- Home
- Linux
  - For Dummies
  - For Jedi
- Coding
  - Python
  - Perl
- HPC Usage
  - HPC
  - HPC Ouaga
  - Advanced HPC
- HPC Admin
  - Module1
  - Module2
- Workflow
  - TOGGLe
  - Galaxy
  - GIGWA
  - Workflow Manager
- NGS
  - SV detection
  - Bacterial genomics
  - RNASeq
  - Meta- barcoding
  - ONT
  - Annotation
  - TrinityTrinotate
- One-week
  - RNASeq Colombie
  - ONT virus Burkina
  - ONT Senegal
  - NGS Senegal
  - GBS Guadeloupe
  - NGS Burkina
- About Us
- GitHub

South Green Trainings pages

Description	Installation of Slurm on centos 7
Related-course materials	HPC Administration Module2
Authors	Ndomassi TANDO (ndomassi.tando@ird.fr)
Creation Date	23/09/2019
Last Modified Date	23/09/2019

Summary

- Definition
- Authentication and databases
- Slurm installation
- Configure usage limit
- Links
- License

Definition

Slurm is an open source, fault-tolerant, and highly scalable cluster management and job scheduling system for large and small Linux clusters.

https://slurm.schedmd.com/

Authentication and databases:

Create the user for munge and slurm:

Slurm and Munge require consistent UID and GID across every node in the cluster. For all the nodes, before you install Slurm or Munge:

```
$ export MUNGEUSER=1001
$ 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=1002
$ groupadd -g $SLURMUSER slurm
$ useradd  -m -c "SLURM workload manager" -d /var/lib/slurm -u $SLURMUSER -g slurm  -s /bin/bash slurm
```

Munge Installation for authentication:

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

Create a munge authentication key:

```
$ /usr/sbin/create-munge-key
```

Copy the munge authentication key on every node:

```
$ cp /etc/munge/munge.key /home
$ cexec cp /home/munge.key /etc/munge
```

Set the rights:

```
$ chown -R munge: /etc/munge/ /var/log/munge/ /var/lib/munge/ /run/munge/
$ chmod 0700 /etc/munge/ /var/log/munge/ /var/lib/munge/ /run/munge/
$ cexec chown -R munge: /etc/munge/ /var/log/munge/ /var/lib/munge/ /run/munge/
$ cexec chmod 0700 /etc/munge/ /var/log/munge/ /var/lib/munge/ /run/munge/
```

Enable and Start the munge service with:

```
$ systemctl enable munge
$ systemctl start munge
$ cexec systemctl enable munge
$ cexec systemctl start munge
```

Test munge from the master node:

```
$ munge -n | unmunge
$ munge -n | ssh <somehost_in_cluster> unmunge
```

Mariadb installation and configuration

Install mariadb with the following command:

```
$ yum install mariadb-server -y
```

Activate and start the mariadb service:

```
$ systemctl start mariadb
systemctl enable mariadb
```

secure the installation:

Launch the following command to set up the root password an secure mariadb:

```
$ mysql_secure_installation
```

Modify the innodb configuration:

Setting innodb_lock_wait_timeout,innodb_log_file_size and innodb_buffer_pool_size to larger values than the default is recommended.

To do that, create a the file `/etc/my.cnf.d/innodb.cnf` with the following lines:

```
[mysqld]
 innodb_buffer_pool_size=1024M
 innodb_log_file_size=64M
 innodb_lock_wait_timeout=900
```

To implement this change you have to shut down the database and move/remove logfiles:

```
$ systemctl stop mariadb
 mv /var/lib/mysql/ib_logfile? /tmp/
 systemctl start mariadb
```

Slurm installation:

Install the following prerequisites:

```
$ yum install openssl openssl-devel pam-devel rpmbuild numactl numactl-devel hwloc hwloc-devel lua lua-devel readline-devel rrdtool-devel ncurses-d
```

Retrieve the tarball

```
$ wget https://download.schedmd.com/slurm/slurm-19.05.0.tar.bz2
```

Create the RPMs:

```
$ rpmbuild -ta slurm-19.05.0.tar.bz2
```

RPMs are located in /root/rpmbuild/RPMS/x86_64/

Install slurm on master and nodes

In the RPMs'folder, launch the following command:

```
$ yum --nogpgcheck localinstall slurm-*
```

Create and configure the slurm_acct_db database:

```
$ mysql -u root -p
 mysql> grant all on slurm_acct_db.* TO 'slurm'@'localhost' identified by 'some_pass' with grant option;
 mysql> create database slurm_acct_db;
```

Configure the slurm db backend:

Modify the `/etc/slurm/slurmdbd.conf` with the following parameters:

```
AuthType=auth/munge
 DbdAddr=192.168.1.250
 DbdHost=master0
 SlurmUser=slurm
 DebugLevel=4
 LogFile=/var/log/slurm/slurmdbd.log
 PidFile=/var/run/slurmdbd.pid
 StorageType=accounting_storage/mysql
 StorageHost=master0
 StoragePass=some_pass
 StorageUser=slurm
 StorageLoc=slurm_acct_db
```

Then enable and start the slurmdbd service

```
$ systemctl start slurmdbd
$ systemctl enable slurmdbd
$ systemctl status slurmdbd
```

This will populate the slurm_acct_db with tables

Configuration file /etc/slurm/slurm.conf:

using the command `lscpu` on each node to get processors' informations.

