Testing HPC Software Stack with Virtualization

Ahmed Shamsul Arefin, PhD
IT Advisor (HPC Systems), Scientific Computing, IM&T

CSIRO INFORMATION MANAGEMENT & TECHNOLOGY
www.csiro.au

In today's world HPC systems are complex, made from hardware and software not necessarily designed to work together as one complete system. This can result in regular downtimes, hardware errors, as well as software malfunctions. To prevent software faults sysadmins can take a number of predictive measures, such as running a separate test cluster or allocating resources for testing. **In this work, we present a simple and cost effective virtual cluster deployment process, which can facilitate an effective playground for the sysadmins and help to eliminate many simple bugs in HPC environments, such as kernel incompatibility, compiler, software library or drivers issues and so on.**

Test Requirements

We utilized the following hardware and software to create our test environment.

Software: We used the following three main software tools:

- **VMWare [1]**
- **Bright Cluster Manager [2], Easy8 License**
- **SLES15 ISO, from Bright Computing.**

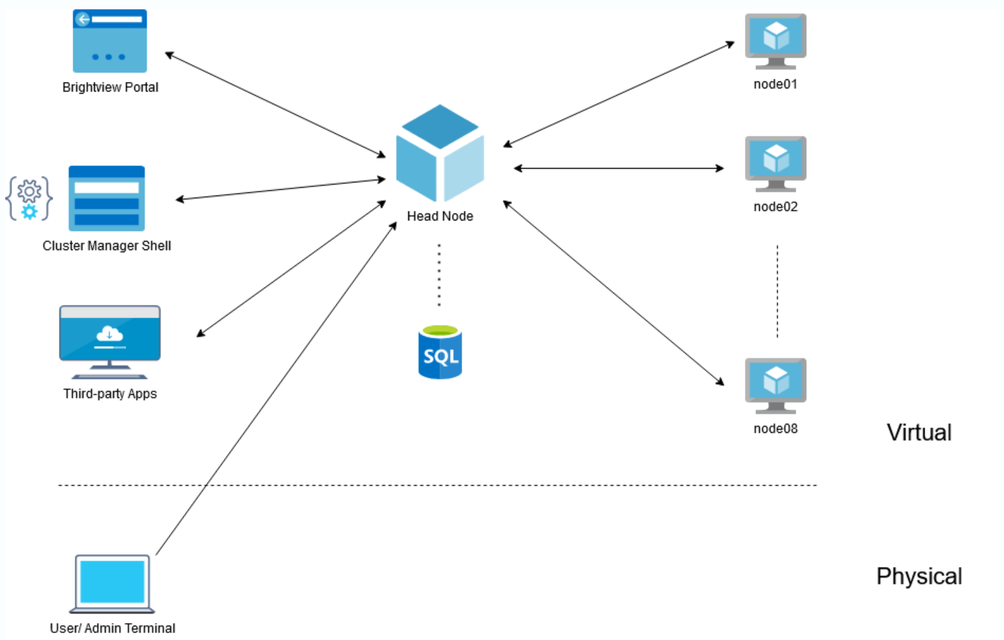

Figure 1: **Creating a virtualized HPC Cluster environment with Bright Cluster Manager.**

The Bright Cluster Manager is a commercially-developed software, but with the Easy8 version it allows to create a free cluster with up to 8 nodes. The Easy8 license also supports CentOS and Ubuntu operating systems. Therefore, the virtual cluster could be **created completely for free replacing** VMWare with KVM or Virtual Box and SLES15 with CentOS/ Ubuntu.

Hardware: We used a recently decommissioned compute node from our production HPC cluster with 16 CPU cores in 2 x Intel Xeon CPU E5-2650 0 @ 2.00GHz, 128GB RAM, 500GB local HDD. In reality is also possible to create proposed the 8-nodes cluster simply on a laptop machine equipped with adequate RAM and CPU cores.

