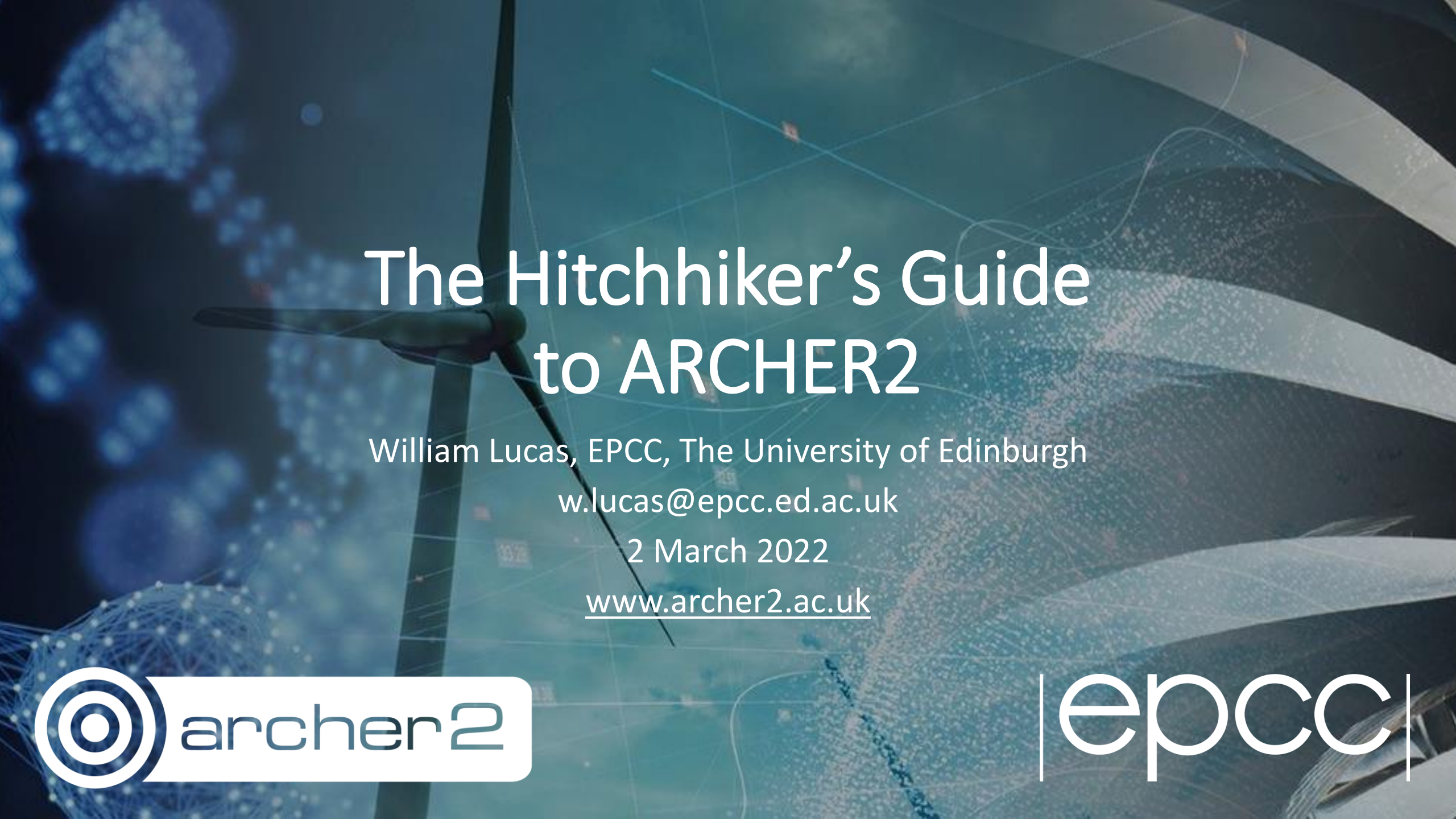


The Hitchhiker's Guide to ARCHER2

William Lucas, EPCC, The University of Edinburgh
w.lucas@epcc.ed.ac.uk

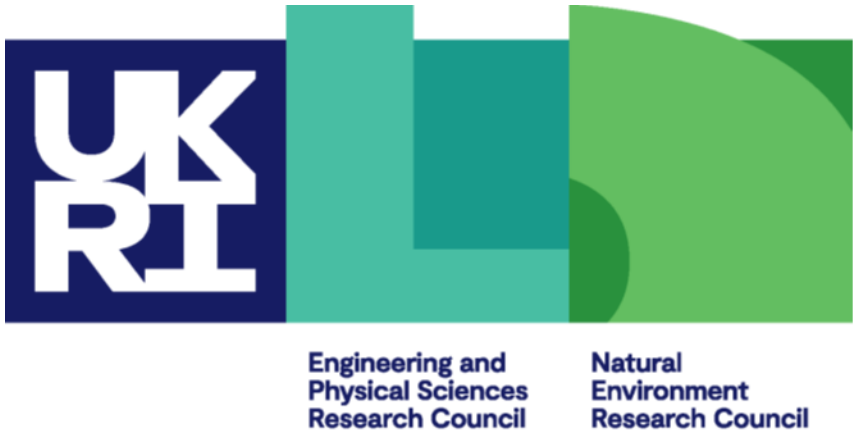
2 March 2022

www.archer2.ac.uk

Reusing this material

Partners

epcc



epcc

THE UNIVERSITY of EDINBURGH

Hewlett Packard Enterprise

What *is* ARCHER2?

ARCHER2 is…

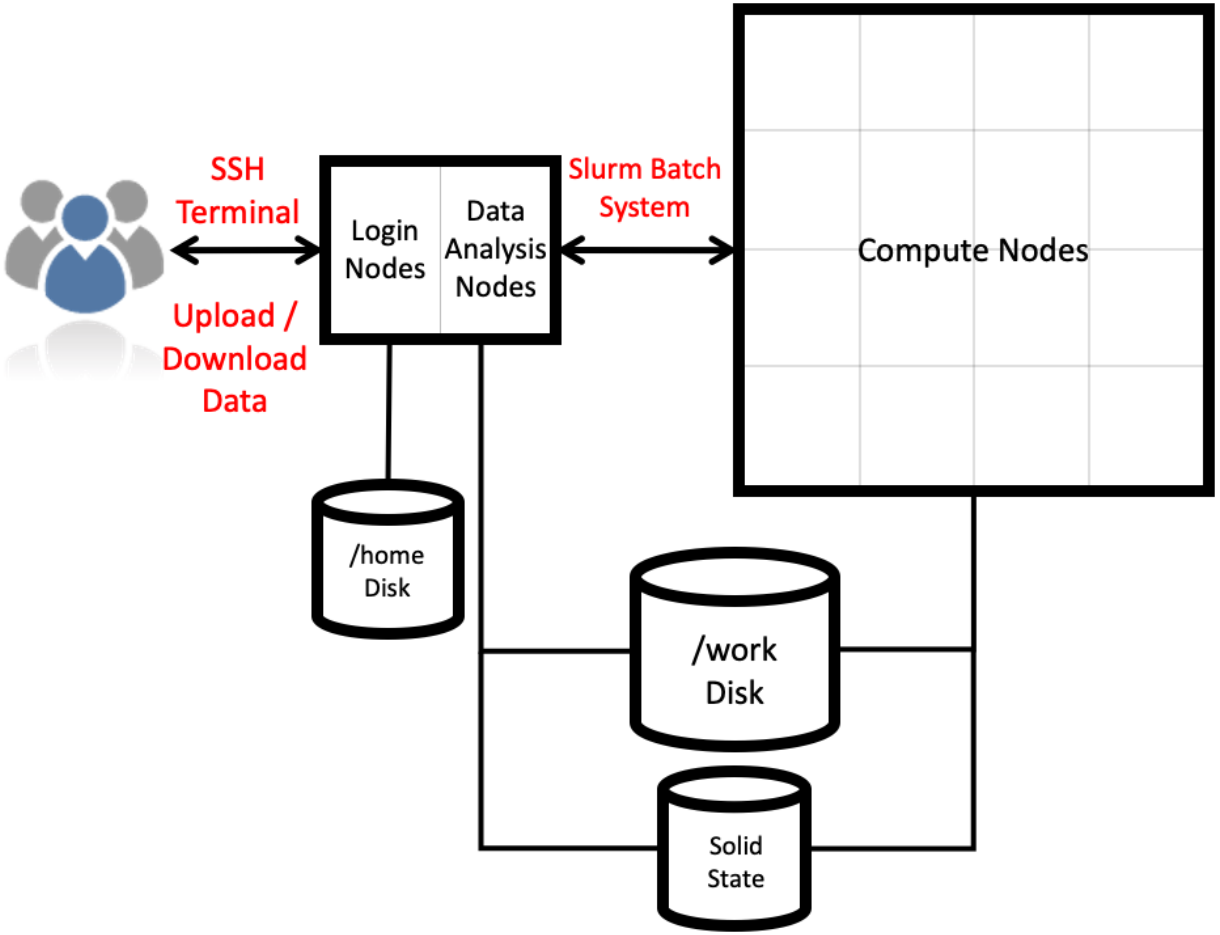
- An HPE Cray EX Supercomputer.
- The UK's national supercomputing service.
- Hosted at EPCC, The University of Edinburgh.
- 5,860 compute nodes (the main resource)
  - Each has dual socket AMD EPYC™ 7742 (Rome) with 64 cores each @ 2.25 GHz
  - 128 cores on each node give 750,080 CPU compute cores.
- It's pretty beefy!
- #22 in November 2021 Top500 (top500.org)
- Or, #5 for CPU-based systems.

