Introduction to Artemis
Changes since the upgrade

Presented by
Dr. Stephen Kolmann
Information and Communications Technologies
Allocation of compute nodes

Existing Artemis
1512 cores

Upgraded Artemis

All users 3016 cores

Civil Engineering: 416 cores

Strategic allocations: 832 cores
## Artemis Compute Nodes

### Existing Artemis Nodes

<table>
<thead>
<tr>
<th></th>
<th>Standard Memory</th>
<th>High Memory</th>
<th>GPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Nodes</td>
<td>56</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Cores per node</td>
<td>2 x 12</td>
<td>2 x 12</td>
<td>2 x 12 CPUs 2 GPUs</td>
</tr>
<tr>
<td>RAM per node</td>
<td>128 GB</td>
<td>512 GB</td>
<td>128 GB</td>
</tr>
</tbody>
</table>

### New Nodes

<table>
<thead>
<tr>
<th></th>
<th>Standard Memory</th>
<th>High Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Nodes</td>
<td>80</td>
<td>3</td>
</tr>
<tr>
<td>Cores per node</td>
<td>2 x 16</td>
<td>4 x 16</td>
</tr>
<tr>
<td>RAM per node</td>
<td>128 GB</td>
<td>6 TB</td>
</tr>
</tbody>
</table>
Job Scheduler Upgrade

“Lanes on a highway”
Fair share changes

- Fair Share: Your priority decreases as you use more computation time
- Calculated at the project level
  - Previously was calculated at faculty level
- Fair Share “Half-Life” is 2 weeks
New Job Queues

#PBS -q

defaultQ  small-express  scavenger

- Small
- Normal
- Large
- High Mem
- GPU

Automatically assigned
### "default" sub-queue limits

<table>
<thead>
<tr>
<th>Queue</th>
<th>Max Walltime</th>
<th>Max Cores per User</th>
<th>Memory per node</th>
<th>Fair Share weight</th>
<th>Nodes used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>1 day</td>
<td>96</td>
<td>&lt;125GB</td>
<td>10</td>
<td>Existing</td>
</tr>
<tr>
<td>Normal</td>
<td>7 days</td>
<td>96</td>
<td>&lt;125GB</td>
<td>10</td>
<td>New</td>
</tr>
<tr>
<td>Large</td>
<td>21 days</td>
<td>600</td>
<td>&lt;125GB</td>
<td>10</td>
<td>New</td>
</tr>
<tr>
<td>High Mem</td>
<td>7 days</td>
<td>192</td>
<td>125 GB to 6TB</td>
<td>50</td>
<td>New/Existing</td>
</tr>
<tr>
<td>GPU</td>
<td>7 days</td>
<td>24</td>
<td>&lt; 128GB</td>
<td>50</td>
<td>Existing</td>
</tr>
</tbody>
</table>
small-express and scavenger queue

<table>
<thead>
<tr>
<th>Queue</th>
<th>Max Walltime</th>
<th>Max Cores/User</th>
<th>Fair Share weighting</th>
</tr>
</thead>
<tbody>
<tr>
<td>small-express</td>
<td>12 hours</td>
<td>96</td>
<td>50</td>
</tr>
<tr>
<td>scavenger</td>
<td>2 days</td>
<td>288</td>
<td>0</td>
</tr>
</tbody>
</table>

IMPORTANT: Jobs in the scavenger queue can be terminated before they complete. Use at your own risk!
Strategic Allocations

Each allocation has their own queue:

- #PBS -q alloc-dh
  - Owner: Prof. David Hensher, 192 cores (6 new nodes)
- #PBS -q alloc-jr
  - Owner: Prof. John Rasko, 96 cores (3 new nodes)
- #PBS -q alloc-nw
  - Owner: Dr. Nicholas Williamson, 96 cores (3 new nodes)
- #PBS -q alloc-am
  - Owner: Dr. Alejandro Montoya, 288 cores (12 existing nodes)
- #PBS -q alloc-md
  - Owner: A/Prof. Meredith Jordan, 160 cores (5 new nodes)
Civil Engineering queue

#PBS -q condo-civil
- 416 cores (13 new nodes)
Interactive access

- Two dedicated nodes for interactive access!
- Gain interactive access via the command line:
  
  ```
  $ qsub -I -P ProjectName -l select=1:ncpus=1:mem=16GB,walltime=1:00:00
  ```

- High Memory and GPU nodes can’t be used interactively.
- High Fair Share weight (weighting is 100). Use sparingly.
Example PBS script

