HPC Cluster Onboarding

Research Computing

Medical College of Wisconsin

Updated Feb 2026

Overview

- Research Computing
- Accounts & Access
- Cluster Resources
- Storage & Data Transfer
- Using Software in the Cluster
- Submit, Monitor and Troubleshoot Jobs
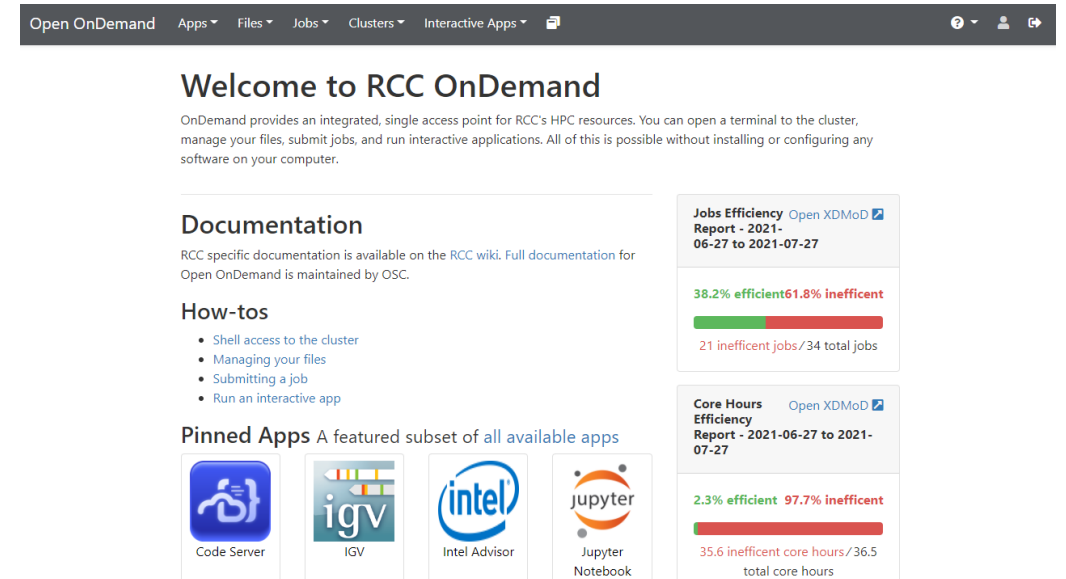- Getting Help

Research Computing

- ## What is Research Computing?
  - Research Computing is a division of MCW-IS that **provides campus-wide access to high performance computing** (HPC) resources that are designed for computational biomedical research.

- ## What is HPC?
  - HPC enables users to **tackle problems too large for a typical desktop or laptop** by leveraging high-speed compute, storage, and network resources to perform complex calculations.

- ## How much does it cost?
  - All MCW faculty are eligible for an allocation of free storage and unlimited computing time for their lab. Additional storage is available for fee.

Request an RCC account

- RCC user accounts are available to all MCW researchers

- Visit https://docs.rcc.mcw.edu/user-guide/accounts/

- Most requests are handled same day

- *To obtain RCC access, you must:*
  - *Be a PI or sponsored by a PI*
  - *Have an active MCW account.*
  - *A PI may sponsor students, postdoctoral fellows, staff, or colleagues with whom they are collaborating on research.*

Access HPC Cluster

- Open OnDemand – web-based portal
  - Access cluster CLI
  - Manage files
  - Manage, write and submit jobs
  - Run pre-configured interactive apps (i.e. Rstudio, Jupyter, Remote Desktop)
  - All of these are possible without leaving your browser!
  - **This is the recommended login method for most users.**

- More information: https://docs.rcc.mcw.edu/user-guide/access/ondemand/

- To connect remotely you must have VPN or use Citrix.

