Introduction to HPC
Overview

• System architecture
• Logging in
• Software
• Modules
• Environment
• Preparing jobs
• Monitoring job
• Example scripts
• Getting help
• Applying for an account
Phase 1

Standard Nodes:
• 96 Compute nodes (4 cores per node)
• 8GB RAM per node
• Infiniband High Speed Network
• IBM GPFS Filesystem: 11 TB Storage

Large Memory Nodes:
• 4 Compute Nodes (8 cores per node)
• 32 GB RAM per node
Phase 2

Standard Nodes:
• 416 Compute nodes (8 cores per node)
• 8GB RAM per node
• Infiniband High Speed Network
• IBM GPFS Filesystem: 100 TB Storage

Large Memory Nodes:
• 2 Compute Nodes (24 cores per node)
• 256 GB RAM per node
Phase 3

Standard Nodes:
• 312 Compute nodes (16 cores per node)
• 64GB RAM per node
• Infiniband High Speed Network
• Panasas Parallel Filesystem : 300 TB Storage

Large Memory Nodes:
• 18 Compute Nodes (16 cores per node)
• 256 GB RAM per node

GPU Nodes
• 76 GPU Enabled Nodes: NVIDIA Tesla K20
Bluecrystal Phase 1

Bluecrystal Phase 1 is near end of life
Half of it has been repurposed and the
Remainder will be turned off Easter 2015

Use either Bluecrystal Phase 2 or 3
System Configuration
System Software

Operating System
GNU/Linux (Scientific Linux)

Queuing System

• Torque, PBS
• Torque, MOAB
System Software

New to GNU/Linux attend the Introduction to Linux Course

E-mail caroline.gardiner@bristol.ac.uk for details

Or Take a look at the Following Online Tutorial:

www.ee.surrey.ac.uk/Teaching/Unix
Phase 1
Phase 2 and 3
Type Of Jobs

Serial Code:

• High Capacity/Throughput Jobs

Parallel Code:

• Requires Additional Programming
• Uses the Infiniband High Speed Network

Two Types of Parallelism:

• MPI Message Passing Interface
• OpenMP
Logging In
Windows
Logging In

Windows
Logging In

Linux and Mac OSX

[cjw@hpc-03 ~]$ ssh -X iszcjw@bluecrystalp3.acrc.bris.ac.uk
iszcjw@bluecrystalp3.acrc.bris.ac.uk's password:
Last login: Thu Sep 18 13:13:44 2014 from newblue4.cm.cluster
[iszcjw@newblue1 ~]$
Logging In

Use the following commands to adjust your environment:

'module avail' - show available modules
'module add <module_name>' - adds it to your environment

You should add modules you use regularly (e.g. mpich) in your startup file.

Modules that add all default modules for a specific interconnect/compiler at once are called e.g. 'default-infiniband-pgi', 'default-infiniband-intel'.

Use the command showquota to see your current disk use, quota etc.

Also see the file /exports/gpfs/Diskuse to see how much free disk space there is, and how much space users/groups are using.

Please note user data stored on the system is not backed up and it is therefore your responsibility to ensure that you have adequate back up.

[fiszcjw@bluecrystal2 ~]$
Logging In

Transferring Data To The HPC Systems

Use:
• scp on Linux and Mac
• WinSCP on Windows
Logging In

Access From Outside The UoB Domain

From 20 September 2013 the University has implemented a new VPN service for the University. The new VPN, a Juniper Pulse appliance, is ready for Windows, Mac OS X, iOS and Android operating systems. For more information see the news item.

University’s virtual private network provides secure access to University network resources from offsite. Depending on your requirements other forms of offsite access are also available, which are often easier to set up and use.

On 30 June 2014 the old VPN service will be decommissioned and all users of the old service must set up a connection to the new Juniper Pulse service.

Instructions to setup your computer to connect to the UoB-VPN are here.
Logging In

Log in to Bluecrystal Phase 2

bluecrystalp2.acrc.bris.ac.uk
Available Software

Languages:
GCC-4.6.4
Intel Compiler XE
Intel Compiler XE
Intel Compiler XE
Intel Compiler XE
Java JDK 1.7.0-40
Mono-3.0.1
PERL 5.14.2
Python 2.6.8
Python 2.7.5
Python 2.7.6
R 2.15.1
R 3.0.2

