MPI on Cirrus and ARCHER2

Reusing this material

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

http://creativecommons.org/licenses/by-nc-sa/4.0/

This means you are free to copy and redistribute the material and adapt and build on the material under the following terms: You must give appropriate credit, provide a link to the license and indicate if changes were made. If you adapt or build on the material you must distribute your work under the same license as the original.

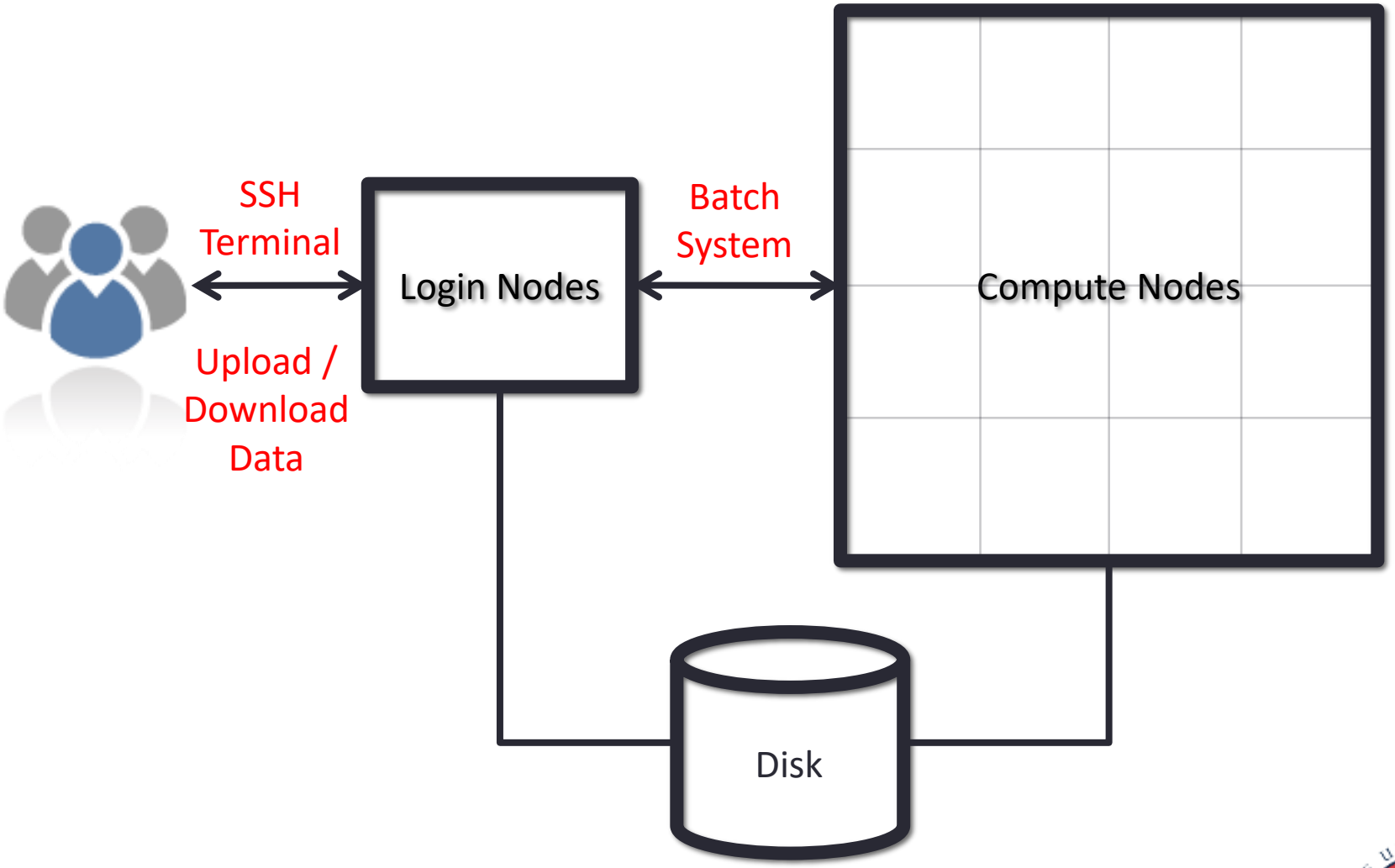
Acknowledge EPCC as follows: "© EPCC, The University of Edinburgh, www.epcc.ed.ac.uk"

Note that this presentation contains images owned by others. Please seek their permission before reusing these images.