What goes into a supercomputer?

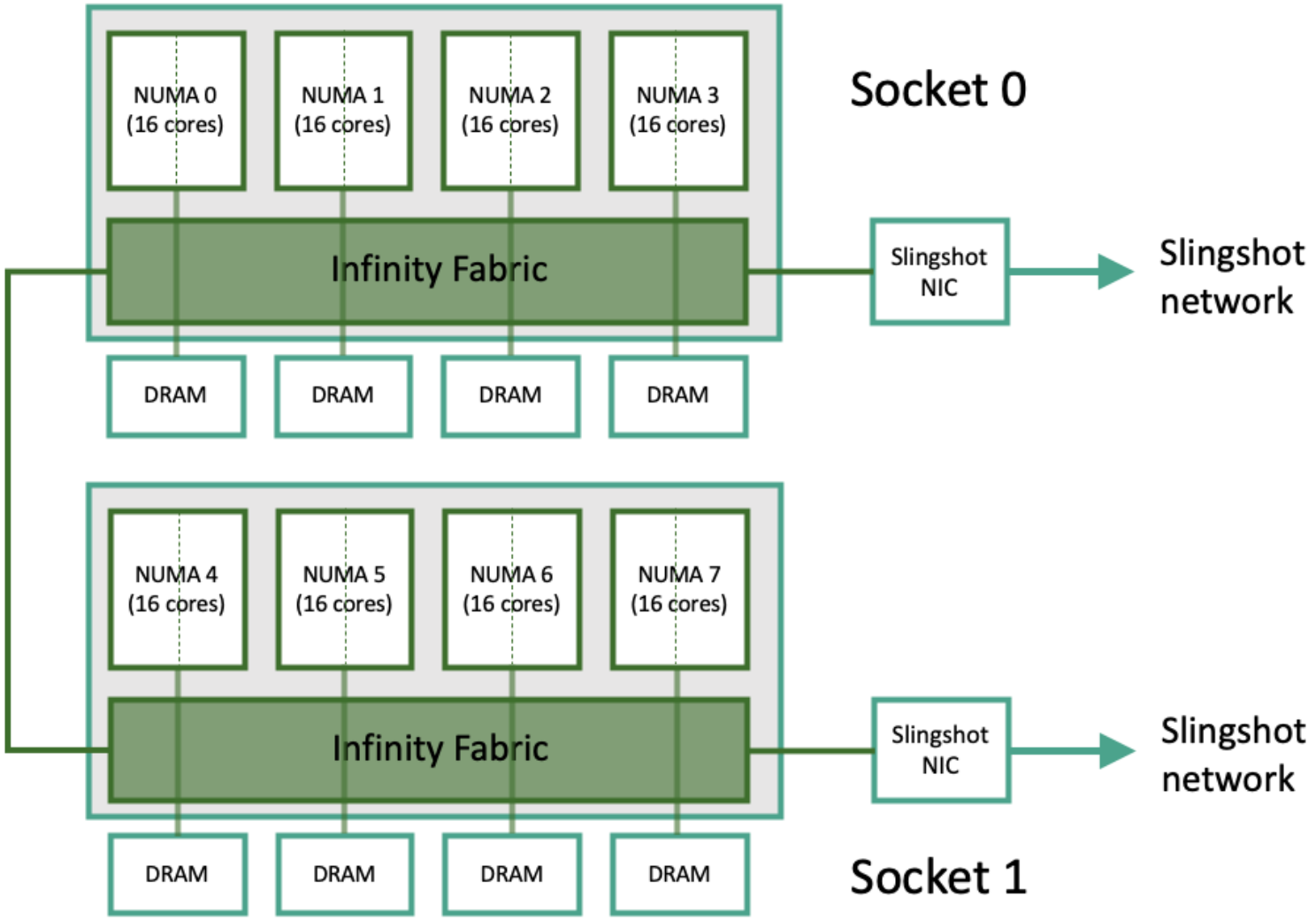
- ARCHER2 is a cluster – it's a supercomputer made up of many individual computers which we call nodes.
- A network links these nodes, turning them into a usable cluster.
- Several types of node, with different purposes:
  - Login nodes x 4
  - Compute nodes x 5,860
    256 GB on a standard node, 512 GB on high memory nodes.
  - Data analysis nodes x 2
- Filesystems - you need somewhere to store files.
  - `/home`: 1 PB network filesystem, backed up
  - `/work`: 14.5 PB high performance Lustre filesystem
  - Solid state: 1.1 PB NVMe storage

