OpenHPC: Beyond the Install Guide

for PEARC24

Sharon Colson Jim Moroney Mike Renfro

Tennessee Tech University

2024-07-22

Acknowledgments and shameless plugs

OpenHPC especially Tim Middelkoop (Internet2) and Chris Simmons (Massachusetts Green High Performance Computing Center). They have a BOF at 1:30 Wednesday. You should go to it.

Jetstream2 especially Jeremy Fischer, Mike Lowe, and Julian Pistorius. Jetstream2 has a tutorial at the same time as this one. Please stay here.

NSF CC* for the equipment that led to some of the lessons we're sharing today (award #2127188).

ACCESS current maintainers of the project formerly known as the XSEDE Compatible Basic Cluster.

Where we're starting from

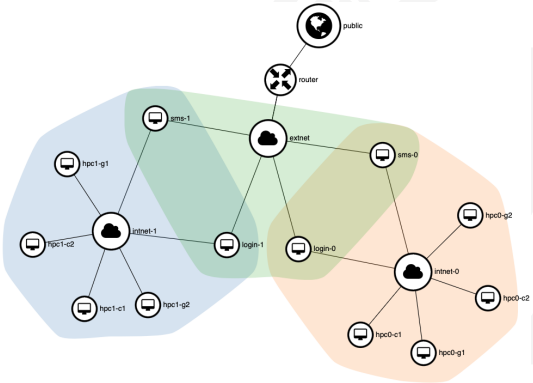

Figure 1: Two example HPC networks

31 HPC clusters (2 shown) with:

1. Rocky Linux 9
2. OpenHPC 3
3. Warewulf 3
4. Slurm
5. 2 non-GPU nodes
6. 2 GPU nodes (currently without GPU drivers, so: expensive non-GPU nodes)
7. 1 management node (SMS)
8. 1 unprovisioned login node

Where we're starting from

We used the OpenHPC automatic installation script from Appendix A with a few variations:

1. Installed `s-nail` to have a valid `MailProg` for `slurm.conf`.
2. Created `user1` and `user2` accounts with password-less `sudo` privileges.
3. Changed `CHROOT` from `/opt/ohpc/admin/images/rocky9.3` to `/opt/ohpc/admin/images/rocky9.4`.
4. Enabled `slurmd` and `munge` in `CHROOT`.
5. Added `nano` and `yum` to `CHROOT`.
6. Removed a redundant `ReturnToService` line from `/etc/slurm/slurm.conf`.
7. Stored all nodes' SSH host keys in `/etc/ssh/ssh_known_hosts`.

Where we're going

1. A slightly more secured SMS
2. A login node that's practically identical to a compute node (except for where it needs to be different)
3. GPU drivers on the GPU nodes
4. Using node-local storage for the OS and/or scratch
5. De-coupling the SMS and the compute nodes (e.g., independent kernel versions)
6. Easier management of node differences (GPU or not, diskless/single-disk/multi-disk, Infiniband or not, etc.)
7. Slurm configuration to match some common policy goals (fair share, resource limits, etc.)

A bit more security for the SMS

(goind to talk about fail2ban here, maybe also firewalld)

Assumptions

1. We have a VM named `login`, with no operating system installed.
2. The `eth0` network interface for `login` is attached to the internal network, and `eth1` is attached to the external network.
3. The `eth0` MAC address for `login` is known—check the **Login server** section of your handout for that. It's of the format `aa:bb:cc:dd:ee:ff`.
4. We're logged into the SMS as `user1` or `user2` that has `sudo` privileges.

Creating a new login node

Working from section 3.9.3 of the install guide:

Make sure to replace the __ with the characters from your login node's MAC address!

```
[user1@sms-0 ~]$ sudo wwsh -y node new login -D eth0 \
    --ipaddr=172.16.0.2 --hwaddr=__:__:__:__:__:__
[user1@sms-0 ~]$ sudo wwsh -y provision set login \
    --vnfs=rocky9.4 --bootstrap=`uname -r` \
    --files=dynamic_hosts,passwd,group,shadow,munge.key,network
```

Semi-stateful node provisioning

(talking about the gparted and filesystem-related pieces here.)

Management of GPU drivers

(installing GPU drivers – mostly rsync'ing a least-common-denominator chroot into a GPU-named chroot, copying the NVIDIA installer into the chroot, mounting /proc and /sys, running the installer, umounting /proc and /sys, and building a second VNFS)

Configuration settings for different node types

(have been leading into this a bit with the wwsh file entries, systemd conditions, etc. But here we can also talk about nodes with two drives instead of one, nodes with and without Infiniband, nodes with different provisioning interfaces, etc.)

Automation for Warewulf3 provisioning

(here we can show some sample Python scripts where we can store node attributes and logic for managing the different VNFSes)

Configuring Slurm policies

Can adapt a lot of Mike's CaRCC Emerging Centers talk from a couple years ago for this. Fair share, hard limits on resource consumption, QOSes for limiting number of GPU jobs or similar.

Sample slide

Left column

This slide has two columns. They don't always have to have columns. It also has a titled block of content in the left column. Make sure you've always got a `::: notes` block after the slide content, even if it has no content.

Use # and ## headers in the Markdown file to make level-1 and level-2 headings, ### headers to make slide titles, and #### to make block titles.