

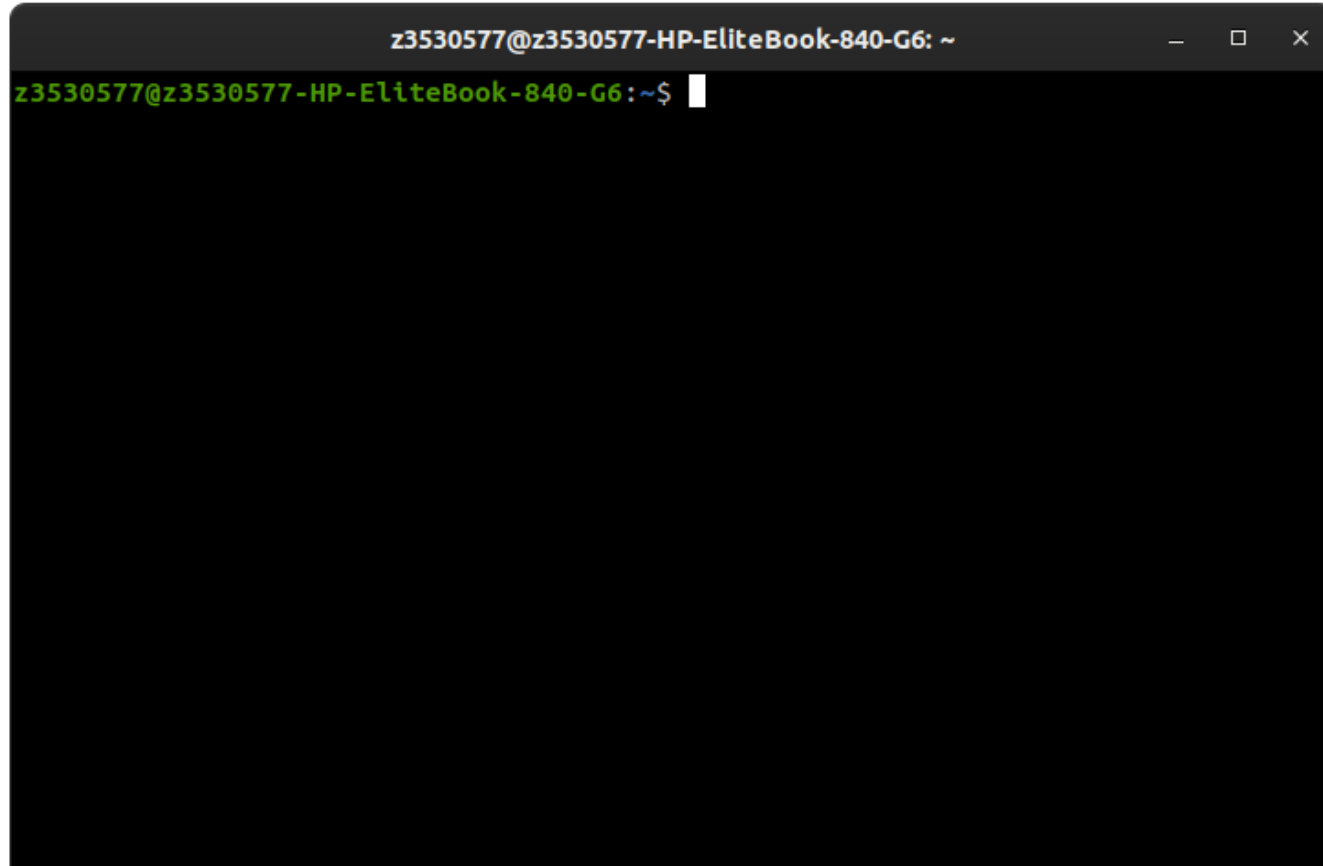
High Performance Computing

An introduction to Supercomputing

What you will learn

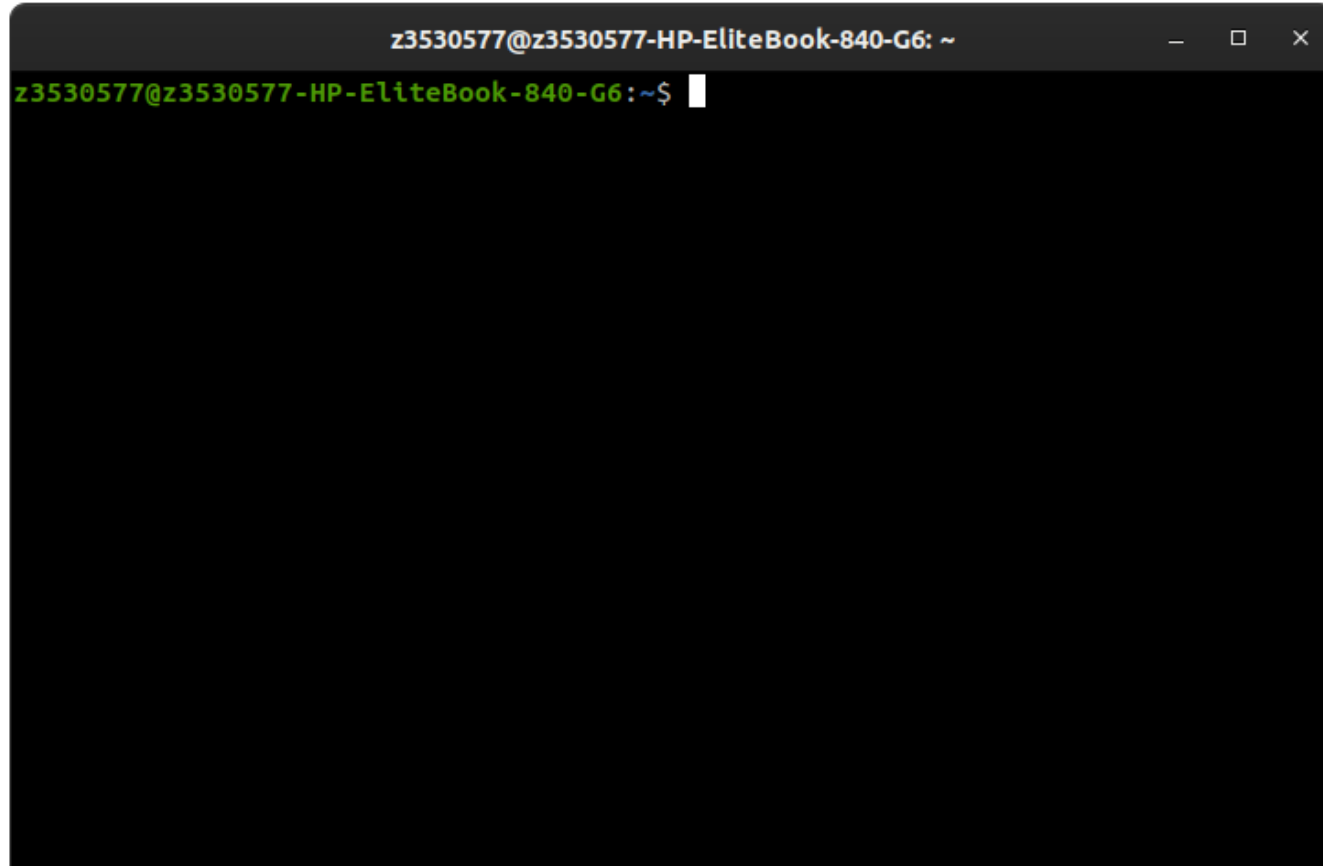
- **Why** you would use HPC
- **Conceptualisation** of a Supercomputer – the systems architecture
- Basics on **how** you will use UNSW's HPC

If HPC is a fancy way of saying
Supercomputer, why does it look like this?

A terminal window with a dark background. The title bar at the top reads "z3530577@z3530577-HP-EliteBook-840-G6: ~" and includes standard window control icons (minimize, maximize, close). The main area of the terminal shows a green prompt "z3530577@z3530577-HP-EliteBook-840-G6:~\$" followed by a white cursor block.

```
z3530577@z3530577-HP-EliteBook-840-G6: ~  
z3530577@z3530577-HP-EliteBook-840-G6:~$
```

If HPC is a fancy way of saying Supercomputer, why does it look like this?

A terminal window with a dark background. The title bar at the top reads "z3530577@z3530577-HP-EliteBook-840-G6: ~". The main area of the terminal shows a green prompt "z3530577@z3530577-HP-EliteBook-840-G6:~\$" followed by a white cursor block. The rest of the terminal is empty.

```
z3530577@z3530577-HP-EliteBook-840-G6: ~  
z3530577@z3530577-HP-EliteBook-840-G6:~$
```

This does not look super.

