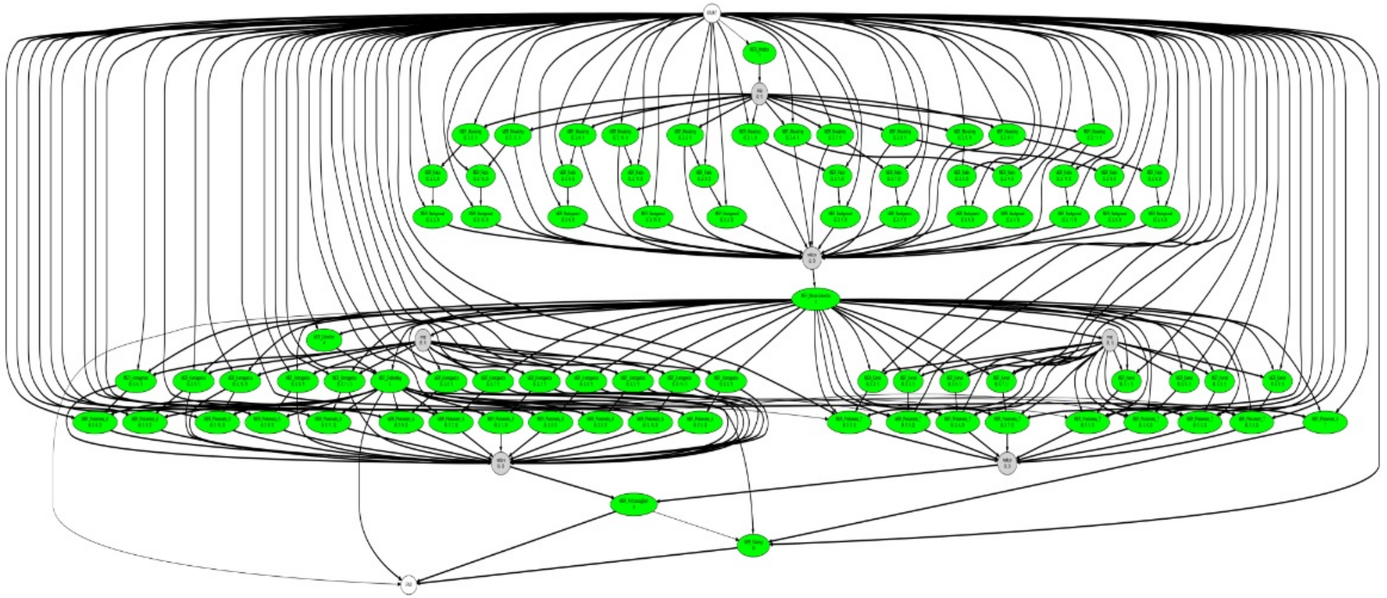
Euclid on IRIS - Show and Tell

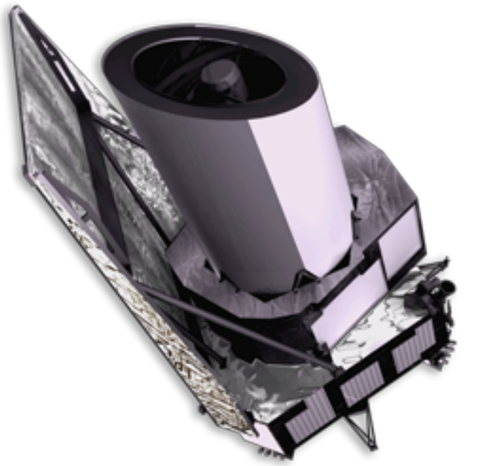
Stig Telfer, StackHPC Ltd

22nd October 2019

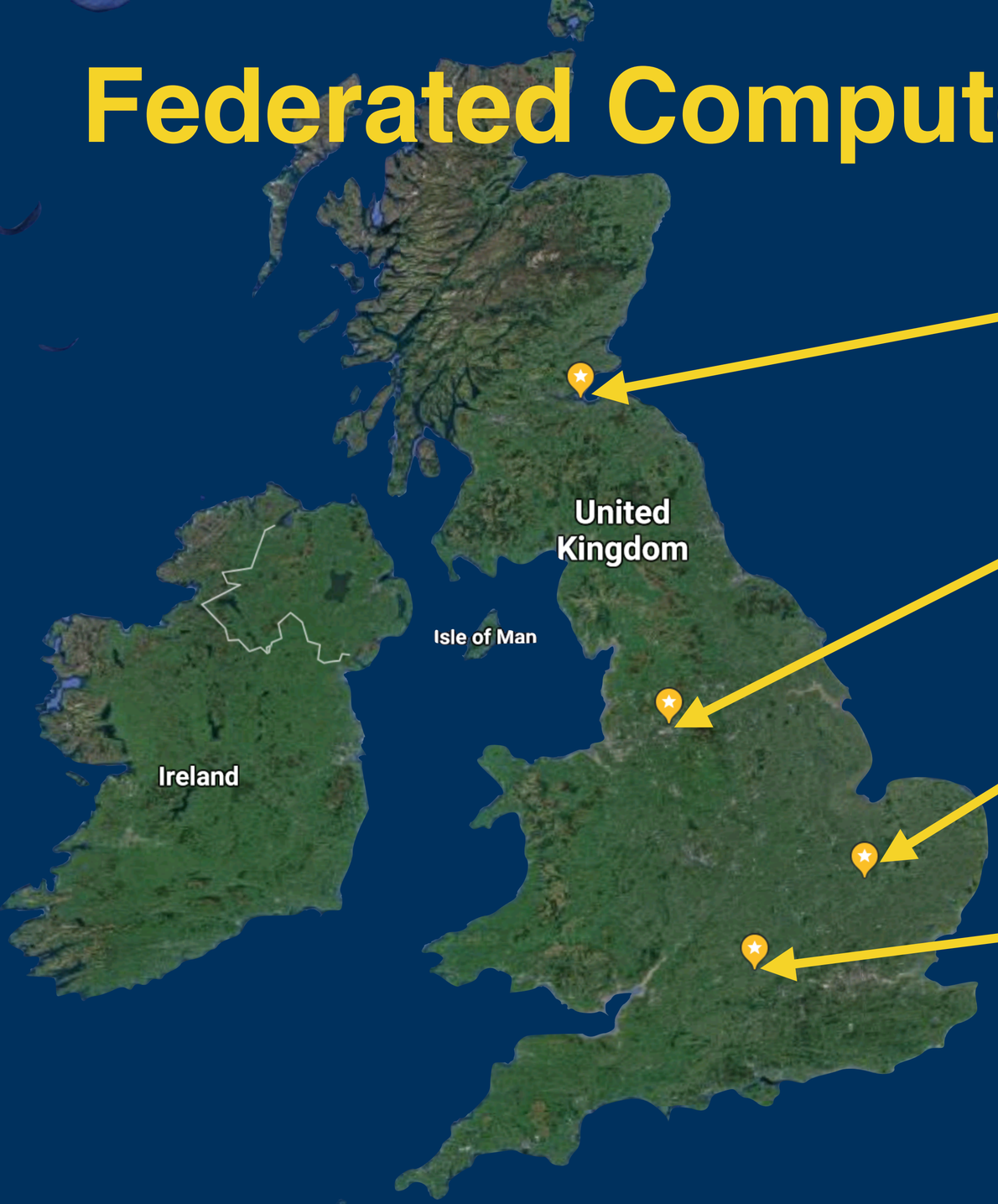
Euclid's Compute Requirements



- First IRIS runs summer 2018
- Data-flow application model
