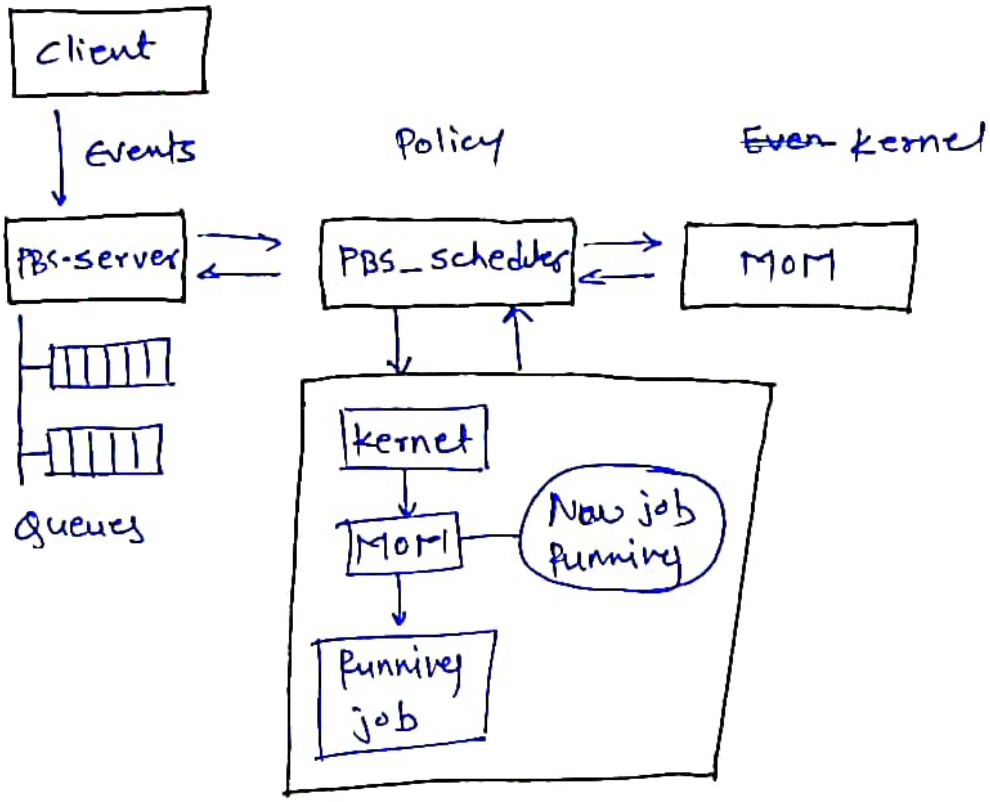
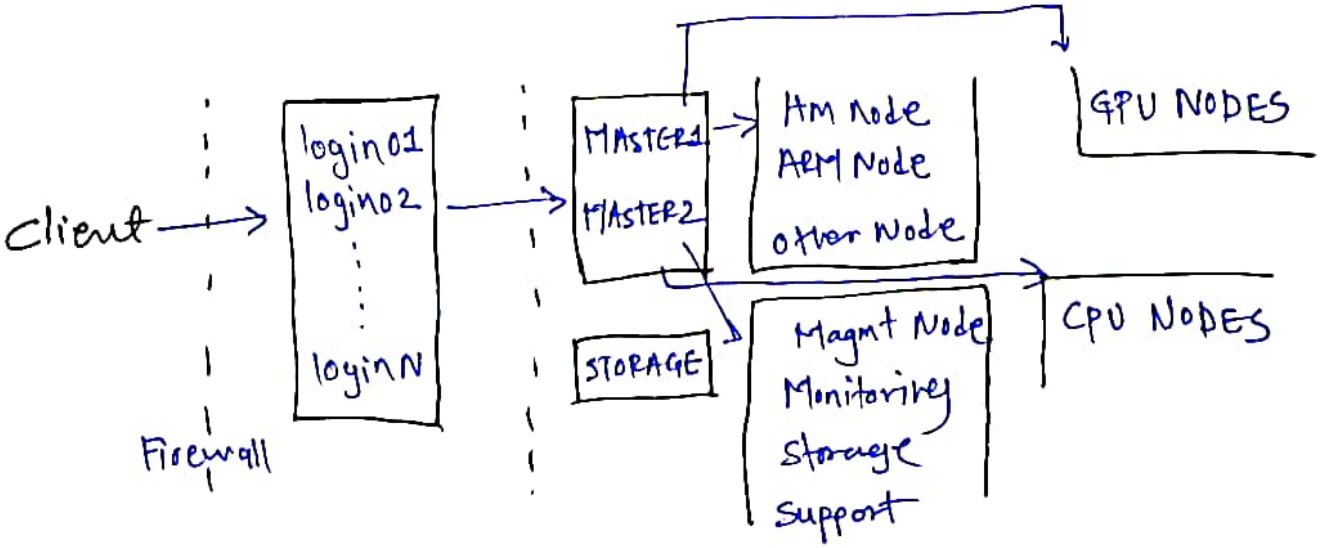
# Open PBS (Portable Batch System)

