

High Performance Computing: Short Introduction

U. of Manitoba and Compute Canada clusters.

Dr. Ali Kerrache

ali.kerrache@umanitoba.ca

University of Manitoba, Sep 20th, 2019.

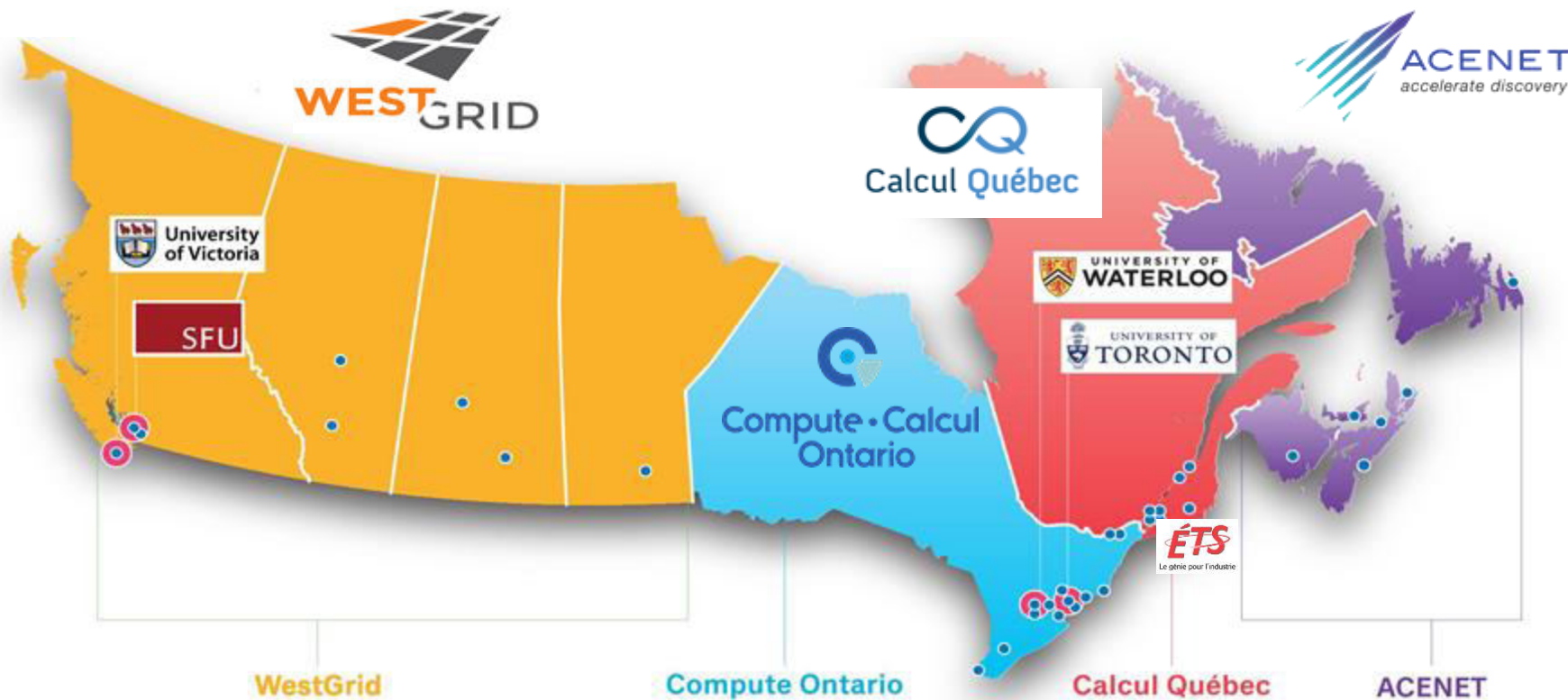
- ❖ *U. of Manitoba and Compute Canada HPC clusters:*
Greex (UofM), Cedar (SFU), Graham (UW), Beluga (ETS), ...
- ❖ Connect & transfer files to **HPC** clusters:
Linux, Mac, Windows, ssh, Putty, MobaXterm, ...
- ❖ Software environment on HPC clusters:
Compilers, Software Installation, Modules, ...
- ❖ Running, submitting and monitoring jobs:
Prepare, compile, submit, run, monitor your Jobs, ...
- ❖ Conclusions and more readings ([links](#)).

WestGrid and Compute Canada

- ❖ *WestGrid systems:* <https://www.westgrid.ca/>
GreX (UofM): *inherited from WG.*
- ❖ *Compute Canada systems:* Cedar, Graham, Beluga, ...
https://docs.compute canada.ca/wiki/Compute_Canada_Documentation
- ❖ *Accounts:*
 - *Compute Canada Account:* <https://ccdb.compute canada.ca/security/login>
 - *WestGrid Account:*
https://www.westgrid.ca/support/accounts/getting_account
- ❖ *Support and Help:*
 - support@westgrid.ca
 - support@compute canada.ca

Note: *read and understand the privacy consent.*

Compute Canada new clusters

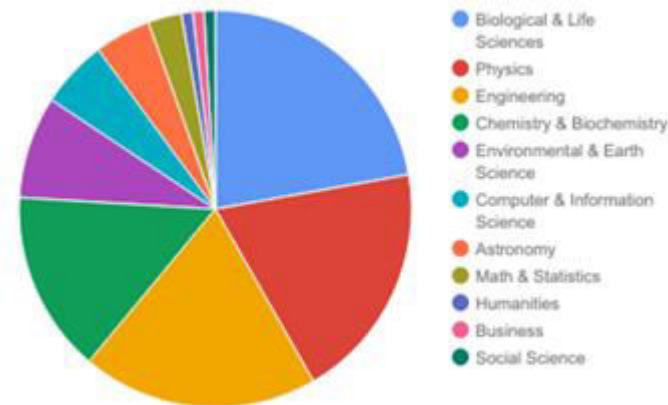
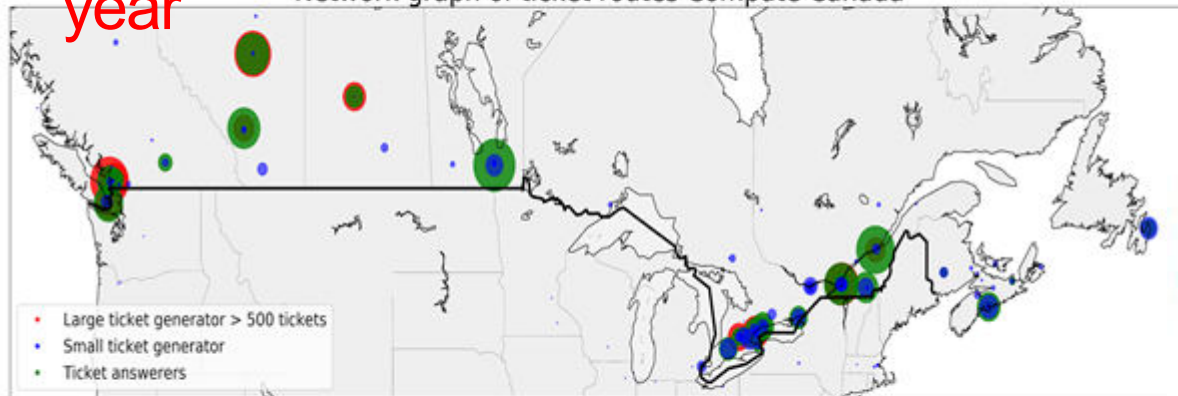


Compute Canada/Grex: people

~200 technical staff

~15,000 user accounts: 20% growth per year

Network graph of ticket routes Compute Canada



U. of Manitoba:

2 analysts and 2 system admins (IST).
250 roles in CCDB.