- Uses cluster filesystem
- Simulation run can take 150,000 core hours
- IRIS resource reservation at multiple sites
- Limited options for cluster filesystem



Federated Compute Platform



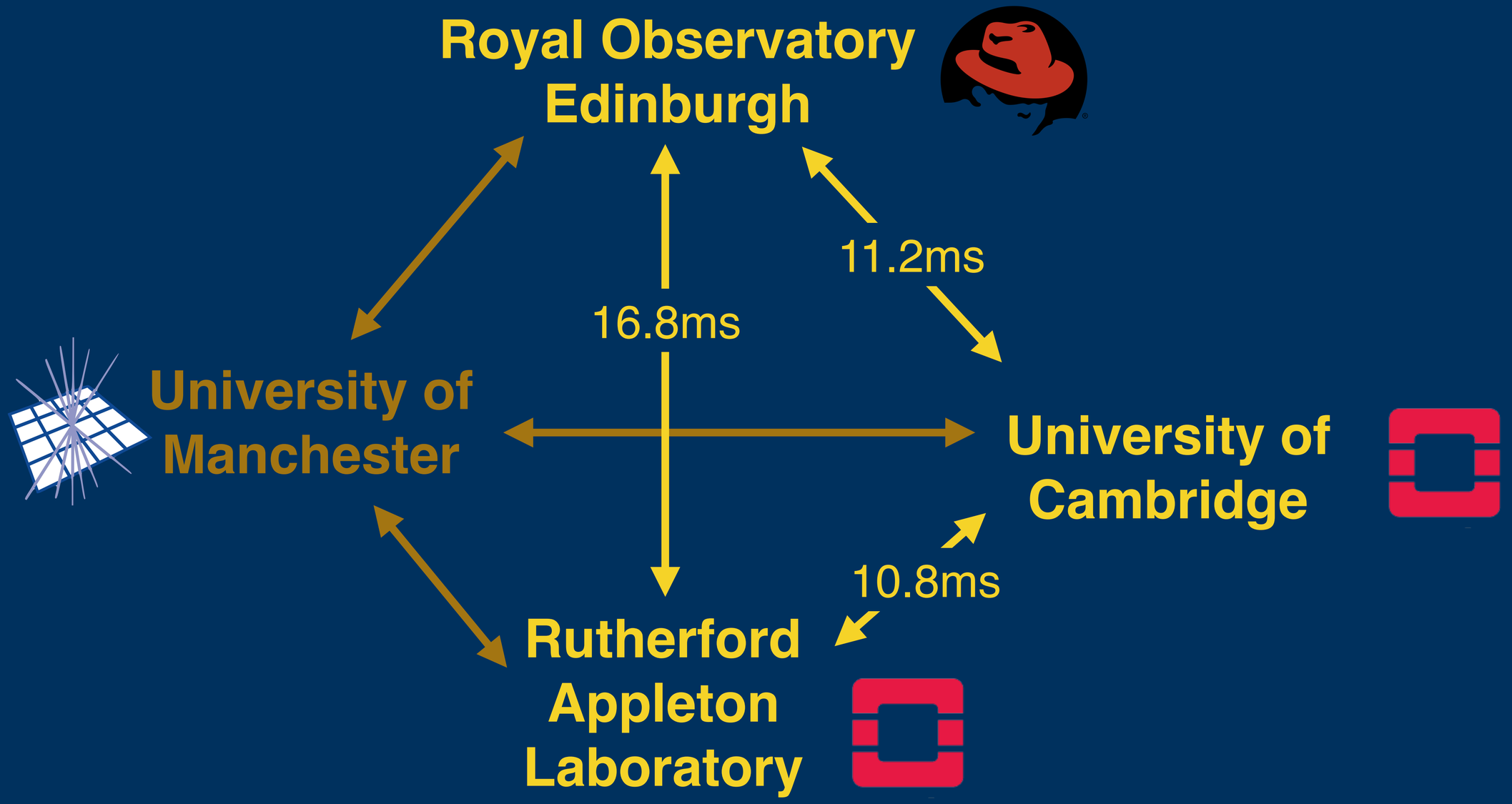
Royal Observatory, Edinburgh
2 VMs plus long-term storage