Access

- ARCHER2: `ssh -X user@login.archer2.ac.uk`
- Cirrus: `ssh -X user@login.cirrus.ac.uk`
- You can access EPCC systems using ssh from anywhere
  - Linux: terminal + command line
  - Mac: Mac terminal + command line + enable an X server (e.g. xquartz) to display any graphics
  - Windows: need to install an ssh program such as MobaXterm
- This gives you access to a login node
  - a few login nodes, users assigned to different nodes depending on system load
  - login nodes shared by all users
  - the many hundreds of compute nodes are accessed exclusively via the SLURM batch system
  - on ARCHER2, MPI programs can only be run on the compute nodes
- Full instructions on access & login in separate document on course web pages

Typical HPC system layout



Useful files and templates

- Take a copy of **`MPP-templates.tar`**
  - see the course web pages
- unpack: **`tar -xvf MPP-templates.tar`**
- Crib sheets for MPI programs available on course web pages

Setting up Cirrus environment

- Load the Intel Compilers
  - module load intel-20.4/compilers
- Load the Message-Passing Toolkit
  - module load mpt
- To automate, add these lines to your ".bash_profile" file

```
[user@cirrus] emacs -nw ~/.bash_profile
```

Compiling MPI Programs on Cirrus

- C programmers use: **`mpicc -cc=icc`**
- C++ programmers use: **`mpicxx -cxx=icpc`**
- Fortran programmers use: **`mpif90 -fc=ifort`**

- There is nothing magic about these MPI compilers!
  - simply wrappers which automatically include various libraries etc
  - compilation done by standard (e.g. Intel) compilers
    - icc, icpc and ifort

- You can use the supplied Makefiles for convenience
  - **`make -f Makefile_c`**
  - **`make -f Makefile_cc`**
  - **`make -f Makefile_f90`**

- Easiest to make a copy of one of these called “Makefile”
  - also need to change the line “MF=“ in the Makefile itself

Running interactively on Cirrus

- Timings will not be reliable
  - shared with other users, many more processes than processors
  - but **very useful** during development and for debugging

- **`mpirun –n 4 ./mpiprog.exe`**
  - runs your code on 4 processes

- NOTE
  - output might be buffered
  - if your program crashes, you may see no output at all

- It *may* help to explicitly flush prints to screen
  - **`fflush(stdout);`**
  - **`FLUSH(6)`**

Running batch jobs on Cirrus

- Run via a batch system
  - Cirrus uses SLURM: you submit a batch script that launches your program
- In **`MPP-templates/`** is a standard batch script: **`cirrusmpi.job`**
  - set up to run a program called “hello” on 4 processors
- To run on 4 processors: **`sbatch cirrusmpi.job`**
  - runs executable called “hello”
  - output will appear in a file called **`hello-XXXX.out`**
  - can follow job progress using **`squeue`** or **`squeue -u <user>`**
  - to see only your own jobs: **`squeue --me`**
  - full instructions included as comments in the template

Cirrus Filesystems

- Cannot run from the home file system
  - on logging in, working directory is /home/*project*/*project*/*username*/
    - *project* is a code number, e.g. **`m24oc`**
  - compute nodes can only see the /work/ file system, not /home/

- Recommendation
  - do everything in /work/
  - i.e. change directory to /work/*project*/*project*/*username*/

