Quick Reference Guide to the High Performance Computer Environments @ NeSI

New Zealand eScience Infrastructure

New Zealand eScience Infrastructure (NeSI)
a partnership for all New Zealand researchers delivered by
The University of Auckland, NIWA, University of Canterbury,
Landcare Research, Otago University and MBIE

Contact Information

- **web**: `http://www.nesi.org.nz`
- **wiki**: `https://wiki.auckland.ac.nz/display/CERES/`
- **ganglia**: `http://ganglia.uoa.nesi.org.nz`
- **support** : support@nesi.org.nz

Useful Quick References

- VI Quick Reference
- BASH Quick Reference
- Linux Quick Reference
- OpenMP Fortran Syntax
- OpenMP 3.1 API C/C++ Syntax
- MPI Quick Reference

Hardware Information

Architecture	Westmere	SandyBridge	LargeMem	GPU
Model	X5660	E5-2680	E7-4870	X5660 (M2090)
Clock Speed	2.8 GHz	2.7 GHz	2.4GHz	2.8 (1.3)GHz
Cache	12MB	20MB	30MB	12MB
Intel QPI speed	6.4GT/s	8 GT/s	6.4GT/	6.4GT/s
Cores/socket	6	8	10	6(512)
Cores/node	12	16	40	12(1024)
Mem/node	96GB	128GB	512GB	96GB(6GB)
GFLOPS/node	134.4	172.8	384.0	134.4 (1330DP)

Disk Spaces + Default Quota

FileSystem	Space	Quota	ACL	Backup	Type	Usage
$HOME	120TB	30GB	rw	yes	GPFS	Archive
/share	120TB	-	ro	yes	GPFS	Archive
$TMP_DIR	240GB	-	rw	NO	EXT4	io
$SCRATCH_DIR	8.8TB	-	rw	NO	GPFS	io

Environment Modules

Environment Modules is very useful to manage environment variables for each application and it is very easy to use. It loads the needed environment by a certain application and its dependencies automatically.

- **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.

Available Compilers

Compiler	Intel	GNU	PGI
Fortran77	ifort	g77	pgf77
Fortran90	ifort	gfortran	pgf90
Fortran95	ifort	gfortran	pgf95
C	icc	gcc	pgcc
C++	icpc	g++	pgCC
Debug	idb	gdb	pgdbg
Profile	vtune	gprof	gpprof

Available MPIs

MPI	version
Intel MPI	4.1.0.024
OpenMPI	1.4,1.6
MPICH2	1.5,3.0.4
PlatformMPI	08.02
MVAPICH2	1.4.1p1

Optimization flags

Compiler	Intel	GNU	PGI
High Opt.	-fast	-O3 -ffast-math	-fast -Mipa=fast,inline
OpenMP	-openmp	-fopenmp	-mp=nonuma
Debug	-g	-g	-g
Profile	-p	-pg	-p
Westmere	-mtarget	-march=corei7	-tp=nehalem-64
SandyBridge	-mtarget	-march=corei7-avx	-tp=sandybridge-64
SSE	-xsse4.2	-msse4.2	-Mvect=[prefetch,sse]
AVX[1]	-xavx	-mavx	-fast

1. Advanced Vector Extension (AVX) streaming SIMD instructions. Sandy Bridge processor only.

Link MKL with OpenMPI, CDFT, ScaLAPACK, BLACS and Intel Compilers

Link line:
```
-L${MKLROOT}/lib/intel64 -lmkl_scalapack_ilp64 \
-lmkl_cdft_core -lmkl_intel_ilp64 -lmkl_sequential \
-lmkl_core -lmkl_blacs_intelmpi_ilp64 -lpthread -lm
```
Compiler options: `-DMKL_ILP64 -I${MKLROOT}/include`
More information at `http://software.intel.com/sites/products/mkl/`

Link MKL with OpenMP and Intel compilers

Link line:
```
-L${MKLROOT}/lib/intel64 -lmkl_intel_ilp64 \
-lmkl_intel_thread -lmkl_core -lpthread -lm
```
Compiler options: `-openmp -DMKL_ILP64 -I${MKLROOT}/include`

LoadLeveler Commands

- **llsubmit** sends a new job to the scheduler for execution
- **llclass** show the current list of classes.
- **llq** returns queue information
  - `llq -l <job id>` gives extended job record information.
  - `llq -s <job id>` gives specific information about why the job is held.
  - `llq -u <user name>` lists only jobs initiated by the user.
- **llcancel** kills a queued job: `llcancel <job id>`

