Using Compute Canada Resources

An Introduction for Newcomers

Guillaume Perrault-Archambault

University of Ottawa

November 16, 2018

Outline

1. Overview
2. Setting Up
3. Running Jobs
4. Managing Resources

Outline

1. Overview
2. Setting Up
3. Running Jobs
4. Managing Resources

What is Compute Canada?

- Partnership between 37 member research institutions
- Provides access to world-class computing resources
- Supported by 200 experts in member institutions
- Computing Resources are made available via clusters

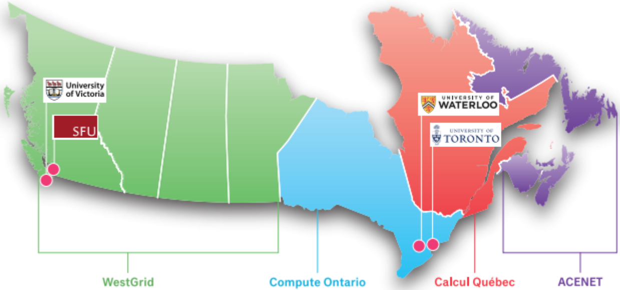
Compute Canada Systems, or Clusters

- What are they?

Niagara supercomputer. Source: University of Toronto (https://goo.gl/Jzw5xz)

Compute Canada Systems, or Clusters

- Where are they?



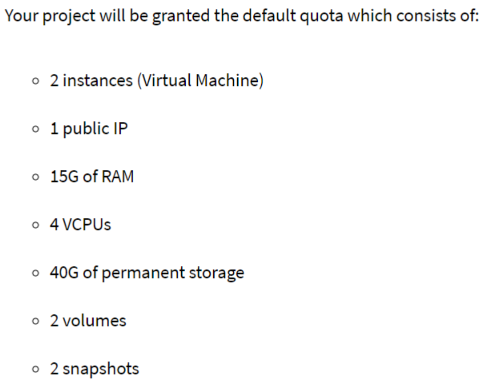
Map of national systems. Source: Compute Canada (https://goo.gl/XbgF4H)

Cloud Compute Systems

- Remote virtual machines
- Useful for custom software install
- Dedicated compute resource
- Not designed for high computation speed
- Think of it as remote laptop with:
    - More power
    - Better software
    - A support team

Cloud System – Example Allocation

- Default Allocation



Your project will be granted the default quota which consists of:

- 2 instances (Virtual Machine)
- 1 public IP
- 15G of RAM
- 4 VCPUs
- 40G of permanent storage
- 2 volumes
- 2 snapshots

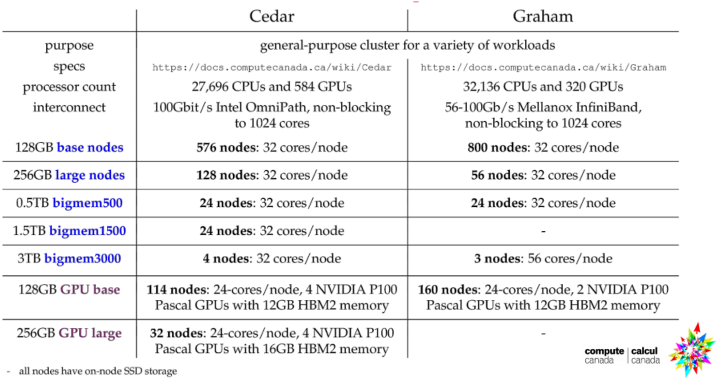
Default project allocation for cloud account. Source: Compute Canada (https://goo.gl/ML5Lnr)

Performance Compute Systems

- High performance parallel computing
- Shared between users – not dedicated
- Not designed for visualization
  - Debug on your laptop and launch big jobs on PCS