University of Manchester
Work in progress

University of Cambridge
39 VMs, 1026 vCPUs, 5.8TB

Rutherford Appleton Laboratory
190 VMs, 1728 vCPUs, 29.3TB

Federated Compute Platform

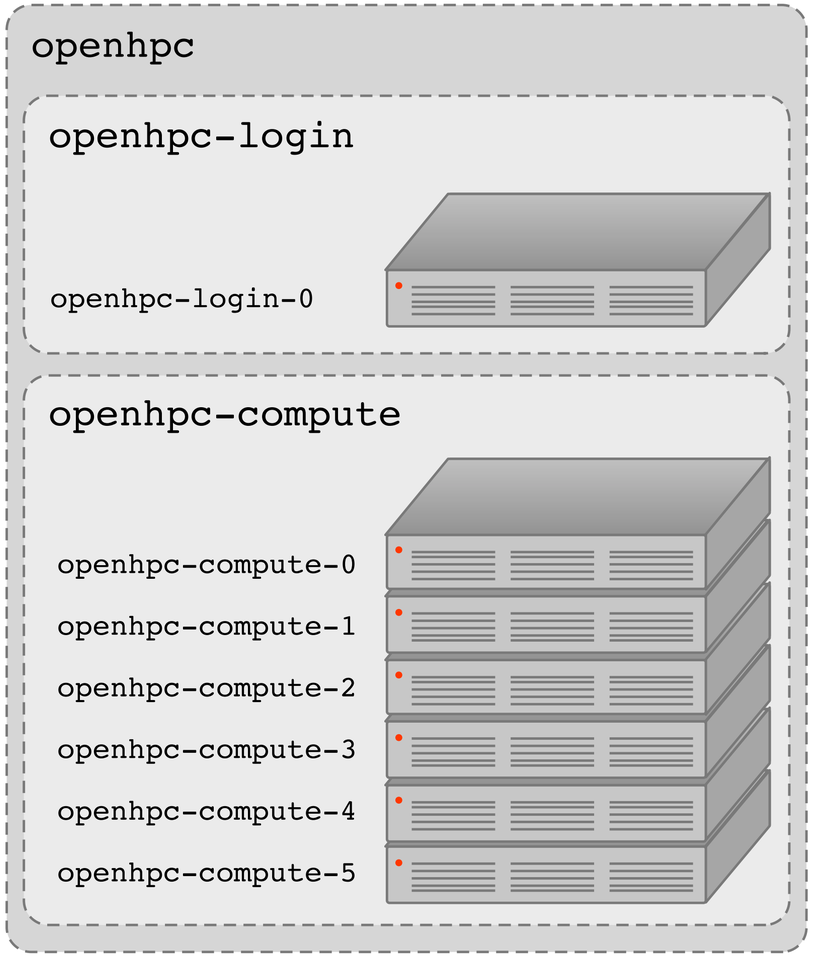


Software-Defined Infrastructure

- We define a cluster as a number of groups of nodes.
- We tell OpenStack to create a number of nodes in each group.
- The definitions are written in Ansible YAML data.
- After authentication with AAI services, Ansible uses the OpenStack Heat APIs to create all the nodes.
- An inventory is produced of everything that was created.

Slurm Infrastructure using Heat & Ansible

```
cluster_name: openhpc

cluster_groups:
  - "{{ slurm_login }}"
  - "{{ slurm_compute }}”

slurm_login:
  name: "login"
  flavor: "m2.medium"
  image: "CentOS7.5-OpenHPC"
  user: “centos"
  num_nodes: 1

slurm_compute:
  name: "compute"
  flavor: "m2.large"
  image: "CentOS7.5-OpenHPC"
  user: "centos"
  num_nodes: 6
```



