ExCALIBUR/SiMLInt workshop June 2023
Using Cirrus

EPCC

June 6, 2023

Introduction

We will be using the Cirrus cluster for the workshop. Cirrus is an EPSRC Tier-2 system hosted by EPCC. It has 280 CPU-based compute nodes, each with two 18-core Intel Xeon E5-2695 processors and 256 GB of memory. A further 38 nodes are equipped with GPUs.

A new project has been created on Cirrus for this workshop. You should create a new account on Cirrus, even if you already have a Cirrus account. This will allow you to use the project's resources and node reservations and more easily share files with the other attendees. If you've ever created a Cirrus or ARCHER/ARCHER2 account, this process will be familiar to you. The steps to doing so are laid out in order below, starting with the creation of an SSH key.

If you run into any problems setting up your Cirrus account, please feel free to email Elena Breitmoser at e.breitmoser@epcc.ed.ac.uk. Once you have your account, you may wish to familiarise yourself with Cirrus. Some information is provided at the end of this document, but full documentation is available online at https://cirrus.readthedocs.io/.

1 Create an SSH keypair

Authenticating yourself during the Cirrus login process requires both a password and an SSH key. We'll create your SSH key first so that you'll have it ready when you then create your account on Cirrus. How you do this will depend on your operating system and SSH client.

In all cases you will be asked to set a passphrase for your key. Please do so as this increases the level of security the key provides. It can also be helpful to note here that the word 'passphrase' will always refer to the security word required to authenticate the use of a key, while 'password' will always refer to the security word associated with your Cirrus account itself. Prompts for either will use the same terms.

1.1 Linux/macOS/MobaXterm Terminal/Windows Subsystem for Linux

Create an SSH keypair with this method if you intend to log in to Cirrus using `ssh` from the command line.

1. Run the command

   ```
   $ ssh-keygen -t rsa -C "your@email.com"
   ```

   where you should use your email address as the key comment following the `-C` option, or else provide some other phrase to help identify the key.

2. You will be prompted to choose the full path to the private key. By default this will be something like `/home/user/.ssh/id_rsa` where the path is to the hidden `.ssh` directory in your home directory, and `id_rsa` is the default name of the private key. You may want to use this, in which case simply press enter, but if you don't, provide your choice. You may *e.g.* want to name the key `id_rsa_cirrus` instead. I'd recommend keeping the key in this `.ssh` directory as this is the default and makes it harder to lose.

3. You will then be prompted twice to enter the passphrase you want to use to unlock the key.

4. Finally, the private and public key files will be created. The private key will have the name and be at the location chosen previously, so `/home/user/.ssh/id_rsa_cirrus`, for example. The public key will be placed in the same directory and have the same name but with an extra `.pub` extension.

1.2 Windows with MobaXterm GUI

Though there are *many* SSH clients to choose from on Windows, we strongly recommend that you use MobaXterm, as this is the one with which we have the most experience setting up for Cirrus. If you want to use another client you may do so, but if you run into any problems it could potentially take quite some time to resolve.

You may want to run `ssh` from within the MobaXterm terminal as though it were Linux, in which case you can follow the instructions previously given in Section 1.1. If you follow the method given here in this section, you will be working natively within MobaXterm and also have access to a GUI file browser once connected.

1. Start the MobaXterm client, then click on the 'Tools' item in the menu, followed by 'MobaKeyGen'. The MobaKeyGen window will open. You should keep this open until you have added your public key to your account in Section 2 as it is easiest to copy the key directly from this window.

2. In the new window, make sure that the options at the bottom are set to create an RSA-encrypted key with 4096 bytes.

3. Click on the 'Generate' button. You'll see that you're prompted to move the cursor around within the window's blank space 'to generate randomness'. Do so until the bar fills.

4. Once the key has been generated, you will need to enter a comment (such as your email address) to help identify it as well as your passphrase.

5. Now save the public and private keys. You should place them side-by-side in the same location and make sure you can remember where this is. The private key should have a name ending in the `.ppk` extension (standing for PuTTY Private Key), while the public key should go without an extension.