Performance System – Example System Specs

- Cedar, Graham Clusters



	Cedar	Graham
purpose	general-purpose cluster for a variety of workloads	
specs	https://docs.computecanada.ca/wiki/Cedar	https://docs.computecanada.ca/wiki/Graham
processor count	27,696 CPUs and 584 GPUs	32,136 CPUs and 320 GPUs
interconnect	100Gbit/s Intel OmniPath, non-blocking to 1024 cores	56-100Gb/s Mellanox InfiniBand, non-blocking to 1024 cores
128GB **base nodes**	**576 nodes**: 32 cores/node	**800 nodes**: 32 cores/node
256GB **large nodes**	**128 nodes**: 32 cores/node	**56 nodes**: 32 cores/node
0.5TB **bigmem500**	**24 nodes**: 32 cores/node	**24 nodes**: 32 cores/node
1.5TB **bigmem1500**	**24 nodes**: 32 cores/node	-
3TB **bigmem3000**	**4 nodes**: 32 cores/node	**3 nodes**: 56 cores/node
128GB **GPU base**	**114 nodes**: 24-cores/node, 4 NVIDIA P100 Pascal GPUs with 12GB HBM2 memory	**160 nodes**: 24-cores/node, 2 NVIDIA P100 Pascal GPUs with 12GB HBM2 memory
256GB **GPU large**	**32 nodes**: 24-cores/node, 4 NVIDIA P100 Pascal GPUs with 16GB HBM2 memory	-

- all nodes have on-node SSD storage

Cedar, Graham Specifications. Source: Compute Canada (https://goo.gl/DvJY2N)

National Systems

- You get access automatically with your CC account
- Standardized workflow
- National clusters:
  - Cedar
  - Graham
  - Beluga[1]
  - Arbutus[2]

[1] Available in January 2019

[2] Cloud only

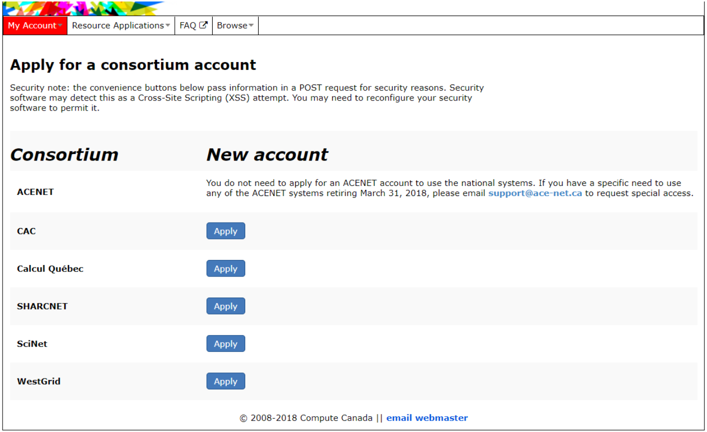
Regional Systems

- Requires creation of regional or "consortium" account
- Each may have unique workflow
- Are being phased out in favor of national systems
    - Compute Canada calls them "Legacy Systems"
- Noteworthy regional clusters:
    - Hades (Calcul Quebec)
    - Guillimin (Calcul Quebec)
    - Mammouth[1] (Calcul Quebec)
    - Accelerator Research Cluster (SciNet)
    - Niagara[1] (SciNet)
- Use only if needed

[1] CPU only

Regional Systems – Applying

- Application Page



Apply for a consortium account

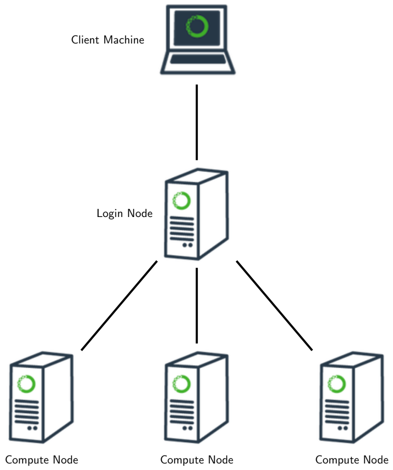
Security note: the convenience buttons below pass information in a POST request for security reasons. Security software may detect this as a Cross-Site Scripting (XSS) attempt. You may need to reconfigure your security software to permit it.