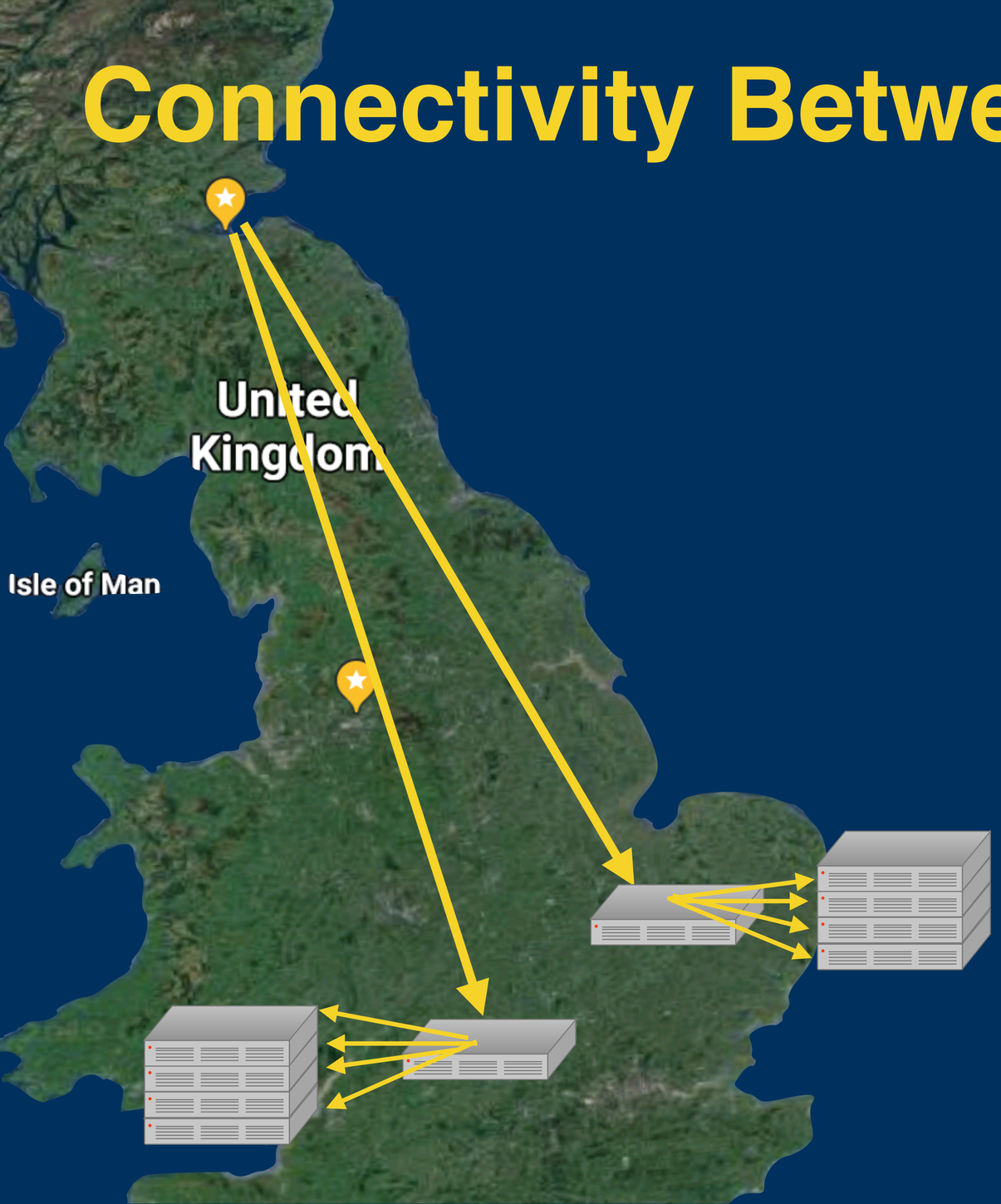
StackHPC

https://galaxy.ansible.com/stackhpc/cluster-infra

Multi-Cloud Portability

- Authenticated access to OpenStack APIs
- Compute node attributes at different sites (flavors and images)
- Presentation of storage resources
- Handling of publicly-addressable IP addresses
- Firewalling and Security, SELinux

Connectivity Between Sites



All Configuration from Edinburgh
Single checkout of Euclid infra repo

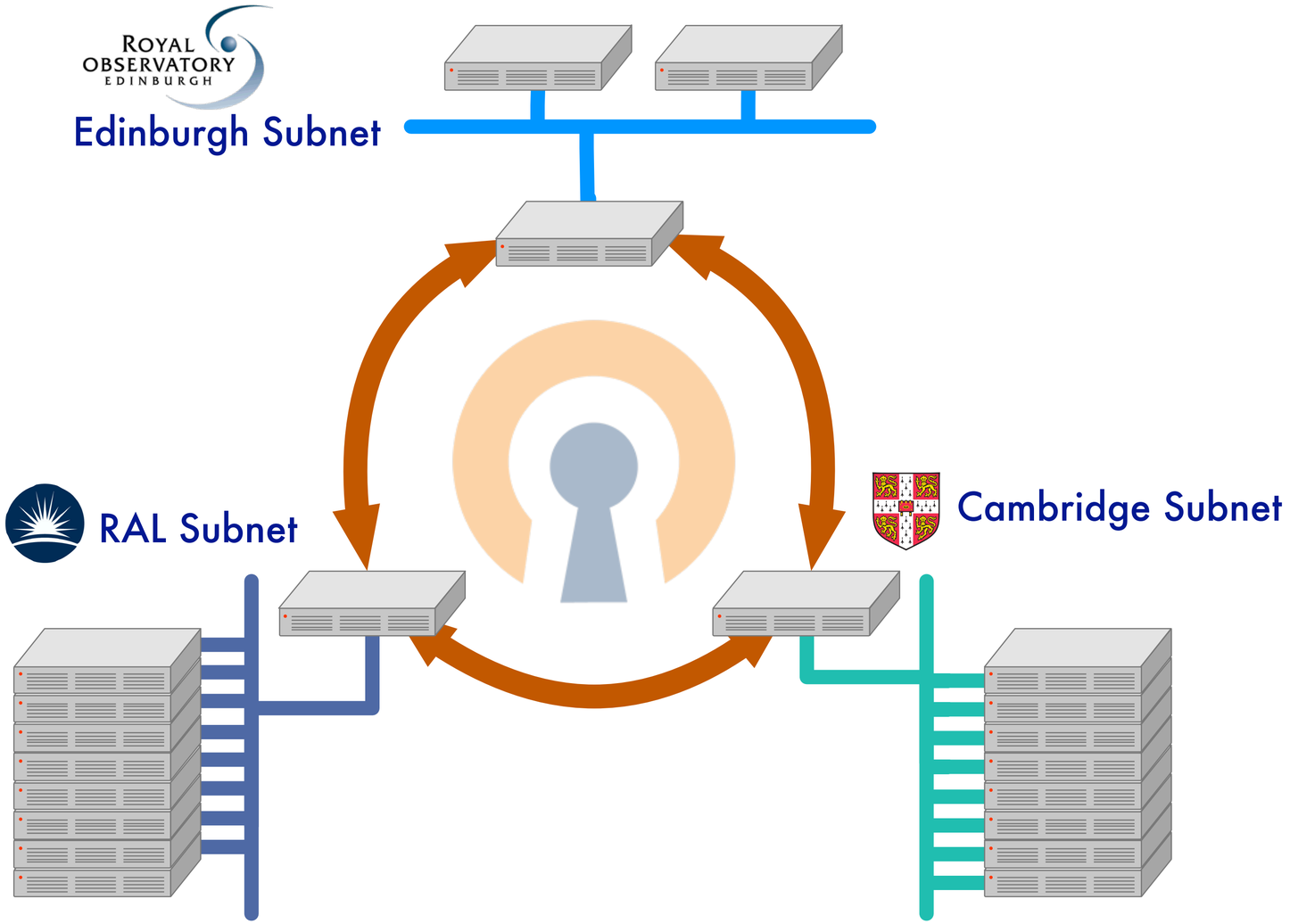
Single Public IP address at each site
Used for admin SSH and OpenVPN

