University of the Witwatersrand, Johannesburg
Mathematical Sciences Support

The Mathematical Sciences HPC Cluster

Community Guidelines

September 24, 2023

Contents

1 Introduction

mscluster, the Mathematical Sciences High Performance Computing (HPC) Cluster is managed by Mathematical Sciences Support (MSS) on behalf of staff and students in the Mathematical Sciences at the University of the Witwatersrand, Johannesburg.

While this resource in intended primarily for those within the Mathematical Sciences, staff and students from other Schools and Faculties are welcome, although usage may need to be negotiated when demand exceeds availability.

1.1 CHPC

The system is meant to compliment larger national HPC systems such as the Center for High Performance Computing (CHPC), by completing 'smaller' workloads faster and closer to home before burdening larger national systems. We find that many Honours, M.Sc, and Ph.D projects can be completed successfully using mscluster, with only a few needing to be migrated to the CHPC.

Using mscluster also provides you with an opportunity to learn how to be a 'good HPC citizen' by making big mistakes on our cluster first, thereby ensuring that your time at the CHPC is used responsibly.

1.2 A Cast of Many

mscluster is possible only because of partnerships with the CHPC, international donors such as Texas Advanced Computing Center (TACC), Wits ICT, PIMD, all Schools in the Mathematical Sciences, and the Technical Laboratory Assistants (TLAs) who help maintain the cluster. mscluster is brought to you by a cast of many.

Use of the cluster is an advantage for Wits staff and students who require HPC. This is done without any cost recovery.

At the time of writing (September 2023), the value of equivalent world-class HPC services internationally is around R18.75, per node, per hour.

1.3 Federation

Wits handles HPC operations via a federated approach. This was not by design, but by systems evolving organically. This federated approach has proven to be a great advantage to our University.

There are currently three openly available clusters.

Wits Core Cluster: Electrical Engineering. Medicine. Bioinformatics. CERN.

The CrunchYard: Commercial code bases. Commercial users. Advanced UI.

mscluster: Teaching, Honours, M.Sc, Ph.D research. SKA.

In addition to these, there are number of other smaller clusters, that "run hot". These are research specific machines that typically run 24/7/365 with no headroom. This is often the case

where a cluster is intended to run very mature pieces of code, and there is no downtime due to code development. mscluster is by comparison, "bursty". mscluster also aims to be lean, agile, and student driven which has its benefits.

As a staff member or student in the Mathematical Sciences, mscluster is what you have access to by default, and where MSS can provide the best support.

1.4 Electricity and Cooling

Offering an open and multi-purpose HPC service within the context of load shedding is not trivial. Many local Universities simply don't.

Please be aware of load shedding schedules, both reported and actual, for the Braamfontein area. If there is any load shedding, expect access, availability, and performance to be affected.

With its many potential points of failure, and many required service providers, recovering a damaged cluster can be complex, and can take time. System recovery after load shedding events can vary from a simple reboot, to repairs lasting days.

From our experience, the mscluster, even with a UPS and generator protecting it, can survive one load shedding event per day. With three events a day, problems are inevitable, and staff are working outside of office hours. With more than three events a day, equipment is physically breaking.

2 Community Matters

The mscluster is not a commercial HPC environment, and it cannot be run as one. Here community matters. Service providers, donors, administrators, supervisors, students, courses, TLAs, Interest Groups, all have a role to play.

By using mscluster, you acknowledge to have received and read this document, which encourages the following operational and community values.

2.1 The Responsibility of the Supervisor

Users may only run code as part of a supervised research project or coursework. In the case of coursework, the course lecturer is considered to be the Supervisor.

We ask that supervisors ...

- Make certain that problems being tackled are indeed HPC problems.
- Routinely monitor computational results.
- Make sure that computational results are moving towards a solution or at least a better understanding of the problem.
- Have basic familiarity of mscluster, and what it can and cannot do.
- Consider signing up as a Principle Investigator at the CHPC and keep a reserve of credits for jobs that the mscluster cannot handle.

- Use monthly usage statistics for their research groups, to make sure the amounts seem sensible for the problems being tackled.
- As discussed later, provide the first escalation point beyond MSS when resolving technical problems.

2.2 The Responsibility of the Research Student

As a member of a community, research students have some responsibilities.

