High Performance, Advanced, and Cloud Research Computing for Windsor Researchers

Paul Preney, M.Sc, B.Ed., B.Sc.
Email: preney@sharcnet.ca

University of Windsor, Office of Research and Integrity Services (ORIS)
SHARCNET / Compute Ontario / Digital Research Alliance of Canada

Support Email: support@tech.alliancecan.ca

Oct. 3, 2025

Land Acknowledgement

I am located at the University of Windsor.

The University of Windsor sits on the traditional territory of the **Three Fires Confederacy of First Nations**, which includes the **Ojibwa**, the **Odawa**, and the **Potawatomi**. We respect the longstanding relationships with First Nations people in this 100-mile Windsor-Essex peninsula region and the straits —les détroits— of Detroit.

Abstract

The University of Windsor is a founding member of **SHARCNET**, now a consortium of approximately 20 Ontario academic institutions, which provide high performance computing clusters, storage, and cloud resources and services to its researchers. SHARCNET is also part of Compute Ontario which is part of the **Digital Research Alliance of Canada (formerly Compute Canada)**. Collectively these organizations work together to seamlessly provide Canadian researchers access to **compute clusters**, **storage**, and **clouds** to do high-performance, advanced, and cloud computing research. This presentation will introduce and explain these facilities and services, how such can be **accessed and used** by researchers and their students. This presentation will also discuss the **Resource Allocation Competition (RAC)** (this year's applications are due by Nov. 4).

Outline

Academic Research Computing at UWindsor

Campus IT:

- ITS (Information Technology Services): `https://www.uwindsor.ca/itservices/`
- Excepting what is in an individual lab or what is negotiated with ITS, making use of ITS **only-used-on-campus** storage is appropriate when using *under* 1 TB (approx.) of storage without (currently) any significant CPU/GPU computational needs associated with that data.

SHARCNET

https://www.sharcnet.ca

SHARCNET:

- **S**hared **H**ierarchical **A**cademic **R**esearch **C**omputing **Net**work
- provides free access to high performance, advanced, and cloud computing and corresponding **compute, cloud, and storage** resources and services to researchers
- biweekly webinars, self-paced courses, summer school courses, and more

SHARCNET (cont.)

- Started in 2001.
- Founding members:
  - Fanshawe College, University of Guelph, McMaster University, Sheridan College, Western University, Wilfred Laurier University, University of Windsor
- Today's members include these institutions:
  - … Brock, Conestoga, Durham, Fields Institute, Lakehead, Laurentian, Nipissing, Ontario College of Art and Design University (OCAD), Ontario Tech University, Perimeter Institute, Trent, Waterloo, York
- Part of *Compute Ontario* which is part of the *Digital Research Alliance of Canada*.

SHARCNET (cont.)

SHARCNET provides:

- compute (clusters)
- clouds
- storage
- Research Data Management (RDM) resources
- training on various advanced research computing, data science, machine learning topics

Compute Ontario

Compute Ontario

https://computeontario.ca

Compute Ontario:

- Plays a key role in coordinating Ontario's advanced research computing and big data focus
- CO's partners include:
  - Centre for Advanced Computing (CAC)
  - HPC4Health
  - SciNet
  - SHARCNET

Digital Research Alliance of Canada

Digital Research Alliance of Canada

https://www.alliancecan.ca

Alliance regional partners include:

- ACENET for New Brunswick, Nova Scotia, P.E.I., Newfoundland and Labrador
- Calcul Québec (CQ) for Québec
- Compute Ontario (CO) for Ontario
- Prairies DRI (P-DRI) for Alberta, Saskatchewan, and Manitoba
- BC DRI (BC-DRI) for British Columbia

When Should Our Resources/Services Be Used

Consider using our resources and services when:

- computing, cloud, and storage resources are **needed** that **exceed what one has available** at his/her school

Who uses our resources and services across Canada?