Ansible SSH to public gateway
SSH Proxy Jump to internal resources

Establishment of VPN connectivity
VPN data plane between sites

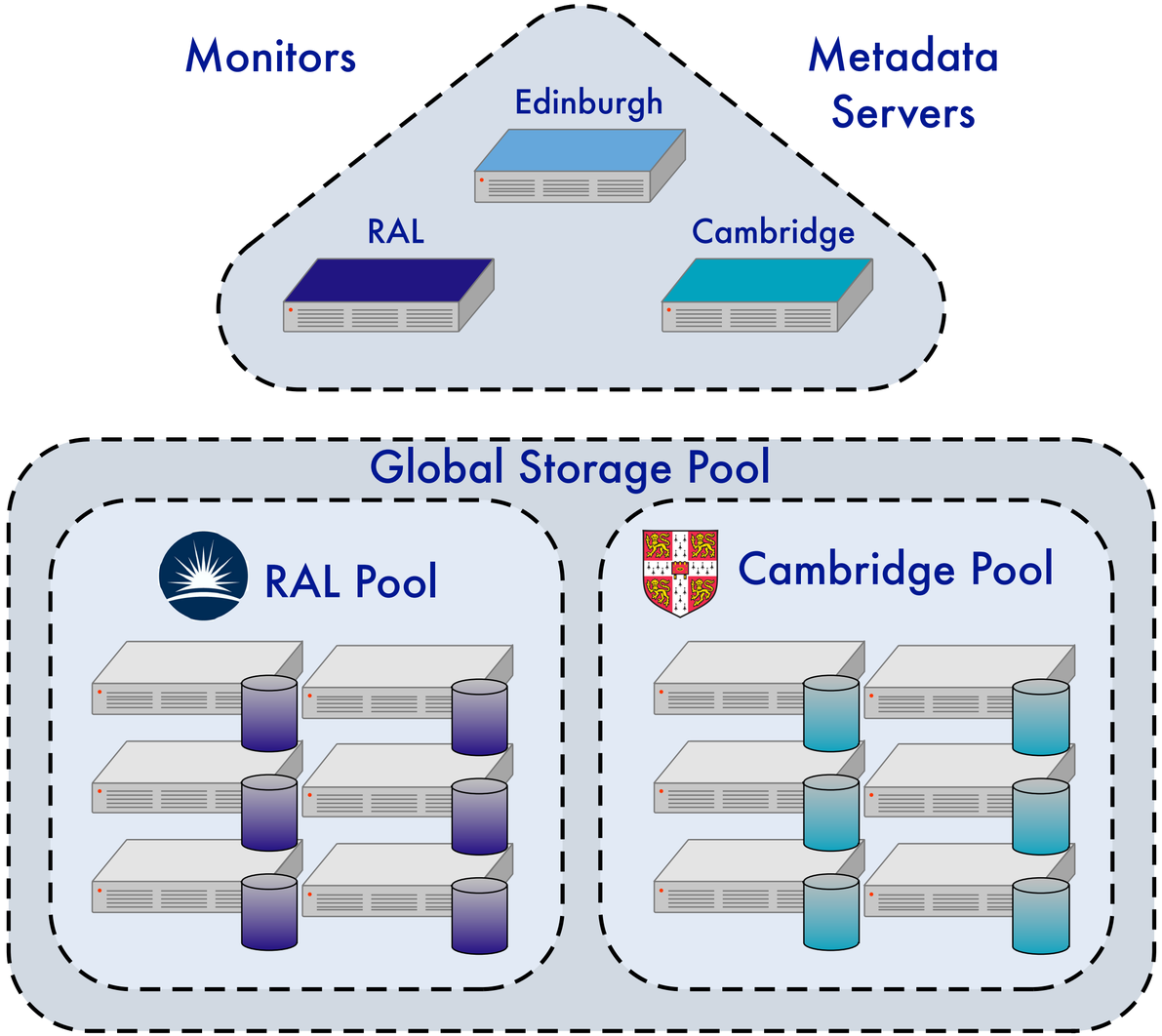
The Illusion of Proximity

- A functional solution using Ansible playbooks
- Create OpenVPN mesh on gateway nodes
- Define routes between subnets
- Open firewall rules to other sites



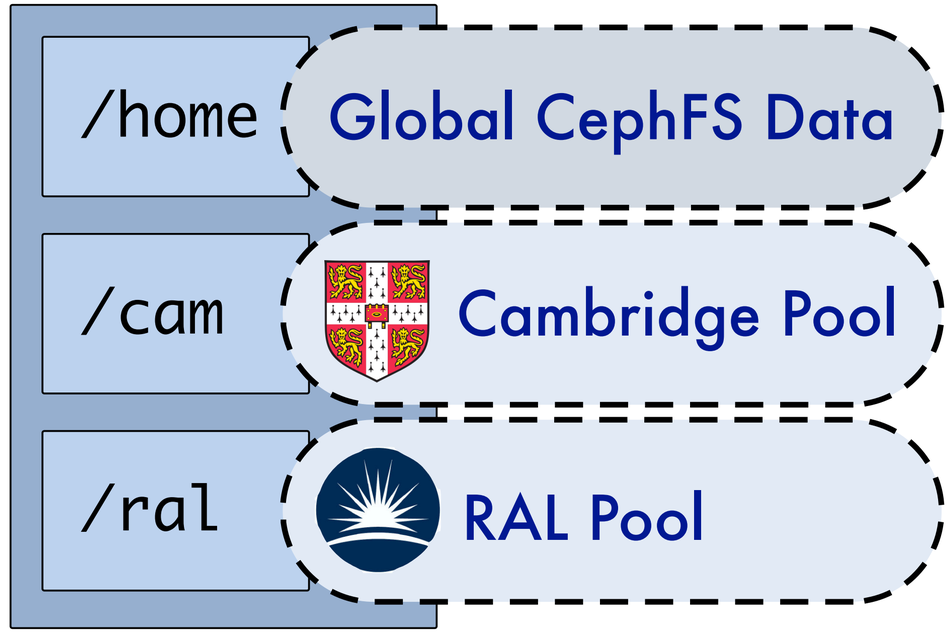
Multi-Site Ceph

- Compute nodes provide converged cluster storage
- Each site runs monitors and metadata servers
- Storage pool across entire deployment - 2-way replication for global data
- Storage pool per site - no replication for local “hot buffer”