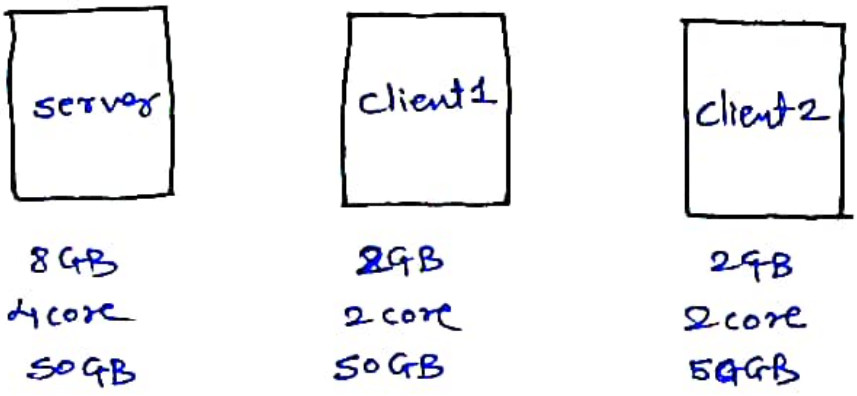
- PBS is the name of computer software that performs job scheduling. its primary task is to allocate computational tasks i.e batch jobs, among the available computing resources.



# Practical Open PBS

## Required Machines




Server :- NAT - 192.168.44.144 Hostname - pbs-master
Host - 10.10.10.133

Node1 :- NAT - 192.168.44.142 Hostname - pbs-node1
Host - 10.10.10.131

Node2 :- NAT - 192.168.44.143 Hostname - pbs-node2
Host - 10.10.10.132

## On all machines

```
# systemctl stop firewalld
# systemctl disable firewalld
# vi /etc/selinux/config
   ↳ SELINUX=disable
# useradd hpcsa
```

## On Master

# vi /etc/hosts

↳ 192.168.44.144 pbs-master
192.168.44.142 pbs-node1
192.168.44.143 pbs-node2

# rsync /etc/hosts root@pbs-node1:/etc/hosts
# rssync /etc/hosts root@pbs-node2:/etc/hosts

# ssh-keygen

# ssh-copy-id root@pbs-node1
# ssh-copy-id root@pbs-node2

# yum install git
# git clone https://github.com/openpbs/openpbs.git

# yum group ~~install~~ install "Development Tools"
↳ if ~~all ready~~ already install check install groups by command # yum group list

# cd /openpbs/
~~# rpmbuild -ba openpbs.spec~~
# cd

# mv /root/openpbs/ /root/openpbs-23.06.06

# tar -cvf /root/rpmbuild/SOURCES/openpbs-23.06.06.tar.gz ~~/root/~~openpbs-23.06.06~~/~~*

# cd openpbs-23.06.06

# rpmbuild -ba openpbs.spec → if ~~a~~ error comes install these 10-12 packages ~~in rpms~~ which mention in error

# cd

# cd rpmbuild/RPMS/x86_64/

# yum install openpbs-server-23.06.06-0.x86_64.rpm

# chmod 4755 /opt/pbs/sbin/pbs_iff /opt/pbs/sbin/pbs_rcp

# systemctl start pbs.service

# systemctl enable pbs.service

# . /etc/profile.d/pbs.sh

# qstat -B → Display server status

## On Node1

# yum install git

# git clone https://github.com/openpbs/openpbs.git

# cd openpbs/

# ./autogen.sh

# yum install autoconf automake libtool

# ./configure

# mkdir /opt/pbs

# ./configure --prefix=/opt/pbs/

if error: "Required header file is missing" occurs then run this command.