The ARCHER2 layout

- Log in with SSH – land on one of the login nodes.
  - Access to both /home and /work.
- Any sort of ‘real’ work (jobs) will be run on the compute nodes or data analysis nodes.
  - Accessed and used via the Slurm batch system.
  - **Compute nodes can’t see the /home filesystem.**
- Solid state – coming soon!



ARCHER2 compute node



Each 16-core NUMA region is made up of 2 8-core compute complex dies.

If using OpenMP, you should use 2, 4, 8 or 16 threads per task.

Each 8-core CCD contains:

- 8 compute cores with 256-bit AVX2 and FMA
- Each compute core has 512 KB L2 cache
- 16 MB shared L3 cache for each of two 4-core compute complexes.

Passport control

archer2

epcc

Getting access

- ARCHER2 login accounts are requested from the EPCC SAFE at https://safe.epcc.ed.ac.uk/: 'Login accounts' > 'Request login account'
- You will end up with one SAFE account and at least one ARCHER2 login account (potentially many).
- To get an ARCHER2 account you will need:
  - The code for the project you are joining (likely provided by your PI).
  - An SSH public key (more next slide).
- Once requested, the project PI and managers will receive an email asking for one of them to approve it.
- You will receive another email once your account has been created.

SSH keys

- The SSH key can be generated by ssh-keygen if you are using a Linux/macOS command line:
  `ssh-keygen -t rsa -C your@email.com`
  - This will generate two files: the one with the .pub extension is your public key, and the other without an extension is your private key.
- On Windows, you can use MobaKeyGen (Tools->MobaKeyGen) if using MobaXterm to generate an RSA-encrypted key.
  - The file with the .ppk extension is your private key—easiest to copy and paste the public key directly from MobaKeyGen into the SAFE.
- You can add multiple SSH keys to your account after creation for access from multiple machines.