2.2.1 Logging Tickets

Please do not email staff members directly. Instead open a single ticket per issue by sending an email to the MSS Help Desk. This is detailed later.

2.2.2 The Escalation Path

If you experience a problem that you feel was not resolved via the MSS Help Desk, the correct escalation path is ...

- Arrange a meeting with the MSS team via the Help Desk.
- Failing a resolution, ask your Supervisor to discuss the problem with MSS.
- Failing a resolution, your Supervisor may raise the matter with your Head of School.
- Failing a resolution, your Head of School may raise the matter with the appropriate Dean or Assistant Dean in the Faculty of Science.

Resolving technical problems via this route will reveal, preferably sooner than later, if the problem was transient and needs no further attention, if the problem is solvable given our constraints, or if the problem is in fact unsolvable given our constraints.

Failure to follow this escalation path is both unprofessional and in bad taste.

2.2.3 Understanding Constraints And How HPC Clusters Work

You are using a scientific instrument. It makes sense to get a high-level understating of how the instrument works, and what its bottlenecks and points of failure are, so you can moderate your expectations and figure out out how to use the cluster creatively. We have been amazed at how users have found ways to push ever closer to the limits of mscluster hardware.

CHPC courses are a good place to start.

2.2.4 Be Aware Of Load Shedding

As discussed above, load shedding is a reality for our cluster. UPS and generator systems have physical limits, and will only go so far to protect our cluster. The popular ESP app provides a good way to be aware when there is load shedding.

2.2.5 Avoid Hogging the Login Nodes

Please avoid doing any work on the login node (146.141.21.100) that makes heavy use of CPU, memory, disk IO, or network IO as as this will affect all users. If you are not using sbatch or srun, you are affecting everyone else.

A second login node (146.141.21.101) is provided should the primary node become unusable due to misuse. Extra security measures are taken on this second node to ensure that it is not easily abused. Please avoid using this node, except when you have reason to believe that there is a problem with the primary login node.

2.2.6 Avoid Ghost Processes

Often users write code that does not exit cleanly, leaving ghost processes running on the worker nodes. Please verify that your code exits cleanly, or prepare a clean-up process.

2.2.7 Target Computations To The Appropriate Hardware

Please use hardware responsibly, in the sense of not hogging high-end nodes without making effective use of them. We suggest first running on stampede. If this does not meet your requirements, only then scale up to bigbatch. And only if bigbatch does not meet your needs, then scale up to biggpu.

biggpu in particular should only be used for mature debugged code - because those three nodes are very expensive to run and maintain, and are in high demand due to the limited supply.

Each partition has unique hardware benefits that are noted later.

2.2.8 Don't Flood The Cluster

In a pay-per-use scenario, (Quality of Service) QoS is a clear-cut matter, and enforcible at a system level. Our environment is more communal and more complex. Enforcing QoS measures solves one class of problem by creating others.

We'd prefer to not overly police the mscluster with QoS. Be considerate. Don't flood the cluster. Give others a chance. MSS will step in with QoS measures when needed, but we'd prefer not to.

2.2.9 Personal Storage and Computation Is Not Allowed

Please do not use the cluster to store any data not required for the computational part of your research. The cluster is not intended as general purpose storage.

Similarly, the cluster is meant as a HPC computational research tool. Using it for personal use, or for any kind of non-HPC computation, is not allowed.

2.2.10 Work With Your Peers

Working with your research peers is the best way to rapidly develop and benefit from a community of practice. As an example, apptainer (previously called singularity) containers provide

opportunities for groups of students to work together to encapsulate and share common workflows, and circumvent common software configuration challenges in their domain.

2.3 Acknowledgements Matter

mscluster is dependent on the kindness of our donors - who usually ask no more than to see results. To assist fund-raising efforts for continued service for future staff and students we would appreciate acknowledgement in any thesis, paper, research report, project, publication or presentation that references results computed on Mathematical Sciences infrastructure.

The suggested form of acknowledgement is ...

"Computations were performed using High Performance Computing infrastructure provided by the Mathematical Sciences Support unit at the University of the Witwatersrand, Johannesburg."

