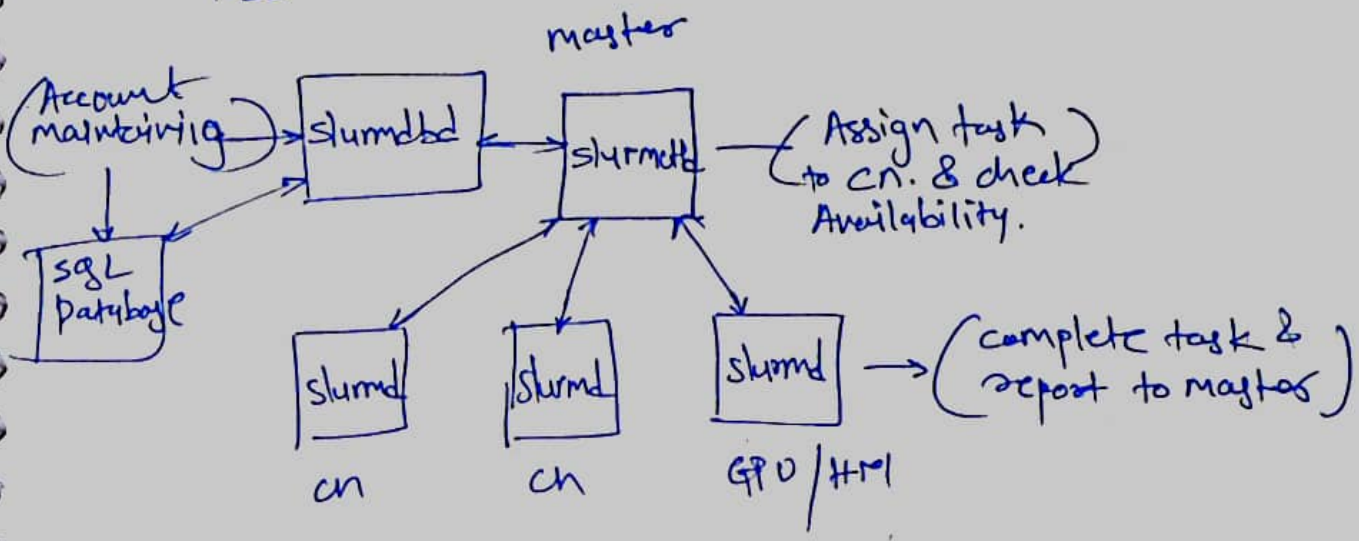
Resource Management & Accounting

- Batch system
- Scheduler
- Resource Manager
  - slurm
  - PBS

SLURM uses:

Backfill Algorithm



slurm - Architecture

Slurm :- Job scheduler, Resourse Manager, Account maintaining.

Munge key :- A symetric key

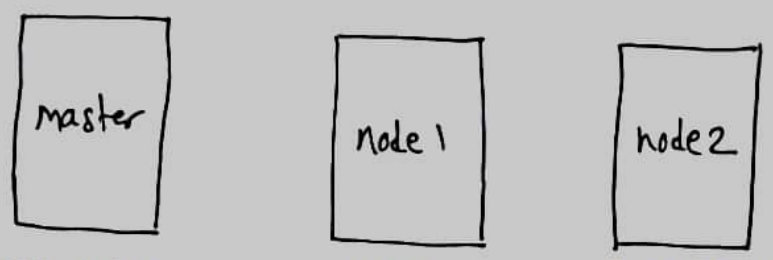
slurmctld - ~~681~~ 6817
slurmdbd - 6819
slurmd - 6818
} Port Number

MUNGE KEY

1. Key Generation
2. Msg Encryption
3. Msg Authentication
4. Access control

Slurm Practical

Requirement :-



master	node 1	node 2
8GB RAM	2GB RAM	2 GB RAM
100 GB HD	50 GB HD	50 GB HD
4 core	2 core	2 core

Master — NAT - 192.168.44.174
Host - 10.10.10.146

Node 1 — NAT - 192.168.44.173
Host - 10.10.10.147

Node 2 - NAT - 192.168.44.172
HOST - 10.10.10.148

Run these commands in all 3 machines.

```
# systemctl stop firewalld.service
# systemctl disable firewalld.service
# vi /etc/selinux/config
   ↳ SELINUX=disabled
# yum install nfs-utils.x86_64
# yum install epel-release
```

