WORKSHOP:
Using Tizard

Paul Coddington, Deputy Director eRSA

9 March 2016

Tizard User Guides

- Tizard web page
  http://www.ersa.edu.au/tizard
- Tizard User Guide
  http://support.ersa.edu.au/hpc/tizard-user-guide.html
- eRSA User Guides
  http://support.ersa.edu.au/
- Tizard training tutorial
  eRSA User Guides > HPC > HPC Quick Start
  Unix tutorial and Unix cheat sheet
  eRSA User Guides > HPC > Torque PBS queuing system

Tizard

- Tizard is a heterogeneous supercomputer
- Different systems optimised for different tasks
- Total of 40 TFlops
- Cluster of CPU nodes
- Cluster of GPU nodes
  - High-end technical
  - Consumer gaming
- Big memory nodes
- Virtualization server
- CentOS Linux

Tizard CPU cluster

- 48 SGI compute nodes connected by a high-speed Infiniband network
  - 28 nodes are shared access for all users
- Each node has:
  - 48 cores (4 AMD 6238 12-core 2.6Ghz CPUs)
  - 128GB memory (2.7GB per core)
- A total of 2304 cores and 24 TFlops
- General purpose jobs - single processor, multi-core on one node, or multiple nodes.

Tizard Big Memory Nodes

- 1 Dell R910 server
  - 4 Intel Xeon E7-8837 8-core 2.66 GHz CPUs
  - 1TB memory, 3 TB of local scratch disk
- 1 Dell R810 server
  - 4 Intel Xeon E7-4830 8-core 2.13 GHz processors
  - 512GB memory, 1.7 TB of local scratch disk
- For applications that require relatively small numbers of cores and large memory per core.
- Use this is you need > 4GB memory per core.

Tizard GPU Tesla nodes

- Each node has:
  - 4 nVIDIA Tesla M2090 GPUs (6GB GPU memory per card) designed for high-end technical computing
  - 2 Intel Xeon L5630 4-core CPUs, 96GB memory
  - 2.7 TFlops (single precision) from the GPUs
  - 1/2 of this for double precision
- 5 nodes giving 13.5 TFlops total single precision (7 TFlops double precision).
- For applications that have been ported to run on GPUs and need good double precision performance, large GPU memory and error correcting (ECC) GPU memory.

Tizard GPU GTX nodes

- Each node has:
  - 4 nVIDIA GTX580 GPUs (3GB GPU memory per card) designed for consumer gaming
  - 2 Intel Xeon L5630 4-core CPUs, 24GB memory
  - 2.7 TFlops (single precision) from the GPUs
  - 1/4 of this for double precision
- 12 nodes giving 32 TFlops total single precision (8 TFlops double precision).
- For applications that have been ported to run on GPUs and are mostly single precision calculations and don't need large GPU memory or error correcting (ECC) GPU memory.

Tizard Virtualisation Server

- 1 Dell R815 server with
  - 4 AMD Opteron 6128 8-core processors
  - 256GB memory and 3.6 TB disk.
- For hosting virtual machines
- Supports applications that require interactive access (e.g. using a GUI) and/or don't run on the operating system used on eRSA clusters.
- Can now use the cloud for some of this

Running Compute Jobs

- Your software must run on Unix
  - Option to use cloud or VM server if Windows
- You can compile and run your own programs
- or use programs that are already installed
- You need to know (or figure out from user guide) how to run the software
- Think about how to run jobs using multiple processors (we can give guidance)
- Then make a script to run the software in *batch mode* on the supercomputer

Application Software

- Some of the software available on Tizard:
  - **Chemistry**: Gaussian09, NWChem, Terachem, NAMD, Amber, Dalton
  - **Bioinformatics**: Blast, BEAST, MrBayes, Paup, ClustalW2, RepeatMasker, Bowtie, STACKS, R, etc etc.
  - **Engineering & Maths**: OpenFOAM, R, Matlab
- To find all software installed, type

```
module avail
```

- Other software can be installed on request to the eRSA helpdesk.
- User must purchase (network) license for commercial licensed software

Compilers

- If you write your own code it needs to be compiled and linked against system libraries.
- GNU Compilers, gcc, g77, g++
- Intel compiler suite, icc, ipCC, ifort. Includes Intel's MKL with optimised BLAS, LAPACK, PARDISO libraries. Usually best performance.
- OpenMPI for parallel MPI programs.
- CUDA development environment for GPUs.

Software Libraries

- OpenMPI (MPI library)
- Common maths libraries
  - fftw
  - lapack, scalapack, atlas
- Other libraries
  - hdf5
  - guile
  - oomph
  - bioperl, biopython

Others can be installed on Tizard by request to the helpdesk.