6. You'll see that there is a large text box in the previously blank area which now contains the public key in a format which can be read by OpenSSH. As said above, keep this window open until you've had the chance to apply this key to your new Cirrus account. In case you do close MobaKeyGen first, you should be able to start it again and then load the key files once more.

2 Request an account on Cirrus via the SAFE

The SAFE, https://safe.epcc.ed.ac.uk/, is a website maintained by EPCC that is used to administer many clusters and services provided by EPCC and other sites around the UK. If you've used it previously, please go ahead and log in to your existing account. You can associate as many machine login accounts as you like with your SAFE account. If you haven't ever used it previously, please create a SAFE account now and then log in.

Once you are logged in to the SAFE, you can request a new Cirrus login account:

1. From the home page on the SAFE, mouse over 'Login accounts' on the top menu and then click on 'Request login account'.

2. On the new page you will be asked to enter the project code. Enter 'tc045', which you will see again and again as a reference to this training event, and then click on 'Next'.

3. The next page will ask which machine to create the account on. 'Cirrus' is the only option and will already be selected. Click 'Next' to continue.

4. On the next page, you will need to enter the username you would like to use on the cluster. You should also take this opportunity to add your public SSH key.

(a) If you created your key on the terminal using `ssh-keygen`, you can click on 'Browse' and navigate to your `.pub` key to upload it. Alternatively, copy the file's contents into the text box on the web page after opening in a text editor or viewing in the terminal with `cat` or similar. You may find the latter easier, as `.ssh` is typically hidden from GUI browsers.

(b) If you created your key with MobaKeyGen, you should go back to that window and copy the public key from the large text box there, and then paste it into the text box in the SAFE.

Then click 'Request' to ask for your account to be created.

5. The project's managers will be emailed to let them know that you have asked for your account to be created. It is up to them to approve or deny it – as long as you are actually one of the training attendees (and as you're reading this, you presumably are!), it will approved. As this part of the process isn't automated, please don't worry if your account isn't approved and created immediately.

6. Once the request has been approved, you will be emailed to let you know and the account will be created on Cirrus.

3 Log in to Cirrus for the first time

It's worth making sure you can successfully log in to Cirrus ahead of arriving at the workshop on 5th June. If you arrive and find you have problems logging in, you may lose valuable time trying to fix them instead of being able to work on the exercises.

3.1 Linux/macOS/MobaXterm Terminal/Windows Subsystem for Linux

If you intend to log in using `ssh` from the terminal and followed the instructions in Section 1.1, follow these steps to log in.

1. The Cirrus login URL is `login.cirrus.ac.uk`. The basic form of the login command is:

   ```
   $ ssh username@login.cirrus.ac.uk
   ```

2. If you used any non-default name or location for your private SSH key, you may find that the connection is immediately closed. In this case, you have a few options to ensure that `ssh` knows which key to use. These examples all assume that your private key is `/home/user/.ssh/id_rsa_cirrus`, which you should change to your key's name and location.

   - The brute-force method is to instruct `ssh` directly to use your new key with the following command:

     ```
     $ ssh username@login.cirrus.ac.uk -i /home/user/.ssh/id_rsa_cirrus
     ```

     You will need to include the `-i /home/user/.ssh/id_rsa_cirrus` option with every Cirrus login session.

   - You can use the SSH agent. This also means you only have to enter the key passphrase once per your machine restart, rather than with every new SSH session (though some Linux distributions will attempt to use the agent no matter what). Run the command

     ```
     $ ssh-add /home/user/.ssh/id_rsa_cirrus
     ```

     and then, when prompted, enter your key's passphrase. The SSH agent continues to run in the background and presents the unlocked key whenever requested. If all is working correctly, you should then be able to log in using the simple

     ```
     $ ssh username@login.cirrus.ac.uk
     ```

     command. The SSH agent will also authenticate you when using other tools such as `scp` and `rsync`. You will only need to re-run `ssh-add` and enter your passphrase again whenever you have restarted your machine.