3 Finding Information

We understand that using the cluster can be daunting. While other applied domains enjoy high-level interaction with HPC, HPC for researchers in the Mathematical Sciences remains very 'close to metal'. Here are some ways that you can learn from and contribute to our community of users.

3.1 Help Desk

Please do not email staff members directly. MSS coordinates work via a virtual Help Desk. Open a single ticket per issue by sending an email to the MSS Help Desk. In this way you will have the benefit of having a track-able ticket number, being assured that the issues will be directed to the correct person, and having multiple eyes on your problem.

Tickets are ideal for usage queries, software changes, training requests, and system problem reports.

Support queries can be sent to support@wits-mss.supportsystem.com, using the subject line "TWK HPC Query".

You must send the email using your student or staff email.

3.2 Undergraduate HPC Interest Groups

Every year, there is one short-form activity, being a general HPC Interest Group for undergraduates. There is also a long-form activity, being the training of undergraduates for competitive HPC events. While aimed at undergraduates, these may still be useful to attend or contribute to.

3.3 Software Carpentry Training

If you can gather a large enough group (between 10 an 15 people), you can request a one-day Software Carpentry style workshop in HPC via the MSS Help Desk.

3.4 The mscluster Wiki

Please consult, and consider contributing content to the mscluster Wiki.

4 Some Tips

Here are some general tips.

- If you are not familiar with Unix like environments, read https://swcarpentry.github.io/shell-novice/.
- As far as is possible, test your code for correctness on your desktop machine first. Avoid using the cluster or the login nodes as your main development environment.
- It is worth reading about Amdahl's Law and Lhadma's Law if you are not already familiar with these.
- Your code needs to be written for parallel execution, using libraries such as MPI, or frameworks such as TensorFlow that are designed to leverage multiple CPUs or multiple GPU cores. The cluster cannot magically take serial code, and run it in parallel for you.
- Always start with a simple program that completes quickly. Get a basic work-flow correct first.
- Your code should periodically write out some indication of its progress. Catching problems early by scrutinizing such intermediate reports benefits everyone. For example, a tiny progress report written every six hours is lightweight, practical, and can save hundreds of compute hours!
- Monitor CPU, RAM, GPU percentage, GPU RAM, Disk IO, and Network IO statistics of the nodes that you're using, using the online monitoring page. Alternatively your code should periodically write out some indication of its resource usage. Make sure that what you're seeing makes sense and is what you expect.
- Understand that even though your code may be making heavy use of the CPU, GPU, storage, or network, doesn't mean you are converging fast enough, or at all, to a meaningful result. Again, discuss checks and measures with your supervisor.
- Historically, the mscluster has close to zero headroom between September and November every year. Those are our busiest months. Start your project early. Your experience will be less frustrating during the other eight months of the year.

5 Partitions

Submit jobs to the partitions with hardware that is best suited for your code. Please work with your supervisor to ensure that your code is indeed leveraging the special hardware being offered by each partition.

Memory is not explicitly limited on partitions, but you are of course limited by the amount of RAM on the nodes as described below.

5.1 stampede

For general purpose use or jobs that can leverage InfiniBand.

40 nodes, each with two Xeon E5-2680 CPUs, two GTX1060 GPUs (6GB per GPU, 12GB per node), and 32GB of system RAM. MaxTime=4320 and MaxNodes=20

Nodes mscluster[2-21] are connected via one InfiniBand switch. Separately, nodes mscluster[22-42] are connected via another InfiniBand. Node selection is important for MPI jobs, to avoid communication across switches.

5.2 batch

For bigger jobs that can leverage a bigger GPU and additional system RAM.

48 nodes each with a single Intel Core i9-10940X CPU (14 cores), NVIDIA RTX3090 GPU (24GB), and 128GB of system RAM. MaxTime=4320 and MaxNodes=48

batch has the potential for 10Gb networking. This has not yet been implemented.

5.3 biggpu

For mature code that can meaningfully leverage very large amounts of GPU and system RAM.

4 nodes, each node has two Intel Xeon Platinum 8280L CPUs (28 cores per CPU, 56 cores per node) with two NVIDIA Quadro RTX8000 GPUs (48GB per GPU, 96GB per node), and 1TB of system RAM. MaxTime=4320 and MaxNodes=4

biggpu has the potential for 10Gb networking. This has not yet been implemented.

