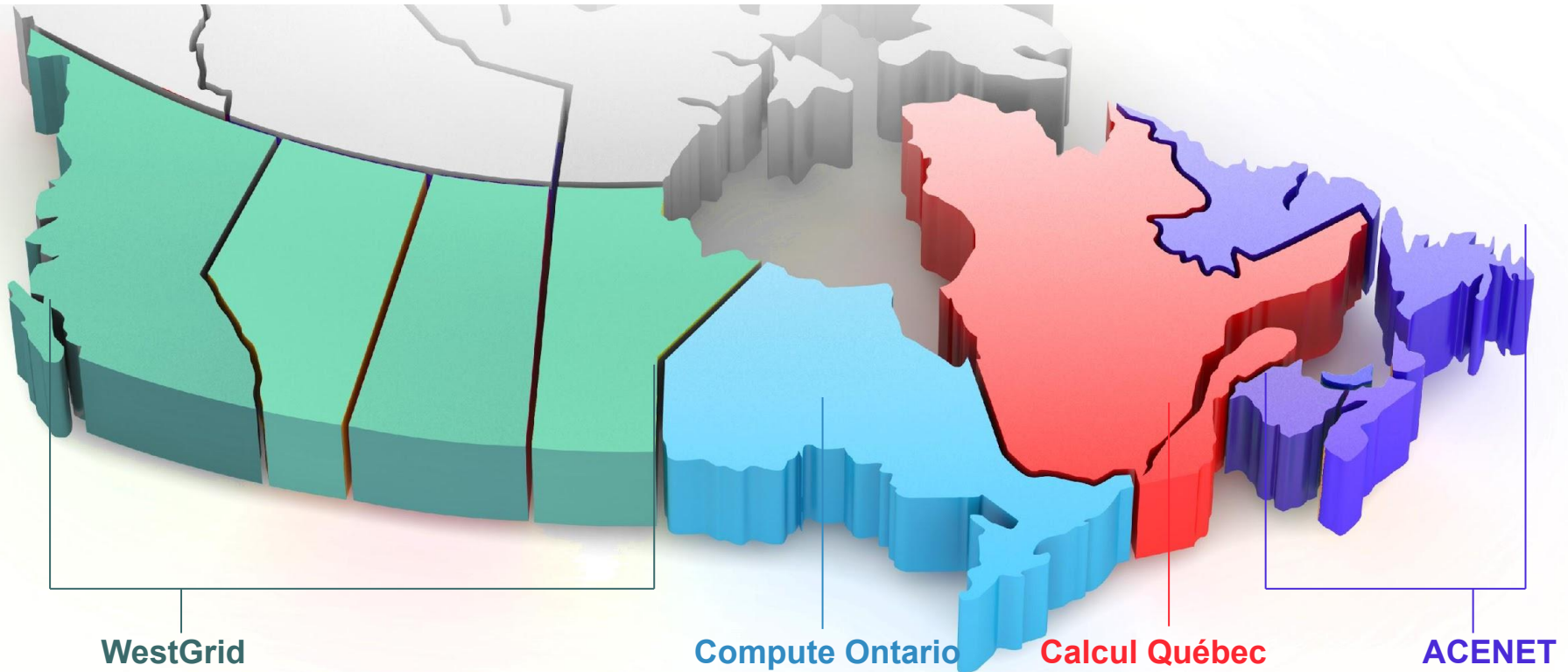


New User Seminar



Single account...

