HPC Documentation Documentation

Monash eResearch Centre

Jan 23, 2019

Contents

Important: Scheduled Outages

We have scheduled quarterly outages for M3. The purpose of this is to ensure that we have communicated outages to our HPC users in advance, and at the same time to ensure that the system is up-to-date as well as apply fixes to bugs and security vulnerabilities.

- Dates are: 06/06/18 (COMPLETED), 05/09/18 (COMPLETED), 12/12/18 (COMPLETED)

This site contains the documentation for the MASSIVE HPC systems. M3 is the newest addition to the facility.

CHAPTER 1

Help and Support

MASSIVE are committed to ensuring that all its users are fully supported so that you receive the maximum benefit from the MASSIVE HPC facilities and services.

MASSIVE provides user support by providing a User Guide, Help Desk support and consulting services.

1.1 Help Desk

For any issues with using MASSIVE or the documentation on this site please contact the Help Desk. Please see *Requesting help on M3* for more details.

Email	Phone
help@massive.org.au	+61 3 9902 4845

1.2 Consulting

For general enquires and enquiries about value added services such as help with porting code to GPUs or using MASSIVE for Imaging and Visualisation, use the following:

Email	Phone
info@massive.org.au	+61 3 9902 4845

1.3 Communications

For enquiries regarding MASSIVE news and events:

Email
communications@massive.org.au

1.4 Other

For other enquiries please contact the MASSIVE Coordinator:

Email
coordinator@massive.org.au

CHAPTER 2

Scheduled Maintenance

We have scheduled quarterly outages for M3. The purpose of this is to ensure that we have communicated outages to our HPC users in advance, and at the same time, to ensure that the system is up-to-date as well as apply fixes to bugs and security vulnerabilities.

Dates are:

- 22/03/17 (COMPLETED)
- 05/07/17 (COMPLETED)
- 20/09/17 (COMPLETED)
- 06/12/17 (COMPLETED)
- 09/01/18 (COMPLETED)
- 07/03/18 (POSTPONED)
- 14/03/18 (COMPLETED)
- 06/06/18 (COMPLETED)
- 05/09/18 (COMPLETED)
- 12/12/18 (COMPLETED)

12 DEC 2018 Maintenance - COMPLETED

The M3 scheduled maintenance is now completed.

All the jobs have been resumed.

For your interest, the following works have been conducted:

- NFS server uplift;
- Upgrade operating system on the login and management nodes; and
- Upgrade operating system on the hypervisors.

---

12 DEC 2018 Maintenance - COMING SOON

Our regular maintenance has been re-scheduled for 12 Dec starting 8:00 a.m. - 5:00 p.m.

A full system outage is required as this will impact the job scheduler. M3 will be offline during the outage window and users will not be able to access the login nodes via ssh or Strudel Desktop (remote desktop).

A reservation has been created for this outage. Jobs that would run into this outage will not start so please take a look at your script if your job doesn't start as expected in the queue.

For your interest, the following works will be conducted:

- Moving home dir to a filesystem with quota enforced;
- Adding more vCPU on the NFS server;
- Upgrade operating system on the login and management nodes; and
- Upgrade operating system on the hypervisors.

If you have any queries or concerns regarding this outage, contact help@massive.org.au.

Research network maintenance 28 Nov Wed 9:00 a.m. - 5:00 p.m.

Monash network team is going to carry out a network change to enable new routing method and automation at the switches in Monash research network. Network switches often require updating to take advantage of improved functionality and stability. During the update, each network fabric will be interrupted for around 30 seconds.

As M3 is sitting on a number of network fabrics, jobs will be paused at 9 a.m. and we are anticipating a disruption to the user home directories, Strudel desktop jobs and login access at different point throughout the day.

If you have any queries or concerns regarding this outage, contact help@massive.org.au.

---

5 SEP 2018 Maintenance - COMPLETED

For this outage, we have completed the following works:

- Upgraded storage firmware;
- Upgraded hardware firmware on the Lustre server; and
- Fixed MTU issue on the host networking;

We are running behind with our schedules due to some unforeseen issues and over the next few days, we are going to complete the following task:

- Upgrade operating system on the login and management nodes

---

5 SEP 2018 Maintenance - UPCOMING

Our regular maintenance has been scheduled for 5 Sep starting 8:30 a.m. - 5:30 p.m.

A full system outage is required as this will impact the job scheduler. M3 will be offline during the outage window and users will not be able to access the login nodes via ssh or Strudel Desktop (remote desktop).

A reservation has been created for this outage. Jobs that would run into this outage will not start so please take a look at your script if your job doesn't start as expected in the queue.

For your interest, the following works will be conducted:

- Upgrade storage firmware;
- Upgrade hardware firmware on the Lustre server;
- Fix MTU issue on the host networking;
- Upgrade operating system on the login and management nodes; and
- Apply bug fixes on the job scheduler.

---

6 JUN 2018 Maintenance - COMPLETED

The M3 scheduled maintenance is now completed.

For this outage, we have completed the following works:

- Upgrade operating system on the switches;
- Upgrade hardware firmware on the GPU nodes; and
- Maintenance on the job scheduler (SLURM).

As part of this maintenance, we have also introduced Quality of Service (QoS) to M3 queue. All existing job submission scripts should still be working. But you should start considering using QoS in the future.

Quality of Service (QoS) provides a mechanism to allow a fair model to users who ask for the right resources in the job submission script. The quality of service associated with a job will affect the job in three ways:

- Job Scheduling Priority
- Job Preemption
- Job Limits

A summary of the available QoS:

Queue	Description	Max Wall-time	Max GPU per user	Max CPU per user	Priority	Preemption
normal	Default QoS for every job	7 days	10	280	50	No
rtq	QoS for interactive job	48 hours	4	36	200	No
irq	QoS for interruptable job	7 days	No limit	No limit	100	Yes
shortq	QoS for job with short walltime	30 mins	10	280	100	No

*Priority: The higher the number the higher is the priority

For more information about how to use QoS, you can follow the link to our documentation page: http://docs.massive.org.au/M3/slurm/using-qos.html

We have also changed an algorithm in calculating fairshare in the queue by enabling priority flag in SLURM.

Lastly, we have now opened up the access to the new Skylake nodes. In order to introduce the new nodes, we have also created a new partition that consists of all the nodes, for more information about the new partition, follow the link to our user documentation page: http://docs.massive.org.au/M3/slurm/check-cluster-status.html

---

6 JUN 2018 Maintenance

Our regular maintenance has been scheduled for 6 June starting 8:30 a.m. - 5:30 p.m.

A full system outage is required as this will impact the job scheduler. M3 will be offline during the outage window and users will not be able to access the login nodes via ssh or Strudel Desktop (remote desktop).

A reservation has been created for this outage. Jobs that would run into this outage will not start so please take a look at your script if your job doesn't start as expected in the queue.

For your interest, the following works will be conducted:

1. Upgrade operating system on the switches;
2. Upgrade hardware firmware on the compute nodes; and
3. Maintenance on the job scheduler.

---

14 MAR 2018 maintenance - COMPLETED

The M3 scheduled maintenance is now completed.

As a result of running the new job scheduler, the software that was previously compiled with old libraries might not work. We have made a new openmpi module available as well, which has been compiled with the new job scheduler libraries:

openmpi/1.10.7-mlx(default)

Here is a list of software that might be affected by the change:

- abyss
- chimera
- dynamo
- emspring
- fasttree
- fsl
- gdcm
- geant4
- gromacs
- lammps
- mpifileutils
- netcdf
- openfoam
- python
- R
- relion
- scipion
- vtk

---

09 JAN 2018 Maintenance - COMPLETED

As part of the filesystem expansion, we need to completely shut down the storage; an outage is scheduled for 9 Jan. 2018 from 8:00 a.m. - 9:00 p.m.

A full system outage is required and this will impact the access to the project directories and running jobs. M3 will be offline during the maintenance window so users will not be able to access the login nodes via SSH or Strudel Desktop (remote desktop).

---

06 DEC 2017 Maintenance - COMPLETED

We are still working on m3-dtn1.massive.org.au server but job scheduler is back online. All the remaining nodes are back online.

The following works were completed during the scheduled outage:

- Physical move of two Lustre servers to make space for the filesystem expansion;
- ZFS updates on NFS services; and
- A set of scripts has been pushed out to lower the resources for Strudel Large Desktop

The reason for this change is to make desktop utilization more efficient and to provide more large-scale desktops available to our user community.

From today, M3 Large desktops will be halved as follows:

Configuration	Old Large Desktop	New Large Desktop
Memory	240GB	120GB
GPU	4 x K80	2 x K80
CPU	2 x 12 Cores	1 x 12 Cores

Based on our analysis we do not expect the majority of users to notice this change or be negatively impacted.

We understand that some users genuinely require a large GPU desktop configuration change might impact you, and if you ever need to run a desktop job that would require more than two GPUs, send us a request and we are more than happy to assist and attend to the request.

---

06 DEC 2017 Maintenance - COMPLETED

The final quarterly maintenance for this year is scheduled for 6 Dec. from 8:30 a.m. - 4:30 p.m.

A full system outage is required and this will impact access to the project directories and running jobs. M3 will be offline during the maintenance window so users will not be able to access the login nodes via SSH or Strudel Desktop (remote desktop).

For your interest, the following works will be conducted:

- Physical move of two Lustre servers to make space for the filesystem expansion
- System updates for NFS services
- A set of scripts will be pushed out to lower the resources for Strudel Large Desktop

A reservation has been created for this outage. Jobs that would run into this outage will not start so please take a look at your script if your job doesn't start as expected in the queue.

Based on usage data and user demand, we are making changes to the Desktop offerings on M3. The main change is the introduction of more Large Desktop resources to users. This involves decreasing the number of the GPUs allocated to each of these jobs. This will allow more users to utilise the GPU nodes in a more efficient way. If you ever need to run a desktop job that would require more than two GPUs, send us a request and we will be more than happy to assist you.

---

20 SEP 2017 Maintenance - COMPLETED

The following works were completed during the scheduled outage:

- NIC firmware upgraded on the management servers;
- Kernel update on the m3-login1;
- Kernel update on management servers' hypervisor;
- ZFS on the NFS services were updated.

Over the next few days, we are going to continue upgrading the cluster nodes. Nodes will be offline as required but there won't be any interruption to the job scheduler.

Jobs should be running now and please report any issues you encounter with M3 to help@massive.org.au.

Attention: This documentation is under active development, meaning that it can change over time as we refine it. Please email help@massive.org.au if you require assistance.

CHAPTER 3

MASSIVE Terms of Use

Use of the MASSIVE facility is subject to Monash University's Information Technology Acceptable Use Policy [PDF] and Information Technology Acceptable Use Procedure [PDF].

When reading the Information Technology Acceptable Use Procedure [PDF], note in particular section 2 *Responsibilities of users*:

Regarding Use of Monash University Computer Accounts each authorised user is responsible for:

- The security of personally owned computers and equipment used in conjunction with the University's IT Services
- Usage of the unique computer accounts which the University has authorised for the user's benefit, these accounts are not transferable
- Selecting and keeping a secure password for each of these accounts, including not sharing passwords and logging off after using a computer
- Familiarising themselves with legislative requirements which impact on the use of IT Resources and acting accordingly, using the University IT Resources in an ethical and lawful way, in accordance with Australian laws/relevant local laws where a student is based in another country
- Cooperating with other users of the ICT facilities to ensure fair and equitable access to the facilities
- Observing the obligations under these Procedures
- Observing the Terms of Service or Acceptable Use policies of third party products or services that have been engaged by the University
- IT Resources must not be used for private commercial purposes except where the paid work is conducted in accordance with the University Practice and Paid Outside Work Policy, or the work is for the purposes of a corporate entity in which Monash University holds an interest

The University accepts no responsibility for:

- Loss or damage or consequential loss or damage, arising from the use of its IT Services for academic or personal purposes
- Loss of data or interference with files arising from its efforts to maintain the IT Services

- Users whose actions breach legislation – for further information refer to the section of this policy titled Relevant Australian Legislation, Policies and Associated Documentation

The above translates to not sharing accounts or account details. For MASSIVE this is particularly important as it is a system linked to many other systems. It is vital to restrict access to the individual that has accepted these terms and conditions, so that the system is used fairly and securely for all other users. Users are limited to one account each. If you need an additional account (e.g. for a particular instrument) contact the MASSIVE helpdesk.

In addition to the University Terms and Conditions we also ask users to follow the HPC etiquette:

1. **Login nodes are used only for light single core processing tasks e.g.**
    (a) Submitting jobs
    (b) Compiling code
    (c) Light pre/post processing of jobs
2. Moving data to and from MASSIVE should be done using the data transfer node (dtn)
3. Do not run jobs on the login nodes
4. Do not write scripts that poll the queuing system continuously (i.e. loops of less than 1 minute)

The above is important as the login nodes are a shared resource and the only entry point to MASSIVE. If all the resources or services are consumed by one user, all other users are denied access to MASSIVE.

Attention: This documentation is under active development, meaning that it can change over time as we refine it. Please email help@massive.org.au if you require assistance.

CHAPTER 4

About M3

M3 is the third stage of MASSIVE.

MASSIVE is a High Performance Computing (HPC) facility designed specifically to process complex data. Since 2010, MASSIVE has played a key role in driving discoveries across many disciplines including biomedical sciences, materials research, engineering and geosciences.

MASSIVE is pioneering and building high performance computing upon Monash's specialist Research Cloud fabric. M3 has been supplied by Dell with a Mellanox low latency network and NVIDIA GPUs.

4.1 System configuration

- Total number of cores: 4,188
- Total number of GPU cores: 741,888 CUDA cores
- Total memory: 32 TByte
- Storage capacity: 2.9 PByte Lustre parallel file system
- Interconnect: 100 Gb/s Ethernet Mellanox Spectrum network

M3 utilises SLURM scheduler to manage the resources with 5 partitions available to users, with different configurations to suit a variety of computational requirements. Details about the partitions can be found here

M3 consists of 165 nodes with the following configurations:

- **"Compute" nodes in 4 configurations**
  - **Standard Memory:**
    - Number of nodes: 22
    - Number of cores per node: 24
    - Processor model: 2 x Intel Xeon CPU E5-2680 v3
    - Processor frequency: 2.50GHz, with max Turbo frequency 3.30GHz

        - Memory per node: 128GB RAM
        - Partition: m3a
    - **Medium Memory:**
        - Number of nodes: 13
        - Number of cores per node: 24
        - Processor model: 2 x Intel Xeon CPU E5-2680 v3
        - Processor frequency: 2.50GHz, with max Turbo frequency 3.30GHz
        - Memory per node: 256GB RAM
        - Partition: m3d
    - **High-Density CPUs:**
        - Number of nodes: 48
        - Number of cores per node: 36
        - Processor model: 2 x Intel Xeon Gold 6150
        - Processor frequency: 2.70 GHz, with max Turbo frequency 3.70GHz
        - Memory per node: 192GB RAM
        - Partition: m3i
    - **High-Density CPUs with High Memory:**
        - Number of nodes: 1
        - Number of cores per node: 36
        - Processor model: 2 x Intel Xeon Gold 6150
        - Processor frequency: 2.70 GHz, with max Turbo frequency 3.70GHz
        - Memory per node: 1TB RAM
        - Partition: m3m
- **"GPU" nodes in 6 configurations**
    - **Desktops:**
        - Number of desktop sessions: 32
        - Number of cores per desktop session: 3
        - Processor model: 2 x Intel Xeon CPU E5-2680 v3
        - Processor frequency: 2.50GHz, with max Turbo frequency 3.30GHz
        - GPU model: nVidia Grid K1
        - Number of GPU per desktop session: 1
        - GPU cores : 192 CUDA cores
        - Memory per desktop session: 16GB RAM
        - Partition name: m3f
    - **K80:**
        - Number of desktop sessions: 26

    - Number of cores per desktop session: 12
    - Processor model: 2 x Intel Xeon CPU E5-2680 v3
    - Processor frequency: 2.50GHz, with max Turbo frequency 3.30GHz
    - GPU model: nVidia Tesla K80
    - Number of GPUs per desktop session: 2
    - GPU cores per card: 4,992 CUDA cores
    - Total GPU cores per desktop session: 9,984 CUDA cores
    - Memory per desktop session: 128GB RAM
    - Partition name: m3c

- **K80 with High-Density GPUs:**
    - Number of nodes: 1
    - Number of cores per node: 24
    - Processor model: 2 x Intel Xeon CPU E5-2680 v3
    - Processor frequency: 2.50GHz, with max Turbo frequency 3.30GHz
    - GPU model: nVidia Tesla K80
    - Number of GPUs per node: 8
    - GPU cores per card: 4,992 CUDA cores
    - Total GPU cores per node: 39,936 CUDA cores
    - Memory per node: 256GB RAM
    - Partition name: m3e

- **P100:**
    - Number of nodes: 14
    - Number of cores per node: 28
    - Processor model: 2 x Intel Xeon CPU E5-2680 v4
    - Processor frequency: 2.40GHz, with max Turbo frequency 3.30GHz
    - GPU model: nVidia Tesla P100
    - Number of GPUs per node: 2
    - GPU cores per card: 3,584 CUDA cores
    - Total GPU cores per node: 7,168 CUDA cores
    - Memory per node: 256GB RAM
    - Partition name: m3h

- **V100:**
    - Number of nodes: 20
    - Number of cores per node: 36
    - Processor model: 2 x Intel Xeon Gold 6150
    - Processor frequency: 2.70 GHz, with max Turbo frequency 3.70GHz

    - GPU model: nVidia Tesla V100
    - Number of GPUs per node: 3
    - GPU cores per card: 5,120 CUDA cores
    - Total GPU cores per node: 15,360 CUDA cores
    - Memory per node: 384GB RAM
    - Partition name: m3g

- **DGX1:**
    - Number of nodes: 1
    - Number of cores per node: 40
    - Processor model: 2 x Intel Xeon CPU E5-2698 v4
    - Processor frequency: 2.20GHz, with max Turbo frequency 3.60GHz
    - GPU model: nVidia Tesla GP100
    - Number of GPUs per node: 8
    - GPU cores per card: 3,584 CUDA cores
    - Total GPU cores per node: 28,672 CUDA cores
    - Memory per node: 512GB RAM
    - Partition name: TBU

CHAPTER 5

Requesting an account

Attention: If you already have an HPC ID account (used to log in to MonARCH), you do not need to create a new account.

5.1 Overview of the process

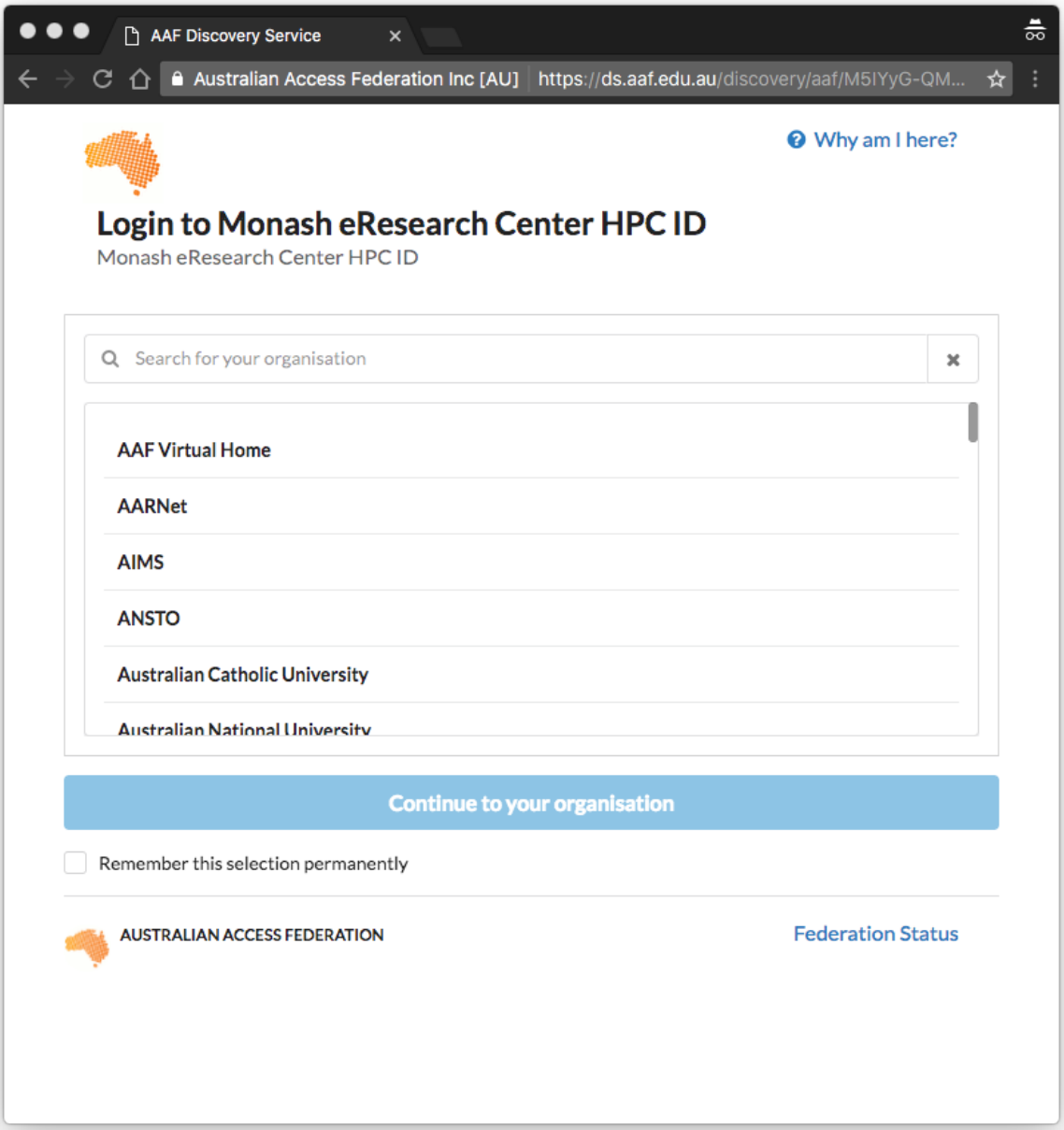
1. Log in to the HPC ID system that manages M3 accounts
2. Create an M3 account
3. Create or join an M3 project
4. Set your M3 account password
5. Log in to M3

5.1.1 1. Log in to the HPC ID system

The HPC ID system manages accounts for M3. Once you have logged in to HPC ID you can create an account on M3. If you already have a MonARCH account, you can skip this step.

