Parallel Computing in R

Morning session : Intro to HPC and the Sulis tier 2 cluster

14th March / University of Warwick

WARWICK
THE UNIVERSITY OF WARWICK

This workshop

- Pilot event
- Funded by UKRI Digital Research Infrastructure programme (phase 1)
- Using HPC Midlands+ ”Sulis” cluster for exercises but much will be applicable to other HPC systems

Format / plan

- Monday AM: Introduction to HPC clusters and Sulis (DQ)
  [Quite general but with R focused examples]

- Monday PM onward: Parallel programming in R (MS)

- Hands-on as much as possible!

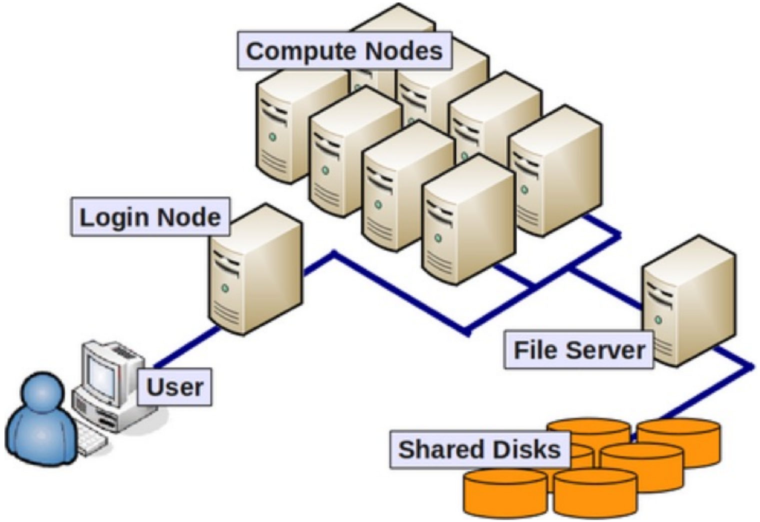
Lunch is at 1-2 every day in the restaurant. Coffee/snacks available in the lounge whenever you need a break.

Morning session

- Introduction to HPC clusters in general and Sulis in particular
- Get signed up and registered with Sulis
- Logging in, moving files to/from the cluster, accessing software
- Submit an example R job from the Sulis website

Before we continue….

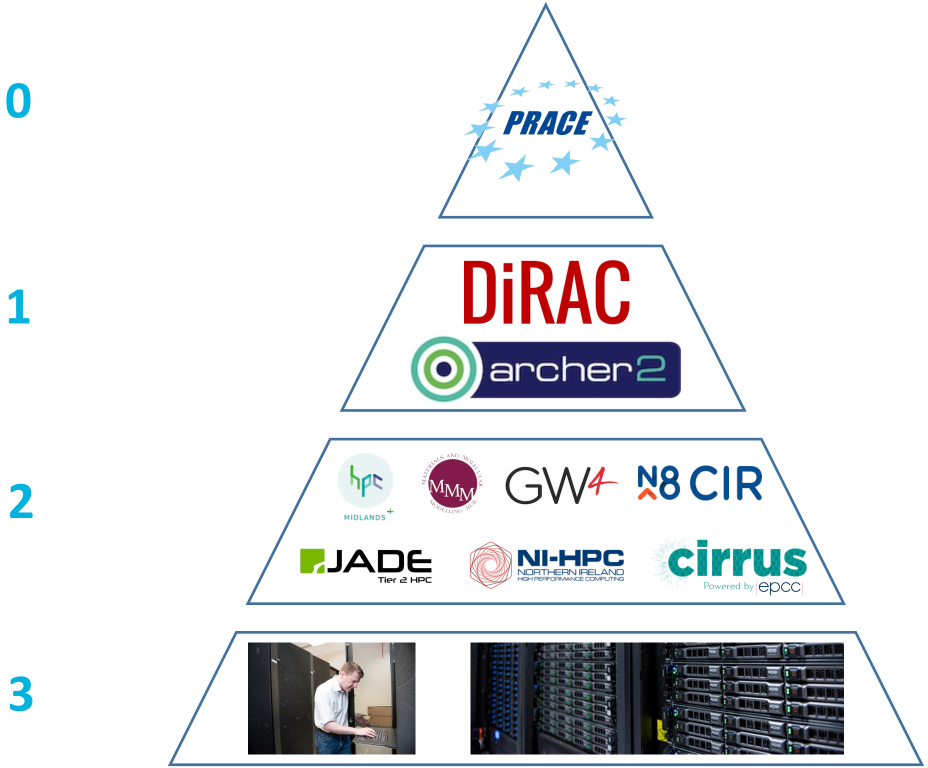
- It will be helpful for you to click links and copy text from these slides during the morning exercises.
- Download from: tinyurl.com/RHPCintro
- Material for this afternoon onwards: mschubert.github.io/R-hpc



https://hbctraining.github.io/Intro-to-shell-flipped/

High Performance Computing cluster

- User connects to login node via terminal client (ssh)
- Works on login node to prepare jobs for submission to a *batch system* (we will use SLURM)
- Submits jobs to a batch queue to be processed non-interactively on a pool of compute nodes
- Compute nodes can be reserved for interactive work
- All node share the same storage via (usually) very high speed networking internal to the cluster


