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".

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 on the ARCHER2 website.

If you want to copy a particular file to Cirrus (rather than clicking on a link and downloading to your laptop, or cloning the entire repository) you can "copy link location" from your browser on the github pages, then issue "wget" on Cirrus, e.g. for `MPP-templates.tar` on the May 2020 run:

```
wget https://github.com/EPCCed/archer2-MPI-2020-05-14/blob/master/exercises/MPP-templates.tar
```

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-compilers-18
```

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

```
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-18` 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 PBS batch system.

The PBS template file is called `cirrusmpi.pbs` 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: `qsub cirrusmpi.pbs`

Note that during the practical sessions there may be a dedicated queue - e.g. R123456 - which you can submit to using: `qsub -q R123456 cirrusmpi.pbs`

The batch system will respond with a unique ID for your job: `XXXXXXX.sdb`.

You can monitor the progress of your job with `qstat -u username`. The letter "Q" indicates the job is queued, "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.pbs.XXXXXXX`

To change the number of processes you run on, edit the batch file and change the value of the `-n` argument to `mpiexec_mpt`. To run on more than 36 processes (i.e. more than one Cirrus node) you will also need to change the line `select=1:ncpus=36`. For example, for 4 nodes (a maximum of 144 MPI processes) use `select=4:ncpus=36`.

Regardless of how many processes you are running on, you should always specify `ncpus=36` and `-ppn 36`: you only need to change the value of `select`.