We wanted to create the test environment close to our production HPC cluster at the CSIRO, therefore used the Bright Cluster Manager and SLES15.

Cluster Design & Development

The development process is fairly simple and brief, the outcome with one master and eight HPC compute nodes is outlined in the **Figure 1**.

We started the process by installing a base operating system and a virtualization tool VMWare on the physical hardware. Next, we downloaded the Bright Cluster Manager 9.0 integrated SLES15 ISO software image from the Bright portal. Note that the download process needs a product key which we obtained as part of the Bright Easy8 license. **The VMs deployment process is described below:**

Deployment Steps

We first created the head node VM using this ISO image downloaded as above, the head node installation process is simple and GUI guided, it automatically deployed the DHCP server, schedular and file server. Further details in the Bright Cluster Manager manual [3]. During installation, we can choose how many compute nodes the cluster will have and what the schedular we want to use. We have chosen Slurm as the job schedular.

Next, we created the compute node VMs. While creating compute VMs we pre-allocated the disk storage and assigned MAC address to each node. We only created the VMs, but did not install any OS at this stage.

We then used Bightview, a Bright Cluster Manager admin portal from a browser inside the head node VM and created 8 x compute node profiles where we assigned the previously created MAC addresses. We then turned on the compute VMs, the compute node variant of SLES15 OS installed automatically where the head node dynamically served the OS image, IP addresses and hostnames `node [01-08]`. We used the head node used both as the job schedular and a file server. The default installation came with a number of pre-installed apps, we installed further tools in the NFS mounted `/cmshared` folder as per our testing requirements.

There are various benefits of creating virtualized HPC environment. For instance, by using VMs, different resource configurations, operating systems, and HPC software can be flexibly mixed on the same physical hardware. Time-to-solution for admins and users as per each user's requirements is much reduced, e.g., admins can dynamically resize, pause, take snapshots, back up, replicate to other virtual environments, or simply wipe and redeploy VMs based on their role-based permissions. Since configurations and files are encapsulated within each VM, the VMs can be archived and rerun. Compute resources for VMs can be prioritized individually or in a pool. It's also possible to migrate running VMs and their encapsulated workloads across the cluster for load-balancing. By running jobs in an isolated VM environment, each job is protected from potential faults caused by jobs running in different VMs. Furthermore, hardware maintenance is possible without impacting operational HPC workflows or serviceability, such as automatic restart on another physical servers within the cluster following a server failure or live migration to another physical host when resources of a given host are at capacity.

Conclusions

Our virtualized test cluster (fully deployed < 45 mins) is capable of scheduling jobs and running apps. We tested the latest Slurm, in which we also checked MPI functionality via the virtualized adapters. We also checked the compilers e.g., the latest GCC, ForTran, Python, Bash and some of the commercial software compatibility against the latest OS kernel and libs. We checked the in-house admin scripts, cron jobs and ssh keys and security and some of the firewall features. While the test cluster helped us to create a simple playground for testing software incompatibility issues, **it didn't facilitate any network, GPUs or the OFED/ interconnect tests,** i.e., any actual HPC performance improvements that might happen due to the software updates not tested.

Overall, this development helped us before deploying a new OS/ Kernel / software image into the production quickly and easily, reducing bugs in the later system and enhancing user experience. **We believe it can aid system admins to develop a simple proof of concept HPC environment for a free for cost and help making the end user's HPC environment a bit more tested, easy and conveniently.**

FOR FURTHER INFORMATION
CSIRO IM&T Scientific Computing
p +61 3 9545 7900
e schelp@csiro.au
w www.csiro.au

REFERENCES
1. VMWare https://www.vmware.com/au.html
2. Bright Cluster Manager [2], Easy8 License https://www.brightcomputing.com/easy8
3. Bright Admin Manual https://support.brightcomputing.com/manuals/8.1/admin-manual.pdf

NCI PAWSEY supercomputing centre MASSIVE