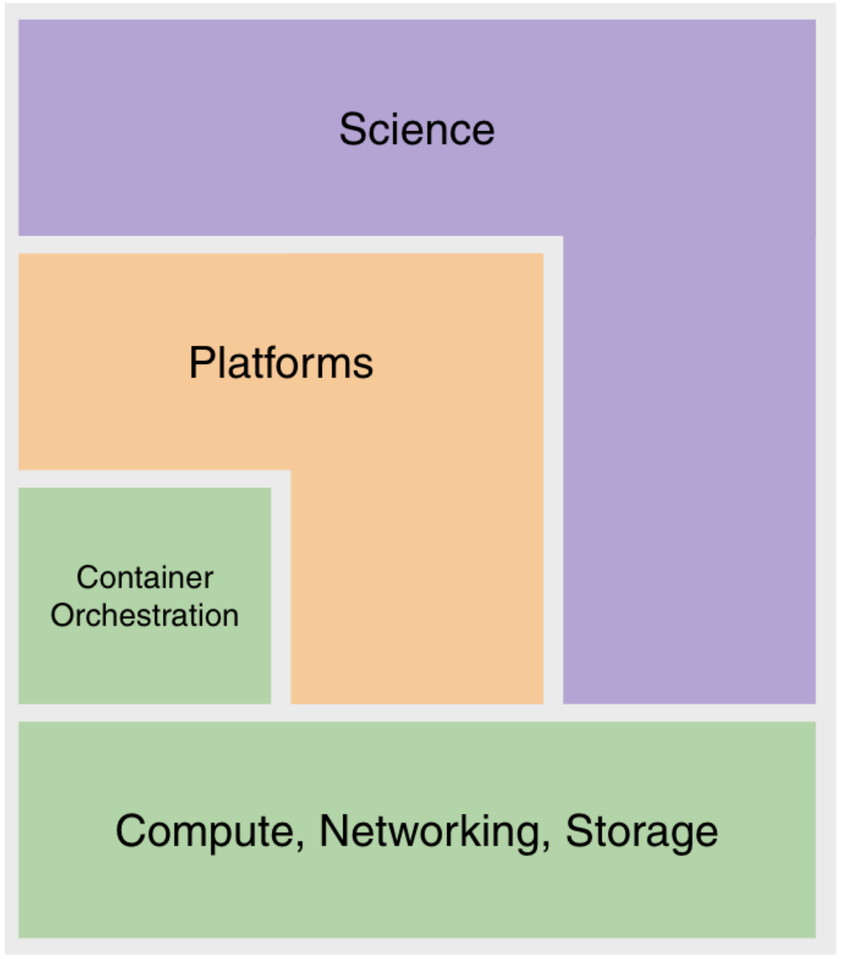
CephFS: A Federated Filesystem?

- Metadata pool across entire deployment
- Data pool across entire deployment
- Data pool and Metadata server per site
- Custom CRUSH rules for each data pool per site
- POSIX extended attributes set the back-end storage pool per file or directory



ceph

Euclid Users



Developer

Platform Ops

Infrastructure Ops

- **Mark, Bryan, Nick**
  POSIX users defined
  Home directories on CephFS
  Access to Slurm and IAL
  ssh public key authentication
- **Mark & StackHPC team**
  Login on Edinburgh control host
  Access to `centos` guest on all hosts in Euclid deployment
- **Mark & StackHPC team**
  Accounts on OpenStack at Cambridge and RAL
  Deploy from Edinburgh control host

The View from Edinburgh

- About 3.2m jobs run so far on the current deployment

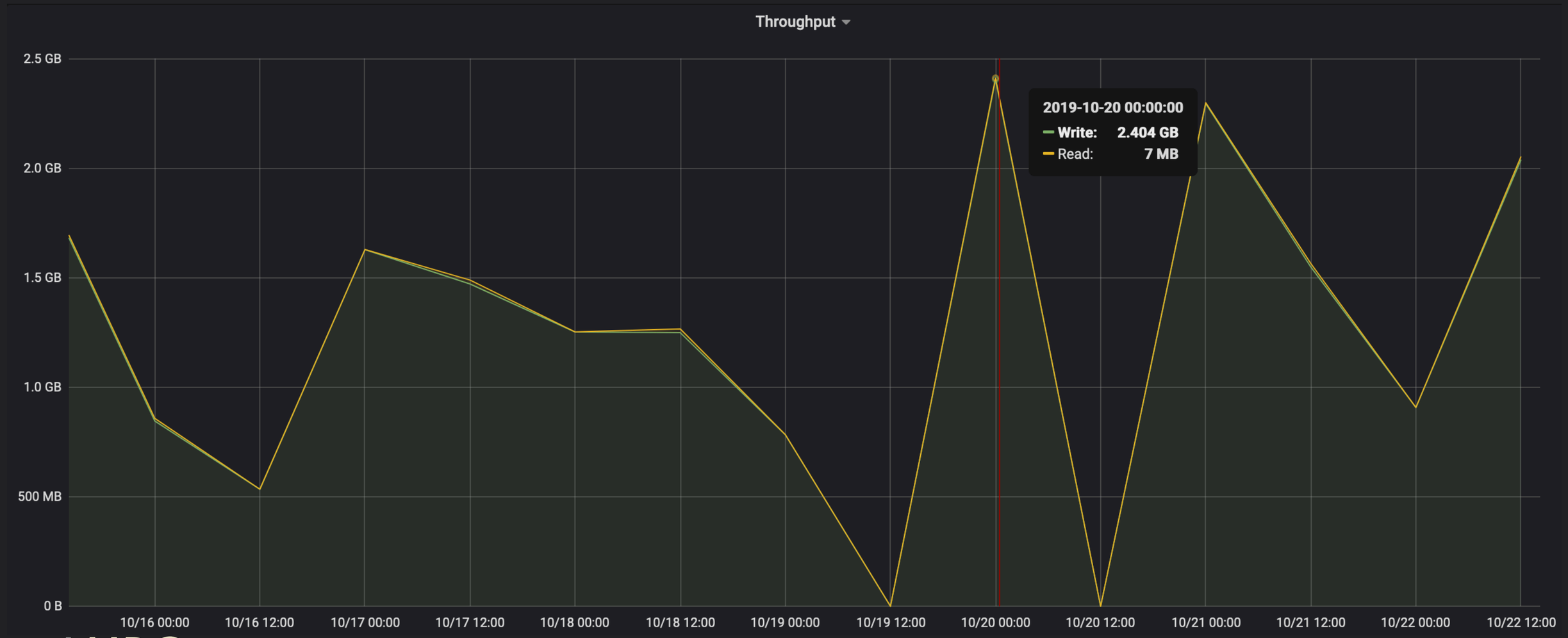
```
[user@euclid-edi-login-0 ~]$ sinfo

PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
cam*         up 1-00:00:00     36  idle  euclid-cam-compute-[0-35]
ral*         up 1-00:00:00    176  idle  euclid-ral-compute-[0-175]
mcr*         up 1-00:00:00    ...  idle  euclid-mcr-compute-[0-..]
```