Logging in to ARCHER2

- On Linux, macOS and WSL, use `ssh` on the command line.
  - If you are using the standard name `id_rsa` for the SSH private key:
    `ssh username@login.archer2.ac.uk`
  - You may need to specify the name of your SSH private key:
    `ssh username@login.archer2.ac.uk -i /path/to/private/key`
  - Alternatively, could use SSH agent beforehand (once per restart or login):
    `ssh-add /path/to/private/key`
- Otherwise on Windows, MobaXterm is generally easiest and best.
- On logging in you will need to provide your key's *passphrase* (unless the SSH agent already knows it) and your account's *password*.
- Your account's initial password is found on your account's page on the SAFE ('View login account password'). Use this when you first log in.
  - You will then be asked to change your password by entering the current one again, and then the new one twice.

Don’t panic!

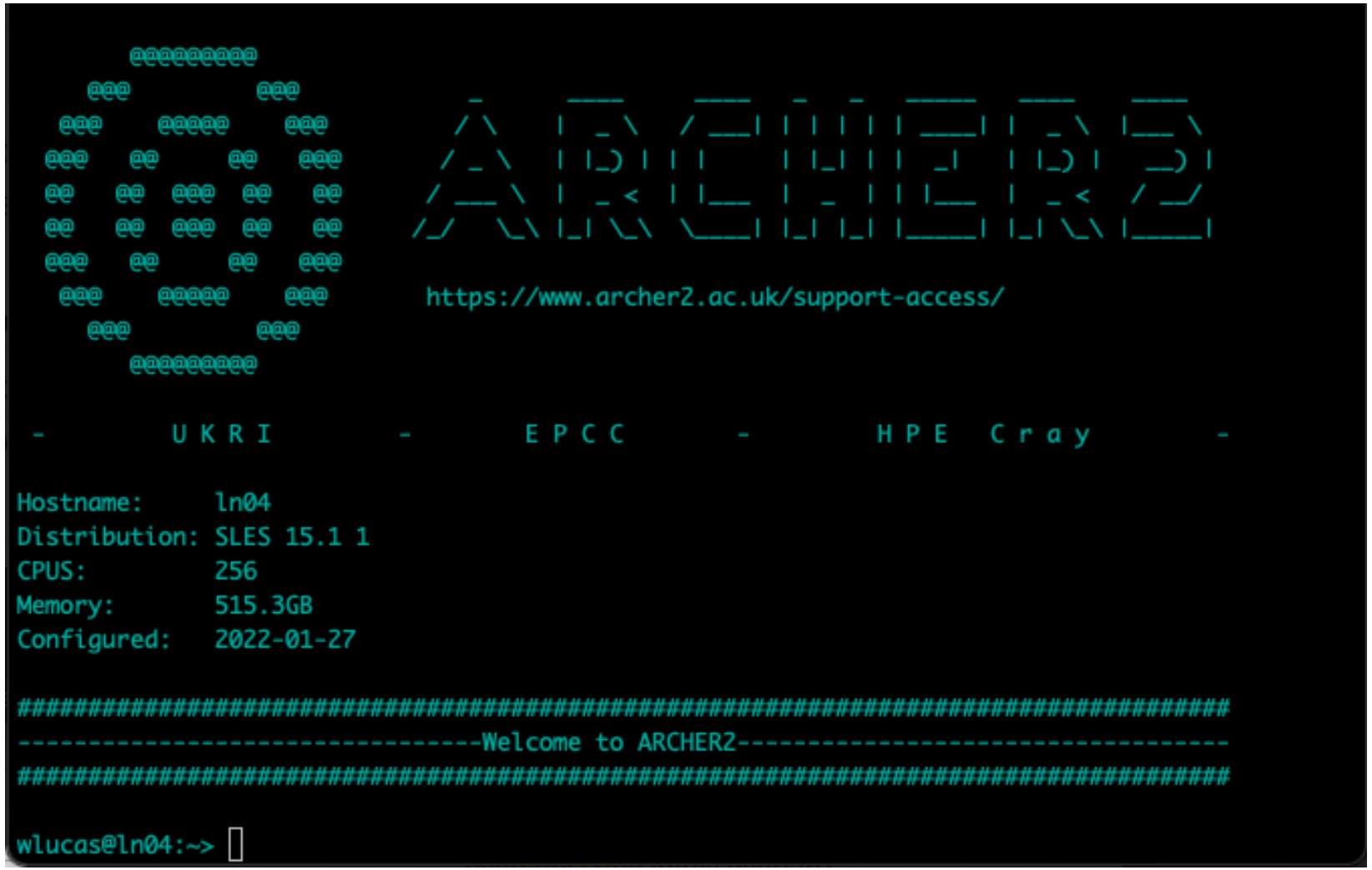
- Logging in is the first hurdle to using the system.
- If you’re having problems:
- See https://docs.archer2.ac.uk/user-guide/connecting for more details, advice and help diagnosing what the problem is.
- Or get in touch with us on the ARCHER2 Service Desk at: support@archer2.ac.uk

