

HPC Speed Cluster

Date: 2024

Serguei Mokhov
Gillian Roper
Carlos Alarcon Meza
Farah Salhany

High-Performance Computing (HPC) Facility: Speed - Concordia University
GitHub - NAG-DevOps/speed-hpc: Speed: Gina Cody School HPC Facility: Scripts, Tools, and Refs
Support: rt-ex-hpc@encs.concordia.ca

HPC Team

The team can be reached at rt-ex-hpc@encs.concordia.ca

- **Serguei Mokhov, PhD**
[Manager, Networks, Security, and HPC, AITS; Affiliate at CSSE and CIISE]
- **Gillian Roper**
[Senior Systems Administrator, HPC and Research Support, AITS]
- **Carlos Alarcón Meza**
[Systems Administrator, HPC and Networking, AITS]
- **Farah Salhany**
[IT Instructional Specialist, AITS]

Speed Resources

- **Speed official page:**
<https://www.concordia.ca/ginacody/aits/speed.html>
- **Speed manual:**
 - **PDF:** <https://github.com/NAG-DevOps/speed-hpc/blob/master/doc/speed-manual.pdf>
 - **HTML:** <https://nag-devops.github.io/speed-hpc/>
- **GitHub page includes examples (PRs are welcome):**
<https://github.com/NAG-DevOps/speed-hpc>
- **Slurm project account:** <>

ENCS Account and GCS Webmail

- To set up ENCS account for the **first time** OR if you cannot remember your ENCS credentials, go to the ENCS service desk (you could also send an email to help@concordia.ca if off campus)
- To change your existing ENCS password, visit the [password reset page](#)
- To access ENCS webmail, visit <https://webmail.encs.concordia.ca/>
 - access from off campus needs VPN
 - use same login as in the lab PCs or Speed

WHY?

- HPC announcements are sent to your ENCS email.
- Job notifications are also sent there by default

IMPORTANT

If you don't read it regularly, set up forwarding to a desired email address:

- Mail => Filters => Forward

=> **Save and Enable**

Mail ▾ Address Book ▾ Tasks ▾ Notes ▾ Others ▾ ⚙

Forward [Disabled] ?

Keep a copy of messages in this account? ☐

Address(es) to forward to:

Save Save and Enable Return to Rules List

Policies and Procedures

- Initial quota is **32 CPUs** and **4 GPUs**.
 - Can be expanded, for a fixed duration, upon request.
- A mailing list has been established for announcements and discussions. All users of the Speed cluster are automatically subscribed to it: hpc-ml@encs.concordia.ca
- Support requests: rt-ex-hpc@encs.concordia.ca

IMPORTANT

- *speed-submit* is a tiny virtual used for dispatching to execution nodes and is SSH accessible from Concordia subnets or from the VPN.
 - Excessive process or IDE executed on *speed-submit* will be terminated.
 - Only compile things on execution nodes (same for using IDEs)

Getting Started

- **Connect to Speed**

- Requires VPN (if off-campus)
- SSH to the cluster's submit node `[speed-submit]`
 - PuTTY or MobaXterm on Windows
 - Terminal on Mac or Linux

```
> ssh <ENCUsername>@speed.encs.concordia.ca
```

- **Go to a working directory**

`$USER` is a parameter (either manually replace it by your ENCS username or the system will automatically do the work.)

- First time (create a directory):

```
[speed-submit] [/home/../../] > mkdir /speed-scratch/$USER  
[speed-submit] [/home/../../] > cd /speed-scratch/$USER  
[speed-submit] [/speed-scratch/../../] >
```

- Not the first time (navigate to your directory):

```
[speed-submit] [/home/../../] > cd /speed-scratch/$USER  
[speed-submit] [/speed-scratch/../../] >
```

Storage

Students have **10GB** quotas (50GB for faculty), **however**, NFS speed scratch has **10TB**

► Hence why we recommend that you use *speed-scratch* for your project

To check your quota [in terminal]

```
> quota
```

WHERE	Diskspace			Files			volume/qtrees
	used	warn	limit	used	warn	limit	
Basic Home and Web	9.2G	9.8G	10.0G	56K	-	-	vol: users
INBOX	7.4M	80.0M	85.0M	1	-	-	vol: mailspool
/nettemp	0.0K	750.0M	800.0M	0	-	-	vol: nettemp
/speed-scratch	5.6G	9.8T	10.0T	48K	-	-	vol: speed_scratch