It looks like more work.

Why you would use HPC

Basic

- Long running jobs or large number of very similar jobs
- Very large data sets
- Doesn't turn off at bedtime
- Rarely needs disruptive reboot

Advanced

- Multi-threaded code
- High speed network and interconnect
- GPU access
- Very high memory access

Why you would use HPC

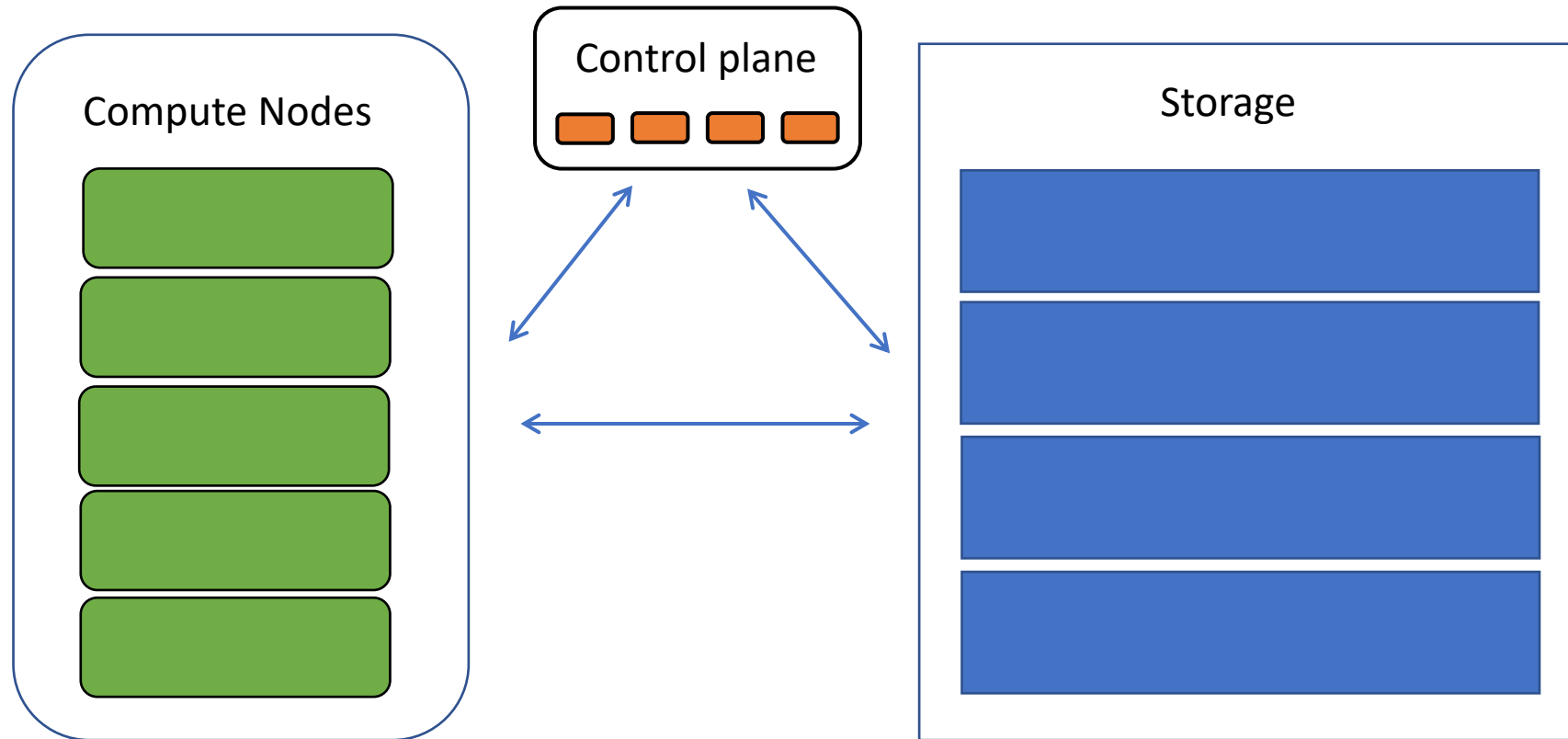
Adjacent Learning

You will need to learn new things that aren't your dissertation subject, but they will be generally useful in the future and you often don't need to go deep.

