Install Slurm

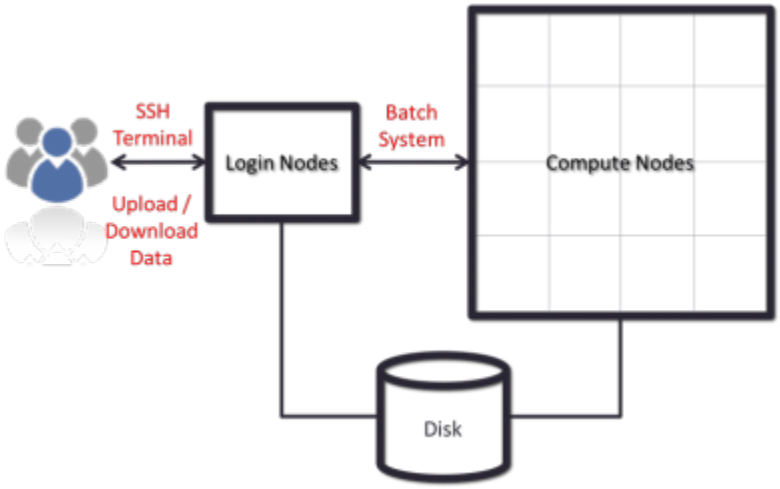
What is a High-Performance Computer?

A high-performance computer (HPC system) is a tool used by computational scientists and engineers to tackle problems that require more computing resources or time than they can obtain on the personal computers available to them.

What is a HPC (High-Performance Computer)?

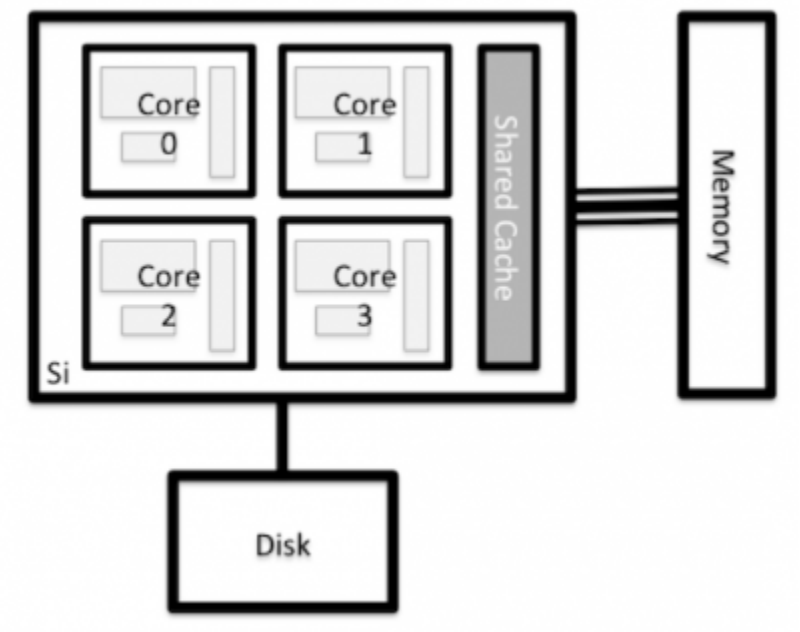
* Computers connected by some type of network (ethernet, infiniband, etc.). * These computers is often referred to as a node. * Several different types of nodes, specialized for different purposes. * Head (front-end or/and login): where you login to interact with the HPC system. * Compute nodes (CPU, GPU) are where the real computing is done. Access to these resources is controlled by a scheduler or batch system.

HPC architecture



Nodes

Each node on an HPC system is essentially an individual computer (physical or virtual):



Parts of HPC system

- Servers:

1. Server Master: master1, [master2 (Backup)]
2. Server login
3. Server accounting: mysql server
4. Nodes: Physical or/and VM

- Storage: (CEPH, Lustre, etc.)
- Scheduler: Slurm, torquePBS, etc.
- Applications: Monitoring, software control, etc.
- Users database: OpenLDAP, AD, etc.

