



INTERSECT

# Intermediate HPC

Introduction to Unix for HPC

# What is HPC?

- HPC, or high-performance computing, refers to the application of supercomputers or clusters of computers to computational problems that typically arise through scientific inquiry
- HPC is useful when a computational problem:
  - **Is too large** to solve on a conventional laptop or workstation (because it requires too much memory or disk space) or
  - **Would take too long** (because the algorithm is complex, the dataset is large, or data access is slow) or
  - **Are too many** – High Throughput Computing

# Parallelism on HPC

- HPC systems often derive their computational power by exploiting parallelism
- Programs for HPC systems must be split up into many smaller “sub-programs” which can be executed in parallel on different processors
- HPC systems can offer parallelism at a much larger scale, with 100’s or 1000’s, or (soon) even millions of tasks running concurrently.
- Writing parallel software can be challenging, and many existing software packages do not already support parallelism & may require development.  
**NOTE: Many tasks cannot be parallelised**

# Reasons to use HPC

- You have a program that can be recompiled or reconfigured to use optimized numerical libraries that are available on HPC systems but not on your own system.
- HPC applications are already installed on the HPC machines which is a non-trivial task
- You have a "parallel" problem, e.g. you have a single application that needs to be rerun many times with different parameters.
- You have an application that has already been designed with parallelism
- To make use of the large memory available
- Our facilities are reliable and regularly backed up

# When not to use HPC?

- You have a single threaded job which will only run one job at a time (typical of MatLab users)
- You rely on Databases
- You have a lot of data to transfer between your local machine and the HPC on a continuous basis (e.g. per job)
- You need to have a GUI to interact with your program