Access HPC Cluster

- SSH (Secure SHell) client
  - Secure method of connecting to an RCC server
  - Command-line interface
  - Windows clients:
    - Secure Shell Client
    - Putty
  - Mac clients:
    - Built-in Terminal App
    - Iterm2
  - Linux
    - Built-in terminal
- Info available on https://docs.rcc.mcw.edu/user-guide/access/login/

HPC Cluster Resources

- Login Nodes
  - 4 login nodes
  - 24 cores/node
  - 128GB memory/node

- Use:
  - **Not** for computationally intensive work
  - Editing scripts
  - Submitting jobs
  - Checking the status of jobs
  - Troubleshooting jobs
  - Interactive tasks

Script editing,
Job submission,
Pre/post processing

User

Login Node

Resource management and
job management

Scheduler

Batch job processing
Interactive jobs
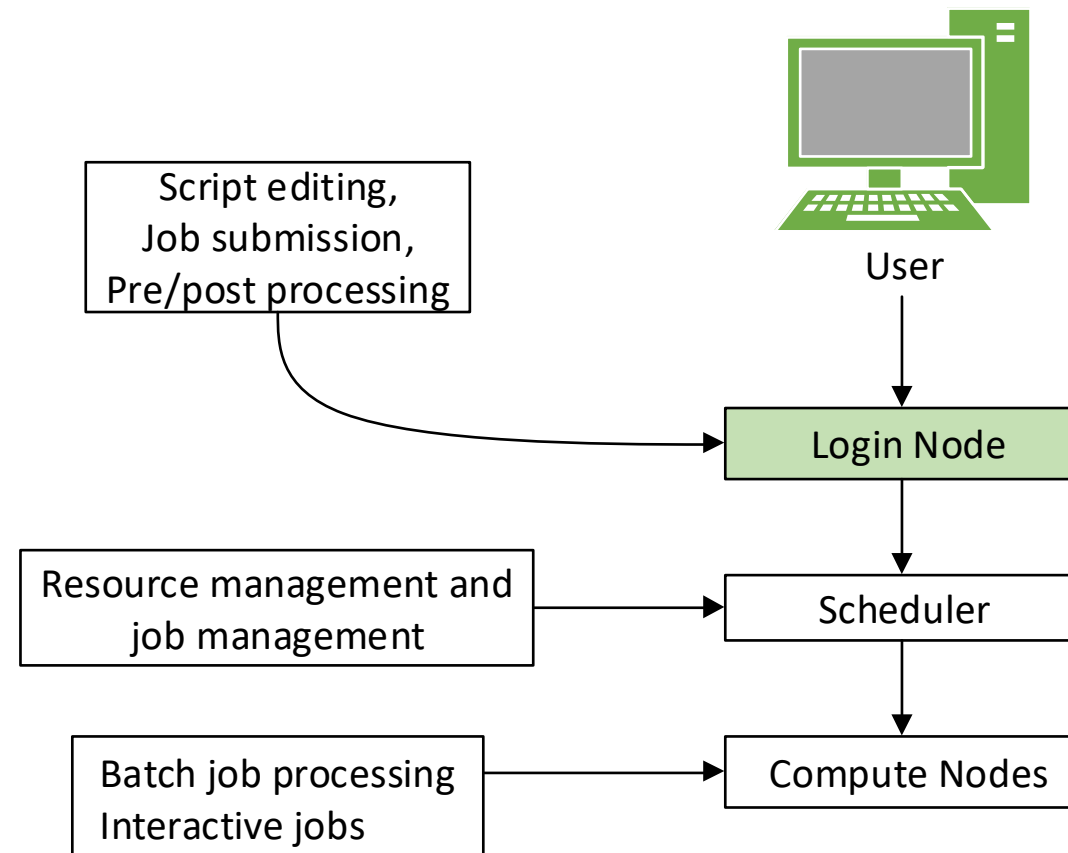
Compute Nodes

HPC Cluster Resources

- Compute Nodes

Standard Nodes	Large memory nodes	GPU nodes
60 nodes	2 nodes	9 GPU nodes
48 cores/node	48 cores/node	48 cores/node
7.5 GB memory/core	32 GB memory/core	7.5 GB memory/core
480GB SSD	480GB SSD	480GB SSD 4 V100 NVIDIA GPUs