Recent usage: 35 active groups (80 users).

- All research disciplines supported.
- Free access for any researcher (and their external collaborators) at a Canadian institution.

Compute Canada Resources

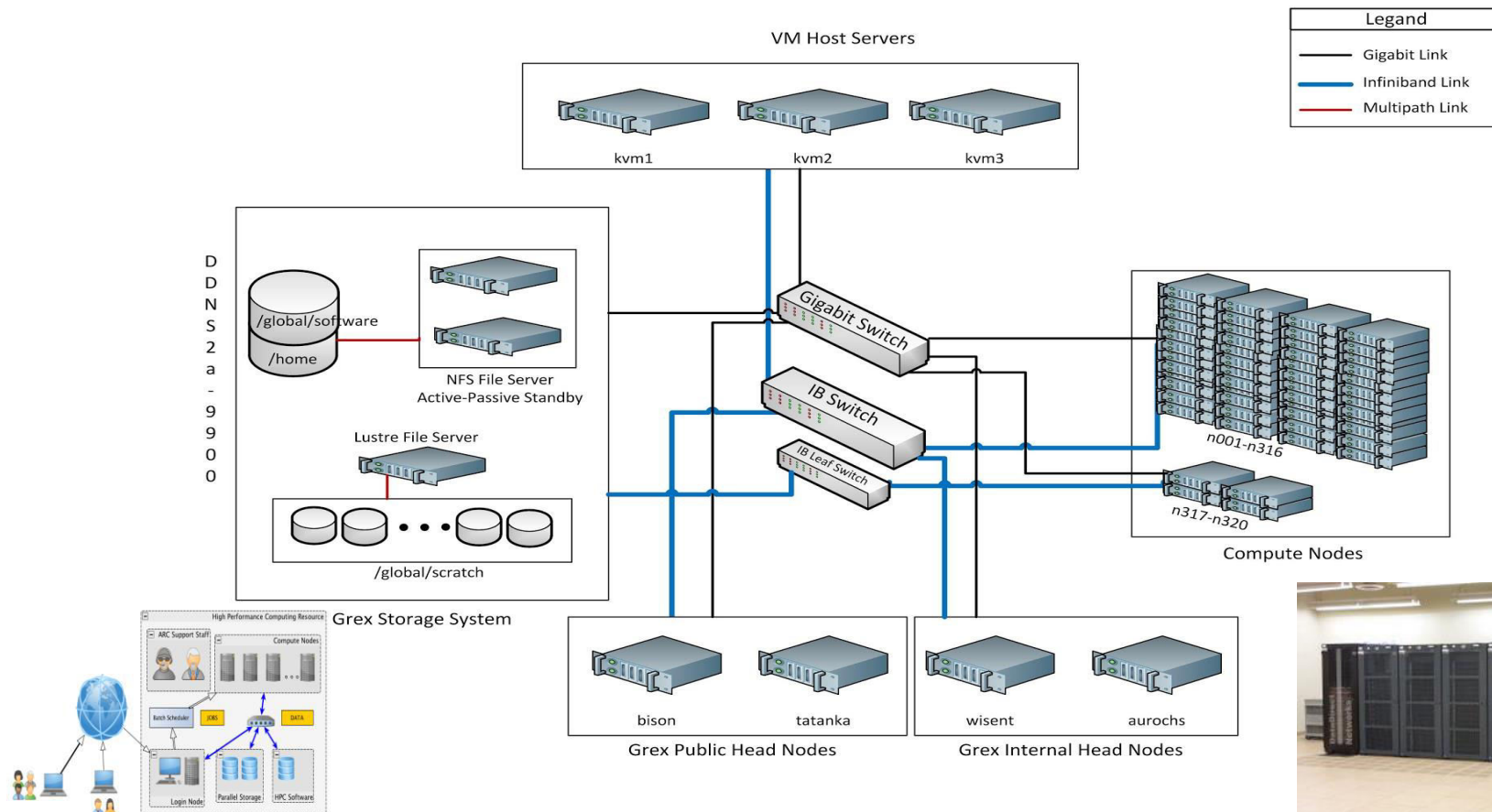
System	Cores	GPUs	Storage	Notes
Cedar	58,416	584	13 PB	NVidia P100 GPUs
Graham	33,472	320	12 PB	NVidia P100 GPUs
Beluga	34,880	688	2.6 PB	NVidia V100 GPUs
Niagara	60,000	0	2 PB	No GPUs; Large parallel jobs.
Arbutus (cloud)	9,048	0	5.7 PB	Physical cores: generally hyper-threaded.
GP Cloud partitions				Cloud partitions are available on GP systems for special purposes.

Resources at U. of Manitoba

	CPU's	GPU's	RAM/node	Storage	Notes
GreX	3780 (315 x 12 CPU) Intel Nehalem 2010	0	48 GB 94 GB	0.4 PB /global/scratch	WG: defunded in 2018 Now: UofM
Free for all CCL	768 (12 * 64 CPU) AMD Opterons 2011	0	256 GB	?	Free for all, use UM NetID
OpenStack	256 cores?	0			By request

- <https://www.westgrid.ca/support/systems/GreX>
- http://umanitoba.ca/computing/ist/service_catalogue/index.html

Structure of a HPC clusters: Grex



Accessing Compute Canada

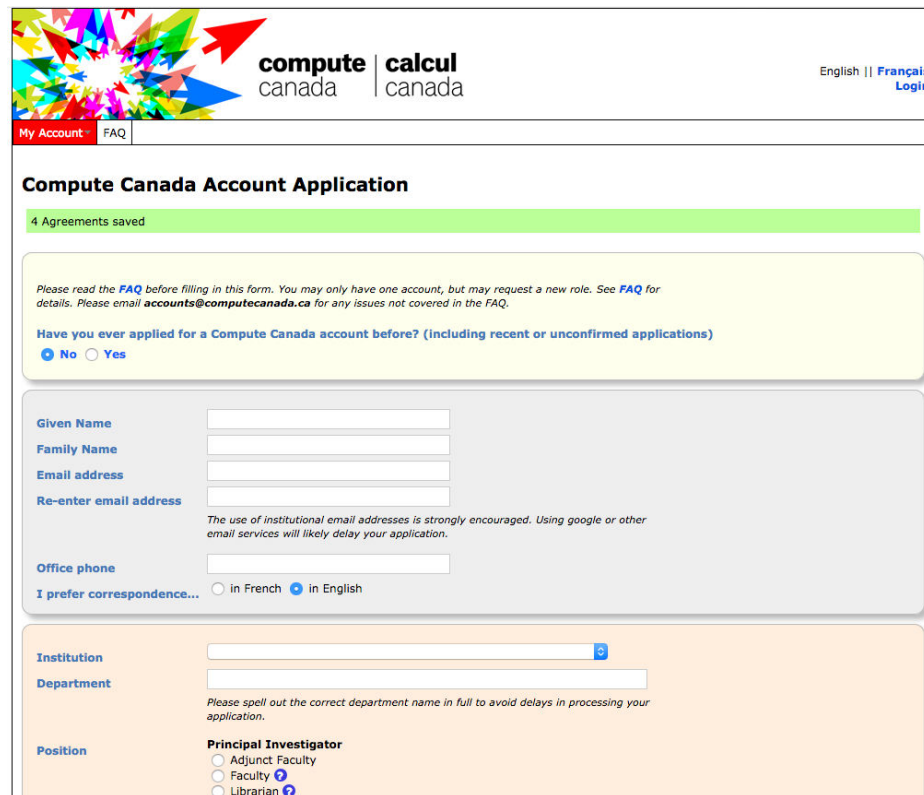


Step 1:

- Faculty member registers in the Compute Canada Database (CCDB)
- <http://ccdb.computeCanada.ca>

Step 2:

- Once account is approved, students / colleagues can register as group members:
- CCDB account Gives access to the New systems:
- ❖ Access to Compute Canada resources is free for the eligible researchers.
- ❖ Everyone gets a “Default” share.
- ❖ Every PI gets 1 TB of storage by default.
- ❖ Resource Allocation Competitions are held annually, to distribute resources based on proposal’s merit.



