Crib Sheet: Cirrus MPI Exercises

David Henty

1 Logging on

Mac / Linux Open a command-line terminal and use your username and password to access Cirrus using a secure shell: `ssh -XY username@login.cirrus.ac.uk`

Windows You should install MobaXterm from `https://mobaxterm.mobatek.net/`. After starting MobaXterm, click on "Sessions -> New Session" from the top bar and then select "SSH". You should enter the Cirrus login address `login.cirrus.ac.uk` as the "Remote host".

You **must** change directory to the `/work/` filesystem:

```
cd /work/ta178/ta178/username/
```

2 Obtaining source code

The source code is stored on github alongside the slides and other documentation. See the "Course materials" link from the main MPI course page at `https://github.com/EPCCed/BMKG-MPI-121124/`.

If you want to copy a file to Cirrus (rather than clicking on a link and downloading to your laptop, or cloning the entire repository), issue "wget" on Cirrus, e.g. for `MPP-templates.tar`:

```
wget https://github.com/EPCCed/BMKG-MPI-121124/raw/main/MPI/exercises/MPP-templates.tar
```

Note that, due to pecularities with github, this is **not** the link you get if you "copy link location" from your browser on github. To use `wget`, you must replace `blob` in the URL with `raw`.

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

3 Compiling code

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

```
module load mpt
module load intel-20.4/compilers
```

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
mpif08 -fc=ifort -o hello hellof08.f90
```

or using the supplied Makefiles

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

```
make -f Makefile_f90
make -f Makefile_f08
```

This uses the Intel compilers. If you want to use the GNU compilers then you should unload the `intel-compilers-19` module and remove flags that specify the Intel compilers (e.g. omit `-cc=icc`).

4 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.

The Slurm template file is called `cirrusmpi.job` and is set up to run an MPI program called `hello` on 4 processes. You will need to edit the file if you want to change this.

You can submit to the default queue in the batch system using: `sbatch cirrusmpi.job`

The batch system will respond with a unique ID for your job, e.g: `Submitted batch job 123456`

You can monitor the progress of your jobs with `squeue --me`. The letters “PD" indicates the job is queued (i.e. 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-123456.out`

You can kill a queued or running job using: `scancel 123456`

To change the number of MPI processes, or to use the reserved queue, you will need to edit the SLURM batch file.