5.4 Quality of Service

Various Quality of Service (QoS) measures may be implemented throughout the year. For example, reservations may be negotiated, time extensions of jobs may be granted, personal and group quotas may be enforced, MaxJobs and MaxSubmitJobs may be controlled per user, per partition, or per research group.

In general, we prefer to not use QoS measures.

6 Basic Usage Examples

Try these examples to get a basic idea of how to use the cluster.

6.1 Getting Some System Information

Run these commands, and try to understand their output. 12345 is of course an example JOBID. Use a real JOBID.

```
sinfo
squeue
scontrol show job 12345
```

6.2 Running Commands Directly

Run these commands, and try to understand their output

```
srun -N6 -p batch -l /bin/hostname
srun -N2 -p biggpu -l cat /proc/cpuinfo | grep model
srun -N4 -p stampede -l /usr/bin/uptime
```

In general, avoid using srun. Interactivity can cause problems, and load shedding complicates these jobs. Use sbatch as far as possible.

6.3 Running A Job Via A Batch File

If mmouse is your username, for example, given a text file called test.sh, and with the contents ...

```
#!/bin/bash
#SBATCH --job-name=test
#SBATCH --output=/home-mscluster/mmouse/result.txt
#SBATCH --partition=batch
sleep 60
/bin/hostname
echo ‘expr 3 + 2‘
```

... submit a job to the queue using the command ...

```
sbatch test.sh
```

... and monitor its progress using.

```
squeue
```

Once the job has completed, check if the contents of the file result.txt makes sense.

6.4 Cancelling Jobs

12345 and mmouse are of course examples. Use a real JOBID and your username here.

To cancel a specific job.

```
scancel 12345
```

To cancel all your jobs.

```
scancel --user=mmouse
```

6.5 Other Useful Flags

A specific node can be requested by including a -w flag. For example ...

```
#SBATCH -w mscluster61
```

For parallel jobs, a specific set of nodes can be requested as follows. Of course, you code needs to be written to exploit multiple nodes.

```
#SBATCH --nodelist=mscluster[2-4]
```

For parallel jobs, the number of nodes can be requested as follows. Of course, you code needs to be written to exploit multiple nodes.

```
#SBATCH --nodes=2
```

7 The Different Storage Areas And Their Uses

We provide various caching and networking opportunities to allow the wrangling of large data sets. In general wrangling a lot a data is best done locally within TW Kambule Laboratories.

7.1 Home Directories

If your username is mmouse, you have a system-wide folder called /home-mscluster/mmouse/ for you to use. This is where you land when you log into the login node.

This is your general working space intended for your development environment, your code, and up to 50GB of data. Datasets that exceed 50GB need to be kept in /datasets/mmouse/ as described later.

7.2 Fast Local Scratch Space

Jobs with high levels of disk IO may affect the cluster performance by flooding the network. For cases such as these we provide limited scratch space on every node. If your username is mmouse, each node has a local folder called /scratch/mmouse/ for you to use.

These folders are not for long term storage. Up to a few weeks is fine. If it's affecting other users or the overall system, we may clear out this data with little warning.

Please be sure to clear out files in all of your scratch folders as leaving them there will affect other users. A tool such as cssh is useful for managing all of your scratch folders - just be careful to not run any code directly using cssh as you will likely affect other users. Use it only to stage and delete your data.

7.3 Fast Network Scratch Space

Staging data as with the fast local scratch spaces (as described above) can be tedious because you are managing multiple folders across multiple nodes. As an alternative we offer fast network

scratch space via a GluserFS filesystem. If your username is mmouse, you have a system-wide folder called /gluster/mmouse/ for you to use.

This folder is not for long term storage. Up to a few months is fine. If it's affecting other users or the overall system, we may clear out this data with little warning.

7.4 Large Datasets

We provide storage for very large datasets, typically range of 50GB to a few terabytes. If your username is mmouse, you have a system-wide folder called /datasets/mmouse/ for you to use.

8 Sideloading Large Datasets

In unusual cases involving multi-terabyte data sets, we do have the capability to load data onto the cluster using dedicated NICs and disks, to avoid affecting other users. Similarly, multi-terabyte data sets can be copied off the cluster, without affecting other users.