The screenshot shows the 'Compute Canada Account Application' page. At the top, there is a header with the 'compute | calcul canada' logo and a language selector for 'English' and 'Français'. Below the header, there are links for 'My Account' and 'FAQ'. The main title is 'Compute Canada Account Application'. A green bar indicates '4 Agreements saved'. A yellow box contains a disclaimer: 'Please read the FAQ before filling in this form. You may only have one account, but may request a new role. See FAQ for details. Please email accounts@computeCanada.ca for any issues not covered in the FAQ.' Below this is a question: 'Have you ever applied for a Compute Canada account before? (including recent or unconfirmed applications)' with radio buttons for 'No' (selected) and 'Yes'. The form then has several input fields: 'Given Name', 'Family Name', 'Email address', 'Re-enter email address', 'Office phone', and a language preference section with radio buttons for 'in French' and 'in English' (selected). The bottom section has 'Institution' and 'Department' dropdown menus, a 'Position' dropdown, and a 'Principal Investigator' section with radio buttons for 'Adjunct Faculty', 'Faculty' (selected), and 'Librarian'.

Accessing Grex

Step 3:

➤ Optional, Regional accounts:

<https://ccdb.compute canada.ca/me/facilities>

❖ Needed for Legacy HPC Resources (Grex, MP2, etc.)

❖ WG: Grex, ownCloud



WestGrid Portal

portal.westgrid.ca [Login]

Applications for WestGrid Accounts

WestGrid account applications are now made through the Compute Canada Database site (CCDB)

<https://ccdb.compute canada.ca>

See the WestGrid web pages for further information:

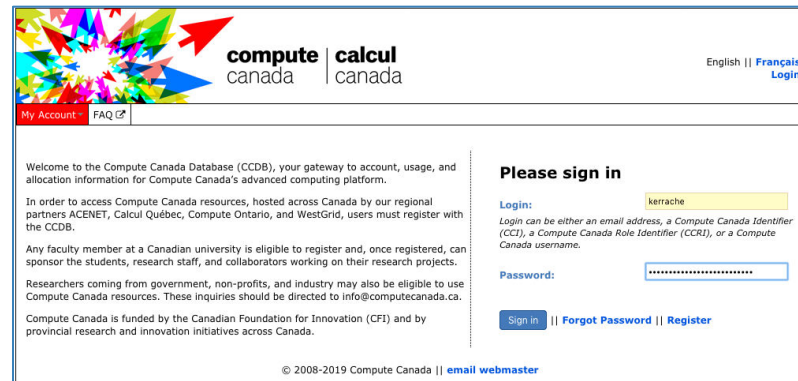
- http://www.westgrid.ca/support/prospective_users
- <http://www.westgrid.ca/support/accounts>
- http://www.westgrid.ca/support/accounts/applying_for_an_account

On the WestGrid account request pages, you will be asked to

1. Confirm your eligibility and accept the conditions of use.
2. Enter username and select a password.

If at any time during the application process you have any questions or encounter any problems, please contact us at accounts@westgrid.ca.

(Your browser must accept cookies.)



compute canada | calcul canada

English || Français Login

My Account - FAQ

Welcome to the Compute Canada Database (CCDB), your gateway to account, usage, and allocation information for Compute Canada's advanced computing platform.

In order to access Compute Canada resources, hosted across Canada by our regional partners ACENET, Calcul Québec, Compute Ontario, and WestGrid, users must register with the CCDB.

Any faculty member at a Canadian university is eligible to register and, once registered, can sponsor the students, research staff, and collaborators working on their research projects.

Researchers coming from government, non-profits, and industry may also be eligible to use Compute Canada resources. These inquiries should be directed to info@compute canada.ca.

Compute Canada is funded by the Canadian Foundation for Innovation (CFI) and by provincial research and innovation initiatives across Canada.

Please sign in


Login:

Login can be either an email address, a Compute Canada Identifier (CCI), a Compute Canada Role Identifier (CCRI), or a Compute Canada username.

Password:

[Sign in](#) | [Forgot Password](#) | [Register](#)

© 2008-2019 Compute Canada || [email webmaster](#)



compute canada | calcul canada

English || Français Logged in as Jamie Rosner (CCI: xdn-485) || Logout

My Account - Resource Applications - FAQ - Browse -

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.

Servers from the former CLUMEQ and RQCHP are now available through Calcul Québec.

Consortium	New Account
ACENET	Apply
CAC	Apply
Calcul Québec	Apply
SHARCNET	Apply
SciNet	Apply
WestGrid	Apply

© 2008-2017 Compute Canada || [email webmaster](#)

How to connect to HPC systems?

Connect to a cluster:

❖ Linux:

ssh; X2Go

❖ Mac:

ssh, X2Go

❖ Windows:

Putty, MobaXterm, ...

HPC

- ☐ Connect
- ☐ Transfer files
- ☐ Compile
- ☐ Test
- ☐ Run
- ☐ Analyze Data
- ☐

Transfer files:

❖ Linux:

scp; sftp; rsync; ...

❖ Mac:

ssh, sftp; rsync; ...

❖ Windows:

WinScp, FileZilla,
MobaXterm, ...

Connect from Mac/Linux to Grex

Connect to Grex from a terminal:

```
$ ssh -Y username@grex.westgrid.ca
```

Password:

Some Information
Message of The Day

```
[username@bison ~]$
```

```
[username@tatanka ~]$
```

File system:

Home directory: /home/username/

Scratch directory: /global/scratch/username/

30 GB per user; 500 K files; Backup.

2 TB per user; 1 M files; No Backup.

Very Important:

- ✓ **Don't share** your password with anyone.
- ✓ **Don't send** your password by email.
- ✓ In case you forgot your password, it is possible to **reset it**.

Connect to national clusters

Connect from a terminal:

Cedar: \$ ssh -XY username@cedar.computecanada.ca

Graham: \$ ssh -XY username@graham.computecanada.ca

Beluga: \$ ssh -XY username@beluga.computecanada.ca

Niagara: \$ ssh -XY username@niagara.computecanada.ca

Compute Canada credentials:

- ❖ user name
- ❖ password (CCDB)

File system: /home and /scratch

Home directory: /home/username/ 50 GB per user; 500 K files; Backup.

Scratch directory: /scratch/username/ 20 TB per user; 1000 K files; No Backup.

File system: /project 1 TB per group; 500 K files; Backup.

