QUICK GUIDE

TO CFSA COMPUTING

PRODUCED BY: ALUN REES

ALUN.REES@WARWICK.AC.UK

IF YOU COME ACROSS ANY PROBLEMS, CONTACT ME AT

CFSACOMPUTING@WARWICK.AC.UK

Table of Contents

SECTION 1

Registering for a SCRTP account

When you first start using the Scientific Computing Research Technology Platform (SCRTP) machines, you will need to register for an account if this hasn't been done already. A SCRTP account will have a username that starts with *ph* and is then followed by 4 other letters. The ph stands for physics, if you are from another department, this prefix will be different. Note that before registering for a SCRTP account you must have a Warwick ITS/email account. These accounts have login usernames starting with "u" followed by 9 numbers. This username is found on your Warwick University student/staff card.

1.1 Getting started

Here are the steps you will need to follow for getting a SCRTP account:

1. Go to `https://warwick.ac.uk/research/rtp/sc/` or google "Warwick scrtp" and chose the top link.

2. On this page you will see a box titled "Linux Desktop" as shown below:

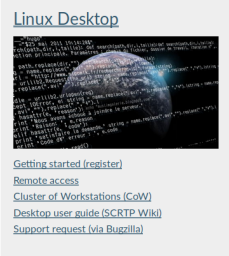

Figure 1: Linux desktop box on SCRTP landing page.

Click the "Getting started (register) link or follow this url: https://warwick.ac.uk/research/rtp/sc/desktop/gettingstarted/

3. From this page look for the "Apply for an account section". Towards the bottom of this section is a link with the text **account creation page**. This is shown below:

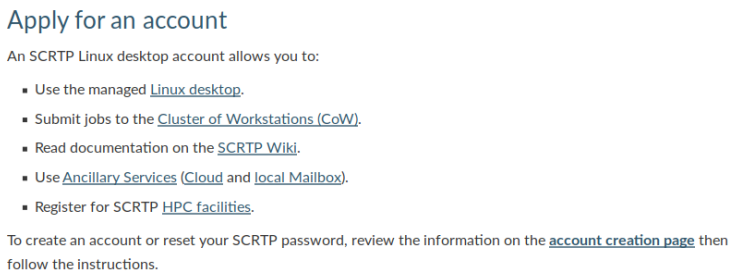

Figure 2: Apply for an account section of page.

4. You should now be on this page: https://warwick.ac.uk/research/rtp/sc/desktop/myaccount/

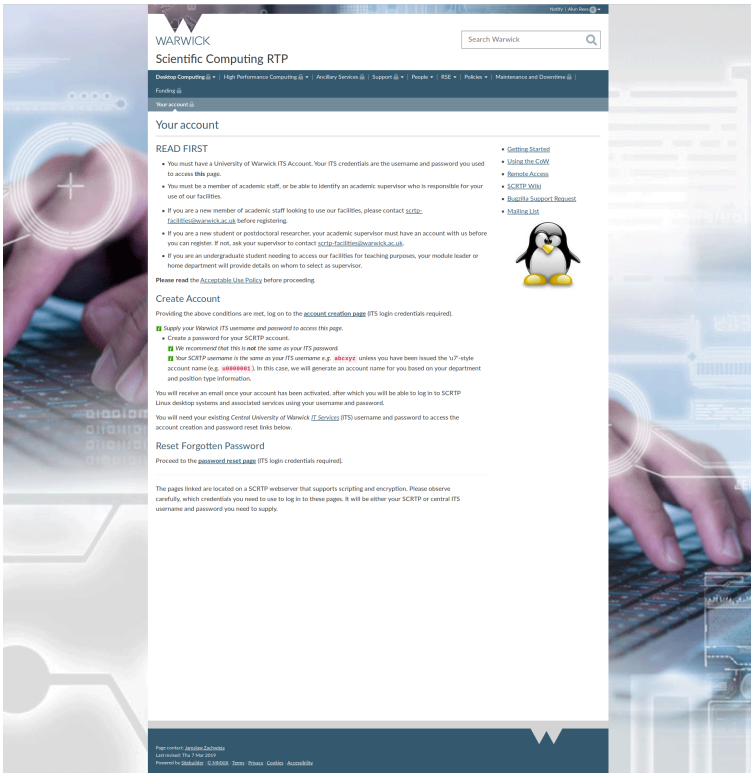

Figure 3: Account creation page.