Script editing, Job submission, Pre/post processing

User

Login Node

Resource management and job management

Scheduler

Batch job processing Interactive jobs

Compute Nodes
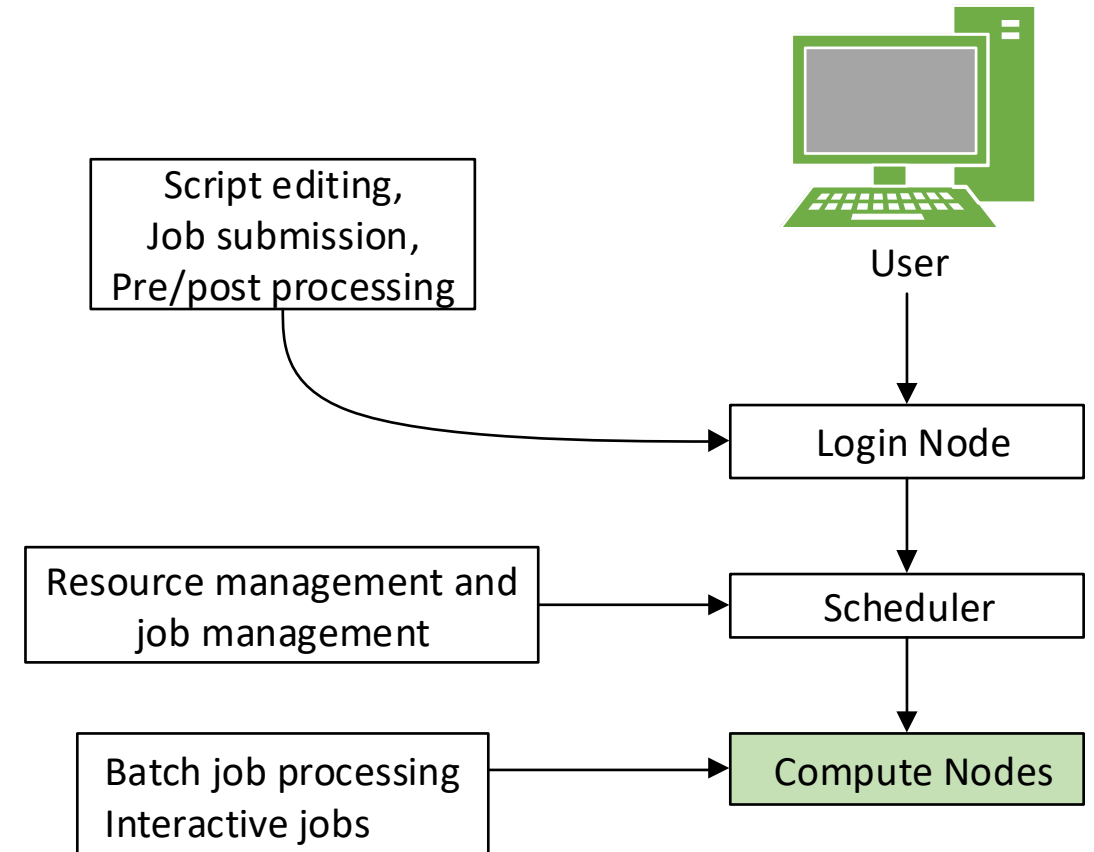
Available Storage

- Free to all users:
  - /home/*NetID* – 100Gb hard limit
  - /group/*PI_NetID* – 1TB free limit
  - /scratch/g/*PI_NetID* – Group scratch directory
    - 25TB temporary (NVMe) storage for runtime files
  - /tmp – local scratch storage for active jobs
    - Local disk (480GB) on compute nodes
    - Do not persist across nodes or jobs, not backed up

- For fee:
  - Research Group Storage - /group/*PI_NetID*
    - First 1TB is free
    - $80/TB/year for additional storage

You can easily find your available storage directories and current utilization on the cluster with the `mydisks` command.

```
$ mydisks
=====My Lab=====
Size  Used Avail Use% File
47G    29G   19G  61% /home/user
932G  158G  774G  17% /group/pi
4.6T     0  4.6T   0% /scratch/u/user
4.6T     0  4.6T   0% /scratch/g/pi
```

Disk space will show as less than quota limit. This is due to base-2 vs. base-10 math. Rest assured, the actual limit is being enforced despite *mydisks* output

File Transfer – Command line SCP

- ## Copy a file to the HPC Cluster:

  `scp local_file user@login-hpc.rcc.mcw.edu:/path/to/remote/target-directory`

- ## Copy a directory to the HPC Cluster:

  `scp -r local_directory user@login-hpc.rcc.mcw.edu:/path/to/remote/target-directory`

- ## Copy a file from the HPC Cluster:

  `scp user@login-hpc.rcc.mcw.edu:/path/to/remote_file /path/to/local/target-directory`

- ## Copy a directory from the HPC Cluster:

  `scp -r user@login-hpc.rcc.mcw.edu:/path/to/remote_directory /path/to/local/target-directory`