In Master Node

```
# systemctl start nfs-server.service
# systemctl enable nfs-server.service
# chmod 777 /home/
# vi /etc/exports
   ↳ /home/ 10.10.10.147(rw,sync,no_root_squash)
     /home/ 10.10.10.148(rw,sync,no_root_squash)
# exportfs -avr
```

on node1 & node2

```
# mount -t nfs 10.10.10.146:/home/ /home/
# df -TH → /home/ is mounted
```

Make passwordless SSH

1. master to client (node1 & node2)
2. node1 to master
3. node2 to master

Run this command in all ~~44~~ 3 machines

ssh-keygen

Now copy public key of master

cat .ssh/id_rsa.pub

↳ copy public key

Now paste the public key of master in node1 & 2

vi .ssh/authorized_keys

↳ paste public key of master which you have copied.

Now add user 'admin' in all 3 machine

useradd admin

Now set password for admin user

passwd admin.

Now do passwordless ssh of all 'admin' user

In admin@master machine

ssh-keygen

ssh-copy-id admin@ node1 host ip address

ssh-copy-id admin@ node2 host ip address

Host file entry

↳ In master machine

vi /etc/hosts

↳ master 10.10.10.146

node1 10.10.10.147

node1 10.10.10.148

rsync /etc/hosts ~~admin~~ root@node1:/etc/hosts.

rsync /etc/hosts root@node2:/etc/hosts.

Now run below comand on all 3 machines

yum install munge munge-libs munge-devel

on master

/usr/sbin/create-munge-key -r

↳ munge key is generated in `/etc/munge'/'

scp /etc/munge/munge.key node1:/etc/munge/

scp /etc/munge/munge.key node2:/etc/munge/

systemctl start munge.service

systemctl enable munge.service.

on client manchines (node1 & node2)

chown munge:munge /etc/munge/munge.ke

systemctl start munge.service

systemctl enable munge.service.

On master

wget https://dawnload.schedmd.com/slurm/slurm-20.11.9.tar.bz2

↳ slurm source code tar downloaded

yum install rpm-build

rpmbuild -ta slurm-20.11.9.tar.bz2

yum install pam-devel python3 readline-devel perl-ExtUtils-MakeMaker mysql-devel

rpmbuild -ta slurm-20.11.9.tar.bz2

~~# yum install gcc~~

On both client machine

yum install pam-devel python3 readline-devel perl-ExtUtils-MakeMaker mysql-devel

on all three machines.

export SLURMUSER=900

groupadd -g $SLURMUSER slurm

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

on master

```
#ll /root /rpmbuild/RPMS/x86_64/
# mkdir /home/rpms
# cd /root/rpmbuild/RPMS/x86_64/
# cp * /home/rpms/
```

on all 3 machines

```
# cd /home/rpms
# yum --nogpgcheck localinstall *
```

Now remove slurmctld & slurmdbd from all client

```
# rpm -e slurm-slurmctld-20.11.9-1.el7.x86_64
        slurm-slurmdbd-20.11.9-1.el7.x86-64.
```

we don't need these packages in node1 & node2

on all 3 machines

```
# mkdir ## /var/spool/slurm
# chown slurm:slurm /var/spool/slurm/
# chmod 755 /var/spool/slurm/
# mkdir /var/log/slurm/
# chown -R slurm. /var/log/slurm
```

On master

touch /var/log/slurm/slurmctld.log
chown slurm:slurm /var/log/slurm/slurmctld.log
touch /var/log/slurm_jobacct.log
touch /var/log/slurm_jobcomp.log

chown slurm: /var/log/slurm_jobacct.log
/var/log/slurm_jobcomp.log

~~vi /etc~~ cp /etc/slurm/slurm.conf.example /etc/slurm/slurm.conf

vi /etc/slurm/slurm.conf

↳ 11. clustername = hpcsa
12. ControlMachine = master.
92. NodeName ------ (comment this line)

on both client

slurm -C

↳ copy all line.

on master

in /etc/slurm/slurm.conf file

93. slurm -C output line of client 1
94. slurm -C output line of client 2

scp /etc/slurm/slurm.conf node1 :/etc/slurm/

scp /etc/slurm/slurm.conf node2 :/etc/slurm/

systemctl start slurmctld
systemctl enable slurmctld

on client nodes.