computecanada



*One can access all national supercomputers
across the country, for free.*

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

computecanada

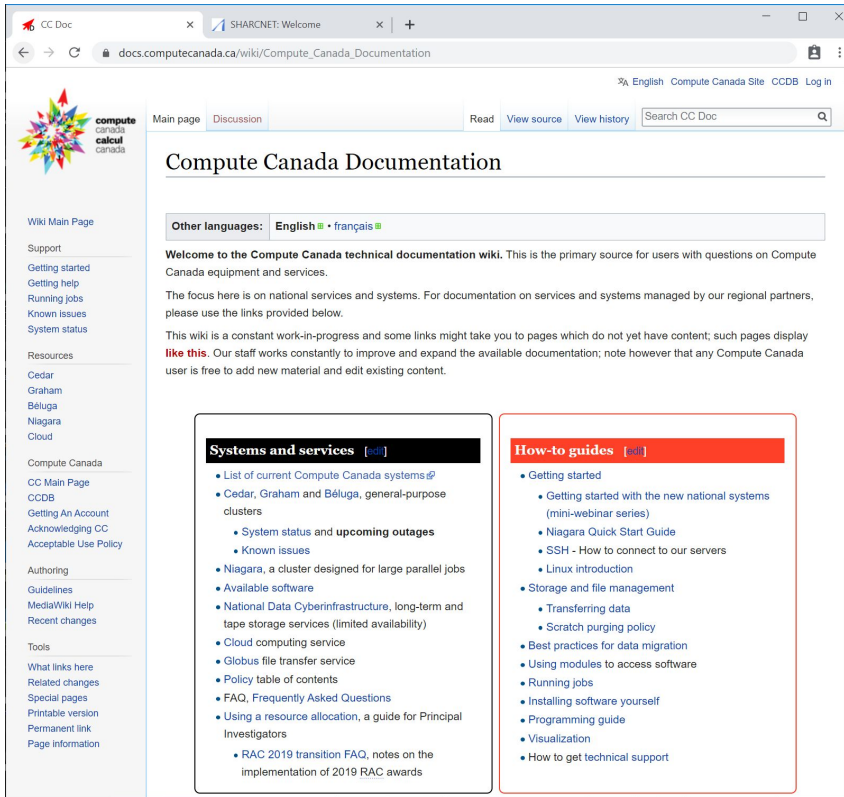
Shared
Hierarchical
Academic
Research
Computing
NETwork



- Member of Compute Canada and Compute Ontario
- 3,000+ Canadian and international users
- ~50,000 CPU cores
- 370+ GPUs
- 10 Gb/s network
- 100 Gb/s between nat'l centres

Where to look for information and get help

Online documentations



The screenshot shows the Compute Canada Documentation website. The header includes the Compute Canada logo and navigation links like 'Main page', 'Discussion', 'Read', 'View source', and 'View history'. A search bar is present. The main content area is titled 'Compute Canada Documentation' and includes a welcome message and a list of 'Systems and services' and 'How-to guides'. The left sidebar contains a 'Wiki Main Page' and various links for support, resources, and tools.

Compute Canada Documentation

Other languages: English • français

Welcome to the Compute Canada technical documentation wiki. This is the primary source for users with questions on Compute Canada equipment and services.

The focus here is on national services and systems. For documentation on services and systems managed by our regional partners, please use the links provided below.

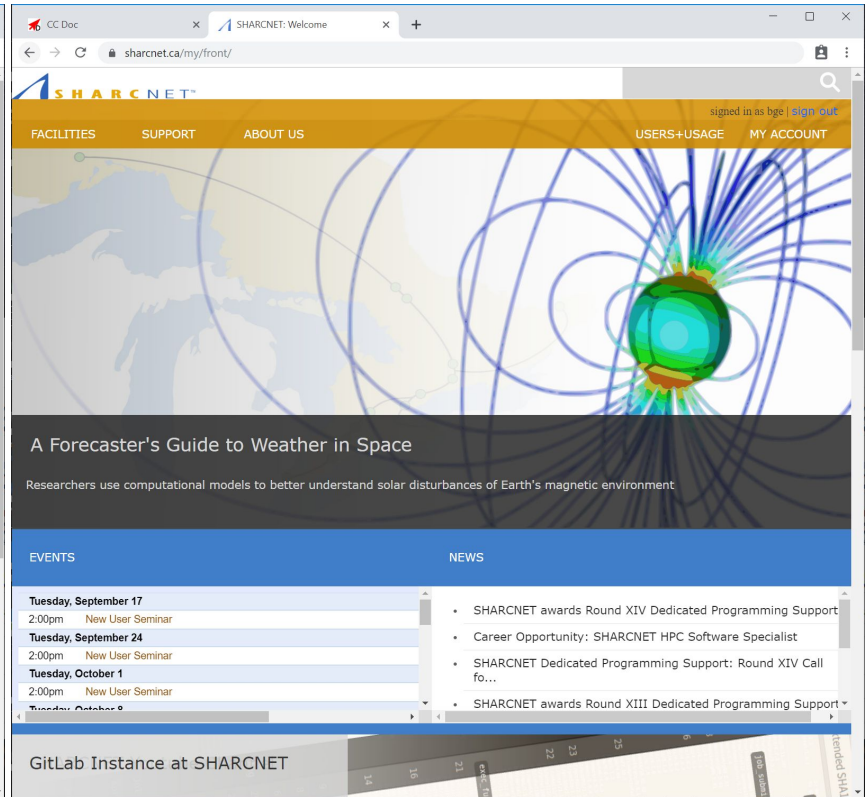
This wiki is a constant work-in-progress and some links might take you to pages which do not yet have content; such pages display **like this**. Our staff works constantly to improve and expand the available documentation; note however that any Compute Canada user is free to add new material and edit existing content.

