SHARCNET™

# New User Seminar Part II

by Sergey Mashchenko
SHARCNET / Compute Canada
based @ McMaster University

# Outline

- Interactive jobs
- Requesting alternative GPUs
- Installing software yourself
- Advanced use of the module command
- Remote data copying with Globus
- Automation of job submission
- Graham’s VDI nodes

# Interactive jobs

Main reference: https://docs.computecanada.ca/wiki/Running_jobs

- Using `salloc` command instead of `sbatch`.
- Salloc accepts the same arguments as sbatch, but only via command line.
- As interactive jobs tend to be wasteful, this is primarily for **quick tests** / **debugging** / **profiling**.
- One can have all kinds of interactive jobs - serial, multithreaded, MPI, CUDA.
- There is a limit of 24 hours imposed on the runtime of interactive jobs.
- There is a limited number of cluster nodes set aside for interactive use; as a consequence, sometimes the wait time can be considerable.

# Requesting alternative GPUs

Main references: https://docs.computecanada.ca/wiki/Graham (Graham)
https://docs.computecanada.ca/wiki/Using_GPUs_with_Slurm (the rest)

- By default, you get P100 GPUs on Graham.
- Graham also has other types of GPUs:
  - 64 V100’s (newer generation - Volta - general purpose GPUs). For out-of-Ontario users, runtime is limited to one day.
  - 144 T4’s (even newer generation - Turing - GPUs, with Turing Tensor cores, but with poor double precision performance; great for machine learning and such).
- For non-default GPUs, one has to request the specific GPU type via `--gres` switch (in `sbatch` and `salloc`). E.g., “`--gres=gpu:v100:1`”.

# Installing software yourself

Main reference: [https://docs.computecanada.ca/wiki/Installing_software_in_your_home_directory](https://docs.computecanada.ca/wiki/Installing_software_in_your_home_directory)

- Sometimes you need to install software yourself
  - If you need a bleeding edge version of a package (which is not installed yet system-wide)
  - If you need to customise a package
  - If this software is not installed on the system
- Many open source packages use standard "`configure`, `make`, `make install`" sequence of commands for compiling and installation.
- In many  cases this will also work on our systems, once you load all the required modules.
- Make sure to provide "`--prefix=installation_directory`" switch to the configure command, otherwise "`make install`" command might fail as you don't have a root access.

# Installing software yourself (2)

- Sometimes the software is not smart (or standard) enough to find all the libraries it needs, based on the current environment (variables `$LIBRARY_PATH`, `$CPATH`, `$PATH` etc.).
- In such situations, one can make use of the per-module environment variables which are named `$EBROOT{PACKAGE_NAME}` and contain the path to the root of the package.
- For example, for FFTW modules, first one needs to load the corresponding FFTW module, and then provide the following flags to the compiling command:

```
  -I $EBROOTFFTW/include -L $EBROOTFFTW/lib
```

  followed by the usual library references (e.g. `-lfftw3_mpi -lfftw3`).

# Advanced use of the module command

Main reference: [https://docs.computecanada.ca/wiki/Utiliser_des_modules/en](https://docs.computecanada.ca/wiki/Utiliser_des_modules/en)

- Basic `module` commands: `list`, `spider`, `load`, `unload`.
- Resetting to the system default set of modules (after you messed up with loading/unloading modules): `module reset`.
- Commands to create and operate custom collections of modules:
  - `module save [name]` - save the current list (optionally, to “name”)
  - `module restore [name]` - restore the default (or “name”) list
  - `module savelist` - list your saved collections
  - `module disable name` - delete the “name” collection

# Remote data copying with Globus

Main reference: **https://docs.computecanada.ca/wiki/Globus**

- For copying data between national systems, or between your machine and a system, the usual commands (scp, rsync) work fine for up to a few GBs.
- For larger amounts of data, one is advised to use browser-based tool Globus.
- Login to http://globus.computecanada.ca/ using your Compute Canada credentials.
- Go to File Manager tab to initiate a data transfer. Search for system End Points by typing “graham”, “cedar” etc.
- You can create End Point for your computer (apps exist for Windows, Mac, Linux).

# Automation of job submission

Main references: https://docs.computecanada.ca/wiki/Job_arrays
https://docs.computecanada.ca/wiki/META_package_for_serial_farming

- A simple way to automate submission of many jobs is the Job Arrays feature of the scheduler. Inside your job script, variable `$SLURM_ARRAY_TASK_ID`(containing ID of the current sub-job) can be used to customize sub-jobs execution.
- A more powerful method to automate job submission is to use our serial farming package META. Start by installing the package on a cluster with

```
git clone git@git.sharcnet.ca:syam/META.git
```

# Graham's VDI nodes

Main reference: **https://docs.computecanada.ca/wiki/VNC**

- Simple GUI tasks can be executed on cluster login nodes (just enable X11 forwarding in your SSH client settings; it's ON by default in MobaXTerm).
- GUI-intensive tasks (visualization etc) are best to be done on dedicated Graham nodes, gra-vdi, using VNC mechanism.
- Install TigerVNC app on your computer, open the VNC viewer, and provide the VDI nodes address:

```
gra-vdi.computecanada.ca
```

# Questions?

Email: help@sharcnet.ca