Crib Sheet: Cirrus MPI Exercises

David Henty

1 Cirrus account

This document assumes you have already been given access to a login account on Cirrus. In the instructions below, you should replace `username` with your own login, and the project `taXXX` with your own project (e.g. `ta012`).

2 Logging on

Use your user name to access the Cirrus login nodes:

```
ssh -XY username@login.cirrus.ac.uk
```

and enter your Cirrus password. You may also be prompted for the pass-phrase for your SSH key.

If your login fails a few times, ask for help and **do not repeatedly try and log in** as your IP address may become temporarily blocked.

3 Obtaining source code

Remember that the compute nodes on Cirrus cannot see the `/home/` filesystem (your default home directory) so you should change directory to `/work/`, i.e.:

```
cd /work/taXXX/taXXX/username/.
```

You can obtain `MPP-templates.tar` from the MPI course pages on Learn. This will be downloaded to your local machine - you can copy them to Cirrus using scp (note that the ":" at the end is required!):

```
scp MPP-templates.tar username@login.cirrus.ac.uk:
```

Now unpack the tar file: `tar -xvf MPP-templates.tar` and change directory into it: `cd MPP-templates`

4 Compiling code

You **must** load non-default modules to access the correct version of MPI:

```
module load mpt
module load intel-compilers-19
```

You can compile the C, C++ and Fortran codes directly:

```
mpicc  -cc=icc   -o hello hello.c
mpicxx -cxx=icpc -o hello hello.cc
mpif90 -fc=ifort -o hello hello.f90
```

or using the supplied Makefiles (note you will have to edit these to change the compiler settings which default to ARCHER2):

```
make -f Makefile_c
make -f Makefile_cc
make -f Makefile_f90
```

This uses the Intel compilers. If you want to use the GNU compilers then you should unload the `intel-compilers-19` module and omit the `-cc=` or `-fc=` flags.

5 Running

You can run parallel jobs interactively from the command line, e.g.:

```
mpirun -n 4 ./hello
```

Running on the front-end like this you are sharing resources with other users. This is fine for development work, but if you want accurate timings for benchmarking you must run on the compute nodes using the Slurm batch system.

Now submit to the batch system: `sbatch cirrusmpi.job`

The batch system will respond with a unique ID for your job: `Submitted batch job XXXXX`.

You can monitor the progress of your job with `squeue -u username`. The letters "PD" indicate the job is queued ("pending"),"R" that it is running, and no listing that it has finished.

When the job has finished, the output will appear in a file called `hello-XXXXX.out`.

To change the number of processes you run on, edit the batch file and change the value of the `--ntasks=4`.

To run on more than 36 processes (i.e. more than one Cirrus node) you will also need to change value of `--nodes=1`. For example, for between 37 and 72 MPI processes you will need to specify `--nodes=2`.

To run on more than 2 nodes, or for more than 20 minutes, you will need to change `#SBATCH --qos=short` to `#SBATCH --qos=standard` and delete the line `#SBATCH --reservation=shortqos`

The script is set up to run an executable called `hello` - you can edit the final argument to `srun` to change this to your own parallel program.