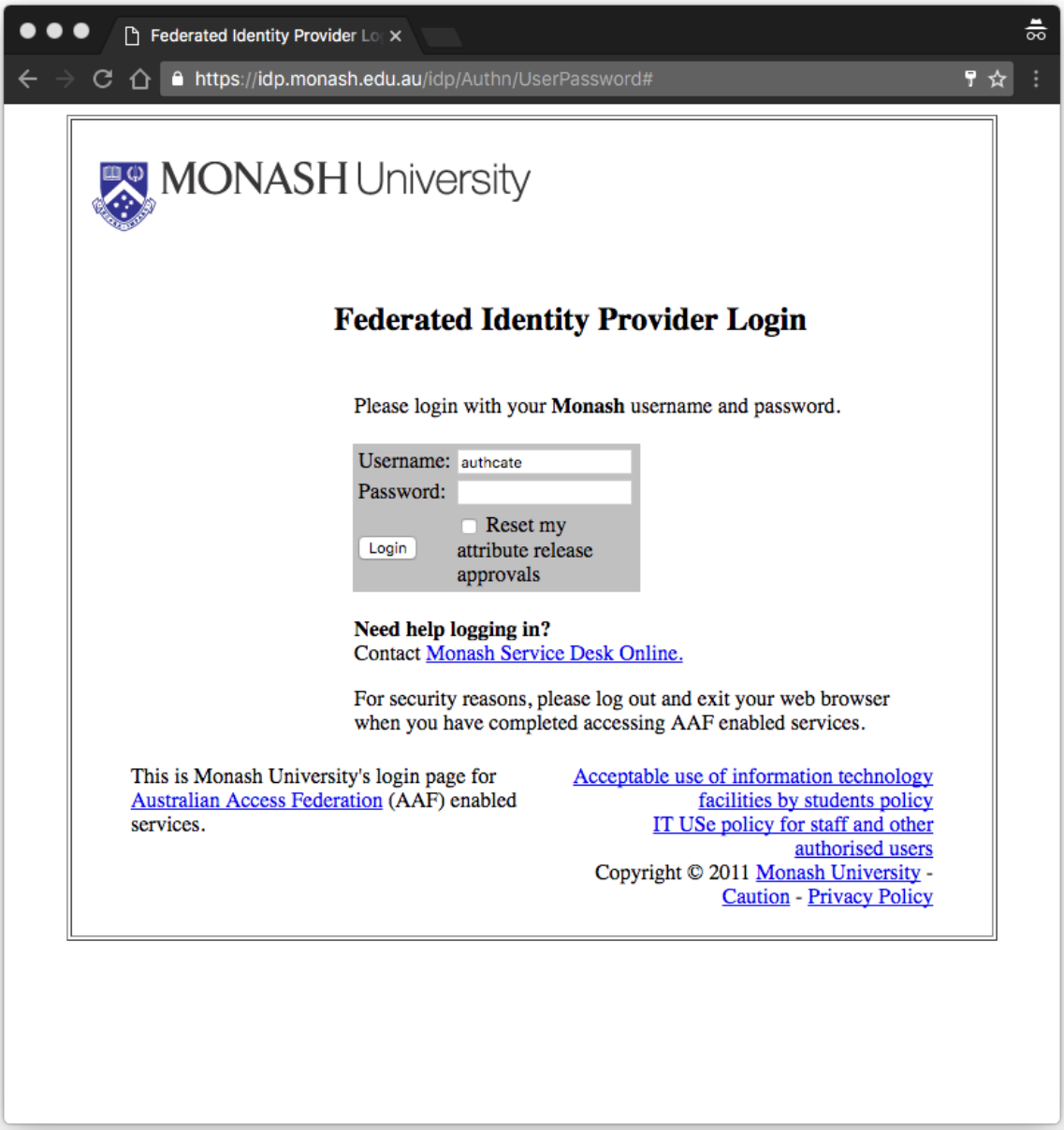
Log in to the HPC ID system.

Select your organisation. Starting typing in the search field to initiate the autocomplete function. Once you've found your organisation, select the Continue to your organisation button. If your organisation isn't displayed in the list, skip to the section below: How to request an M3 account via email.



You will be presented with the login form for the selected organisation. Enter your authentication credentials for your organisation. The image below demonstrates the Monash University login service.

Important: The first time you attempt to access the HPC ID service it will ask you for permission to access the information provided by your organisation's login process (typically your name and email address). This information will be used by the HPC ID service to authenticate you and create your HPC ID account. The form may also provide you with a range of release consent duration periods, in which case you should select one and select the Accept button to proceed.

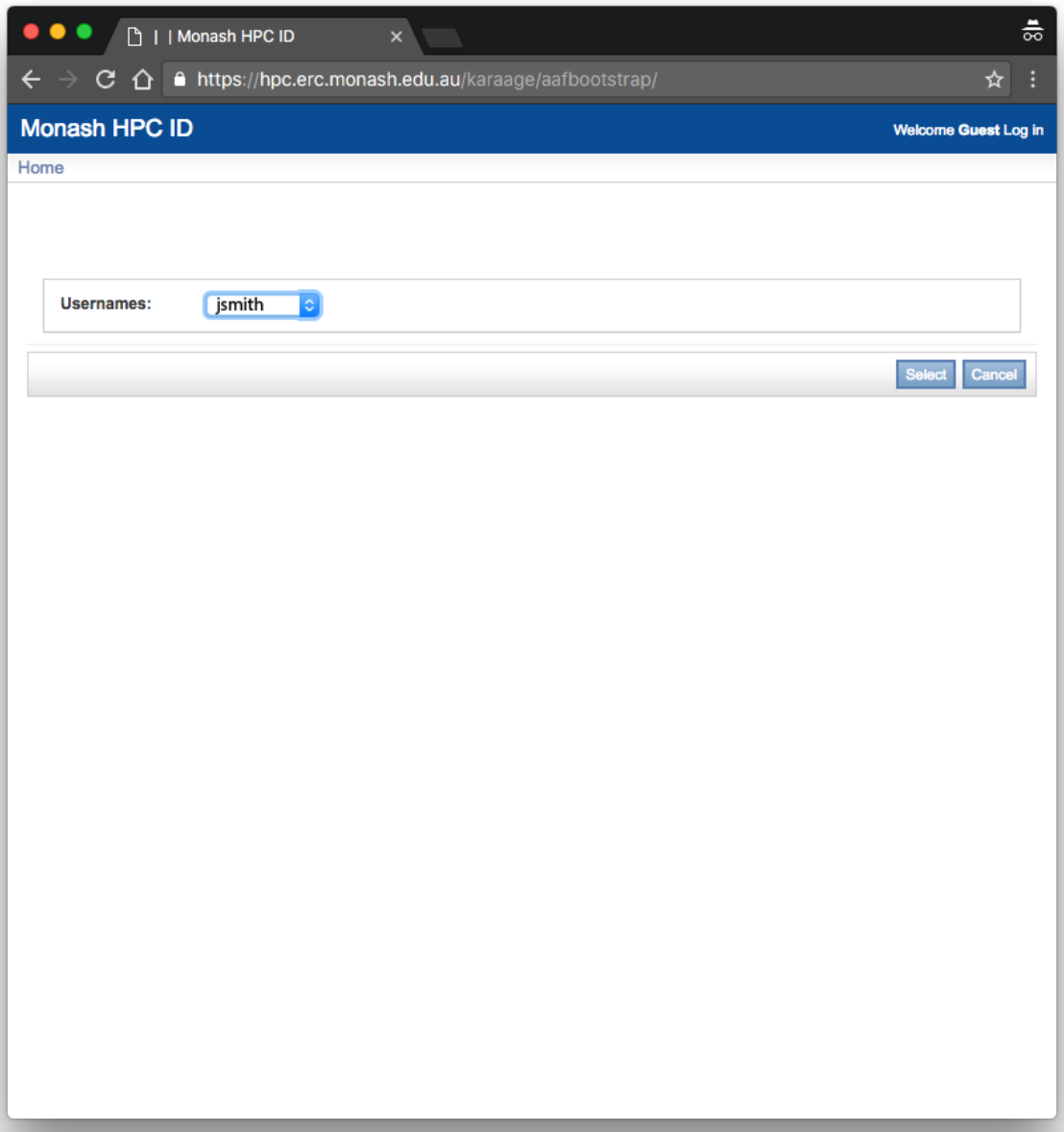

Federated Identity Provider Lo
https://idp.monash.edu.au/idp/Authn/UserPassword#
MONASH University
Federated Identity Provider Login
Please login with your Monash username and password.
Username: authcate
Password:
Login
Reset my attribute release approvals
Need help logging in?
Contact Monash Service Desk Online.
For security reasons, please log out and exit your web browser when you have completed accessing AAF enabled services.
This is Monash University's login page for Australian Access Federation (AAF) enabled services.
Acceptable use of information technology facilities by students policy
IT USe policy for staff and other authorised users
Copyright © 2011 Monash University - Caution - Privacy Policy

5.1.2 2. Create an M3 account

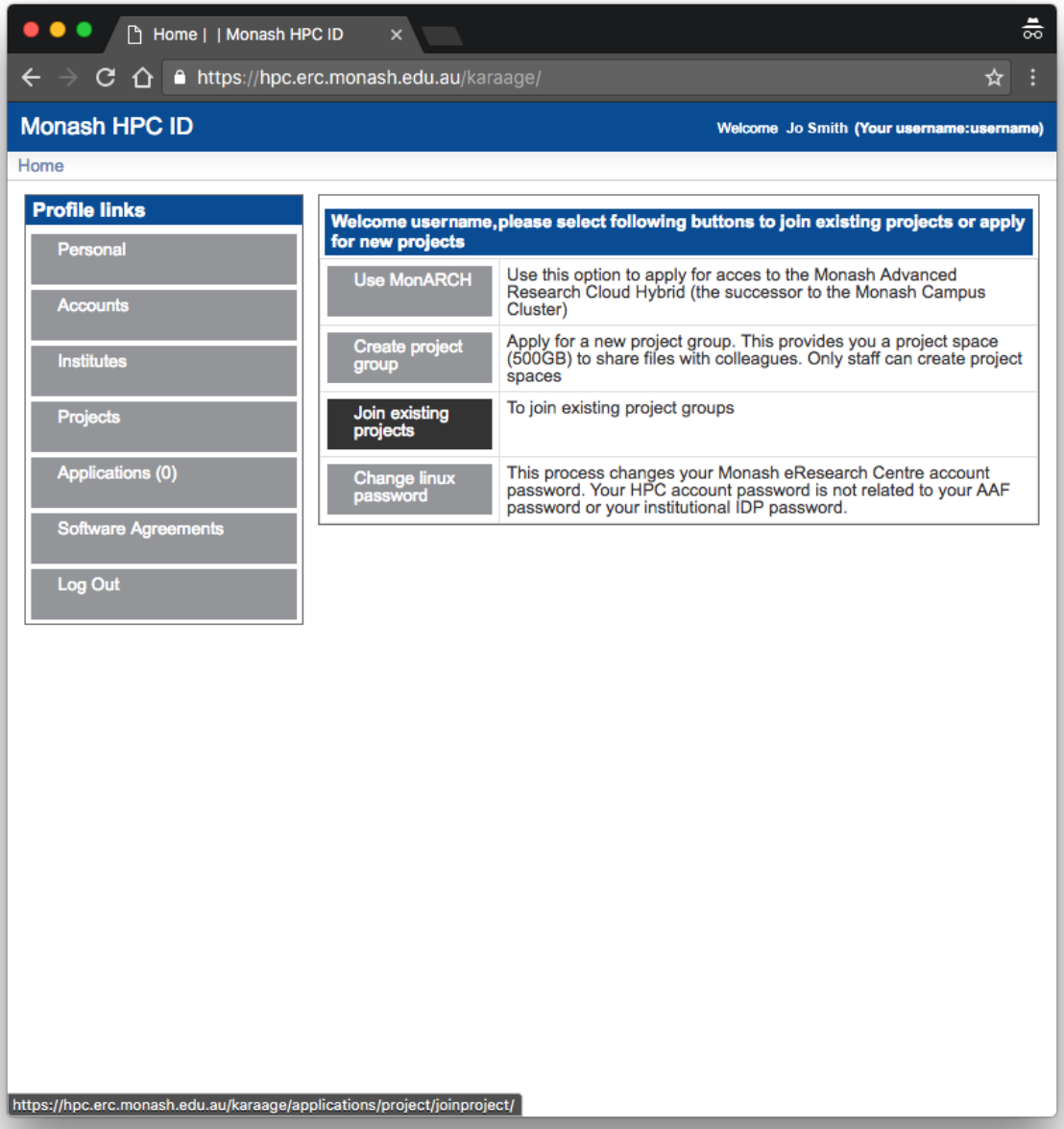
Once you've logged in to the HPC ID system you can begin the process of creating an M3 user account.

Select a username that will be used for your M3 account from the options available in the drop down list. Hit Select.

Note: You will access your account on the HPC ID system with your organisational credentials, but may have a different username and password for your M3 account.



The next screen is your HPC ID account's home page.


Home | | Monash HPC ID
https://hpc.erc.monash.edu.au/karaage/
Monash HPC ID
Welcome Jo Smith (Your username:username)
Home
Profile links
Personal
Accounts
Institutes
Projects
Applications (0)
Software Agreements
Log Out
Welcome username,please select following buttons to join existing projects or apply for new projects
Use MonARCH
Use this option to apply for acces to the Monash Advanced Research Cloud Hybrid (the successor to the Monash Campus Cluster)
Create project group
Apply for a new project group. This provides you a project space (500GB) to share files with colleagues. Only staff can create project spaces
Join existing projects
To join existing project groups
Change linux password
This process changes your Monash eResearch Centre account password. Your HPC account password is not related to your AAF password or your institutional IDP password.
https://hpc.erc.monash.edu.au/karaage/applications/project/joinproject/

5.1.3 3. Create or join an M3 project

You require an active project in order to access M3. M3 projects have codes, typically in the form of *ab12*, or *character character number number*. If you have been provided with a project code, click on the Join existing projects button and enter the code in the search field. If you do not know the project code, consult your project leader.

If you need to start a new project **do not select the Create project group button**; email help@massive.org.au instead.

Note: Only staff members can apply to create projects; PhD/RHD students must ask their supervisor(s) to apply for a new project.

5.1.4 4. Set your M3 account password

Select the option below that applies to you to proceed and set your M3 account password.

I created my account via the HPC ID system

If this is the first time you have requested a Monash HPC cluster account through the HPC ID system, you will have to set your M3 password in order to log in to M3. Note that this password is NOT the same as your organisational login password, it is specific to the HPC systems. Never disclose your password to anyone, including the MASSIVE helpdesk! To change your password, select the Change Linux Password button in the right hand panel of your account Home page. This is your HPC password, to be used for HPC services including SSH access for M3 (i.e. once your request for an account has been approved). Note that MASSIVE does not store your organisational password through the AAF login.

I requested my account via email

If you submitted your request for an account via email, the HPC team will have provided you with your username in the email response to your request. To get your password, you must contact the helpdesk by phone on 03 990 24845.

The HPC team strongly recommend you change your preset password at the earliest opportunity. There are two ways to change your password:

- Log in to M3 using SSH and change your password using the `passwd` command
- Connect to M3 using Strudel desktop client, open a terminal and use the `passwd` command.

5.1.5 5. Log in to M3

Once your account has been provisioned you will receive an email from the MASSIVE helpdesk with further instructions.

See also the *Connecting to M3* page, which includes instructions regarding software that you may need to install in order to connect to M3.

If you encounter any issues, contact the MASSIVE helpdesk.

5.1.6 Request an M3 account via email

You should only use this option if your organisation doesn't display in step 1 above. Email the MASSIVE helpdesk with the following details:

- Subject line: Request for account on M3
- Your full name
- Your organisational email address
- Name of the organisation
- Contact number (office telephone)
- Your preferred username (this should be based on your name, e.g. jsmith, jocelines, smithj)
- Project code for the project you wish to join (if known)

You will receive an email response within two business days. If your request has been approved, the message will include your new M3 username. To get your password, you must contact the MASSIVE helpdesk by phone on 03 990 24845.

Attention: This documentation is under active development, meaning that it can change over time as we refine it. Please email help@massive.org.au if you require assistance.

CHAPTER 6

Requesting help on M3

To get help with M3 issues, please email help@massive.org.au with the following details:

- The commands that you have used
- A simple example that will allow us to replicate the error
- The full path to your `slurm` submission script and input files (if applicable)

As well as the following information for specific issues:

6.1 Software issues

- The software package(s) that you are attempting to use
- What you expect the software to do
- What the software has done
- The full path to your error logs

6.2 System issues

- Error logs (with full path), or other details describing the error

6.3 Contacting us

6.3.1 Help Desk

For technical help and enquires please email help@massive.org.au.

6.3.2 General Enquiries

For general enquiries, please email info@massive.org.au or phone us on +61 03 9902 4845.

6.3.3 Postal Address

MASSIVE Coordinator
Monash e-Research Centre
Building 75, The STRIP
Monash University, Clayton Campus
Wellington Road, Clayton
Australia

Attention: This documentation is under active development, meaning that it can change over time as we refine it. Please email help@massive.org.au if you require assistance.

CHAPTER 7

Connecting to M3

There are a number of ways that you can connect to M3. To connect via a command line, see the ssh section. To connect to a desktop session (for graphical applications) see the strudel section.

Attention: This documentation is under active development, meaning that it can change over time as we refine it. Please email help@massive.org.au if you require assistance.

7.1 Connecting to M3 via ssh

To connect to M3, you will be required to use `ssh`. If you're already confident with the command line, you can simply `ssh` into M3 using your HPCID username and password.

```
ssh username@m3.massive.org.au
```

If you need further instructions, consult the relevant section below for your Operating System.

7.1.1 Linux and OS X Users

To connect to M3, Linux and OS X users need to start a terminal session. The process is very similar, however OS X users may need to take some additional steps.

Note: OS X Users: To start a terminal session, in Finder, navigate to Applications > Utilities and double click on Terminal.app

Once you have launched a terminal session, execute either one of the following commands:

```
ssh -l username m3.massive.org.au
```

```
ssh username@m3.massive.org.au
```

Enter your password at the prompt and you will be directed to the login node. If this is the first time that you have logged into M3, you will be asked if you wish to accept the host key of the node you are connecting to. This is just an identifier for the login node. Enter `yes` at the prompt and you will not be asked again; the login node's credentials will be stored on your local system.

X11 port forwarding

To enable X11 port forwarding the `-Y` or `-x` flag is required when you use the `ssh` command, depending upon which version of `ssh` you have installed on your system.

```
ssh -l username m3.massive.org.au -Y
```

To test that the session supports port forwarding, use the `xclock` or `xeyes` command.

Note: OS X Users: To enable X11 port forwarding OS X users using OS X 10.4 and above need to install the XQuartz application as X11 is no longer shipped with OS X.

7.1.2 Windows Users

The recommended software is Putty which is available from: http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

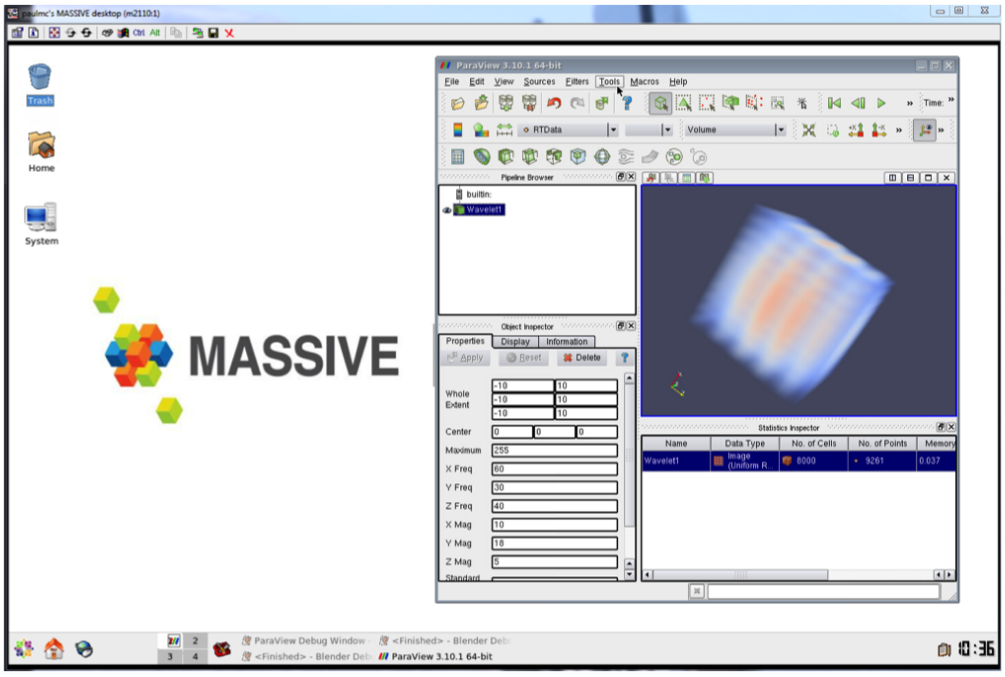
Note: if you are connected to the Monash network, you must download the `zip` file and *not* the `exe` file.

Attention: This documentation is under active development, meaning that it can change over time as we refine it. Please email help@massive.org.au if you require assistance.

7.2 Connecting to M3 via the MASSIVE desktop

What is the MASSIVE Desktop?