Logging in to Tizard

- To use Tizard you first log in to one of the two *head nodes* using ssh
- You need an ssh client such as PuTTy for Windows or native ssh for Linux or Mac.

```
ssh
username@tizard1.ersa.edu.au
```

- Use tizard1 for job submission or compiling.
- Use tizard2 for compiling and job submission to mecheng nodes
- Can run short (<10mins) test jobs directly on the head nodes

Queueing System

- To ensure efficient and fair use of resources all jobs run on eRSA's facilities need to be submitted as batch jobs to a *queuing system*.
- This is standard practice in HPC centres.
- The queuing system we use is Torque, a variant of the PBS queuing system.
  - very similar to the system run by NCI and Pawsey
  - similar to other HPC clusters e.g. Phoenix, Colossus
- Note that you can't (easily) run jobs interactively, they must be specified in a script so they run automatically when the resources they need (memory and CPU cores) become available.

Overview of Torque

- Jobs are allocated dedicated compute resources (CPUs, memory) on the cluster.
- Users request the resources in a Torque *job script*, a Unix shell script with formatted comments to specify Torque attributes.
- The job script is submitted to the Torque batch queue, where it waits until the required resources are available.
- The *scheduler* (we use one called Maui) allocates resources (nodes, cores, memory) to a job
- Torque runs the job on the appropriate nodes.

Example job script - sequential

```
#!/bin/tcsh
### Job name
#PBS -N MyJobName
### Output files
#PBS -j oe
### Mail to user when job ends or aborts
#PBS -m ae
#PBS -M fred.bloggs@ersa.edu.au
### Queue name
#PBS -q tizard
### Number of nodes, memory, walltime.  REQUIRED
#PBS -l nodes=1:ppn=1
#PBS -l mem=Xmb,vmem=Ymb
#PBS -l walltime=01:00:00
cd $PBS_O_WORKDIR
# Load modules if required
module load application
# Run the executable
applicationExe < InputFile.dat > OutputFile.log
```

Example job script - MPI

```
##!/bin/tcsh
### Job name
#PBS -N MyJobName
### Output files
#PBS -j oe
### Mail to user when job ends or aborts
#PBS -m ae
#PBS -M fred.bloggs@ersa.edu.au
### Queue name
#PBS -q tizard
### Number of nodes, memory, walltime.  REQUIRED
#PBS -l nodes=N:ppn=P
#PBS -l mem=Xmb,vmem=Ymb
#PBS -l walltime=01:00:00
cd $PBS_O_WORKDIR
echo Using nodes
cat $PBS_NODEFILE
# Load modules if required
module load application
# Run the executable
mpirun -np NP applicationExe < InputFile.dat > OutputFile.log
```

Parameters to job script

```
### Number of nodes, memory, walltime.
#PBS -l nodes=N:ppn=P
#PBS -l mem=Xmb,vmem=Ymb
#PBS -l walltime=01:00:00
```

- The parameters in red need to be specified for your job
- What should you set them to be?

Walltime

```
### Number of nodes, memory, walltime.
#PBS -l walltime=01:00:00
```

- Walltime is an estimate of the time your job should take to run (HH:MM:SS)
- **Don't underestimate** – if the job takes longer than the specified walltime Torque will kill it!
  - Checkpoint if you can
- Don't overestimate too much – shorter jobs are likely to run (be scheduled) earlier
- There is a max walltime – 100 hours on Tizard

Number of processors

```
### Number of nodes, memory, walltime.
#PBS -l nodes=N:ppn=P
```

- N is the number of nodes
- P the number of processors per node
  - N <= 28 and P <= 48 on Tizard
- Total number of processors is N x P
  - Make sure you get this right when you specify the number of processors to your program!
- Check that your application can actually make use of multiple processors before putting P>1

Number of processors

```
### Number of nodes, memory, walltime.
#PBS -l nodes=N:ppn=P
```

- Check that your application can actually make use of multiple nodes before putting N>1!
  - Usually this means it's an MPI program
  - More nodes often easier to schedule
- Don't use more processors than your applications can effectively use.
- Start with small numbers of processors and increase, compare the execution time, check that you are getting speedup with more processors

Memory

```
### Number of nodes, memory, walltime.
#PBS -l mem=Xmb,vmem=Ymb
```

- Read the user guide for your application to see if you can estimate memory requirements
- If not, run a test job with mem 2GB per processor
  - vmem is tricky – just set Y to be 2X
- If the program fails, double mem and try again
- If the program runs, check the PBS output file which will tell you how much mem and vmem were actually used, and use that (but may change with problem size)
- If you need >4GB per processor use Big Memory node

Submitting a job

To submit your job simply type:

qsub <shellscript>

where shellscript is the name of your Torque job submission script file, eg

```
qsub submit.pbs
```

You will get a response something like:

```
Job submitted.
Torque JobId: 28195.tizard
```

