

# Introduction to HPC Workshop

Center for e-Research  
([eresearch@nesi.org.nz](mailto:eresearch@nesi.org.nz))

# Outline

## ① About Us

- About CER and NeSI
- The CS Team
- Our Facilities

## ② Key Concepts

- What is a Cluster
- Parallel Programming
- Shared Memory
- Distributed Memory

## ③ Using the Cluster

- Suitable Work

- What to expect

- Parallel speedup

- Data

- Getting to the Login Node

## ④ Submitting a Job

- Documentation

- Basic Job Properties

- Outputs

- Grisu

- LoadLeveler

- Software

- Best practices and advice

# About Us

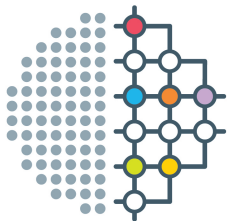
## CER: Center for e-Research

- Part of the University of Auckland
- User support and system maintenance

## NeSI : New Zealand eScience Infrastructure

- **NeSI** provides
  - high performance computing
  - a national data storage and sharing service
  - expert support, including engineering
  - single-sign on across the NZ research sector

# About Us



# NeSI

New Zealand eScience  
Infrastructure



# About Us

## Computational Science Team

- We support researchers to get the most out of our platforms and services.
- The CS Team has a lot of experience in HPC that spans many science domains.
- Collaboratively enhance the performance of research software codes.
  - Troubleshoot memory and other or I/O bottlenecks.
  - Connect researchers and scientific software experts.
  - The team is available to support researchers across any research institution in New Zealand.

# About Us

## Support

- Email [eresearch@nesi.org.nz](mailto:eresearch@nesi.org.nz)
- Creates a support 'ticket' where we can track the history of your request
- You can also arrange to meet us to discuss any issues



# Our Facilities

## NeSI Facilities

- NeSI provides several kind of HPC architectures and solutions to cater for various needs.
  - Bluegene/P
  - Power6 and Power7
  - Intel Westmere
  - Intel SandyBridge
  - Kepler and Fermi GPU servers
  - Intel Xeon Phi Co-Processor
- Supported applications can run on across several NeSI architectures.
- We can install and study the scalability in all the NeSI resources and find the most suitable environment for your case.
- See NeSI website for facility specs and application details.

# Our Facilities

## NeSI CeR Supercomputing Center

- funded by the **University of Auckland**, **Landcare Research** and the **University of Otago** with co-investment from the NZ Government through **NeSI**.
- Currently have around 5,000 Intel CPU cores across about 300 hosts.
- About 35 TB of memory and 80 TFLOPS (distributed).
- Shared storage of 200 TB with a 40 Gbit/s Infiniband network.
- Linux RHEL 6.3



# Our Facilities

## NeSI Pan Cluster

Architecture	Westmere	SandyBridge	LargeMem
Model	X5660	E5-2680	E7-4870
Clock Speed	2.8 GHz	2.7 GHz	2.4GHz
Cache	12MB	20MB	30MB
Intel QPI speed	6.4GT/s	8 GT/s	6.4GT/
Cores/socket	6	8	10
Cores/node	12	16	40
Mem/node	96GB	128GB	512GB
GFL OPS/node	134.4	345.6	384.0
# nodes	76	194	4

# Our Facilities

## NeSI Pan Cluster - Co-Processors

Architecture	Nvidia Fermi	Nvidia Kepler	Intel Phi
Main CPU	X5660/E5-2680	E5-2680	E5-2680
Model	M2090	K20X	5110P
Clock Speed	1.3GHz	0.732GHz	1.053GHz
Cores/Dev.	512	2688	60 (240)
Dev./node	2	2	2
Mem/Dev.	6GB	6GB	8GB
TFLOPS/Dev	1.33	1.17	1.01
# nodes	16	5	2

# Key Concepts

## What is a cluster

- A cluster is a network of computers, sometimes called nodes or hosts.
- Each computer has several processors.
- Each processor has several cores.
- A core does the computing.
- If your application uses more than one core, it can run faster on our cluster.