The MASSIVE Desktop is a remote desktop environment that allows researchers to access interactive desktop applications alongside the MASSIVE HPC environment. We provide two simple utilities to access the MASSIVE Desktop; `Strudel` and Strudel Web.



Why use the MASSIVE Desktop?

This environment is great for:

- researchers who are working with increasingly large data sets that cannot be moved to the local desktop for analysis and visualisation (for example, researchers working with high resolution volumetric data produced at the IMBL at Australian Synchrotron);
- scientists who are undertaking imaging or visualisation work;
- researchers who are relatively new to centralised computing infrastructure or HPC and don't yet have a strong grasp of the command line environment; and
- users who need to run interactive visualisation applications on high-end hardware.

What software is supported on the MASSIVE Desktop?

The list of tools and applications is increasing. Check our *Software on M3* page to see a regularly updated list. If we don't currently support the application you require then we might be able to install it. Please contact us.

7.2.1 MASSIVE Desktop (Strudel)

Quick start guide for MASSIVE Desktop

Follow this link: https://www.massive.org.au/userguide/cluster-instructions/strudel

> **Attention:** The MASSIVE team have developed a web based version of Strudel. Please feel free to trial this new service called Strudel Web.

Frequently Asked Questions

How does Strudel work and is it secure?

Strudel launches a remote desktop instance by programmatically requesting a visualisation node on the MASSIVE server, creating an SSH tunnel and launching TurboVNC.

Can I use Strudel at my HPC or cloud site?

Yes, this is easily implemented with just a simple configuration file and has been done for a range of sites other than MASSIVE or the CVL. Instructions to achieve this will be published soon. Until then please feel free to email us for more information.

I cannot find M3 under the list of the sites on Strudel Desktop, what should I do??

You can add M3 to the list of the sites on Strudel Desktop by opening Strudel, then select "Manage sites" on the menu bar. This will allow you to select "MASSIVE M3". Click OK and come back to the main screen of Strudel Desktop.

Now you should be able to select either "M3 Standard Desktop" or "M3 Large Desktop" under the "Site" section.

Can I change the size of the Desktop after it has been started?

You can change the desktop size by entering a value in the "xrandr" command from a terminal on the desktop (e.g. *xrandr -s "1920x1080"*). If that does not work check the options in TurboVNC (ctl-alt-shift-o), newer versions have a "Remote desktop size" under "Connection Options", set this to "Server" or the size you would like.

I have a slow connection can I make TurboVNC perform better?

From the options menu (ctl-alt-shift-o) you can set the connection speed "Encoding method" to one of the "WAN" options. This will reduce the quailty of the rendering but increase the interaction speed.

I have forgotten my passphrase, how do I proceed?

You can recreate your passphrase key by deleting the old one, this will prompt you to create a new passphrase when you first login with your MASSIVE id. To delete the key: Indentity > Delete Key, from the Strudel menu. You can also avoid the key by using Identity > Don't Remember Me.

7.2.2 MASSIVE Desktop in browser (Strudel Web)

The Strudel Web service offers the same functionality and easy access to MASSIVE as the Strudel desktop client, but does not require you to install any additional software on your local machine. Log in to the MASSIVE Desktop via Strudel Web.

Attention: This documentation is under active development, meaning that it can change over time as we refine it. Please email help@massive.org.au if you require assistance.

16/08/2018: We have updated instructions on how to access `market` storage.

CHAPTER 8

File Systems on M3

The M3 File System is arranged in 3 parts:

- your home directory
- your project directory
- project scratch space

In your home directory you should find a number of symbolic links (`ls -al`). These point to the project space and scratch you have been allocated. You can request additional space via help@massive.org.au

For example if your project name is "YourProject001" you will see the following two links:

```
ls -al ~/
YourProject001 -> /projects/YourProject001
YourProject001_scratch -> /scratch/YourProject001
```

The first link points to your `project` data that is backed up weekly (as with your `home` directory). The second one points to the faster system `scratch` space, which is not backed up and is used for temporary data.

8.1 What to put on each file system?

That is up to you, but as a general guide:

8.1.1 Home directory (~2GB)

This should contain all of your hidden files and configuration files. Things like personal settings for editors and other programs can go here.

8.1.2 Project directory (Shared with everyone on your project)

This area is backed up but limited in space. It should contain all of your input data, a copy of all your job scripts and final output data. It might be worth while to keep copies of any spreadsheets you might use to analyse results or any `matlab` or `mathematica` scripts/programs here as well. Basically anything that would be hard to regenerate.

Generally each user in the project should create a subdirectory in the project folder for themselves.

8.1.3 Scratch directory (Shared with everyone on your project)

This area is not backed up. Generally all your intermediate data will go here. Anything that you can recreate by submitting a job script and waiting (even the job runs for quite a long time) can go here. Anything that you can't regenerate automatically, things that you have to think about and create rather than asking the computer to calculate should go in the project directory because that is backed up.

8.2 Disk Quotas

M3 uses `soft` and `hard` quota limits.

A `soft` limit allows you to have more than your allocated space for a short period of "grace" time (in days). After the grace time has been exceeded, the filesystem will prevent further files being added until the excess is removed.

A `hard` limit prevents further files being added.

The quotas on the Project directories are much larger than the space users get in their own Home directories, so it is much better to use the Project space. Also the project space is available for all members of your project, so you can use it to share data with your colleagues.

8.3 Default Quotas for New Projects

By default, quotas for `/projects` directory will be applied as below:

- Default projects for Cryo-Electron Microscopy: 5TB
- Default project for MX: 5TB
- Other projects: 500GB

Default quota for `/scratch` directory is 3TB.

Please use the `user_info` command to view your usage and quota on all your projects and `/scratch`.

If you need higher allocation for project storage spaces, please send your request via quota request form or contact help@massive.org.au.

8.4 Scratch Usage Policies

Demand for scratch space is high so the following policies are now in force to ensure fair access to this high performance resource.

- Scratch space is only to be used for data that is actively being processed.
- The system quota (above) will allow data up to 20T to be stored for up to 30 days before new data will be prevented from being created

- Data which is older than 60 days will be targeted for deletion

Due the specialist facility nature of M3, exceptions to the above policies can be catered for and can be requested via help@massive.org.au. We can also help make archival storage (such as RDS) available on M3 for integrating into your workflows directly.

8.5 System Backups and File Recovery

The data storage on M3 is based on Lustre which distributes data across a number of disks and provides a mechanism to manage disk failures. This means that the M3 file system is fault tolerant and provides a high level of data protection.

In addition to this, the home directories are backed up to tape every night. This means that if you create a file on Tuesday, on the following day there will be a copy of the file stored in the backup system. Backups are kept for 30 days, before the system permanently removes the file to make space for new data.

File System	Type	Quota	Policy	How long are backups kept?
Home Directory	NFS	Yes	Daily Backup	30 days
Project Directory	Lustre	Yes	Daily Backup	30 days
Scratch Directory	Lustre	Yes	No	

8.5.1 File Recovery Procedure

If you delete a file/directory by mistake, you will be able to recover the file by following the following the following procedure:

- Email a request to help@massive.org.au.
- Please include the the full path to the missing data, as well as information on when it was last seen and when it was deleted
- We will be able to restore files modified within the 30 day window. Beyond that time, any changes in the file will be lost.

The project scratch space is not backed up. Files are purged from this space as new space is required.

8.6 Information for Desktop Users

Desktop users should be aware that many application and Desktop defaults dump data to your home directory. Care must be taken when dealing with large files as these can create large amounts of hidden data and that can cause your home directory to go over quota.

The following is some information for solving common issues:

8.6.1 Thumbnails Generating Too Much Data

The act of viewing large amounts of images in a file browser cause the generation of many Gigabytes of thumbnail images. To fix:

- Go to Applications - System Tools - File Browser
- At your File Browser, Go to Edit - Preference
- At preference, Go to Preview

- At Other Previewable files - Show thumbnails - Change to 'Never'
- Ok .

8.6.2 Remember to empty your trash folder

Some users may still encounter disk quota full messages when they have already removed many files from their Home directories.

Files in the trash folder count towards a user's home directory quota.

Ensure that you clear your trash folder when you exit your MASSIVE Desktop Session.

8.6.3 Already over quota?

If you are over quota, and cannot login via the desktop, you can login using a login shell and use the commands described above in "Tools for Helping Manage Files".

If you need higher allocation for project storage spaces, please send your request via quota request form or contact help@massive.org.au.

8.7 Storage outside of M3

With your M3 project, you have an allocation of storage on its high performance Lustre file system. This storage space is intended for data analyses and has a limited capacity. For large-scale, secure, and long-term research data storage, Monash University has the following offerings available through VicNode:

- `Vault` – primarily used as archive, is a tape-based system specifically for long-term storage; this is best used to free up space on your M3 project, allowing for more data to be staged into your project for analyses. For further information, please visit: https://vicnode.org.au/products-4/vault-tape-mon/
- `Market` – is backed-up storage intended for active data sets and is accessible through the Windows, Linux, or Mac desktop environments at your research laboratory for convenient sharing of data files. For further information, please visit: https://vicnode.org.au/products-4/market-mon

All additional storage requests can be lodged with the Research Storage team via the Data Dashboard or contacting researchdata@monash.edu

8.7.1 Instructions to access Ceph Market

> **Attention:** Update: 16th August 2018
>
> Issues with connecting with the method below have been resolved. Please note that unmounting is using a different flag.

The `Market` allocation is presented as a CIFS share with a given name, usually of the form: `RDS-R-<Faculty>-<Name>`. This share can be mounted within an M3 Desktop session as follows:

1. Open a Terminal window within your M3 Desktop session and issue this command:

```
gvfs-mount smb://storage.erc.monash.edu.au/shares/<sharename>
```

- Replace the `<sharename>` with the one provided by your allocation;

- Enter your Monash ID (previously known as Authcate) username, when prompted;
- enter `MONASH` when prompted to enter the "Domain"; and
- finally your Monash ID password on the "Password" prompt.

Note: `gvfs-mount` is not available on M3 login nodes, use desktop (Strudel) to access the share.

2. If successful, the mounted share will be visible through the file browser. If the user is not a member of the group, an "access denied" message will be displayed.

3. It is best to cleanly `unmount` the share when it is no longer needed, by using this command:

```
gvfs-mount -u smb://storage.erc.monash.edu.au/shares/<sharename>
```

However, the share will be automatically unmounted once the desktop session terminates.

The collection owner will/should be able to add and/or remove collaborators who can mount the share; through the eSolution's Group Management page: https://groupadmin.monash.edu/ On this page, a list of shares that you have admin privileges will appear, each of this shares will appear as: `RDS-R-<Faculty>-<Name>-Shared`.

Important Note: It is a known issue that the available storage for the share is *incorrectly* reported. Users are advised to simply ignore the warning, and allow a copy/move to proceed. We are unable to add non-Monash users to mount a share, since this authenticates against the Monash AD.

8.7.2 Instructions to access Vault

Coming soon.

In the meantime, please contact: `help@massive.org.au`.

Attention: This documentation is under active development, meaning that it can change over time as we refine it. Please email help@massive.org.au if you require assistance.

CHAPTER 9

Copying files to and from M3

To copy files to and from M3, see the appropriate section for your operating system below.

9.1 Linux and OS X Users

9.1.1 Secure Copy (scp)

Use the `scp` (Secure Copy) command for transferring files to M3. The following example copies a local file in your current directory to the `destinationdirectory` on M3. If you do not have ssh keys configured, you will be asked for your password.

```
scp afile username@m3-dtn.massive.org.au:~/destinationdirectory/
```

To copy files from M3 to your local machine, reverse the command. When run on your local machine, the following example copies a file on M3 to your current directory on your local machine.

```
scp username@m3-dtn.massive.org.au:~/sourcedirectory/afile afile
```

To copy files from M3 to another system, specify a username and hostname for both the source and destination. When run on a login node, the following example copies a file on M3 to the destination directory on another system (assuming that you have `ssh` access to the remote system).

```
scp afile usernameatanothersystem@anothersystem.org.au:~/destinationdirectory
```

To recursively copy all files in a directory, add the `-r` flag.

```
scp -r adirectory usernameatanothersystem@anothersystem.org.au:~/destinationdirectory/
```

For more details type `man scp` on M3.

9.1.2 rsync

Use `rsync` to synchronise file systems and to transfer large amounts of files, with the ability to stop and restart the file transfers. `rsync` will replicate all files in a folder from one spot to another. It first analyses both file systems to find the difference and then transfers only the changes.

A typical command to synchronise files from M3 to a local folder is:

```
rsync -auv -e ssh adirectory username@m3-dtn.massive.org.au:~/destinationdirectory/
```

`rsync` is very powerful and has many options to help transfer data. For example it can delete unwanted files (`--delete`), compress data before transfer (`-z`) or can you let you see what command options might do without actually executing them (`--dry-run`). For more info on `rsync` try `man rsync`.

9.1.3 OS X

Cyberduck is our recommended client for OS X users, however other clients like FileZilla are available and may have better performance. Whichever client you use, please be aware that some download links may try to get you to install additional unnecessary software. If you have concerns about which software to download and which to ignore, please contact your institutional IT department.

9.2 Windows Users

Warning: This content is under development, please check back shortly.

CHAPTER 10

Copying files to M3 from M2

To copy files from M2 to M3

Connect to Data Transfer Node on M3 to do your rsync or scp

Login to *m2.massive.org.au*, then run command:

```
rsync -auv -e ssh somedirectory {username}@m3-dtn.massive.org.au:~/{project_dir}
```

Note that:

- *username* on M3 may not be the same as on M2.
- *project_dir* is a symlink to your M3 project directory within your M3 home directory.

CHAPTER 11

Software on M3

M3 uses a modular system to manage software.

11.1 Modules

`Modules` is software installed on M3 that provides an easy mechanism for updating your environment variables, so that your environment uses the correct version of software and libraries installed on M3.

11.2 Installed software modules

```
------------------------ /usr/share/Modules/modulefiles ----------------------
↪--
dot         module-git  module-info modules     null        use.own

------------------------ /usr/local/Modules/modulefiles ----------------------
↪--
3daprecon/0.0.1
3depict/0.0.15
3dslicer/4.6.0
3dslicer/4.8.1
abaqus/6.14
abyss/2.0.2
adxv/1.9.12
afni/16.2.16
afni/17.0.11
amber/18-multi-gpus
amber/18-parallel
amber/18-serial
amber/18-single-gpu
amira/6.3.0
```

```
anaconda/4.3.1-Python3.5
anaconda/4.3.1-Python3.5-gcc5
anaconda/5.0.1-Python2.7-gcc5
anaconda/5.0.1-Python3.5-gcc5
anaconda/5.0.1-Python3.6-gcc5
anaconda/5.1.0-Python3.6-gcc5
analyze/12.0
ansys/18.1
ants/1.9.v4
ants/2.2.0
arpack/2.1
arpack/3.1.3-2
ascp/3.5.4
atlas/3.10.2-gcc4
atlas/3.10.2-gcc5
atomprobedevcode/1.0.0(default)
attr/2.4.46-12
autodock_vina/1.1.2
avizo/9.0.1
avizo/9.3.0
avizo/9.3.0.1
avizo/9.4.0
axel/2.12
bamtools/2.4.1
bcbtoolkit/4.0.0
bcftools/1.6
bcftools/1.7
bcl2fastq/2.19.1
beagle/2.1.2
beast1/1.10.0
beast1/1.8.4
beast2/2.4.7
bedtools/2.26.0
bigdatascript/v0.99999e
biscuit/0.2.2
bismark/v0.19.1
blas/3.8.0-gcc5
boost/1.58.0
boost/1.62.0
boost/1.62.0-gcc4
boost/1.67.0-gcc5
bowtie/1.1.2
bowtie2/2.2.9
bsoft/1.9.2
bsoft/2.0(default)
buster/20170508
bwa/0.7.12
bzip2/1.0.6
caffe/1.0.0
caffe/1.0.0-protbuf32
caffe/caffe-matlab
caffe/caffe-tsn
caffe/deepvistool
caffe/latest(default)
```

```
caffe/rc4
caffe2/0.8.1
canu/1.7.1
caret/5.65
caw/0.2.4(default)
cblas/20032302-gcc5
ccp4/7.0
ccp4/ccp4i
ccp4/ccp4i2
ccp4/ccp4mg
ccp4/coot
cellprofiler/2.2.0
cellprofileranalyst/2.2.0
cellranger/2.0.1
chimera/1.10.2
chimera/1.11
chrome/default
cistem/1.0.0-beta
cloudstor/2.3.1-1.1
cmake/3.10.2-gcc4
cmake/3.10.2-gcc4-system
cmake/3.10.2-gcc5
cmake/3.5.2
cmake/3.5.2-gcc4
cmake/3.5.2-gcc5
comsol/5.2a
connectome/1.2.3
convert3d/1.0.0
coventormp/1.002
cplex/12.8.0
cpmd/3.17.1
cryosparc/beta
crystallography/0.0.3(default)
cst/2017
ctffind/4.0.17
ctffind/4.1.10
ctffind/4.1.3
ctffind/4.1.4
ctffind/4.1.8
cuda/4.1
cuda/6.0
cuda/7.0
cuda/7.5
cuda/8.0
cuda/8.0.61
cuda/8.0-DL
cudnn/5.1
cudnn/5.1-DL
cudnn/7.1.2-cuda8
cufflinks/2.2.1
cutadapt/0.16
cytoscape/3.4.0
daris-utils/1.0
deep-complex-networks/2017
```

```
deepmedic/0.6.1
detectron/20180322
dials/1.5.1
dicomnifti/2.32.1
dmtcp/2.5.2
dos2unix/7.4.0
drishti/2.6.3
drmaa/1.0.7
dti-tk/2.3.1
dtk/0.6.4.1
dynamo/1.1.178
eclipse/4.7.3a
effoff/0.2.1(default)
eigen/3.2.9
eigen/3.3.0
eiger2cbf/1.0
eman/2.12
eman/2.2
emapaccess/1.0(default)
emap-galaxy-shortcut/1.0.0(default)
emap-mytardis-shortcut/1.0.0(default)
emap-wiki-shortcut/0.0.1(default)
emspring/spring_v0-84-1470
emspring/spring_v0-84-1470_mlx
fastqc/0.11.7
fasttree/2.1.10
fastx-toolkit/0.0.13
fcsalyzer/0.9.12
feedback/1.0.1(default)
ffmpeg/3.4.2
fftw/3.3.4-gcc
fftw/3.3.5-gcc
fftw/3.3.5-gcc5
figtree/1.4.3
fiji/20160808
fiji/20170223
fiji/current
fiji/MMI-MHTP
fix/1.064
flexbar/3.4.0
fmriprep/1.0.15
fmriprep/1.1.1
fouriertransform/0.2.3(default)
freesurfer/20160922
freesurfer/5.3
freesurfer/6.0
freesurfer/devel-20171013
freesurfer/devel-20180612
fsl/5.0.11
fsl/5.0.9
fsleyes/0.22.4
fsleyes/0.23.0
gamess/16srs1
gap/4.8.10
```

```
gatk/3.4
gatk/3.7
gatk/4.0.1.1
gaussian/g16a03
gautomatch/0.53
gautomatch/0.56
gcc/4.9.3
gcc/5.4.0
gcc/6.1.0
gcc/8.1.0
gctf/0.50
gctf/0.66
gctf/1.06
gctf/1.06_cuda8
gctf/1.08_cuda8
gctf/1.18
gdcm/2.6.6-gcc4
gdcm/2.6.6-gcc5
geant4/10.02.p03
geant4/10.03.p01
geos/3.6
gflags/master
gflags/master-gcc4
gimp/2.8
gingerale/2.3.6
git/2.8.1
glew/2.0-gcc4
glew/2.0-gcc5
glog/master
glog/master-gcc4
glpk/4.60
gmp/6.1.2
gmsh/3.0.3
gnuparallel/20160822
gnuplot/5.2.1
gpu_burn/0.9
gpucomputingsdk/4.0.17
graphviz/2.30.1
gromacs/2016.3-openmpi-cuda8.0
gromacs/2016.4-openmpi-cuda8.0
gromacs/2018-openmpi-cuda8.0
gromacs/2018-openmpi-cuda8.0-NVML(default)
gsl/2.2-gcc4(default)
gsl/2.2-gcc5
gst-devel/1.4.5
gst-libav/1.10.4
gst-libav/1.4.5
gurobi/7.5.1
h5toxds/1.1.0
hdf5/1.10.0-patch1
htop/2.0.1
htseq/0.10.0
htslib/1.7
huygens/16.10.1-p1
```

```
icm/3.7-3b
icm/3.8.7
idl/8.6
igv/2.3.81
ihrsr++/v1.5
ilastik/1.2.0
imagemagick/7.0.5-7
imagescope/11.2.0.780
imod/4.8.54
imod-raza/4.7.12
imosflm/7.2.1
impute2/2.3.1
intel/2015.0.090
intel/2016
intel/2017u4
iqtree/1.5.3
iqtree/1.6.2
itk/4.10.0-gcc4
itk/4.10.0-gcc5
itk/4.8.2-gcc4
itk/4.8.2-gcc5
itksnap/3.3.x(default)
jags/3.3.0
jags/3.4.0
jags/4.3.0
java/1.7.0_67
java/1.8.0_77
jdk/10-20180320
jspr/2017-7-20
kallisto/0.43.0
kilosort/1.0
lammps/20180510(default)
lammps/nil
lapack/3.6.1-gcc4
lapack/3.6.1-gcc4-opt
lapack/3.6.1-gcc5
lapack/3.8.0-gcc5
leveldb/master
leveldb/master-gcc4
levelset/0.0.2(default)
libint/1.1.4
libjpeg-turbo/1.4.2
libjpeg-turbo/1.5.1
libsmm/20150702
libtiff/3.9.7
libxsmm/1.9
lmdb/latest
macs2/2.1.1.20160309
mafft/7.310
magma/1.6.1
magma/2.0.2
mango/4.0.1
mantid/3.8.0
mantid/3.9.0
```

```
mathematica/11.0.1
mathgl/1.11.2
mathgl/2.0.3(default)
matlab/r2015b
matlab/r2016a
matlab/r2017a
matlab/r2017b
matlab/r2017b-caffe
matlab/r2018a
maven/3.3.9
merantk/1.2.1
mesa/13.0.5
mesa/default
meshlab/2016.12-gcc5
mevislab/2.8.1-gcc-64bit
miakat/4.2.6
minc-lib/2.2-git-gcc4
miniconda3/4.1.11-python3.5
mne/TF-201804
motioncor2/2.1
motioncor2/2.1.10-cuda8
motioncorr/2.1
motioncorr2/20160822
mpfr/3.1.5
mpifileutils/20170922
mrbayes/3.2.6
mrf/0.2.2(default)
mricrogl/1.0.20170207
mricron/06.2013
mricron/30apr2016
mriqc/0.9.7
mrtrix/0.3.15-gcc4
mrtrix/0.3.15-gcc5
mrtrix/0.3.16
mrtrix/20170712
muscle/3.8.31
mxtools/0.1
mytardis/0.1(default)
namd/2.12-ibverbs-smp-cuda
namd/2.12-multicore
nanofilt/201807
nccl/master
nccl/master-gcc4
netcdf/4.4.1.1
neuro_workflow/2017v2
niftilib/2.0.0
niistat/9.oct.2016
nis-elements-viewer/4.20
nn/0.2.4(default)
novactf/03.2018
objexport/0.0.4(default)
opencv/3.4.1
opencv/3.4.1-gcc4
openfoam/4.1
```

```
openfoam/5.x
openmpi/1.10.3-gcc4-mlx
openmpi/1.10.3-gcc4-mlx-cuda75
openmpi/1.10.3-gcc4-mlx-verbs
openmpi/1.10.3-gcc4-mlx-verbs-cuda75
openmpi/1.10.3-gcc5
openmpi/1.10.3-gcc5-mlx
openmpi/1.10.7-intel
openmpi/1.10.7-mlx(default)
orca/4.0.1
paml/4.9
pbzip2/1.1.13
peakseq/1.3.1
perl/5.24.0
phenix/1.11.1
phyml/3.1
picard/2.9.2
pigz/2.3.3
pigz/2.3.4
plink/1.7
plink/1.9
pointless/1.10.28
posgen/0.0.1(default)
posminus/0.2.3(default)
protobuf/master
protobuf/master-gcc4
psi4/v1.1
pv/1.6.6
pyem/v0.1
pyem/v0.1-201806
pymol/1.8.2.1
pymol/1.8.6
python/2.7.11-gcc
python/2.7.12-gcc4(default)
python/2.7.12-gcc5
python/3.5.2-gcc
python/3.5.2-gcc4
python/3.5.2-gcc5
python/3.6.2
pyxnat/20170308
qatools/1.2
qiime2/2017.9
qiime2/2018.2
qiime2/2018.4
qt/5.7.1-gcc5
quit/1.1
R/3.3.1
R/3.4.3
R/3.5.0
raxml/8.2.9
rdf-kd/0.0.1(default)
relion/1.4
relion/2.02
relion/2.0.6
```

```
relion/2.0beta
relion/2.1(default)
relion/2.1.b1
relion/2.1.b2
relion/2.1-openmpi-1.10.7-mlx
resmap/1.1.4
resmap/1.1.5
resmap/1.9.5
rest/1.8
rest/1.8-matlab2017a.r6685
rings/1.3.3
r-launcher/0.0.1(default)
root/5.34.32
rosetta/2018.09
rsem/1.3.0
rstudio/1.0.143
rstudio/1.0.44
rstudio/1.1.414
rstudioserver_epigenetics/1.0
rstudioserver_epigenetics/1.0-20171101
samtools/1.3.1
samtools/1.6
samtools/1.7
sbt/0.13.15
scalapack/2.0.2
scipion/devel
scipion/devel-20170327
scipion/v1.0.1_2016-06-30
scipion/v1.1
scipion/v1.1.1
scipion/v1.2
scipion/v1.2.1
sdm_1d_calculate/2.0.2(default)
sdm_1d_plot/0.0.4(default)
sdm_2d_calculate/2.0.2(default)
sdm_2d_plot/0.0.4(default)
shapeit/v2_r837
simnibs/2.0.1g
simple/2.1
simple/2.5
singularity/2.2
singularity/2.3.1
singularity/2.4
singularity/2.4.2
singularity/2.4.5
singularity/2.5.1
singularity/c840a691d772c5a9d3981176ff3a1b849c6221c8
singularity/d3d0f3fdc4390c7e14a6065543fc85dd69ba42b7
singularity/dc816fa86ac79a6f20a1b1e8592a571498148825
skewer/20170212
smux/0.0.1
snappy/master
snappy/master-gcc4
snpm/13
```

```
soapdenovo2/2.04-r241
sourcetracker/2.0.1
sparsehash/2.0.3
spider/21.11
spm12/matlab2015b.r6685
spm8/matlab2015b.r6685
spm8/matlab2017a.r6685
squashfs-tools/4.3-0.21
sra-tools/2.7.0
star/2.5.2b
stata/14
strelka/2.8.4
structure/2.3.4
subread/1.5.1
surfice/7_feb_2017
synopsys/3.1(default)
tannertools/2016.1
tannertools/2016.2
tapsim/v1.0b_r766
tbb/20180312oss
tempest/1.5
tensorflow/1.0.0-python2.7.12-gcc5
tensorflow/1.3.0-python2.7.12-gcc5
tensorflow/1.4.0-python2.7.12-gcc5
tensorflow/1.4.0-python3.6-gcc5
terastitcher/20171106
texlive/2017
tiff/4.0.8
tigervnc/1.8.0
tmap/3.0.1
tophat/2.1.1
tracer/1.6
trackvis/0.6.1
trim_galore/0.4.5
trimmomatic/0.38
turbovnc/2.0.2(default)
turbovnc/2.1.0
ucsc-genome-tools/201806
unblur/1.0.2
unrar/5.0
valgrind/3.13
varscan/2.3.9
vasp/5.4.4
vep/90
vim/8.0.0596
virtualgl/2.5.0(default)
virtualgl/2.5.2
visit/2.12.3
vmd/1.9.3
voro++/0.4.6
vt/0.57
vtk/7.0.0
vtk/7.0.0-gcc5
wfu_pickatlas/3.0.5b
```

```
workspace/4.0.2
wxgtk/3.0.2
wxwidgets/3.0.3
xds/20170302
xds/monash(default)
xds/mxbeamteam
xjview/9.0
xjview/9.6
xnat-desktop/0.96
xnat-utils/0.2.1
xnat-utils/0.2.5
xnat-utils/0.2.6
xvfb/1.19.3
yade/1.20.0-cpu
yade/1.20.0-gpu
yasm/1.2.0-4
zetastitcher/0.3.3
```