/project/{project-number1}/{user01;user02; ...}

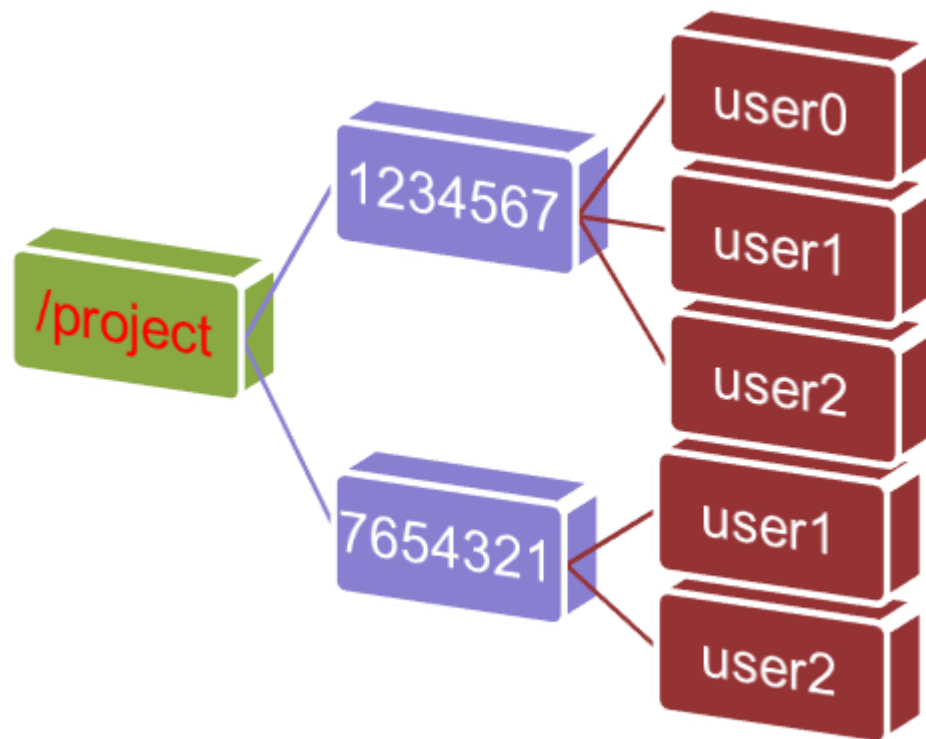
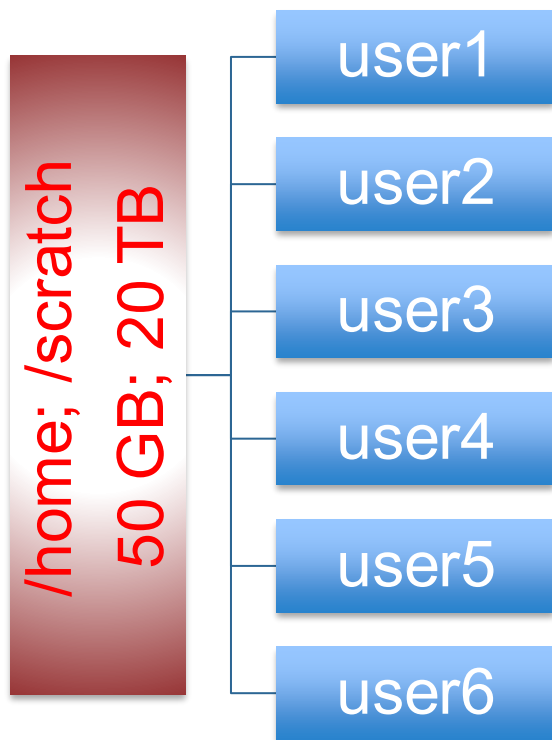
/home/{username}/projects/{project-id-1}/{username}

/project/{project-number2}/{user01;user02; ...}

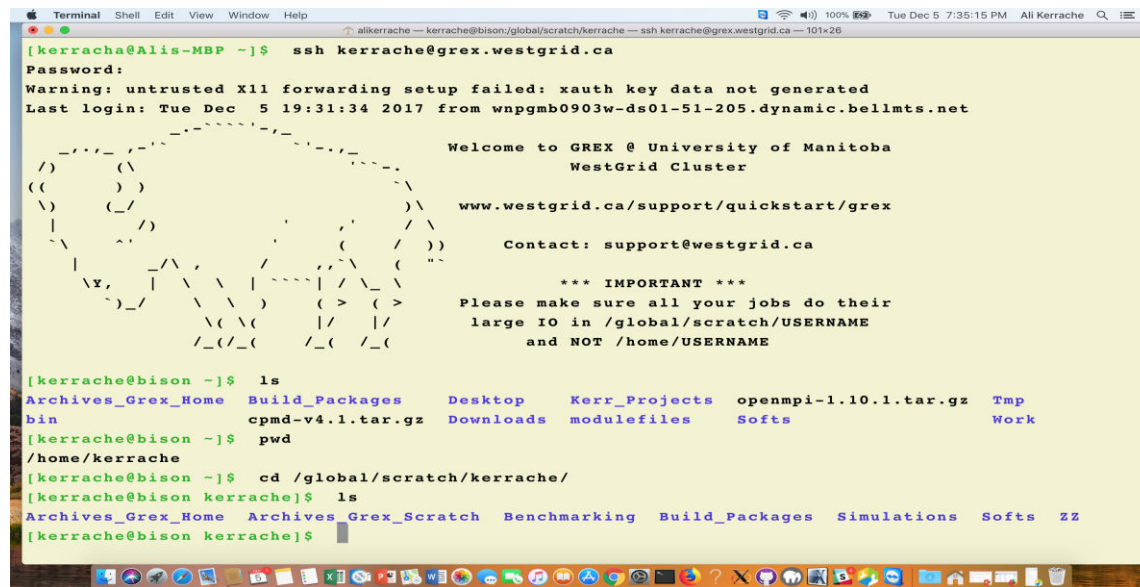
/home/{username}/projects/{project-id-2}/{username}

- ❖ Up to 10 TB on request from the PI.
- ❖ More than 10 TB: Requires RAC allocation.

File system structure



Connect from a terminal to Grex



```

[kerracha@Alis-MBP ~]$ ssh kerrache@grex.westgrid.ca
Password:
Warning: untrusted X11 forwarding setup failed: xauth key data not generated
Last login: Tue Dec 5 19:31:34 2017 from wnpqmb0903w-ds01-51-205.dynamic.bellmts.net

Welcome to GREX @ University of Manitoba
WestGrid Cluster

www.westgrid.ca/support/quickstart/grex

Contact: support@westgrid.ca

*** IMPORTANT ***
Please make sure all your jobs do their
large IO in /global/scratch/USERNAME
and NOT /home/USERNAME

[kerrache@bison ~]$ ls
Archives_Grex_Home  Build_Packages  Desktop  Kerr_Projects  openmpi-1.10.1.tar.gz  Tmp
bin                cpmv-v4.1.tar.gz  Downloads  modulefiles    Softs                Work
[kerrache@bison ~]$ pwd
/home/kerrache
[kerrache@bison ~]$ cd /global/scratch/kerrache/
[kerrache@bison kerrache]$ ls
Archives_Grex_Home  Archives_Grex_Scratch  Benchmarking  Build_Packages  Simulations  Softs  ZZ
[kerrache@bison kerrache]$
    
```

Short demonstration:

- Open a terminal
- Connect to a cluster:

- ❑ Grex: `ssh username@grex.westgrid.ca`
- ❑ cedar: `ssh username@cedar.computecanada.ca`
- ❑ graham: `ssh username@graham.computecanada.ca`
- ❑ beluga: `ssh username@beluga.computecanada.ca`

How to connect from Windows?