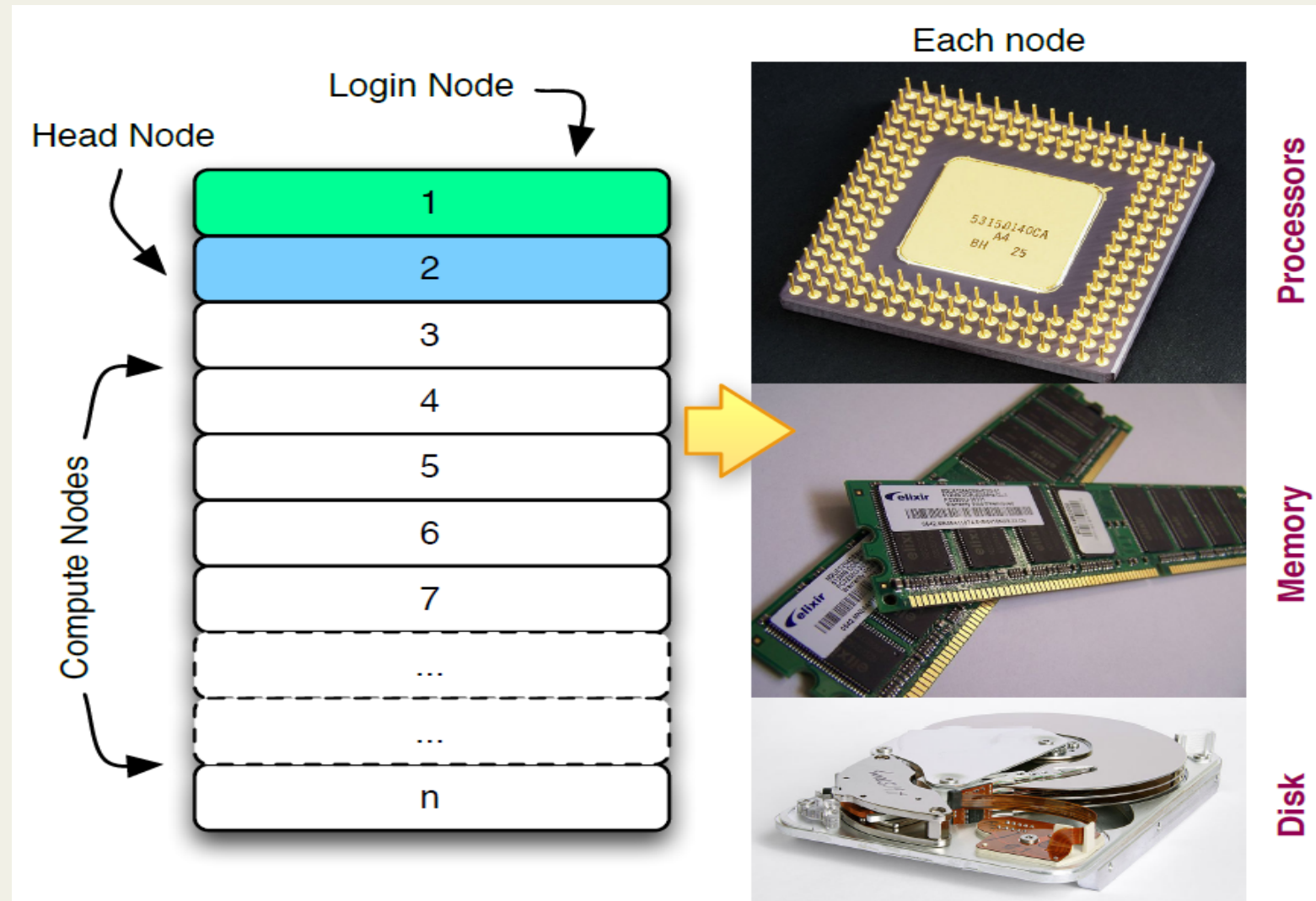
# HPC machines

System	Memory Architecture	Cores	Nodes	Memory
Octane (training machine)	Distributed	48	3	48GB
<b>Orange</b>	<b>Distributed</b>	<b>1,600</b>	<b>100</b>	<b>8TB</b>
<b>NCI – (Raijin)</b>	<b>Distributed</b>	<b>57,472</b>	<b>3592</b>	<b>158TB</b>

# The typical HPC workflow

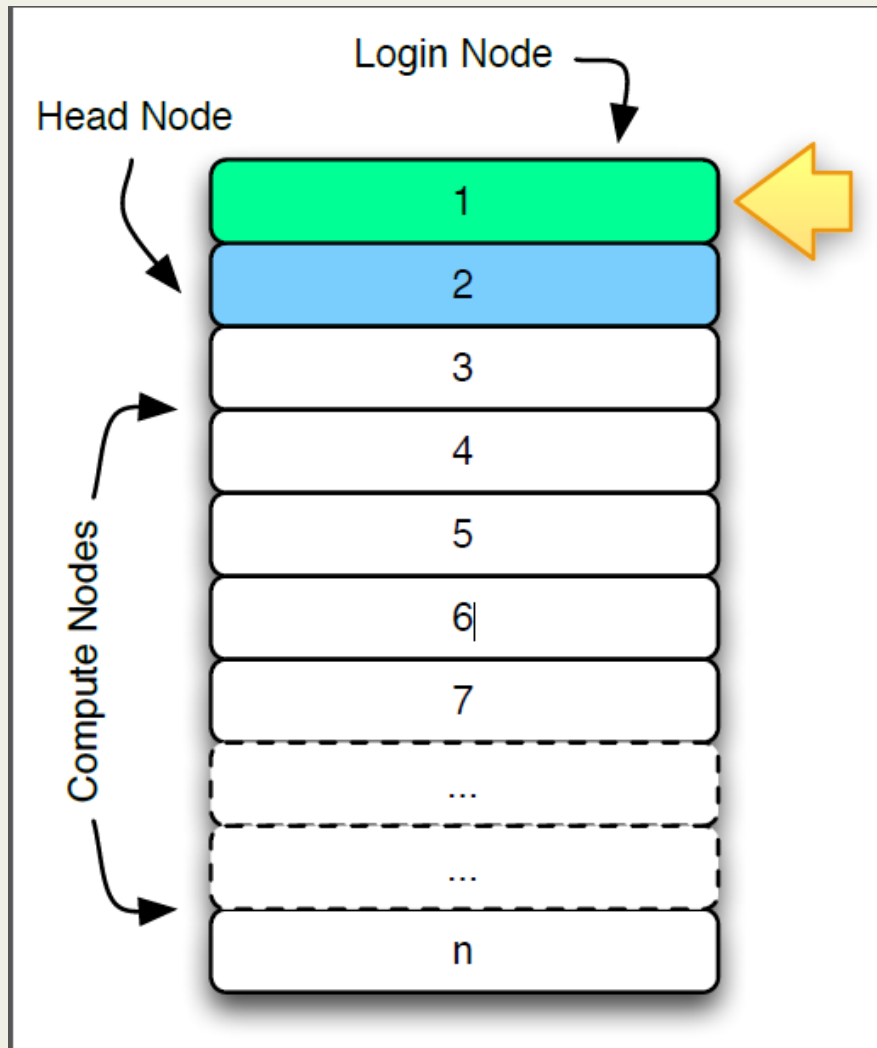
- In HPC we talk about **jobs**, these are simply commands we wish to run and requests for resources (e.g. compute time, disk space, memory requirements, setup of s/w env's etc.)
- Generally time consuming & resource intensive.
- Jobs are typically run **non-interactively**,
- Can be run **interactively** for testing purposes
- We add our jobs to a **queue**.
- When machines have **free resources** jobs run
- Once jobs are complete, we can inspect their output.

