SUSE Linux Enterprise High Performance Computing

Installation Guide

SUSE Linux Enterprise High Performance Computing provides a preconfigured and well integrated environment to make deploying and running a High Performance Computing Cluster easy. While providing the same components as SUSE Linux Enterprise, it comes with number of additional tools and libraries specifically targetted at HPC use cases.

Publication Date: February 15, 2022

Contents

1 Conceptual Overview

High Performance Computing (HPC) is the ability to process data and perform complex calculations at high speeds. For example, a supercomputer is a well-known application of HPC. Such supercomputer contains thousands of "compute nodes" working together to complete complex tasks.

These tasks can be weather predictions, analyzing the stock market, or streaming a live event. To process and analyze these massive amount of data, organizations need a reliable and high speed infrastructure like the SUSE Linux Enterprise High Performance Computing.

SUSE Linux Enterprise High Performance Computing contains the following highlights:

- Slurm workload manager
- Module support (Lmod)
- Spack
- Rapid node deployment using Clustduct
- **Performance.** TBD
- **Reliability.** TBD
- **Scalability.** TBD

2 Terminology

TDB

clustduct

A framework which connects a genders database to dnsmasq.

This framework feeds dnsmasq with the node information of a genders database. It can also create a PXE boot file structure with the possiblity to update node MAC addresses in the genders database. In addition, boot images can be managed in the PXE environment.

compute nodes

A node which receives tasks to be run as jobs.

See *head node(s)* and *storage nodes*

genders database

TBD

head node(s)

A node which manages the cluster of the HPC system. Also named as *management node*. See also *compute nodes* and *storage nodes*

HPC cluster

Consists of hundreds or thousands of compute nodes networked together and working in paralell.

storage nodes

A node which stores or provides data to the compute nodes as fast as possible.

3 Usage scenario

SUSE Linux Enterprise High Performance Computing can be used in many variations and scenarios. This is why it is hard to specify a general scenario which fits everywhere. Depending on your use case, your HPC system can be small or large.

Therefor, the procedures in this document will focus on a minimal High Performance Computing system for smaller needs. As a basic requirement you need:

Head node (host name `md0`)

One machine (bare metal or virtual) with two (Ethernet) network cards which has been installed with SLE-HPC15 SP3 chosing the `HPC Managment Server (Head Node)` system role.
Usually, the dimensioning of your head node depends on the size of your cluster.

Compute nodes (host names `c001` and `c002`)

Two machines (bare metal or virtual) which have been installed with SLE-HPC15 SP2 using the system role `Text Mode`. The second node may be prepared by cloning the first one. Each machine needs at least one (Ethernet) network card.
The hardware setup of your compute nodes depends on your use case.

This setup can be used for .

3.1 Required modules

All machines need the product SUSE Linux Enterprise High Performance Computing with the following modules:

- Basesystem Module
- Server Applications Module
- Desktop Applications Module
- Web and Scripting Module
- Development Tools Module
- HPC Module

3.2 Required service components

DNS

DNS is reachable on each machine.
Later, we will discuss the deployment of compute nodes using 'clustduct', as part of this, we will set up our own DNS.

Firewall

The cluster is sufficiently protected by an external firewall.
Later, we will discuss the installation of dedicated login nodes. It is assumed that there is no firewall running on any interface connected to the internal cluster network. Until then, it is assumed that (single) master will serve as a login node for users.

Host name and IP address

all nodes on the internal network have their IP addresses configured (preferrably thru DHCP). However, a static configuration would be sufficient for testing purposes as well.

LDAP, NIS, or other means

A central user account management.

NTP

This service is available on the m01. This server will be set up as NTP server for the internal network.

SSH

Most configurations on the various components on the cluster will be performed from the main management node (m01). Thus we need to be able to access any other nodes as `root` without password.

No AppArmor

Security extensions like AppArmor are either not installed or disabled.

4 Preparing the head node

After logging into the head node (or a separate storage node) as root, perform the following steps:

1. Install SLE-HPC15 SP3 and choose the `HPC Managment Server (Head Node)` system role.
2. *Setting up Genders database*
3. *Setting up SSH access*
4. *Set up network storage*

4.1 Setting up Genders database

While not absolutely essential for this cluster, the genders database offers a good way to classify the different systems in a cluster and refer to them by their roles when performing any configuration.

On the head node, perform the following steps:

1. Install the package on m01:

   ```
   # zypper install -y genders
   ```

2. Add the main management node to create the database:

   ```
   # echo "m01  all,management=main" >> /etc/genders
   ```

3. Repeat this step, if there are further management nodes (replace *N* with the respective number of the node):

   ```
   # echo "mN  all,management=main" >> /etc/genders
   ```

4. If there are separate storage nodes:

```
# echo "sN    all,storage" >> /etc/genders/
```

5. Add the compute nodes:

```
# echo "cN    all,storage" >> /etc/genders/
```

4.2 Setting up SSH access

Most configurations on the various components on the cluster will be performed from the main management node (m01). Thus we need to be able to access any other nodes as root without password. Therefore as a first step an ssh key will be generated and deployed to all other nodes.

For most remote operations we will use pdsh which allows one to perfrom the same operation on multiple nodes. pdsh is able to use mrsh as remote shell using the munge authentication. For root access, munge authentication is disabled by default, therefore this guide relies on ssh for remote root access. This will require the root ssh public key to be deployed to all systems - unfortunately, we won't be able to use pdsh for this.

It should be noted that ssh uses priviledged ports obtained by rresvport() for communication. The number of ports available here is limited, therefore these may get exhausted if multiple pdsh's are running on a machine and the fanout is too high. 'mrsh' does not use privileged ports and thus is not affected by this problem as severely.