11.3 Requesting an install

If you require additional software please email help@massive.org.au with the following details:

- software
- version
- URL for download
- urgency
- if there are any licensing restrictions we need to impose

11.4 Docker based work flows

Many fields are beginning to distribute fully self contained pieces of software in a container format known as docker. Unfortunately docker is unsuited as a container format for shared user systems, however it is relatively easy to convert most docker containers for scientific work flows to the Singularity format. If you wish to run software based on a Docker container, please email help@massive.org.au and let us know where we can obtain the container and we will be happy to convert it for you.

11.5 Running QIIME on M3

QIIME (Quantitative Insights Into Microbial Ecology) 2 is installed on M3. To use this software:

```
 # Loading module
 module load qiime2/2017.9

# Unloading module
 module unload qiime2/2017.9
```

If you encounter issues with this install, please contact help@massive.org.au

CHAPTER 12

Running jobs on M3

Launching jobs on M3 is controlled by SLURM, the Slurm Workload Manager, which allocates the compute nodes, resources, and time requested by the user through command line options and batch scripts. Submitting and running script jobs on the cluster is a straight forward procedure with 3 basic steps:

- Setup and launch
- Monitor progress
- Retrieve and analyse results

For more details please see:

12.1 Partitions Available

Nodes belong to different partitions which allow corresponding jobs to run on them. There are two partitions on m3 at the moment:

The default partition is `comp` which consists of:

- m3a: 21 nodes, each with 24 cores plus 128GB RAM
- m3c: 13 nodes, each with 24 cores, 4 x nVidia Tesla K80, 256GB RAM
- m3d: 13 nodes, each with 24 cores, 256GB RAM
- m3h: 14 nodes, each with 28 cores, 2 x nVidia Tesla P100, 256GB RAM
- m3i: 43 nodes, each with 36 cores, 196GB RAM

12.2 Other partitions

- short:
  - use this when the jobs can be completed within ten minutes

  - two nodes, each with 24 cores, 100GB RAM

- **rtqp:**
  - real-time partition to be used on demand
  - batch jobs shouldn't be running here
  - use this partition only with two QoS, rtq and irq
  - ten nodes, a total of 412 cores, three nodes with V100 and one node with P100

12.3 Checking the status of M3

On M3, users can check the status of all nodes via the `show_cluster` command. The output of this command should be similar to:

```
$ show_cluster
    NODE             TYPE      PARTITION*         CPU      Mem (MB)   GPU/Phi
↪STATUS
                                                (Free)       (Free)    (Free)
    ------------------------------------------------------------------------------
↪-----
    m3a000            CPU         m3a/all        (24)      (116588)     (N/A)
↪Reserved
    m3a001            CPU         m3a/all        (24)      (116588)     (N/A)
↪Reserved
    m3a002            CPU         m3a/all        (24)      (116588)     (N/A)
↪Reserved
    m3a003            CPU         m3a/all          24        116588       N/A
↪Idle
    m3a004            CPU         m3a/all          24        116588       N/A
↪Idle
    m3a005            CPU         m3a/all          24        116588       N/A
↪Idle
    m3a006            CPU         m3a/all          24        116588       N/A
↪Idle
    m3a007            CPU         m3a/all          24        116588       N/A
↪Idle
    m3a008            CPU         m3a/all        (24)      (116588)     (N/A)
↪Reserved
    m3d007            CPU         m3d/all           8         15614       N/A
↪Running
    m3d008            CPU             m3d           8         15614       N/A
↪Running
```

12.3.1 The STATUS field explained

The `STATUS` field can show:

- Idle - Node is completely free. No jobs running on the node.
- Running - Some jobs are running on the node but it still has available resources for new jobs.
- Busy - Node is completely busy. There are no free resources on the node. No new jobs can start on this node.
- Offline - Node is offline and unavailable due to a system issue.

- Reserved - Node has been booked by other users and is *ONLY* available for them.

12.4 Slurm Accounts

Slurm accounts are used to account for CPU/GPU usage, as well as setting job priorities, and are an important part of the job scheduler. Each M3 project has a corresponding Slurm account.

12.4.1 Default accounts

Some users on M3 will have a single project, which means that they'll have a single project code and won't need to specify an account. Other users will have multiple projects, which means they'll have multiple Slurm accounts and may need to specify an account.

To view your default slurm account:

```
sacctmgr show user $USER format=User,DefaultAccount
```

To change your default slurm account:

```
# Replace nq46 with your account code
sacctmgr modify user where name=$USER set DefaultAccount=nq46
```

12.4.2 Setting the account for a job

Depending on how you're accessing M3, the mechanism for setting the account to charge a job to changes:

sbatch and smux

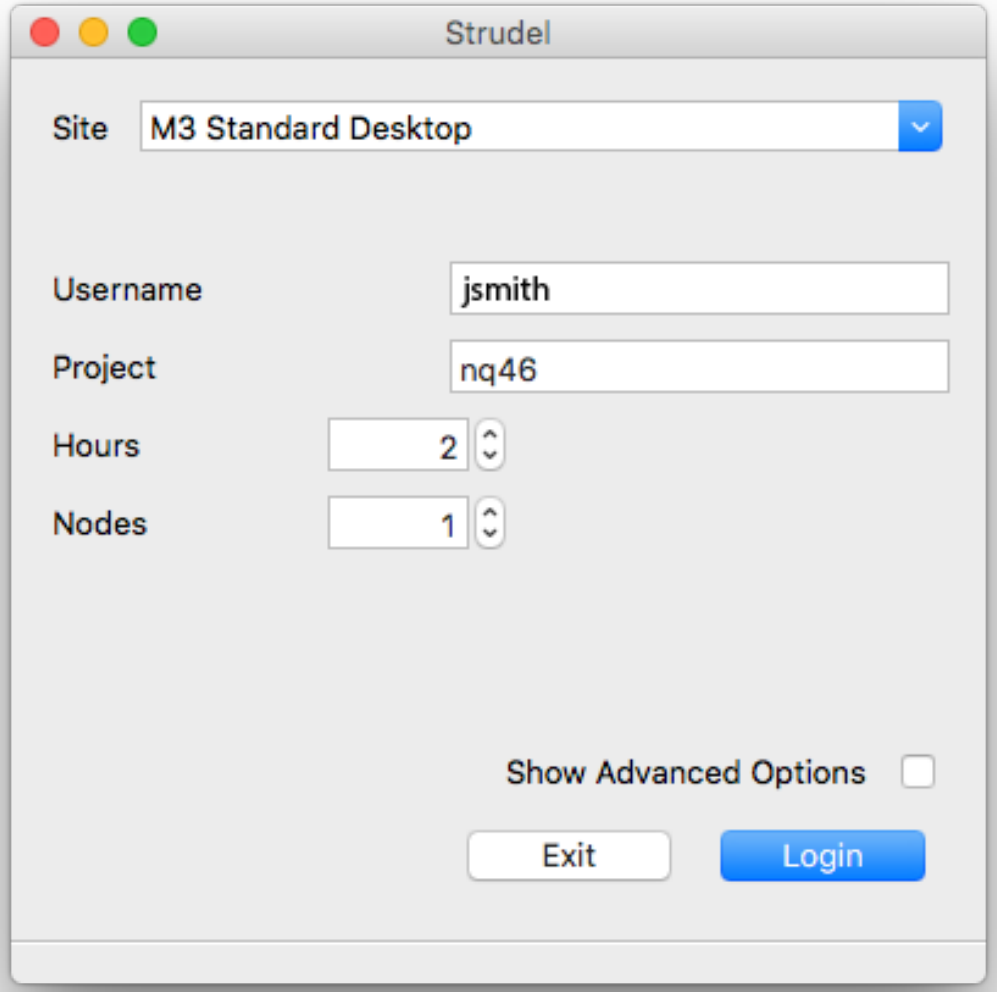
To specify an account for `sbatch` and `smux` jobs, use the `-A` or `--account` option:

```
sbatch --account nq46 job.script
# OR
sbatch -A nq46 job.script
```

```
smux new-session --ntasks=12 --account=nq46
# OR
smux new-session --ntasks=12 -A=nq46
```

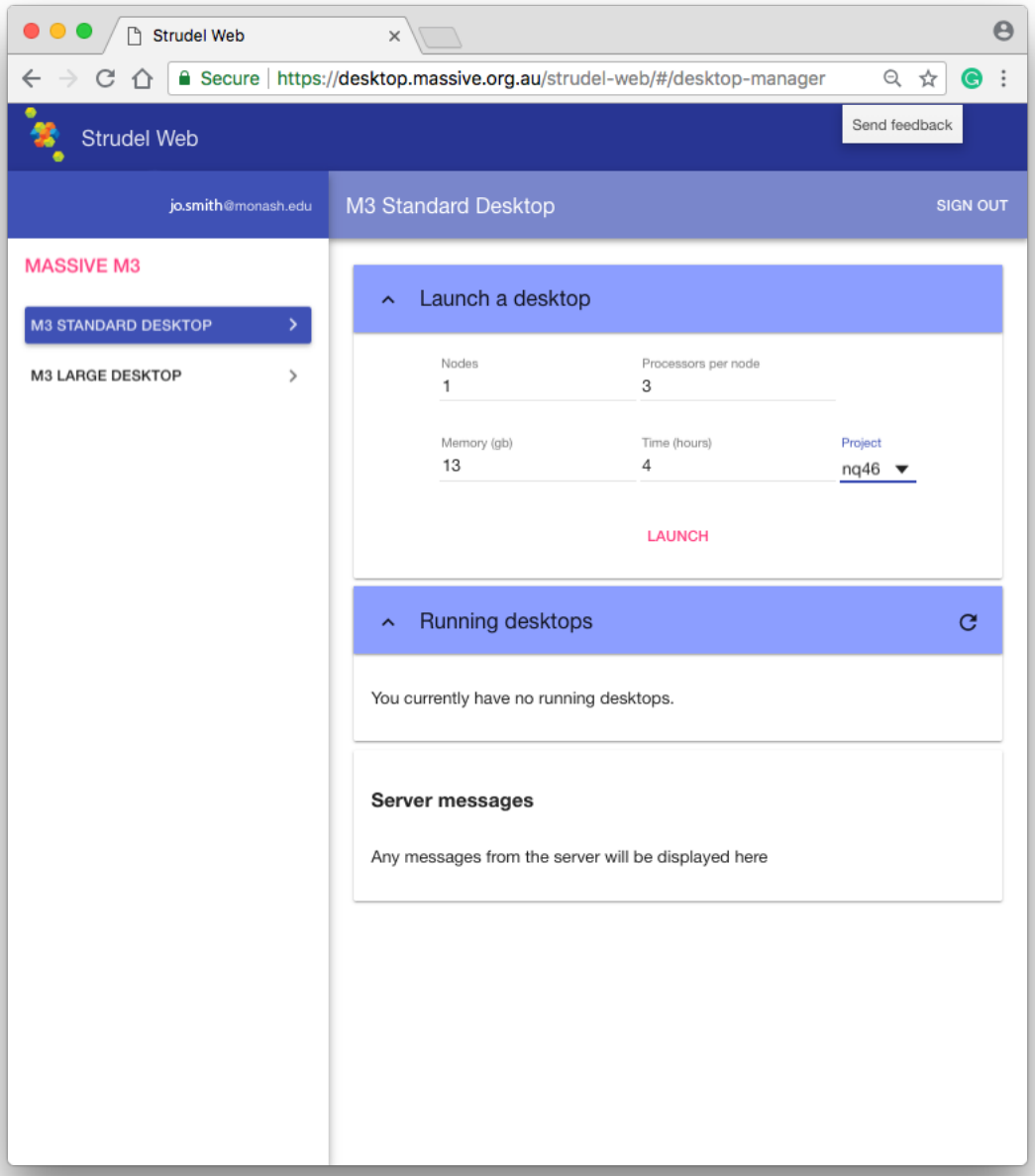
Strudel Desktop

To specify an account for Strudel Desktop, enter the account code in the *Project* box:



Strudel Web

To specify an account for Strudel Web, when requesting a desktop, use the drop down menu in the *Launch a desktop* section.



12.4.3 Questions about slurm accounts

If you have any enquiries with regards to your project resources and Slurm accounts, please do not hesitate to contact us on help@massive.org.au

12.5 Quick Start

A submission script is a shell script that consists of a list of processing tasks that need to be carried out, such as the command, runtime libraries, and input and/or output files for the tasks. If you know the resources that your tasks need

to consume, you may also modify the SBATCH script with some of the common directives, e.g.:

```
Short Format    Long Format             Default         Description
------------    -----------             -------         -----------
-N count        --nodes=count           One             One node will be used.  Used␣
↪to allocate count nodes to your job.
-A accountID    --account=accountID     One             Enter the account ID for your␣
↪group. You may check your available account(s) with id command.
-t HH:MM:SS     --time=HH:MM:SS         02:00:00        Always specify the maximum␣
↪wallclock time for your job, max is 7 days.
-p partition    --partition=partition   m3a             Always specify your partition␣
↪(i.e. m3c, m3d, m3f)
-n count        --ntasks                One             Controls the number of tasks␣
↪to be created for the job
N/A             --ntasks-per-node       One             Controls the maximum number␣
↪of tasks per allocated node
-c count        --cpus-per-task         One             Controls the number of CPUs␣
↪allocated per task
N/A             --mem-per-cpu           4096MB          Memory size per CPU
-m size         --mem=size              4096MB          Total memory size
-J jobname      --job-name=job_name     slurm-{jobid}   Up to 15 printable, non-
↪whitespace characters
N/A             --gres=gpu:1            N/A             Generic consumable resources␣
↪e.g. GPU
N/A             --no-requeue            --requeue       By default, job will be␣
↪requeued after a node failure
```

12.6 Running Simple Batch Jobs

Submitting a job to SLURM is performed by running the `sbatch` command and specifying a job script.

```
sbatch job.script
```

You can supply options (e.g. `--ntasks=xx`) to the `sbatch` command. If an option is already defined in the `job.script` file, it will be overridden by the commandline argument.

```
sbatch [options] job.script
```

12.6.1 An example Slurm job script

```
#!/bin/bash
#SBATCH --job-name=MyJob
#SBATCH --account=nq46
#SBATCH --time=01:00:00
#SBATCH --ntasks=1
#SBATCH --mem-per-cpu=4096
#SBATCH --cpus-per-task=1
./helloworld
```

This script describes the job: it is a serial job with only one process (`--ntasks=1`). It only needs one CPU core to run the `./helloworld` process. The default memory per CPU has been set to 4GB and you should adjust the script based on how much your job needs.

12.6.2 Cancelling jobs

To cancel *one* job

```
scancel [JOBID]
```

To cancel *all* of your jobs

```
scancel -u [USERID]
```

12.7 Running MPI Jobs

The Message Passing Interface (MPI) is a library specification for message-passing. It is a standard API (Application Programming Interface) that can be used to create parallel applications. An MPI job can considered as a cross-node and multi-process job.

12.7.1 An example Slurm MPI job script

```
#!/bin/bash
#SBATCH --job-name=MyJob
... ...
#SBATCH --ntasks=32
#SBATCH --ntasks-per-node=16
#SBATCH --cpus-per-tasks=1
... ...
module load openmpi-gcc
mpiexec <program>
```

This script tells Slurm this is a multi-processing job. It has 32 MPI processes, with 16 MPI processes on each node (implicitly requesting 2 nodes). For each MPI process, it needs 1 CPU core.