# The HPC "Cluster"





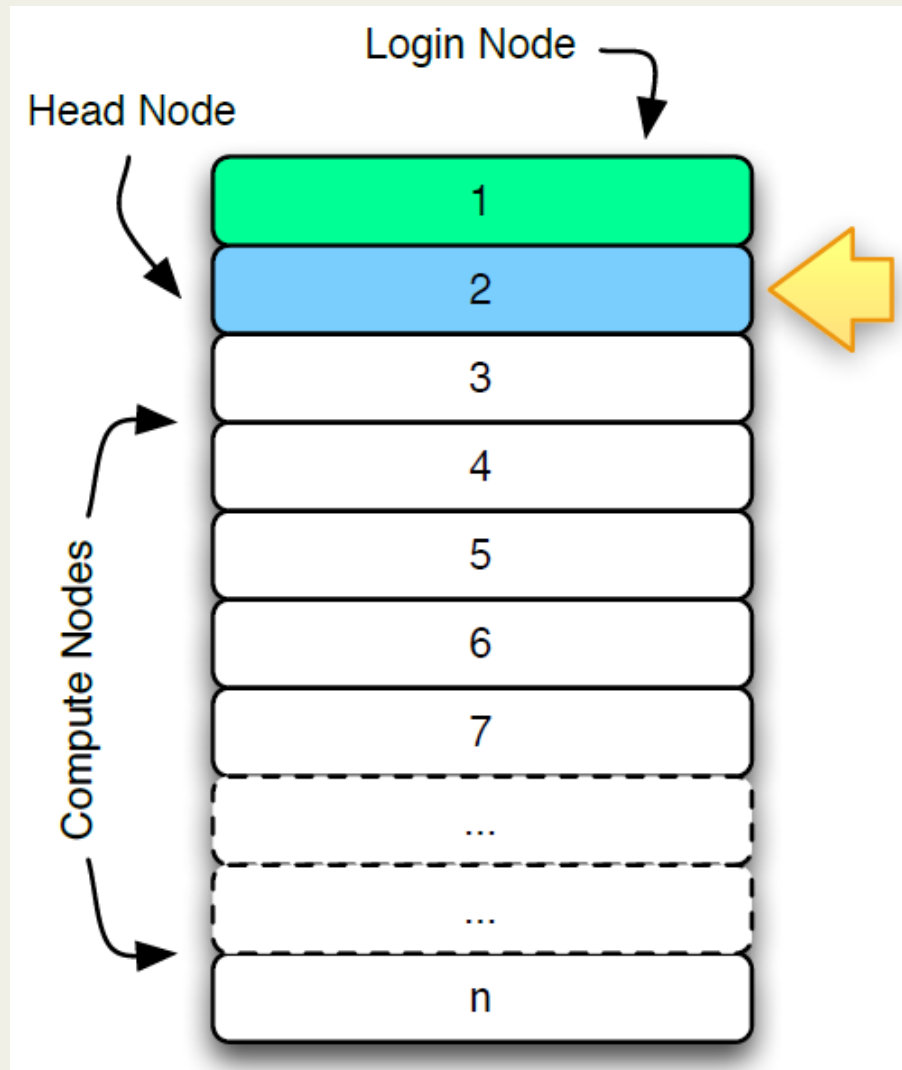
# The Login Node



## Login Node

- Interactive programs
- SSH sessions
- Testing
- Compiling

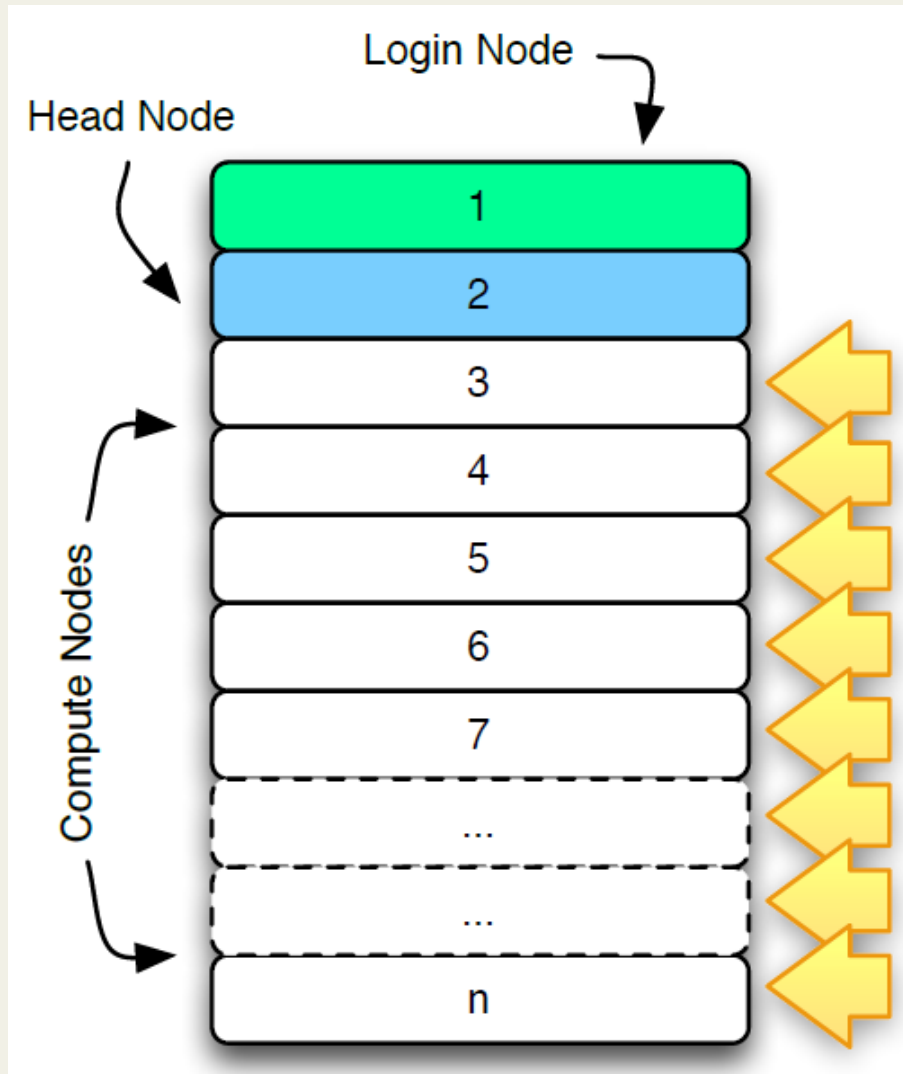
# The Head Node



## Head Node

- Queuing jobs

# Compute Nodes



- These nodes run your jobs
- Managed by the **scheduler**
- Typically you won't interact with the nodes directly
- Some users may need to!

# Queuing Systems

- **Portable Batch System** (PBS) is the name of computer software that performs job scheduling. Its primary task is to allocate computational tasks, i.e., batch jobs, among the available computing resources.
- The following versions of PBS are currently available:
  - OpenPBS
  - TORQUE
  - PBS Professional (PBS Pro)
- Guide to PBS: <http://hpc.sissa.it/pbs/pbs.html>

# Queuing Systems cont.

- Another popular batch system is **SLURM** (Simple Linux Utility for Resource Management )
  - Open source, fault-tolerant, and highly scalable cluster management and job scheduling system for large and small Linux clusters.
  - Very useful for use on clusters
  - Platform Tools used by IBM
  - Used by many supercomputers, e.g. TERA 100 at CEA (Europe's most powerful supercomp.)
- Many banks and commercial entities using batch systems