More information about LoadLeveler in User Manual :
`http://www.redbooks.ibm.com/redbooks/pdfs/sg246038.pdf`

Limits

Name	MaxJobCPU	Description
small1h	41	Jobs running for under 1 hour (high priority)
small6h	41	Jobs running for under 6 hours (high priority)
medium	12500	Jobs running for under 1 week (medium priority)
long	unlimited	Jobs running for longer than a week (low priority)
bigmem	unlimited	Jobs that needs more than 12GB/core

OpenSSH Access

Parameters

- host: login-01.uoa.nesi.org.nz
- port: 22

Limits

- CPU time: 60 minutes
- Memory : 2GB

Suggested Software

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

Building Nodes

The building nodes will allow you to build the binaries for specific architecture. The binaries compiled in Westmere can run in Sandy Bridge, but it can **NOT** exploit all the Sandy bridge features. The binaries compiled in Sandy Bridge can **NOT** run in the Westmere nodes. The interactively usage is limited up to **24h** of **CPUTime** and **70GB** of **memory**.

- Westmere + GPU : build-gpu-p
- SandyBridge + GPU : build-gpu-sb-p
- SandyBridge + Phi : build-phi-p

Remote File System Access

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

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

Remote File System Transfer with RSYNC (Unix Only)

RSYNC over SSH protocol is the best choice to transfer big data volumes.

- Transfer data from your machine to the server:
  `rsync -avHl /path/origin/* sshserver:/path/destination/`
- Transfer data from the server to your machine:
  `rsync -avHl sshserver:/path/destination/* /path/origin/`

Remote File System Transfer with scp/sftp (Unix Only)

SCP and **SFTP** are the most popular software to transfer data across the SSH protocol.

- **SCP**: `scp -pr sshserver:/path/destination/* path/destination/`
- **SFTP**: `sftp sshserver:/path/destination/* path/destination/`

Get Involved!

We would like to encourage you to send suggestions and feedback to the NeSI Team.

Quick Reference Guide to the High Performance Computer Environments @ NeSI

New Zealand eScience Infrastructure

New Zealand eScience Infrastructure (NeSI)
a partnership for all New Zealand researchers delivered by
The University of Auckland, NIWA, University of Canterbury,
Landcare Research, Otago University and MBIE

SubmitScript Example : SMP job using local disk

```
#!/bin/bash
# Gaussian SubmitScript
# Optimized for run parallel job of 8 Cores in NeSI Pan Cluster
#########################################################################
#@ 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 = 8
#@ environment = COPY_ALL,OMP_NUM_THREADS=8
#@ queue
#########################################################################
###  Load the Enviroment Modules for Gaussian G09 C.01
. /etc/profile.d/modules.sh
module load g09/C.01
#########################################################################
###  Transfering the data to the temporal disk  ($TMP_DIR)
cd $TMP_DIR
cp -r $HOME/Gaussian/h2o_opt.dat .
setenv GAUSS_SCRDIR $TMP_DIR
#########################################################################
###  Run the Parallel Program
smpexec g09 < ./h2o_opt.dat > h2ol_opt.log
#########################################################################
###  Transfering the results to the home directory ($HOME)
cp -pr $TMP_DIR $HOME/results/
```

Most important OpenMPI (mpirun) Options

To find the full list of the available component types under the MCA architecture load the OpenMPI module and execute : `ompi_info --param all all`. More information at **OpenMPI WebSite**

- `-c, -n, --n, -np <#>` Run this many copies of the program on the given nodes.
- `-npersocket <#persocket>` Launch this processes for each sockets on the node.
- `-pernode` On each node, launch one process.
- `--app <appfile>` Provide an appfile, ignoring all other command line options.
- `-bind-to-core, -bind-to-socket` For process binding
- `-debug` Invoke the user-level debugger.
- `-debugger` Sequence of debuggers to search for when -debug is used.
- `-mca, --mca <key> <value>` Send arguments to various **MCA modules**.
  - `btl self,sm,openib` Select which network to use for MPI communications
  - `btl_sm_use_knem 1` Use KNEM for intra-node MPI.
  - `coll_sm_control_size` Buffer size (in bytes) chosen to optimize collective operations
  - `coll_fca_enable 1` Use Fabric Collective Accelerator (Only Mellanox)
  - `coll_base_verbose 1` Verbosity level for the coll framework
  - `coll_tuned_use_dynamic_rules 1` Dynamic optimization of collective operations
  - `coll_tuned_alltoall_algorithm 3` AllToAll Collective communication algorithm
  - `coll_tuned_allreduce_algorithm 4` AllReduce Collective communication algorithm
  - `mtl mxm` Use MellanoX Messaging library which provides enhancements to parallel communication libraries.