File Transfer – Command line RSYNC

- Copy a file to the HPC Cluster:

`rsync -avz local_file user@login-hpc.rcc.mcw.edu:/path/to/target-directory`

- Copy a directory to the HPC Cluster:

`rsync -avz local_directory user@login-hpc.rcc.mcw.edu:/path/to/target-directory`

- Copy a file from the HPC Cluster:

`rsync -avz  user@login-hpc.rcc.mcw.edu:/path/to/remote_file /path/to/local/target-directory`

- Copy a directory from the HPC Cluster:

`rsync -avz  user@login-hpc.rcc.mcw.edu:/path/to/remote_directory /path/to/local/target-directory`

File Transfer – Command line RCLONE

- Command-line tool for syncing, copying and transferring files between the cluster (or local) storage and cloud services (i.e. OneDrive, Google Drive, S3, Dropbox).
- Ideal for moving large datasets to or from the cluster through the Terminal.
- It is a more advanced tool, for more details on how to configure and use it please visit [our documentation.](#)

File Transfer – FTP clients

Tool	Platforms	Pros	Cons	Heavy Transfers
MobaXterm	Windows	• SSH Terminal • Run windowed apps from the cluster	• Windows only • Heavier app	• Not optimized for huge multi-threaded queues • Stable for routine SFTP
WinSCP	Windows	• Scripting to automate file transfers	• Windows only • No SSH Terminal • No windowed aps	• Good for long SFTP jobs • Reliable queues and resume
FileZilla	Windows MacOS Linux	• Cross-platform • Simple GUI • Good queue manager	• No SSH Terminal • No windowed apps	• Fast and stable for large batches
CoreFTP	Windows	• Lightweight • Simple GUI	• Windows only • No SSH Terminal • No windowed apps	• Good for heavy transfers • Resume, bandwidth control, scheduling

HPC Software Environment

- Most packages installed as modules
- Modules dynamically load software package environments
- Commands:
  - *module avail* – list all installed software modules
  - *module avail gcc* – list installed versions of gcc compiler
  - *module load gcc (ml gcc)* – load default module for gcc compiler
  - *module load gcc/9.3.0 (ml gcc/9.3.0)* – load specific a version of gcc module
  - *module list (ml)* – display your currently loaded modules
  - *module unload gcc/9.3.0* – unload module
  - *module help gcc/9.3.0* – display help information
- How do I find my software?
  - *module avail*
- Request software install – email help-rcc@mcw.edu

SLURM Job Script Format

test-job.slurm

```bash
#!/bin/bash
#SBATCH --job-name=test-job
#SBATCH --ntasks=1
#SBATCH --mem-per-cpu=1gb
#SBATCH --time=00:01:00
#SBATCH --account=PI_NetID
#SBATCH --partition=partition
#SBATCH --output=%x-%j.out
#SBATCH --mail-type=ALL
#SBATCH --mail-user=NetID@mcw.edu

echo "Starting at $(date)"
echo "Job name: ${SLURM_JOB_NAME}, Job ID: ${SLURM_JOB_ID}"
echo "I have ${SLURM_CPUS_ON_NODE} CPUs on compute node $(hostname -s)"
```

SLURM Job Script Format

- In our YouTube channel you will be able to find a video explaining how to create a simple and an advanced SLURM job.

- We also have SLURM Guides in our documentation.

- The OpenComposer App in OnDemand will help you create a SLURM job.
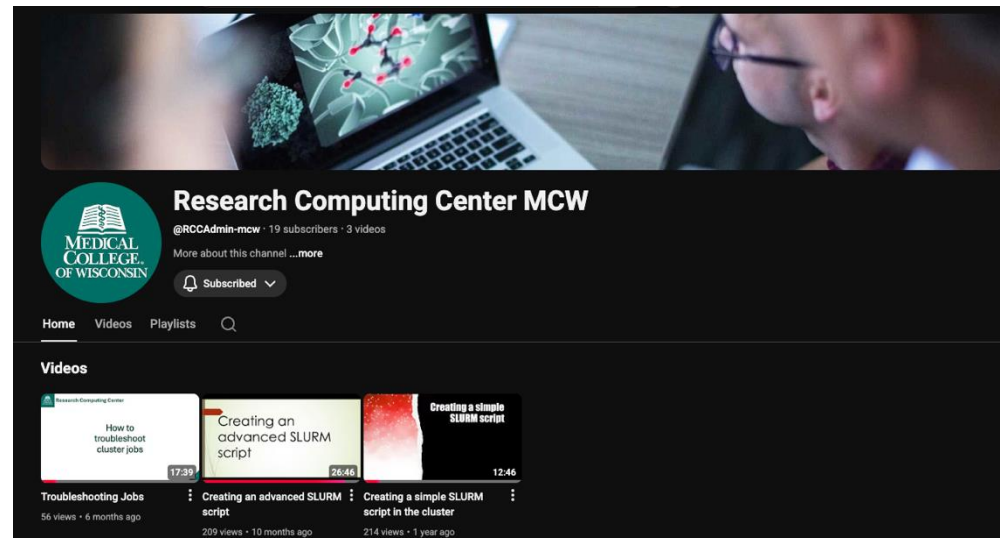
Submit a Job

