

# HPC Workshop

S. T. Balan, L. Whiteway, T. Wilson, A. Palmese

Department of Physics and Astronomy  
University College London

22 October 2015

# Information on the Web

## This presentation

[https://github.com/Astrophysics-UCL/HPCInfo/blob/master/training/workshops\\_2015/hpc\\_workshop/slides/hpc\\_workshop\\_oct\\_2015.pdf](https://github.com/Astrophysics-UCL/HPCInfo/blob/master/training/workshops_2015/hpc_workshop/slides/hpc_workshop_oct_2015.pdf)

## Splinter on the UCL Astrophysics Wiki

<https://wiki.ucl.ac.uk/display/PhysAstAstPhysGrp/Splinter+User+Guide>

## UCL Research Computing Platforms

[https://wiki.rc.ucl.ac.uk/wiki/Main\\_Page](https://wiki.rc.ucl.ac.uk/wiki/Main_Page)

## DiRAC

<http://www.dirac.ac.uk/>

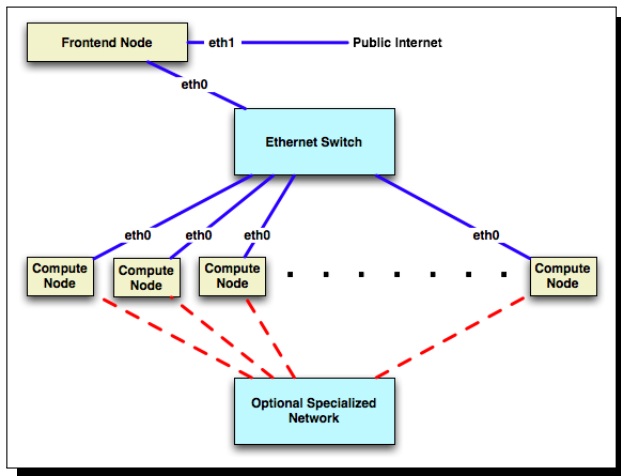
# What will you learn?

- ▶ Running your programs in HPC machines

# HPC Facilities

machine	type	cores	memory
SPLINTER-1	distributed	96	48GB
SPLINTER-2	shared	40	1TB
PHALANX	shared	32	512GB

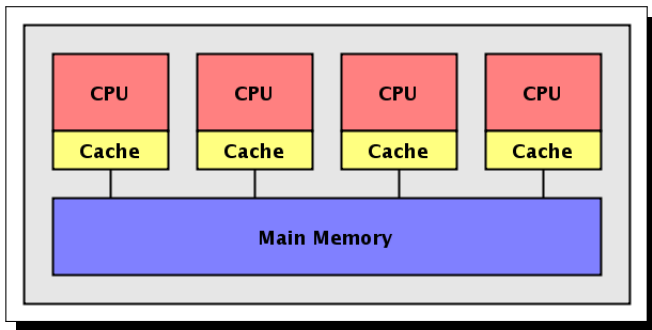
## SPLINTER distributed



1

<sup>1</sup><http://www.rockclusters.org/>

## SPLINTER shared



2

---

<sup>2</sup><http://www.cs.rit.edu/>

# Best practices I

- ▶ Choose the machines that are suited for your problem
- ▶ Read the User Guide
- ▶ Do not run your programs in the login node
- ▶ Do not install common software locally
- ▶ Request optimum resources
- ▶ Minimise data transfer between nodes,
- ▶ **Backup! Backup! Backup!**

# Submitting jobs

## commands

```
qsub jon_script
qsub -I
checkjob job_id
qstat
showq
qdel
```

## Example

```
#!/bin/bash
#PBS -N hello_world_program
#PBS -l nodes=1:ppn=4
#PBS -l mem=2gb
#PBS -j oe
#PBS -V

# source the required scripts
# this sets the PATH
source /home/sbalan/binpaths.sh
# this sets the LD_LIBRARY_PATHS
source /home/sbalan/libpaths.sh

# run my program
/home/sbalan/hello.exe
```



# Exercises I

1. Login to your HCP machine and find the path to your HOME directory and your quota
2. Find the processor type and the version of your operating system
3. Request an interactive queue and run the program `hello_world.exe`
4. Submit `hello_world.exe` using a job script, find its jobid, check the output log.
5. Compile `big_mem_example`, submit it using a job-script and see how much memory it uses
6. Compile `time_pause_example`, submit it using a job-script and kill this job using its jobid.
7. In the previous example see what happens when you play with the time requested.