"# yum install openssl-devel libXt-devel libtool-ltdl-devel hwloc-devel libedit-devel libical-devel ncurses-devel postgresql-devel postgresql-contrib python3-devel tcl-devel tk-devel zlib-devel expat-devel"

# ./configure --prefix=/opt/pbs/ (install libraries in /pbs directory)

# make ⟶ recompilation

# make install

# . /opt/pbs/etc/pbs.sh

# qstat ⟶ if command not found then do following steps

# chmod +x /opt/pbs/etc/pbs.sh

# export PATH=${PATH}:/opt/pbs/bin

Now copy file "/etc/pbs.cont" file from master node to node1 "/etc/" path

# scp /etc/pbs.cont root@pbs-node1:/etc/

# vi /etc/pbs.cont

↳ PBS_START_SERVER=0
PBS_START_SCHED=0
PBS_START_COMM=0
PBS_START_MOM=1

# systemctl start pbs
# systemctl enable pbs
# qstat -B

Mount "/root/rpmbuild/RPMS/x86_64/" to node2

on master & node2

# yum install nfs-utils *
# systemctl start nfs
# systemctl enable nfs.

on master

# chmod 777 /root/rpmbuild/RPMS/x86_64/
# vi /etc/exports
↳ /root/rpmbuild/RPMS/x86_64/ node2 ip 192.168.44.143 (rw,sync, no_root_squash)

# exportfs -avr

# showmount -e → check whether path is exported or not

On node 2 # mkdir /root/pbs

# mount -t nfs 192.168.44.144 :/root/rpmbuild/RPMS/x86_64/ /root/pbs

# df -h → check /x86_64/ is mount or not

# cd /pbs/

# yum install openpbs-execution-23.06.06-0.1.x86_64.rpm

On ~~#~~ node 1

#rsync /etc/pbs.conf root@pbs-node2:/etc/

On node 2

# systemctl start pbs.service

# systemctl enable pbs.service

# vi /var/spool/pbs/mom_priv/config

↳ $logevent 0x1ff

~~#~~$ clientname pbs-node2 (comment this line)

$ restrict_user_maxsysid 999

#systemctl restart pbs.service

## On Node 1

# vi /var/spool/pbs/mom_priv/config
↳ $logevent 0x1ff
# $clientname pbs-node1 (comment this line)
# $restrict_user_maxsysid 999 (com
# systemctl restart pbs.service.

## On master

# vim /var/spool/pbs/server_priv/nodes
↳ pbs-node1 np=1
pbs-node2 np=1

# qmgr

Now we get Qmgr shell

#: create node pbs-node1
#: create node pbs-node2
#: exit

# pbsnodes -a → list of all nodes & their attribute

Now Add master as client

# vi /etc/pbs.conf ...
↳ PBS_START_MOM = 1

# systemctl restart pbs.service

# vim /var/spool/pbs/mom_priv/config

↳ $logevent 0x1ff

# $ clientname pbs-master

# restrict_user_maxsysid 999

# systemctl restart pbs.service

# vim /#var/spool/pbs/server_priv/nodes

↳ pbs-master np=1

# qmgr

#: create node pbs-master

#: exit

# pbsnodes -a

↳ you can see 'pbs-master' is added as node

Now switch to normal user

# su - hpcsa → switched user to hpcsa

# qsub -I → The job is queued & scheduled as any PBS batch job

Here we get interactive terminal of 'node2 hpcsa' user ~~test~~

# Queue Creation in pbs

## on master

```
# qmgr
#: create queue pbs-queue queue_type = execution
#: set queue pbs-queue enabled = True
#: set queue pbs-queue started = True
#: ~~queue~~ set ~~queue~~ queue pbs-queue resources_default.
                                        nodes = 1
#: set queue pbs-queue resources_default.walltime = 360
#: set server default_queue = pbs-queue
#: P S ⟶ print server
#: exit
# su - hpcsa
# qsub -I
```

Now open new terminal & run this command

```
# qstat -f
```

you can see the job is in Running state & queue name - pbs-queue

# Assign queue to particular node

On master

```
# qmgr
#: set node pbs-node2 queue = pbs-queue
```

Now run the job

```
# su - hpcsa
# qsub -I
```

```
# qmgr
#: print node pbs-node2
```

Here you can see queue is assign to pbs-node2