- Most jobs that run on HPC are batch jobs
  - Submitted with the *sbatch* command and requires a job script
  - Best method for production job as it allows you to submit many jobs and let SLURM do the work
  - No requirement that you sit and watch the command-line
- Submit the job
  - *sbatch slurmScript*
- Check the job status
  - *squeue –u NetID*
- Cancel a job
  - *scancel jobID*
- Additional info: https://docs.rcc.mcw.edu/user-guide/jobs/running-jobs/
- The videos in our YouTube channel also explain how to submit jobs
- The Open Composer can also be used to submit jobs.

Monitoring and Troubleshooting Jobs

- A little time spent here dramatically increases success rate
- Tools
  - *squeue*
  - *sacct -j JobID*
  - *scontrol show job JobID (only works with running job)*
  - Output/error files
  - SSH to compute node and run "top" command
- Make sure your job is doing what you intended
- In our YouTube video [How to troubleshoot cluster jobs](#) you will find information on how to use the tools above and investigate failing jobs.
- You can also monitor jobs through the OpenComposer.

Getting Help

- Research Computing docs – https://docs.rcc.mcw.edu/
  - Help documentation specific to our systems
  - Subjects include accessing resources, running jobs, troubleshooting jobs, and software package
  - FAQ – https://docs.rcc.mcw.edu/faq/
- Man pages
  - Manual pages are included within the Linux operating system for many commands
  - *man ls*
- Forums
  - Stack Overflow
  - Stack Exchange
  - Many others
- Send a help request to help-rcc@mcw.edu

Getting Help

- Sending a help request to RCC
  - Contact help-rcc@mcw.edu
  - Include the following when applicable:
    - Your MCW NetID (username)
    - Machine name where the problem occurred (usually at the top of output file)
    - Job number of problem job (if applicable)
    - Name of problem software package (if applicable)
    - Brief explanation of what happened
    - Steps to reproduce the issue (if applicable)
    - Any fixes you've tried (if applicable
  - RCC may request further information
    - Please be timely in your response

Recommended

- Explore the docs – https://docs.rcc.mcw.edu/
- Work on being comfortable in the command line
  - https://www.codecademy.com/learn/learn-the-command-line
  - https://monicagiraldochica.github.io/bash_tutorial/
- Man pages
  - Linux includes manual pages for most common commands
  - Information about flags and functions of each command is presented in a uniform way
- Linux Forums – Google is your friend
- Don't copy paste everything ChatGPT recommends. Make sure it's correct for your specific situation.
- Visit our YouTube Channel

Supporting RCC

- Publications and Acknowledgement
  - For projects that have received support from the RCC, we appreciate an acknowledgement in your publication
  - Suggested:

    "This research was completed in part with computational resources and technical support provided by the Research Computing Center at the Medical College of Wisconsin."

  - We're excited when our work helps result in publications. Please send us an email to let us know!

Getting Started

- Docs – https://docs.rcc.mcw.edu/
- New Users – HPC Quick Start
  - Request an account
  - https://docs.rcc.mcw.edu/user-guide/quickstart/
  - https://docs.rcc.mcw.edu/user-guide/access/remote-access/
  - https://docs.rcc.mcw.edu/user-guide/jobs/running-jobs/
- Experienced Users – SLURM Guide
  - https://docs.rcc.mcw.edu/user-guide/jobs/running-jobs/
- Send a help request to help-rcc@mcw.edu