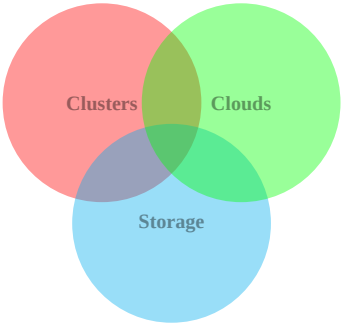
- **graduate students** and **postdocs**
- **faculty** and other researchers at various institutions

Our Resources

A supercomputer is:

- a **networking** of many **computers** in a **single** location that
- are able to function as a single computer or as many individual/groupings of computers
- lots of **CPU cores**
- lots of **GPU cores** (where available)
- lots of **RAM**
- lots of **disk space**

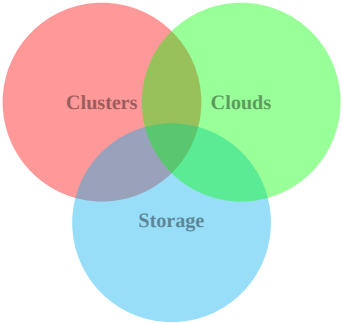
Available Resources



HPC clusters:

- have lower latencies, higher bandwidths, and more CPU, GPU, RAM, and storage resources than typical consumer devices
- can run large and exascale simulations that need dedicated access to dozens/hundreds/thousands of CPU and GPU cores
- jobs run with the resources requested

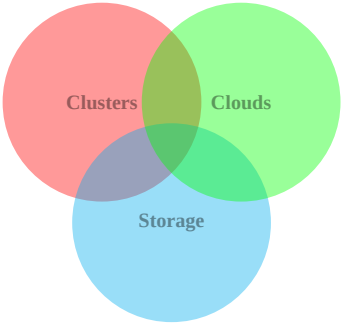
Available Resources (cont.)



Clouds:

- e.g., to run web services (24/7/365)
- CPU and GPU resources are usually over-committed (shared)
  - This can be discussed with cloud admins when the cloud instance is being set up.

Available Resources (cont.)



Storage:

- e.g., to store terrabytes of research data

Available Resources (cont.)

Compute Cluster	Purpose	Managed By	Location
fir.alliancecan.ca	general	BC-DRI	Simon Fraser University (British Columbia)
killarney.alliancecan.ca	PAICE	Vector Institute, SciNet	University of Toronto (Ontario)
narval.alliancecan.ca	general	CQ	École de technologie supérieure (Québec)
nibi.alliancecan.ca	general	SHARCNET	University of Waterloo (Ontario)
rorqual.alliancecan.ca	general	CQ	École de technologie supérieure (Québec)
tamia.alliancecan.ca	PAICE	Mila, CQ	Université Laval (Québec)
trillium.alliancecan.ca	large parallel	SciNet	University of Toronto (Ontario)
vulcan.alliancecan.ca	PAICE	P-DRI	University of Alberta (Alberta)

- The main clusters are fir, narval, nibi, and trillium.
- PAICE: *Pan-Canadian AI Compute Environment* which are specialized GPU clusters

Available Resources (cont.)

Compute cluster hardware summary (approximate):

Resource	Fir	Killarney	Narval	Nibi	Rorqual	Tamia	Trillium	Vulcan
Purpose	General	PAICE	General	General	General	PAICE	LP	PAICE
Nodes	1,032	178	1,340	752	767	46	1,224	205
RAM (TB)	856	104	433	647	560	23	949	102.5
CPU Cores	175,104	11,232	83,216	140,928	136,896	2,272	241,056	13,120
CPU Kinds	EPYC	Intel	EPYC	Intel, MI300A	EPYC, Intel	Intel	EPYC	Intel
Physical GPUs	640	752	636	312	324	168	252	820
GPU Kinds	H100	L40S, H100	A100	H100, MI300A	H100	H100	H100	L40S
Storage (PB)	51	1.7	41	25	68.6	0.4	29	5

- LP: “Large Parallel”
- Killarney: 10 nodes have 8xH100 GPUs and each of these nodes has 48 CPU cores
- Nibi: 6 nodes have 4xMI300A GPUs and each of these nodes has 96 CPU cores
- **NOTE:** For interactive visualization use, L40S is (likely) better than H100.