2013-10-09

# Introduction to HPC Workshop

## Key Concepts

### What is a Cluster

#### Key Concepts

#### Key Concepts

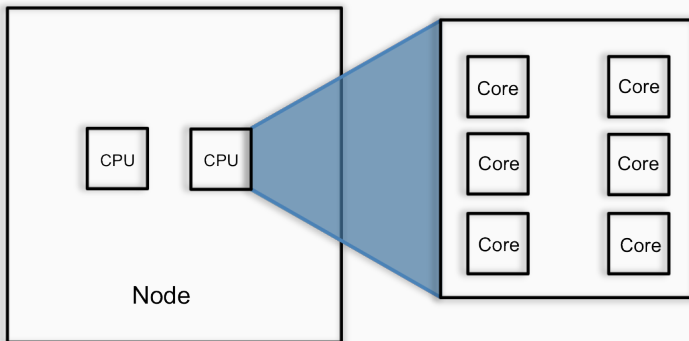
##### What is a cluster

- A cluster is a network of computers, sometimes called nodes or hosts.
- Each computer has several processors.
- Each processor has several cores.
- A core does the computing.
- If your application uses more than one core, it can run faster on a multi-processor.

Can the content of the node be changed by the image of the cpu's and the shared memory of a couple of slides further?

## Key Concepts

## HPC node overview



# Key Concepts

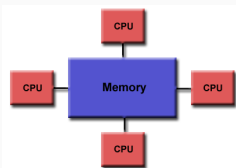
## Parallel Programming

- There are several ways to make a program use more cores and hence run faster.
- Many scientific software applications will be able to use multiple cores in some way. But this is often done explicitly by the user, not automatically.
- We can help you improve the performance of your code or make better use of your application.

# Key Concepts

## Shared Memory

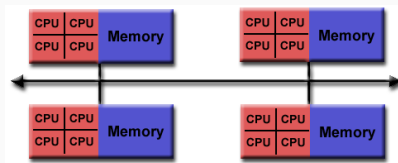
- Symmetric multiprocessing (SMP) : two or more identical processors are connected to a single shared main memory.
- This shared memory may be simultaneously accessed by single program using multiple threads.
- There are different frameworks for utilizing SMP capabilities.



# Key Concepts

## Distributed Memory

- Multiple-processor computer system in which each process has its own private memory.
- Computational tasks can only operate on local data.
- If remote data is required, the computational task must communicate with one or more remote processors.
- The most popular parallel programming paradigm is MPI.





# Key Concepts

## Useful Quick References

- [VI Quick Reference](#)
- [BASH Quick Reference](#)
- [Linux Quick Reference](#)
- [OpenMP Fortran Syntax](#)
- [OpenMP C/C++ Syntax](#)
- [MPI Quick Reference](#)

# Using the Cluster

## Overview

- The cluster is a shared resource and work must be scheduled.
- Jobs are queued by LoadLeveler (LL) and are executed on the compute nodes.
- The login node is not for running jobs, it is only for file management and job submission.

## Compiling and Testing Software

- In each NeSI facility you will find building/development nodes.
- We have the most up to date development tools ready to use.
- You can build and test your software and then submit a job.

## Introduction to HPC Workshop

## └ Using the Cluster

## └ Using the Cluster

## Using the Cluster

## Overview

- The cluster is a shared resource and work must be scheduled.
- Jobs are created by load balancer (LB) and are executed on the compute nodes.
- The login node is not for running jobs, it is only for file management and job submission.

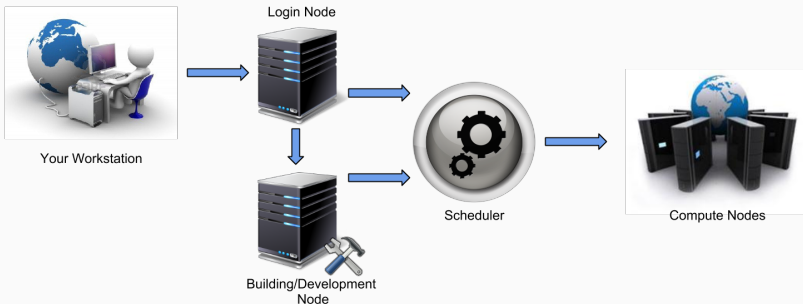
## Compiling and Testing Software

- In each HPC facility you will find building/development nodes.
- We have the most up to date development tools available.
- You can build and test your software and then submit a job.