Storage and file systems

- Lustre: is a parallel distributed file system.
- Spectrum Scale: is a scalable high-performance data management solution.
- BeeGFS: was developed for I/O-intensive HPC applications.
- OrangeFS: is a parallel distributed file system that runs completely in user space.
- Ceph: that offers file-, block- and object-based data storingon a single distributed cluster.
- GlusterFS: has a client-server model but does not need a dedicatedmetadata server.

Scheduler

In order to share these large systems among many users, it is common to allocate subsets of the compute nodes to tasks (or jobs), based on requests from users. These jobs may take a long time to

complete, so they come and go in time. To manage the sharing of the compute nodes among all of the jobs, HPC systems use a batch system or scheduler.

The batch system usually has commands for submitting jobs, inquiring about their status, and modifying them. The HPC center defines the priorities of different jobs for execution on the compute nodes, while ensuring that the compute nodes are not overloaded.

A typical HPC workflow could look something like this:

- Transfer input datasets to the HPC system (via the login nodes)
- Create a job submission script to perform your computation (on the login nodes)
- Submit your job submission script to the scheduler (on the login nodes)
- Scheduler runs your computation (on the compute nodes)
- Analyze results from your computation (on the login or nodes, or transfer data for analysis elsewhere)

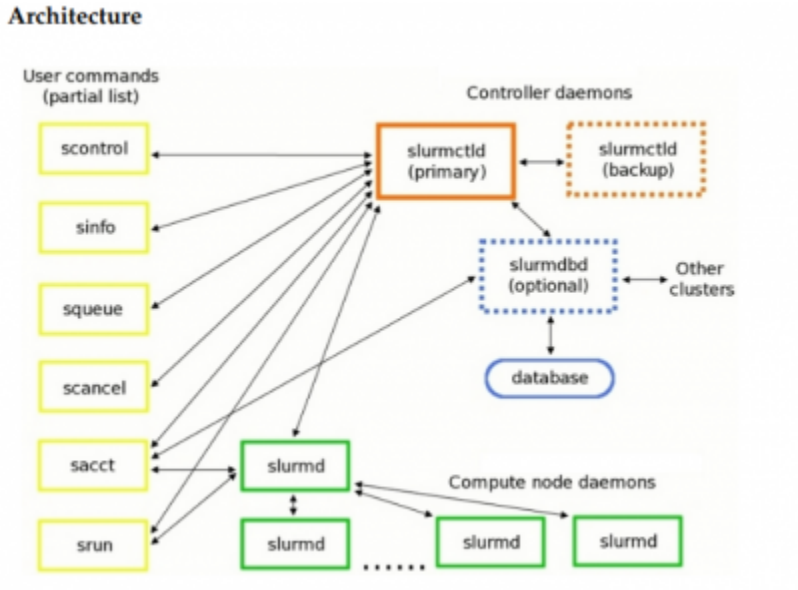
Slurm:

- Is an open source, and highly scalable cluster management and job scheduling system.
- Requires no kernel modifications for its operation
- Is relatively self-contained.

Slurm has three key functions

- It allocates exclusive and/or non-exclusive access to resources (nodes) to users for some duration of time so they can perform work.
- It provides a framework for starting, executing, and monitoring work on the set of allocated nodes.
- It arbitrates contention for resources by managing a queue of pending work.

Slurm architecture



Commands:

- scontrol: is the administrative tool used to view and/or modify Slurm state.
- sinfo: reports the state of partitions and nodes managed by Slurm.
- squeue: reports the state of jobs or job steps.
- scancel: to cancel a pending or running job or job step.
- sacct: is used to report job or job step accounting information about jobs.
- srun: to submit a job for execution or initiate job steps in real time.

