

Why the Trillium Cluster is Different

Ramses van Zon

Compute Ontario Colloquium - May 6, 2026

We want computational resources, and now!

Computational demands



Digital Research
Alliance of Canada

You may know that DRAC has an annual resource allocation competition (RAC).

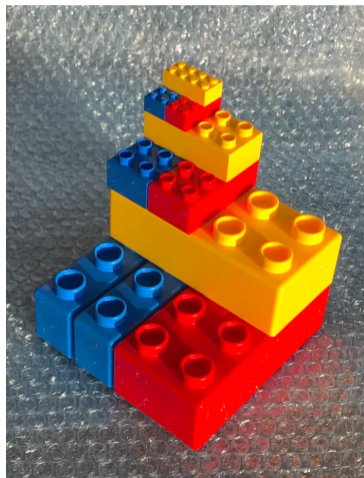
- For 2026, 766 research groups applied for resources on the DRAC clusters.
- Not everyone can get what they want: 90% of CPU asks and 23% of GPU asks were granted.
- Very small requests are not allowed (≤ 200 core years, ≤ 2 GPUs); those groups are served by 'opportunistic' allocation of altogether $\sim 25\%$ of the systems

So more resources are requested than can be delivered.

⇒ **We have a responsibility to ensure that the available resources are well-utilized.**

Varied requirements

All those resource requests do not all have the same computational characteristics.



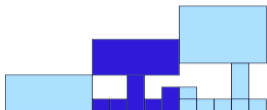
One could have projects that need:

- Many short single-core computations
- Long running single-core computations
- Single node parallel computations (still ~192 cores each!)
- Large, multi-node parallel computations
- Large memory computations

And some may use GPUs, while others do not.

This poses a scheduling issue.

Scheduling issue



To allow fair sharing of the clusters, users must to submit compute jobs that request resources from the scheduler (SLURM).

The scheduler then uses a priority queue, to ensure the resources are fairly shared.

Problem

Large compute jobs need:

- Many, full nodes available at the same time
- Preferably physically close in the network topology
- Consistent hardware

If scheduling allows both small and large jobs, small jobs would get in the way of scheduling large jobs. So the scheduler would have to leave resources unused for those large jobs.

E.g. imagine one serial job is still running on a 192-core node. All other 191 cores are then unused until that serial job is done.

Solution

Do we need one?

Yes! The RAC applications show that there is demand for substantial large parallel capability from researchers in aerospace, astrophysics, atmospheric and climate physics, nuclear physics, material science, engineering, etc.. to conduct large-scale simulations.

To be able to serve these researchers as well as projects which require smaller computations, first Compute Canada and then DRAC has chosen for a two-pronged approach:

General Purpose Clusters

Optimize most clusters for throughput of relatively small, modest size computations.

Fir, Nibi, Rorqual, Narval

Large Parallel Cluster

Dedicate one cluster to large parallel computations.

Trillium

Requirements for Trillium



- Dedicated to large parallel jobs with intense computation and/or communication
 - Many cores
 - good interconnect and topology
 - fast and large storage
- Large CPU-only portion (many computations are still CPU based)
- Compatibility and coordination with other national systems
 - Ask access on CCDB
 - Use same folder layout and permissions
 - Mount CC software stack
 - Use CC support email
 - Use SLURM scheduler

Architecture differences

| | Fir | Nibi | Rorqual | Narval | Trillium |
|-----------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|--|
| CPU Nodes | 872 nodes x 192 zen5 cores | 710 nodes x 192 gen6 cores | 686 nodes x 192 zen4 cores | 1181 nodes x 64 zen2 cores | 1224 nodes x .192 zen5 cores |
| GPU Nodes | 160 nodes x 4 H100 | 36 nodes x 8 H100 | 93 nodes x 4 H100 | 159 nodes x 4 A100 | 63 nodes x .4 H100 |
| Local disk | 7.8TB | ≥ 3TB | ≥ 480GB | ≥ 960GB | 0 |
| Network | 400 Gbit/s | 200 Gbit/s | 200 Gbit/s | 100 Gbit/s | 400/800 Gbit/s |
| Max.nonblocking | 1 node | 1 node | 31 nodes | 56 nodes | 1224 nodes |
| Blocking factor | 27:5 | 2:1 | 34:6 | 33:7 | 1:1 |
| File system(s) | Lustre | VAST | Lustre | Lustre | VAST |

Source: <https://docs.alliancecan.ca>

Most of these are obvious improvements, but be aware:

- The nodes have no local disk. Usually, VAST is fast enough. If not, you could often use ramdisk.
- Modules from the StdEnv/2020 may not work on the Zen5 architecture, and we're not fixing that; use StdEnv/2023.