❖ Install ssh client:

- Putty: <http://www.putty.org/>
- MobaXterm: <https://mobaxterm.mobatek.net/>

❖ How to connect:

- ✓ Login: your user name
- ✓ Host: grex.westgrid.ca
- ✓ Password: **your password**
- ✓ Port: 22

Use the appropriate address:

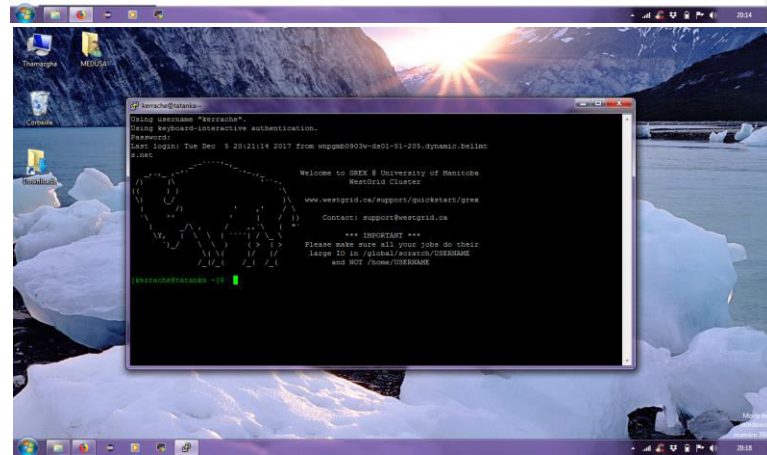
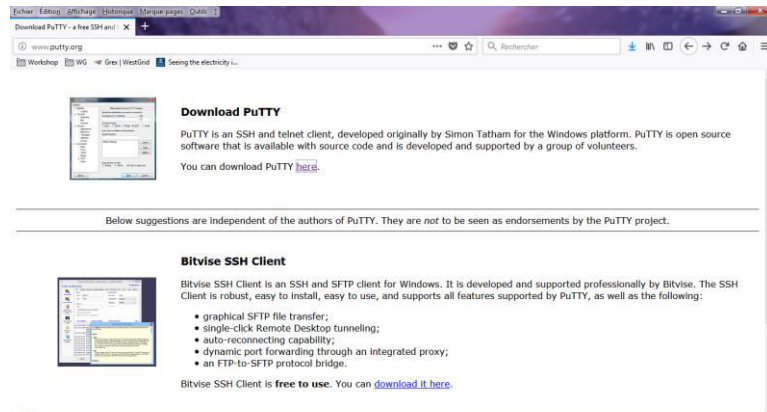
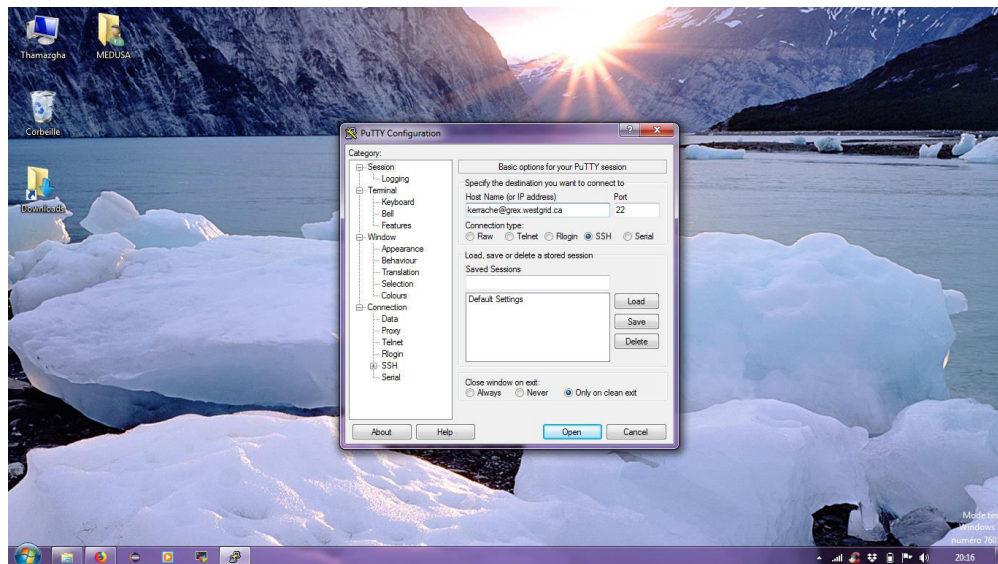
Grex: grex.westgrid.ca
Cedar: cedar.computeCanada.ca
Beluga: beluga.computeCanada.ca
Graham: graham.computeCanada.ca
Niagara: niagara.computeCanada.ca

❖ Use Cygwin: same environment as Linux

- <https://www.cygwin.com/>

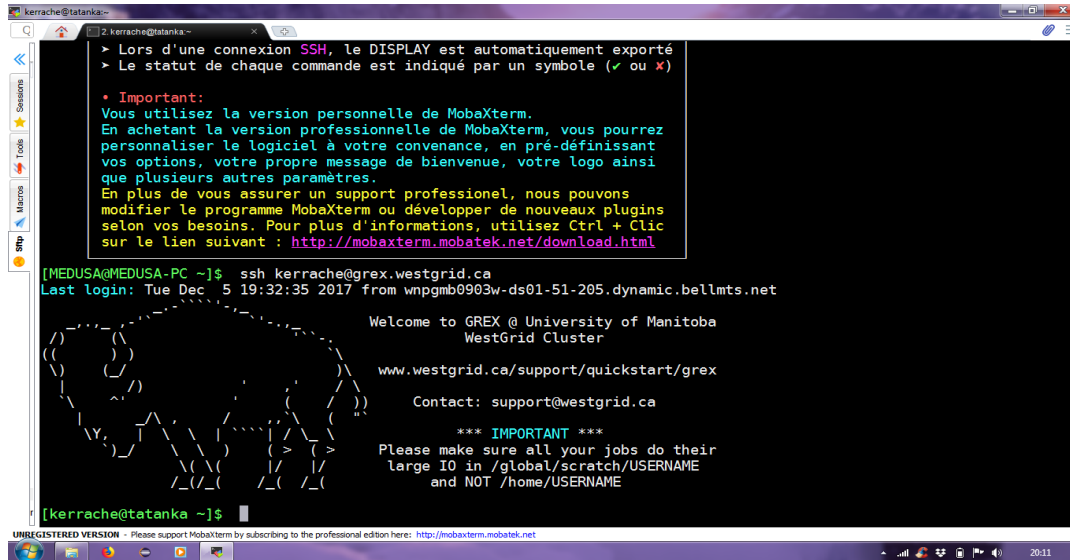
❖ How to use Putty?

- Download Putty.
- Install it on your laptop or Desktop
- Launch Putty and connect.



❖ How to use MobaXterm?

- *Download MobaXterm.*
- *Install it on your laptop or Desktop*
- *Launch MobaXterm and connect.*



```
kerrache@tatanka:~$ ssh kerrache@grex.westgrid.ca
Last login: Tue Dec 5 19:32:35 2017 from wnpqmb0903w-ds01-51-205.dynamic.bellmts.net

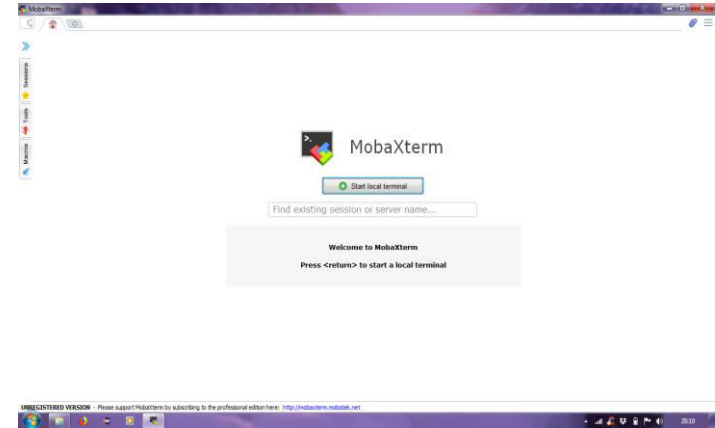
Welcome to GREX @ University of Manitoba
WestGrid Cluster

www.westgrid.ca/support/quickstart/grex

Contact: support@westgrid.ca

*** IMPORTANT ***
Please make sure all your jobs do their
large IO in /global/scratch/USERNAME
and NOT /home/USERNAME

[kerrache@tatanka ~]$
```