Consortium	New account
ACENET	You do not need to apply for an ACENET account to use the national systems. If you have a specific need to use any of the ACENET systems retiring March 31, 2018, please email support@ace-net.ca to request special access.
CAC	Apply
Calcul Québec	Apply
SHARCNET	Apply
SciNet	Apply
WestGrid	Apply

© 2008-2018 Compute Canada || email webmaster

Consortium account application page. Source: Compute Canada (https://goo.gl/sGZCW3)

Compute Cluster - Client Model



- Clients connect to login node
- Jobs submitted to compute node
- Multiple login nodes
- Clients share login nodes

Outline

1. Overview
2. Setting Up
3. Running Jobs
4. Managing Resources

Setting Up an Account

- Go to registration page and fill out form
  - Takes 5 minutes
  - Full instructions in Wiki
- Compute Canada will review then e-mail prof Mao to approve
  - Process may take a few days

Accessing a System

- ssh to Graham cluster (demo)
  - Can set up to be passwordless
  - Can set up a profile (putty example in Wiki)
- We are now on one of Graham's login nodes!
- Notice that the login nodes Unix-based
  - You will need to learn some basic Unix commands
  - See the wiki for two excellent Unix resources

Transferring Files To and From the Cluster

- Several methods
  - FTP client tool (demo with FileZilla)
  - SFTP mounting – described in the wiki
  - Globus – better for huge files, but works only off campus
- We are almost ready to run simulations!

Using Modules

- Compute Canada systems have a big stack of pre-installed software
- Some software cannot coexist in the same environment
  - eg two different versions of python
  - Collision can cause unintended behavior
- Solution: modules
  - Ensures a software environment with a consistent set of software
  - No collisions
- Simple to use
  - module avail
  - module list
  - module load/unload <module name>
- demo
- More info on Compute Canada Using Modules page

Installing Python Modules

- Compute Canada pre-installed python packages comes with limited number of python modules
- To install tensorflow or pytorch, follow special instructions
- For other modules, use

  ```
  pip install --user <name of module>
  ```
- demo
- Prescribed method by Compute Canada is different
  - Uses virtual environments; more complicated
  - Better if working on multiple big projects
  - Detailed here
  - Both flows are officially supported

Outline

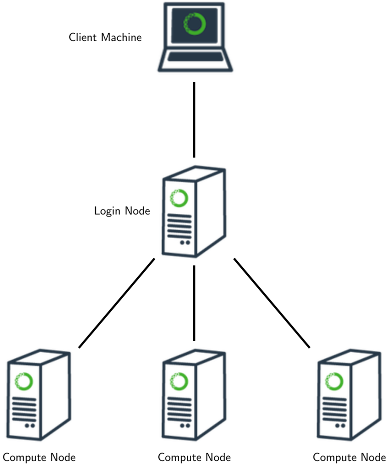
1 Overview

2 Setting Up

3 Running Jobs

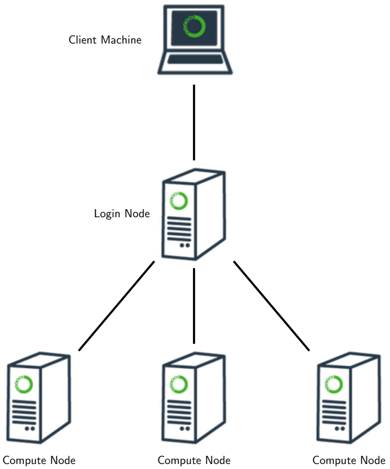
4 Managing Resources

Running Jobs



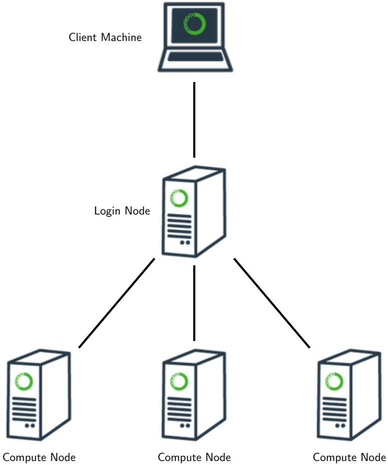
- Now that we are on the login node

Running Jobs



- Now that we are on the login node
- How do we launch jobs on the compute nodes?

