HIGH PERFORMANCE COMPUTING

The guide I wish I had

I WANTED TO SEND A JOB TO THE CLUSTER..

how it started

how it ended

CLUSTERS

- At some point in your research, you might/will need your code to run **fast.** You might need a more powerful (i.e. more cores) computer or you might just want to avoid running everything on your laptop ("*can't shut down, code is running*")

- You can use one of the clusters available (see SCRTP.md).

- You can access them from your laptop in a number of ways: **ssh terminal**, free X2Go interface, standard RTP or via the web.

STEP 0 – USING THE TERMINAL

The shell command line

- The command line interface provides powerful computing tools and it is essential when using HPC facilities.
- Bash comes as default user shell in Linux and Mac and can be installed for use in Windows.

File Manipulation

Aim is to gain familiarity with command line inputs as oppose to the graphical interface.

Essential commands:

mkdir *dir*	– create a directory dir	**cp *file1 file2***	– copy file1 to file2 (see mv)
cd *dir*	– change directory to dir	**rm -r *dir***	– delete directory dir
cd ..	– move to super directory	**nano/vim**	– open command-line text editor
ls	– directory listing		

---> more on downloadable **cheat sheet**

STEP 1 – CONNECT TO THE CLUSTER

Create your scrtp account

- Start here:
  https://warwick.ac.uk/research/rtp/sc/desktop/gettingstarted/
- Get username and password

Choose a machine

See SCRTP.md for the complete list.

Connect

Open terminal window:

```
$$ ssh username@machine.scrtp.warwick.ac.uk
```

Where am I?

```
$$ pwd
```

STEP 2 – GET STARTED WITH YOUR DESKTOP

Load modules

You need to load all environment libraries *everytime*.

```
module load GCC/7.3.0-2.30 OpenMPI/3.1.1
module load IPython/7.2.0-Python-3.6.6
```

Everytime? No..

```
module save list-name
```

Next time:

```
module restore list-name
```

Create a virtual environment

(Optional) Like anaconda, it is a way to have environments with different library versions.

```
python3 -m venv env-name
```

activate

```
source ~/env-name/bin/activate
```

install packages

```
pip3 install somepackage
```

*** troubleshooting for abcpy: python -m pip install --upgrade pip

STEP 3 – IMPORT CODE AND RUN FILES

Copy files: from local to remote machine

It's suggested to create a folder in your laptop with the cluster name to put all files and code you need to use.

```
scp filename username@machine.scrpt.ac.uk:~/directory
```

Run code via .sh file

- **.sh** files are used to **execute shell scripts:** these are commands for the terminal (see **sh** and **bash**).
- They all start with: `#!/bin/bash`
- They tell the machine which modules to load, which environment to use and technical details e.g. cores, running time.
- Make sure it is in the same directory as your files and code.

STEP 4 – RUN WITH SLURM PROTOCOL

Slurm protocol

- Slurm is a protocol and a system for cluster management.
  It will do everything for you when you need to run parallel code in the clusters.
- Now you are really ready to run your code

  sbatch run.sh
- You can even disconnect and shut down your laptop, the job is already sent, queued and running!
- Check the status:

  squeue -u username
- Oh, no! I made a mistake

  scancel jobnumber

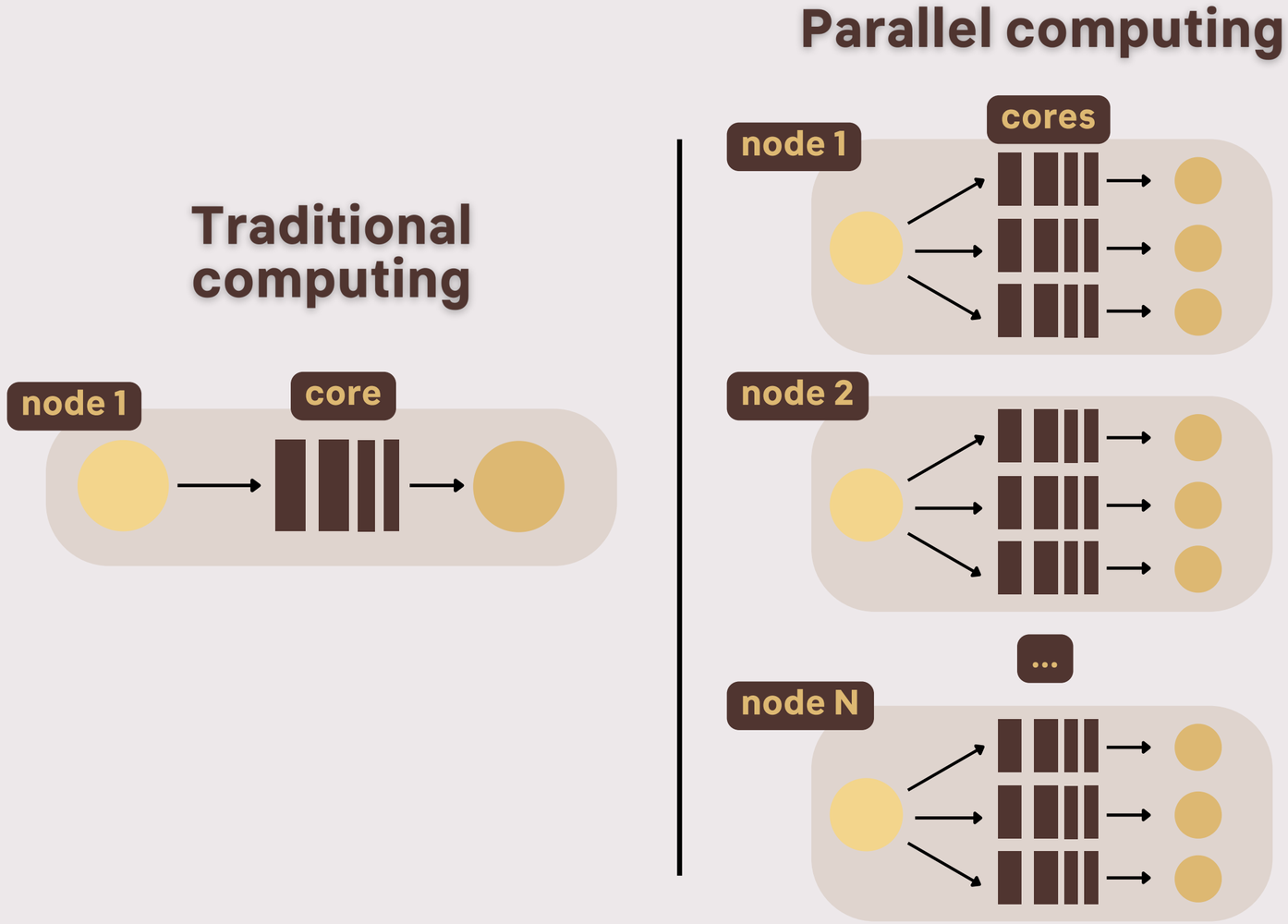
The output

- At the end of the job, you will find the desired outputs (csv, png, txt. etx) and a **jobnumber.out** which acts as a python console.
- Get your files back with reversed **scp** command.

BONUS – HOW TO PARALLELISE YOUR CODE

The MPI backend

- Offers a base class for every parallelization backend.
- It essentially resembles the map/reduce API from Spark.
- Allows to parallelise code over multiple processes and nodes.



Too fast?

Reach out to us.
We have been there and we had someone that helped us too!

more on Computational Techniques