0
PRACE
1
DiRAC
archer2
2
hpc MIDLANDS+
MMM MATERIALS AND MOLECULAR MODELLING HUB
GW4
N8 CIR
JADE Tier 2 HPC
NI-HPC NORTHERN IRELAND HIGH PERFORMANCE COMPUTING
cirrus Powered by epcc
3

UK HPC tiers

- 3: local (e.g. department or university)
- 2: regional / specialist
- 1 : national
- 0 : international (e.g. PRACE but with Brexit impact)

Shared facilities

Pros and cons – no free lunch

Benefits of using shared HPC platforms

- Each compute node is much more powerful than an individual PC or commodity server
- Ability to run across multiple nodes via fast interconnect
- **Run many calculations concurrently**
- (Usually) free at the point of access
- Somebody else runs it, pays for it and supports it
- Can vastly increase the scope and ambition of research, and enable calculations that otherwise take years or decades

Barriers to use shared HPC platforms

- Parallel programming is *hard*, and not all tasks will benefit from running on larger/multiple servers
- Interaction is (mostly) via text only command line interface, which can be daunting
- Most calculations done via batch mode (although interactive sessions for testing/developing encouraged)
- Not your computer. You don’t get to use it when you like, for as long as you like
- No administrative access for users (i.e. no `sudo`..)

Sulis tier 2 HPC system

- Specialises in high throughput or ensemble computing (i.e. trivially parallel workloads)
- 167 standard compute nodes with 512GB RAM and 128 AMD EPYC compute cores
- 30 GPU nodes with 3x Nvidia A100
- 4 high memory nodes with 1.5TB RAM
- 3 very high memory nodes with 4TB RAM
- 2 PB of storage including 200TB of SSD scratch

https://sulis.ac.uk

WARWICK
THE UNIVERSITY OF WARWICK

Morning exercises part 1

- Register for Sulis and login (some have done this already)

  https://sulis-hpc.github.io/gettingstarted/getaccount.html

- Create (on your local machine) a *shell script,* i.e. a text file named `hello.sh` with content:

```
#!/usr/bin/env bash

echo Hello world from `uname –n` working directory `pwd`
```

- Upload this to Sulis and execute via the command line

Demonstration

Useful CLI commands – cheat sheet

THE UNIVERSITY OF WARWICK

- Ninja terminal skills not necessary to use HPC, but need the basics

`cd`	Change working directory to home directory
`pwd`	Print current working directory
`ls`	List files in current working directory
`mkdir Rworkshop`	Create sub-directory of working directory called Rworkshop
`cd Rworkshop`	Change working directory to the sub-directory Rworkshop
`pwd`	Print current working directory
`echo “some text” > file1`	Echo a string and redirect the output into a file file1
`mv file1 file2`	Move (rename) file1 to file2
`cp file2 file1`	Create a copy of file2 called file1
`cat file1`	Show the contents (if text) of file1

Advanced CLI

- The command line can be more powerful than many expect!
- Tools for querying internet resources

```
$ curl wttr.in
```

- Tool for manipulating text, ”piping” the output of one command into another and manipulating text/data (spoilers!)

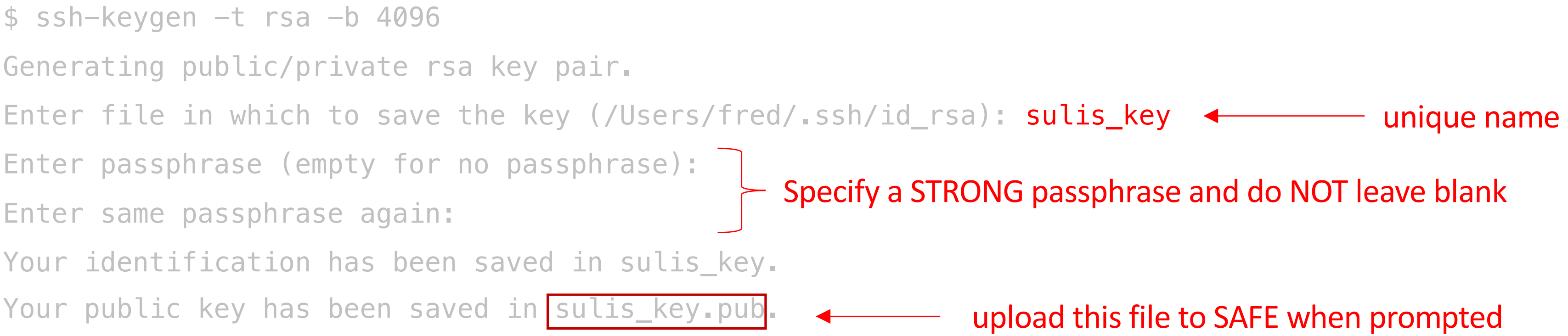
```
    $ wget -q -O- https://www.nytimes.com/games/wordle/main.bfba912f.js | cut -c
31558-50028 | head -1 | awk -F "," -v day=$((($(date "+%s")-$(date -d "2021-06-18"
"+%s"))/(3600*24)))  '{print $((day-1))}'
```