The View from Inside

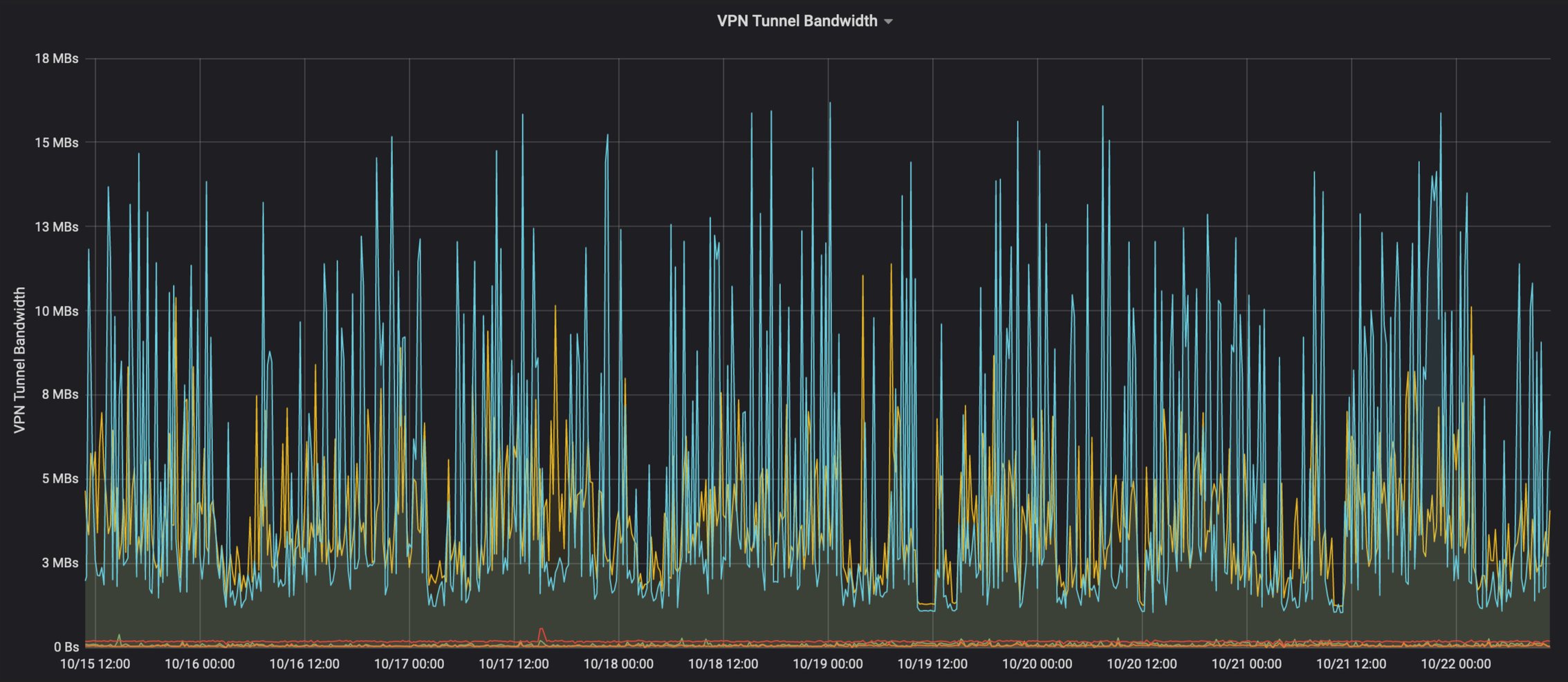
- Ansible playbooks for adding performance telemetry
- Use Prometheus and Grafana for gathering and presentation of telemetry data
- Initial emphasis on Ceph and the VPN tunnels
- Custom telemetry agents for sampling TCP stream back-pressure