Cirrus idiosyncrasies

- By default, MPI wrappers are not in your path

```
user@cirrus$ mpicc
-bash: mpicc: command not found
```

- To access correct version: **`module load mpt`**
  - defaults to GNU compilers: gcc, g++ and gfortran
  - in SLURM batch system, job launcher is called **`srun`**

- Intel compilers: **`module load intel-20.4/compilers`**
  - can add these to end of your **`.bash_profile`** file in home directory
  - to check you have the right version (similarly for mpif90)

```
user@cirrus$ which mpicc
/opt/hpe/hpc/mpt/mpt-2.25/bin/mpicc
```

  - to use Intel C compiler: **`mpicc -cc=icc`**
  - to use Intel C++ compiler: **`mpicxx -cxx=icpc`**
  - to use Intel Fortran compiler: **`mpif90 -fc=ifort`**

Compiling MPI Programs on ARCHER2

- C programmers use **`cc`**
- C++ programmers use **`CC`**
- Fortran programmers use **`ftn`**
- There is nothing magic about these MPI “compilers” !
  - simply wrappers which automatically include various libraries etc
  - compilation done by standard (Cray) compilers
    - **`clang`**, **`clang++`** and **`crayftn`**
- You can use the supplied Makefiles (C, C++, Fortran) for convenience
  - **`make -f Makefile_c`**
  - **`make -f Makefile_cc`**
  - **`make -f Makefile_f90`**
- Easiest to make a copy of your choice called “Makefile”
  - e.g. **`cp Makefile_c Makefile`**
    - also need to change the first line “MF=“ in the Makefile itself
  - then you can just type “**`make`**“

ARCHER2 idiosyncrasies

- Not possible to run MPI programs directly on login nodes
- Can be a substantial delay in batch queues
  - have dedicated queues for some courses for more rapid turnaround
  - or use the short queue for development
- Cannot run from the home file system
  - on logging in, your working directory is /home/taXXX/taXXX/username/
  - compute nodes can only see the /work/ file system, not /home/
- Recommendation
  - do everything in /work/
  - i.e. change directory to /work/taXXX/taXXX/username/

Running on ARCHER2 back-end

- Run via a batch system
  - on ARCHER2 we use SLURM
  - submit a script that then launches your program
- In MPP-templates/ is a standard batch script: **`archer2mpi.job`**
  - set up to run a program called **`hello`** in the short queue
- Submit: **`sbatch archer2mpi.job`**
  - runs on 4 processes of a single ARCHER2 node (each node has 128 CPU-cores)
  - output will appear in a file called **`hello-XXXXXX.out`**
  - can follow job progress using command: **`squeue -u $USER`**
  - full instructions included as comments in the template
- Batch script has instructions on how to use different queues

Running interactively

- Cannot run on ARCHER2 login node
  - must run on compute nodes via SLURM and sbatch
- However, can run “interactively”
  - sbatch command waits until resources are available
  - output appears on the screen in real time as your program runs
  - can be very useful for debugging (e.g. to detect deadlock)

```
srun --partition=standard --qos=short --reservation=shortqos \
--time=00:01:00 --unbuffered --cpu-bind=cores --nodes=1 --tasks=4 ./hello
```

C++ Interface

- MPI is not an OO interface
  - however, can be called from C++
- Originally had different function calls, e.g.
  - `MPI::Intracomm comm;`
  - `...`
  - `MPI::Init();`
  - `comm = MPI::COMM_WORLD;`
  - `rank = comm.Get_rank();`
  - `size = comm.Get_size();`
- Compiler is called `CC`
  - see `hello.cc` and `Makefile_cc`