Checking status of a job

To view the status of your job simply type:

qstat –a

```
tizard:
                                                                  Req'd  Req'd   Elap
Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
25016.tizard     sandery  seq      sef8        21497   1  --     --  1000: E 627:1
28167.tizard     hongyi   seq      newtest64s  29764   1  --     --  2440: R 02:03
28168.tizard     hongyi   seq      newtest64s    --    1  --     --  2440: Q   --
28193.tizard     honcao   seq      xcomp13     24585   1  --     --  1000: R 29:47
28195.tizard     honcao   seq      xcomp15     32087   1  --     --  1000: S 29:47
28196.tizard     honcao   seq      xcomp16     29707   1  --     --  1000: R 29:47
28201.tizard     honcao   seq      rxttfunc3    4870   1  --     --  1000: R 27:52
```

To see other options to qstat consult the Unix manpage.

```
man qstat
```

Delete a job

To delete a job from the queues simply type:

qdel <JobId>

where the JobId is the numeric part of the job identifier given to your job by the queuing system, eg

```
qdel 28195
```

you can find the JobId from qstat

Queues

- The queueing system on Tizard has several different queues for different types of jobs.
  - tizard : CPU nodes (the default queue)
  - gpu : GPU nodes
  - bigmem : big memory nodes
  - short : jobs with walltime < 5 hours and <= 16 cores
  - workshop : queue for this training workshop

- Specify the queue you want in your job script.
- For any Torque command specify a queue using –q *queuename*

Modules

- The Environment Modules package provides for the dynamic modification of a user's environment via module files.
- Better than static settings, e.g. in .cshrc.
- Each module file contains the information to configure the environment for an application.
- Modules can be loaded and unloaded dynamically and automically.
- Modules are useful for managing different versions of applications, or multiple applications with conflicting paths or environment variables.
- Look at the HPC user guides for more information.

Module commands

- **module avail** – what modules are available
- **module list** – what modules are loaded
- **module show** – what environment will be set
- **module load** – load the module
- **module unload** – unload the module
- **module whatis** – other information including what other modules need to be loaded (e.g. compiler for MPI)

File systems

- User files are stored in a large, scalable, high-speed storage system.
- Accessible on the supercomputers via NFS.
- /home/users/username - home directory
- /scratch – large scratch area of high speed disk, available on all compute nodes
- /tmp – fast local temp space on each node
- Please delete files from /scratch and /tmp after your job has finished.
- Home directory backed up, but not scratch, tmp

Copying files

- Copying files to and from eRSA file system and your PC (or elsewhere) can be done using a few protocols
  - sftp, scp, rsync
- Can use simple GUI drag-and-drop client programs
  - WinCSP, Filezilla, Cyberduck, etc
  - See eRSA sFTP user guide for more information
- Different tradeoffs for convenience, ease of use, speed, firewalls.
- Note that text files copied from Windows PCs will have extra ^M characters under a Linux file system that can mess up some things, **including Torque job scripts**.
  - Use the dos2unix command to fix this.

Storage

- Each HPC user gets 200GB by default
- Users, research groups, Schools can purchase additional storage (at cost)
- RDSI provides large amounts (Pbytes) of additional storage for data collections, or developing or analysing data collections
- Currently at no cost to researchers, with allocation process requiring uni approval
- Can be mounted (NFS) on Tizard or Emu

File I/O

- Performance of some applications can be limited by speed of file I/O (input/output).
- In this case, try copying job files (job script, input files) to /scratch and submit job from there, then copy output to home directory.
- If writing temporary files (e.g. Gaussian) use local /tmp on the node.
- Doing lots of small reads/writes and/or reading/writing lots of small files is inefficient and puts high load on file server. Try to aggregate them.

GPUs

- Consumer computer gaming market has driven huge performance increases in GPUs, which now have higher performance than CPUs.
- Modern GPUs are like parallel computers on a chip.
- nVIDIA Tesla M2090 GPU has 512 cores and peak performance of 1.3 TFlops single precision, 0.66 TFlops double precision.

GPUs

- But GPUs have a specialised architecture and programming model, so programs need to be rewritten for GPUs.
- Many applications have now been ported to GPUs.
- Some applications run very well on GPUs and can scale across multiple GPUs.
- However some give little or no performance benefit over a many-core compute node.
- Your mileage may vary – check speedups for the application you are interested in.

Applications on GPUs

- Terachem
- OpenFOAM
- BeagleBEAST
- NAMD
- Amber
- Matlab
- LAMMPS
- Many others

- Some are installed on Tizard already. Ask the helpdesk if there are others you want installed.

eRSA

Advancing Research Innovation

Upcoming workshops

Running a Virtual Machine in the cloud
14 April, University of Adelaide

R-Studio in the Cloud
15 April, University of Adelaide

Talk to us after the workshop to register