What will I find on logging in?

archer2

epcc

On login


https://www.archer2.ac.uk/support-access/
- UKRI - EPCC - HPE Cray -
Hostname: ln04
Distribution: SLES 15.1 1
CPUS: 256
Memory: 515.3GB
Configured: 2022-01-27
Welcome to ARCHER2
wlucas@ln04:~>

On login

- ARCHER2 runs Linux: the HPE Cray Linux Environment
  - Based on SLES 15 (SUSE Linux Enterprise Server).
- You'll find yourself on one of the four login nodes, in your home directory on the /home filesystem:
  - `/home/<projectname>/<projectname>/<username>`
  - So, I arrive in `/home/z19/z19/wlucas`.
- You also have a work directory on the /work filesystem which is where everything you will use in jobs should go:
  - `/work/<projectname>/<projectname>/<username>`
  - My work directory is at `/work/z19/z19/wlucas`.
- Use standard tools to get started (`ls`, `cd`, `mkdir`, `cp`, `mv`, `vim`…)

Modules on ARCHER2

- Installed software is accessed via environment modules (Lmod).
- Loading a module makes the corresponding software available to you.

Command	Action
`module list`	Show which modules are loaded at the moment.
`module avail [string]`	Show all modules that can be loaded *now*. (Optionally limit to modules whose names contain `string`.)
`module load <modulename>`	Load the `<modulename>` module (optionally with version).
`module unload <modulename>`	Unload the `<modulename>` module (optionally with version).
`module swap <module1> <module2>`	Unload `<module1>` and load `<module2>`.
`module spider [string]`	Show all modules. (Optionally search for `string` and display prerequisite modules to be loaded first – helpful if you can't find a module with `avail`.)

What's loaded at login?

```
wlucas@ln04:~> module list

Currently Loaded Modules:
  1) cce/11.0.4                                 8) cray-mpich/8.1.4
  2) craype/2.7.6                               9) cray-libsci/21.04.1.1
  3) craype-x86-rome                           10) PrgEnv-cray/8.0.0
  4) libfabric/1.11.0.4.71                     11) bolt/0.7
  5) craype-network-ofi                        12) epcc-setup-env
  6) perftools-base/21.02.0                    13) load-epcc-module
  7) xpmem/2.2.40-7.0.1.0_2.7__g1d7a24d.shasta
```

Modules on ARCHER2

- Some will stand out, such as `cray-mpich` and `cray-libsci`.
- Some of these modules are lower-level support software enabling the others to function – you should leave these loaded.
- More HPE Cray modules are available:
  - Libraries (e.g. `cray-fftw`, `cray-hdf5`)
  - Debugging tools (e.g. `gdb4hpc`, `valgrind4hpc`, `atp`)
  - Profiling tools (e.g. `perftools`, `perftools-lite`)
- The ARCHER2 CSE service at EPCC provides more modules for:
  - Simulation software (e.g. VASP, CASTEP, GROMACS, OpenFOAM)
  - Utilities (e.g. CDO, ParaView, NCO)
  - More libraries (e.g. Boost, GSL, PETSc)

Modules for package users

- If you only intend to use centrally installed software, such as CP2K:
  `module load cp2k`
  will load the default `cp2k` module, version 8.1.
- If `module avail` shows there are other versions, load those with:
  `module load cp2k/cp2k-8.2`
- If you want to use licensed software such as VASP, you will need to request access to the package via the SAFE.
  - Log in, go to 'Login accounts' > 'username@archer2'.
  - Click the 'Request access to package' button.
  - Choose which software you want to use and provide your licence details.
  - The licence will be checked – informed via email once access is given.