5. Follow the link titled **account creation page** just under the "Create Account" subtitle.

6. Fill in the form and submit. Make sure you choose CFSA as your department - this is important for setting you up in the correct directory.

7. Wait for SCRTP to setup your account - this usually takes a day.

2

SECTION

Using ssh

Secure Shell (SSH) is a cryptographic network protocol for operating network services securely over an unsecured network.Typical applications include remote command-line, login, and remote command execution, but any network service can be secured with SSH. SSH is an essential tool for using High Performance Computing (HPC) and can also be used to analyse data on a more powerful remote machine.

2.1 Getting started

1. In the command line type:
   ssh-keygen -t rsa -b 4096

2. Choose a directory to store the keys in. Typically, it is stored in the home directory, in a hidden folder named ".ssh". Two keys are generated by running this command, a public key and a private key.

3. Choose a passphrase for the ssh keys, or you can leave it blank if you prefer.

4. Test remotely accessing another computer. A frequently used one for CSC users is nenneke. Using your CSC user name (ph****), use the following command to remotely access nenneke:
   ssh ph**@nenneke.space.warwick.ac.uk**

5. It is likely that a message will appear to ask if you want to add nenneke to the list of authorised hosts, type:
   yes

6. You will also be asked to type the password for nenneke - this will be the same as your CSC desktop password (the one you use with your ph**** account).

7. You should now be in the home directory of nenneke. The files on nenneke should be the same as on your csc desktop. Go to your Documents directory adn check that this is correct:
   cd Documents
   ls

8. You can end a session by typing
   logout

9. Note that in order to see plots and pdf files through ssh you should add the following options to your ssh command
   ssh -Y -A ph**@nenneke.space.ac.uk**

2.2 The config file

In order to avoid typing the full domain every-time you want to log in to a remote machine, you can set up a config file that will allow you to refer to a ssh connection with a hostname. Here is an example of a config file:

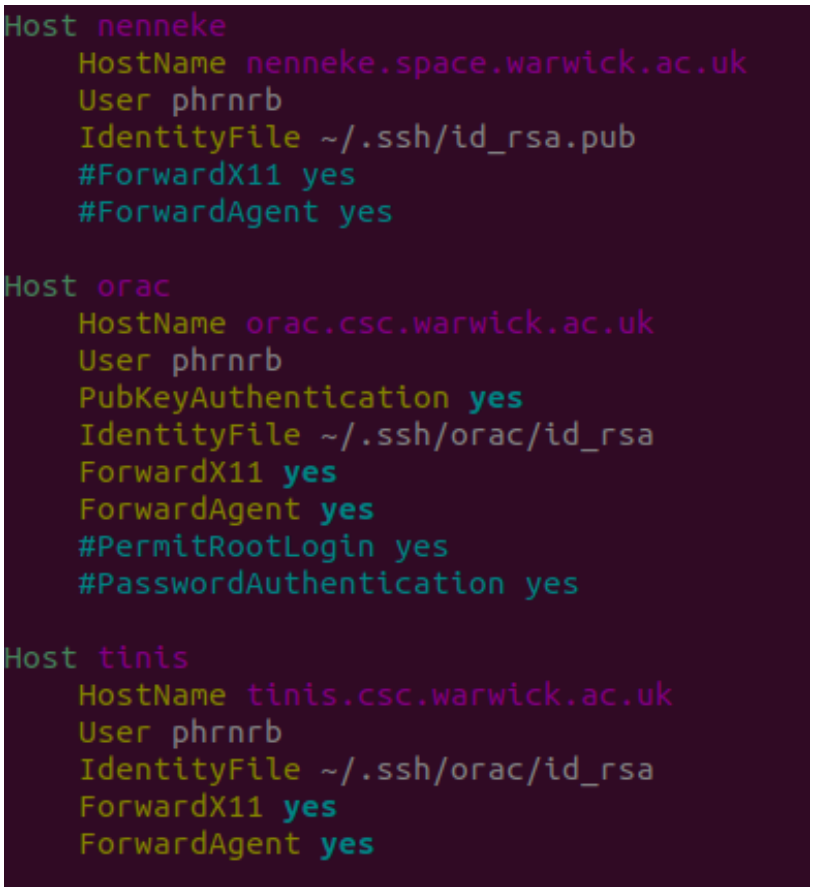

Figure 4: Config file with details for three remote connections to nenneke, orac, and tinis. Note that certain ssh options have been commented out. Using your private id rsa key for the IdentityFile means that you won't be asked for the password on login.