C++ interface is now removed

Must therefore cross-call to C

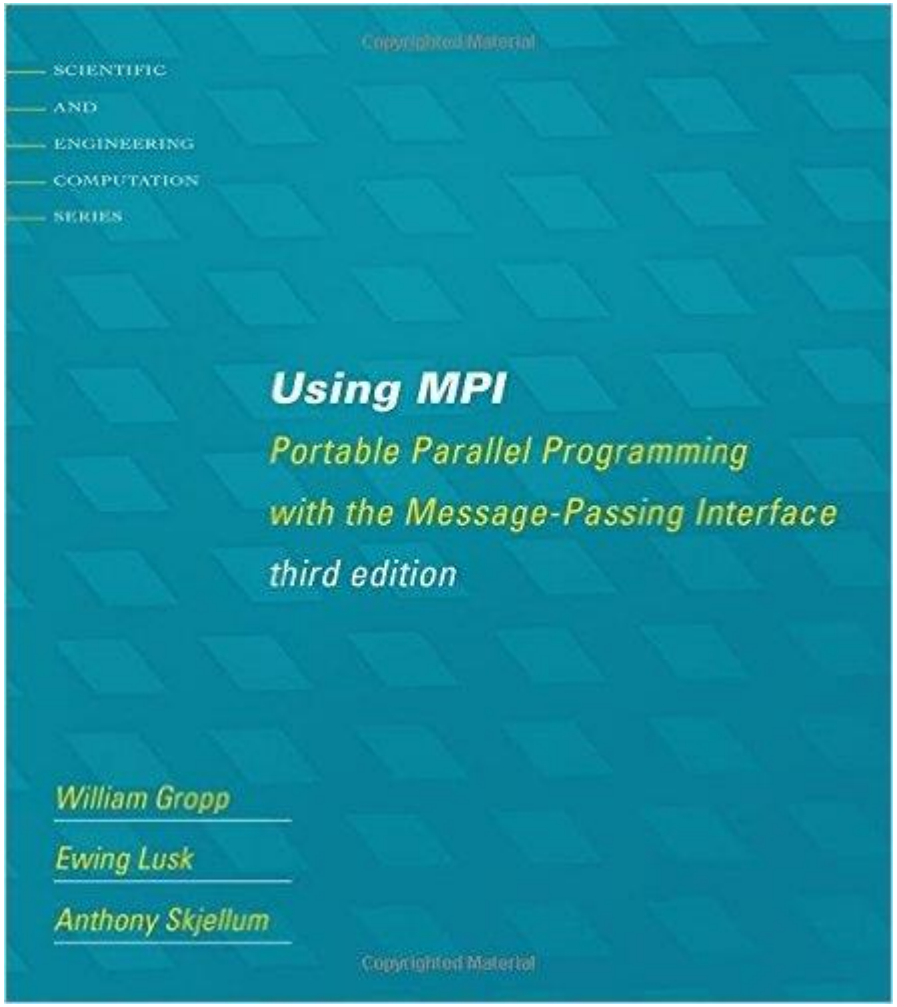
Documentation

- MPI Standard available online
  - See: http://www.mpi-forum.org/docs/
  - currently version 4.0

- Available in printed form
  - http://www.hlrs.de/mpi/mpi31/

- Man pages available on Cirrus and ARCHER2
  - must use the C style of naming: **`man MPI_Routine_name`**, e.g.:
  - **`user@computer$ man MPI_Init`**

MPI Books


Copyrighted Material
SCIENTIFIC
AND
ENGINEERING
COMPUTATION
SERIES
Using MPI
Portable Parallel Programming
with the Message-Passing Interface
third edition
William Gropp
Ewing Lusk
Anthony Skjellum
Copyrighted Material

Exercise: Hello World

The minimal MPI program

- See Exercise 1 on the exercise sheet
- Write an MPI program that prints a message to the screen
- Main purpose is to get you compiling and running parallel programs on HPC system
  - both interactively and in batch via SLURM and sbatch
  - also illustrates the SPMD model and use of basic MPI calls
- We supply some very basic template code
  - you need to add appropriate calls to compute rank and size