CITS3402 and CITS5507 High Performance Computing
Laboratory Sheet 2 - How to use Setonix

How to login to Setonix:

You should read this page from Setonix documentation. The instructions are briefly as follows:

- You have to `ssh` to Setonix:

  ```
  ssh -l your_user_name setonix.pawsey.org.au
  ```

- Move to the `scratch` area:
  `cd $MYSCRATCH`

When you login, you are at the login node of Setonix. The login node is a relatively simple node with a few cores and you are not supposed to run your jobs in the login node. You should first change to your working directory to the `scratch` area. This is already set in your profile and the `cd $MYSCRATCH` command takes you to the scratch are, e.g., for me `/scratch/courses0101/adatta`.

All your files will be stored here, and you will submit jobs and get the output from your job here. I would strongly suggest you learn the vi editor for writing your programs and batch files (see below). Here is a tutorial. It takes only a few commands to write text documents in `vi`. However, if you wish, you can write your code using other editors in your local computer and transfer those to your directory in Setonix. I am giving details below.

All jobs in Setonix are submitted using batch files. This file stores all information for compiling your source code, requesting resources (for example, the number of cores) and any other information like required libraries. We will not require any external libraries, and the batch file is quite simple for us.

Creating a batch file:

You have to create a batch file (you can give any name) called `myscript.sh`. This file looks like the following (this is a minimal batch file):

```
#!/bin/bash

#SBATCH --nodes=1
#SBATCH --ntasks=28
#SBATCH --partition=work
#SBATCH --account=courses0101
#SBATCH --mem=4G
#SBATCH --time=00:01:00

cc -o test -fopenmp  ./test.c
srun ./test
```

The first line specifies the shell. The second line specifies the number of Setonix nodes you need. We will need only one node for openMP programs (We will need multiple nodes later for MPI). The next line indicates how many cores you need. Each Setonix node has 28 cores. You can run at most 28 independent threads in one CPU, however, we will experiment with more threads later. The next line indicates the partition we will use. You can use the `work` partition. The next line indicates our account, which is `courses0101`, this is needed for accounting, e.g., how much resources we are using for the entire class. The next line indicates how much RAM we need. 4GB will be sufficient for us initially, you can find out the maximum amount of RAM available per node from the documentation, and you can request that maximum amount. The next line tells the scheduler how much time you need for your job. Here I am running a small program so I have asked for 1 minute. You can find out from the documentation what is the maximum time you can request. You should be aware that your job will be delayed if you request more resources like number of nodes or time. Usually the scheduling time does not vary much if you request more cores on a single node. As we have mentioned above, we need only one node for shared memory programming, but you should try to estimate the required time judiciously.

You submit the job by:
`sbatch myscript.sh`

And you will get a prompt like:
`Submitted batch job 41048` (some number like this). You will get an output file `slurm-41048.out` once your job completes. This file will also contain

all error messages, either errors in your batch file, compilation errors, or run-time errors (41048 will be replaced by your job number).

Now run the very first opemMP program from the lecture slides (or any other openMP program) and check the output.

How to transfer a file to Setonix from your local machine:

You can use the secure copy protocol or scp. Assume that your local file name is test.c and you are currently in the directory where the file is located (you have to otherwise provide the path of the file. The following command executed in your computer will transfer the file to your directory in Setonix (I am using my directory adatta, you have to use your user name).

```
scp test.c adatta@setonix.pawsey.org.au:/scratch/courses0101/adatta
```

You have to provide your password in Setonix after executing this command. Similarly, you can transfer files from your directory in Setonix to your local directory. Assume that I am transferring the file test.c from my directory in Setonix to my local directory

```
scp adatta@kaya.hpc.uwa.edu.au:/group/cits5507/adatta/test.c .
```

The last '.' indicates your current directory in your local machine. You have to again provide your password.

Amitava Datta
August 2024