   - You can create an SSH `Host` config to use. This also makes your `ssh` login command shorter. You will need to open and edit the `/home/user/.ssh/config` file, changing the `/home/user` path to the correct one for your home directory. You can create this file if it doesn't yet exist. Within it, you will need to enter a few lines:

```
Host cirrus-pax
    HostName login.cirrus.ac.uk
    User username
    IdentityFile /home/user/.ssh/id_rsa_cirrus
    IdentitiesOnly yes
```

You will need to replace `username` with your Cirrus username. Once this is done, you can log in with `ssh` referencing only the name of the `Host` from the config:

```
$ ssh cirrus-pax
```

You can also refer to `cirrus-pax` when using tools like `scp` and `rsync`.

- You can combine the previous two methods (this is the way I do it) with an SSH config and the SSH agent. Write an entry in your `/home/user/.ssh/config` file with the following contents:

```
Host cirrus-pax
    HostName login.cirrus.ac.uk
    User username
    ForwardAgent yes
```

Then, every time you restart your machine, add your private key to the SSH agent:

```
$ ssh-add /home/user/.ssh/id_rsa_cirrus
```

You now have the best of both worlds: you only need to enter the key's passphrase once, when you restart your machine, and you can log in to Cirrus using the short command

```
$ ssh cirrus-pax
```

3. When you are prompted for your account password (*not* your key's passphase) you are connected. You will now need to provide the initial login password and then change it to a new one of your choice. Go to Section 3.3 below to see how to proceed.

3.2 Windows with MobaXterm GUI

If you intend to log in using the MobaXterm GUI and followed the instructions in Section 1.2, follow these steps to log in.

We'll set up a new MobaXterm session and configure it to connect to Cirrus using the SSH key you added via the SAFE.

1. Click on the 'Session' button at the top left of the MobaXterm window (or equivalently click on 'Sessions' in the top menu, followed by 'New session').
2. In the new window, click the 'SSH' button to create a new session of this type.
3. In the 'Basic SSH settings' panel enter `login.cirrus.ac.uk` in the 'Remote host' text box, tick the 'Specify username' box and then enter your Cirrus username next to it. Leave the port set to 22.
4. Click on the 'Advanced SSH settings' tab below to open a new pane in the window.
5. Tick the 'Use private key' box. You should see a small file icon in the text box to the right. If you click on it, you can navigate to and select your private SSH key – the file ending with the `.ppk` extension.
6. Click on 'OK' to complete the setup and connect to Cirrus.
7. A terminal will open and you will be prompted to provide your key's passphrase.
8. Next, you will be prompted for your account password. At this point you are connected, though not yet logged in. You will now need to provide the initial password and then change it to a new one of your choice. Go to Section 3.3 below to see how to proceed.

Depending on the setup within MobaXterm and your installation's version, there is a chance you may be prompted twice to provide your password and passphrase. This is because MobaXterm provides a file browser GUI which may open via a connection separate from your terminal. If it asks you to do so, provide your key passphrase and account password as before to authenticate the second session.

You will only need to follow the above process once. Whenever you want to connect to Cirrus again, you can select the 'Sessions' tab on the left hand pane, if it isn't already open, and you should see 'login.cirrus.ac.uk (username)' listed. Double click on this item and you will be connected.

3.3 Initial password

Once you are connected and authenticated via your SSH key, you will be asked to enter your password. For your first login you'll use one that was set automatically for you and which is stored in the SAFE. If this is successful, you'll then be asked immediately to set a new password of your choice.

To retrieve your initial login password, head back to the SAFE. Mouse over 'Login accounts' on the menu. You should now see your new Cirrus account listed as 'username@cirrus'. Click on it to see an overview of your account. Near the bottom of the page you should see several buttons. Click on the one labelled 'View Login Account Password'. You can copy and paste this directly into the password prompt in your SSH session to complete the login.

An aside here: if you find that your initial password is being rejected, it is worth checking that you are actually pasting it into the prompt. Some terminals and clients do not use the common Ctrl+V shortcut for paste, and the terminal-based password prompt doesn't display any characters when they are entered. This makes it possible to spend a long time trying to log in while unknowingly failing to provide any password at all. Some Linux terminals use Ctrl+Shift+V as the paste shortcut. MobaXterm uses Shift+Insert. It may be best to right-click inside the terminal and select paste, or else to enter the password manually.