❖ How to connect:

- ✓ *Login: your user name*
- ✓ *Host: grex.westgrid.ca*
- ✓ *Password: your password*
- ✓ *Port: 22*

Use X2Go: Linux/Mac/Windows

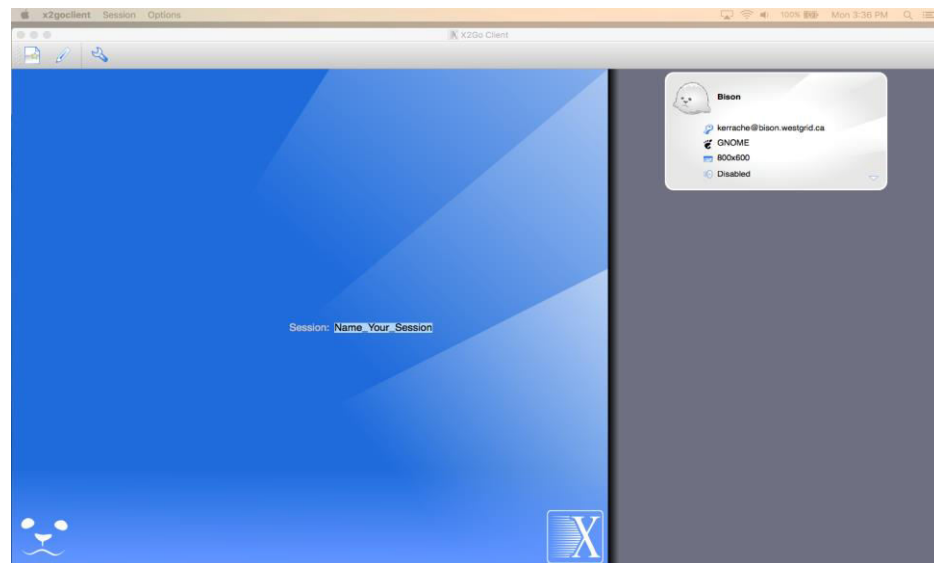
❖ **Why X2Go:** Access to GUI

❖ **How to use X2Go?**

- Ask first if X2Go is installed on the remote machine.
- If yes, install X2Go client on your laptop or Desktop.
- Linux, Windows, Mac (XQuartz)

- Launch X2Go.
- Create a session and connect.

- ✓ **Login:** **your user name**
- ✓ **Host:** **bison.westgrid.ca**
- ✓ **Port:** **22**
- ✓ **Session:** **GNOME**



Use X2Go: Linux/Mac/Windows

❖ **Why X2Go:** access to GUI

❖ **How to use X2Go?**

➤ Ask first if X2Go is installed on the remote machine.

➤ If yes, install X2Go client on your laptop or Desktop.

➤ Linux, Windows, Mac (XQuartz)

➤ Launch X2Go.

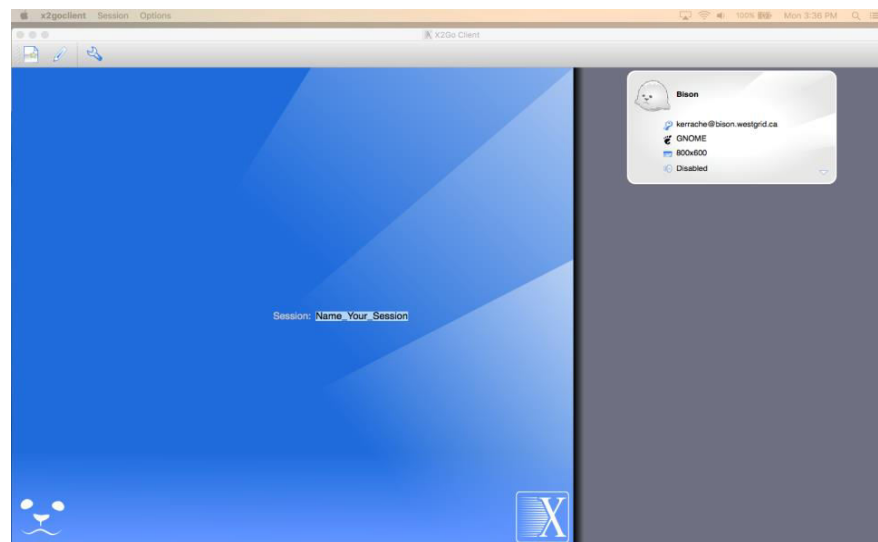
➤ Create a session and connect.