12.8 Running Multi-threading Jobs

Multi-threading is a type of execution model that allows multiple threads to exist within the context of a process. Simply speaking, a Slurm multi-threading job is a single process, multi-core job. Many application can belong to this category

- OpenMP programs,
- Matlab programs with (Parallel Computing Toolbox) enabled,
- and so on

12.8.1 An example Slurm Multi-threading job script

```
#!/bin/bash
#SBATCH --job-name=MyJob
... ...
#SBATCH --ntasks=1
#SBATCH --cpus-per-tasks=8
```

(continues on next page)

(continued from previous page)

```
... ...
./<program>
```

This script tells Slurm this is a multi-threading job. It has only 1 process, but that process needs 8 CPU cores to execute.

12.9 Running Interactive Jobs

12.9.1 Submitting an Interactive Job

Interactive sessions allow you to connect to a compute node and work on that node directly. This allows you to develop how your jobs might run (i.e. test that commands run as expected before putting them in a script) and do heavy development tasks that cannot be done on the login nodes (i.e. use many cores).

To launch an interactive job on MASSIVE, use the `smux new-session` command. Please note it will launch the job with the following defaults:

- `--ntasks=1` (1 cpu core)
- memory=4G
- time=2 hours
- no other resources such as GPU

`smux` has a complete help available by typing `smux -h` or just `smux`. Commands such as `new-session` and `attach-session` can be abbreviated to `n` and `a`. It is designed to mirror the options used by `tmux`.

If you have a reservation and your reservation nodes are not in the default partition

```
smux new-session --reservation=hpl --partition=m3c
```

If you want to change default time to 2 days

```
smux new-session --time=2-00:00:00
```

If you need multi-core

```
smux new-session --ntasks=12
```

If you need two GPU cards

```
smux new-session --ntasks=2 --gres=gpu:2 --partition=m3c
```

Available partitions:

Partition	Types of GPU	Per node
m3c	K80	4
m3h	P100	2

If the job launches immediately (as it should most of the time) you will be connected automatically. If it does not launch immediately, you can use `smux attach-session` or just `smux a` once it hast started.

12.9.2 How long do interactive jobs last for?

Interactive Jobs will remain active until "exit" or the job is cancelled

The mechanism behind the interactive job is:

- User runs the `smux new-session` command
- `smux` schedules a Slurm batch job to start a `tmux` session on a compute node
- Slurm grants the user `ssh` access to the node
- `smux attach-session` connects the user to the node and attaches to the `tmux` session
- The job is completed whenever the `tmux` session ends or the job is cancelled
- Therefore, an interactive job will not be automatically terminated unless the user manually ends the session.

To end a session:

- Option 1: Run `exit` on the compute node. This terminates the `smux` job and returns you to the login node.
- Option 2: Cancel the job directly (from compute nodes or login nodes) using `scancel [JOBID]`.

12.9.3 Reconnecting to/Disconnecting from an Active Interactive Job

Since an interactive job is a `tmux` session, you can reconnect to/disconnect from it at any time. Here is a real-world scenario.

I am in the office and have an interactive job running (ID#70064). Now I plan to go home but I want to leave this job running so I can reconnect to it when I am home. The steps are:

1. Disconnect the screen session for the existing interactive job. You can either just close your laptop or the terminal and walk away, or type "ctrl-b d" (That is, hold the ctrl key, and press the b key, release both keys then press the d key) (ctrl-b is the standard tmux escape sequence, it can be changed)
2. Now `show_job` to see if the job is still running:

```
JOBID           JOB NAME     Project          QOS       STATE     RUNNING       TOTAL␣
↪NODE                     DETAILS
                                                                      TIME        TIME
-------------------------------------------------------------------------------------
↪-----------------------------
70064          _interact     default      normal     Running          7    1-00:00      1␣
↪                      m3a009
```

3. Once I am home, I can reconnect

```
m3-login1:~ ctan$ smux a
```

Note: to access to the cluster node, first you need to ssh to the login nodes

Please visit *Connecting to M3* for more information

4. Continue working until the wall-time limit is reached, or I end the job.

12.10 Running GPU Jobs

12.10.1 Running GPU Batch Jobs

Requesting GPU resources

M3 is equipped with the following NVIDIA GPU cards:

- 60 Tesla V100
- 28 Tesla P100
- 44 Tesla K80
- 8 Grid K1

When requesting a Tesla V100 GPU, you need to specify `--partition=m3g`

```
#SBATCH --gres=gpu:V100:1
#SBATCH --account=nq46
#SBATCH --partition=m3g
```

When requesting a Tesla P100 GPU, you need to specify `--partition=m3h`

```
#SBATCH --gres=gpu:P100:1
#SBATCH --account=nq46
#SBATCH --partition=m3h
```

When requesting a Tesla K80 GPU, you need to specify `--partition=m3c`

```
#SBATCH --gres=gpu:1
#SBATCH --account=nq46
#SBATCH --partition=m3c
```

When requesting a Grid K1 GPU, you need to specify `--partition=m3f`

```
#SBATCH --gres=gpu:1
#SBATCH --account=nq46
#SBATCH --partition=m3f
```

Sample GPU Slurm scripts

To submit a job, if you need 1 node with 3 cores and 1 GPU, then the slurm submission script should look like:

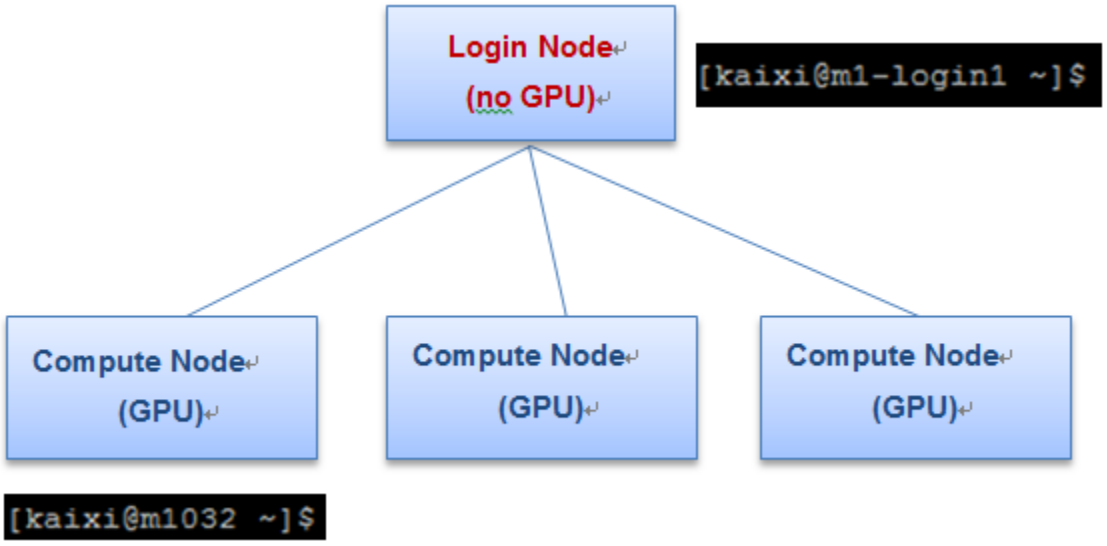
```
#!/bin/bash
#SBATCH --job-name=MyJob
#SBATCH --account=nq46
#SBATCH --time=01:00:00
#SBATCH --ntasks=3
#SBATCH --cpus-per-task=1
#SBATCH --gres=gpu:1
#SBATCH --partition=m3f
```

If you need 6 nodes with 4 cpu cores and 2 GPUs on each node, then the slurm submission script should look like:

```
#!/bin/bash
#SBATCH --job-name=MyJob
#SBATCH --account=nq46
#SBATCH --time=01:00:00
#SBATCH --ntasks=24
#SBATCH --ntasks-per-node=4
#SBATCH --cpus-per-task=1
#SBATCH --gres=gpu:2
#SBATCH --partition=m3c
```

12.10.2 Compiling your own CUDA or OpenCL codes for use on M3

M3 has been configured to allow CUDA (or OpenCL) applications to be compiled (device independent code ONLY) on the Login node (no GPUs installed) for execution on a Compute node (with GPU).



`Login node`: can compile some of CUDA (or OpenCL) source code (device independent code ONLY) but cannot run it

`Compute node`: can compile all CUDA (or OpenCL) source code as well as execute it.

We strongly suggest you compile your code on a compute node. To do that, you need to use an `smux` session to gain access to a compute node

```
smux new-session --gres=gpu:1 --partition=m3c
```

Once your interactive session has begun, load the cuda module

```
module load cuda
```

To check the GPU device information

```
nvidia-smi
deviceQuery
```

Then you should be able to compile the GPU code. Once compilation has run to completion, without error, you can execute your GPU code.

Attention: If you attempt to run any CUDA (or OpenCL) application (compiled executable) on the Login node, 'no CUDA device found' error may be reported. This is because no CUDA-enabled GPUs are installed on the Login node. You must run GPU code on a compute node.

12.11 Running Array Jobs

Job arrays allow you to run a group of identical/similar jobs. The Slurm script is EXACTLY the same. The only difference between each sub-job is the environment variable, `$SLURM_ARRAY_TASK_ID`. So it can be a good idea if you want to do some data level parallelization. E.g. let sub-job 1 `(SLURM_ARRAY_TASK_ID=1)` process data chunk 1, sub-job 2 processes data chunk 2, ... etc.

To do that, just add the following statement in your submission script, where `n` is the number of jobs in the array:

```
#SBATCH --array=1-n
```

12.11.1 An example of Slurm Array job script

```
#SBATCH --array=1-20
```

Or you can specify an array job at submission time, without modifing your submission script:

```
sbatch --array=1-20 job.script
```

In Slurm, the job array is implemented as a group of single jobs. E.g. if you submit an array job with #SBATCH –array=1-4. When the starting job is ID=1000, the ids of all jobs are: 1000, 1001, 1002, 1003.

Note: There is a limit of 1000 jobs per array. Slurm also has a bug where it will not allow array id's above this limit, this can be worked around with a prefix in the script (e.g. for "1001-1020" use –array=01-20 and reference variables with a prefix 10$SLURM_ARRAY_TASK_ID)

A maximum number of simultaneously running tasks from the job array may be specified using a "%" separator. For example "–array=0-15%4" will limit the number of simultaneously running tasks from this job array to 4.

12.12 QoS (Quality of Service)

We have implemented Quality of Service (QoS) starting 6th of June 2018.

The QoS can be added to each job that is submitted to Slurm. The quality of service associated with a job will affect the job in three ways:

1. Job Scheduling Priority
2. Job Preemption
3. Job Limits

12.13 How to run jobs with QoS

A table to show the differences between QoS:

Queue	Description	Max Wall-time	Max GPU per user	Max CPU per user	Prior-ity	Preemp-tion
nor-mal	Default QoS for every job	7 days	10	280	50	No
rtq	QoS for interactive job	48 hours	4	36	200	No
irq	QoS for interruptable job	7 days	No limit	No limit	100	Yes
shortq	QoS for job with short walltime	30 mins	10	280	100	No

12.13.1 Explanation

These QoS are applied to the partition `comp`.

`--qos=normal`

This is the QoS for all the jobs that do not specify a QoS. Jobs that run here won't be interrupted.

`--qos=rtq`

This is intended to be used by jobs that have an instrument or a real-time scenario and therefore can't be interrupted and must be available on demand. You can only use a few CPUs and GPUs, but jobs will start as soon as possible (before `normal` jobs).

`qos=irq` This is intended to be used by jobs that are interruptible. To use the `irq` you have to be prepared to either restart from scratch (if the job was short anyway) or restart from a checkpoint. Jobs will start very quickly and use all the availble resources but may be stopped at short notice to allow `shortq` or `rtq` jobs to run.

The mechanism to checkpoint depends on the software that you are running.

Please contact us if you have any questions with regards to job checkpointing.

`--qos=shortq` This is intended to be used by short and uninterruptible jobs. These jobs will run before `normal` but the walltime is limited.

An example of Slurm GPU job script

```
#!/bin/bash
#SBATCH --job-name=MyJob
#SBATCH --account=<my_account>
#SBATCH --qos=<irq,shortq,rtq>
#SBATCH --gres=gpu:<K80,P100,V100>:1
#SBATCH --ntasks=2

<GPU processing program>
```

An example Slurm job script

```
#!/bin/bash
#SBATCH --job-name=MyJob
#SBATCH --account=<my_account>
#SBATCH --qos=<irq,shortq,rtq>
#SBATCH --ntasks=2
```

(continues on next page)

(continued from previous page)

```
openmpi/1.10.7-mlx
mpirun <program>
```

Warning: This command is not yet implemented on M3. Please check the message of the day (MOTD) displayed at login to see which system commands are currently implemented on M3.

12.14 Checking job status

There are two methods to check your job status.

12.14.1 Method 1: `show_job`

We provide a `show_job` script. This script groups information, filters, sorts, and provides statistics to provide a clean, tidy, and user-friendly output.

```
$ show_job

                        ************************************************
                        *               MY JOB SUMMARY                 *
                        *               Cluster:    m3                 *
                        ************************************************
                          User Name                          Massive User
                          User ID                                 masusr
                        ------------------------------------------------
                          Num of Submitted Jobs                        0
                          Num of Running Jobs                          0
                          Num of Pending Jobs                          0
                          Num of CPU Cores                             0
                        ------------------------------------------------


                        ************************************************
                        *               Job Details on m3                *
                        ************************************************

JOBID           JOB NAME     Project          QOS       STATE    RUNNING       TOTAL
↪NODE                    DETAILS
                                                                    TIME        TIME
-------------------------------------------------------------------------------------------
↪----------------------------
```

Hint: To check the status of a single job use `show_job [JOBID]`.

12.14.2 Method 2: Slurm commands

To display all of your running/pending jobs use `squeue -u `whoami``.

Hint: `whoami` returns your M3 username, and is a handy shortcut.

```
$ squeue -u `whoami`
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
```

If you want to view the status of a single job

```
$ scontrol show job [JOBID]
```

12.15 Project Allocation

Resources on M3 are allocated by the SLURM job scheduler. Unlike M1 & M2, on M3 we use the SLURM multifactor plugin to assign priorities to jobs. This means that the scheduler sets priority based upon job age, queue, partition, job size, QoS and fair-share.

Slurm's fair-share factor is a scoring system that reflects the shares of a computing resource that a user has been allocated and the number of computing resources the user's jobs have consumed. The fair-share component to a job's priority influences the order in which a user's queued jobs are scheduled to run.

The MASSIVE team is working on a mechanism to represent this score in a more meaningful way.

Please refer to SLURM documentation for more information: https://slurm.schedmd.com/priority_multifactor.html#fairshare

12.15.1 Project Space

To check your quota on /projects

```
lfs quota -h -g "project_id" /projects
```

and to check your quota on /scratch

```
lfs quota -h -g "project_id" /scratch
```

Please also visit the link below for more information about our filesystem:

http://docs.massive.org.au/M3/file-systems-on-M3.html

12.15.2 Questions about allocation

If you have any enquiries with regards to your project resources and space allocation, please do not hesitate to contact us on help@massive.org.au

CHAPTER 13

Lustre File System Quickstart Guide

13.1 About Lustre

Attention: This content is still under development. Please check back again shortly.

Lustre is a parallel filesystem that is highly scalable and it is a shared filesystem for your `/projects` and `/scratch` directory.

Lustre servers consist of two major components: **Metadata** and **Object Storage**.

One or more metadata servers (MDS) nodes that has one or more metadata target (MDT) devices are assigned to store namespace metadata, such as filenames, directories, access permissions, and file layout. One or more object storage server (OSS) nodes store file data on one or more object storage target (OST) devices. Lustre software presents the OSTs as a single unified filesystem.

On M3, `/projects` is made up of 10 OSTs and each OST has a space of 29TB and this is the space for you to store your primary data.

`/scratch` is presented for you to store reproducible data, this space is often bigger than your `/project` space and is not backed-up.

One of the benefits of using Lustre is file striping, which means files can be divided into chunks that are written or read simultaneously across a set of OSTs within the filesystem. The chunks are distributed among the OSTs using a method that ensures load balancing.

For both `/projects` and `/scratch` directories, we have set the default stripe count for all directories to 1.

To find out the stripe info on your file, you may use **lfs getstripe** command, e.g.:

```
lfs getstripe /projects/pMOSP/gins/dd-baseline01
```

If you have a directory that consists of files that are larger than 100GB, please consider striping your files across two or more OSTs; to assign an appropriate stripe count, use the command:

```
lfs setstripe -c 2 /projects/pMOSP/gins/
```

Only files created **after** running this command will have the new stripe settings.

For more information about how to do file striping on Lustre, please contact help@massive.org.au, detailing the directory to be striped and suggestions will be provided to you.

In order to use Lustre in a more efficient way, avoid having a lot of small files inside a directory. A better practice is to split a large number of files (in the thousands or more) into multiple subdirectories to minimize the contention to the metadata servers.

We have enforced quota on both `/projects` and `/scratch` directories. To find out your project usage:

```
lfs quota -g {project_id} /projects

lfs quota -g {project_id} /scratch
```

Attention: This documentation is under active development, meaning that it can change over time as we refine it. Please email help@massive.org.au if you require assistance, or have suggestions to improve this documentation.

CHAPTER 14

DRAFT Instructions for MX2 Eiger users

This document supersedes earlier instructions on how to download your MX2 data without assistance from the MASSIVE team. Earlier instructions are preserved here MX Eiger V1 instructions.

The MASSIVE team can now download the data on your behalf and store the data without counting it towards your disk quota. This document also updates and simplifies the procedure for setting up for reprocessing data. M3 is available to users associated with MASSIVE partners for transferring and processing Eiger data. Please follow these instructions to get started.

14.1 The Big Picture

With recent detector upgrades, the MX beamline has started to produce a relatively high data rate. As archival storage is considerably cheaper than using high speed file systems for data storage, the MASSIVE team aims to help high data rate producers in two primary ways:

1. By making an archival copy of all their data
2. While continuing to make it available on a high speed file system for processing

The following process is designed to limit the length of time that data is kept available on the high speed file system by moving it to archival storage, with the option of retrieving it again latter.

1. The MASSIVE team creates one project per each data producer. In the case of MX data at Monash, this means one project for each Monash CAP.
2. The MASSIVE team creates an additional project for each research subgroup to do their work. There may be one or more research projects per CAP.
3. Each CAP leader authorises the MASSIVE team to download data from the Australian Synchrotron on their behalf (see the section *Authorising MASSIVE to download your data*).
4. The MASSIVE team downloads a copy of each experiment to the M3 high performance file system, and makes this data readable to the appropriate users.
5. The MASSIVE team makes additional copies of each experiment to archival storage.

6. After some time (typically around three months) the MASSIVE team removes the copy of data on M3 (preserving the copies on archival storage).
7. Each researcher is able to "symlink" large data files (avoiding taking up additional disk space) and copy smaller input files (allowing them to be changed) and perform re-processing.

14.2 Authorising MASSIVE to download your data (CAP Leaders only)

In order to use this service each CAP leader needs to authorise the M3 service to download data collected by their CAP for each EA, using the following steps:

1. Open the Australian Synchrotron Portal (https://portal.synchrotron.org.au)
2. Open the Experiment Application (EA) for your CAP
3. Select Add user and search for the user "Monash Datasynchroniser" with the email address "help@massive.org.au".

Note: This user needs to be present in the EA for each beam time, but since common practice is to copy the EA from the previous beam time, you should only need to do this once. This user does not count for the maximum of 15 users per experiment.

14.3 Creating research projects (research lab leaders only)

M3 use is organised into projects that are lead by a project lead. The project lead is usually a research lab leader, although we can be flexible around this, and have some projects which consist of a single person rather than a lab; please contact us to discuss cases like this. The project lead is responsible for managing users within the project and communicating with the MASSIVE team about allocations and project reporting. We recommend that M3 projects are lead by research laboratory leads.

Note: Note that M3 projects are not the same as CAP projects; research laboratory leads must apply for a separate M3 project to process MX data. If your group already has an M3 project, proceed directly to the next section.

To request an M3 project:

1. Fill out the project request form at https://goo.gl/forms/YtZoTU98ZU9GrlUI2
2. The MASSIVE team will email your project team members with instructions on how to create an account and join your project

Creating projects and user accounts may take up to two business days. If you do not receive a confirmation email within two business days, contact the MASSIVE help desk at help@massive.org.au.

14.4 Requesting access to M3

The M3 identity management system is currently undergoing an overhaul designed to make it easier for project leaders to authorise collaborators to share access to their data and resource allocations. In the meantime, follow these steps to request access to M3:

1. Go to the HPC identity managment portal
2. Enter your University and your university username and password
3. Select a user name
4. Email help@massive.org.au providing:
   (a) your user name
   (b) which CAP you are a part of
   (c) which research subgroup you are a part of (if you are not a leader)

Once the MASSIVE team has finished setting up your account, you will receive an email asking you to go back to the HPC identity managment portal and set a password to log in to M3.

14.5 Getting started on M3

Go to http://docs.massive.org.au/ for information to get you started with the M3 system. Alternatively, contact help@massive.org.au to schedule an on-boarding session for your lab group.

14.6 Connecting to M3

Connect to M3 and start a shell/terminal session. This can be done in one of three ways. Please see *Connecting to M3* for details on each of these options.

- SSH into m3.massive.org.au as per instructions at *Connecting to M3 via ssh*
- Log in to a desktop session using Strudel as per instructions at *Connecting to M3 via the MASSIVE desktop*
- Log in to a desktop session through the web using https://desktop.massive.org.au

Users who have logged in using a desktop session will need to launch a terminal, by double clicking on the terminal icon.

14.7 Accessing your MX data

You should find your data located in: `/scratch/<CAPprojectcode>/<epn>`

`<CAPprojectcode>` is the project code of your CAP. If you do not know your project code, you can type `id` at the command line and it will list all projects you are a member of. `<epn>` is the Australian Synchrotron EPN number.