- The Command Line Interface (aka **cli**)
- Software: reading, writing, running and source control (**git**)
- General IT tools and problem solving
- Data shape, sharing and management
- Community engagement

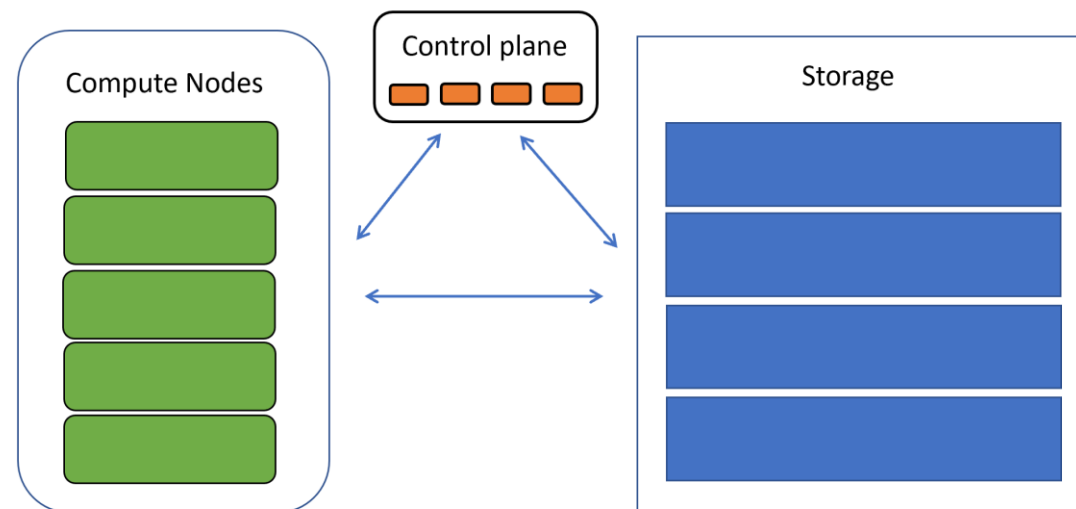
While secondary to your actual topic, these skills will almost certainly be required in your future research or work.

HPC: Conceptualization



HPC: Conceptualization

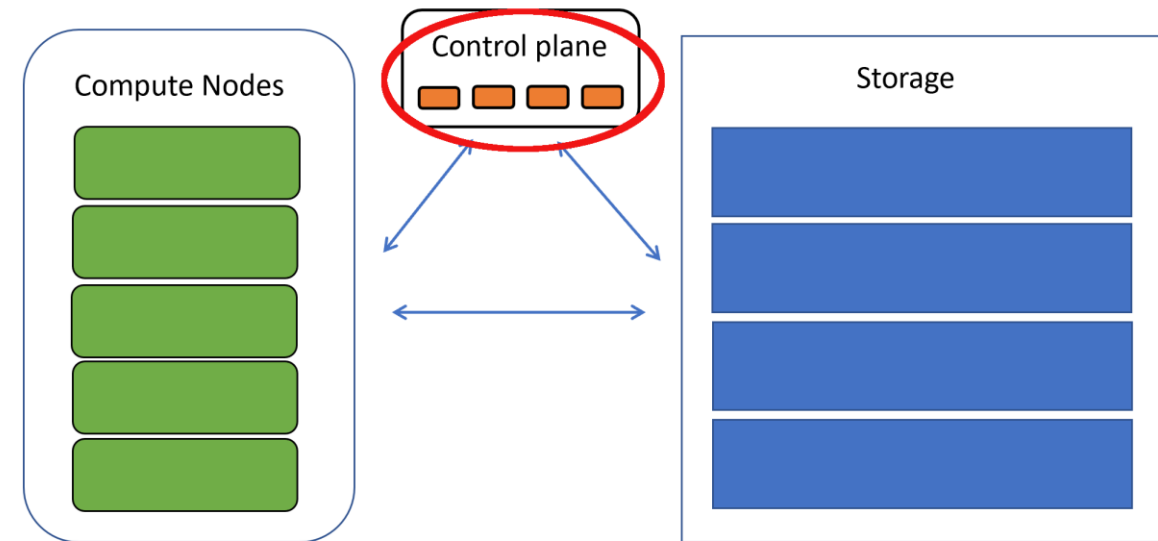
In a traditional computer, these would all be in the same box – a laptop or a desktop or a handheld device. Let's break it down.