Visit http://slurm.schedmd.com/configurator.easy.html to make a configuration file for Slurm.

Modify the following parameters in `/etc/slurm/slurm.conf` to match with your cluster:

```
ClusterName=IRD
ControlMachine=master0
ControlAddr=192.168.1.250
SlurmUser=slurm
AuthType=auth/munge
StateSaveLocation=/var/spool/slurmd
SlurmdSpoolDir=/var/spool/slurmd
SlurmctldLogFile=/var/log/slurm/slurmctld.log
SlurmdDebug=3
SlurmdLogFile=/var/log/slurm/slurmd.log
AccountingStorageHost=master0
AccountingStoragePass=3devslu!!
AccountingStorageUser=slurm
NodeName=node21 CPUs=16 Sockets=4 RealMemory=32004 CoresPerSocket=4 ThreadsPerCore=1 State=UNKNOWN
PartitionName=r900 Nodes=node21 Default=YES MaxTime=INFINITE State=UP
```

Now that the server node has the slurm.conf and slurmdbd.conf correctly filled, we need to send these filse to the other compute nodes.

```
$ cp /etc/slurm/slurm.conf /home
$ cp /etc/slurm/slurmdbd.conf /home
```

```
$ cexec cp /home/slurm.conf /etc/slurm
$ cexec cp /home/slurmdbd.conf /etc/slurm
```

Create the folders to host the logs

On the master node:

```
$ mkdir /var/spool/slurmctld
$ chown slurm:slurm /var/spool/slurmctld
$ chmod 755 /var/spool/slurmctld
$ mkdir  /var/log/slurm
$ touch /var/log/slurm/slurmctld.log
$ touch /var/log/slurm/slurm_jobacct.log /var/log/slurm/slurm_jobcomp.log
$ chown -R slurm:slurm /var/log/slurm/
```

On the compute nodes:

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

test the configuration:

```
$ slurmd -C
```

You should get something like:

```
NodeName=master0 CPUs=16 Boards=1 SocketsPerBoard=2 CoresPerSocket=4 ThreadsPerCore=2 RealMemory=23938 UpTime=22-10:03:46
```

Launch the slurmd service on the compute nodes:

```
$ systemctl enable slurmd.service
$ systemctl start slurmd.service
$ systemctl status slurmd.service
```

Launch the slurmctld service on the master node:

```
$ systemctl enable slurmctld.service
$ systemctl start slurmctld.service
$ systemctl status slurmctld.service
```

Change the state of a node from down to idle

```
$ scontrol update NodeName=nodeX State=RESUME
```

Where nodeX is the name of your node

Configure usage limits

Modify the /etc/slurm/slurm.conf file

Modify the `AccountingStorageEnforce`parameter with:

```
AccountingStorageEnforce=limits
```

Copy the modified file to the several nodes

Restart the slurmctld service to validate the modifications:

```
$ systemctl restart slurmctld
```

Create a cluster:

The cluster is the name we want for your slurm cluster.

It is defined in the `/etc/slurm/slurm.conf` file with the line

```
ClusterName=ird
```

To set usage limitations for your users, you first have to create an accounting cluster with the command:

```
$sacctmgr add cluster ird
```

Create an accounting account

An accounting account is a group under slurm that allows the administrator to manage the users rights to use slurm.

Example: you can create a account to group the bioinfo teams members

```
$ sacctmgr add account bioinfo Description="bioinfo member"
```

You can create a account to group the peaople allow to use the gpu partition

```
$ sacctmgr add account gpu_group Description="Members can use the gpu partition"
```

Create a user account

You have to create slurm user to make them be able to launch slurm jobs.

```
$ sacctmgr create user name=xxx DefaultAccount=yyy
```

Modify a user account to add it to another accounting account:

```
$ sacctmgr add user xxx Account=zzzz
```

Modify a node definition

Add the amount of /scratch partition

In the file `/etc/slurm/slurm.conf`

Modify the TmpFS file system

```
$TmpFS=/scratch
```

Add the TmpDisk value for /scratch

The TmpDisk is the size of the scratch in MB , you have to add in the line starting with NodeName

For example for a node with a 3TB disk:

```
$ NodeName=node21 CPUs=16 Sockets=4 RealMemory=32004 TmpDisk=3000 CoresPerSocket=4 ThreadsPerCore=1 State=UNKNOWN
```

Modify a partition definition

You have to modify the line starting with PartitionName in the file `/etc/slurm/slurm.conf` .

Several options are available according to what you want

Add a time limit for running jobs (MaxTime)

A limitation time on partitions allows slurm to manage priorities between jobs on the same node.

You have to add it in the PartitionName line with the amount of time in minutes.

For example a partition with a 1 day max time the partition definition will be:

```
PartitionName=short Nodes=node21,node[12-15]  MaxTime=1440 State=UP
```

Add a Max Memory per CPU (MaxMemPerCPU)

As memory is a consumable resource MaxMemPerCPU serves not only to protect the node's memory but will also automatically increase a job's core count on submission where possible

You have to add it in the PartitionName line with the amount of memory in Mb.

This is normally set to MaxMem/NumCores

for example 2GB/CPU, the partition definition will be

```
PartitionName=normal Nodes=node21,node[12-15] MaxMemPerCPU=2000 MaxTime=4320 State=UP
```

Links

- Related courses : HPC Trainings

License


Institut de Recherche
pour le Développement
FRANCE


cirad


INRA
SCIENCE & IMPACT

WebMaster: Christine & Séb