3

SECTION

Using X2Go

X2Go is an open source remote desktop software for Linux that uses a modified NX-3 protocol. X2Go gives remote access to a Linux system's graphical user interface. For CFSA users it provides a way of accessing the CSC computers/network, such as nenneke from a self-managed desktop or laptop.

3.1 Downloading X2Go

X2Go is free to download and is available on all types of operating systems. To download X2Go, click the url to the right: https://wiki.x2go.org/doku.php/download:start

3.2 Using X2Go on a Mac

Machines that use OS X, also need to download XQuartz (The XQuartz project is an open-source effort to develop a version of the X.Org X Window System that runs on OS X) in order to be able to view the GUI generated by X2Go. XQuartz is available for download from the following link: https://www.xquartz.org/

3.2.1 Getting Started

Now that you have downloaded X2Go (and XQuartz if you are using a Mac) then you can get started by following these instructions:

1. Open X2Go. If you are on a mac, you can use the search bar or look for it using Launchpad. The logo looks like this:

Figure 5: The X2Go logo.

2. Once you have opened it, X2Go will ask you to input the details for a new connection. Here are the details for connecting to nenneke, the CSC's cluster of workstations. The dialogue box will look like this:

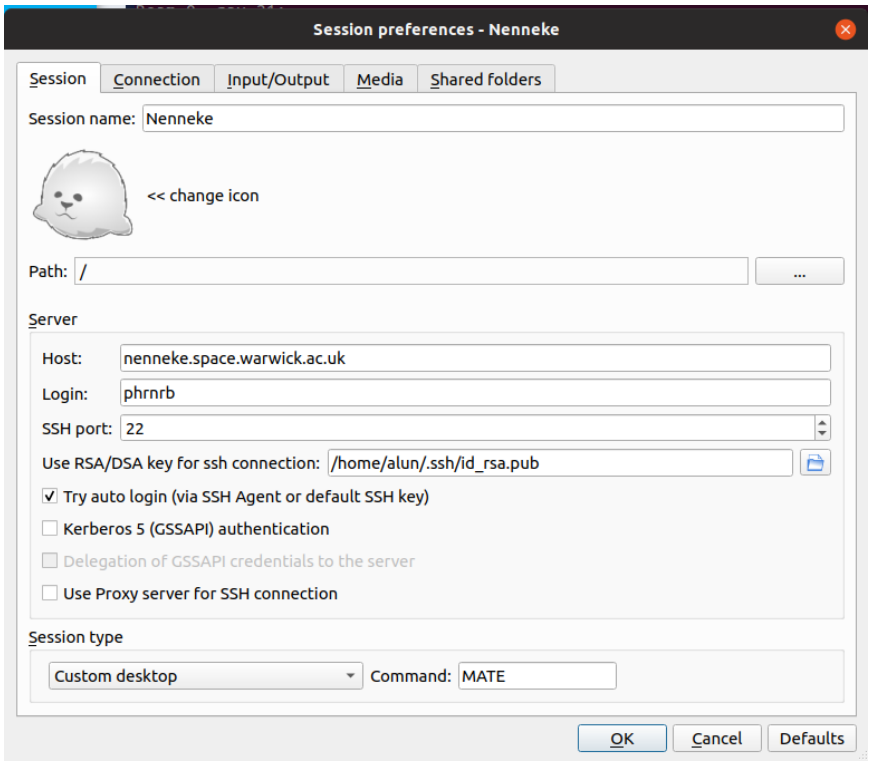

Figure 6: New session dialogue box with the details for nenneke input.

- **Session name**: whatever you want to call this remote connection
- **Host**: nenneke.space.warwick.ac.uk
- **Login**: ph**** (your CSC username)
- **SSH port** : 22 (default)
- **RSA key for ssh connection**: Path to "id_rsa.pub" file (as discussed in Section 1)
- **Session type**: Custom desktop - MATE (these are determined on the bindings set up on the remote computer - for CSC machine this is MATE).