Libraries:
GNU:
ATLAS
FFTW 3.3.4
GSL 1.16
Intel:
ATLAS

Tools:
CMAKE 2.8.1
CMAKE 2.8.12
GIT 1.8.4.2
Subversion-1.8.4

Profiling:
Intel VTune
TAU 2.23
# Available Software

**Applications:**

<table>
<thead>
<tr>
<th>Software</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abaqus</td>
<td>6.12</td>
</tr>
<tr>
<td>Amber</td>
<td>12</td>
</tr>
<tr>
<td>Beast</td>
<td>1.7.5</td>
</tr>
<tr>
<td>Comsol</td>
<td>4.3b</td>
</tr>
<tr>
<td>GROMACS</td>
<td>5.0</td>
</tr>
<tr>
<td>GCTA</td>
<td>1.24.3</td>
</tr>
<tr>
<td>LS-DYNA 971R6.1.0</td>
<td></td>
</tr>
<tr>
<td>Matlab</td>
<td>R2013b</td>
</tr>
<tr>
<td>Meep</td>
<td>1.2.1</td>
</tr>
<tr>
<td>NASTRAN 2012.1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Software</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Netlogo</td>
<td>5.0.5</td>
</tr>
<tr>
<td>OpenBabel</td>
<td>2.3.2</td>
</tr>
<tr>
<td>ParaView</td>
<td>4.0.1</td>
</tr>
<tr>
<td>PAML</td>
<td>4.7</td>
</tr>
<tr>
<td>PhyloBayes</td>
<td>3.3f</td>
</tr>
<tr>
<td>Plink</td>
<td>1.0.7</td>
</tr>
<tr>
<td>QuantumEspresso</td>
<td>5.1</td>
</tr>
<tr>
<td>Scilab</td>
<td>5.4.1</td>
</tr>
<tr>
<td>Trinity</td>
<td>2013.8.14</td>
</tr>
</tbody>
</table>
Available Software

If there is any software that you need that’s not already installed contact us and we’ll install it for you.

This applies to Python and R packages as well
Modules

Module Commands

module avail

module add  \textit{module-name}

module del  \textit{module-name}

module list  \textit{module-name}

Remember, modules that are required by a job need to be added to your .bashrc file
Environment

[iszczw@bigblue4 ~]$ pwd
/gpfs/cluster/isys/iszcjw

[iszczw@bigblue4 ~]$ ls -l .bashrc
-rw-r--r-- 1 iszcjw isys 7746 Aug 29 15:32 .bashrc

[iszczw@bigblue4 ~]$ more .bashrc
#/ .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

module add shared moab/5.2.3 torque/2.3.3
module add languages/R-2.15.1
e tc.
Preparing Jobs

Steps Required To Run A Job

• Ensure the required application module is included in your .bashrc file

• Or Compile your code (If Required)

• Copy Any Required Data Onto The System

• Create a Job Submission Script

• Submit The Job Script To The Queuing System
How The Queuing System Works

- The job script contains the commands required to run the job
- Submit the job script to the queuing system
- The queuing system then executes the commands in the script on the compute nodes

- Don’t expect your jobs to start instantly
- The Queuing system runs a fair share policy
- Users with a lot of jobs can not take over the system
- Get jobs in the queue sooner rather than later
Copy The Workshop tar File Into Your $HOME Directory

[iszcjw@newblue2 ~]$ cd
[iszcjw@newblue2 ~]$ cp ../..workshop.tar .
[iszcjw@newblue2 ~]$ ls -l workshop.tar
-rw-r-xr-x 1 iszcjw isys 10240 Sep 25 14:23 workshop.tar
[iszcjw@newblue2 ~]$

Unpack The tar File

[iszcjw@newblue2 ~]$ tar xvf workshop.tar
./workshop/job1.sh
./workshop/job2.sh
./workshop/job3.sh
./workshop/job4.sh
./workshop/job5.sh
[iszcjw@newblue2 ~]$

Workshop Package
Simple Job Scripts

Change Directory into workshop

[iszcjw@newblue2 ~]$ cd workshop
[iszcjw@newblue3 workshop]$ more job1.sh
#!/bin/bash
#
#
# Define working directory
export WORK_DIR=$HOME/workshop

# Change into working directory
cd $WORK_DIR

# Execute code
/bin/hostname

sleep 20
Queuing System Commands

qsub \textit{job\_script}

qstat \textit{job\_id\_number}

qdel \textit{job\_id\_number}

showstart \textit{job\_id\_number}

showq
Queuing System Commands

[iszczw@bigblue1 workshop]$ qstat 2630827
Job id          Name             User            Time Use S Queue
------------------------- ---------------- --------------- -------- - ----- 
2630827.bluequeue1 TenPerNode.txt  mp1728          0 R long

[iszczw@bigblue1 workshop]$ qstat -an1 2630827

bluequeue1.cvos.cluster:

<table>
<thead>
<tr>
<th>Job ID</th>
<th>Username</th>
<th>Queue</th>
<th>Jobname</th>
<th>SessID</th>
<th>NDS</th>
<th>TSK</th>
<th>Memory</th>
<th>Time</th>
<th>S Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2630827.bluequeuel</td>
<td>mp1728</td>
<td>long</td>
<td>TenPerNode.txt</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>241:0R</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>
Submit your first job

[iszcjw@bigblue4 workshop]$ qsub job1.sh
2630148.bluequeue1.cvos.cluster
[iszcjw@bigblue4 workshop]$
Monitoring Jobs

Use the –an1 switch on qstat to find where the job is running

[iszcjw@bigblue1 workshop]$ qstat -an1 2630627

bluequeue1.cvos.cluster:

<table>
<thead>
<tr>
<th>Job ID</th>
<th>Username</th>
<th>Queue</th>
<th>Jobname</th>
<th>SessID</th>
<th>NDS</th>
<th>TSK</th>
<th>Memory</th>
<th>Time</th>
<th>S Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2630627.bluequeu</td>
<td>phxct</td>
<td>short</td>
<td>Geant4Sim_0.sh</td>
<td>5144</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>

The log into that node and run top

[iszcjw@bigblue1 workshop]$ ssh u01n001
Last login: Fri Sep 26 10:53:02 2014 from bigblue4.cvos.cluster
[iszcjw@u01n001 ~]$ top
Monitoring Jobs

[iszcjw@u01n001 ~]$ top

top - 14:25:09 up 2:46, 1 user, load average: 8.01, 8.00, 7.34
Tasks: 223 total, 9 running, 214 sleeping, 0 stopped, 0 zombie
Cpu(s): 1.2%us, 0.9%sy, 21.6%ni, 76.1%id, 0.2%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 8155224k total, 2803888k used, 5351336k free, 89160k buffers
Swap: 15553312k total, 0k used, 15553312k free, 665884k cached

<table>
<thead>
<tr>
<th>PID</th>
<th>USER</th>
<th>PR</th>
<th>NI</th>
<th>VIRT</th>
<th>RES</th>
<th>SHR</th>
<th>%CPU</th>
<th>%MEM</th>
<th>TIME+</th>
<th>COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>5714</td>
<td>el14718</td>
<td>39</td>
<td>19</td>
<td>1090m</td>
<td>204m</td>
<td>79m</td>
<td>R 100.9</td>
<td>2.6</td>
<td>38:50.97</td>
<td>sander.MPI</td>
</tr>
<tr>
<td>5715</td>
<td>el14718</td>
<td>39</td>
<td>19</td>
<td>1089m</td>
<td>202m</td>
<td>29m</td>
<td>R 100.9</td>
<td>2.5</td>
<td>38:48.68</td>
<td>sander.MPI</td>
</tr>
<tr>
<td>5720</td>
<td>el14718</td>
<td>39</td>
<td>19</td>
<td>1088m</td>
<td>201m</td>
<td>29m</td>
<td>R 100.9</td>
<td>2.5</td>
<td>38:47.29</td>
<td>sander.MPI</td>
</tr>
<tr>
<td>5716</td>
<td>el14718</td>
<td>39</td>
<td>19</td>
<td>1089m</td>
<td>203m</td>
<td>30m</td>
<td>R 98.9</td>
<td>2.5</td>
<td>38:49.64</td>
<td>sander.MPI</td>
</tr>
<tr>
<td>5717</td>
<td>el14718</td>
<td>39</td>
<td>19</td>
<td>1088m</td>
<td>202m</td>
<td>29m</td>
<td>R 98.9</td>
<td>2.5</td>
<td>38:48.19</td>
<td>sander.MPI</td>
</tr>
<tr>
<td>5719</td>
<td>el14718</td>
<td>39</td>
<td>19</td>
<td>1088m</td>
<td>201m</td>
<td>29m</td>
<td>R 98.9</td>
<td>2.5</td>
<td>38:42.80</td>
<td>sander.MPI</td>
</tr>
<tr>
<td>5721</td>
<td>el14718</td>
<td>39</td>
<td>19</td>
<td>1088m</td>
<td>200m</td>
<td>29m</td>
<td>R 98.9</td>
<td>2.5</td>
<td>38:51.49</td>
<td>sander.MPI</td>
</tr>
<tr>
<td>5718</td>
<td>el14718</td>
<td>39</td>
<td>19</td>
<td>1088m</td>
<td>202m</td>
<td>29m</td>
<td>R 96.9</td>
<td>2.5</td>
<td>38:48.76</td>
<td>sander.MPI</td>
</tr>
</tbody>
</table>
Simple Job Script

```
iszcjw@newblue3 workshop]$ more job2.sh
#!/bin/bash
#
#
#PBS -l nodes=1:ppn=1,walltime=1:00:00

# Define working directory
export WORK_DIR=$HOME/workshop

# Change into working directory
cd $WORK_DIR

# Execute code
/bin/hostname

sleep 20
```
Simple Job Script