Systems and services

- List of current Compute Canada systems
- Cedar, Graham and Béluga, general-purpose clusters
 - System status and upcoming outages
 - Known issues
- Niagara, a cluster designed for large parallel jobs
- Available software
- National Data Cyberinfrastructure, long-term and tape storage services (limited availability)
- Cloud computing service
- Globus file transfer service
- Policy table of contents
- FAQ, Frequently Asked Questions
- Using a resource allocation, a guide for Principal Investigators
 - RAC 2019 transition FAQ, notes on the implementation of 2019 RAC awards

How-to guides

- Getting started
 - Getting started with the new national systems (mini-webinar series)
 - Niagara Quick Start Guide
 - SSH - How to connect to our servers
 - Linux introduction
- Storage and file management
 - Transferring data
 - Scratch purging policy
- Best practices for data migration
- Using modules to access software
- Running jobs
- Installing software yourself
- Programming guide
- Visualization
- How to get technical support



The screenshot shows the SHARCNET website. The header includes the SHARCNET logo and navigation links like 'FACILITIES', 'SUPPORT', 'ABOUT US', 'USERS+USAGE', and 'MY ACCOUNT'. A search bar is present. The main content area features a large image of a globe with magnetic field lines and the title 'A Forecaster's Guide to Weather in Space'. Below this, there are sections for 'EVENTS' and 'NEWS'.

SHARCNET

A Forecaster's Guide to Weather in Space

Researchers use computational models to better understand solar disturbances of Earth's magnetic environment

EVENTS

- Tuesday, September 17
 - 2:00pm New User Seminar
- Tuesday, September 24
 - 2:00pm New User Seminar
- Tuesday, October 1
 - 2:00pm New User Seminar

NEWS

- SHARCNET awards Round XIV Dedicated Programming Support
- Career Opportunity: SHARCNET HPC Software Specialist
- SHARCNET Dedicated Programming Support: Round XIV Call for...
- SHARCNET awards Round XIII Dedicated Programming Support

GitLab Instance at SHARCNET

<https://docs.compute canada.ca/>

<https://www.sharcnet.ca/>


Where to look for information and get help

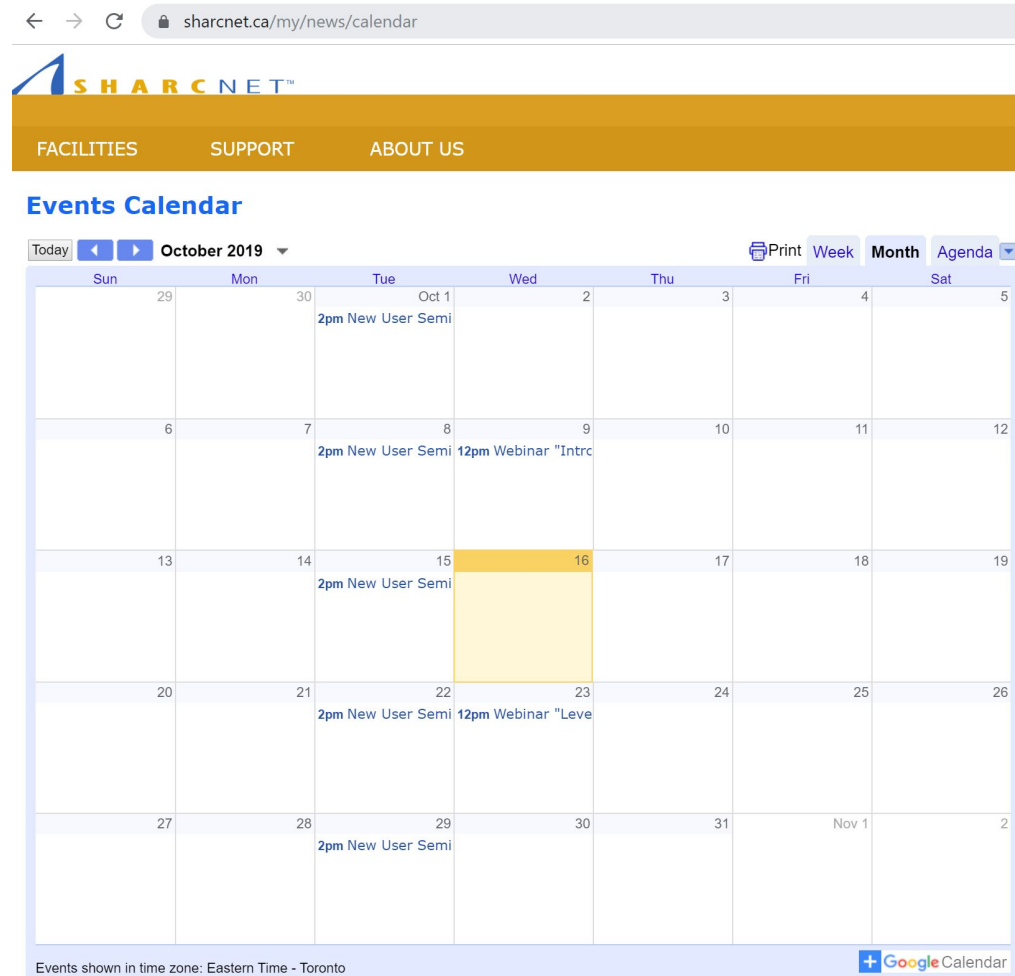
Youtube videos - search for SHARCNET channel 

