Practical session 2: Running a serial job

1. **On the CSF**, make sure you are in the `mace-course` folder by using the `pwd` command (seen in ex1).

   0). If you didn't get the training files in exercise 1, run the command:

   ```
   module load training/mace
   ```

   1). If you are not in the folder, then move to it using the `cd` command from exercise 1.

   ```
   cd ~/mace-course
   ```

   The ~ is Linux short-hand meaning "your home directory" - it saves you a lot of typing! Remember, the "home" directory is your private directory, a bit like "My Documents" on Windows.

2. With the command `ls` you should be able to see several folders. We are going to run a job in found in the `python-example` folder. Let's go into that folder:

   ```
   cd python-example
   ls
   ```

3. This exercise is going generate the famous Mandelbrot set. Look at the contents of the jobscript:

   ```
   cat mandelbrot.txt
   ```

   Reminder of what it means

   - The first line indicates there will be commands to run using the BASH Linux shell.
   - Next, we specify various job requirements for Slurm – it will run the job when they become available.
   - The remaining two lines (beginning with "module" and "python") are the commands we want the job to run. We'll cover the "module" line later in the course. For now, hopefully you can see that this job will be running a python script named `mandelbrot.py`. It uses the `python` command to do this.

4. Now "submit" the job to the batch system with this command:

   ```
   sbatch mandelbrot.txt
   ```

   Note: If you are trying this outside of a taught course session, you should remove the "reservation" line in the jobscript, or submit using:

   ```
   sbatch --reservation="" mandelbrot.txt
   ```

   You will be told the job has been submitted and what its **JobID** is.

5. Check the job:

   ```
   squeue
   ```

   Can you see the **JobID**? Every 10 seconds or so type `squeue` and you should see the status change from **PD** (pending) to **R** (running).

6. When the job is no longer visible with `squeue`, it means the job has finished. Check for output files by listing the files using the familiar `ls` command (The `l` is a loweracse letter L, not a number one):

   ```
   ls
   ```

   A useful version of this command is

   ```
   ls -ltr
   ```

   which lists the most recently written files at the bottom of the list. This is a quick way of seeing whether new files have been created or updated since you last ran the `ls` command.

7. Look at the contents of the job output file using the cat command (which simply displays the entire file in your terminal):

   ```
   cat slurm-JobIB.out
   ```

   This python script includes some code to tell you how long it took to calculate the Mandelbrot set. It should be about 30s. Was it?

8. Did you notice that a PNG image file had been created? Look at it using an image viewer (it is OK to run this on the CSF login node – it doesn't use much resource):

   ```
   eog mandelbrot.png
   ```

   (`eog` is the name of a Linux image viewer – it stands for Eye of Gnome! Linux likes to use short command names like `eog`, or `ls`, or `cat`.)

That's it! You have now run a serial (1-core) job on the CSF. First you submitted the job to the batch system. When the job was selected to be run by the batch system, it ran on a compute-node in the CSF (not on the login node.) The compute nodes are more powerful. It used no more than 5GB of RAM (memory) and it ran for about 30s. The output files it generated have been saved in the same directory from where you submitted the job.