

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

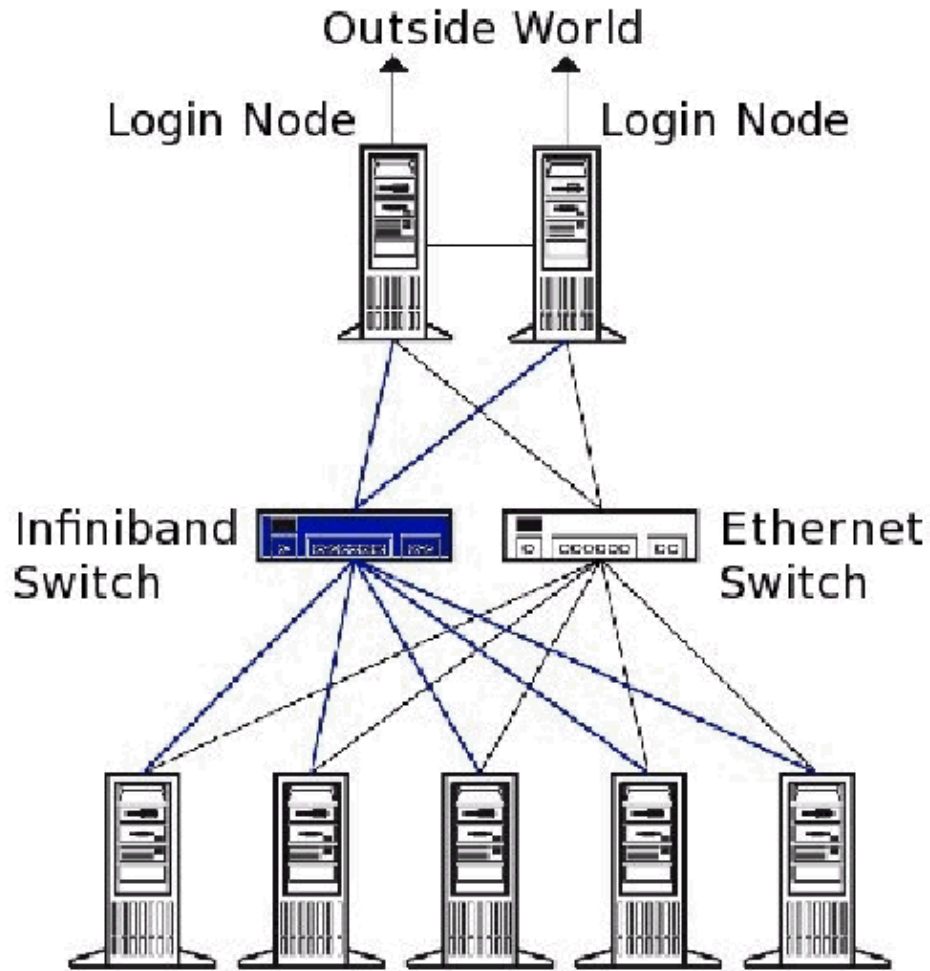
- 76 GPU Enabled Nodes: NVIDIA Tesla K20

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:

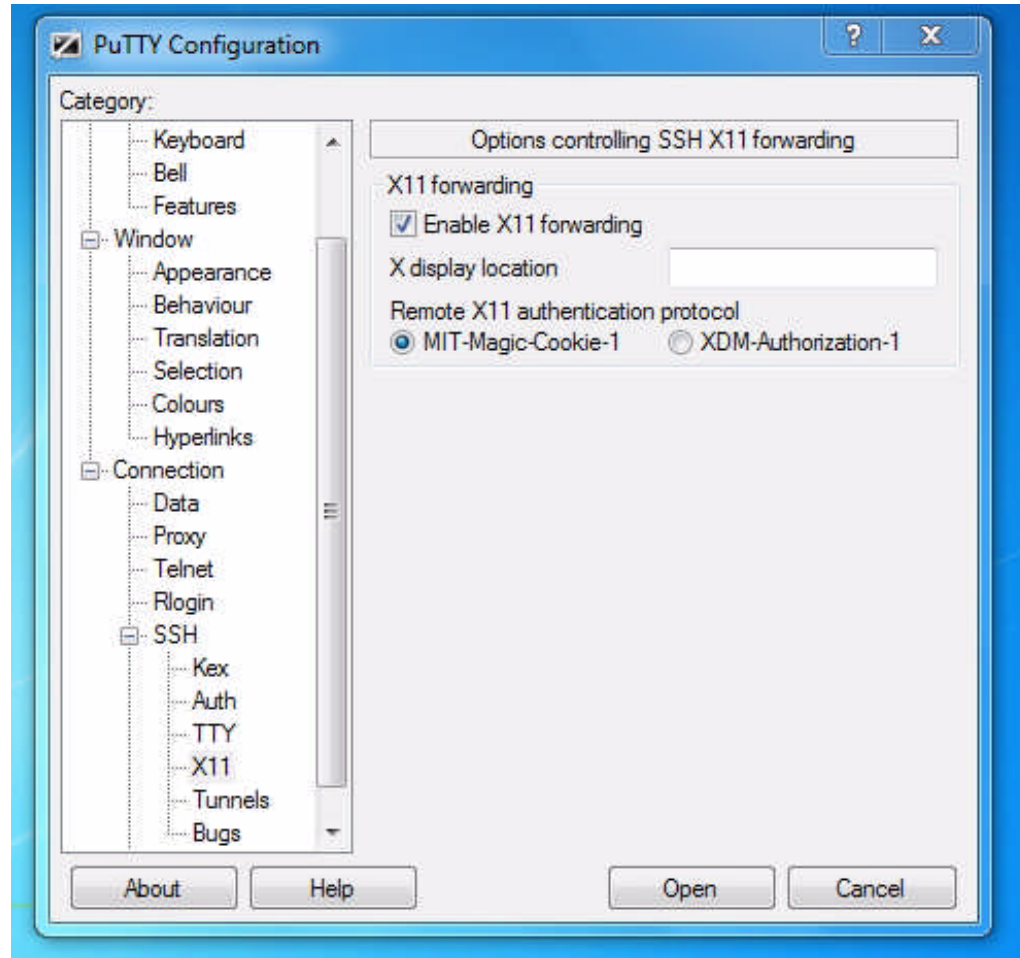
- Requires Additional Programming
- Uses the Infiniband High Speed Network

Two Types of Parallelism:

- MPI Message Passing Interface
- OpenMP

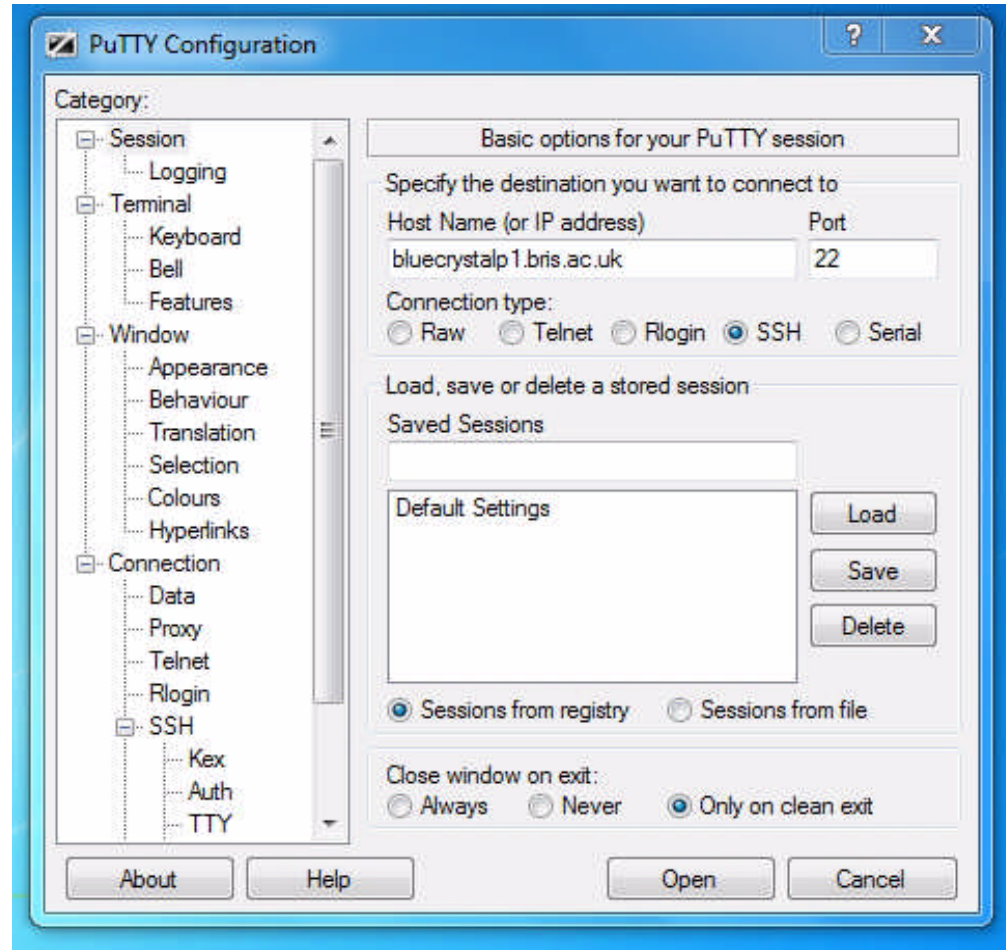
Logging In

Windows



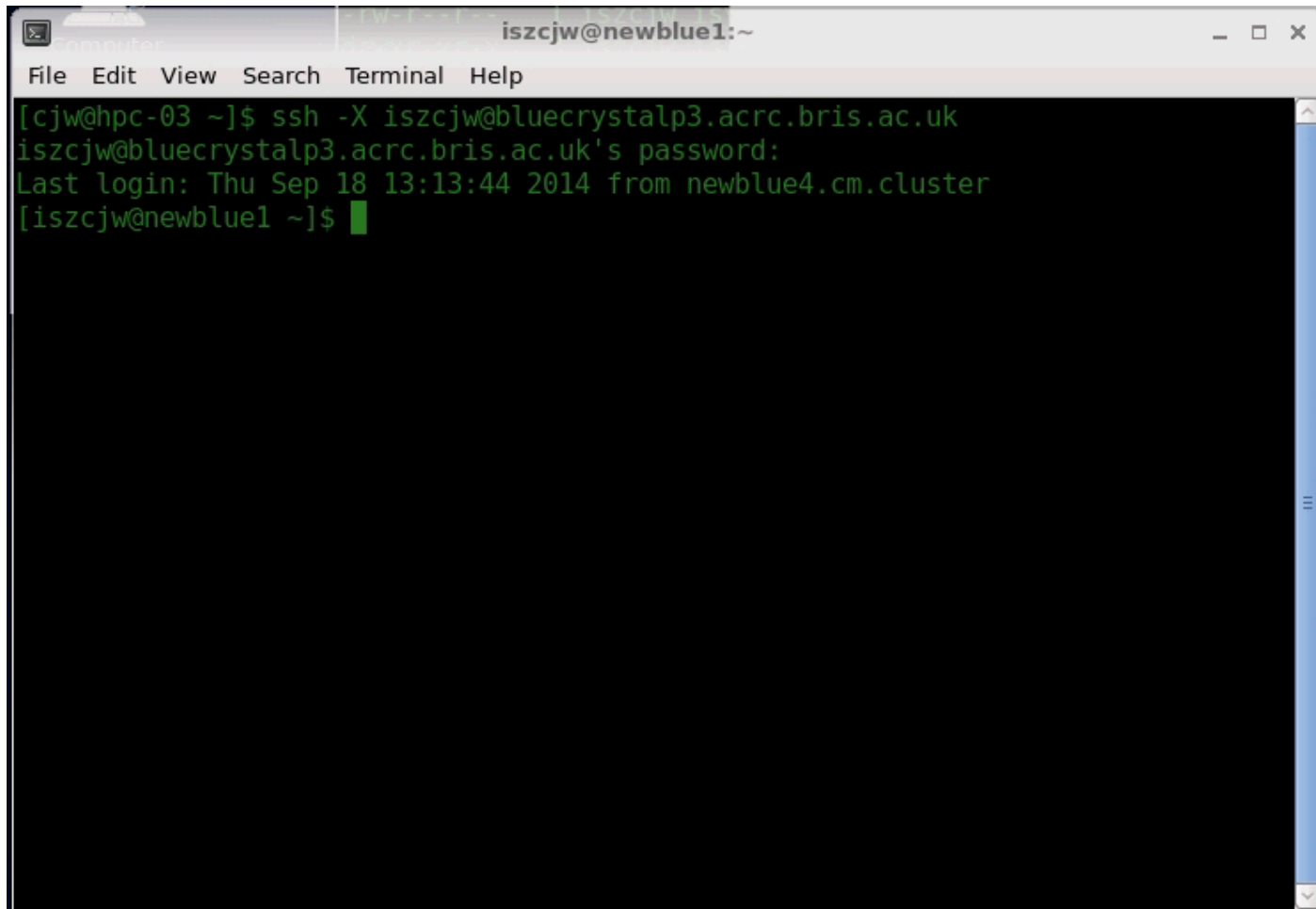
Logging In

Windows



Logging In

Linux and Mac OSX



A terminal window titled 'iszcjw@newblue1:~' showing an SSH session. The user 'iszcjw' is logged in from 'newblue4.cm.cluster'. The terminal output is as follows:

```
iszcjw@newblue1:~  
File Edit View Search Terminal Help  
[cjh@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

```
iszcjw@bluecrystal2:~  
login as: iszcjw  
iszcjw@bluecrystalp1.bris.ac.uk's password:  
Last login: Sun Sep 26 11:57:38 2010 from user-154-013-vpr.nomadic.bris.ac.uk  
-----  
|  |  | Welcome to                               Based on Scientific Linux 4.3  
-(-)- ClusterVisionOS 2.1                       ClusterVision ID: #50197  
|  |  |  
-----  
  