source: https://lordgrenville.github.io/posts/wordle/

Connecting to Sulis (1)

- Access is via ssh key pairs rather than username and password
  - Use strong passphrase to encrypt your private key
  - Use a *different* pair for Sulis (with a different filename) to any other system

- Be sure to upload the ***public*** part of the key to SAFE



```
$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/fred/.ssh/id_rsa): sulis_key
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in sulis_key.
Your public key has been saved in sulis_key.pub.
```

unique name

Specify a STRONG passphrase and do NOT leave blank

upload this file to SAFE when prompted

Connecting to Sulis (2)

- Specify which private key file to use when connecting

your username

```
$ ssh –i sulis_key <username>@login.sulis.ac.uk
Enter passphrase for key 'id_sulis':
```

- May wish to add an entry to `~/.ssh/config` (Mac/Linux/WSL)

```
Host sulis
   HostName login.sulis.ac.uk
   IdentityFile <path_to_key_pair>/id_sulis
   User <username>
```

Customise for your username and location
Use pwd in directory with key pair to check

Connecting to Sulis (3)

- Two factor authentication is mandatory once per day
- Must set this up on first login, have an authenticator app ready

  https://sulis-hpc.github.io/gettingstarted/connecting/firsttime.html
- Other options for generating OTPs are available (ask for help) for if you don’t like or trust smartphones (like me), but make sure you copy/screengrab the first login text

  https://sulis-hpc.github.io/gettingstarted/connecting/TOTP.html

File transfer

- Once you’ve created `hello.sh`, `use rsync` to copy to sulis

```
$ rsync –av hello.sh sulis:
```

assumes ~/.ssh/config entry for sulis

- Note the colon “:” after the Sulis hostname. Anything after the colon specifies *where* (relative to your home directory) to put the file on the remote machine.

- Graphical file transfer clients are available, e.g. FileZilla and CyberDuck. Ask for help if you’d like to try using these with Sulis.

Morning exercises part 2

- Run hello.sh on the login node and via the batch system

  https://sulis-hpc.github.io/gettingstarted/batchq/singlenode.html

- Request an interactive session

  https://sulis-hpc.github.io/gettingstarted/batchq/interactive.html

- Load the R environment module and install a package from CRAN

  https://sulis-hpc.github.io/gettingstarted/software/R.html

- Submit an example parallel R job to the batch system

  https://sulis-hpc.github.io/gettingstarted/batchq/singlenode.html#parallel-package-in-r

Demonstration

Software and Environment modules

- Software environment on the cluster is customisable

`module spider R`	Search for software matching “R”
`module spider R/4.1.2`	Ask for info on how to load specific search result R/4.1.2
`module load <module_name>`	Load environment module named <module_name>
`module purge`	Remove all modules from environment

- To load latest version of R (as of March 2022)

```
module load GCC/11.2.0 OpenMPI/4.1.1 R/4.12
```

Minimal SLURM submission script

- Example job using a single processor core

```
#!/usr/bin/env bash
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=1
#SBATCH --mem-per-cpu=3850
#SBATCH --time=01:00:00
#SBATCH --account=su105
#SBATCH --reservation=rworkshop

echo Hello world from `uname -n` working directory `pwd`
```

specific to *this* workshop

Glossary : https://sulis-hpc.github.io/gettingstarted/batchq/slurmnotes.html

Interactive job on compute node

- Request session on a compute node using 8 processor cores

```
$ salloc --account=su105 --reservation=rworkshop -N 1 -n 8 --mem-per-cpu=3850 --time=1:00:00
```

specific to *this* workshop

- Subsequent commands executed on the allocated compute node

- Exit interactive session (return control to login node) when finished

```
$ exit
```

Basic SLURM cheat sheet

- SLURM generic commands

squeue	Show running/queued jobs
squeue –u $USER	Show only my jobs
sbatch <filename>	Submit job script <filename> to the default queue (compute)
scancel <job_number>	Cancel job with job number <job_number>
sinfo	Show state of cluster resources

- Specific to Sulis

account–balance	Show CPU time remaining in allocated budget