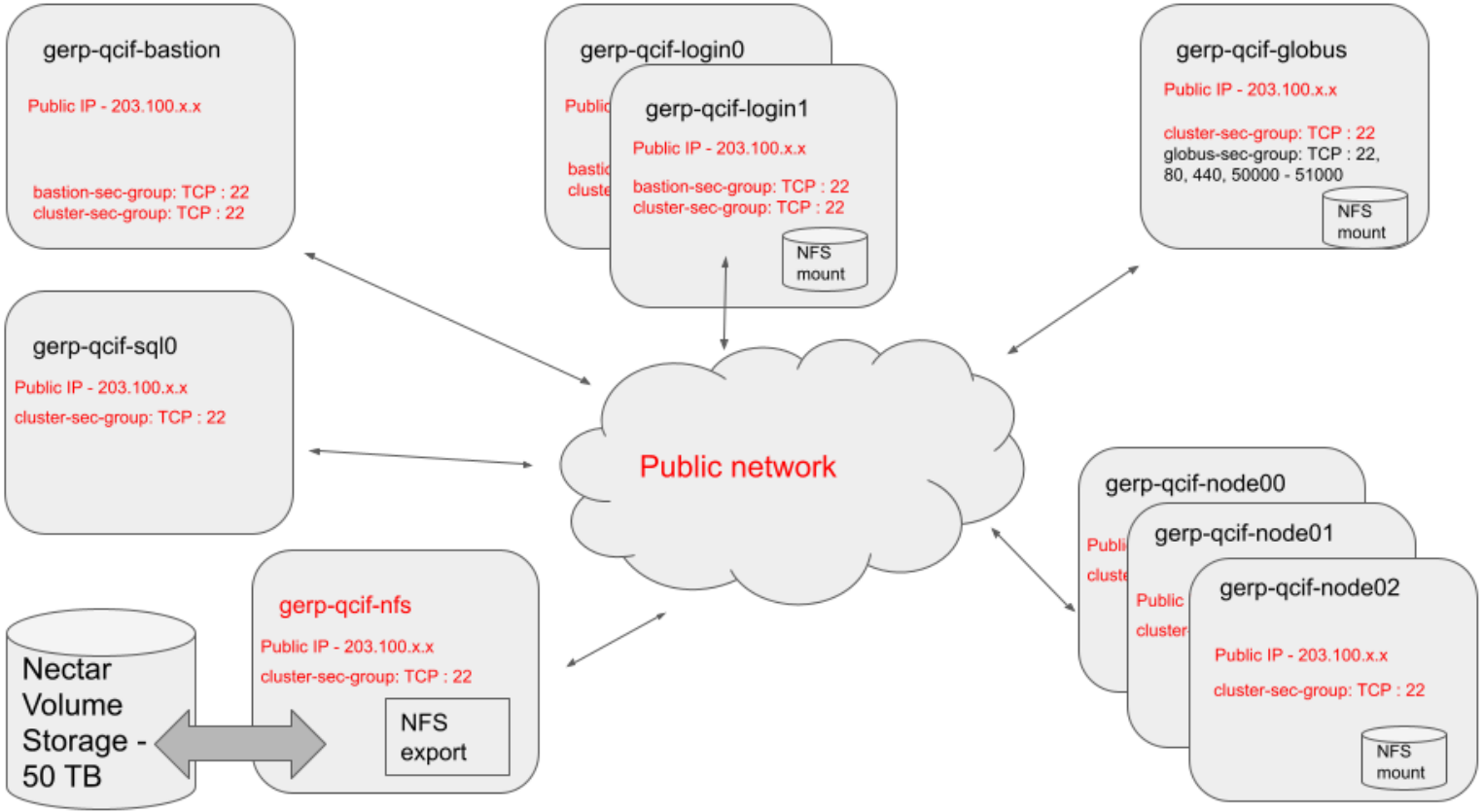
- Several different types of storage available on the cluster
  - local nvme, nectar volume storage, nfs mounted volume storage
- rsync the cryosparc installation, lots of little files, just like cryosparc creates.

Source (machine - storage)	Destination (machine - storage)	Speed	Note:
gerp-qcif-sql0 - mounted volume storage	gerp-qcif-sql0 - mounted volume storage	23,682,455.49 bytes/sec	rysnc on the same machine to the same storage
gerp-qcif-sql0 - mounted volume storage	gerp-qcif-node00 - nvme	21,568,032.47 bytes/sec	rsync to nvme on another machine. A bit slower but using the network between machines.
gerp-qcif-node00 - nvme	gerp-qcif-node00 - nfs mounted volume storage	**3,319,455.77 bytes/sec**	rsync on the same machine from nvme to nfs mounted storage.
gerp-qcif-node0 - nvme	gerp-qcif-node01 - nvme	145,267,517.41 bytes/sec	rsync between compute nodes, nvme to nvme.

Network Configuration



Network Configuration



CryoSPARC and Globus fixed

- CryoSPARC on QRISCloud. Test workflow no longer times out.
- Globus on QRISCloud functional, no timeout received.
  - 500 GB dataset from ‘AARNet Readonly Public Test Share’ to ‘EM Data Processing Portal at QCIF’
  - 317.49 MB/s transfer rate
  - Globus on Monash-02, intermittent timeout fixed.

User Provisioning
Strudel v2

- Users accounts are provisioned using a combination of python, cron jobs and Gitlab runners
- Strudel v2 is used to provide web access to the service
- https://github.com/Characterisation-Virtual-Laboratory/EM-Data-Processing-Portal/blob/main/USER_PROVISIONING.md

- https://gerp.rc.edu.au/ - GPU eResearch Platform
  - EM Data Processing Portal – QCIF
  - EM Data Processing Portal - MeRC

Compute Resources

```
12 x qld.64c600g.A100.nvme
  - 64 CPUs
  - 600 GB RAM
  - 1 x A100 40 GB, with MIG - 2 x 10GB, 1 x 20 GB, therefore 3
GPUs
```

- Effectively 36 GPUs at QRISCLOUD.
- There are enough compute resources to run a total of 36 CryoSPARC installations simultaneously
- similar for Monash-02, effectively 36 GPUs.

Storage Challenges

- 50 TB at QRISCloud
- 50 TB at Monash-02
- Hardware to run 36 CryoSPARC instances at each cluster
- CryoEM datasets are large, multi-TB.
- Each user provisioned with 5 TB total, 10 users per cluster.

Acknowledge:

Andreas Hamacher, Kiowa Scott-Hurley: HPCasCode

Chris Hines: Strudel v2, user provisioning

Swe Aung: Nectar advanced networking

Michael Mallon: QCIF, storage performance investigation

References:

https://github.com/Characterisation-Virtual-Laboratory/EM-Data-Processing-Portal

- HPCasCode: https://gitlab.erc.monash.edu.au/hpc-team/HPCasCode
- CryoSPARC: https://cryosparc.com
- SLURM: https://slurm.schedmd.com
- LiberTEM https://libertem.github.io/LiberTEM/install.html
- Globus https://www.globus.org

- ACCS Imaging Tools: https://imagingtools.au