#!/bin/bash
#PBS -l select=3:ncpus=16:mem=32GB:mpiprocs=16
#PBS -l walltime=24:00:00
#PBS -q defaultQ

module load intel-mpi

cd $PBS_O_WORKDIR
<commands to run job>

All jobs must specify memory and walltime. Jobs which exceed memory/walltime limits will be terminated.
GPU PBS script

#!/bin/bash

#PBS -l select=1:ncpus=1:ngpus=1
#PBS -l walltime=24:00:00
#PBS -q defaultQ

module load intel-mpi
module load cuda

cd $PBS_O_WORKDIR
<commands to run GPU job>

Ask for GPUs like this. It will automatically be placed in the GPU queue.
Job Accounting

- A resource usage file is written at the end of every job
- Generic resource usage file format:
  - `<PBS Script Name>.o<Job ID>_usage`
- Example file for a PBS script called “rundmc”:
  - Filename is “rundmc.o701491_usage”

Job Id: 701491.mgmt1 for user skol2049 in queue short
Job Name: rundmc
Project: RDS-ICT-V3TEST-RW
Walltime requested: 00:01:00 : Walltime used: 00:01:16
  Cpus requested: 1
  Mem requested: 2gb : Mem used: 1206768kb
  Pmem requested: - : VMem used: 1531820kb
  Cpu Time: 00:01:17 : Cpu percent: unavailable
    : Cpu utilization: unavailable
Summary

- Update your PBS scripts
  - Add memory, check queue
- Optimise your memory and walltime requests.
- Use small-express and interactive queues to test programs.
- Talk to me for assistance using high memory nodes!
  - Look me up in the university staff directory!
- Or, request assistance via IT self-service portal: HPC request.
- Use Artemis HPC Users Yammer page! Yammer link:
  - https://www.yammer.com/sydney.edu.au/#!/threads/inGroup?type=in_group&feedId=6833468&view=all
Logging into Artemis

Access to Artemis has not changed.

- Need to be on campus, or logged into the University VPN to access Artemis.
- If using MacOS or Linux, open a terminal and type:
  \$ ssh <UniKey>@hpc.sydney.edu.au
- If you’re using Windows, you can use the PuTTY program to login.
Artemis File Systems - Lustre

Filesystems are unchanged.

- `/project` is a dedicated allocation for your project
  - Temporary storage for “output”

- `/scratch` is available for anyone to use.
  - Intended for data to be stored while job runs
  - Delete/move your files from `/scratch` when your job finishes.

- `/home` is a small storage area for important data
  - Eg sourcecode, programs, important results.
  - Backed up in case of HDD failure only

<table>
<thead>
<tr>
<th>Storage Space</th>
<th><code>/project</code></th>
<th><code>/scratch</code></th>
<th><code>/home</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage Space</td>
<td>1 TB</td>
<td>117T (shared by everyone)</td>
<td>10 GB</td>
</tr>
<tr>
<td>Backed up?</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
Data Storage

Artemis
/home
/scratch
/project

- Short-term data storage for output from compute jobs

RCOS
- Linux filesystem for storing important data.
- Backed up regularly.

Classic RDS
- Windows file system for storing important data
- Backed up regularly
Data Storage

Artemis
/home
/scratch
/project

/home
• Storage of programs, input files
• Not intended for storage of large data.

/project
• Storage for large data output from computations
• Intended to be short term
• Important data should be saved to RCOS

/scratch
• "Scratch" space for data created during a job.
• Delete data before terminating job.
Data Storage

RCOS

- Linux server for storing important research data
- Mounted on Artemis:
  - /rds/PRJ-<ShortProjectName>
  - E.g. /rds/PRJ-PANDORA
- Backed up regularly
- Intended for storage of important research data
- Amount of storage available was specified in your RDMP.

- To copy data from RCOS to your own computer, use SFTP:
  - `sftp <UniKey>@rcos.sydney.edu.au`

IMPORTANT:
- Store data in /rds only:
- /rds/PRJ-<ShortProjectName>
- E.g. /rds/PRJ-PANDORA
Data Storage

Classic RDS

- Windows server for storing important research data
- `\research-data.sydney.shared.sydney.edu.au<Volume>`
- Backed up regularly
- Intended for storage of important research data
- Amount of storage available was specified in your RDMP.

- Accessible as a network drive from Windows/MacOS.
- Accessible via `smbclient` from Linux.