Later, a scalable method for provisioning compute nodes will be introduced. This method will not suffer from this problem.

1. Generate an SSH key:

```
# ssh-keygen -t rsa
```

   You may want to specify a password here. This will require you to run an ssh-agent and add your private key to it to avoid being promted for a password. If you want to script some configuration commands, you may want to create a separate, password-less key which you deploy to the nodes that you want to (re-)configure from a script.

2. Run bash as ssh-agent and add your key:

```
# ssh-agent bash
# ssh-add
```

3. Enter your password.

4. Copy your SSH key to all nodes:

   ```
   # for i in $(nodeattr -n "management=aux||storage||compute"); do \
     ssh-copy-id -o "StrictHostKeyChecking accept-new" root@$i; \
   done
   ```

5. Enter your passwords on every system that is to be accessed.
   Once the root ssh key has been deployed do all other nodes in the cluster there is no need to access nodes individually.

6. On the management node, install the following packages:

   ```
   # zypper install pdsh pdsh-genders
   ```

7. To test whether the pdsh setup works, run the following command:

   ```
   # pdsh -R ssh -g all hostname -f
   ```

   The command prints a list of fully qualified hostnames.

8. Fix the names that are known under the internal network.
   The hostnames known to the nodes may not correspond to the hostnames known to the system. For some of the HPC services it is important, that the system know the names they are known under the internal network. Run the following command to fix this:

   ```
   # pdsh -R exec -g all ssh root@%h /bin/sh -c 'echo %h > /etc/hostname'
   ```

4.3 Set up network storage

1. Make sure, the NFS server is installed:

   ```
   # zypper install nfs-kernel-server
   ```

2. Export NFS file systems:

   a. Export home:

      ```
      # IF=eth1; network=${ip route | grep $IF |  cut -f 1 -d' '} \
      echo "/home   $network(rw,sync,nohide)" >> /etc/exports
      ```

b. Export data store

3. Enable and start the NFS server:

```
# systemctl enable --now nfs-server
```

5 Preparing cluster setup

5.1 Install munge and distribute munge key to all cluster nodes

1. 'munge' is the authentication used by mrsh, also configure the workload manager Slurm to use munge:

```
# pdsh -R ssh -g "management||compute" zypper install -y munge
# pdcp -R ssh -g "management=aux||compute" /etc/munge/munge.key /etc/munge
# pdsh -R ssh -g "management||compute" chmod 0400 /etc/munge/munge.key
# pdsh -R ssh -g "management||compute" chown munge:munge /etc/munge/munge.key
```

2. Enable and start munged on each system:

```
# pdsh -R ssh -g "management||compute" systemctl enable --now munge
```

5.2 Distribute NTP configuration and synchronize time across all cluster nodes

1. Set up m01 as time server:

```
# zypper install -y chrony
```

2. Set m01 as master server for the internal network (eth1 is assumed to be the network interface of the internal network):

```
# IF=eth1; network=${ip route | grep $IF |  cut -f 1 -d' '}; \
echo "allow $network > /etc/chrony.d/master.conf
# systemctl enable --now chronyd
```

A default timeserver is set in /etc/chrony.d/pool.conf at installation time. For the purpose of this guide it is assumed that this time source is accessible from m01. If another source is desired, this setting may be changed.

3. Set m01 as time source for the other nodes on the cluster, enable und start chronyd:

```
# pdsh -R ssh -g "~management=main" zypper install -y chrony
# pdsh -R ssh -g "~management=main" sh -c 'echo "pool m01 iburst" \
  > /etc/chrony.d/pool.conf'
# pdsh -R ssh -g "~management=main" systemctl enable --now chronyd
```

For redundancy it is possible to set up a second time server on another management node. If this node has no access to the outside network, it will may use m01 as its time source. In case of faiure of m01 it will continue to serve time using its internal clock thus making sure the clocks within the cluster remain in sync.

5.3 Syncing genders database across all cluster nodes

Use the following command:

```
# pdcp -R ssh -g "~management=main" /etc/genders /etc
```

5.4 Add mount option for NFS to all nodes on a cluster

Make sure, all NFS exported file systems (ie /home and possibly other storage partitions) are mounted on all the nodes within the cluster. Here it is assumed only /home is exported from m01. If a separate storage server is used or more directories are exported, the commands will have to be adapted.

```
# server=m01; pdsh -R ssh -g "~(management=main||storage)" \
sh -c "echo \"${server}/home  /home nfs  defaults 0 1\" >> /etc/fstab"
# psdh -R ssh -g "~(management=main||storage)" mount /home
```

5.5 Install the Slurm workload manager across the cluster

If the management node has been installed using the Management Node profile, Slurm should be installed there already. To make sure, run:

```
# zypper install -y slurm slurm-munge
```

We will configure Slurm with 'munge' authentication. For this, we need to ensure, that the `slurm` user has the same UID across the entire cluster. When installing Slurm on a node, this user will be created if it doesn't exist already. To ensure the UID is uniform across the cluster, the `slurm` user should be configured on the nodes before installing any Slrum packages.

6 Further adaptions

In case your computing power is not enough, you can extend your HPC system:

More than one head node

Can be a clone of `m01`.

Storage node

Whose data is exported via NFS. This is used to export user homes as well as provide shared application and library stacks.

If such a node is not available, it is assumed that the single head node is installed in such a way that an additional physical partitions are available from which this data can be exported. One of these partitions is mounted to `/home`.

Database node

To store accounting data.

High-speed network

In addition to the IP networking, a high-speed network (Infiniband, Omni-Path) may exist which is also connected to each node. This high speed interconnect is used for application Message Passing (MPI) and optionally for connecting a parallel file system.