HPC: The Control Plane

The Control Plane

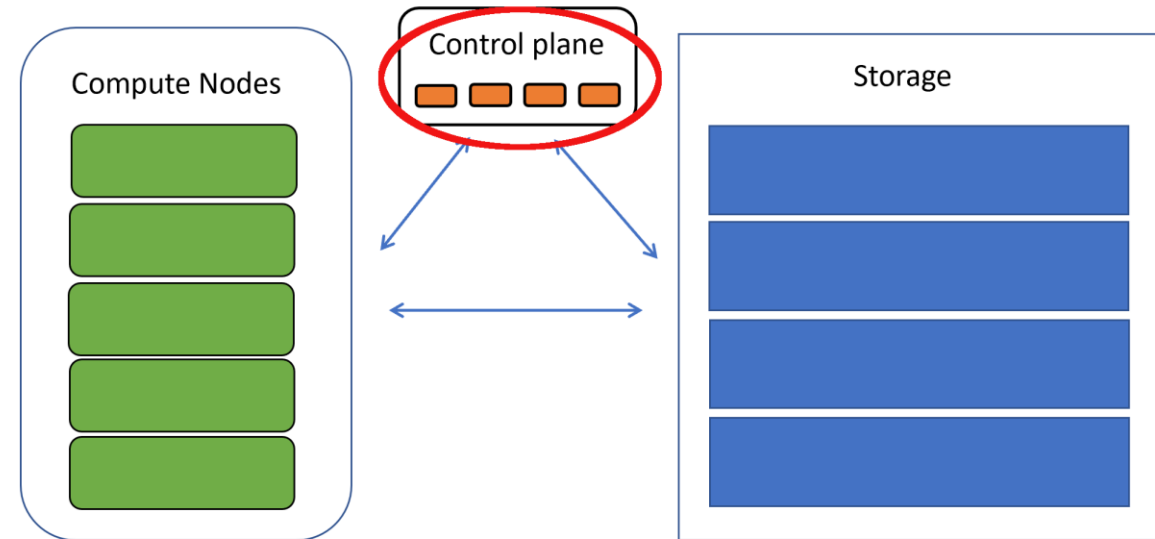
- Collection of servers used to “control” the cluster: the software, login process, data transfers etc.
- Login nodes (katana1, katana2)
- Data management nodes (kdm)
- Administration nodes (no user access)
- No need to remember the name ‘control plane’, it’s just a collective term we use out the back.



HPC: The Control Plane

Login nodes

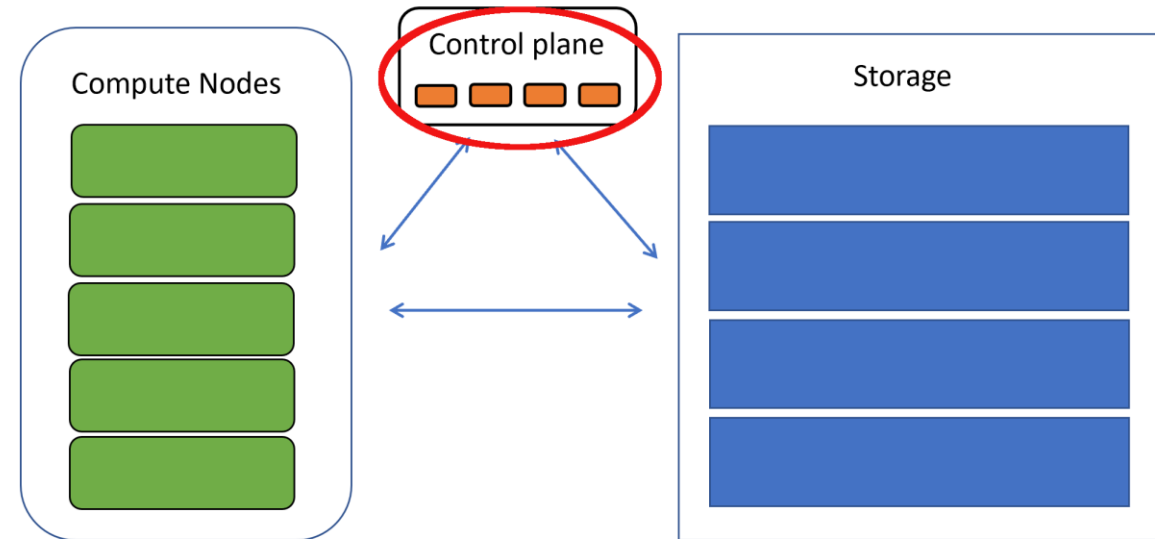
- Names: **katana1.restech.unsw.edu.au** and **katana2.restech.unsw.edu.au**
- Method of access to the HPC
- Many people at the same time (~100)
- Submit jobs to the queues from login nodes
- Different hardware to compute nodes
- We don't run analysis on login nodes