systemctl start slurmd
systemctl enable slurmd

on master

sinfo

if ~~ideal~~ idle* " is there then you need to restart slurmd service on node1 & node2.

scontrol update node=node1 state=idle
scontrol update node=node2 state=idle

slurmctld -Dvv → to debug slurmctld
sinfo -R → to check unhealthy node

on client

slurmd -Dvv → to debug slurmd service on client

On Master

srun -w node1 --pty /bin/bash → to submit the job node 1

srun -N1 --pty /bin/bash → to submit job on N1 (node No. 1)

scontrol update node=node1 state=down reason=maintenance → to down the node for maintenance

scontrol update node=node1 state=resume
→ resume node1 state to idle from down

vi demosbatch.sh

```
#!/bin/bash
#SBATCH --partition=standard
#SBATCH --job-name=myjob
#SBATCH --nodes=2
#SBATCH --ntasks=2
#SBATCH --cpus-per-task=1
#SBATCH --time=00:05:00
#SBATCH --output=myjob_output_%j.log

date
sleep 3000
```

sbatch demosbatch.sh → submit job

squeue → show the running job ~~into~~.

scontrol show job 5 → job into with details
job id

scancel 6 job id → to concel job 6

sshare → show account details in slurm

Installation of Database

yum install mariadb-server mariadb-devel -y

systemctl start mariadb

systemctl enable mariadb

mysql

#> GRANT ALL ON slurm_acct_db.* To 'slurm'@'localhost'
IDENTIFIED BY '1234' with grant option;

#> ~~SHow~~ SHOW VARIABLES LIKE 'have_innodb';

#> FLUSH PRIVILEGES

#> CREATE DATABASE slurm_acct_db;

#> show databases;

#> quit;

Verify the databases grants for slurm user

mysql -p -u slurm

#> show grants; → show grants for slurm user

vi /etc/my.cnf.d/innodb.cnf

↳ innodb_butter_pool_size = 1024M
innodb_log_file_size = 64M
innodb_lock_wait_timeout = 900

to implement this change you have to shut down database & move logfiles.

systemctl stop mariadb

mv /var/lib/mysql/ib_logfile'? /temp/

systemctl start mariadb

mysql

#> SHOW VARIABLES LIKE 'innodb_butter_pool_size';

Create slurmdb conf. file

vim /etc/slurm/slurmdb.conf

DbdAddr= localhost
DbdHost = localhost
DbdPort = 6819
StoragePass= P34
StorageLoc = slurm_acct_db

chown slurm: /etc/slurm/slurmdbd.conf
chmod 600 /etc/slurm/slurmdbd.conf
touch /var/~~slurmdbd~~log/slurmdbd.log
chown slurm: /var/log/slurmdbd.log

slurmdbd -D -vvv → see slurmdbd log

systemctl start slurmdbd.service
systemctl enable slurmdbd.service
systemctl start slurmctld.service
systemctl enable slurmctld.service.

← Accounting →

vim /etc/slurm/slurmdbd.conf
- AccountingStorageType = accounting_storage/mysql
  StorageHost = localhost
  storageUser = slurm
  storagePass = 1234
  storageLoc = slurm_acct_db

chown slurm:slurm /etc/slurm/slurmdbd.conf
mkdir -p /var/log/slurm
touch /var/log/slurm/slurmdbd.log
vim /etc/slurm/slurm.conf
- AccountingStorageHost = localhost
  AccountingStorageUser = slurm
  AccountingStorageType = accounting_storage/slurmdbd

#systemctl restart slurmctld
#systemctl restart slurmdbd
#systemctl restart mariadb

#mysql

#> create user 'slurm'@'master';
#> grant all on slurm_acct_db.* To 'slurm'@'master';

#>show databases;

#>exit

#sacctmgr → workload manager, view & modify slurm account information

#: add cluster Accounting-Cluster

#: add account slurm-accounting

#: add user ~~add~~ admin account= slurm-accounting
present user

#: create qos slurm-qos maxwall = 2-00:00:00

(qos = Quality of service)

#:show qos

#:exit

#sshare → show accounting details.