✓ **Login:** your user name

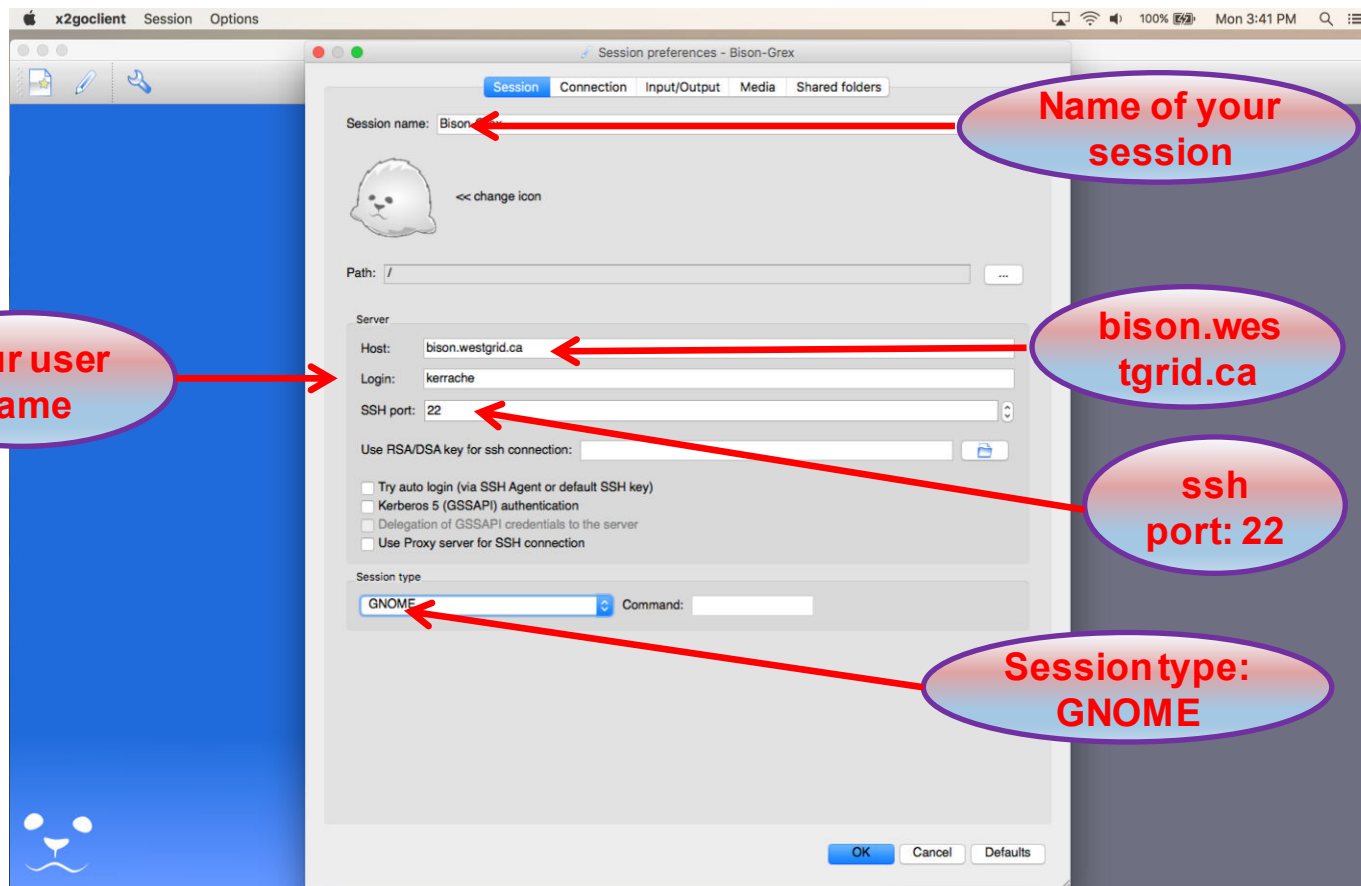
✓ **Host:** bison.westgrid.ca

✓ **Port:** 22

✓ **Session:** GNOME



Create a session with X2Go



The screenshot shows the 'Session preferences' dialog box for a session named 'Bison-Grex'. The dialog has tabs for 'Session', 'Connection', 'Input/Output', 'Media', and 'Shared folders'. The 'Session' tab is active. The 'Session name' field is set to 'Bison-Grex'. Below it is a section for the session icon, showing a seal icon and a '<< change icon' button. The 'Path' field is set to '/'. The 'Server' section contains fields for 'Host' (bison.westgrid.ca), 'Login' (kerrache), and 'SSH port' (22). There is a checkbox for 'Use RSA/DSA key for ssh connection' and a list of authentication options: 'Try auto login (via SSH Agent or default SSH key)', 'Kerberos 5 (GSSAPI) authentication', 'Delegation of GSSAPI credentials to the server', and 'Use Proxy server for SSH connection'. The 'Session type' dropdown is set to 'GNOME'. The 'Command' field is empty. At the bottom are 'OK', 'Cancel', and 'Defaults' buttons. Red arrows point from text labels to specific fields: 'Name of your session' points to the 'Session name' field; 'Your user name' points to the 'Login' field; 'bison.westgrid.ca' points to the 'Host' field; 'ssh port: 22' points to the 'SSH port' field; and 'Session type: GNOME' points to the 'Session type' dropdown.

Session preferences - Bison-Grex

Session | Connection | Input/Output | Media | Shared folders

Session name: Bison-Grex

<< change icon

Path: /

Server

Host: bison.westgrid.ca

Login: kerrache

SSH port: 22

Use RSA/DSA key for ssh connection:

☐ Try auto login (via SSH Agent or default SSH key)

☐ Kerberos 5 (GSSAPI) authentication

☐ Delegation of GSSAPI credentials to the server

☐ Use Proxy server for SSH connection

Session type

GNOME

Command:

OK Cancel Defaults

Name of your session

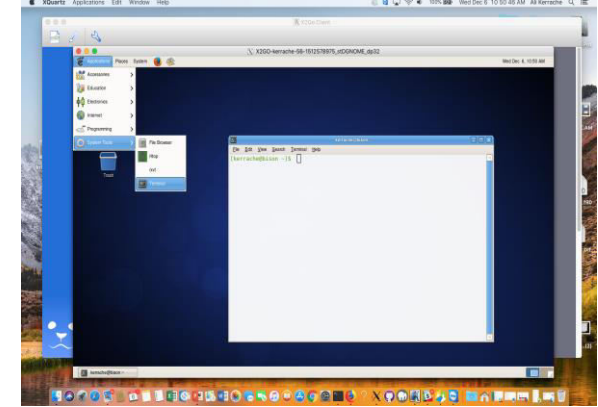
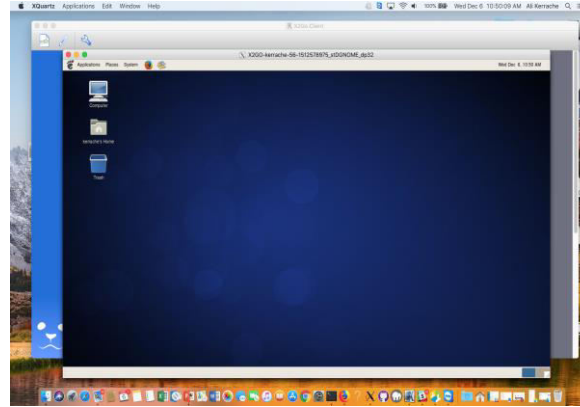
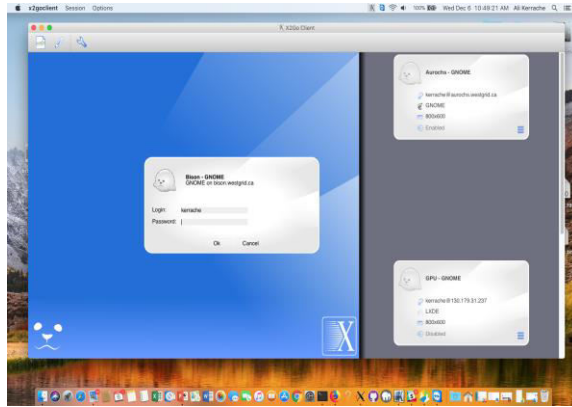
Your user name

bison.westgrid.ca

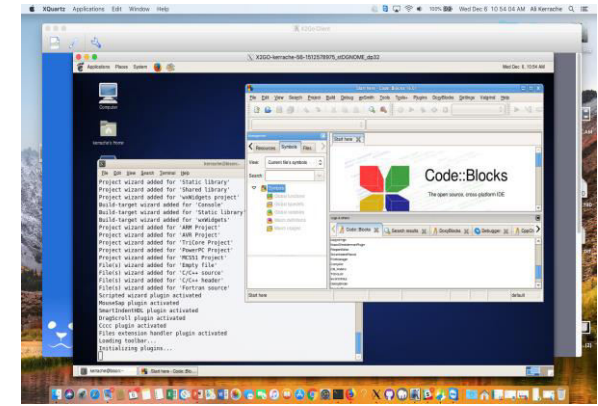
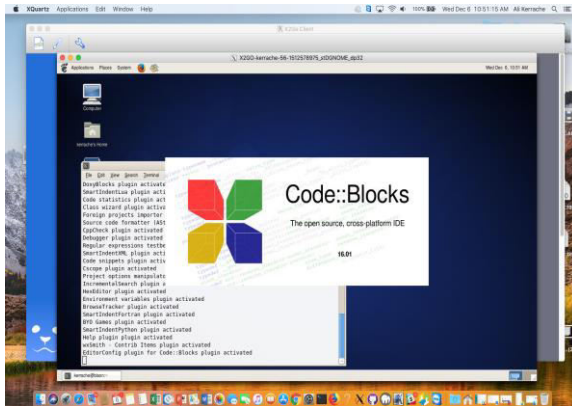
ssh port: 22

Session type: GNOME

Using a session with X2Go



- ❖ **Using X2Go:**
- Launch X2Go.
- Create a session.
- Connect.
- Start a terminal.
- Load a module.