Applications

- Lifecycle management tool: Foreman or PXE server [PXE-DNS-DHCP-TFTP]
- Monitoring services: Nagios, Icinga, etc.
- Monitoring resources (menory, CPU load, etc): gaglia
- SSH server: openssh
- Central control configurations server: Puppet, ansible,
- manage (scientific) software: Easybuild
- Environment Module System: LMOD
- Central users database: OpenLDAP, Active Directory, etc.====

Slurm Installation

1. System operation: CentOS 7 or 8
2. Define name of every server (hostname): master, node01.. nodeN
3. Install database server (MariaDB)
4. Create global users (munge user). Slurm and Munge require consistent UID and GID across every node in the cluster.
5. Install Munge
6. Install Slurm

Create the global user and group for Munge

For all the nodes before you install Slurm or Munge, you need create user and group using seem UID and GID:

```
export MUNGEUSER=991
groupadd -g $MUNGEUSER munge
useradd  -m -c "MUNGE Uid 'N' Gid Emporium" -d /var/lib/munge -u $MUNGEUSER -g
munge  -s /sbin/nologin munge
```

```
export SLURMUSER=992
groupadd -g $SLURMUSER slurm
useradd  -m -c "SLURM workload manager" -d /var/lib/slurm -u $SLURMUSER -g
slurm  -s  /bin/bash slurm
```

Install Slurm dependencies

In every node we need install a few dependencies:

```
yum install openssl openssl-devel pam-devel numactl numactl-devel hwloc hwloc-
devel lua lua-devel readline-devel rrdtool-devel ncurses-devel man2html
libibmad libibumad perl-ExtUtils-MakeMaker gcc -y
```

Install Munge

In every node we need install that:

We need to get the latest EPEL repository:

```
yum install epel-release
yum update
```

For CentOS 8, we need edit file /etc/yum.repos.d/CentOS-PowerTools.repo, and enable repository

Change:

```
enable=0 by enable=1
```

- Update database repository:

```
yum update
```

After that, we can install Munge

```
yum install munge munge-libs  munge-devel -y
```

In the server master, we need create Munge key and copy that to all another server

```
yum install rng-tools -y
rngd -r /dev/urandom
```

Creating Munge key

```
/usr/sbin/create-munge-key -r
dd if=/dev/urandom bs=1 count=1024 > /etc/munge/munge.key
chown munge: /etc/munge/munge.key
chmod 400 /etc/munge/munge.key
```

Copying Munge key to another server

```
scp /etc/munge/munge.key root@node01:/etc/munge
scp /etc/munge/munge.key root@node02:/etc/munge
.
.
.
scp /etc/munge/munge.key root@nodeN:/etc/munge
```

In every node we need correct the permissions as well as enable and start the Munge service.

```
chown -R munge: /etc/munge/ /var/log/munge/
chmod 0700 /etc/munge/ /var/log/munge/
```

```
systemctl enable munge
systemctl start munge
```

To test Munge, we can try to access another node with Munge from our server node.

```
munge -n
munge -n | unmunge
munge -n | ssh node01.cluster.test unmunge
remunge
```

Create Slurm rpm packages

At the server, download the last version of Slurm. At this moment the last version es 19.05.5

```
cd /tmp
wget https://download.schedmd.com/slurm/slurm-19.05.5.tar.bz2
yum install rpm-build
```

```
rpmbuild -ta slurm-19.05.5.tar.bz2
```

Copying the Slurm rpm files for installation from the master to the other servers o to a shared folder.

```
cd ~/rpmbuild/RPMS/x86_64
cp slurm*.rpm /fns/shared_folder
```

Install Slurm in the Master, compute, and Login nodes:

The slurm-torque package could perhaps be omitted, but it does contain a useful /usr/bin/mpiexec wrapper script.

