Topics

● **SHARCNET**

● Where to look for information and get help

● Essentials
  ○ What are available
  ○ How to connect to graham
  ○ How to manage disk storage
  ○ How to run your programs - *jobs*

● Common mistakes to avoid

● Q & A
What is SHARCNET?

A consortium of 19 Ontario institutions providing advanced computing resources and support...

- Member of Compute Canada and Compute Ontario
- 3,000+ Canadian and international users
- Almost 50,000 CPU cores
- 320 P100 GPUs
- 10 Gb/s network
Topics

- SHARCNET
- Where to look for information and get help
- Essentials
  - What are available
  - How to connect to graham
  - How to transfer files
  - How to compile programs
  - How to submit jobs
  - Manage files
- Do’s and don’t do’s
- Q & A
Where to look for information

help@sharcnet.cat or support@computecanada.ca
Getting Help:

SHARCNET web siteL https://www.sharcnet.ca/

- **Documentation and training materials:**
  - Help pages, tutorials: Support > Wiki
  - SHARCNET’s youtube channel: youtube.sharcnet.ca

- **Ticketing system**
  - Send an email to help@sharcnet.ca
Getting Help: Compute Canada site

Compute Canada web site (docs.computecanada.ca)

- How-to guides
- Systems and services
- Discipline guides
- Regional partners
- Compute Canada’s problem tracking system
  - Email to support@computecanada.ca
  - Email to help@sharcnet.ca
Topics

- SHARCNET
- Where to look for information and get help
- **Essentials**
  - What are available
  - How to connect to graham
  - How to manage disk storage
  - How to run your programs - *jobs*
- Common mistakes to avoid
- Q & A
Essentials: Computing Environment

- **Systems**
  - Clusters, Cloud facilities

- **Operating Systems**
  - Linux (64-bit CentOS)

- **Languages**
  - C/C++, Fortran, Matlab/Octave, Python, R, Java, etc.

- **Key Parallel Development Support**
  - MPI, pthreads, OpenMP, CUDA, OpenACC, OpenCL

- **Software Modules**
  - select pre-built and configured software, as well as versions, with the `module` command

- **Batch Scheduling**
  - SLURM scheduler
Essentials: Graham cluster

- Number of CPU cores: 33,448
- Number of nodes: 1043
- 32 cpu cores per node
- Between 128 and 3072 GB of RAM per node
- Number of NVIDIA P100 GPUs: 320
- Networking: EDR (cpu nodes) and FDR (GPU and cloud nodes)
  InfiniBand
Essentials: Access to SHARCNET

Connecting to clusters

- All systems are only accessible via secure shell (ssh), using your Compute Canada credentials
  
  ```
  $ ssh user@graham.computecanada.ca
  ```
  
- We recommend authenticating using an ssh key agent. See the SSH page in our help wiki for details

Connection and file transfer programs

- Unix / Mac
  - scp or sftp, rsync

- Windows
  - MobaXterm
  - Cygwin (a full Unix-like suite)
  - Windows Subsystem for Linux

- Any OS (from a browser)
  - Globus
## Essentials: File systems

<table>
<thead>
<tr>
<th>File system</th>
<th>Quotas</th>
<th>Backed up?</th>
<th>Purged?</th>
<th>Available by Default?</th>
<th>Mounted on Compute Nodes?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home Space /home</td>
<td>50 GB and 0.5M files per user</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Scratch Space /scratch</td>
<td>20 TB and 1M files per user, can request increase to 100 TB</td>
<td>No</td>
<td>Yes, all files older than 60 days</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Project Space /project</td>
<td>1 TB and 0.5M files per group, can request increase to 10 TB</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Nearline Space</td>
<td>5 TB per group</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

- Run `quota` command on Graham/Cedar to find out if you are approaching or over the disk quota.
Essentials: Running programs with SLURM

- Jobs are submitted using the `sbatch` command with a script, e.g. `run_job.sh`, containing

```bash
#!/bin/bash
#SBATCH --time=00:05    # Run time limit (DD-HH:MM)
#SBATCH --account=def-user
#SBATCH --ntasks=32     # Number of MPI processes, default 1
#SBATCH --cpus-per-task=32  # Normally defined for threaded jobs
#SBATCH --gres=gpu:2      # request GPU "generic resource", 4 on Cedar, 2 on Graham
#SBATCH --mem=1024M      # memory; default unit is megabytes
#SBATCH --mem-per-cpu=1024M  # Optional, for user’s reference
#SBATCH --job-name=hello  # You give any name
#SBATCH --output=%x-%J.log
./myprog                # Replace with `mpiexec ./myprog` or `srun ./myprog` for MPI jobs
```

- `squeue`: to list the status of submitted jobs.
- `sacct`: to show details of recent jobs.
- `scancel`: to kill jobs.
Why is my job not starting?

- There may be multiple reasons
- Graham/Cedar are very busy clusters, with ~15% of the cycles available to non-RAC jobs. **Tip: consider applying for RAC.**
- Requesting much more resources (runtime, CPU cores, memory) than what is actually needed will result in a longer queue wait time, for no good reason. **Tip: request only what the job needs, with a bit of leeway.**
- If your job uses multiples of 32 cpu cores, sometimes the queue wait time can be much shorter if you do a by-node reservation, instead of the default by-core one. **Tip: use --nodes=N and --ntasks-per-node=32 sbatch arguments to request the by-node reservation.**
Topics

● SHARCNET
● Where to look for information and get help
● Essentials
  ○ What are available
  ○ How to connect to graham
  ○ How to manage disk storage
  ○ How to run your programs - *jobs*
● Common mistakes to avoid
● Q & A
Common mistakes to avoid

- Do not run significant programs on login nodes, nor run programs directly on compute nodes.
- Do not specify a maximum job run time blindly (say, 7 days), or more memory than required for your program
  - pick an appropriate value, eg. 130% of the measured/expected run time or memory per processor
- Do not create millions of tiny files, or large amounts (> GB) of uncompressed (eg. ASCII) output
  - aggregate files with tar, use binary or compressed file formats
Common mistakes to avoid

● Do not run “watch squeue”. If it is required please add a delay of at least 60 seconds between updates, with “watch -n 60 squeue”.
● Do not submit more than 1 job per second to the scheduler. Add a delay of at least 1 second between each job submission.
● Do not submit very short jobs (less than 10 minutes). In this case combine tasks into longer jobs.
● If you have many jobs to submit, consider using alternatives
  ○ Check “Serial farming on Graham” webinar on youtube.sharcnet.ca