3. Close the dialogue box and try to connect
4. Insert rsa key passphrase and password for nenneke/CSC
5. X2Go should now show your nenneke desktop

4

SECTION

Using modules

Most popular software is readily available on the CSC machines and only need to be loaded onto the computer in order to be up and running.
In order to load a certain module or software package, type the following command into the terminal:

- load module _____

Or to find out more about a module program then use:

- module spider _____

The function will give a brief description of the module/package and a list of what versions are available. An example using MATLAB is shown below:

Figure 7: Screenshot of using the *module spider* function.

Most programs/packages should be available on the CSC machines. If not, then it is possible to request this software by filing a Bugzilla request.

4.1 Customising your .bashrc script

Now that you have loaded the modules that you want, you can ensure that they are ready loaded every-time you use your computer using your *bashrc* script.

This file is usually located in your home directory:

/home/space/ph****/ or ~/

The file comes with some things already included (so that the default environment for the terminal is bash), but specifies where you should put your own preferences and functions. An example of a customised *bashrc* script is shown below: Here are a few

```
# SAVE THIS FILE AS ~/.bashrc
# Read by bash EVERY TIME for interactive non-login shells

### FIRST: source global files which may have been missed
export _bashrc=${BASH_SOURCE}
if [[ -e $HOME/.bash_profile ]] ; then
    source $HOME/.bash_profile
elif [[ -e $HOME/.bash_login ]] ; then
    source $HOME/.bash_login
elif [[ -e $HOME/.profile ]] ; then
    source $HOME/.profile
fi
if [[ -e /etc/bashrc ]] ; then
    source /etc/bashrc
fi


### ... YOUR STUFF HERE ...
## the gets executed EVERY time you start a bash shell
## DO NOT PUT 'module load' COMMANDS HERE
alias store='cd /storage/space2/phrnrb'
export COMPILER=gfortran
### LAST: stop processing if non-interactive
[[ $- != *i* ]] && return
## 'module load' commands and anything else which may produce output on STDERR or STDOUT
date # for example
module load intel/2017.4.196-GCC-6.4.0-2.28  impi/2017.3.196
module load Python/3.6.6
module load matplotlib
```

Figure 8: Example of a customised *bashrc* script

things to notice about this particular *bashrc* script:

1. **alias** store=_____
    - whenever the user types **store** in the command line, then it will execute whatever it is defined as in the *bashrc* file
    - In this case, **store** will change the directory to /storage/space2/phrnrb/
2. **export COMPILER=gfortran**
    - Changes the default compiler to *gfortran*

SECTION 5

Printing

Here are the different options available for printing:

- Direct printing from computer
- CSC computer printing
- Kyocera printing

5.1 Direct Printing

This method allows users to connect their own desktop/laptop and print from there directly.

Instructions for various devices can be found on the following website ITS Printing Instructions.

Many people have reported issues with connecting to the nearest printer (PTGCOLOUR1 - located in the upstairs PS common room), and it is advised that you contact ITS to establish this connection. You can make an ITS service request from the following website, ITS Help Desk, and use the Self-Service Online link.

5.2 CSC computer printing

Note that CSC managed desktops are readily connected to all University printers. If your default computer is CSC managed then printing will not be an issue. If this isn't the case, then you can use X2go and connect to either *nenneke* or *godzilla*. The printers will be listed as follows:

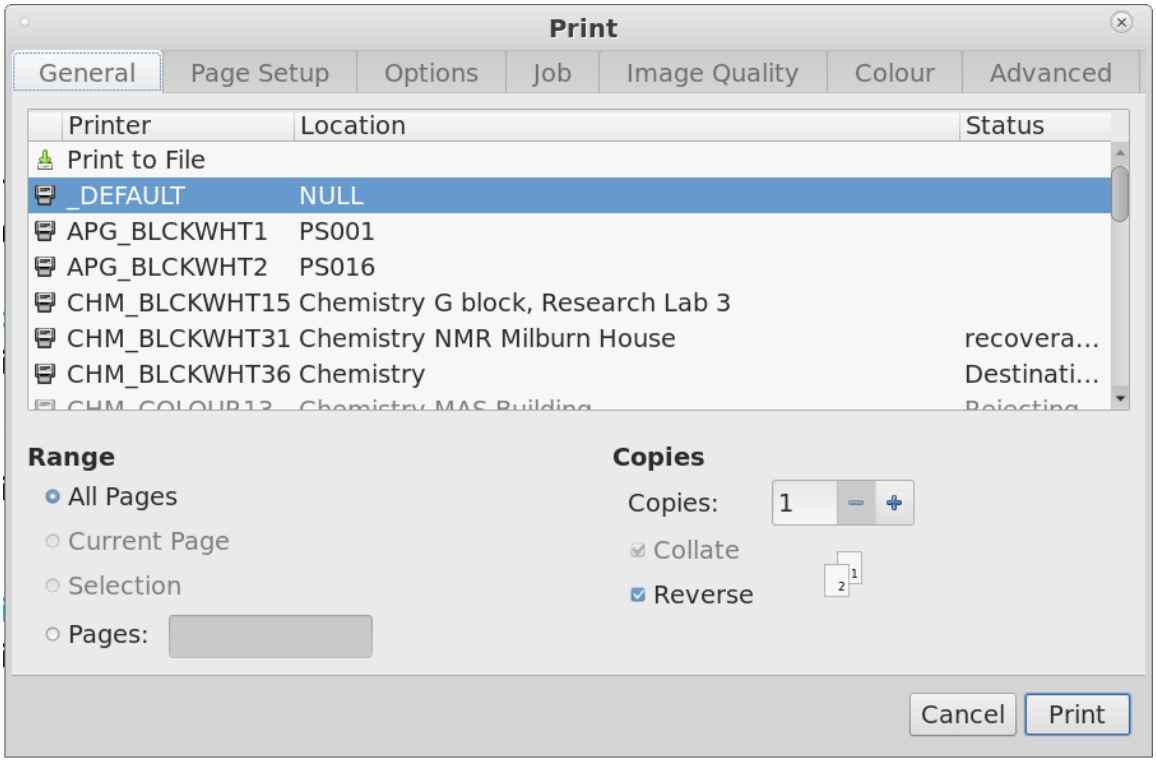

Figure 9: Printing pop-up box on CSC computers

It's also possible to print from a CSC machine using *ssh*. Having logged into the CSC computer using ssh, go to the directory of the document/photo you would like to print off, and then type the following command:

lp -d PTG_COLOUR1 -o sides=two-sided-long-edge *filename*

For more on command line printing, visit lp command

5.3 Kyocera printing

Another option for printing around the University is using the Kyocera printers. These will print documents in black and white, and double-sided. In order to send a document for printing, then you need to send the document via email to:

mobile.printing@live.warwick.ac.uk

You can sign in to any Kyocera printer by using you Warwick University card, and print your document there.

6

SECTION

Using Python

Please note, that this section requires some knowledge of using python already.

Here are the steps needed in order to get started using Python on your CSC computer:

- Load a version of Python
- e.g Use: *module load Python/3.6.6*
- Type python in the command line to start a terminal session
- You can edit your python script in the terminal using vim *example.py*
- It is possible to load your favourite GUI for python too
  - module spider _____
  - For Spyder use:
  - module load spyder/3.3.1-Python-3.6.6
  - Type *spyder* in terminal to start a session
- It is possible to have ready loaded modules and constants for your python terminal sessions:
  - Create a .pythonrc script in your home directory
  - Add the follwoing line to your .bashrc script:
  - *export PYTHONSTARTUP=~/.pythonrc*

– *source .bashrc*

The format of the .pythonrc script is the same as any .py script. An example of a .pythonrc script is shown below:

```
import os
import sys
sys.path.append('/home/space/phrnrb/Documents/epoch2/epoch/SDF/utilities/sdf_helper')
sys.path.append('/home/space/phrnrb/Documents/baldur')
import numpy as np
import matplotlib.pyplot as plt
import sdf
import sdf_helper as sh
import menu as mu

#Define constants
G  = 6.67430e-11
h  = 6.62607015e-34
c  = 299792458
u0 = 1.256637e-6
E0 = 8.854187e-12
mp = 1.672621e-27
me = 9.109383e-31

#Message for me
print('Additional modules and constants loaded')
```

Figure 10: An example of a .pythonrc script

7

SECTION

Using IDL

Simialrly to the previous section, this requires some preliminary knowledge of IDL and how it works.
Here are the steps needed for using IDL:

- Load a version of IDL.
- Type *module load IDL/8.7.1* in the terminal
- Type *idl* to start a terminal session (no GUI)
- It's possible to start a GUI for IDL by adding a few lines to your .bashrc script (see below)

Lines to add to .basrhc script for IDL GUI:

- IDL startup:
- *export IDL_STARTUP=/home/shared/space/ssw/idlstart.pro*
- Location of script to load ssw:
- *alias mirror='/home/shared/space/ssw/gen/mirror/mirror.pl'*
- *alias sswde='csh /home/shared/space/lf/sswde.csh'*
- *alias ssw='csh /home/shared/space/lf/ssw.csh'*

- *export SSW_INSTR="aia"* - this line is optional, it adds commands for specific solar instrument SDO/AIA. Should be used if a user intends to use AIA for research. In principle, ssw will work without it.
- Exit the .bashrc script and type *source .bashrc* in the terminal to load changes

If you type "sswde" or "ssw" in the terminal then the CFSA adapted, IDL Guided User Interface (GUI) will open.

To check that the commands are working properly, follow these instructions:

- Type *x = findgen(100)* in the GUI (this creates array of 100 elements from 0 to 99)
- *print, minmax(x)* (minmax is a function unique to ssw and sswde)
- If ssw works properly, it should return 0.000.. and 99.000

SECTION 8

Using HPC Systems

The University offers two HPC systems for running large computing jobs. These computers are called *Orac* and Tinis.
In order to use these systems, you must register:

1. Fill in the form located at `https://warwick.ac.uk/research/rtp/sc/hpc/register/`
2. If you want access to both *Orac* and *Tinis* then you'll need to complete the form twice (one time for each system)
3. Once the form is completed, it will take some time for you account to be registered (about 1-2 days).
4. Your id_rsa.pub file will be added to the computer allowing you to access via *ssh*
5. You can access the computer(s) by typing:
6. ssh ph****@orac.csc.warwick.ac.uk
7. ssh ph****@tinis.csc.warwick.ac.uk

You'll need to upload the code you intend on running and create a *job.sh* script. Details on how to do this are available on HPC Wiki - note that you'll need to your CSC credentials to sign in.

Here is an example of a job.sh script adapted for running an epoch simulation:

```
#!/bin/bash

# Resource requests
#SBATCH --time=48:00:00
#SBATCH --job-name=CB_Foil
#SBATCH --nodes=12
#SBATCH --ntasks-per-node=28

# Output and Errors
#SBATCH --output=/home/space/phrnrb/CeriBrenner-Foil2/std.out
#SBATCH --error=/home/space/phrnrb/CeriBrenner-Foil2/std.err

# load intel libraries
module purge
module load intel
module load impi
module load imkl

DIR=/home/space/phrnrb/CeriBrenner-Foil2/
echo $DIR | srun $HOME/epoch2/epoch2d/bin/epoch2d
```

Figure 11: A job.sh script used for running an EPOCH simulation.

9

SECTION

Going Self-managed

If you would rather not have a computer that is maintained by the CSC, it is possible to go *self-managed*. By choosing this, you can remove the default MATE Linux operating system (OS), and install your preferred OS. Note that the most common self-managed OS in CFSA is Ubuntu as it is easily customised and very user-friendly for people who are unfamiliar with linux. Note, that other OS are available on request. Windows is available through ITS.

I would recommend asking the CFSA computing team (Ben McMillan and computing assistant) to help with this, since there are a few problems that can arise. And it will be necessary to set up a Bugzilla request to change the settings of the ethernet port from CSC to self-managed.

- Note that if you have a CSC computer, then your ethernet port with be configured for CSC usage. This is how the computer installs the OS, and performs any updates that are performed system wide.
- Please think carefully about the decision to go self-managed, since it will be very time-consuming to swap your system back to the CSC OS.
- If you would prefer a Mac, the department has a few spare that could be adopted as your default computer.

SECTION 10

Additional Training/Resources

If you would like to learn more about using UNIX systems, then there are courses available at the University via the RSE courses.
They provide courses on:

- Python
- FORTRAN
- HPC systems
- Software development
- Java
- MPI
- C and C++

ITS also offer training in some more fundamental computing topics, ITS Learning.