In this exercise, we will work through some of the basics of using a cluster and running parallel programs. Initially, make a bookmark at the lab notes on your browser. They are in the Handouts section of the class home page at http://cs.boisestate.edu/~amit/teaching/430/CS430.html.

1. Login in to the onyx server (or any workstation in the ENGR 213-214 MetaGeek lab). Start up a console window. Make sure you have enough disk space to do this lab. Check with the command:
   
   quota -v

   If needed, make room in your account first by cleaning up!

2. Create a directory cs430 (or cs530) for the class. Checkout the MPI examples folder using the following command:
   
   svn checkout https://cs.boisestate.edu/repos/csdept/courses/cs430/lab/MPI

3. Login to the master node, onyx, using the command ssh -Y onyx.

4. **SSH setup.** Type in the following command to setup password less login in the lab from any machine to another machine.
   
   ssh-setup

   You only need to run the above command once and from now on you do not need a password to log in to onyx or any other node from a node in the lab.

5. **Cluster monitoring.** Here we will try a few cluster monitoring commands. Try the following commands: cchk, cdate, cmips, cfree, cnodes, cdisks, and ctemp.

6. **Cluster control with parallel shell.** The pdsh shell is a simple parallel shell that is useful for doing tasks across the cluster. The utility dshbak helps with sorting the output from pdsh. Try the following commands: pdsh -a date
   
   pdsh -a uptime

   pdsh -a ps augx | grep pbs | grep -v grep

   pdsh -a -x node00 ls -l /etc/passwd

   pdsh -w node[16-26] who

   pdsh -w node[16-26] who | dshbak

   pdsh -a date | dshbak -c

7. **Compiling a MPI program.** Change directory to the parallel_sum directory. Build the spmd_sum program using the command:
   
   make

8. **Acquiring nodes for running programs.** Acquire 4 nodes using the command:
   
   pbsget -4
9. Check the nodes allocated to you with the command `qstat -n`.

10. Run your MPI program with 8 processes as follows. We are using two cores per node in this example (there are four cores per node).

```
mpiexec -n 8 spmd_sum_3 1000000
```

11. Try running the program with different number of processes.

```
mpiexec -n 4 spmd_sum_3 1000000
mpiexec -n 16 spmd_sum_3 1000000
```

12. Always exit out of the PBS shell after a pbsget with the `exit` command. You can try that now. It should change the prompt.

```
exit
```

13. Now check the `sum.pbs` batch script. Update it to reflect your folder structure. Submit that to run in batch mode. You don’t need to acquire nodes for this purpose. Use the command shown below. It names the job name as your login name.

```
qsub -N test1 -m be sum.pbs
```

Wait for two emails from the PBS telling you that the job has started and then a little bit later one that tells you that the job has finished. If you forward your email from onyx, then you will see the emails at your regular email address. Now check the output. It should be in the files `test1.o<jobid>` and `test1.e<jobid>`. Check those files. See the lab notes for how to run a job at a certain time at night.

14. Check how much cluster resources you used with the command:

```
pbs_usage
```

The above command prints your usage in the current month. Using the command line option `-t` will give you the total since the cluster was installed.