- <http://youtube.sharcnet.ca>
- Two-minute short videos on a variety of simple topics
- Programming clusters with/without MPI
- Parallel programming using threads
- Programming GPUs
- Understanding how scheduler works
- Visualization of scientific data
- Tools for bioinformatics
- Tools and platforms for data science
- Popular programming languages such as C/C++, Fortran, Matlab/Octave, R, Python, Julia, CUDA, etc.

Where to look for information and get help

Online events

- New user seminar every Tuesday at 2pm.
- Bi-weekly general interest seminars at noon on Wednesday.
-  @SHARCNET



The screenshot shows the SHARCNET website's Events Calendar for October 2019. The calendar is a grid view with days of the week as columns and dates as rows. Events are listed in the cells for specific dates. The events are:

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	Oct 1 2pm New User Semi	2	3	4	5
6	7	8 2pm New User Semi	9 12pm Webinar "Intrc"	10	11	12
13	14	15 2pm New User Semi	16	17	18	19
20	21	22 2pm New User Semi	23 12pm Webinar "Leve"	24	25	26
27	28	29 2pm New User Semi	30	31	Nov 1	2

At the bottom of the calendar, it says "Events shown in time zone: Eastern Time - Toronto". The Google Calendar logo is in the bottom right corner.

Where to look for information and get help

Interactive help

- Ticketing system via support@computeCanada.ca or help@sharcnet.ca
- E-mail us - check staff contact info on <https://www.sharcnet.ca/>
- Phone us
- Office visit

Use of systems

Installation of software

Access to commercial software and site licence

Debugging and optimizing code

Programming

Consultation on various research problems

Grant application for compute hardware

...

The computing environment

Clusters across the country

- cedar.computecanada.ca
- graham.sharcnet.ca
- niagara.computecanada.ca
- beluga.computecanada.ca

Cloud services

- arbutus.cloud.computecanada.ca
- graham.cloud.comptecanada.ca
- east.cloud.computecanada.ca

NB. Files are NOT shared across systems. You need move and copy files around as needed.

The computing environment

Cluster computing environment

- OS: 64-bit Linux - CentOS
- Languages: C/C++, Fortran, R, Python, Matlab/Octave, Java, Julia, CUDA, etc.
- Access to a variety of software packages
- Parallel development support:
 - **MPI**, **OpenMP**, Pthreads, **CUDA**, OpenACC, OpenCL, DDT
 - **C++**: Language support for multithreading (since C++-11 standard)
 - **Fortran**: Language support for parallel programming (since 2003 standard)
 - **Julia**: Parallel processing constructs, shared and distributed objects
- Data science support:
 - R, Python, Julia, Spark, DASK, etc.
- Batch computing via slurm

Connecting to clusters via SSH

```
bge@DESKTOP-QFKH8Q4: ~  
bge@DESKTOP-QFKH8Q4:~$  
bge@DESKTOP-QFKH8Q4:~$  
bge@DESKTOP-QFKH8Q4:~$ ssh graham.sharcnet.ca -l bge
```