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 (eg. 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.  
-----  
[iszcjw@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

The screenshot shows a Mozilla Firefox browser window displaying the University of Bristol IT Services website. The address bar shows the URL: <https://www.bris.ac.uk/it-services/advice/homeusers/uobonly/uobvpn/>. The page title is "UoB VPN - Connecting from off campus". The navigation menu includes: IT Services homepage, Contact us, About, News, IT status, Locations, A-Z index, and Service catalogue. The main content area is titled "UoB VPN - Connecting from off campus" and contains the following text:

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.](#)

On the right side of the page, there is a "AskIT - get help from Bristol's staff & student community" box with a search input field and a "Continue" button.

The footer contains the following information:

IT Services home | IT staff area | create short link

Updated 5 June 2014 by IT Services | [Feedback](#)
University of Bristol, Computer Centre, 5 Tyndall Avenue, Bristol BS8 1UD, UK - Tel: +44 (0)117 928 7870

university home | a-z index | help | terms and conditions | privacy and cookie policy | © 2002-2014 University of Bristol

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

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:

Abaqus 6.12

Amber 12

Beast 1.7.5

Comsol 4.3b

GROMACS 5.0

GCTA 1.24.3

LS-DYNA 971R6.1.0

Matlab R2013b

Meep-1.2.1

NASTRAN 2012.1

Netlogo-5.0.5

OpenBabel 2.3.2

ParaView 4.0.1

PAML 4.7

PhyloBayes-3.3f

Plink-1.0.7

QuantumEspresso-5.1

Scilab 5.4.1

Trinity 2013.8.14

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 *module-name*

module del *module-name*

module list *module-name*

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

Environment

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

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

```
[iszcjw@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  
etc.
```


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

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

Workshop Package

Copy The Workshop tar File Into Your \$HOME Directory

```
[iszcjw@newblue2 ~]$ cd
[iszcjw@newblue2 ~]$ cp ../../workshop.tar .
[iszcjw@newblue2 ~]$ ls -l workshop.tar
-rwxr-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 ~]$
```

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 *job_script*

qstat *job_id_number*

qdel *job_id_number*

showstart *job_id_number*

showq

Queuing System Commands

```
[iszcjw@bigblue1 workshop]$ qstat 2630827
```

Job id	Name	User	Time Use	S	Queue
2630827.bluequeue1	TenPerNode.txt	mp1728		0	R long

```
[iszcjw@bigblue1 workshop]$ qstat -an1 2630827
```

```
bluequeue1.cvos.cluster:
```

Job ID	Username	Queue	Jobname	Req'd SessID	Req'd NDS	Elap TSK	Memory	Time	S	Time
2630827.bluequeue1	mp1728	long	TenPerNode.txt	--	2	--	--	241:0	R	--

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:
```

Job ID	Username	Queue	Jobname	Req'd SessID	Req'd NDS	Elap TSK	Memory	Time	S	Time
2630627.bluequeue	phxct	short	Geant4Sim_0.sh	5144	--	--	--	13:00	R	03:57 u01n001

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
```

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

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

```
[iszcjw@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

```
[iszcjw@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

```
[iszcjw@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  
  
# 1. Edit this  
export MYDIR "${HOME}/Test/loop_test"  
#  
#-----  
#  
cd $MYDIR  
#-----
```


Example Scripts

Create a main.sh script containing the following:

```
cut here -----
#!/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
cut here -----
```

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

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

bluequeue1.cvos.cluster:

Job ID	Username	Queue	Jobname	SessID	Req'd NDS	Req'd TSK	Elap Memory	Time	S	Time
2631674-1.bluequ	iszcjw	veryshor	job6.sh-1	--	2	--	--	01:00	R	--
2631674-2.bluequ	iszcjw	veryshor	job6.sh-2	--	2	--	--	01:00	R	--
2631674-3.bluequ	iszcjw	veryshor	job6.sh-3	--	2	--	--	01:00	R	--

Array Jobs

```
[iszcjw@bigblue4 workshop]$ qstat -an1 2631674-3
```

```
bluequeue1.cvos.cluster:
```

Job ID	Username	Queue	Jobname	SessID	Req'd NDS	Req'd TSK	Elap Memory	Time	S	Time
2631674-3.bluequ	iszcjw	veryshor	job6.sh-3	--	2	--	--	01:00	R	--

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

First Name *

Surname *

Email *

Department *

Faculty *

Institution *

Telephone

User Type STAFF

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

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 *

(kilohours)

Estimated Disk usage *

(GB)

Estimated duration of project *

(Months)

Funding *

Please select...

Project proposal (500 Words Max)