Programming Environments

- With the `PrgEnv-cray` module loaded, as at login, we will be using the HPE Cray compilers as well as libraries for those compilers.
- ***Always*** use compilers via the compiler wrappers, below – they will automatically correctly link the correct HPE Cray libraries.
- Also available are GNU (GCC) and AMD (AOCC) compilers.
- Swap the `PrgEnv-*` module to change compilers.

Language	Compiler wrapper
Fortran	`ftn`
C	`cc`
C++	`CC`

Compiler Suite	PrgEnv
HPE Cray	`PrgEnv-cray`
GNU GCC	`PrgEnv-gnu`
AMD AOCC	`PrgEnv-aocc`

Example ‘Hello, World’ build

- Let’s imagine we have `hello.f90` and `hello-mpi.f90`.
- Following on from initially logging in (i.e. `PrgEnv-cray` is loaded):

```
wlucas@ln04:/work/z19/z19/wlucas/hello> ftn hello.f90 -o hello
wlucas@ln04:/work/z19/z19/wlucas/hello> ./hello
 Hello, World!
wlucas@ln04:/work/z19/z19/wlucas/hello> module swap PrgEnv-cray PrgEnv-gnu

Due to MODULEPATH changes, the following have been reloaded:
  1) cray-mpich/8.1.4

wlucas@ln04:/work/z19/z19/wlucas/hello> ftn hello.f90 -o hello
wlucas@ln04:/work/z19/z19/wlucas/hello> ftn hello-mpi.f90 -o hello-mpi
```

General build tips

- Use the compiler wrappers `ftn`, `cc` and `CC`. May help to set environment variables before `configure`/`make` e.g.

```
export CC=cc
export MPICC=cc
```

- Install software to a location on the `/work` file system.
  - Remember that `/home` isn't visible on the compute nodes!
- Run large and long builds as jobs on the compute nodes.
- You may want to read further about the development environment at https://docs.archer2.ac.uk/user-guide/dev-environment/.

Login node etiquette

- The login nodes are a ‘shared resource’
  - Many other people will be using the same login node as you.
- You can use them to:
  - Set up new jobs
  - Develop code
  - Run shorter builds
  - Prepare data to be moved elsewhere
- You shouldn’t:
  - Run anything long (around 5 minutes +)
  - Run anything CPU-intensive
  - …such as simulations or long, heavy builds.
- Anything intensive or long should go on compute or data analysis nodes.

Moving code and data to/from ARCHER2

- If transferring many files, it's often best to create an archive with `tar` or `zip` beforehand to lower per-file overhead.
- Push to and pull from ARCHER2—i.e. you'll be running the commands to move data from your own machine.
- On Linux, macOS, WSL, use `scp` or `rsync` from your machine:

```
scp local-archive.tar username@login.archer2.ac.uk:/work/z19/z19/wlucas/dest
scp username@login.archer2.ac.uk:/work/z19/z19/wlucas/remote-archive.tar .
```

- MobaXterm has a file browser which allows drag and drop transfers.
- Other tools available on ARCHER2:
  - Globus (GridFTP) for very large transfers
  - git to manage repositories and clone from GitHub etc.

Running jobs on the compute nodes

archer2

epcc

Slurm batch system

- Slurm is used to run jobs on the compute and data analysis nodes.
- Jobs are submitted to the queue. Slurm will schedule jobs and then run them once the resources they need become available.
- Resources:
  - How many nodes?
  - For how long?
- Most users will run jobs by submitting job scripts to Slurm.
  - Include information on the resources required and how they should be used.
  - And what commands to run (as a bash script)
- Can also run jobs interactively—useful for small tests and debugging.