Why are the compute nodes connected with the earth? It's only the login node, and perhaps the build node that have access to the www.

# Using the Cluster

## Using the cluster



# What to Expect

## Suitable Work

- Problems that can be solved with parallel processing.
- Problems that consume large amounts of memory.
- Problems that render your desktop useless for long periods of time.

## Less suited

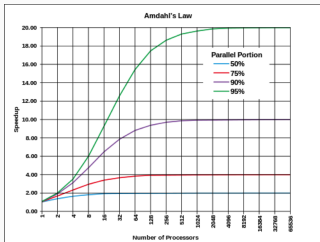
- Windows only software (work in progress).
- Interactive software, e.g. GUI, only available for development.

# What to expect

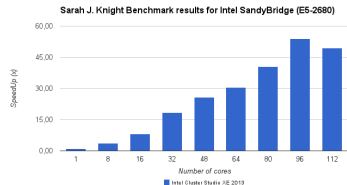
## Suitable Work

- Some problems are “embarrassingly parallel” i.e. it is trivial to divide the problem and solve independently.  
e.g. run simulation with 1000 different initial conditions.
- Approximately linear speedup.
- Other problems have dependencies, they cannot be separated  
e.g. simulate the weather.
- Speed up depends what % of the program runtime can be parallelised.

## Amdahl's Law



## Real Case: more cores $\neq$ more speed



## Parallel execution time

- Single core computation time: computation only.
- Parallel computation time: computation + communication + waiting.
- E.g.
  - writing results (to one file) is often a bottleneck.
  - Small problem on many cores: communication cost will dominate.
  - Unbalanced load: one core will mainly wait on the other.
- Conclusion: Test which number of cores is best suited for your problem.



# Using the Cluster

## Data

- Upload input data to the login node for use on the cluster.
- Download results from the login node to your local drive.
- The home directory has a rather small quota, project directories can be larger.
- For long term storage and back-up, ask your IT department.
- Things do go wrong, make sure to have a back-up.
- Files on the login node are shared across the build and compute nodes

# Using the Cluster

## Connection via SSH

Each terminal client has its own way of using the Secure Shell (SSH) protocol

- Windows: mobaxterm
- MacOSX: Terminal(Included in the OS), iTerm2
- Linux: Konsole, GnomeTerminal, yakuake

On Unix based systems you need to do something like:

```
ssh jbon007@login.nesi.org.nz
```

# Using the Cluster

Each NeSI Supercomputing Center has one or more Login Nodes

- **CeR**
  - `login.uoa.nesi.org.nz` which is the RHEL linux login node.
- **Bluefern**
  - `kerr.canterbury.ac.nz` which is the AIX unix login node.
  - `beatrice.canterbury.ac.nz` which is the SUSE linux login node.
  - `foster.canterbury.ac.nz` which is the BlueGene/P login node
  - `popper.canterbury.ac.nz` which is the Visualization Cluster login node.
- **NIWA**
  - `fitzroy.nesi.org.nz` which is the AIX unix login node.

# Using the Cluster

## Remote File System Access

In order to access the file system (/home) remotely from your machine, we recommend:

- **Windows** (mobaxterm) : mobaxterm
- **Windows** (SSHFS) :  
<http://code.google.com/p/win-sshfs/>
- **MacOSX** (SSHFS) : <http://code.google.com/p/macfuse/>
- **Linux** (SSHFS) :  
<http://fuse.sourceforge.net/sshfs.html>
- **KDE** (Konqueror) : type fish://user@host:port
- **Gnome** (Nautilus) : type sftp://user@host:port

# Submitting a Job

## Documentation

- Center specific documentation:
  - Bluefern :  
<http://wiki.canterbury.ac.nz/display/BlueFern>
  - NIWA : <http://teamwork.niwa.co.nz/display/HPCF/NIWA+HPCF+User+Documentation>
  - CeR : <http://wiki.auckland.ac.nz/display/CERES/>
- Examples for submitting jobs are on our Wiki page
- See the "Getting Started section"
- Take a look to the Quick Reference Guide.  
<http://goo.gl/ytbRWy>
- You will also find links to available software on the cluster

# Submitting a Job

## Basic Job Properties

- **Name** So you can identify the output later.
- **Job Type** How many processes and how many threads?
- **Walltime** How long the job can run for. The job will be cancelled if the walltime is exceeded.
- **Memory** How much memory to use? Job will die if memory is exceeded.
- **CPU cores** How many to use? Your program may try to use more than you request e.g. MATLAB.
- **Account or Group information** Especially important for access to licensed software and funded research allocations
- **Emails** Notification of job starting, also scheduler errors.