ANSIBLE

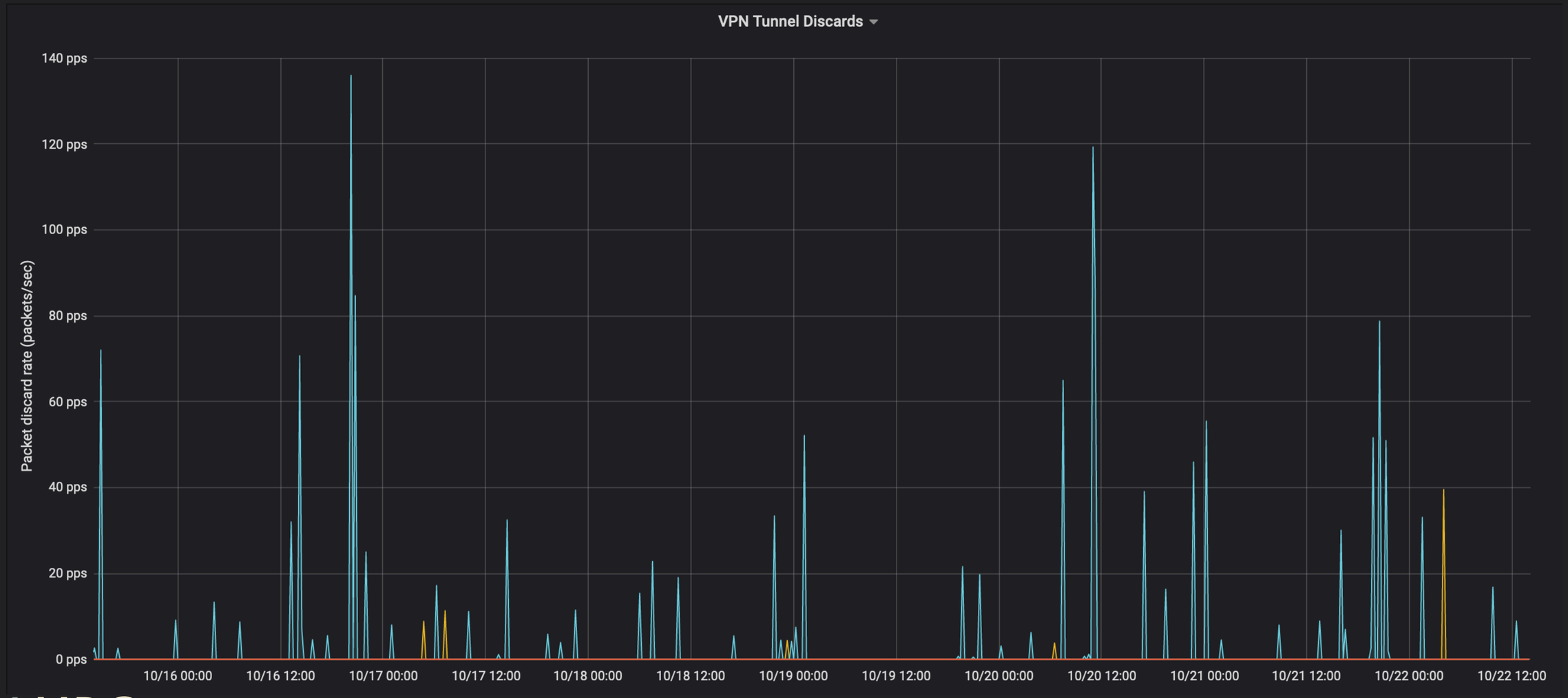
Aggregate Filesystem Bandwidth - ~GB/s



VPN Tunnel Bandwidth - ~MB/s



VPN Tunnel Discards - Occasional Spikes



I/O Latency can be high (as expected)

Bringing it all together

1. Deploy Infrastructure at multiple sites and combine the inventories generated.

2. Resolve cross-cloud portability issues early to make a consistent infrastructure.

3. Connect the sites using OpenVPN tunnels and IP routing between.

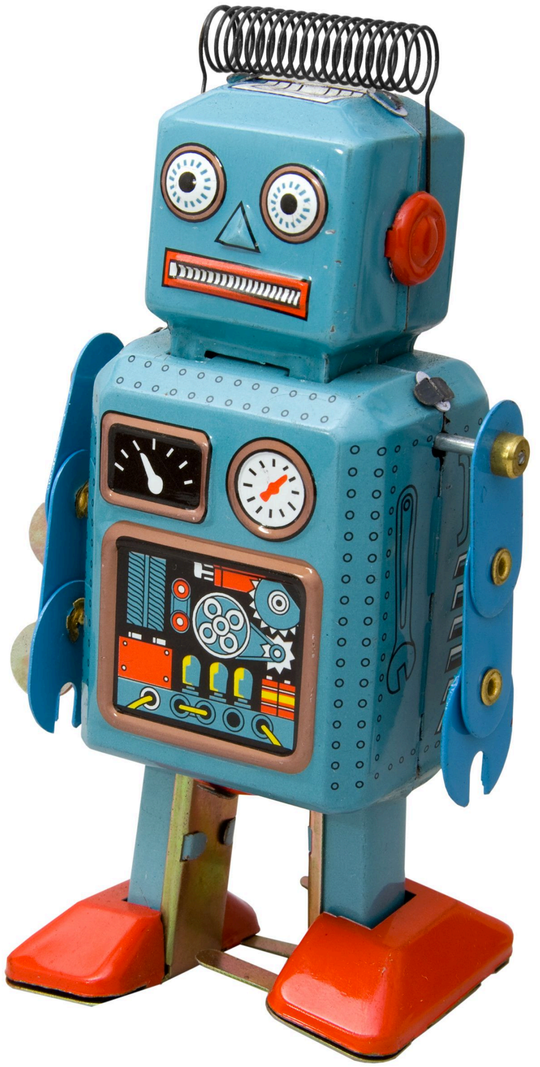
4. Deploy a globally-accessible cluster filesystem.

5. Define science users with local home directories.

6. Install and configure Slurm and science platform software packages.

Share and Enjoy…

- Euclid's "Euclid-as-a-Service" playbook: https://github.com/astrodb/euclid-saas
- Ansible role for deploying software clusters with Heat: https://galaxy.ansible.com/stackhpc/cluster-infra
- Ansible role for OpenVPN federated mesh: https://galaxy.ansible.com/stackhpc/openvpn
- Ansible role for OpenHPC/Slurm deployment: https://galaxy.ansible.com/stackhpc/openhpc
- Prometheus exporter for TCP stream back-pressure: https://github.com/stackhpc/sockpuppet (WIP)



Thank You

Special thanks to:

Mark Holliman, *ROE*

Alex Dibbo, *STFC/RAL*

Paul Browne, *Cambridge University*

John Taylor, Mark Goddard, John Garbutt, Doug Szumski, Bharat Kunwar, StackHPC Ltd

stig@stackhpc.com