[iszczw@newblue3 workshop]$ more job3.sh
#!/bin/bash
#
#
#PBS -l nodes=1:ppn=1,walltime=1:00:00

# Define working directory
export WORK_DIR=$HOME/workshop

# Define executable
export EXE=/bin/hostname

# Change into working directory
cd $WORK_DIR

# Execute code
$EXE
sleep 20
Simple Job Script

[iszczw@newblue3 workshop]$ more job4.sh
#!/bin/bash
#
#
#PBS -l nodes=1:ppn=1,walltime=1:00:00
# Define working directory
export WORK_DIR=$HOME/workshop
# Define executable
export EXE=/bin/hostname
# Change into working directory
cd $WORK_DIR

echo JOB ID: $PBS_JOBID

echo Working Directory: `pwd`
echo Start Time: `date`
# Execute code
$EXE

echo End Time: `date`
Simple Parallel Job

[iszczw@newblue3 workshop]$ more job4.sh
#!/bin/bash
#
#
#PBS -l nodes=2:ppn=4,walltime=1:00:00
# Define working directory
export WORK_DIR=$HOME/workshop

# Define executable
export EXE=$HOME/workshop/hello
# Change into working directory
cd $WORK_DIR

Continued on next slide
Simple Parallel Job

# Generate the list of nodes the code will run on -------------------------

cat $PBS_NODEFILE
export nodes=`cat $PBS_NODEFILE`
export nnodes=`cat $PBS_NODEFILE | wc -l`
export confile=inf.$PBS_JOBID.conf

for i in $nodes; do
    echo $i >> $confile
done

# Execute the code -------------------------------------------------

mpirun -np $nnodes -machinefile $confile $EXE
Submit A Parallel Job

[iszcjw@bigblue1 workshop]$ mpicc helloworld.c -o hello

[iszcjw@bigblue1 workshop]$ ls -l hello
-rwxr-xr-x 1 iszcjw isys 7864 Sep 26 11:14 hello

[iszcjw@bigblue1 workshop]$ qsub job5.sh
2630626.bluequeue1.cvos.cluster
[iszcjw@bigblue1 workshop]$
Example Scripts

In order to run a number of parallel Abaqus jobs we can do the following:
Assume all the input file *.inp are in the working directory

Copy the following job submission template script into the working directory:

cut here ------------------------------------------
#!/bin/bash
#
#
#----------------------------------------------------------------------------------
#PBS -l walltime=12:00:00,nodes=1:ppn=4
#PBS -q abaqus
#
#PBS -q abaqus

# 1. Edit this
export MYDIR "${HOME}/Test/loop_test"
#
#----------------------------------------------------------------------------------
# cd $MYDIR
#----------------------------------------------------------------------------------
Example Scripts

Create a main.sh script containing the following:

```bash
#!/bin/bash
# Find each input file
# and strip off .inp to avoid confusing the script
for f in `ls *.inp | sed s/.inp//`
do

# Create a job script for each .inp file
cp qabaqus.parallel.sh qabaqus.parallel.sh.$f

# Add the execution line to the end of the job script
echo "abaqus job="$f "cpus=4 analysis double interactive" >> qabaqus.parallel.sh.$f

# Submit the job script to the queue
qsub qabaqus.parallel.sh.$f

sleep 10

done
```

The above script searches the current directory for input files and creates a unique job submission script for each from the job submission script template. It then submits each of the unique job submission scripts to the queue.
Array Jobs

Array jobs allow us to submit a number of jobs with a single command

#!/bin/bash
#
#
#PBS -l nodes=2:ppn=4,walltime=1:00:00

# Define working directory
export WORK_DIR=$HOME/workshop

# Define executable
export EXE=$HOME/workshop/hello.$PBS_ARRAYID

# Change into working directory
cd $WORK_DIR

Continued on next slide
Array Jobs

Continued

# Generate the list of nodes the code will run on -----------------------------

cat $PBS_NODEFILE
export nodes=`cat $PBS_NODEFILE`
export nnodes=`cat $PBS_NODEFILE | wc -l`
export confile=inf.$PBS_JOBID.conf

for i in $nodes; do
echo ${i} >>$confile
done

# Execute the code ----------------------------------------------------------

mpirun -np $nnodes -machinefile $confile $EXE
Array Jobs

Submit an array job

[iszczw@bigblue4 workshop]$ qsub -t 1-3 job6.sh
2631674.bluequeue1.cvos.cluster
[iszczw@bigblue4 workshop]$ qstat -u iszczw

bluequeue1.cvos.cluster:

<table>
<thead>
<tr>
<th>Job ID</th>
<th>Username</th>
<th>Queue</th>
<th>Jobname</th>
<th>SessID</th>
<th>NDS</th>
<th>TSK</th>
<th>Memory</th>
<th>Time</th>
<th>S Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2631674-1.bluequ</td>
<td>iszczw</td>
<td>veryshor</td>
<td>job6.sh-1</td>
<td>--</td>
<td>2</td>
<td>--</td>
<td>--</td>
<td>01:00</td>
<td>R</td>
</tr>
<tr>
<td>2631674-2.bluequ</td>
<td>iszczw</td>
<td>veryshor</td>
<td>job6.sh-2</td>
<td>--</td>
<td>2</td>
<td>--</td>
<td>--</td>
<td>01:00</td>
<td>R</td>
</tr>
<tr>
<td>2631674-3.bluequ</td>
<td>iszczw</td>
<td>veryshor</td>
<td>job6.sh-3</td>
<td>--</td>
<td>2</td>
<td>--</td>
<td>--</td>
<td>01:00</td>
<td>R</td>
</tr>
</tbody>
</table>
Array Jobs

[iszcjw@bigblue4 workshop]$ qstat -an 2631674-3

bluequeue1.cvos.cluster:

<table>
<thead>
<tr>
<th>Job ID</th>
<th>Username</th>
<th>Queue</th>
<th>Jobname</th>
<th>SessID</th>
<th>NDS</th>
<th>TSK</th>
<th>Memory</th>
<th>Time</th>
<th>S Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2631674-3.bluequ</td>
<td>iszcjw</td>
<td>veryshor</td>
<td>job6.sh-3</td>
<td>--</td>
<td>2</td>
<td>--</td>
<td>--</td>
<td>01:00</td>
<td>--</td>
</tr>
</tbody>
</table>

u03n007+u03n007+u03n007+u03n007+u03n009+u03n009+u03n009+u03n009
Matlab Script

Matlab jobs must be run through the queing system not the Matlab GUI

```
#!/bin/bash
#
#PBS -l walltime=1:00:00
#PBS -joe
#PBS -q testq

# Change into the working directory
cd /exports/gpfs/iszcjw/Test/matlab

# Execute the code
matlab -nodisplay -nojvm -nodesktop -nosplash < test.m
```
Getting Help

ACRC Website
https://www.acrc.bris.ac.uk

Service Desk
hpc-help@bristol.ac.uk
Applying For An Account

ACRC Website

https://www.acrc.bris.ac.uk
# Application Form

## BlueCrystal Application

### Personal details

<table>
<thead>
<tr>
<th>Field</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Name</td>
<td>Callum</td>
</tr>
<tr>
<td>Surname</td>
<td>Wright</td>
</tr>
<tr>
<td>Email</td>
<td><a href="mailto:C.Wright@bristol.ac.uk">C.Wright@bristol.ac.uk</a></td>
</tr>
<tr>
<td>Department</td>
<td>IT Services</td>
</tr>
<tr>
<td>Faculty</td>
<td>Other</td>
</tr>
<tr>
<td>Institution</td>
<td>University of Bristol</td>
</tr>
<tr>
<td>Telephone</td>
<td>(0117) 331 4319</td>
</tr>
<tr>
<td>User Type</td>
<td>STAFF</td>
</tr>
</tbody>
</table>

### Project details

Staff may submit a new project proposal or choose to join an existing project. To join an existing project you will need to know the project code, which is created and advised to the user when a project is approved.

- Join an existing project
- Create a new project
- Project code

### Additional information

The following information is not essential, but, if known, will help us to assess the requirements for your job.

- Preferred log-in shell: bash

You may use the box to provide details of the code you wish to run and, if known, compilation details and platform dependencies.
Application Form

Project details

Staff may submit a new project proposal or choose to join an existing project. To join an existing project you will need to know the project code, which is created and advised to the user when a project is approved.

- Join an existing project
- Create a new project

Project title *

Estimated CPU usage *
(kilo)hours

Estimated Disk usage *
(GB)

Estimated duration of project *
(Months)

Funding *
Please select...

Project proposal (500 Words Max)