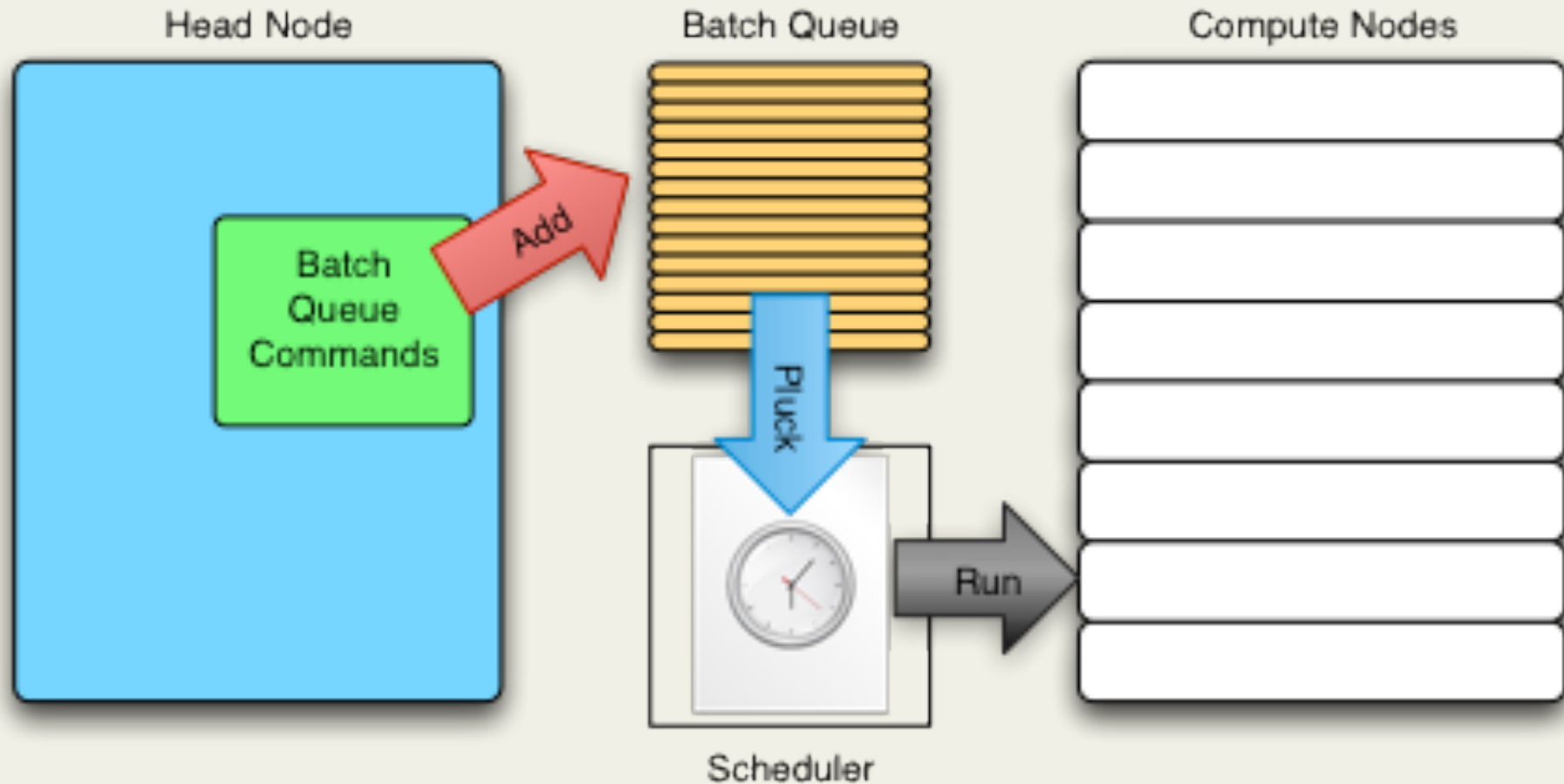
# PBS Pro

**PBS Pro** is the batch system used on all three systems available through your Intersect membership:

- Octane (training machine)
- Orange
- Raijin(NCI)

Batch System	PBS PRO
Machines using	Orange, Octane, Raijin
Code Base	PBS Professional
Licence	Altair Licence