Once you are logged in, you will see the Cirrus welcome message and then be informed that you need to change your password. You will firstly need to enter once more the same initial password from the SAFE, and then to enter twice your new password. This should of course be of reasonable complexity.

Please note that the password shown in the SAFE *does not* change from the initial one after you set your own. Only you will know your password. If you lose your password after this point, you will need to request a reset. There's another button to let you do this on your Cirrus account page in the SAFE, 'Request password reset', next to the same button to view the initial login password.

4 Using the system

Now you are logged in to Cirrus, you are free to take a look around.

4.1 File systems

There are two file systems we will concern ourselves with, `/home` and `/work`. On logging in you will find yourself in your home directory at

```
/home/tc045/tc045/username
```

The `/home` file system is not particularly large but can be used to store some important files. Importantly, it is not mounted on the compute nodes. Jobs will instead be run from within the `/work` file system. You will have your own work directory at

```
/work/tc045/tc045/username
```

Disc space is shared between all members of the project (*i.e.* all attendees and demonstrators at this event), with 500 GiB available to us on `/work`. Please keep this for the exercises only, and clean up if you accidentally produce any huge files.

Your home and work directories are kept private to you alone. If you want to share files with anyone else, or if the demonstrators want to share files with you, the `shared` directories can be used. These exist on both file systems in two hierarchies. To share with other users in this project (who are also members of the `tc045` Unix group), use

```
/work/tc045/tc045/shared
```

and to share with anyone on any other project you can use

```
/work/tc045/shared
```

You may still need to set read permissions on anything you copy into these directories. For example, to recursively set group read and execute permissions on a directory, allowing other `tc044` project members to read it and its contents:

```
$ chmod -R g+rX /work/tc045/tc045/shared/mydirectory
```

4.2 Modules

Environment modules are available. You can use the commands you are probably used to

```
$ module list
$ module avail
$ module load <modulename>
```

to list the currently loaded modules, see what other modules are available, and then to load a module.

In this training you will typically only need to load a compiler and an MPI library.

4.3 Running jobs via the batch system

Jobs are run via the Slurm batch system. You will be able to run jobs via the normal QoS if you like, but we have also set up reservations on Cirrus to allow our group exclusive access to ten compute nodes, ensuring quick job throughput.

In a Slurm job script the account (budget to charge), partition (group or type of nodes to run on) and QoS (type of job, determining the limits that apply) will be specified by options to the `sbatch` command used to submit the job. To use a given reservation, you must also provide its code. All in all, you should provide the following options in your scripts:

```
#SBATCH --account=tc045
#SBATCH --partition=standard
#SBATCH --qos=reservation
#SBATCH --reservation=<reservation code>
```

The reservation codes for each day are given in the following table:

Table 1: Project `tc045` reservations.

Day	Time active	Reservation code
Monday 5th June	10:00 – 13:00	`tc045_956402`
Tuesday 6th June	10:00 – 13:00	`tc045_956408`
Wednesday 7th June	10:00 – 13:00	`tc045_956414`

4.4 Running the BOUT++ example - interactively

The following example is run as an interactive job using a project reservation code (see section 4.3) which can only be used during the time periods listed in table 1.

```
export HOME=/work/tc045/tc045/shared/
source /work/tc045/tc045/shared/.bashrc
module load gcc/8.2.0 mpt/2.25
conda activate /work/tc045/tc045/shared/bpp_5_0_0_ss_0_4_2
module load bout-dep/2023-06-01
cd /work/tc045/tc045/shared/bpp_5_0_0_ss_0_4_2/examples/conduction
export OMP_NUM_THREADS=4
srun --nodes=1 --ntasks=4 --ntasks-per-node=4 \
  --cpus-per-task=4 --exclusive --time=00:10:00 \
  --partition=standard --qos=reservation --reservation=<reservation code> \
  --account=tc045 ./conduction
```

Outwidth of these times it is

```
--qos=standard
```

and the reservation flag will not be needed.

4.5 Further reading

If you would like to read more about using Cirrus, the documentation is available online at https://cirrus.readthedocs.io/.