# Submitting a Job

## Two main tools for submitting a job

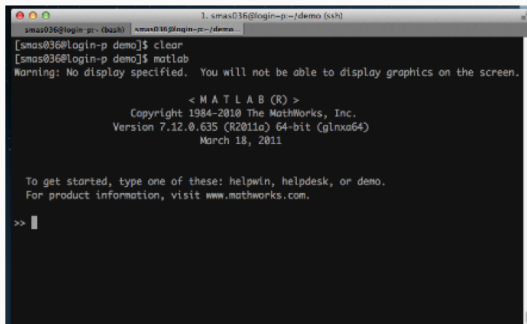
- LoadLeveler – for people comfortable with the Linux command line
- Grisu Template Client – for a graphical interface

## Which one to use? In general

- LL for complex workflows or large numbers of jobs
- Grisu for simple workflows or few jobs

# Submitting a Job

## Outputs

A terminal window titled '1. smas036@login-p-~/demo (ssh)' showing a user logging in as 'smas036' on a node named 'login-p demo'. The user enters 'clear' and 'matlab'. A warning message states: 'Warning: No display specified. You will not be able to display graphics on the screen.' The MATLAB startup banner follows, including the version '7.12.0.635 (R2011a) 64-bit (glnxa64)' and the date 'March 18, 2011'. It also provides instructions on how to get started and where to find product information. The prompt '>>' is visible at the bottom.

```
smas036@login-p- (bash) 1. smas036@login-p-~/demo (ssh)
[smas036@login-p demo]$ clear
[smas036@login-p demo]$ matlab
Warning: No display specified. You will not be able to display graphics on the screen.

< M A T L A B (R) >
Copyright 1984-2010 The MathWorks, Inc.
Version 7.12.0.635 (R2011a) 64-bit (glnxa64)
March 18, 2011

To get started, type one of these: helpwin, helpdesk, or demo.
For product information, visit www.mathworks.com.

>> █
```

Jobs have no interactive interface, but command line output and can write to files. Graphical tools are, however, available on the login and build/development nodes.



# Submitting a Job

## Outputs

- Information output while the job runs is written to a text file.
- Standard output goes to stdout, standard error goes to stderr.
- These should have unique names for a given job directory (see job Name)
- If your application writes to other files e.g. output data, that stays the same
- When your job fails, first look at stdout and stderr for clues

# Submitting a Job

## Quick Intro to Grisu

- Cross platform Java client: Windows, Mac, Linux
- Grisu interfaces with LoadLeveler to submit and monitor jobs
- Basic workflow:
  - Login
  - Set requirements
  - Attach files
  - Submit job
  - Wait ... check status
  - Download results

# Submitting a Job

## Quick Intro to LoadLeveler

- You need to access the login node and work from a terminal
- Requires basic knowledge of the Linux command line:
  - How to navigate file system and edit files
  - Shell scripting is very useful for automation
  - Tutorials available online at Software Carpentry – computing basics aimed at researchers

# Submitting a Job

## Setup a Job Description

Can use macros in job attributes

e.g. `#@ output = $(job_name).$(jobid).out`

MPI jobs

```
#@ job_type = MPICH | parallel
```

```
#@ total_tasks = 16
```

```
#@ blocking = 4 | unlimited
```

# Submitting a Job

## Setup a Job Description

### GPUs

```
#@ resources = ... GPUDev(1)
```

### Specific architectures

```
#@ requirements = (Feature=="sandybridge")
```

```
#@ requirements = (Feature=="Kepler")
```