# The Batch Queuing System



# Batch Queuing component

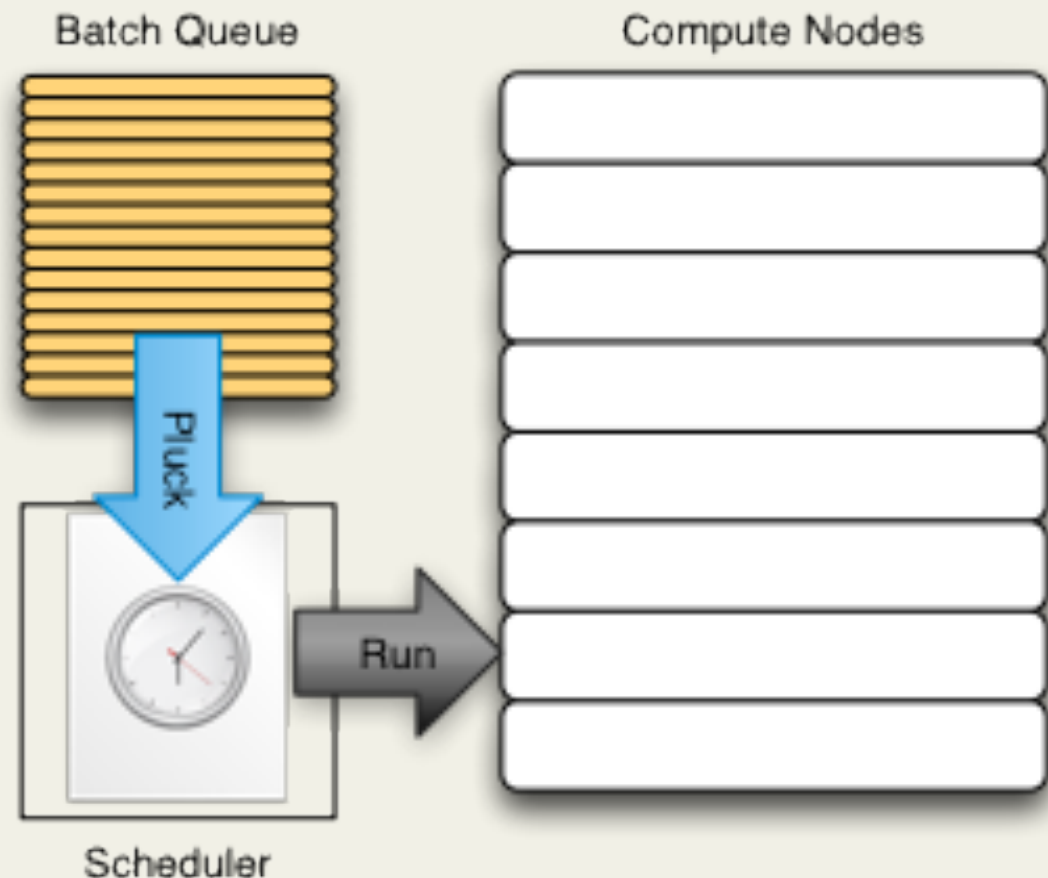


- The batch system is a normal program
- Lets you add and remove jobs from the queue and monitor the queue
- Script/command line driven



# The Scheduler component

- Allocates jobs to compute nodes
- Optimizes usage of resources
- “Optimize” can mean many things
- Non-trivial
- Never interact with directly



# PBS Pro Commands

In order to use the batch system productively, we need to know how to perform three actions:

- Add a job to the queue
- Remove a job from the queue
- See where our job is in the queue

Command	Description
<code>qsub &lt;job-script&gt;</code>	Submit a job (add to queue) Returns a <job-number>
<code>qdel &lt;job-number&gt;</code>	Delete job (remove from the queue)
<code>qstat &lt;job-number&gt;</code>	Monitor jobs
<code>qalter &lt;job-number&gt;</code>	Modifies the attributes of the job or jobs

# Exercise 1

## Monitoring the queue with qstat

Command	Description
<code>qstat -a</code>	List all jobs in the queue
<code>qstat -u &lt;username&gt;</code>	List all jobs of a particular user
<code>qstat -f &lt;job-number&gt;</code>	Show detailed information about a job

# All about Modules

- **What do Modules do?**
  - Set up the environment for a software package
  - Adds paths to executables to \$PATH
  - May change other shell variables, and/or load other modules.
  - Allow you to have different versions of the same software package, e.g. load the Intel compilers v23 with module load intel/12, or if you require an older version, you can load it with module load intel/9

# All about Modules (cont.)

## Things to know about modules

- No default module loaded when you login. You can change this by adding lines to your .bashrc file (BASH users) or similar for other Shells
- Modules exist for many packages but not for all!
- Some modules exclude each other, e.g.
  - You can't load Intel compilers v8 & v9. You can only load one.
  - You can only load 1 MPI (e.g. MPT or Intel or OpenMPI)

# Module Commands

Command	Description
<code>module avail</code>	Will list all available module files in the current MODULEPATH
<code>module load/unload</code>	Will load/unload a modulefile into the shell environment e.g. <code>module load mpt/2.06</code>
<code>module list</code>	List all loaded modules
<code>module show [modulefile]</code>	Will show information about the modulefile

# Add a job to the queue

- To add a job to the queue, we write a **job script**. A job script is simply a script.
- The `#` symbol signifies a comment for the Shell
- It has some special comments that pass info to PBS Pro.
- **#PBS** is a keyword for PBS and specifies that this line is for PBS. The Shell will ignore it
- When we want to queue the job, we pass its filename as a parameter to **qsub**, e.g.
  - **qsub <job-script>**
- The batch queuing system will return a number that uniquely identifies the job.

# Useful Environment Variables

- These are available in the context of your job script.

Command	Description
PBS_O_WORKDIR	The directory the job was submitted from
PBS_JOBID	The job number given when the job was submitted



# A sample PBS job script for Octane

Note the **"-1"** below is a lowercase letter "L" not a one "1"!

```
#!/bin/bash
# Request resources
# * 10 minutes wall time to run
#PBS -l walltime=00:10:00
# * 1 node, 1 processor
# * 100 megabytes physical memory allocated to job
#PBS -l select=1:ncpus=1:mem=100mb
# Specify a project code (for accounting)
#PBS -P a40
cd $PBS_O_WORKDIR
# Specify the job to be done
date
sleep 10
date
```