Available Resources (cont.)

Cloud	Managed By	Location
arbutus.cloud.alliancecan.ca	BC-DRI	University of Victoria (British Columbia)
beluga.cloud.alliancecan.ca	CQ	École de technologie supérieure (Québec)
cedar.cloud.alliancecan.ca	BC-DRI	Simon Fraser University (British Columbia)
nibi.cloud.alliancecan.ca	SHARCNET	University of Waterloo (Ontario)

NOTE: There is no “default” cloud acount: one needs to first submit a ticket and work with cloud admins to set their cloud instance up.

Available Resources (cont.)

Approximate compute cluster storage amounts:

Space	Initial Quota	Backed Up	Description
`/home`	50 GB	Yes	personal data
`/project`	1 TB	Yes	research group data
`/scratch`	20 TB	No	files deleted after 60d
nearline	1+ TB	n/a	offline group storage

- Actual nearline quota differs by cluster.
- PAICE systems have different initial quota amounts and backup policies.

Accessing Resources

- Cost:
  - Access is **free** to research faculty at Canadian academic research institutions.
  - We are funded directly and indirectly through various federal and provincial grants.
- Limited to **research**, e.g., undergraduate course work is not permitted.
- Faculty members (PIs) **sponsor** some number of users, e.g., their students.
- Compute jobs are associated with a **sponsor**.
- Users can have more than one **sponsor**.
- All accounts must be renewed **every year** to remain active.

Accessing Resources (cont.)

IMPORTANT

- If you already have/had a CCDB account, a new one will not be made so don't apply for a new account!
- Instead **to re-activate an inactive account** log in to the CCDB web site, i.e., `https://ccdb.alliancecan.ca/`, and **apply for a new role**.

Accessing Resources (cont.)

To obtain an account, apply for it by following these instructions:

1. Go to `https://docs.alliancecan.ca`.
2. Click on **Getting an account** in the left-hand side menu which will take you to:
   - `https://docs.alliancecan.ca/wiki/Apply_for_a_CCDB_account`

Accessing Resources (cont.)

If you are doing work under a sponsor…

- It is very important that your sponsor **first** acquires/activates his/her account **before** you can apply for your own account.
- Ask your sponsor for his/her **CCRI number** before you apply for your own account.
- If your sponsor's account becomes inactive and that is your only sponsor then your account is implicitly deactivated (even if you activated your account).
- There is only one sponsor per research group, but, a sponsor is also able to be sponsored and therefore can run jobs under the other sponsor's account.

Accessing Resources (cont.)

Multi-factor Authentication (MFA)

- To access most resources, you **must** have multi-factor authentication (MFA) configured in your CCDB account.
- For more information, see: `https://docs.alliancecan.ca/wiki/Multifactor_authentication`

Accessing Resources (cont.)

Compute cluster login access using a web browser:

- Fir: `https://jupyterhub.fir.alliancecan.ca/`
- Narval: `https://jupyterhub.narval.alliancecan.ca/`
- Nibi: `https://ondemand.sharcnet.ca`
- Rorqual: `https://jupyterhub.rorqual.alliancecan.ca/`
- Trillium: `https://ondemand.scinet.utoronto.ca`

NOTE: Command line login node and graphical desktop access can be done using only your web browser.

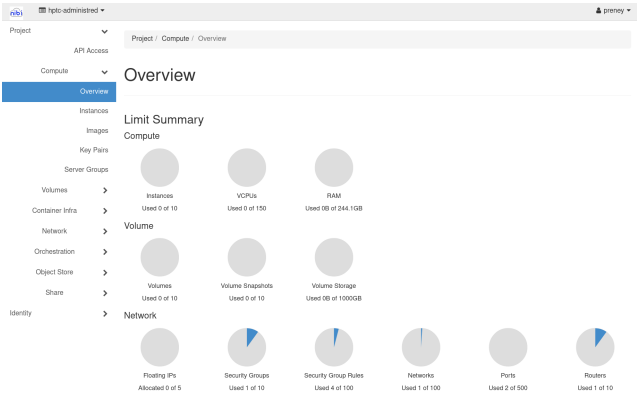
Accessing Resources (cont.)