Note: This copy of the data is read-only and cannot be changed.

14.8 Reprocessing your MX data

1. Use the command `module load mxtools` to load a set of tools to assist in reprocessing MX data
2. Create an appropriate directory to store reprocessed output
3. Use the command `xds_reprocess <autoprocessing directory> <reprocessing directory>`

Note: The auto-processing directory contains files generated by the Australian Synchrotron during auto-processing. You will find it somewhere under `/scratch/<CAPprojectcode>/<epn>/home/<ASUsername>/`. The reprocessing directory can be anywhere you like. We recommend putting it somewhere under `/projects/<ResearchProjectcode>/<M3Username>`.

Use the commands:

```
module load xds/monash
xds_par
```

This will run the command `xds_par`. Wait for the process to finish. You will be returned to your terminal prompt. The output of the command is displayed on the terminal screen and is also stored in `xds_stdout` file in the same folder where `xds_par` was run.

14.9 Recommended Settings

You might also wish to include these values in your `XDS.INP` file, as recommended by the MX Beamline scientists.

```
JOB=ALL
FRIEDEL'S_LAW=FALSE
```

14.10 Known Issues

14.10.1 Cannot open or read filename.tmp error

The following error may appear when you are attempting to process data:

```
!!! ERROR !!! CANNOT OPEN OR READ FILE LP_01.tmp

Presumably, the independent jobs supposed to have been started
in the COLSPOT step have not run properly. This could be due
to incomplete installation of XDS where some of the executables
of the package are not included in the search path.

!!! ERROR !!! CANNOT OPEN OR READ "mcolspot.tmp"
```

To fix this error, run the `module purge` command, followed by reloading the `xds` module:

```
module purge
module load xds/monash
xds_par
```

14.11 Need help?

If you need help or have any queries, contact help@massive.org.au

Attention: This documentation is under active development, meaning that it can change over time as we refine it. Please email help@massive.org.au if you require assistance, or have suggestions to improve this documentation.

CHAPTER 15

Deep Learning on M3

15.1 Software

There are a number of deep learning packages available on M3.

15.1.1 Caffe

To use Caffe on M3:

```
nvidia-modprobe -u
module load caffe
your-caffe-script-here
```

15.1.2 TensorFlow

To use TensorFlow on M3:

```
# Loading module
module load tensorflow/1.3.0-python2.7.12-gcc5

# Unloading module
module unload tensorflow/1.3.0-python2.7.12-gcc5
```

15.2 Reference datasets

15.2.1 ImageNet

ImageNet is a collection of approximately 14 million annotated images often used to benchmark and test image classification and object detection models. Each image has been categorised according to a WordNet ID, and additional

annotations, such as bounding boxes, can be obtained via ImageNet APIs. The dataset is described in full at http://www.image-net.org.

Users can access the Fall 2011 version of the ImageNet database by registering their acceptance of the terms of access.

15.3 Quick guide for checkpointing

15.3.1 Why checkpointing?

Checkpoints in Machine/Deep Learning experiments prevent you from losing your experiments due to maximum walltime reached, blackout, OS faults or other types of bad errors. Sometimes you want just to resume a particular state of the training for new experiments or try different things.

Pytorch:

First of all define a save_checkpoint function which handles all the instructions about the number of checkpoints to keep and the serialization on file:

```
def save_checkpoint(state, condition, filename='/output/checkpoint.pth.tar'):
    """Save checkpoint if the condition achieved"""
    if condition:
        torch.save(state, filename)  # save checkpoint
    else:
        print ("=> Validation condition not meet")
```

Then, inside the training (usually a for loop with the number of epochs), we define the checkpoint frequency (at the end of every epoch) and the information (epochs, model weights and best accuracy achieved) we want to save:

```
# Training the Model
for epoch in range(num_epochs):
    train(...)  # Train
    acc = eval(...)  # Evaluate after every epoch

    # Some stuff with acc(accuracy)
    ...
    # Get bool not ByteTensor
    is_best = bool(acc.numpy() > best_accuracy.numpy())
    # Get greater Tensor to keep track best acc
    best_accuracy = torch.FloatTensor(max(acc.numpy(), best_accuracy.numpy()))
    # Save checkpoint if is a new best
    save_checkpoint({
        'epoch': start_epoch + epoch + 1,
        'state_dict': model.state_dict(),
        'best_accuracy': best_accuracy
    }, is_best)
```

To resume a checkpoint, before the training we have to load the weights and the meta information we need:

```
checkpoint = torch.load(resume_weights)
start_epoch = checkpoint['epoch']
best_accuracy = checkpoint['best_accuracy']
model.load_state_dict(checkpoint['state_dict'])
print("=> loaded checkpoint '{}' (trained for {} epochs)".format(resume_weights,␣
↪checkpoint['epoch']))
```

Keras

Keras provides a set of functions called callback: you can think of it as events that will triggered at certain training state. The callback we need for checkpointing is the ModelCheckpoint which provides all the features we need according to the checkpoint strategy adopted.

```
from keras.callbacks import ModelCheckpoint
# Checkpoint In the /output folder
filepath = "/output/mnist-cnn-best.hdf5"

# Keep only a single checkpoint, the best over test accuracy.
checkpoint = ModelCheckpoint(filepath,
                             monitor='val_acc',
                             verbose=1,
                             save_best_only=True,
                             mode='max')
# Train
model.fit(x_train, y_train,
              batch_size=batch_size,
              epochs=epochs,
              verbose=1,
              validation_data=(x_test, y_test),
              callbacks=[checkpoint])  # <- Apply our checkpoint strategy
```

Keras models have the load_weights() method which load the weights from a hdf5 file. To load the model's weight you have to add this line just after the model definition:

```
... # Model Definition
model.load_weights(resume_weights)
```

Tensorflow

The tf.train.Saver class provides methods to save and restore models. The tf.saved_model.simple_savefunction is an easy way to build a saved model suitable for serving. Estimators automatically save and restore variables in the model_dir.

```
# Create some variables.
v1 = tf.get_variable("v1", shape=[3], initializer = tf.zeros_initializer)
v2 = tf.get_variable("v2", shape=[5], initializer = tf.zeros_initializer)

inc_v1 = v1.assign(v1+1)
dec_v2 = v2.assign(v2-1)

# Add an op to initialize the variables.
init_op = tf.global_variables_initializer()

# Add ops to save and restore all the variables.
saver = tf.train.Saver()

# Later, launch the model, initialize the variables, do some work, and save the
# variables to disk.
with tf.Session() as sess:
  sess.run(init_op)
  # Do some work with the model.
  inc_v1.op.run()
  dec_v2.op.run()
```

(continues on next page)

(continued from previous page)

```
  # Save the variables to disk.
  save_path = saver.save(sess, "/tmp/model.ckpt")
  print("Model saved in path: %s" % save_path)
```

After a model has been trained, it can be restored using tf.train.Saver() which restores Variables from a given checkpoint. For many cases, tf.train.Saver() provides a simple mechanism to restore all or just a few variables.

```
# Create some variables.
v1 = tf.Variable(..., name="v1")
v2 = tf.Variable(..., name="v2")
...
# Add ops to restore all the variables.
restorer = tf.train.Saver()

# Add ops to restore some variables.
restorer = tf.train.Saver([v1, v2])

# Later, launch the model, use the saver to restore variables from disk, and
# do some work with the model.
with tf.Session() as sess:
  # Restore variables from disk.
  restorer.restore(sess, "/tmp/model.ckpt")
  print("Model restored.")
  # Do some work with the model
  ...
```

Attention: This documentation is under active development, meaning that it can change over time as we refine it. Please email help@massive.org.au if you require assistance, or have suggestions to improve this documentation.

CHAPTER 16

Neuroimaging

This section covers workflows to assist the Neuroimaging Community.

16.1 Human Connectome Project Data Set

MASSIVE hosts a copy of the Human Conectome Project (HCP) Data, 900 and 1200 subject series. This data is restricted to M3 users that have their email registered with HCP (i.e. an account) **AND** have accepted the HCP terms and conditions associated with the datasets.

The data is located at:

```
/scratch/hcp
/scratch/hcp1200
```

The following outlines the process of getting access to this data:

16.1.1 Human Connectome Project Steps for Access

Create an Account

1. Follow the "Get Data" link to: http://www.humanconnectome.org/data/
2. Select "Log in to ConnectomeDB"
3. Select "Register"
4. Create an account and make sure that your email address matches the email associated with you MASSIVE account
5. You will recieve an email to validate your account

Accept Terms and Conditions

1. Login to https://db.humanconnectome.org/

#. Accept terms and conditions using the "Data Use Terms Required" button on all 3 datasets (WU-Minn, WU-Minn HCP Lifespan Pilot and MGH HCP Adult Diffusion)

16.1.2 MASSIVE Steps for Access

Request Access to HCPdata

If you have completed the Human Connectome Project steps above you can request access with this link: HCPData.

We will verify your MASSIVE email against the HCP site and grant access.

Accessing the Data

The data is available on M3 via */scratch/hcp* and */scratch/hcp1200* You can also create a symbolic link to this in your home directory using

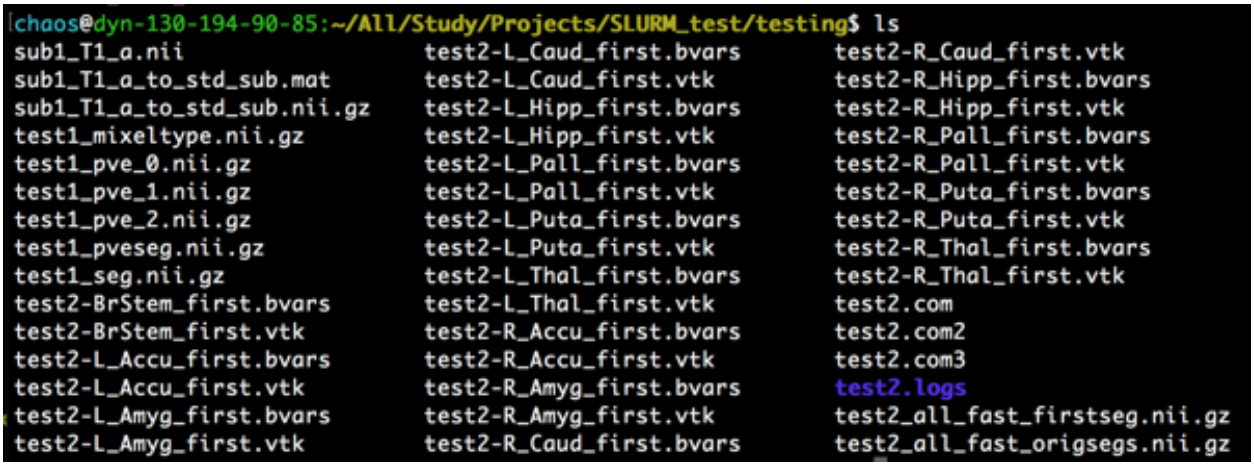
```
ln -s /scratch/hcp ~/hcp
ln -s /scratch/hcp1200 ~/hcp1200
```

Note: The Connectome Workbench is available via the MASSIVE Desktop. Any updates to this or other software requirements can be directed to help@massive.org.au

16.2 Using SLURM to submit a simple FSL job

Example data at /home/chaos/kg98/chaos/SLURM/first/example

- FIRST is a simple tool in FSL library to segment sub-cortical regions of a structural image. https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FIRST/UserGuide
- It usually takes 20 mins to process each image.
- The typical output are a group of image masks of all sub-cortical regions, e.g., hippocampus, amygdala...



Data and scripts

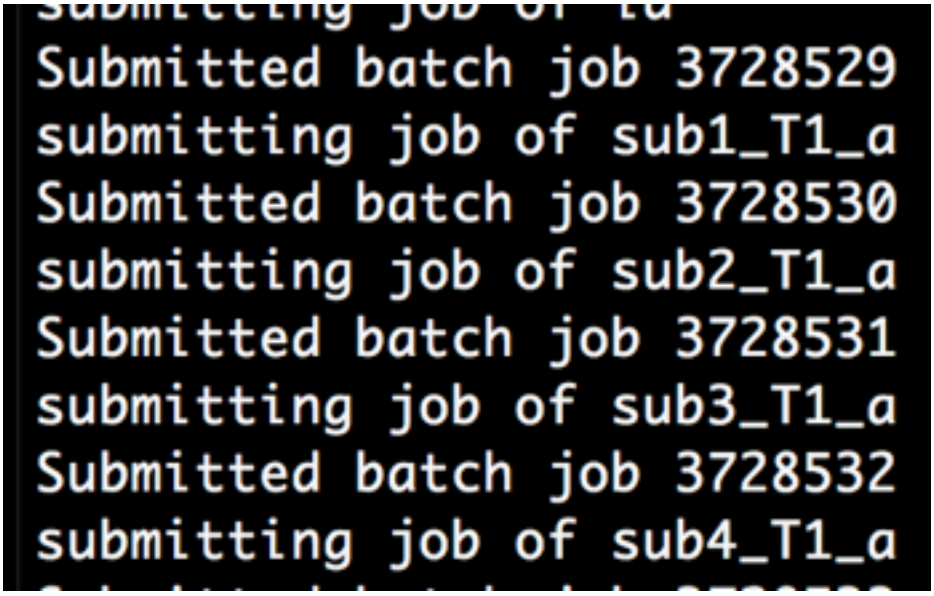
- 6 T1-weighted structural images

```
sub1_T1_a.nii  sub3_T1_a.nii  sub5_T1_a.nii
sub2_T1_a.nii  sub4_T1_a.nii  sub6_T1_a.nii
```

- **Three files in the script directory**
  - Id.txt = base name list of these imaging files (check basename function for further help)
  - First_task.txt = file with the resource and environment detail of one fsl-first job **NEED TO change the DIR for the correct path of input data**
  - Submit_template= loop to submit all six jobs.

To run the job, simply do

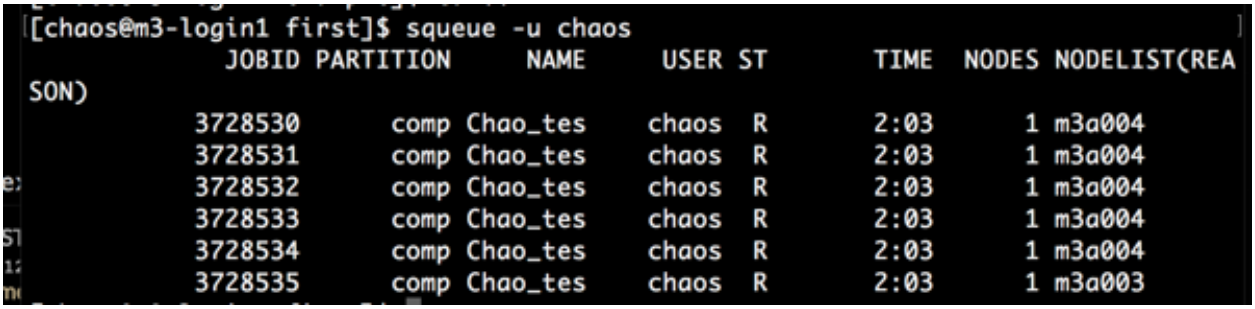
```
$bash submit_template
```



Try to use

```
squeue -u (your_masssive_accName)
```

to check the job.



Output log files:

- Log files should be appear in the script folder, recording all the logs of fsl-first command

```
chaos@m3-login1 scripts]$ ls
irst_task.txt     slurm-3728530.out  slurm-3728533.out  submit_template.txt
d.txt             slurm-3728531.out  slurm-3728534.out
lurm-3728529.out  slurm-3728532.out  slurm-3728535.out
chaos@m3-login1 scripts]$ more slurm-3728530.out
TRUCTURES: L_Accu L_Amyg L_Caud L_Hipp L_Pall L_Puta L_Thal R_Accu R_Amyg R_Cau
 R_Hipp R_Pall R_Puta R_Thal BrStem
usr/local/fsl/5.0.9/fsl/bin/first_flirt /projects/kg98/chaos/SLURM/first/exampl
/sub1_T1_a /projects/kg98/chaos/SLURM/first/example/sub1_T1_a_to_std_sub
usr/local/fsl/5.0.9/fsl/bin/run_first -i /projects/kg98/chaos/SLURM/first/examp
e/sub1_T1_a -t /projects/kg98/chaos/SLURM/first/example/sub1_T1_a_to_std_sub.ma
 -n 50 -o /projects/kg98/chaos/SLURM/first/example/sub1_T1_a_first-L_Accu_first
```

Output data:

- Output data should start to pop up in the data folder

```
[chaos@m3-login1 example]$ ls
scripts                                sub4_T1_a_first-L_Accu_first.vtk
sub1_T1_a_first.com                    sub4_T1_a_first.logs
sub1_T1_a_first-L_Accu_corr.nii.gz     sub4_T1_a.nii
sub1_T1_a_first-L_Accu_first.bvars     sub4_T1_a_to_std_sub.mat
sub1_T1_a_first-L_Accu_first.nii.gz    sub4_T1_a_to_std_sub.nii.gz
sub1_T1_a_first-L_Accu_first.vtk       sub5_T1_a_first.com
sub1_T1_a_first.logs                   sub5_T1_a_first-L_Accu_corr.nii.gz
sub1_T1_a.nii                          sub5_T1_a_first-L_Accu_first.bvars
sub1_T1_a_to_std_sub.mat               sub5_T1_a_first-L_Accu_first.nii.gz
sub1_T1_a_to_std_sub.nii.gz            sub5_T1_a_first-L_Accu_first.vtk
sub2_T1_a_first.com                    sub5_T1_a_first.logs
sub2_T1_a_first-L_Accu_corr.nii.gz     sub5_T1_a.nii
sub2_T1_a_first-L_Accu_first.bvars     sub5_T1_a_to_std_sub.mat
sub2_T1_a_first-L_Accu_first.nii.gz    sub5_T1_a_to_std_sub.nii.gz
sub2_T1_a_first-L_Accu_first.vtk       sub6_T1_a_first.com
sub2_T1_a_first.logs                   sub6_T1_a_first-L_Accu_corr.nii.gz
sub2_T1_a.nii                          sub6_T1_a_first-L_Accu_first.bvars
sub2_T1_a_to_std_sub.mat               sub6_T1_a_first-L_Accu_first.nii.gz
sub2_T1_a_to_std_sub.nii.gz            sub6_T1_a_first-L_Accu_first.vtk
sub3_T1_a_first.com                    sub6_T1_a_first-L_Amyg_corr.nii.gz
sub3_T1_a_first.logs                   sub6_T1_a_first-L_Amyg_first.bvars
sub3_T1_a.nii                          sub6_T1_a_first-L_Amyg_first.nii.gz
sub3_T1_a_to_std_sub.mat               sub6_T1_a_first-L_Amyg_first.vtk
sub3_T1_a_to_std_sub.nii.gz            sub6_T1_a_first.logs
sub4_T1_a_first.com                    sub6_T1_a.nii
sub4_T1_a_first-L_Accu_corr.nii.gz     sub6_T1_a_to_std_sub.mat
sub4_T1_a_first-L_Accu_first.bvars     sub6_T1_a_to_std_sub.nii.gz
sub4_T1_a_first-L_Accu_first.nii.gz
```

Attention: This documentation is under active development, meaning that it can change over time as we refine it. Please email help@massive.org.au if you require assistance, or have suggestions to improve this documentation.

CHAPTER 17

Cryo EM

MASSIVE supports the cryo-EM community with storage and compute services. On these pages we will provide suggestions for workflows and software settings that we have tested to be optimal for a range of datasets. If you have a dataset that you think is taking longer than normal to process or you notice performance issues please let us know and our help team will work with you to resolve the issues.

The three major cryo-EM data processing packages that we have installed for you are:

1. Relion
2. Simple
3. Cryosparc

17.1 Relion

The Relion software is capable of running on single desktops, right through to multiple nodes on a compute cluster. Generally, the Relion workflow consists of numerous computationally light (or interactive) steps, and a few steps which demand significant resources. You will probably wish to use the Relion graphical interface to invoke the light (or interactive) steps directly. For the more computationally demanding steps, it is generally best to submit a job to the M3 queuing system.

17.1.1 The Graphical Interface

To utilise the Relion graphical interface, you have two options. The first option is to create a MASSIVE desktop instance, either using Strudel or Strudel Web. The second option is to utilise X11 forwarding.

For MASSIVE desktop usage, note that currently the **Standard Desktop** does not currently provision a CUDA compatible GPU (restricting Relion to CPU processing). You will need to launch a **Large Desktop** if you wish to utilise a GPU. Note also that only Relion 1.4 is available via the desktop menu. To access more recent versions, you will need to open a terminal, load the required Relion module, and then run the `relion` executable.

X11 forwarding will utilise your local X11 server to render the Relion instance running remotely on M3. The instance itself may be launched from the login node, although computationally demanding workflow steps must **not** be executed

directly. Login nodes are a shared resource reserved for job orchestration, and as such running heavy calculations there may impede other users (and will not be particularly fast!). If you wish to launch heavy jobs directly from the interface, you may launch the GUI directly from a compute node instead. This will allow you to specify exactly the resources you require, but as with the **Large Desktop** resources may not be immediately available. To access a compute node, launch an interactive command line job. Once your job has started, load the required Relion module and run the `relion` executable.

Tip: For particle picking, Relion uses the middle-click to deselect a particle. If you are using OS X, you might need to use a tool like MiddleClick to enable middle click emulation.

17.1.2 Using the Queue

While running computationally intensive steps directly from the interface is possible (as per instructions above), it is generally most efficient to execute these steps using the job queuing system. Processing such as 2d and 3d classification require hours of CPU/GPU time, and while using a sufficiently provisioned interactive (desktop or command line) instance may be a reasonable option, often these will not be available for immediate launch. The queue is also the most efficient choice where you need to submit numerous jobs for processing.