For Windows users we recommend

Windows Subsystem for Linux

```
MobaXterm  
Terminal Sessions View X server Tools Games Settings Macros Help  
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help X server Exit  
Quick connect... 2. /home/mobaxterm  
• MobaXterm Personal Edition v10.6 •  
(X server, SSH client and network tools)  
• Your computer drives are accessible through the /drives path  
• Your DISPLAY is set to 10.0.2.15:0.0  
• When using SSH, your remote DISPLAY is automatically forwarded  
• Your HOME folder is not persistent: it will be erased on restart  
• Each command status is specified by a special symbol (✓ or ✗)  
• Important:  
This is MobaXterm Personal Edition. The Professional edition  
allows you to customize MobaXterm for your company: you can add  
your own logo, your parameters, your welcome message and generate  
either an MSI installation package or a portable executable.  
We can also modify MobaXterm or develop the plugins you need.  
For more information: https://mobaxterm.mobatek.net/download.html  
[2020-03-25 11:25.59] ~  
[bge.DESKTOP-QFKH8Q4] > ssh graham.sharcnet.ca -l bge
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

MobaXterm for Windows

Connecting to graham virtual desktop via VNC

The screenshot displays a VNC session titled 'x11 - TigerVNC'. The desktop environment includes a taskbar with 'Applications', 'Places', and 'System' menus. A 'Computer' icon is visible on the left. The main window is 'MATLAB R2019b - academic use', showing the 'HOME' tab with various toolbars and a 'Command Window' displaying the command `= rand(10,10)`. A 'Workspace' window shows a variable 'a' with a value of '10x10 double'. A 'Firefox' browser window is open, displaying the 'CC Doc' page for 'Compute Canada Documentation' at https://docs.computeCanada.ca/wiki/Compute_Canada_Documentation. A 'Terminal' window is open, showing the command prompt `bge@gra-vdi4:~` and the output of the `module load nixpkgs` and `module load matlab/2019b` commands. A 'VNC Viewer: Connection Details' dialog box is overlaid, showing the 'VNC server' field with the address `gra-vdi.computeCanada.ca` and buttons for 'Options...', 'Load...', 'Save As...', 'About...', 'Cancel', and 'Connect'.

Terminal output:

```
bge@gra-vdi4:~  
with traditional programming languages such as C, C++, and Fortran.  
Properties:  
Tools for development / Outils de développement  
You will need to load all module(s) on any one of the lines below before the  
"matlab/2019b" module is available to load.  
nixpkgs/16.09  
Help:  
Description  
=====  
MATLAB is a high-level language and interactive environment  
that enables you to perform computationally intensive tasks faster than w  
[bge@gra-vdi4 ~]$ module load nixpkgs  
[bge@gra-vdi4 ~]$ module load matlab/2019b  
[bge@gra-vdi4 ~]$ matlab  
MATLAB is selecting SOFTWARE_OPENGL rendering.  
Opening log file: /tmp/java.log.57058  
GLib-GIO-Message: 10:54:07.700: Using the 'memory' GSettings backend. Your sett  
ings will not be saved or shared with other applications.
```

Command Window output:

```
= rand(10,10)  
Columns 1 through 7  
0.2760 0.7513  
0.6797 0.2551  
0.6551 0.5060  
0.1626 0.6991  
0.1190 0.8909  
0.4984 0.9593  
0.9597 0.5472  
0.3404 0.1386  
0.5853 0.1493  
0.2238 0.2575
```


Accessing and managing files

File systems

- **/home**: 50G, 0.5m files; backed up regularly
- **/project**: 1T per group, 0.5m files, upto 10T per group; backed up.
- **/scratch**: 20T per user, 1m files, upto 100T; 2 months of life.
- **nearline**: to store files not currently in use, but may be needed later. 5T per group. NOT available on compute nodes.

*NB: Please **DO NOT** store everything, remove the files no longer in use to save space.*

How to access

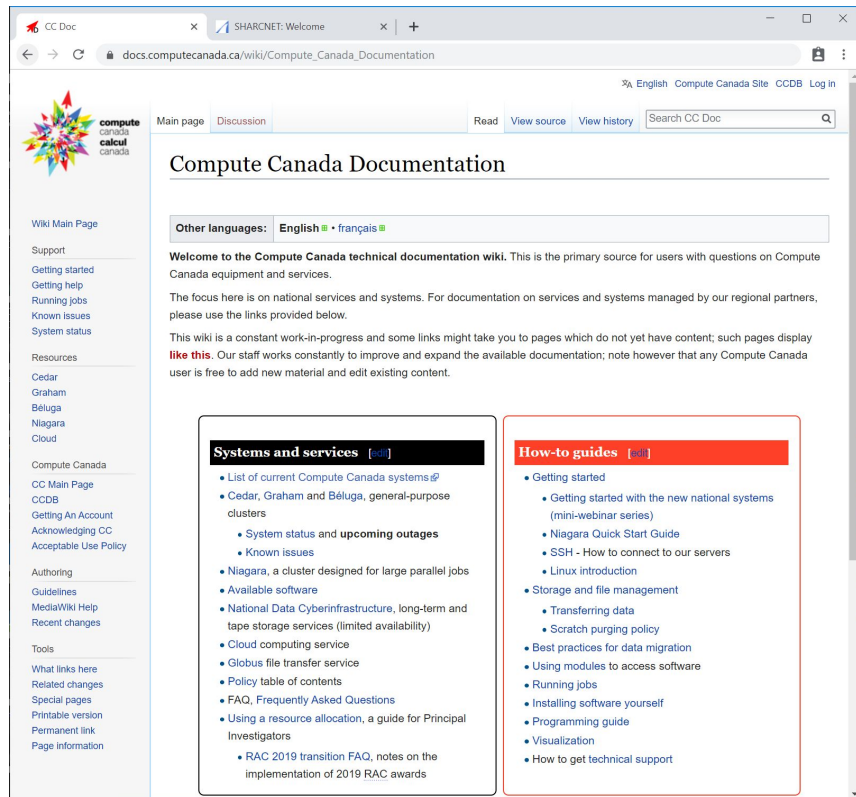
- SSH connect to a cluster and access files there.
- Mount remote cluster file systems on your laptop, desktop and access files as if they were local. See Youtube videos on how to mount cluster file system with SSHFS.  YouTube^{CA}

Moving files to and from

- Via SSHFS mounted drives (as if local)  YouTube^{CA}
- scp, sftp.
- globus via web browser.

What software packages are available?

Check for software on the web site



Check for software while on a cluster

\$ module avail

\$ module spider *keyword*

Running jobs using a slurm script - *myjob.sh*

Submitting a serial job

```
#!/bin/bash
#SBATCH --time=01:00 # DD-HH:MM
#SBATCH --account=def-user

R --no-save --args input 23 output < sample.R
```

To see what account groups you have access to, use command **sshare -U** or **salloc** by itself

sbatch *myjob.sh*

Submitting a series of jobs

```
#!/bin/bash
#SBATCH --time=01:00
#SBATCH --account=def-user
#SBATCH --array=1-200
```

```
R --no-save --args input $SLURM_ARRAY_TASK_ID \
output < sample.R
```

In this example, slurm generate 200 integers from 1 to 200 and stores in the environment variable `$SLURM_ARRAY_TASK_ID` one at a time.

The R script `sample.R` reads 3 args: “input”, an integer *N* ranging from 1 to 200 and “output”. It reads input from **input*N*.csv** and writes results to output **output*N*.csv** for each of the 200 cases.

Running jobs using a slurm script - *myjob.sh*

Submitting a threaded job

```
#!/bin/bash
#SBATCH --account=def-user
#SBATCH --time=0-03:00
#SBATCH --cpus-per-task=32
#SBATCH --ntasks=1
#SBATCH --mem=20G

export \
  OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK

./myprog.exe
```

sbatch *myjob.sh*

Submitting a parallel job

```
#!/bin/bash
#SBATCH --account=def-user
#SBATCH --time=5-00:00
#SBATCH --ntasks=32
#SBATCH --mem-per-cpu=4G

srun ./mympprog.exe
```

Why my jobs don't start

- Your jobs are waiting because of past usage. Everyone has a share. The share usage is completely "reset" in 2 weeks but after 1 week the usage is halved so some of your jobs may start then.
- 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 for time and memory.***
- You may try other clusters such as beluga, cedar and niagara.
- If your job can efficiently use multiples of 32 cpu cores, it gains access to a larger set of nodes, so may wait less than a by-core job. ***Tip: use --nodes=N and --ntasks-per-node=32 sbatch arguments to request by-node allocation.***

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

Q&A