Compute cluster login access using SSH/SFTP:

- Fir: `fir.alliancecan.ca`
- Killarney: `killarney.alliancecan.ca`
- Narval: `narval.alliancecan.ca`
- Nibi: `nibi.alliancecan.ca`
- Rorqual: `rorqual.alliancecan.ca`
- Tamia: `tamia.alliancecan.ca`
- Trillium: `trillium.alliancecan.ca`
- Vulcan: `vulcan.alliancecan.ca`

NOTE: If your computer/site blocks SSH then you will not be able to access these resources with SSH. Use a web portal instead to access such (directly or indirectly).

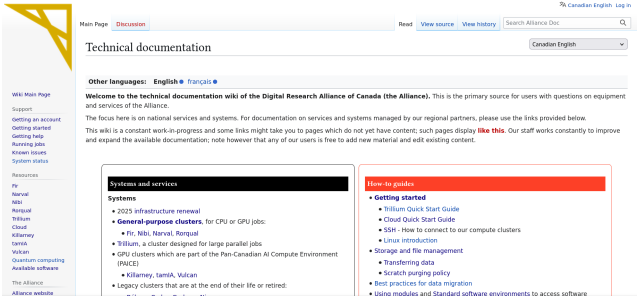
Accessing Resources (cont.)



Our cloud node management portals use OpenStack.

See our “quick start” documentation at `https://docs.alliancecan.ca/wiki/Cloud_Quick_Start`.

Documentation Resources



We have useful technical documentation in our wiki at `https://docs.alliancecan.ca`.

Documentation Resources (cont.)

SHARCNET has:

- Online training materials and courses at `https://training.sharcnet.ca`.

- YouTube channel videos at `https://youtube.sharcnet.ca`.

Documentation Resources (cont.)

Compute Ontario has:

Past summer school courses at `https://training.computeontario.ca`.

NOTE: In summer 2025, ACENET participated in the Compute Ontario Summer School event.

Some webinar videos can be found at `https://www.computeontario.ca/training-colloquia`.

Documentation Resources (cont.)

Online user-specific job portals:

- Narval: `https://portail.narval.calculquebec.ca/`
- Nibi: `https://portal.nibi.sharcnet.ca/`
- Rorqual: `https://metrix.rorqual.alliancecan.ca/`
- Tamia: `https://portail.tamia.ecpia.ca/`
- Trillium: `https://my.scinet.utoronto.ca`
- Vulcan: `https://portal.vulcan.alliancecan.ca/`

Documentation Resources (cont.)

Globus (file transfer) end-points:

- Fir: `alliancecan##fir-globus`
- Narval: `Compute Canada - Narval`
- Nibi: `alliancecan##nibi`
- Rorqual: `alliancecan##rorqual`
- Trillium: `alliancecan##trillium`
- Tamia: `TamIA's Globus v5 Server`
- Vulcan: `Vulcan Globus v5`

Support Resources

Support through tickets.

To open a ticket send an email to:
`mailto:support@tech.alliancecan.ca`.

Support from expert staff via:

- ticketing system
- video conferencing
- direct email (ticketing system is preferred)
- in-person (if at the same institution)
- telephone (if needed)

Compute Cluster: Environment

- Operating System: 64-bit Linux
- Supported programming languages include:
  - C, C++, Fortran, Java, Julia, MATLAB, Octave, Python, R, etc.
- A large variety of open source and some commercial software packages.
- Parallel development including:
  - **C**: multithreading since its 2011 standard
  - **C++**: multithreading since its 2011 standard
  - **Fortran**: since its 2003 standard
  - **Julia**: shared and distributed memory
  - **MPI**, **Chapel**: shared and distributed memory
  - **OpenMP**, **pthreads**: shared memory
  - **CUDA**, **OpenACC**, **OpenCL**: GPUs
- Data science support:
  - DASK, Julia, Jupyter, Python, R, etc.

