

Spring 2007: CS 633/733 Grid Computing
Homework-1

200 points. Due Jan 29, 2007 before class. Individual Work Only.

Part-A: Running parallel applications on clusters using a batch system

1. Login to everest.cis.uab.edu using the user id and password provided in class. From one of the Vulcan machines use the command “ssh <userid>@everest.cis.uab.edu” to login or use an SSH client. You can look at <http://www.cis.uab.edu/cs633/software> for instructions and demo on how to login to the cluster.
2. Download the file *HPL.dat* and the SGE submit script *sge.script* for Everest from <http://www.cis.uab.edu/cs633/software>. Submit the SGE script using the command **qsub sge.script**. Check the status of your job using the command **qstat -u <userid>**. When your job completes look at the output in the file *sge.script.oXXXX*, where XXXX is the jobid (if the job completes successfully then you will see “End of Tests” at the end of the file). You can look at the man pages for qsub and qstat if you are interested in more details. The SGE script submits the HPL benchmark.
3. Execute the HPL benchmark for different number of processors and problem sizes given in the table below (ignore P=Q=8 for Everest, it has only 26 nodes available for execution). Note down the best times and GFlops numbers and plot these on two separate graphs.

Problem size N = 10000					
P	Q	Time (seconds)	GFlops	Speedup	Efficiency
1	1				
2	2				
3	3				
4	4				
8	8				

N	P	Q	Time (seconds)	GFlops	Speedup	Efficiency
10000	1	1				
20000	2	2				
30000	3	3				
40000	4	4				
80000	8	8				

In order to execute the different cases, create a template for the file *HPL.dat* (sample template *HPL.dat.template* is available at: <http://www.cis.uab.edu/cs633/software>). Using the template, generate a new *HPL.dat* using something like this:

```
sed -e 's/$N/10000/' -e 's/$P/1/' -e 's/$Q/1/' HPL.dat.template > HPL.dat
```

4. Repeat steps (1), (2), and (3) for the host olympus.cis.uab.edu.
5. Comment on the performance plots for both machines and the two different cases.

Part-B: Setting up SSH keys for authentication without passwords

6. Read the man pages for ssh-keygen and figure out how to login from one of the Vulcan machines to Everest and Olympus without typing the password. Document all the steps performed on each machine.

Part-C: Job submission to the cluster from a remote host

7. After completing question (6), write a script (say, *runremote.sh*) that will execute on one of the Vulcan machines and perform the following operations (you can use any scripting language available on Vulcan):
 - a. create an SGE submit script (use an SGE template and command-line arguments to qsub)
 - b. copy the SGE submit script and the input file *HPL.dat* to the specified hostname using SCP
 - c. submit the SGE script for execution on the specified hostname using SSH
 - d. wait for the job to complete (Note: When you submit the SGE script it will just return a jobid, how will you know when the job actually completes? – use the class discussion forum to discuss this issue, I will post possible solutions on the discussion forum)
 - e. copy the output file *HPL.out* from the specified hostname to Vulcan after the job completes using SCP.

The script *runremote.sh* must accept the following arguments (either as command-line arguments or by prompting the user to enter appropriate values):

- a. Name of the executable to be used in the SGE submit script (with full path)
- b. Any arguments that must be passed to the executable (problem size and number of processes)
- c. Number of processes to be used with **-pe single** option in the SGE script file
- d. Hostname where the SGE script must be submitted

Part-D: Short Answer Questions (no need to implement this)

8. Describe how login to Everest and Olympus works without passwords after setting up the keys.
9. If there are dozens of machines on which you have to submit a job, will this approach of setting up shared keys work well? What are the drawbacks with this approach?
10. If each of these machines had a different batch system then what changes are required to your script?

Submission:

Create a tar file with all the scripts and documentation and upload the tar file to WebCT. Look at the slides on UNIX shell programming at <http://www.cis.uab.edu/cs333/fall2006/> if you need help with shell scripts.