You have been assigned to run a campus commodity cluster. There are 4 groups on campus, each with 2 users. All groups are already on the system submitting jobs.

**Introduction**

- There are 7 nodes available for you to use for this practical
  - The head node is "Login"
  - The compute nodes are Compute-[0-1]
- These nodes have the following specifications:
  - 2 CPUs
  - 512MB of memory
  - Centos 7 latest
- The home directories are NFS mounted from the scheduler nodes
- Root has ssh keys from the scheduler node to all other nodes
- There are 8 users and 4 groups set up for the practical
  - There are ssh keys between all nodes for these users

**Familiarize yourself with your cluster**
- Slurm is installed with the latest version
  - slurm 18.06.6
- The cluster name is 'lcicluster'
- There is currently 1 partition enabled named 'normal'
  - This has a 20 minute time limit
- Your cluster is currently being used by "users" from a university
  - Jobs will be continually submitted by "users" throughout the session
  - We will tune the cluster to make it run more efficiently
- Submit an interactive job
  - **$ salloc -N 1**
    **$ srun hostname**
- There is currently a 20 minute limit on the single normal partition
- MariaDB is enabled with Slurm for accounting although nothing is enforced

**Exercise 1: Enable Fairshare**

Oh no! A professor Bob is complaining about the fairness of the cluster. They only run 4 core jobs a few times a week and other are running a ton of single core jobs. Enable fairshare to give all users an equal share.

- The documentation for enabling fairshare is located here:
    - https://slurm.schedmd.com/priority_multifactor.html

- The following are the parameters that should be modified in the /etc/slurm/slurm.conf

    - `PriorityType=priority/multifactor`
    - `PriorityDecayHalfLife=0-00:10:00`
    - `PriorityCalcPeriod=1:00`

- We need to apply a weight to each factor in the multifactor equation

    - `PriorityWeightAge=1000`
    - `PriorityWeightFairshare=10000`
    - `PriorityWeightJobSize=1000`
    - `PriorityWeightPartition=1000`

- Reread the configuration
    - **`scontrol reconfigure`**

- Verify job priority is being modified properly
    - **`scontrol show config`**

- Let's check to see if anyone's priority has changed
- We can now read the shares
    - **`sshare`**

**Exercise 2: Modify Fairshare for Groups and enable accounting**

The professors have talk with each other and they want to share the machine equally between groups.

- We need to enable accounting for the groups
- Add the cluster to the accounting
  - `sacctmgr add cluster lcicluster`
- Next add the accounts (see table below)
  - `sacctmgr add Account <account>`
- We know need to associate the users with the accounts
  - `sacctmgr add user <user> Account=<account>`
- We need to also make each account the user's default
  - `sacctmgr add user <user> DefaultAccount=<account>`
  - Verify all associations
    - `sacctmgr show associations`
- Reread the configuration
  - `scontrol reconfigure`
- Wait a bit and we will view the shares
  - `sshare`

| Account | Users |
|---------|-------|
| bio | kara |
|  | tim |
| physics | bob |
|  | mary |
| math | rae |
|  | david |
| astro | joe |
|  | alice |

**Exercise 3: Issues with Priority**

Professor Bob in the physics is back to having issues getting high priority jobs through. He has
a conference coming up and needs the use of the entire cluster. Create a reservation that
allows his group to use the entire cluster for the next 30 minutes. Then create a high partition
with all the nodes that has a higher starting priority.

- Create the reservation for his group
    - `scontrol create reservation account=astro duration=30 nodecnt=2`
    - `scontrol create reservation starttime=now users=bob,mary duration=30 nodecnt=2 partition=normal`
    - Verify the reservation exists
        - `scontrol show res`

- Create a new partition named high
    - Copy the partition line in the /etc/slurm/slurm.conf for normal and rename it high and change the default
        - `PartitionName=high Nodes=Compute-[0-1] Default=NO MaxTime=0-20:00 State=UP`
    - Reread the configuration and verify the partition exists
        - `scontrol reconfigure`
        - `sinfo`

- Create a QOS for each partition and set a priority for high
    - `sacctmgr add qos <partitions>`
    - `sacctmgr modify qos high set priority=1000`
    - `sacctmgr show qos format=name,priority`

- Give QOS a fairshare weight
    - Add the following to the /etc/slurm/slurm.conf
        - `PriorityWeightQOS=10000`
    - Assign the QOS to each partition
        - Add the following to each Partition line in /etc/slurm/slurm.conf
            - `QOS=<qosname>`
        - `scontrol reconfigure`
    - Check jobs and verify the qos is applied

**Exercise 4: Limiting Groups with Accounting**

The IT department has decided they want to sell the resources to each department to help fund the machine. Limit each project to 5 cpu hours per account. Modify partitions so that the high priority is double the cost of the normal partition

- Set all groups to 5 cpu hours
  - `sacctmgr modify account <accountname> set GrpCPUMins=300`

- Verify the accounts
  - `sacctmgr show Association`

- We need to now make sure everything we are doing is enforced
  - Add the following to your /etc/slurm/slurm.conf
    - `AccountingStorageEnforce=associations,limits,qos`

  - `scontrol reconfigure`

- Change the change rate of the high queue
  - `sacctmgr modify qos high set usagefactor=2`

**Exercise 5: Enable Preemption for a Low Partition**

The users would like a low queue in which they can submit jobs that might be preempted but get changed less as well. Make sure these jobs only backfill and are half the cost of normal jobs.

- Copy the partition line in the /etc/slurm/slurm.conf for normal and rename it low and change the qos
  - `PartitionName=low Nodes=Compute-[0-1] Default=NO MaxTime=0-20:00 State=UP QOS=low`

- Create the QOS for the low partition
  - `sacctmgr add qos low`
  - `sacctmgr modify qos low set priority=0`
  - `sacctmgr show qos format=name,priority`

- Make sure the QOS has the Flag NoReserve to make sure it only backfills
  - `sacctmgr modify qos low set FLAGS=NoReserve`

- Bump the normal partition's priority to 100 to be in between the low and high queue
  - `sacctmgr modify qos normal set priority=100`

- Change the charge rate of the low queue
  - `sacctmgr modify qos low set usagefactor=0.5`

- `scontrol reconfigure`

**Exercise 6: Setup a Singularity for a User**

Biology wants to use a Docker image for their genomics project. They need Ubuntu. Follow the instructions to setup an Ubuntu image for them. Singularity has already been installed on the compute nodes and scheduler.

- Create an image
  - **`singularity build ubuntu.sif docker://ubuntu`**
- Get a shell inside of the container
  - **`singularity shell ubuntu.sif`**
    - **`cat /etc/os-release`**
    - **`id`**