Compute Cluster: Environment (cont.)

- Container technology: Apptainer
  - Docker cannot be used at all.
  - Apptainer should be used with Docker images.
  - Since `podman` is also installed, it can also be used instead of Apptainer.
- Compiler toolchains:
  - GNU Compiler Collection (GCC)
  - Clang/LLVM
  - Intel OneAPI
  - NVIDIA nvhpc Toolkit

One can also install additional software in one's own account.

Compute Cluster: Clusters Are Different Than Your Computer

Enterprise supercomputing clusters:

- consume lots of resources including power
- can run 24/7/365 doing computations
  - Don’t run 24/7/365 on your computer or it won’t last!
- require submitting work to be run as “jobs”
  - jobs are (fairly) queued (and so don’t run instantly)
  - Remember there are many other users —not just you!

This following is a video link of the nibi server room:

- Click this link to view with external viewer.

Compute Cluster: Clusters Are Different Than Your Computer (cont.)

On clusters:

- There are many users able to submit jobs to run at any time.
  - unlike on your own computer where you are likely the only user
- Each submitted job needs to provide:
  - how much **RAM** the job requires (at most),
  - how many **CPUs** the job requires,
  - (if needed) how many **GPUs** the job requires,
  - how much **time** the job will run (at most) for, and,
  - which **sponsor account** the job is to be run under.
- Once a job has been submitted, the scheduler determines when the job can be run.
- Jobs normally run **non-interactively** with **no access** to a keyboard, mouse, or screen.
- Job **input and output** typically needs to be from/to **files** stored on disk.

Compute Cluster: Using a Compute Cluster

- Generally each cluster has a **login node** and a set of **compute nodes**.
- Program development, debugging, and testing can be done:
  - Via a login node within an interactive scheduler job,
  - on a login node *if and only if* such does not significantly use resources (time, CPUs/GPUs, and/or RAM), or,
  - using your own computer.
- Jobs are usually submitted from a login node.
- Login node access is via SSH or one of the web portals.

Compute Cluster: Maximizing Job Throughput

All jobs are submitted to a queue for the scheduler to schedule and run:

- Your research team ideally wants to always have jobs **in the queue** waiting to be run to maximize throughput.
- Each individual in every research team has the same fair-share priority of the entire team in terms of job submission.

Compute Cluster: Supported Software

A list of available software on our clusters can be see at:

- `https://docs.alliancecan.ca/wiki/Available_software`

Upon request, we work with researchers to help install and use software on our systems.

Compute Cluster: Why Use Supercomputing Resources?

On your own computers or your lab computers, you:

- do not have sufficient CPU/GPU cores
- do not have sufficient memory (RAM or storage)
- need a lot of disk space, e.g., hundreds of GBs or TBs
- need to run a lot of simulations using hundreds of CPU/GPU cores
- need to run web services
- need to run an SQL database service to service compute jobs / cloud resources
- need to make use of cloud resources

2026 Resource Allocation Competition: Overview

Every year there is a Resource Allocation Competition (RAC):

- peer-reviewed competition
- for research projects that need resources significantly beyond what is normally available

2026 Resource Allocation Competition: Overview (cont.)

There are two kinds of RAC awards:

- **Research Platforms and Portals (RPP)** which primarily involves using cloud(s)
- **Resources for Research Groups (RRG)** which primarily involves using compute cluster(s)

2026 Resource Allocation Competition: Overview (cont.)

Key dates for thr 2026 RAC:

- RRG and RPP application deadline:
  - Sept. 23 to Nov. 4, 2025 (no extensions of deadline)
- Information sessions: see web site link below for slides and recordings.
- 2026 RAC result announcements: Late March 2026
- 2026 RAC allocation start: Early April 2026
- URL: `https://docs.alliancecan.ca/wiki/Resource_Allocation_Competition`