Running Jobs



- Now that we are on the login node
- How do we launch jobs on the compute nodes?
- The answer is to use a Job Scheduler (or Workload Manager)

Job Scheduler

- The scheduler is a program that manages which users get the compute nodes and when
- There are several popular job schedulers: SLURM, LSF, Torque
- The largest Compute Canada systems (Cedar, Graham, Niagara, Mammouth) use SLURM
- Several commands to launch a job: salloc, srun, sbatch
  - salloc allows you to run commands on the compute node itself
  - srun allows you to run an interactive job
  - sbatch allows you to run a batch (non-interactive) job
- Most common command is sbatch

Submitting a batch job

- Tell SLURM what resources you need and what program you want to run
- Put these instructions inside a "batch" file:

```
#!/bin/bash
#SBATCH --gres=gpu:1              # Number of GPUs (per no
#SBATCH --time=00:30:00
#SBATCH --job-name="quick test"
#SBATCH --mem=2gb
#SBATCH --output=%x-%j.out
echo ’Hello, world!’              # Replace this with your
```

- Now we run:

```
sbatch <batch file name>
```

- Quick instructions here
- Compute Canada detailed instructions here

Monitoring and Canceling Jobs

- Monitoring your jobs' status

  ```
  squeue -u <username>
  ```

- Cancelling a job

  ```
  scancel <job id>
  ```

- demo

Outline

1. Overview
2. Setting Up
3. Running Jobs
4. Managing Resources

Check memory and CPU usage

- Max RAM usage

  ```
  sacct -j <job id> --format=JobID,JobName,MaxRSS,Elapsed
  ```

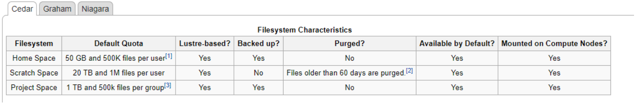
- Job efficiency

  ```
  seff <job id>
  ```

- demo

Storage space

- Cedar storage allocation



Filesystem Characteristics

Filesystem	Default Quota	Lustre-based?	Backed up?	Purged?	Available by Default?	Mounted on Compute Nodes?
Home Space	50 GB and 500K files per user[1]	Yes	Yes	No	Yes	Yes
Scratch Space	20 TB and 1M files per user	Yes	No	Files older than 60 days are purged.[2]	Yes	Yes
Project Space	1 TB and 500k files per group[3]	Yes	Yes	No	Yes	Yes

Cedar Filesystem Characteristics. Source: Compute Canada (https://goo.gl/yZu6Bb)

- Key takeaways
  - Run jobs in scratch disk
  - Only keep important files on project space
  - Don't keep files in home directory

Check storage space usage

- Check usage with:

```
diskusage_report
```

- Or per file/directory:

```
du -sh <file or directory name>
```

- demo

Fairshare

- NormShares
  - Number of compute shares compared to other associations (fraction)
- NormShares
  - Number of compute shares compared to other associations (fraction)
- EffectvUsage
  - Associations' compute usage compared to other associations (fraction)
- UsageRate (my term)
  - EffectvUsage / NormShares
  - How much association has used of its fairshare
- LevelFS ("Level Fairshare")
  - NormShares / EffectvUsage (ie 1/UsageRate)
  - How "underserved" an association is
  - Number above 1.0 means association underserved – jobs will have high priority

Check fairshare level

- Check CPU fairshare with:

```
sshare -l -A def-yymao_cpu -a
```

- Check GPU fairshare with:

```
sshare -l -A def-yymao_gpu -a
```

- demo
- both memory and CPU/GPU years count toward fairshare usage!

Common Project Space

- Used to share common scripts or templates
- Currently on Cedar
    - My bashrc (can be used as template or as-is)
    - SLURM run script
    - directory with simulation example (compatible with run script)
- Work in progress!
- demo

Getting Help

- Compute Canada Wiki
- Our Wiki
- Compute Canada technical support: support@computecanada.ca
    - Excellent!
- Centre for Advanced Computing team
    - support@computecanada.ca usually redirects to them
- Other students
- Me