# You've got mail!

# Set email address

**#PBS -M nobody@intersect.org.au**

# Send an email when jobs

# begins (b), gets aborted (a)

# and ends (e)

**#PBS -m abe**

# A sample PBS job script for Orange

**A small change is required to the job script in order to make it work with the way resources are accounted for on Orange:**

```
#!/bin/bash
# Request resources
# * 10 minutes wall time to run
#PBS -l walltime=00:10:00
# * 1 node, 1 processor
# * 100 megabytes physical memory allocated to job
#PBS -l select=1:ncpus=1:mem=100mb
# Specify a group (for accounting)
#PBS -W group_list=a40
cd $PBS_O_WORKDIR
# Specify the job to be done
date
sleep 10
date
```

# Exercise 2

Create a script and submit a very simple job

Command	Description
<b>#PBS</b>	<b>#PBS</b> is a keyword for PBS and specifies that this line is for PBS. The Shell will ignore it
<b>#</b>	Signifies a comment for the Shell, e.g. <b># Next line will create a job</b>
<b>qsub</b>	Submit a job to PBS
<b>qstat</b>	List jobs in the queue
<b>cat</b> <i>&lt;filename&gt;</i>	Print a file to the terminal (catenate)
<b>less</b> <i>&lt;filename&gt;</i>	Like <b>cat</b> , but less at a time
<b>nano</b> <i>&lt;filename&gt;</i>	Will open file <i>&lt;filename&gt;</i> in a text file editor
<b>Sample PBS Script</b>	<a href="http://www.intersect.org.au/orange-handbook">http://www.intersect.org.au/orange-handbook</a>

# Exercise 3

Create another script and submit a more realistic sample job

Command	Description
<b>#PBS</b>	<b>#PBS</b> is a keyword for PBS and specifies that this line is for PBS. The Shell will ignore it
<b>#</b>	Signifies a comment for the Shell, e.g. <b># Next line will create a job</b>
<b>qsub</b>	Submit a job to PBS
<b>qstat</b>	List jobs in the queue
<b>nano</b> <filename>	Will open file <filename> in a text file editor
<b>module load</b> <module_name>	Loads a software module on the HPC machine

# Job limits on Orange

- 250 hours of **walltime**
- 64GB of **memory** per standard node, e.g. 128GB for 2 nodes etc.
- 256GB of **memory** per large memory nodes
- **NOTE**: If you grab a node with 64GB, you can effectively use about 60GB as the OS uses memory

# Priorities of Jobs

In order of importance, jobs are prioritised in this order:

1. Resources available to the project
2. Walltime
3. Number of jobs (fair share)

# Best strategy

1. Submit jobs constantly/daily
2. Have about 10-20 jobs in the machine
3. Be realistic with walltime
4. Don't ask for resources you don't need!



# NCI Facilities

Disk Type	Disk Usage
NCI Raijin	<ul style="list-style-type: none"><li>• <b>2 sockets with 8-core CPU's = 16 cores</b></li><li>• 57,472 cores in the compute nodes</li><li>• Approximately 160 TBytes of main memory;</li><li>• Infiniband <b>FDR</b> (fourteen data rate) interconnect</li><li>• Approximately 42 PBytes of usable fast file system (for short-term scratch space).</li></ul>

# NCI Facilities

## Raijin

96.5% of Nodes have 24GB/node

3.2% of Nodes have 48GB/node

0.3% of Nodes have 96GB/node

*plus* a small number of nodes with 1TB/node

# Software on NCI

Area	Software
Computational Chemistry	ABINIT, Amber, CPMD*, GULP*, NAMD*, Molpro etc.
Bioinformatics	AbySS, BEAST, BIOPERL, Cufflinks, MAW, etc.
Math Libraries	ARPACK, BLACS, Boost, FFTW, GSL, MKL, Tao
Statistics & Maths Env's	Maple*, Mathematica*, MatLab*, Octave*, R, Stata*

- Asterisked items indicates that discussion with NCI facility staff is required before use (Licensing issues)
- <http://nf.nci.org.au/facilities/software/index.php>

# Software on NCI

Area	Software
Computational Chemistry	ABINIT, Amber, CPMD*, GULP*, NAMD*, Molpro etc.
Bioinformatics	AbySS, BEAST, BIOPERL, Cufflinks, MAW, etc.
Math Libraries	ARPACK, BLACS, Boost, FFTW, GSL, MKL, Tao
Statistics & Maths Env's	Maple*, Mathematica*, MatLab*, Octave*, R, Stata*

- Asterisked items indicates that discussion with NCI facility staff is required before use (Licensing issues)
- <http://nf.nci.org.au/facilities/software/index.php>

# Orange Physical Disks

- Which of the 3 disks to use and when?