Digital Research Alliance of Canada · Compute Ontario · SHARCNET · University of Windsor

2026 Resource Allocation Competition: Minimums

Q. Should your research team apply for a RAC?

A. Only if your team will need to use one or more more than the following slides' minimums.

2026 Resource Allocation Competition: Minimums (cont.)

Each 2026 RRG (compute) RAC application must meet at least one of these minimum requirements:

- CPU core years or equivalent: 200 years
- Reference GPU units (RGPU): 25
- Project storage: 41 TB
- Nearline storage: 101 TB
- A submission is required if any dCache storage is needed.
- URL: `https://docs.alliancecan.ca/wiki/RAC_application_guide`

Digital Research Alliance of Canada · Compute Ontario · SHARCNET · University of Windsor

2026 Resource Allocation Competition: Minimums (cont.)

Each 2026 RPP (cloud) RAC application must meet at least one of these minimum requirements:

- virtual CPU years (VCPU): 81
- virtual GPU years (VGPU) / RGU years: 3.6
- persistent cloud VCPU (VCPU years): 26
- volume and snapshot storage: 11 TB
- shared filesystem storage: 11 TB
- object storage: 11 TB
- URL: `https://docs.alliancecan.ca/wiki/RAC_application_guide`

Digital Research Alliance of Canada · Compute Ontario · SHARCNET · University of Windsor

2026 Resource Allocation Competition: Minimums (cont.)

IMPORTANT!

The 2026 RAC will **not** accept:

- applications that exceed 10 pages (before references),
- references must be submitted as a separate document (if the total number of pages in the entire application including references will exceed 10), and,
- documents must be submitted in PDF format.

See the “RAC application guide” for more detail, templates, etc.

2026 Resource Allocation Competition: Getting Help / Advice

One can request staff interaction for RAC applications by sending an email requesting such to `mailto:support@tech.alliancecan.ca` to open a ticket.

Questions, Answers, and Discussion

Thank you!

Questions, answers, and discussion.

Appendices

These appendices are extra slides provided as examples should some questions be asked about the compute clusters and running/compiling code.

Appendix: Software/Environment Modules on Clusters

- We have many software program versions available on our compute clusters.
- These can be made use of by using the `module` command.
- Our documentation wiki lists these software titles and versions at `https://docs.alliancecan.ca/wiki/Available_software`

Appendix: Software/Environment Modules on Clusters (cont.)

Some `module` commands:

- **`module reset`**: reset loaded modules back to defaults
- **`module avail`**: shows names of software packages that are available with the currently loaded modules
- **`module spider <PKGNAME>`**: shows information about `<PKGNAME>` and its versions
- **`module spider <PKGNAME>/<VERSION>`**: shows information about version `<VERSION>` of `<PKGNAME>` and how it must be loaded
- **`module load <PKGNAME>`**: loads the default version of `<PKGNAME>`
- **`module load <PKGNAME>/<VERSION>`**: loads version `<VERSION>` of `<PKGNAME>`
- **`module unload <PKGNAME>`**: unloads `<PKGNAME>`

Appendix: Slurm Scheduler Use

Jobs are submitted and scheduled using the Slurm scheduler using these commands:

- **`sbatch`**: submits a job into the scheduler's job queue
- **`salloc`**: requests an interactive job
- **`scancel <JOBID>`**: cancels job id `<JOBID>`
- **`squeue`**: see all jobs currently in the job queue
- **`sacct`**: see history of jobs run recently

Appendix: Slurm Scheduler Use (cont.)

The **sbatch** command is typically invoked like this:

- `sbatch --account <ACCOUNT_NAME> <YOURJOBSCRIPT_FILENAME>`

where:

- `<ACCOUNT_NAME>` is the name of the sponsor's account the job will run under
- `<YOURJOBSCRIPT_FILENAME>` is the name of the job script to run

Appendix: Slurm Scheduler Use (cont.)

