HPEDSI Course - “Introduction to High Performance Computing”, Homework 2

* Please use Neches, or your own Linux machine, to run your commands/scripts and generate the results. Use VPN if you connect to Neches from home.
* All files used as input are in the intro2scheduler folder that you downloaded from the course Moodle site.
* Fill your solutions in this file, as well as the output from your commands. If the output is standard output, you can paste the output along with your solution; if the output are files, please attach them. The solution is often not unique, the more details you provide, the better.
* Create a compressed file named as “Intro2LinuxHW2\_yourname.zip” from the solution file (this file, preferably word or pdf file) and the output files. Please make sure the zip file is not empty and upload it to the Moodle site.

1) Write the commands and answers for the following problems.

The user pzhang5 planned to submit 100 long jobs on Neches, named as myjob001, myjob002, ..., myjob100. Each of the job requires 2 nodes (16 processors) for 10 days.

a)  How can he find out the partitions listed on the Neches cluster? Please interpret the output and recommend a partition to him for his jobs.

sinfo -all

b)  His jobs are basically using MATLAB to reverse large matrices. Which command can he use to see available software installed on the Neches cluster? How could he use MATLAB in the SLURM script (suppose his matlab script is mymatlab.m)?

module avail

#!/bin/bash

#SBATCH -n 8 -N 1

#SBATCH -t 0:30:05

#SBATCH --mem=10gb

#SBATCH -o matlabjob.out -J matlabjob

#SBATCH --mail-type=all

#SBATCH --mail-user=ksehta@uh.edu module add matlab

Run Command:

time matlab < mymatlab.m

c)  Suppose pzhang5 submitted 100 long jobs on Neches, named as myjob001, myjob002, ..., myjob100. How could he check the status of his jobs?

Sequence -u $pzhang5

d) Using the command that you provided in (c), pzhang5 can find that there are 10 of them running and the rest are pending in the queue. Could you please tell him possible reasons why his jobs are in the queue?

1. You or your group are at the maximum processor count or running job count and your job is being held.
2. Your job is requesting specialized resources, such as GPU nodes or large memory nodes or certain software licenses, that are in high demand and not available.
3. Your job is requesting a lot of resources. It takes time for the resources to become available.
4. Your job is requesting incompatible or nonexistent resources and can never run.
5. Job is unnecessarily stuck in batch hold because of system problems (very rare).

e) e) pzhang5 plans to kill all his jobs in the pending state. What is the command(s) to do so? (hint: use scancel --help to check usage)

scnacel -u $pzhang5 -t PENDING

2) Your task is to improve the performance of a parallel application by using more compute resources. The application supplied to you is a molecular dynamics simulation package (NAMD). The package is installed on Neches as a module. In the intro2scheduler/SLURM directory, find the SLURM job script to submit jobs to SLURM (fix potential problems in the script if you come across any). Using the sbatch command and job scripts, you need to benchmark the scalability of the software by comparing the simulation time of the same job run on the following resources.

|  |  |  |
| --- | --- | --- |
| **JOB** | **RESOURCE** | **NO OF MPI PROCESS** |
| 1 | 1 NODE | 1 |
| 2 | 1 NODE | 2 |
| 3 | 1 NODE | 4 |
| 4 | 1 NODE | 8 |
| 5 | 2 NODE | 16 |
| 6 | 4 NODE | 32 |
| 7 | 8 NODE | 64 |

Make sure you edit the script(s) so that

• The output files should have a name that is indicative of the job that produced them.

• Job names should be different and convenient for you to check the job status

• Your own email address is provided to receive notifications.

Upload the batch job script(s) that you used along with output files (there should be 7x2 =14 output files in total) from all runs. Don’t forget the discussion of the results.

Hints:

i) Check the time used by the application at the end of the output files

ii) Plot the scaling curve: speedup vs. number of processes