Usage differences

Login

- Password access to the login nodes is disabled. You must use SSH Keys and MFA.¹
- There are separate login and robot nodes for CPU and GPU subclusters.

Job scheduling

- The CPU and GPU subclusters have their own scheduler.
- The login nodes are the job submission nodes.
- The CPU resources are scheduled by whole 192-core node.
- The GPU resources are scheduled by GPU (no “MIG”), or by whole node.
- Strict 24h limit on jobs. Improves turnover and scheduling of large jobs. Also, it’s safer (use checkpointing!).
- No memory specification: You get 1/4 or 1/1 of the node’s memory.

Interactive jobs

- Use debugjob. It’s faster than salloc for up to 4 nodes, up to 1 hour. It does not count to your fairshare.

Usage differences

Home

- \$HOME is read-only in compute jobs.
- You will have space for up to 100 GB (instead of 50GB) or 1 million files in your \$HOME directory.

Project

- Find links to the project spaces to which you have access in the \$HOME/links directory.
- Default accounts have access to a project space of 1 TB per group.
- It is not possible to request more project space on Trillium through RAS.
- \$PROJECT is read-only in compute jobs.

Scratch

- Users have a nominal quota of 25TB, but are expected to clean up all unused data.
- There is no purging policy yet, but this is subject to change in the future.

Nearline space

- Not a mounted file system on Trillium: must use HPSS.

Usage commonalities

Access

- Request on CCDB like for other systems
- Login with ssh using CCDB account

Environment

- Linux environment
- Same software module stack
- Slurm scheduler

File system

- Shared file systems
- Users have a \$HOME, \$SCRATCH, and \$PROJECT space
- \$HOME and \$PROJECT are backed up, \$SCRATCH is not
- File transfer with scp/rsync or Globus
- Nearline storage systems
- Same group membership
- Same directory layout and permissions as on other clusters

**What if I have to do some lighter
post/preprocessing/interactive work?**

Testing and Debugging

If you're not ready to run large jobs, test it first!

Trillium debugjob

```
$ ssh USERNAME@trillium.alliancecan.ca
tri-login01:~$ debugjob
debugjob: Requesting 1 node(s) with 192 task(s) for 60 minutes and 0 seconds
SALLOC: Granted job allocation 1510483
SALLOC: Waiting for resource configuration
SALLOC: Nodes tri0427 are ready for job
tri0427:~$
```

Trillium-GPU debugjob

```
$ ssh USERNAME@trillium-gpu.alliancecan.ca
trig-login01:~$ debugjob
debugjob: Requesting 1 node(s) with 1 gpu(s) for 120 minutes and 0 seconds
SALLOC: Granted job allocation 488955
SALLOC: Waiting for resource configuration
SALLOC: Nodes trig0001 are ready for job
trig0001:~$
```

Open OnDemand

- Interactive work may not require so many resources.
- A lot of interactive work is easier in a graphical environment
- Many clusters have “JupyterHub”
- Instead of JupyterHub, Trillium has Open OnDemand.

Trillium OnDemand

<https://ondemand.scinet.utoronto.ca>

This support a growing number of applications from your browser, such as JupyterLab, VS Code, RStudio, MATLAB, ParaView, the DDT debugger, as well as a terminal and job submissions.

- Interactive Apps on OnDemand can request partial nodes and memory.
- Interactive Apps on OnDemand have write access to HOME.
- Interactive Apps on OnDemand have internet access.
- The OnDemand nodes CPUs are different but compatible with Trillium CPU nodes.

More information

Want to know more?

- <https://docs.alliancecan.ca/wiki/Trillium>
- https://docs.alliancecan.ca/wiki/Trillium_Quickstart
- Intro to Trillium self-paced course <https://scinet.courses/1389>
- https://docs.alliancecan.ca/wiki/Trillium_Open_OnDemand_Quickstart
- Need support? Email trillium@tech.alliancecan.ca.

Thank you for attending! Questions?