ECE 759
High Performance Computing for Engineering Applications
Assignment 1
Due Monday 9/16/2024 at 23:59 PM

Submit responses to tasks which don't specify a file name to Canvas in a file called assignment1.pdf (choose one of the formats). Submit all plots (if any) on Canvas. Do not zip your Canvas submission.

All *source code files* should be submitted in the HW01 subdirectory on the main branch of your GitHub repo (which should be named repo759, as instructed in class). Please use the name HW01 exactly as shown here (both in terms of capitalization & name). Other names like hw1, hw01, HW1 will not be recognized by the repo scripts. The HW01 subdirectory should have no subdirectories. For this assignment, your HW01 folder should contain task4.sh and task6.cpp.

All submissions will be graded on *Euler*. Sometimes, code behaves differently on your computer and on Euler, so it is recommended that you test on *Euler* before you submit. Per CAE policies: do not run computational jobs directly on the head node; and do not launch interactive jobs on *Euler*. Use sbatch to submit your jobs and use the resources responsibly.

Use Piazza for resolving your questions and leaving comments on assignments. Read this document to help you use Piazza correctly, efficiently and wisely.

Finally, remember to invite the TAs and the instructor as collaborators to your GitHub repo. That will allow us to clone your repo and grade your work. You can find their GitHub account on the course Canvas page.

Please submit clean code. If you have a bit of time, learn how to use a formatter like clang-format.

Specify your GitHub link here: https://github.com/Rob12312368/repo759/tree/main/HW01

Note that your link should be of this format: https://github.com/YourGitHubName/repo759/HW01

1. (a) Read the files timing.md and slurm usage.md from the Assignments/general directory of the ECE 759 Resource Repo. These documents will come into play in almost all your assignments throughout the semester.

   (b) Read the hw repos.md file in the same directory. This is very important and must be done in order for you to turn in all the assignments for ECE 759.

   (c) At least skim workflow.md. This contains a quick guide for effectively working between your local computer and *Euler*. This was also discussed in class.

   NOTE: For this problem, when you are ready to state this in good faith, just say "I went through a) through c) and understand how to time code, how to submit my assignments with git, and what the recommended workflow is when it comes to working on my assignment".

I went through a) through c) and understand how to time code, how to submit my assignments with git, and what the recommended workflow is when it comes to working on my assignment.

2. Write one line of bash code for each of the following sub-tasks (assume that all the files and directories mentioned exist). The purpose of this task is to get you familiar a bit with the Linux command line.

   a) Change the current directory to a subdirectory called somedir

   b) Print out to the terminal the contents of a file called sometext.txt. The file exists in the current directory.

c) Print out to the terminal the last 5 lines of a plain text file called sometext.txt. The file exists in the current directory.

d) Print out to the terminal the last 5 lines of *each* file that ends in the extension .txt and lives in the current directory

e) Write a for loop which prints each integer from 0 to 6 (including 0 and 6).

a) cd somedir

b) cat sometext.txt

c) tail -n 5 sometext.txt

d) for file in *.txt; do tail -n 5 $file; done

e) for i in {0..6}; do echo $i; done

3. The purpose of this task is to get you familiar using *Euler*. On *Euler*, using the module command, answer the following questions.

a) Are there any modules loaded (module list) when you log in on *Euler*?

b) What version (version number) of gcc is available to you without loading any modules?

c) List all cuda modules available on *Euler*.

d) List one other piece of software that has a module on *Euler* and write one sentence about what it does. (If you aren't familiar with any of the other software, google one up and write a sentence about it.)

a) No modules loaded
b) 14.1.1
c) nvidia/cuda/10.2.2   nvidia/cuda/11.6.0   nvidia/cuda/12.1.0
   nvidia/cuda/11.0.3   nvidia/cuda/11.8.0   nvidia/cuda/12.2.0
   nvidia/cuda/11.3.1   nvidia/cuda/12.0.0   nvidia/cuda/12.5.0
d) anaconda(mini): It is a python package manager like pip. Also, it lets you create virtual environments and manage them. Usually, it comes along with a few popular packages when installed, so it is quite convenient.

4. Write a bash script called task4.sh with a Slurm header which asks for

• 2 CPU cores

• A job name of FirstSlurm

• An output file called FirstSlurm.out

• An error file called FirstSlurm.err

and runs a single command to print the hostname of the machine (compute node) running the job. This job must be submitted by running sbatch task4.sh on *Euler*'s head node.

5. Research some useful Slurm tools (one sentence responses):

   a) In what directory does a Slurm job on *Euler* begin execution? You may run some jobs in different directories to check this.

   b) Explain what SLURM JOB ID is in the environment of a running Slurm job.

   c) How would you track the status of job(s) run by yourself? Assume that the job(s) have not been completed yet.

   d) How would you cancel a job submitted by yourself? Assume that the job is still in the queue.

   e) Explain what the following script header line specifies: #SBATCH --gres=gpu:1 f) (Optional) Explain what the following script header line specifies: #SBATCH --array=0-9

a) Usually, it runs at the directory that the task is sent from.

b) It is a unique identifier for a job, so that we can do some operation while it is waiting in the queue.

For example, we can cancel it while waiting.

c) We can use "squeue" and check the "ST" column.

d) First "squeue" to find the job id, and "scancel {jobid}" to cancel the job.

e) It is asking for one gpu to run the task.

f) It means running the same job 10 times with id ranging from 0 to 9.

6. Write a C++ program called task6.cpp that:

   a) Takes a command line argument N. (If you are confused about command line arguments, it may be helpful for you to read this)

   b) Prints out each integer from 0 to N (including 0 and N) in ascending order with the printf function.

   c) Prints out each integer from N to 0 (including N and 0) in descending order with std::cout.

For each printing process, the integers should be separated by spaces on a single line ending in a newline.

   • Compile command: g++ task6.cpp -Wall -O3 -std=c++17 -o task6

   • Run command (do not run this on Euler's head node; do it through Slurm): ./task6 N

   • Expected output (followed by a newline):
     0 1 2 3 · · · N
     N · · · 3 2 1 0

   • Example expected output for N = 6 (followed by a newline):
     0 1 2 3 4 5 6
     6 5 4 3 2 1 0