HPC: The Control Plane

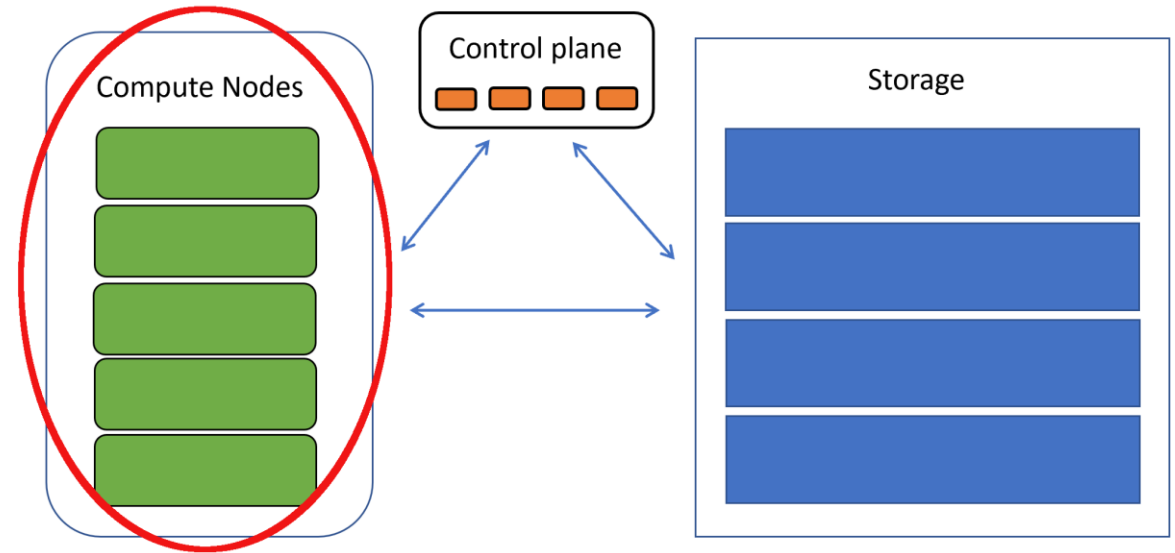
Data Management server

- Name: **kdm.restech.unsw.edu.au**
- Remote access to the HPC
- Many people at the same time (~10)
- Has more memory than login, fewer CPUs
- Specifically designed for moving data around
- Internal and external data movement supported
- Access to the UNSW Data Archive