# Submitting a Job

```
#!/bin/bash
# Optimized for run parallel job of 12 Cores in NeSI (Pandora-westmere)
#####
#@ job_name = Gaussian
#@ class = default
#@ notification = never
#@ group = nesi
#@ account_no = uoa
#@ wall_clock_limit = 1:00
#@ initialdir = $(home)
#@ output = $(home)/$(job_name).txt
#@ error = $(home)/$(job_name).err
#@ job_type = serial
#@ resources = ConsumableMemory(2048mb) ConsumableVirtualMemory(2048mb)
#@ parallel_threads = 12
#@ environment = COPY_ALL,OMP_NUM_THREADS=12
#@ queue
#####
module load g09/C.01
cd $SCRATCH_DIR
cp -r $HOME/Gaussian/h2o_opt.dat .
setenv GAUSS_SCRDIR $SCRATCH_DIR
### Run the Parallel Program
g09 < ./h2o_opt.dat > h2o1_opt.log
### Transferring the results to the home directory ($HOME)
cp -pr $TMP_DIR $HOME/results/
```

# Submitting a Job

```
#!/bin/bash
# Optimized for run parallel job of 512 Cores at NeSI (Pandora-SandyBridge)
#####
#@ job_name = LAMMPS_TEST
#@ class = default
#@ group = nesi
#@ notification = never
#@ account_no = uoa
#@ wall_clock_limit = 00:30:00
#@ resources = ConsumableMemory(4096mb) ConsumableVirtualMemory(4096mb)
#@ job_type = MPICH
#@ blocking = unlimited
#@ node_usage = not_shared
#@ output = $(job_name).$(jobid).out
#@ error = $(job_name).$(jobid).err
#@ requirements = (Feature=="sandybridge")
#@ initialdir = /share/src/LAMMPS/lammps-12Aug13/bench
#@ total_tasks = 512
#@ queue
#####
module load lammps/12Aug13-sandybridge
cd $SCRATCH_DIR
cp /share/test/LAMMPS/* .
### Run the Parallel Program
export OMP_NUM_THREADS=1
MPIRUN lmp_mpi -var x 20 -var y 20 -var z 20 -in in.lj > lj-512.out
### Transferring the results to the home directory ($HOME)
cp -pr $SCRATCH_DIR $HOME/OUT/lammps/
```

# Submitting a Job

## Environment Modules

- Modules are a convenient way to provide access to applications on the cluster
- They prepare the environment you need to run the application
- Commands
  - **module avail** - lists available modules
  - **module show module\_name** - displays full information about the module with name *module\_name*.
  - **module load module\_name** - loads the module with name *module\_name* and its dependencies.
  - **module unload module\_name** - unload the module with name *module\_name* and its dependencies.
  - **module list** - list all modules currently loaded.
- Grisu loads a module when you select an application



# Submitting a Job

## LoadLeveler

- To submit a job  
`llsubmit myjob.ll`
- To monitor a job  
`llq -u "myuserid"`
- Shows job id and status – R, I, etc
- To cancel  
`llcancel "jobid"`

# Submitting a Job

## Notes for Windows Users

- Be careful of Windows end of line (EOL) characters, sometimes Linux will not handle them correctly
- Notepad++ lets you convert between Windows and Unix style line endings
- Even though you can avoid using the Linux command line, having a basic understanding will help you debug your jobs

# Submitting a Job

## Software

- We have many specialised software packages.
- Best way to see what we have is by checking the wiki.
- The Wiki also has a software section
- We can install software that you need:
  - Linux version of the software.
  - Command line mode without user interaction.
  - Interaction possible for small tests on the build nodes.
  - We don't provide licenses.
  - You can install software in your home directory if it is really esoteric.

# Submitting a Job

## Best practices and advice

- Share with us a short test and we will study the scalability of your application.
- Try to be accurate with the walltime, it will help to the LL to schedule the jobs better.
- Be aware that you are sharing resources with other researchers.
- A wrong memory request or a wrong job description setup can potentially affect others.
- If we find some case like that, we may be forced to cancel the job with this behavior and inform the owner by email.

# Our Expectations

## Our Expectations

- We have an acceptable use policy that follows the NeSI IT policies
- We conduct regular reviews of projects to :
  - see how you are going and if you could use some help
  - collect any research outputs from your work on our facility
  - determine how the cluster has helped your research
  - look at the potential for feature stories on your work
- Please contact us if you have any questions
- Please acknowledge us in your publications

# Questions & Answers