In unusual cases, data upload or download using USB external drives is possible. Arrange a meeting via the Help Desk if you think this is required.

In the following three cases, please don't do any work that doesn't involve putting data where it needs to be. Pre-processing of data needs to handled as a job submission, unless it uses a trivial amount of CPU, RAM, or Disk resources.

8.1 Sideloading Large Datasets Onto /datasets/

If your username is mmouse, to sideload data, using wget for example,

```
ssh mmouse@10.100.14.250
cd /datasets/mmouse
wget http://<URL-TO-YOUR-DATA>
```

8.2 Sideloading Large Datasets Onto /gluster/

If your username is mmouse, to sideload data, using wget for example,

```
ssh mmouse@10.100.14.2
cd /gluster/mmouse
wget http://<URL-TO-YOUR-DATA>
```

8.3 Sideloading Large Datasets Onto /home-mscluster/

Ideally, don't.

There are very rare situations where you may need to sideload data into you home directory. If you suspect you have a need for this, please discuss the use case with the MSS team first.

If your username is mmouse, to sideload data, using wget for example,

```
ssh mmouse@10.100.14.252
cd /home-mscluster/mmouse
wget http://<URL-TO-YOUR-DATA>
```

9 Computational Software

A selection of computational software relevant to the Mathematical Sciences is installed. However, maintaining software locally in your home directory is encouraged wherever possible.

9.1 Anaconda

While we do provide a decent Anaconda installation, it is HIGHLY recommended that you install Anaconda in your home directory, and manage your own personal installation. Anaconda installations can vary wildly from person to person, so one size cannot fit all.

9.2 TensorFlow, PyTorch, etc.

Libraries related to Machine Learning are best managed by yourself, typically using Anaconda as described above.

9.3 Apptainer

Apptainer (previously called Singularity) is available. It is strongly advised that work-flows that are common within your research group be encapsulated using containers. The same applies for software chains that are tricky to configure but need to be frequently deployed. Please work with members in your research group and with your supervisor to develop and promote these containers.

9.4 CUDA

We strongly recommend using CUDA version 12.0.

There are however multiple versions of CUDA installed in /usr/local/. Point to the version that is correct for your code. For example,

```
export PATH=/usr/local/cuda-12.0/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/cuda-12.0/lib64:$LD_LIBRARY_PATH
export CUDA_HOME=/usr/local/cuda-12.0
```

If in doubt, point to the latest that is available.

If you run into an incompatibility problem with CUDA versions, please open a ticket via the Help Desk.

9.5 R

While we do provide a decent R installation, it is highly recommended that you compile R, and install R packages into your home directory as you require them.

9.6 MATLAB, Mathematica, and Others ...

We add less frequently used software, such as MATLAB and Mathematica, in /usr/local/.

List the contents of /usr/local/ to see if anything there is useful to you. If you find something you would like to use, we recommend updating your PATH variable in .bashrc to make running that software easier.

Please request software by opening a ticket via the Help Desk.

9.7 Useful Software for Windows Users

We recommend using PuTTy for logging onto the cluster and basic interaction.

Historically we would recommend WinSCP for file transfer, but we have recently seen examples of users experiencing problems with WinSCP.

Both of PuTTy and WinSCP remain available on the Net for free.

In general, we are currently only testing user experience using Ubuntu Linux, so your mileage with Windows or macOS may vary.

10 Thanks

Wits has a long and proud HPC history. Everyone in our community owes thanks to everyone else, current and past. MSS however would like to thank the following people and organizations, without whom, mscluster would not be possible.

- Prof. Ebrahim Momoniat for his leadership in scaling up all MSS operations, including mscluster, to world-class levels.
- Prof. Scott Hazelhurst for being the first strong proponent of HPC at Wits.
- Prof. Turgay Celik for his wise stewardship of the Ranger donation from TACC.
- Mr. Bryan Johnson for growing the local HPC community via the CHPC Ecosystems Project.
- CHPC and TACC for their endless support and encouragement.
- PIMD for their daily efforts to provide power in the face of load-shedding.
- The Wits ICT Networking team for keeping our network up in the face of their many challenges.
- You - a researcher using mscluster. Your daily dive into the unknown is why TLAs and MSS staff come to work every morning.