Type	Disk Type	Disk Type	• Disk Usage
1	Panasas	59Tb	<ul style="list-style-type: none"><li>• Parallel global file system</li><li>• All nodes see the Panasas disks<ul style="list-style-type: none"><li>• directly attached to each node</li></ul></li><li>• Very fast for large files</li><li>• Can be slow if you copy MANY small files</li><li>• System director blade creates metadata for each file</li></ul>
2	SGI	50Tb	<ul style="list-style-type: none"><li>• An NFS mounted file system</li><li>• All nodes see the SGI disks<ul style="list-style-type: none"><li>• can only be seen via NFS backbone</li></ul></li><li>• Uses old technology, therefore very robust</li><li>• Scales nicely for clusters up to 100 Nodes (very good for Orange)</li></ul>

# Orange Disks (cont.)

- Which of the 3 disks to use and when?

Type	Disk Type	Disk Type	• Disk Usage
3	Local Scratch	200Tb	<ul style="list-style-type: none"><li>• <u>Exist in each node</u></li><li>• No network is necessary making these the fastest disk</li><li>• If you have a lot of I/O, you should copy your data to here and work here</li><li>• One 2Tb disk in each compute node</li><li>• Is only accessible within that node as it's not attached to any network, therefore cannot be accessed by another node</li><li>• <b><u>NOTE:</u></b> The lifetime of files on this disk is only for the duration of the runtime of the job – The user must copy back results. If not, the next job will erase any files</li></ul>

# Disk Partitions - Orange

/home	
Mounted under:	<b>/home/username</b>
Disk Type	SGI Disks
Size:	60GB default
Backed up:	Yes
Speed:	Intermediate disk (SGI Disks)
Life time:	Permanent

# Disk Partitions - Orange

## /projects/project-name

Mounted under:	<code>/projects/project-name</code>
Disk Type	Panasas Disk
Size:	no default size
Backed up:	Yes
Speed:	High speed
Life time:	Till end of the running year - merit allocation period

- There will also be some “repository space” for large datasets, such as bioinformatics databases



# Disk Partitions - Orange

/data2	
Mounted under:	/data2 on each node
Disk Type	Scratch Disks
Size:	Limit of disk - 2TB
Backed up:	No
Speed:	Fastest
Life time:	Job duration

**Warning:** This partition is shared among users, so can be “filled up” (with other jobs) while your job is running!

# Disk Partitions

You can find out more about the partitions on the HPC machine using the **df** command.

Command	Description
<b>df -h</b>	Show disk free space for all partitions in human readable format

You can find out more about current disk usage, using the **du** command.

Command	Description
<b>du -hs .</b>	Show disk usage of current directory in human readable format

# Quotas

- There is no quota on scratch disks for performance reasons
- The quota on /projects/project-name depends on your allocation
- 60 GB soft limit for /home
- 80 GB hard limit for /home (30 days)

# More Info on NCI & Orange

- Read more about Orange and NCI Facilities
  - <http://www.intersect.org.au/hpc-news>
  - <http://www.intersect.org.au/orange>
  - [http://www.intersect.org.au/nci\\_next](http://www.intersect.org.au/nci_next)
- Sample PBS Script & Info on Orange
  - <http://www.intersect.org.au/orange-handbook>

# Resource Allocation Round

- Merit-based system by which Intersect members can gain access to our HPC facilities
- Applications reviewed by HPC staff (for technical complexity and track record) and the Intersect Resource Allocation Committee (for research merit)
- Applications to Intersect's HPC systems will be made through NCI's forms in October each year
- Applications must be made by Academic Staff at an Intersect member institutions. PhD students can make use of the facilities, the lead CI must be an academic staff member.
- Questions to: [hpc\\_support@intersect.org.au](mailto:hpc_support@intersect.org.au)

# To Apply

Please follow the instructions at:

<http://intersect.org.au/time/merit>

Briefly this involves:

- Registering for a user account
- Completing personal and career profiles
- Registering a new project
- Completing an application for resources for the project

# Applications continued

- If you're unsure about which machine to get access to, email [hpc\\_support@intersect.org.au](mailto:hpc_support@intersect.org.au) who can advise you
- You can add accounts to an existing project also!

# Conclusion

- In this course we have covered
  - the basics of the Unix command line
  - transferring data
  - the specifics of our HPC machine
- As different machines have different PBS systems, scripts that work on Orange may need to be adapted for Raijin (NCI) and vice-versa
- Sample answers to all exercises in this course [are available on GitHub](#)



# Thanks for attending!

- Please complete our **course survey** at:

- <http://svy.mk/18c8dHa>

Any **further questions**, contact us at

- [training@intersect.org.au](mailto:training@intersect.org.au)

- Find out about **upcoming courses** by signing up to our mailing list

- <http://bit.ly/1aZvRqw>