- “Out of quota” errors can be diagnosed with the `/encs/bin/bigfiles` command

Refer to: <https://nag-devops.github.io/speed-hpc/#how-to-resolve-disk-quota-exceeded-errors>

Storage

If you need to copy between **G:** or **U:** drive and `/speed-scratch/$USER`

- **Unix/Linux home U:**

- Speed to U:

```
> cp -r /speed-scratch/$USER/project ~/
```

- U: to Speed

```
> cp -r ~/project /speed-scratch/$USER/
```

- **Windows home G:**

- Speed to G:

```
> cp -r /speed-scratch/$USER/project /winhome/<1st letter of $USER>/$USER/
```

- G: to Speed

```
> cp -r /winhome/<1st letter of $USER>/$USER/project /speed-scratch/$USER/
```


SLURM tiny Intro

Slurm is workload manager, used for job scheduling

- **Batch job** (to run unattended tasks).
 - Job script acts as a wrapper for your actual job **my-script.sh**
 - Slurm options are usually embedded in a job script prefixed by **#SBATCH** directives.
for example:

```
#!/encs/bin/tcsh
#SBATCH --job-name=my-job --account=<>
#SBATCH --mail-type=ALL
#SBATCH --mem=10G -p pt -t 600
  or #SBATCH --mem=10G -p ps -t 600
#SBATCH --gpus=1 -p pg
```

- ◇ Use **srun** in your script for every complex compute step.
- Submit job to the cluster

sbatch [options] my-script.sh

Required/common Job Parameters:

- --job-name (-J)
- --account (-A)
- --partition (-p)
- --nodes (-N)
- --mem
- --time (-t)
- --ntasks (-n)
- --cpus-per-task (-c)
- --gpus (-G)
- --chdir (-D)
- --output (-o)
- --mail-type

SLURM tiny Intro

- **Interactive jobs** (to set up environments, debugging, and optimizing)

Similar to submitting a batch job but allows you to run shell commands interactively within the allocated resources.

- To request an interactive session, allocate resources with **salloc** [options]
for example:

```
salloc -J my-interactive-job -p pt --mem=10G
salloc: Granted job allocation 339377
salloc: Waiting for resource configuration
salloc: Nodes magic-node-04 are ready for job
```

- ◇ For quick, short job to compile on a GPU node, you can use an interactive **srun** directly, however **salloc** is preferred.

for tcsh:

```
srun --pty --gpus=1 --mem=1G -t 60 /encs/bin/tcsh
```

for bash:

```
srun --pty --gpus=1 --mem=1G -t 60 /encs/bin/bash
```

Running Batch Jobs

- **Check job output** (pay attention to requested vs. used resources and adjust accordingly)

```
[speed-submit] [/speed-scratch/../../speed-hpc/src] > seff 102406
```

⇒ Job ID: 102406
Cluster: speed
User/Group: carlos/carlos
State: COMPLETED (exit code 0)
Cores: 1
CPU Utilized: 00:00:00
CPU Efficiency: 0.00% of 00:00:30 core-walltime
Job Wall-clock time: 00:00:30
Memory Utilized: 4.84 MB
Memory Efficiency: 0.09% of 5.00 GB

Other commands to check:

- **sinfo -la**
- **squeue**
- **squeue -A comp6831f24**
- **showjob JOBID**
- **jobsummary JOBID**
- **sstat JOBID** #for each step

To cancel a job:

- Batch:
 > scancel JOBID
- Interactive:
 > exit

REFERENCES

- HPC GitHub: <https://github.com/NAG-DevOps/speed-hpc>
- Speed Manual: <https://nag-devops.github.io/speed-hpc/>
- High-Performance Computing (HPC) Facility: Speed: <https://www.concordia.ca/ginacody/aits/speed.html>
- Connect to VPN: <https://www.concordia.ca/it/support/connect-from-home.html>
- Connect to a GCS server: <https://www.concordia.ca/ginacody/aits/support/faq/ssh-to-gcs.html>
- MobaXterm: <https://mobaxterm.mobatek.net/download-home-edition.html>
- Linux and scripting lessons: <https://software-carpentry.org/lessons/>
- Udemy: <https://www.concordia.ca/it/services/udemy.html>
- Calcul Québec's example: https://github.com/calculquebec/formation_cpu-gpu
- Launch X (Graphical) applications: <https://www.concordia.ca/ginacody/aits/support/faq/xserver.html>
- The Unix Shell: <https://swcarpentry.github.io/shell-novice/index.html>

For help, email: rt-ex-hpc@encs.concordia.ca

THE END

Thank You