An example `yourjobscript.sh` to run an instance of a sequential program might be:

yourjobscript.sh

```
1 #!/bin/bash
2 #SBATCH --time=0-05:00        # D-HH:MM, i.e., 5 hours (max time)
3 #SBATCH --mem=4000M           # i.e., job needs ~4 GB RAM (max)
4 ./sequential-program.exe
```

e.g., if the job account was `def-myacct` then `sbatch` can be used to submit such as a job as follows:

- `sbatch --account def-myacct yourjobscript.sh`

Appendix: Slurm Scheduler Use (cont.)

An example job script to run 100 instances of the program:

sequential-array.sh

```
#!/bin/bash
#SBATCH --time=0-05:00        # D-HH:MM, i.e., 5 hours (max time)
#SBATCH --mem=4000M           # i.e., job needs ~4 GB RAM (max)
#SBATCH --array=1-100

# e.g., one can pass $SLURM_ARRAY_TASK_ID to tell program which instance it is...
./sequential-program.exe $SLURM_ARRAY_TASK_ID
```

Appendix: Slurm Scheduler Use (cont.)

An example job script to run a single-GPU, single-GPU program:

1cpu1gpu-job.sh

```
#!/bin/bash
#SBATCH --time=0-11:00
#SBATCH --mem=4000M
#SBATCH --gpus-per-node=1

./gpu-program.exe
```

Appendix: Slurm Scheduler Use (cont.)

An example job script to run a multithreaded program using 10 CPU cores (on a single node):

10cpus1node.sh

```
1 #!/bin/bash
2 #SBATCH --time=0-05:00
3 #SBATCH --ntasks=1
4 #SBATCH --cpus-per-task=10
5 #SBATCH --mem-per-cpu=1024M
6
7 ./multithreaded-program.exe
```

Appendix: Slurm Scheduler Use (cont.)

An example job script to run an OpenMP program using 10 CPU cores:

10cpus1node-openmp.sh

```
#!/bin/bash
#SBATCH --time=0-05:00
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=10
#SBATCH --mem-per-cpu=1024M

export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK
 ./openmp-program.exe
```

Appendix: Slurm Scheduler Use (cont.)

An example job script to run a six CPU, single GPU OpenMP program:

6cpus1gpu-openmp.sh

```
#!/bin/bash
#SBATCH --time=0-05:00
#SBATCH --ntasks=1
#SBATCH --gpus-per-node=1
#SBATCH --cpus-per-task=6
#SBATCH --mem=4000M           # total memory needed

export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK
 ./openmp-program.exe
```

Appendix: Slurm Scheduler Use (cont.)

An example job script to run 8 MPI processes (using any number of nodes):

8ranks-mpi.sh

```
#!/bin/bash
#SBATCH --time=0-05:00
#SBATCH --ntasks=8
#SBATCH --cpus-per-task=1
#SBATCH --mem-per-cpu=1024M

srun ./mpi-program.exe
```

Appendix: Slurm Scheduler Use (cont.)

An example job script to run 8 MPI processes all running on one node:

8ranks1node-mpi.sh

```
#!/bin/bash
#SBATCH --time=0-05:00
#SBATCH --nodes=1
#SBATCH --ntasks=8
#SBATCH --cpus-per-task=1
#SBATCH --mem-per-cpu=1024M

srun ./mpi-program.exe
```

Appendix: Slurm Scheduler Use (cont.)

Our wiki has more job script examples at `https://docs.alliancecan.ca/wiki/Running_jobs`

Appendix: Slurm Scheduler Use (cont.)

The **`salloc`** command allows one to get an interactive job session with the requested resources dedicated to it.

To use `salloc`, pass on a single command line all of the `sbatch` job script options needed, e.g.,

- `salloc --account <ACCOUNT_NAME> --time=0-03:00 --cpus-per-task=10 --mem-per-cpu=1024M`

When this command is run, one will have to wait until those resources become available. Once the resources are available the shell prompt will return.

Appendix: Slurm Scheduler Use (cont.)

The **`scancel`** command allows one to cancel a queued or running job using its job ID, e.g.,

- `scancel 32385923`