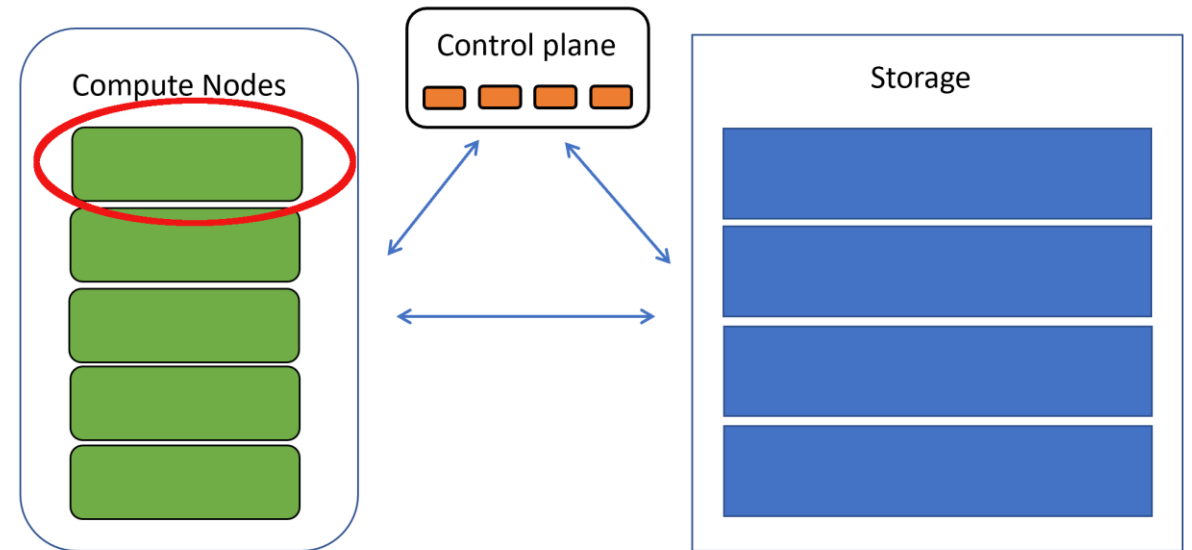
HPC: The Compute Nodes

- ~170 servers
- Grouped into **queues**
- Not usual to login to these individually
- Access via the scheduler (PBSPro)
- Programs run on the nodes
- Interconnected for performance



HPC: A Compute Node

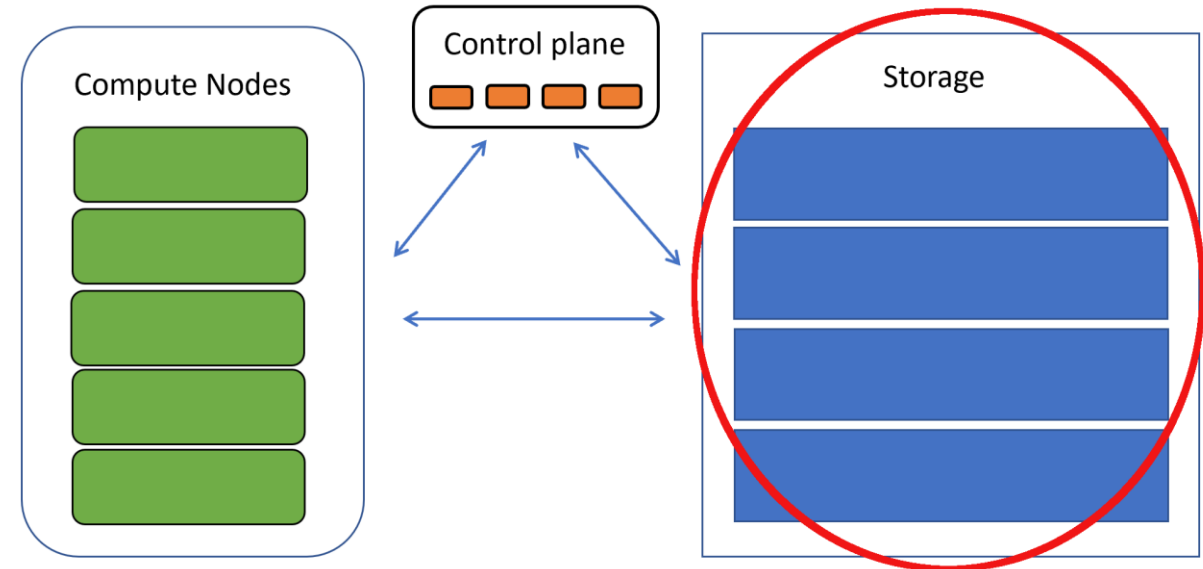
- Variety of brands and models
 - Dell C4140, M620, M630, M640, M710HD, R640
 - Huawei 1288H V5, 2488 V5
 - Lenovo SR570, SR630, SR850
- Between 16 and 44 CPUs per machine
 - Hyperthreading and virtualisation is off
 - 4-10GB memory per CPU, 126-377GB per machine
- Some have special hardware: GPU, Infiniband
- Jobs can run on more than one node at a time



HPC: Storage

- Petabytes of storage shared across whole org
- Shared across all login and compute nodes with same name:

/home/<zid>	Your personal files and programs (10GB)
/scratch/<zid>	Your source, intermediate and result data (hand wavy limit, on 16TB storage)
/apps	Shared programs
/data	Large, shared datasets like the Human Genome or geospatial datasets



Systems

CentOS 7.x Linux

Environment modules

PBSPro

CentOS 7.x

To access

Windows: [Ubuntu in WSL](#) or [Putty](#) or [MobaXTerm](#)

Apple/OSX: Terminal App

Login with ssh

```
ssh <zid>@katana.restech.unsw.edu.au
```

```
ssh <zid>@kdm.restech.unsw.edu.au
```

You will be in your home directory `/home/<zid>`

For more information on using Linux we recommend the [MIT Missing Semester](#) course.