This is a basic queue submission script for Relion:

```
#!/bin/bash
#SBATCH --job-name=MyJob
#SBATCH --time=01:00:00
#SBATCH --ntasks=16
#SBATCH --cpus-per-tasks=1

module load relion
mpirun YOUR_RELION_COMMAND
```

Here the required Relion command (YOUR_RELION_COMMAND) will depend upon the job within your Relion workflow which you wish to execute. You can determine this command via the `Print command` button within the Relion GUI (it will print the command to your terminal, not the GUI).

The other path to submitted M3 queue jobs is via the Relion interface itself. Under the `Running` tab, you will need to set the following options:

Number of MPI procs	MPI rank count.
Number of threads	Threads per rank count.
Submit to queue	Yes.
Queue name	Ignored.
Queue submit command	sbatch
Job Name (no spaces!)	A name for your job.
Job Time	Maximum wall time for your job.
Standard submission script	Default script to use.
Minimum dedicated cores per node	Ignored.

The resultant queue script forms part of the preamble printed to your job standard output file. You may find it useful to copy this for manual job submission.

17.1.3 Motion Correction

Relion interfaces to MotionCor2 or Unblur for beam induced motion correction. If you are using an interactive instance running on a compute node, executing motion directly from the Relion GUI may be preferred option.

Note that MotionCor2 can utilise GPU accelerators where your compute node is appropriately provisioned. For successful GPU usage, your job execution must be configured very specifically. Firstly, the `Number of MPI procs` setting (under the **Running** tab) must equal the number of GPU accelerator cards. Secondly, the `Which GPUs to use` setting (under the **Motioncor2** tab) must provide a colon separated list of integer GPU identifiers. For example, using Motioncor2 on the **m3h** partition which has 2 GPUs per node, you would therefore use 2 MPI processes, and you would set the GPUs to use to "0:1". An equivalent queue submission script would be:

```
#!/bin/bash
#SBATCH --job-name=motion_correction
#SBATCH --time=01:00:00
#SBATCH --ntasks=2
#SBATCH --partition=m3h
#SBATCH --gres=gpu:2

module load relion
module load motioncor2

mpirun `which relion_run_motioncorr_mpi` --i ./Import/job001/movies.star --o␣
↪MotionCorr/job033/ --save_movies  --first_frame_sum 1 --last_frame_sum 16 --use_
↪motioncor2 --bin_factor 1 --motioncor2_exe /usr/local/motioncor2/2.1/bin/motioncor2␣
↪--bfactor 150 --angpix 3.54 --patch_x 5 --patch_y 5 --gpu "0:1" --dose_weighting --
↪voltage 300 --dose_per_frame 1 --preexposure 0
```

Note that for this script, we've added the partition specification `--partition=m3h` to ensure that the job is launched on a GPU enabled node, and we've requested the 2 GPUs available on these nodes `--gres=gpu:2`. Also, note that it may be necessary to explicitly specify the MotionCor2 executable in the `MOTIONCOR2 executable` dialog under the `Motioncor2` tab. You can find the executable by running *which motioncor2* from the command line (after loading the required module).

17.1.4 2d/3d Classification & Refinement

These jobs are likely to be the most computationally demanding in your Relion workflow. Their expense will scale with the number of particles, and the number of classes you request. Although CPU only operation is possible, GPUs usage will generally yield an order of magnitude performance improvement. Timeframes for these jobs are often measures in hours, so you will most likely need to use the queuing system for efficient operation.

Relion will utilise all provisioned GPUs through the use of MPI processes and/or threads. The optimal choice of these configurations will somewhat depend on the specifics of your jobs. Generally speaking, fastest operation is achieved by utilising 2-4 MPI *worker* ranks **per** GPU, with each having 1-2 threads.

Memory usage, however, scales strongly with MPI rank count, and may be prohibitively expensive (depending on your particle box size). It may often be necessary to make a tradeoff between memory usage and performance. Also note that memory usage increases with iteration count as Relion only performs lower resolution reconstructions during early iterations. Relion suggests the following GPU maximum memory requirements (in gigabytes) for **each** MPI worker on a GPU:

$$Mem = \frac{2.5 * 1.05 * 4 * (2 * boxsize)^3}{1024 * 1024 * 1024}$$

For large simulations, it may be useful to run early iterations using more performant configurations, and then continue the job using settings with a smaller memory footprint. To continue a job, use the `--continue` flag to indicate

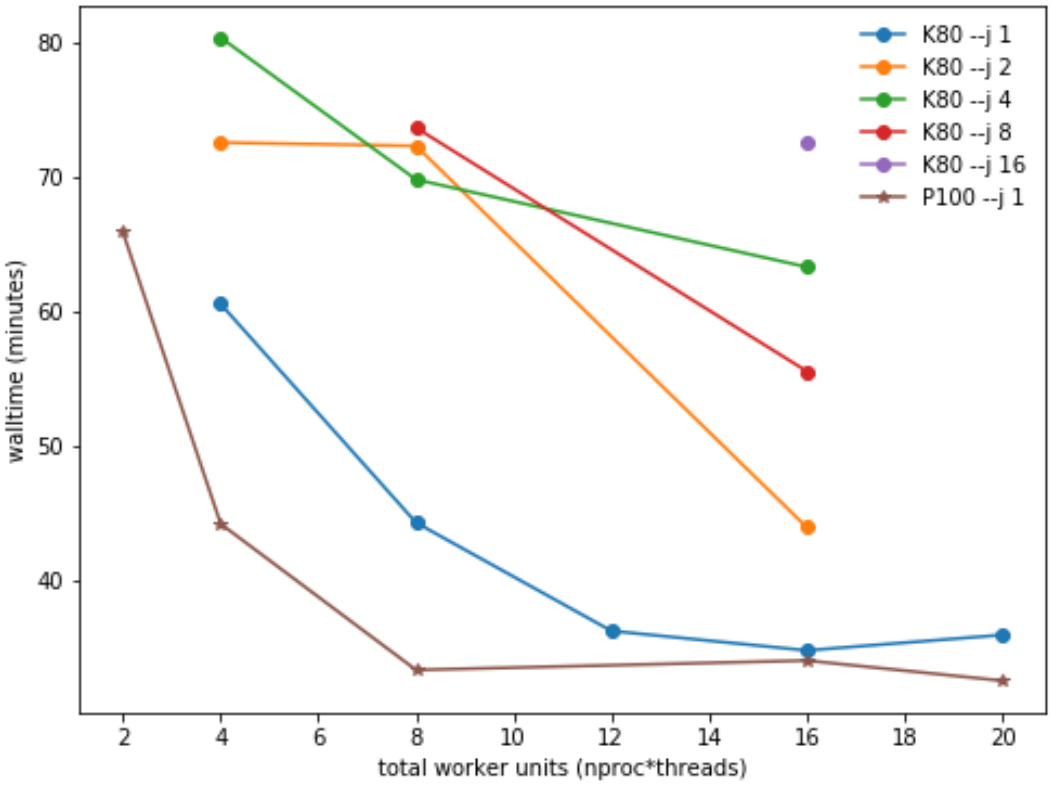


Fig. 1: Walltime vs total workers. 3D Classification benchmark, 5 iteration. M3C partition: Relion 2.04, 4 K80 GPUs. M3H partition: Relion 2.1b1, 2 P100 GPUs.

the required iteration STAR file to continue from. For large enough particle images, GPU usage may not be possible during later iterations, in which case it will be necessary to continue in CPU only mode.

Relion uses a master/worker pattern for MPI job distribution. The total tasks required will therefore equal the number of workers plus one. So for 16 workers, 1 threads per worker, and 2 GPUs, your submission script will be:

```
#!/bin/bash
#SBATCH --job-name=relion_refine
#SBATCH --time=01:00:00
#SBATCH --ntasks=17
#SBATCH --cpus-per-task=1
#SBATCH --partition=m3h
#SBATCH --gres=gpu:2

module load relion

mpirun `which relion_refine_mpi` --i Particles/shiny_2sets.star --ref emd_2660.
↪map:mrc --firstiter_cc --ini_high 60 --ctf --ctf_corrected_ref --iter 5 --tau2_
↪fudge 4 --particle_diameter 360 --K 6 --flatten_solvent --zero_mask --oversampling␣
↪1 --healpix_order 2 --offset_range 5 --offset_step 2 --sym C1 --norm --scale --
↪random_seed 0 --o class3d --gpu --pool 10 --dont_combine_weights_via_disc --j
↪$SLURM_CPUS_PER_TASK
```

Note that we use the SLURM environment variable `SLURM_CPUS_PER_TASK` to set the number of threads within the Relion command so that it automatically matches what you have requested for your SLURM script.

For certain jobs Relion scales well across multiple nodes, with almost linear speedup with GPU count. If you wish to use more than one node on M3 for a single job, there are some further settings you need to invoke to ensure that Relion is appropriately cast across the provisioned nodes. In particular, we wish to ensure that the distribution of workers per GPU is identical for all GPUs. For example, if we wish to have 4 MPI ranks **per** GPU, we need to set a total of 9 MPI ranks to the first node (1 master rank, plus 4 works per GPU), and then each node thereafter should only have 8 MPI workers (4 for each GPU again). So to utilise 3 nodes we would require a total of 25 MPI ranks, and the necessary script would be:

```
#!/bin/bash
#SBATCH --job-name=relion_refine_multinode
#SBATCH --time=01:00:00
#SBATCH --ntasks=25
#SBATCH --ntasks-per-node=9
#SBATCH --cpus-per-task=1
#SBATCH --partition=m3h
#SBATCH --gres=gpu:2

module load relion

mpirun `which relion_refine_mpi` ...
```

Note the addition of the flag `ntasks-per-node` which limits the number of tasks distributed across each node and ensures the required distribution pattern. Submitting multiple node classification/refinement queue jobs directly from the Relion interface is not currently supported, though you can easily generate a single node job and modify its script as instructed above.

Relion provides further options for performance optimisation. Our testing across the standard 2d & 3d Relion benchmarks suggest the following settings will be a good starting point for running on M3:

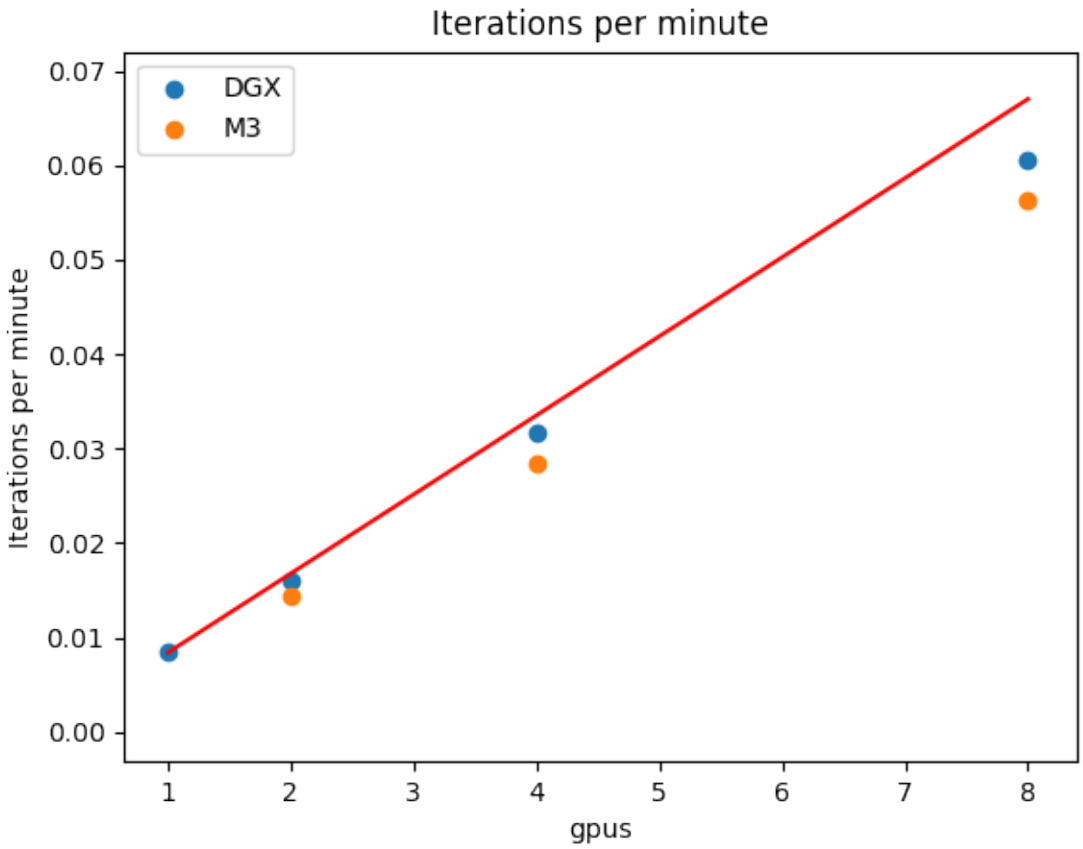


Fig. 2: Iterations per minute vs GPU count. Standard 2D classification benchmark, 5 iterations averaged, Relion 2.1b1. DGX: 8 P100 GPUs. M3, 2 P100 GPUs per node (maximum of 4 nodes used).

dont_combine_weights_via_disc	We use this.
pool	Usually small values (<10) works best.
no_parallel_disc_io	We don't use this.
preread_images	We don't use this, although it might help slightly if your entire dataset fits in memory.

17.1.5 Particle Polishing

Particle polishing jobs are not GPU accelerated. Your maximum MPI rank count should be equal to two times the number frames in your movie files.

Attention: This documentation is under active development, meaning that it can change over time as we refine it. Please email help@massive.org.au if you require assistance, or have suggestions to improve this documentation.

CHAPTER 18

Bioinformatics

MASSIVE supports the bioinformatics, genomics and translational medicine community with storage and compute services. On these pages we will provide examples of workflows and software settings for running common bioinformatics software on M3.

18.1 Requesting an account on M3

If you are requesting an account on M3 and are working in collaboration with the Monash Bioinformatics Platform, please ensure you indicate this in the application and request that the appropriate Platform members are added to your M3 project. This will enable them to assist in your analysis.

18.2 Getting started

18.2.1 Importing the bioinformatics module environment

M3 has a number of bioinformatics packages available in the default set of modules. These include versions of bwa, bamtools, bcftools, bedtools, GATK, bcl2fastq, BEAST, BEAST2, bowtie, bowtie2, cufflinks, cytoscape, fastx-toolkit, kallisto, macs2, muscle, phyml, picard, qiime2, raxml, rstudio, samtools, star, sra-tools, subread, tophat, varscan, vep (this list shouldn't be regarded as exhaustive !).

A software stack of additional packages (known as bio-ansible) is maintained by the Monash Bioinformatics Platform (MBP).

Tools are periodically added as required by the user community.

Modules maintained by MBP are installed at: `/usr/local2/bioinformatics/`

To access these additional tools, type:

```
source /usr/local2/bioinformatics/bioansible_env.sh
```

This loads the bio-ansible modules into your environment alongside the default M3 modules. If you are using this frequently you might like to source this in your `.profile` / `.bash_profile`.

To list all modules:

```
module avail
```

You will see the additionals tools listed under the `/usr/local2/bioinformatics/software/modules/bio` section, followed by the default M3 modules.

18.2.2 Installing additional software with Bioconda

In addition to the pre-installed modules available on M3, Bioconda provides a streamlined way to install reproducible enviroments of bioinformatics tools in your home directory.

conda is already installed on M3 under the *anaconda* module.

```
module load anaconda

conda config --add channels r
conda config --add channels defaults
conda config --add channels conda-forge
conda config --add channels bioconda
```

These channels will now be listed in *~/.condarc* and will be searched when installing packages by name.

Create a conda enviroment for your analysis / pipeline / project

Conda works by installing a set of pre-compiled tools and their dependencies in a self-contained 'enviroments' which you can switch between. Unlike modules loaded via `module load`, you can only have a single Conda environment active at one time.

To create an enviroment `my_proj_env` with a specific version of STAR, subread and asciigenome:

```
# Change this to your M3 project ID
export PROJECT=df22
export CONDA_ENVS=/projects/$PROJECT/$USER/conda_envs

mkdir -p $CONDA_ENVS

module load anaconda

conda create --yes -p $CONDA_ENVS/my_proj_env star=2.5.4a subread=1.5.2 asciigenome=1.
↪12.0

# To use the environment
source activate $CONDA_ENVS/my_proj_env

# Leave the enviroment
source deactivate
```

You can search for packages on the commandline with: `conda search <package_name>`, or on the web using the Bioconda recipes list.

For further detail see the official Bioconda documentation.

18.3 Running the RNAsik RNA-seq pipeline on M3

The RNAsik pipeline runs on the BigDataScript workflow engine. A template BigDataScript config file pre-configured for M3 has been prepared, however you need to create a copy and modify the `tmpDir` setting to reflect your M3 project.

If the file `~/.bds/bds.config` doesn't exist, create a copy and edit the `tmpDir` setting like:

```
# Ensure the BigDataScript module is loaded
source /usr/local2/bioinformatics/bioansible_env.sh
ml load BigDataScript

# Create a copy of the config file in your home directory
mkdir ~/.bds
cp $(which bds).config ~/.bds/

# Change this to your M3 project ID
export PROJECT=df22

# Edit the tmpDir setting to point to your scratch area
# (you can alternatively do this with vim / nano / emacs / $EDITOR)
sed -i 's/.*tmpDir.*/tmpDir = \/scratch\/${PROJECT}\/tmp\//' ~/.bds/bds.config
```

Create a SLURM sbatch script, `rnasik_sbatch.sh`, containing:

```
#!/bin/bash
#SBATCH --account=${PROJECT}
#SBATCH --profile=All
#SBATCH --no-requeue
#SBATCH --ntasks=1
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=2
#SBATCH --mem=2000
#SBATCH -t 3-0:00  # time limit (D-HH:MM)
#SBATCH --mail-type=FAIL,BEGIN,END
#SBATCH --mail-user=example.user@monash.edu
#SBATCH --job-name=rnasik

# NOTE: The SBATCH options above only apply to the BigDataScript (bds) process.
# (This is why --mem and --cpus-per-task allocations are small since these only␣
↪reflect the
# resources allocated to the BigDataScript pipeine runner, not STAR, picard etc).
#
# As part of running the pipeline, bds submits jobs to the queue with the appropriate␣
↪(larger)
# --cpus-per-task and --mem flags. If you need to add SBATCH flags to every job␣
↪submitted by
# bds (eg --account=myacct --reservation=special_nodes), add these flag to the
# clusterRunAdditionalArgs variable in the bds.config file.
#
# Default CPU and memory requirements are defined in bds.config and sik.config.

##
# Modify these variables as required
##

PROJECT=df22
```

(continues on next page)

(continued from previous page)

```
export REFERENCE_GENOMES="/scratch/${PROJECT}/references"
export FASTA_REF="${REFERENCE_GENOMES}/iGenomes/Mus_musculus/Ensembl/GRCm38/Sequence/
↪WholeGenomeFasta/genome.fa"
export GTF_FILE="${REFERENCE_GENOMES}/iGenomes/Mus_musculus/Ensembl/GRCm38/Annotation/
↪Genes/genes.gtf"
export FASTQ_DIR="./fastqs"
export SIK_VERSION="b55f2c7"
#############################

source "/usr/local2/bioinformatics/bioansible_env.sh"

ml unload perl
ml load BigDataScript
ml load RNAsik-pipe/${SIK_VERSION}
ml load R
ml load python/3.6.2
ml load bwa
ml load hisat2
ml load bedtools2/2.25.0
ml load qualimap/v2.2.1
ml load gcc/6.1.0
ml load multiqc/1.4
ml load picard/2.18.0

ml list

# A precaution to ensure Java app wrapper scripts control -Xmx etc
unset _JAVA_OPTIONS

# BigDataScript sometimes needs ParallelGCThreads set, but we don't want jobs
# to inherit these JVM settings.
# export _JAVA_OPTIONS=-Xms256M -Xmx728M -XX:ParallelGCThreads=1

export BDS_CONFIG="${HOME}/.bds/bds.config"

# /usr/local2/bioinformatics/software/apps/RNAsik-pipe-${SIK_VERSION}
export SIK_ORIGIN="$(dirname $(which RNAsik))/../"

# Run RNAsik bypassing the usual wrapper script so that we can specify additional␣
↪options
bds -c ${BDS_CONFIG} -log -reportHtml "${SIK_ORIGIN}/src/RNAsik.bds" \
    -configFile "${SIK_ORIGIN}/configs/sik.config" \
    -fastaRef ${FASTA_REF} \
    -fqDir ${FASTQ_DIR} \
    -gtfFile ${GTF_FILE} \
    -align star \
    -counts \
    -all \
    -extn ".fastq.gz" \
    -paired \
    -pairIds "_R1,_R2"
```

Run like:

```
sbatch rnasik_sbatch.sh
```

18.4 FAQ

Q : You have version xx and I need version YY, how do I get the software?

A : You should consider installing the software in your home directory. The Bioconda project helps streamline this process with many pre-packaged tools for bioinformatics. If you are unable to install the version you need, please contact the helpdesk help@massive.org.au

CHAPTER 19

Frequently asked questions

19.1 Accounts

How do I apply for an account on M3?

- Create an M3 account

How do I apply for an account on M3 if my organisation isn't an AAF member?

- Email the MASSIVE helpdesk and it will reply to you with further instructions

19.2 Jobs

How to check the status of my submitted job or when it's estimated to start?

- Checking job status

How long should sinteractive take to schedule?

- Jobs may start within seconds or take several days depending on the size of the job and current resource availability
- For further information go to Checking job status

How long should my job wait in the short queue?

- The short queue is for short jobs only, so you can expect your job to start within minutes. If a job in the short queue takes longer than 24 hours to start, contact the MASSIVE helpdesk

Why isn't my job running?

- Diagnosing problems with jobs

Why can't I use more resources?

- M3 is operated on shared use model and the MASSIVE team makes a best effort to share resources fairly. If you have special requirements or deadlines contact the MASSIVE helpdesk to discuss your needs

My job is taking a long time to be scheduled - how can I check if there's a problem?