Job charging

- Once a job completes, the resources it used are calculated and subtracted from your budget.
- Compute node resources are charged in CUs
  - 1 CU = 1 node hour
  - 1 node for 2 hours = 2 CUs, 2 nodes for 1 hour = 2 CUs.
- Look at your ARCHER2 account in the SAFE (‘Login accounts’ > ‘username@archer’)
- You will see one or more budgets listed and their resources.
  - E.g. I have access to the z19 budget, so I should tell Slurm to charge to z19.
  - Some project PIs will give individual users their own sub-projects with associated budgets which they will need to use instead.

A simple VASP job script

```
#!/bin/bash

# Set Slurm options for job
#SBATCH --job-name=my_vasp_job
#SBATCH --nodes=8
#SBATCH --ntasks-per-node=128
#SBATCH --cpus-per-task=1
#SBATCH --time=0:10:0
#SBATCH --partition=standard
#SBATCH --qos=standard
#SBATCH --account=z19

# Make VASP 5 available via its module
module load vasp/5

# Prevent threading
export OMP_NUM_THREADS=1

# Use srun for the parallel launch
srun --hint=nomultithread --distribution=block:block vasp_std
```

Partition and QoS

- The partition is the group of nodes which the job should run on.
  - `standard` is the partition containing the 5,860 standard and high memory nodes, but the former are preferred.
  - `himem` is the partition containing the 584 high memory nodes.
  - `serial` is the partition containing the 2 data analysis nodes.
- The Quality of Service (QoS) defines which limits should be applied to the job. Which QoS are available depends on the partition.
  - `standard` allows jobs of up to 1024 nodes to run for up to 24 hours. You can have up to 64 jobs in the queue with up to 16 running, and use up to 1024 nodes at once.
  - Other QoS include `short`, `long`, `himem`, `serial`, `largescale`, …
  - See the QoS section in https://docs.archer2.ac.uk/user-guide/scheduler/

Managing jobs in Slurm

- Write the job script as a simple text file, e.g. `run-job.sh`
- Remember – run jobs on the work file system!
- Submit the script to the queue with
  `sbatch run-job.sh`
  - You will be told the job's ID, a number.
- Check your jobs with
  `squeue -u $USER`
  - Job names, IDs, the nodes in use, how long they've been running.
  - Status: 'PD' pending, 'R' running, 'CG' completing.
  - As well as the reason a job is still pending.
- Cancel a job with
  `scancel <job-id>`

More advanced jobs

- What a job does is entirely up to you.
  - Limited only by the capabilities of bash
- You can:
  - Run pure MPI jobs over one or more nodes.
  - Run hybrid MPI/OpenMP jobs over one or more nodes.
  - Run a job which runs several smaller simulations on fractions of each node e.g. running four simulations of 32 cores each on one node.
  - Run job arrays e.g. running 24 sub-jobs on 1 node each, each sub-job using different arguments or input data.
- Run interactive jobs to work directly on a compute node.
- Lots of information and example scripts in the documentation at https://docs.archer2.ac.uk/user-guide/scheduler/

Next steps

archer2

epcc

Next steps

- The main ARCHER2 website at https://www.archer2.ac.uk provides:
  - Information on getting access, such as via EPSRC calls or the Driving Test
  - A calendar of upcoming training and the material repository
  - The current service status
- The ARCHER2 documentation at https://docs.archer2.ac.uk covers:
  - Logging in and transferring data
  - The application development environment
  - Much more information on how to run jobs
  - Example job scripts for the centrally maintained software
  - Python and R
  - Current known issues and workarounds
- Finally, please contact the Service Desk if you need help at support@archer2.ac.uk.

Upcoming Training

Course	Venue	Dates
ARCHER2 for Package Users	Online	10 March 2022
Software Carpentry	Heriot-Watt, Edinburgh	21-22 March 2022
Message-Passing Programming with MPI	Online	23, 24 and 31 March 2022
Message-Passing Programming with MPI	*Online*	*Self-service – always available*
Shared Memory Programming with OpenMP	*Online*	*Self-service – always available*

Any questions?