Environment Modules

Not all software is always available to all users. Some researchers will need the bleeding edge Python 3.8, others will be reproducing other's work and need Python 2.7

Modules is how users can control what software they have access to.

module avail	List all the available modules
module avail pyth	List all the available modules that start with the letters "pyth"
module load python/3.7.4	Load the module python/3.7.4.
module list	List the modules the user has loaded
module show python/3.7.4	Show me the details of a module
module help python/3.7.4	Show me the help of a module
module unload python/3.7.4	Unload the Python module

Environment Modules

```
1/1 + [T] [M] Tmux: Default
1: 0:4:ssh - "rtsladm1b.restech.unsw.edu.au"
[z3530577@katana2 ~]$ module avail pyth
----- /apps/modules/languages -----
python/2.7.15      python/3.7.3      python/intel-2.7.12  python/intel-3.5.2
python/3.6.5      python/3.7.4      python/intel-2.7.16  python/intel-3.6.8
[z3530577@katana2 ~]$ which python3
/usr/bin/python3
[z3530577@katana2 ~]$ python3 --version
Python 3.6.8
[z3530577@katana2 ~]$ module load python/3.7.4
[z3530577@katana2 ~]$ which python3
/apps/python/3.7.4/bin/python3
[z3530577@katana2 ~]$ python3 --version
Python 3.7.4
[z3530577@katana2 ~]$
```

PBSPro

This is the system that manages the compute resources available, and the incoming jobs.

When you submit, your job is put into a queue, depending on:

- Faculty or research group
- Time requested (in hours: 12, 100, 200, ...)
- Memory requested (in GB: 124, 180, 248, 370, 750 & 1000)

PBSPro

As resources become available in your queue, your job will be sent to the resources you requested. This can take between 5 mins – 5 days.

```
datakid@vr05: ~  
name,department,birthday month  
John Smith,Accounting,November  
Erica Meyers,IT,March  
~  
~  
~  
~  
~  
~  
birthdays.csv 1,1 All  
import csv  
with open('birthdays.csv') as csv_file:  
    csv_reader = csv.reader(csv_file, delimiter=',')  
    line_count = 0  
    for row in csv_reader:  
        if line_count == 0:  
            colnames = ", ".join(row)  
            print('Column names are {}'.format(colnames))  
            line_count += 1  
        else:  
            print(row)  
            line_count += 1  
    print('Processed {} lines.'.format(line_count))  
~  
data_analysis.py 12,9 Bot  
#PBS -k eod  
#PBS -j oe  
  
cd $PBS_O_WORKDIR  
  
module add python/3.7.3  
  
python data_analysis.py  
~  
~  
SUBMIT.sh 6,1 Bot  
<
```

PBSPro

Submitting and checking the queue use two commands:

qsub and **qstat**

There's more to it, but that is the minimum viable use case.

```
z3530577@katana1:~/test
[z3530577@katana1 test]$ qsub SUBMIT.sh
342403.kman.restech.unsw.edu.au
[z3530577@katana1 test]$ qstat -u z3530577

kman.restech.unsw.edu.au:

Job ID          Username Queue   Jobname  SessID NDS TSK Memory Req'd Req'd Elap
-----
342260.kman.res z3530577 general1 TESPYPHON -- 1 8 128gb 04:00 Q --
342387.kman.res z3530577 maths12  TESPYPHON -- 1 8 128gb 04:00 Q --
342403.kman.res z3530577 maths12  TESPYPHON -- 1 2 2gb 04:00 Q --
[z3530577@katana1 test]$
```

More Information

- [Hacky Hour](#) – real humans solving your problems with coffee (or Zoom or Teams or ...)
- [Service Desk](#) – real humans solving your problems with email (mention Katana in the subject line)
- [Documentation](#) – it's getting a face lift we promise!
- Example [PBSPRO scripts](#) – join the [UNSW github account](#) first