- Try Checking job status and if you still believe there may be a problem contact the MASSIVE helpdesk

What is the command to start an interactive session?

- To start an interactive session, you should use smux. For more information, refer to Running interactive jobs

How do I run my job? (using slurm, not on the login node)

- Running jobs on M3

19.3 M3, desktops and Strudel

Why can't I connect to my desktop?

- Your account may not yet be activated. Once you've created your account the MASSIVE team has to associate it with a project in order for it become active. When this occurs the MASSIVE helpdesk will send you an email informing you that your access to CVL or M3 is ready along with instructions on how to set your password
- If you're using Strudel Desktop, have you installed the required software? See the FAQ "Why is Strudel Desktop not working" below for more information
- If your account has been activated, you've set up your password and you've installed all the required software but are still unable to connect to your desktop contact the MASSIVE helpdesk
- **For further information on connecting to your desktop select the link below appropriate to you:**
  - CVL users: Getting started on the CVL@M3 desktop
  - MASSIVE users: Requesting an account, Connecting to M3

What do I do if I forget my password?

- If you have entered an incorrect password more than five times in a ten minute period the system will block further attempts to log in for another ten minutes. If you've forgotten your password you can set it again by logging in to the HPC ID system using your organisational credentials and selecting the Change Linux password button to set your new password

What do I do if I forget my Strudel Desktop passphrase?

- You can recreate your passphrase key by deleting the old one, this will prompt you to create a new passphrase when you first log in with your MASSIVE ID. When you start Strudel select Identity from the menu then Delete Key from the drop down (note: to stop having to use the passphrase select Don't Remember Me; however you will need to use your account password to log in each time you access the system)

Why is Strudel Web not working?

- This problem is usually caused by accounts not yet having been activated: refer to the FAQ above "Why can't I connect to my desktop"

- If you've selected the SHOW DESKTOP button on your Strudel Web console but the tab that opens fails to load the desktop, close the tab, return to the console and select the SHOW DESKTOP button again. If the problem persists contact the MASSIVE helpdesk

Why is Strudel Desktop not working?

- **Have you installed the required software? For more information:**
  - MASSIVE users should go to: MASSIVE desktop (Strudel) using the Strudel Desktop application and select the button that expands the section titled "Instructions for software installation"
  - CVL users: go to Connecting to a CVL@M3 desktop using the Strudel Desktop application and select the button that expands the section titled "Instructions for software installation"
- Has your account been activated? To check this refer to the FAQ above "Why can't I connect to my desktop"
- **For further information on connecting to your desktop select the link below appropriate to you:**
  - CVL users: Getting started on the CVL@M3 desktop
  - MASSIVE users: Requesting an account, Connecting to M3

I have a slow connection, can I make TurboVNC run faster?

- Select Options from the TurboVNC menu. On the Encoding tab select Encoding method and then choose one of the "WAN" options. This will reduce the quality of the rendering but increase the interaction speed.

Is Strudel secure?

- Strudel uses an SSH (Secure Shell) tunnel to connect users to the MASSIVE visualisation nodes. All traffic between your computer and the node is encrypted in the tunnel

Can I use Strudel at my HPC or cloud site?

- Yes, this is easily implemented using a simple configuration file. Contact the MASSIVE helpdesk for more information

What should I do if I can't find M3 under the list of the sites on Strudel Desktop?

- You can add M3 to the list of the sites on Strudel Desktop by opening Strudel, then selecting "Manage sites" on the menu bar. This will allow you to select MASSIVE M3. Select OK to return to the main Strudel Desktop screen. You should be able to now select either "M3 Standard Desktop" or "M3 Large Desktop" under the "Site" section.

How do I request for software to be installed on HPC?

- Requesting an install

How do I request access to restricted software?

- Log in to HPC ID and select the [Software agreement] button from the left side menu then select the [Add software] button. Select the name of the software to open its Software Details page. To submit your request select the [I accept] button. You will receive an email notification with further information concerning your request within two business days

Is M3 suitable for running traditional Computational Fluid Dynamics or Molecular Dynamics that require large scaling?

- Current MASSIVE resources necessitate that you run only one or two large jobs (e.g. CFD, MD) at a time so to MASSIVE service can provide a fair and equitable usage of resources

How do I run my job? (using slurm, not on the login node)

- Running jobs on M3

Attention: This documentation is under active development, meaning that it can change over time as we refine it. Please email help@massive.org.au if you require assistance.

CHAPTER 20

About Field of Research (FOR) and Socio-Economic Objective (SEO) codes

MASSIVE collects Field of Research (FOR) and Socio-Economic Objective (SEO) codes as part of the project application process; these codes form part of the project data. Collecting FOR and SEO codes, as well as other metadata, allows MASSIVE to accurately report on our user base, apply for funding, and demonstrate our continuing contribution to the research community.

The Australian Bureau of Statistics (ABS) publishes a list of FOR codes and a list of SEO codes; we recommend consulting these documents when completing your project application.

We request FOR codes in the format `1234 @ 56.78%`, with a minimum of 4 digits for each FOR code.

If a *single* FOR code applies to your work, supply 100%. If *multiple* FOR codes apply to your work, supply multiple percentages that sum to 100%.

Important: For example: A project has two applicable FOR codes: 080302 Computer System Architecture, and 080501 Distributed and Grid Systems. If these codes are evenly weighted, supply `FOR=080302 @ 50%, 080501 @ 50%`. If one code is weighted more heavily than the other, supply `FOR=080302 @ 75%, 080501 @ 25%` or `FOR=080302 @ 55%, 080501 @ 45%`

We request a single SEO code in the format `123456`, with a minimum of 2 digits for the SEO code.

Further reading material can be found on the Australian Research Council (ARC) website.

CHAPTER 21

Miscellaneous

How does Strudel work and is it secure?

Strudel launches a remote desktop instance by programmatically requesting a visualisation node on the MASSIVE server, creating an SSH tunnel and launching TurboVNC.

Can I use Strudel at my HPC or cloud site?

Yes, this is easily implemented with just a simple configuration file and has been done for a range of sites other than MASSIVE or the CVL. Instructions to achieve this will be published soon. Until then please feel free to email us for more information.

Can I change the size of the Desktop after it has been started?

You can change the desktop size by entering a value in the "xrandr" command from a terminal on the desktop (e.g. *xrandr -s "1920x1080"*). If that does not work check the options in TurboVNC (ctl-alt-shift-o), newer versions have a "Remote desktop size" under "Connection Options", set this to "Server" or the size you would like.

I have a slow connection can I make TurboVNC perform better?

From the options menu (ctl-alt-shift-o) you can set the connection speed "Encoding method" to one of the "WAN" options. This will reduce the quailty of the rendering but increase the interaction speed.

I have forgotten my passphrase, how do I proceed?

You can recreate your passphrase key by deleting the old one, this will prompt you to create a new passphrase when you first login with your MASSIVE id. To delete the key: Indentity > Delete Key, from the Strudel menu. You can also avoid the key by using Identity > Don't Remember Me.

CHAPTER 22

CloudStor Plus

22.1 Getting Started

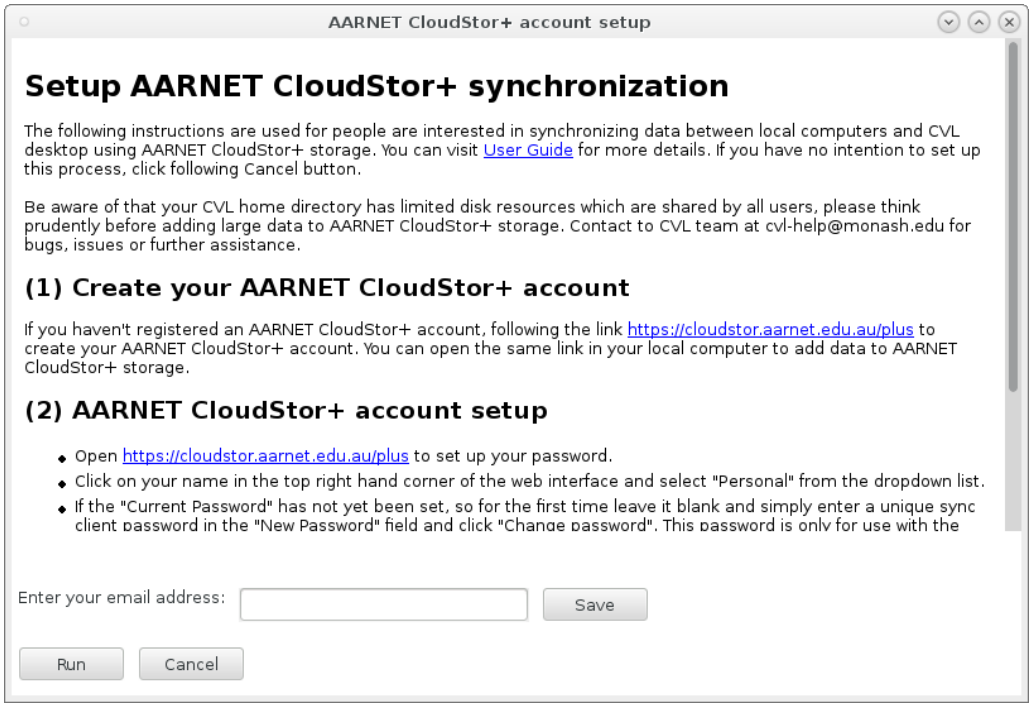
CloudStor Plus can be used to sync files between your laptop and your CVL Desktop. Please read the Cloudstor Getting Started Guide to find out how to register with CloudStor Plus.

Follow the instructions on Cloudstor Getting Started Guide to

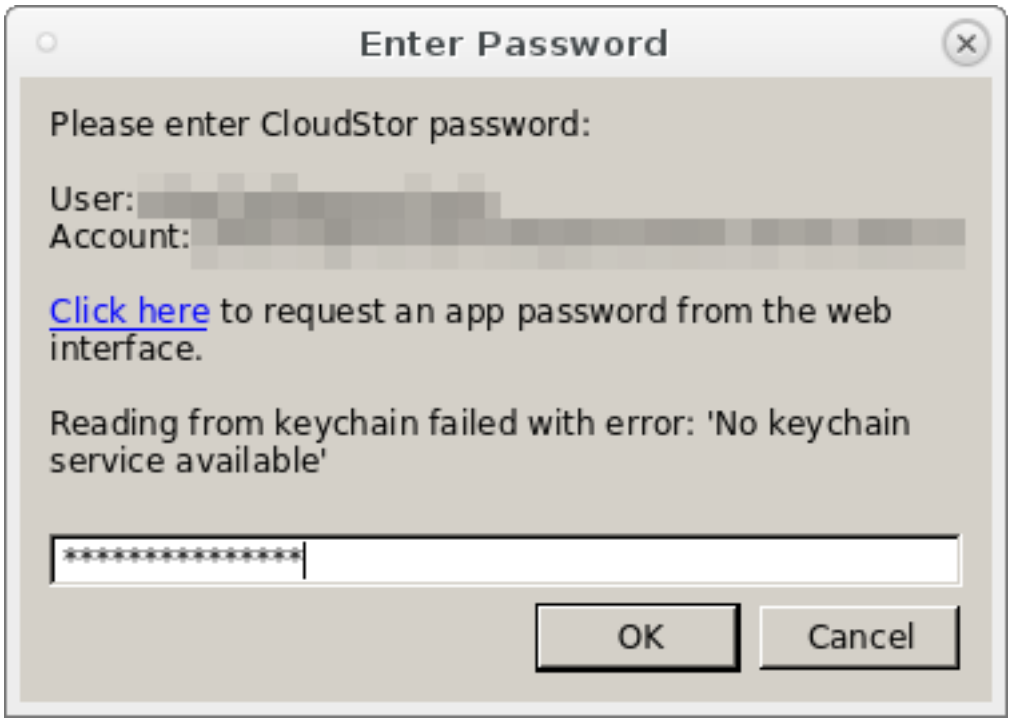
- Create an AARNET CloudStor Plus account if you don't already have one
- Set your CloudStor Plus desktop sync client password on the CloudStor Plus website

22.2 CloudStor Plus on the CVL Desktop

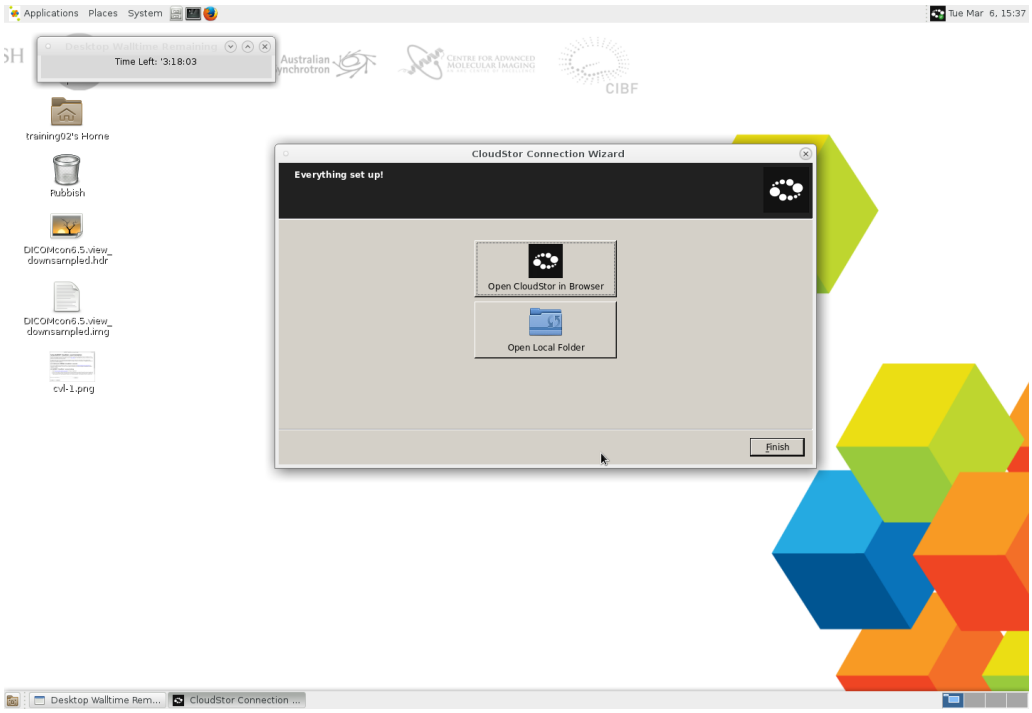
- **Select the following menu option:** *Applications -> Full List of Apps -> Aarnet CloudStor Plus 2.3.1-1.1*
  - Enter your CloudStor Plus email address in the email box. Click 'Save', then click 'Run'.



– Enter your CloudStor Plus desktop sync client password when prompted.

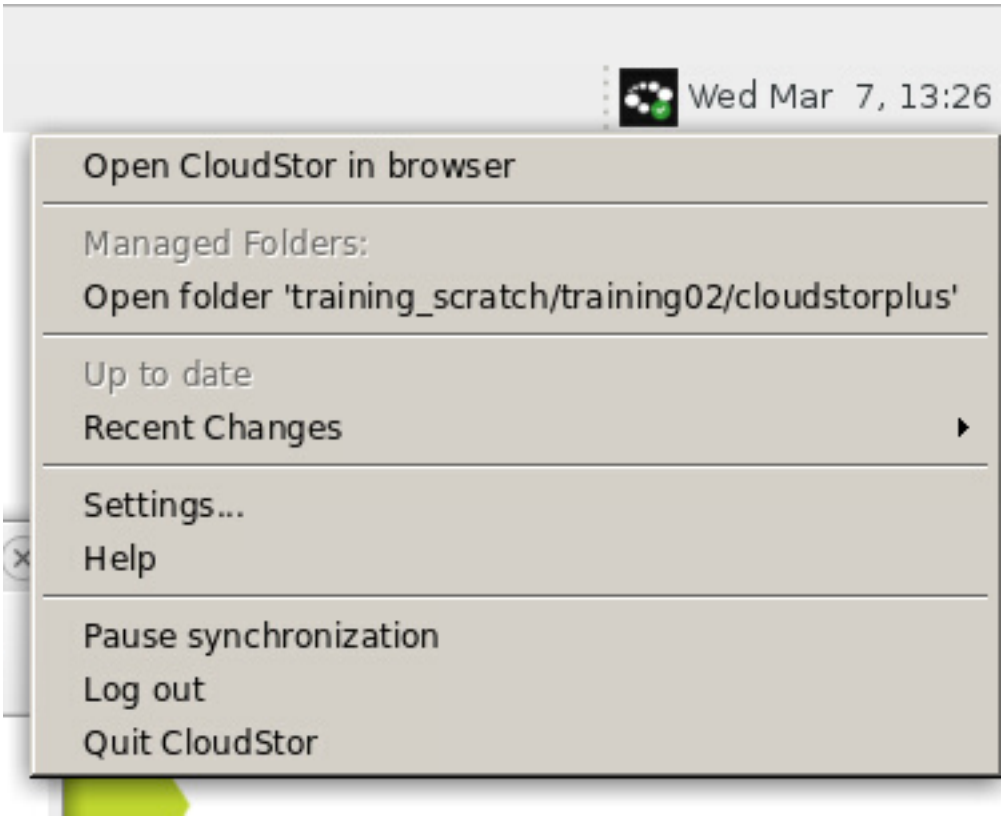


– Once the setup is complete, Cloudstor Conneciton Wizard will show you the option to either browse your files online or open your ClodStor local folder



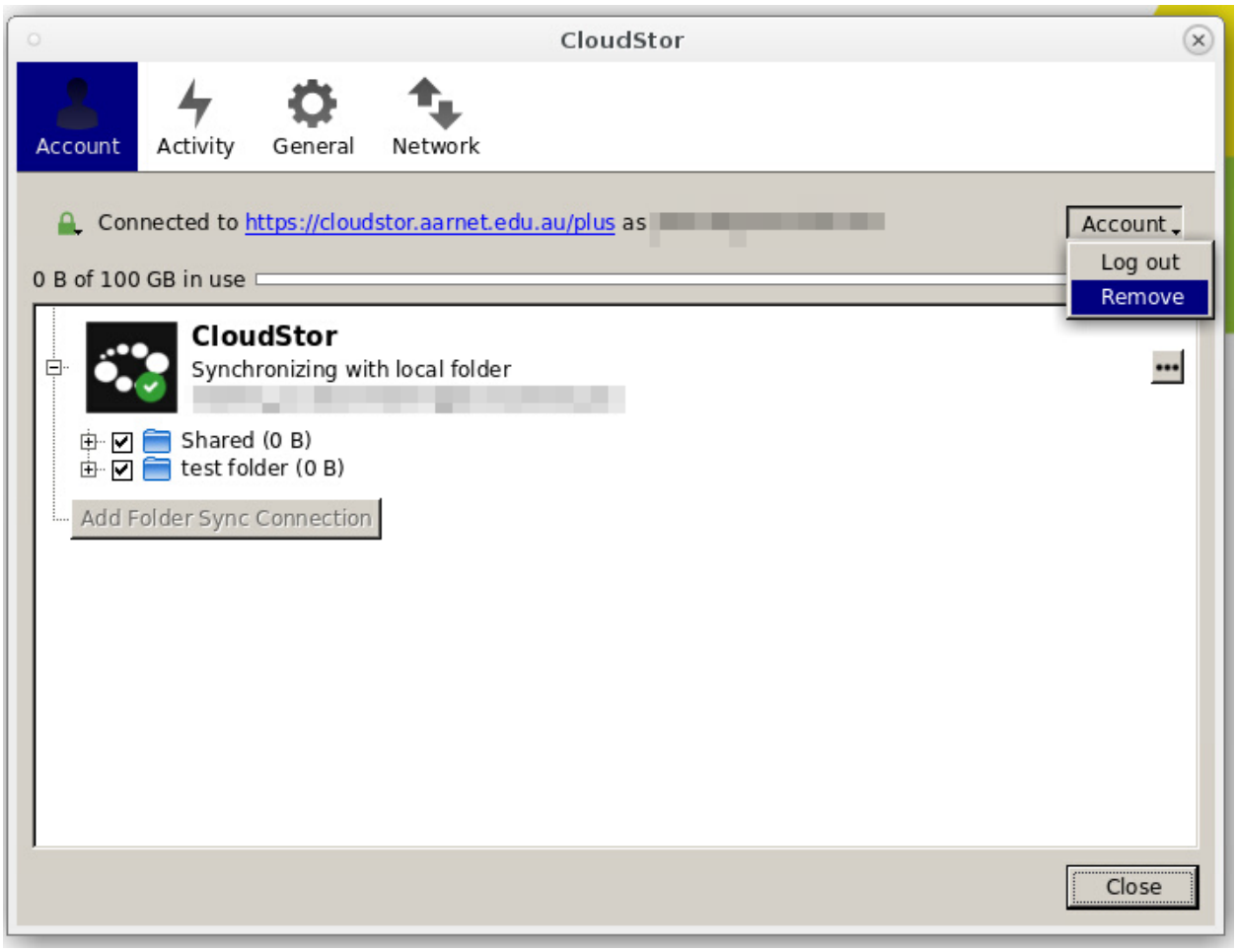
- The CloudStor Plus / OwnCloud icon appears on the top right of the CVL desktop

- You can right-click on the CloudStor Plus icon and select 'Open folder ownCloud' to access your CloudStor Plus storage on the CVL desktop



22.3 Deactivating your CloudStor Plus Account on the CVL Desktop

- Right-click on the CloudStor+ icon on the top right corner of the CVL desktop and select *Settings*
- Click on *Remove* in the Account settings to stop using the CloudStor+ desktop sync client.
- NOTE: Removing your account from the desktop client does NOT delete your files in your CVL home directory
- Stop the desktop session and re-start a new session


CloudStor
Account
Activity
General
Network
Connected to https://cloudstor.aarnet.edu.au/plus as
Account
Log out
Remove
0 B of 100 GB in use
CloudStor
Synchronizing with local folder
Shared (0 B)
test folder (0 B)
Add Folder Sync Connection
Close