Remember that you don't need to setup this options if you are using the MPIRUN wrapper!

Submit Script Syntax

- **account_no** = (uoa, uoc, uoo, landcare or nesiXXXX)
- **blocking** = (unlimited or N). Where N is the number of tasks allocated to nodes
- **group** = Identifies what group the job submission is done under.
  - nesi - users without a particular NeSI project
  - chemistry - University of Auckland Chemistry Department
  - pd - NeSI Proposal Development allocations
  - unfunded - NeSI Research allocations without NeSI HPC funding
  - funded - NeSI Research allocations with NeSI HPC funding
- **job_name** Sets the job name and the $job_name macro
- **job_type** Specifies the job type for the submission.
  - Serial - single- or multi-threaded job, SMP, confined within a single node
  - MPI - Distributed Parallel Environment (includes OpenMPI 1.4 but not 1.6)
  - MPICH - Distributed Parallel Environment includes MPICH and OpenMPI 1.6
- **notification** Specifies under which conditions Loadleveler will send out email notifications to the specified address about the state of the job:
  - always - notifies about every change in the job status
  - start - notifies when the job is activated
  - complete - notifies when the job ends
  - error - notifies only if the job fails
  - never - never send any notifications
- **notify_user** user email
- **parallel_threads** Sets the number of cores to be allocated for each task
- **resources** Specifies what consumable resources the job requires to run per task:
  - ConsumableMemory - the amount of physical memory required.
  - ConsumableVirtualMemory - the amount of virtual memory required.
  - GPUDev - the number of GPU devices required (max 2 GPU).
  - AnsysLicenses - the number of ANSYS licenses to check out (max 40).
- **tasks_per_node** specifies the total number of tasks (cores) to be run on each available node.
- **total_tasks** specifies how many tasks (cores) are to be run in total
- **wall_clock_limit** Sets the limit for the elapsed time for which a job can run

Useful MPIRUN Wrapper

Inside each application module file there is a useful command wrapper of mpirun that can simplify its usage. In your script you only have to write:

```
MPIRUN binary <aplication options>
```

Instead of:

```
mpirun -np 64 -mca btl self,sm,openib \
        -mca btl_sm_use_knem 1        \
        -mca mpi_paffinity_alone 1    \
        --bind-to-core                \
        binary <aplication options>
```

The **MPIRUN** Wrapper can be setup in the module file using the following syntax:

```
set-alias  MPIRUN  "mpirun -np \$LOADL_TOTAL_TASKS OPTIONS \$@"
```

Recommended Best Practices

- Use the **SubmitScripts Templates** that are in /share/SubmitScripts.
- Use the **smpexec** for SMP jobs in order to use the core binding capabilities
  `smpexec myapplication argument1 argument2`
- Ask for the minimum nodes for the specified number of cores for the very large jobs. It will take more time to enter in execution but will finish with less walltime.

SubmitScript Example : MPI job using Shared FileSystem

```
#!/bin/bash
# NAMD SubmitScript
# Optimized for run parallel job of 512 Cores at NeSI (Pandora-SandyBridge)
#########################################################################
#@ job_name = NAMD_TEST
#@ class = default
#@ group = nesi
#@ notification = never
#@ account_no = uoa
#@ wall_clock_limit = 00:59:00
#@ 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/test/NAMD/apoa1
#@ total_tasks = 512
#@ queue
#########################################################################
###  Load the Enviroment
ulimit -m $((4096*1024)) -v $((4096*1024))
. /etc/profile.d/modules.sh
module load NAMD/2.9-sandybridge
#########################################################################
###  The files will be allocated in the shared FS ($SCRATCH_DIR)
cp -pr /share/test/NAMD/apoa1/* $SCRATCH_DIR
cd $SCRATCH_DIR
#########################################################################
###  Run the Parallel Program
export OMP_NUM_THREADS=1
MPIRUN namd2 apoa1.namd
#########################################################################
###  Transfering the results to the home directory ($HOME)
#cp -pr $SCRATCH_DIR $HOME/OUT/namd
```

Grisu Command Line Interface (grislish)

- `help` - displays help menu
- `help <keyword>` - displays help about a keyword
- `attach <path_to_file>` - attaches a file to the next job
- `set <property> <value>` - sets a value for the next job
- `submit <command_to_run>` - submits a job
- `print jobs` - lists all jobs and their statuses
- `print job <jobname>` - displays details about a job