Before install Slurm, we need disable selinux

```
nano /etc/selinux/config
```

change SELINUX=enforcing to SELINUX=disables

```
cd ~/rpmbuild/RPMS/x86_64
export VER=19.05.5-1
yum install slurm-$VER*rpm slurm-devel-$VER*rpm slurm-perlapi-$VER*rpm slurm-
torque-$VER*rpm slurm-example-configs-$VER*rpm
```

Explicitly enable the service in the master

```
systemctl enable slurmctld
```

Only if the database service will run on the Master node: Install the database service RPM:

```
cd ~/rpmbuild/RPMS/x86_64
export VER=19.05.5-1
yum install slurm-slurmdbd-$VER*rpm
```

If you have a server for database, install in this server:

```
export VER=19.05.5-1
yum install slurm-$VER*rpm slurm-devel-$VER*rpm slurm-slurmdbd-$VER*rpm
```

Explicitly enable the service:

```
systemctl enable slurmdbd
```

We need to make sure that the server has all the right configurations and files.

```
mkdir /var/spool/slurmctld
```

```
chown slurm: /var/spool/slurmctld
chmod 755 /var/spool/slurmctld
touch /var/log/slurmctld.log
chown slurm: /var/log/slurmctld.log
touch /var/log/slurm_jobacct.log /var/log/slurm_jobcomp.log
chown slurm: /var/log/slurm_jobacct.log /var/log/slurm_jobcomp.log
```

Compute nodes

On Compute nodes you may additionally install the slurm-slurmd and slurm-pam_slurm RPM package to prevent rogue users from logging in:

```
export VER=19.05.5-1
yum install slurm-slurmd slurm-pam-$VER*rpm_slurm-$VER*rpm
systemctl enable slurmd
```

We need to make sure that all the compute nodes have the right configurations and files.

```
mkdir /var/spool/slurmd
chown slurm: /var/spool/slurmd
chmod 755 /var/spool/slurmd
touch /var/log/slurmd.log
chown slurm: /var/log/slurmd.log
```

Slurm configuration

Slurm provides an example file located at /etc/slurm/slurm.conf.example. You can copy this file to /etc/slurm/slurm.conf

```
cp /etc/slurm/slurm.conf.example /etc/slurm/slurm.conf
```

It also have a web-based configuration tool which can be used to build a simple configuration file, which can then be manually edited for more complex configurations.

After that we need to edit /etc/slurm/slurm.conf and make some modifications. Its

```
vi /etc/slurm/slurm.conf
```

It is important to change the parameters: ClusterName and ControlMachine.

```
ClusterName=vlir-test
ControlMachine=10.10.2.242
SlurmUser=slurm
SlurmctldPort=6817
```

```
SlurmdPort=6818
AuthType=auth/munge
StateSaveLocation=/var/spool/slurm/ctld
SlurmdSpoolDir=/var/spool/slurm/d
SwitchType=switch/none
MpiDefault=none
SlurmctldPidFile=/var/run/slurmctld.pid
SlurmdPidFile=/var/run/slurmd.pid
ProctrackType=proctrack/pgid
ReturnToService=0
```

If the /var/spool directory does not exist, you need to create it.

```
mkdir /var/spool/slurm
chown slurm.slurm -R /var/spool/slurm
```

Slurm logging

The Slurm logfile directory is undefined in the RPMs since you have to define it in slurm.conf. See SlurmdLogFile and SlurmctldLogFile in the slurm.conf page, and LogFile in the slurmdbd.conf page.

Check your logging configuration with:

```
grep -i logfile /etc/slurm/slurm.conf
```

```
SlurmctldLogFile=/var/log/slurm/slurmctld.log
SlurmdLogFile=/var/log/slurm/slurmd.log
```

```
scontrol show config | grep -i logfile
```

```
SlurmctldLogFile        = /var/log/slurm/slurmctld.log
SlurmdLogFile           = /var/log/slurm/slurmd.log
SlurmSchedLogFile       = /var/log/slurm/slurmsched.log
```

If log files are configured, you have to create the log file directory manually:

```
mkdir /var/log/slurm
chown slurm.slurm /var/log/slurm
```

Study the configuration information in the Quick Start Administrator_Guide.

Home Users

For the users folder, you can use the server's local disk or mount the remote storage. For this reason it is

recommended to create a folder to put the information of the users. In this example we created a folder /home/CLUSTER and here we create the folder for every users.

```
mkdir /home/CLUSTER
```

Creating users

For the users you can crate every user manually o you can user an external user database how Active Directory, OpenLDAP or MySQL, etc. For this example we going to create the users manually in every server.