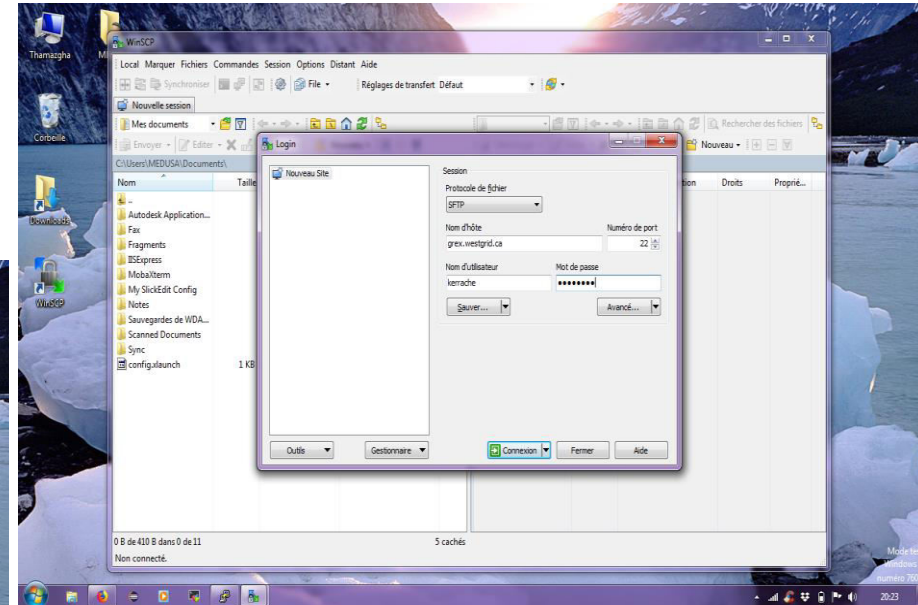
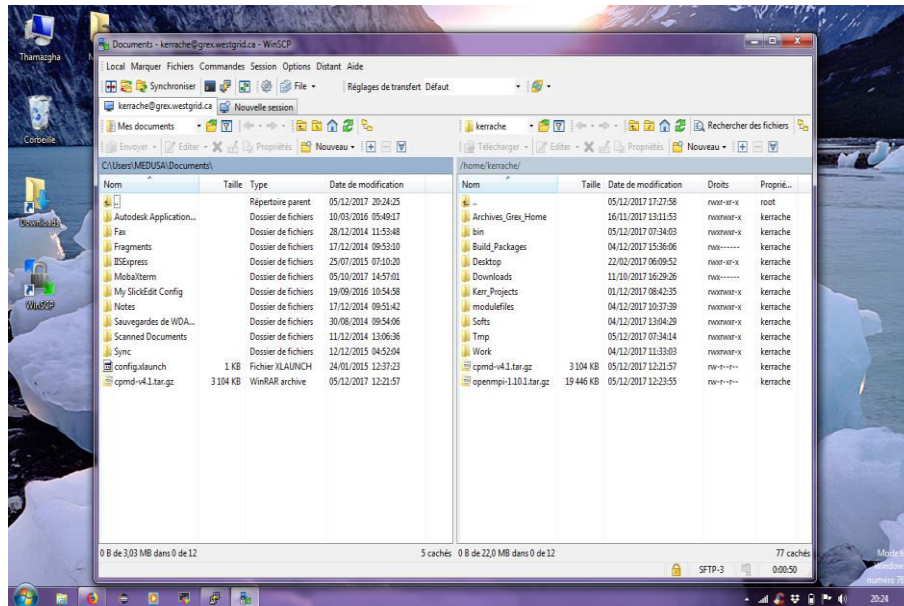


File Transfer: scp; rsync

- ❖ **Terminal:** Linux; Mac; CygWin; MobaXterm.
- ❖ **Check if `scp`; `sftp`; `rsync` are supported.**
- ❑ **Syntax for `scp`:** `scp [Options] [Target] [Destination]`
- ❑ **Syntax for `rsync`:** `rsync [Options] [Target] [Destination]`
- **Options:** for details use `man scp` or `man rsync` from your terminal.
- **Target:** Files or directory to copy (exact path).
- **Destination:** Where to copy the files (exact path).
- ❖ **Path on remote machine:** examples of a path on Grex.
 - `username@grex.westgrid.ca:/home/username/{Your_Dir}`
 - `username@grex.westgrid.ca:~/{Your_Dir}`
 - `username@grex.westgrid.ca:/global/scratch/username/{Your_Dir}`

File Transfer: WinScp, FileZilla

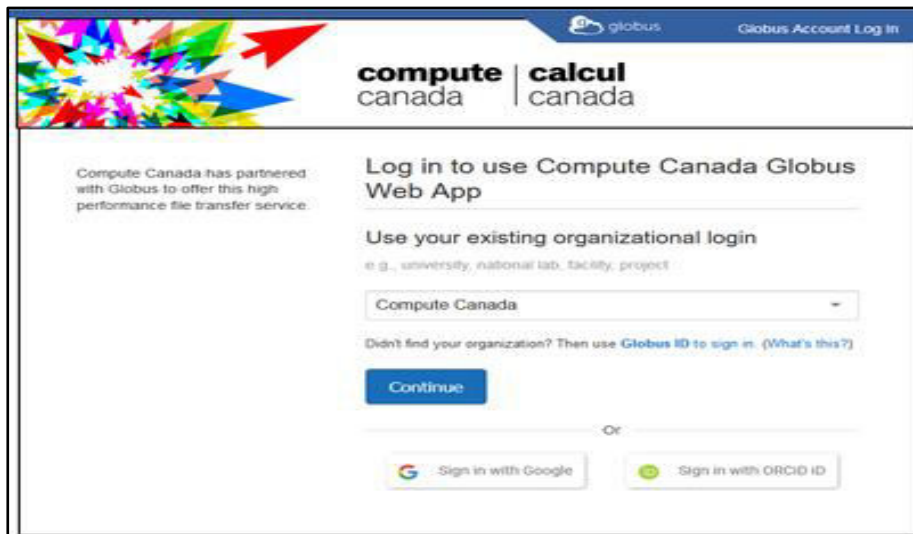
- Install WinScp or FileZilla.
- Launch the program.
- Connect with your credentials.



- Navigate on your local machine.
- Navigate on remote machine.
- Copy your files (both ways).

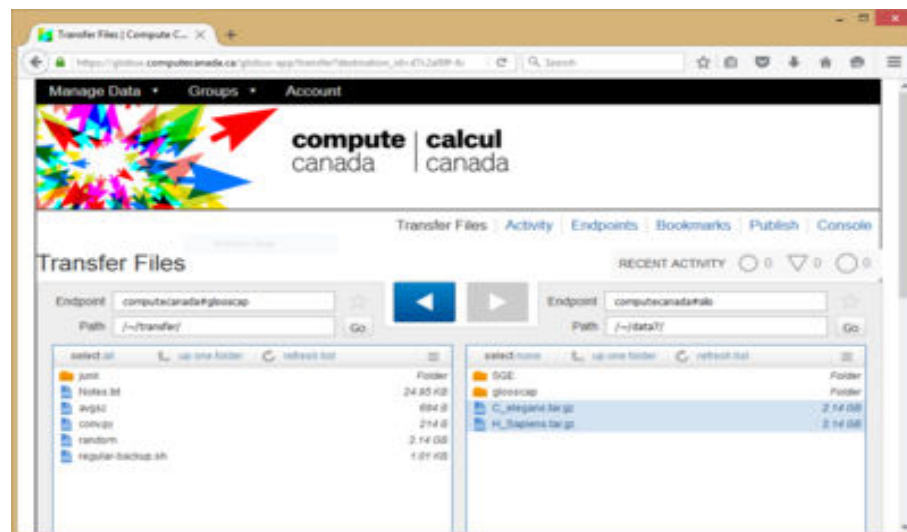
File Transfer: Globus

- Globus is a service for fast, reliable, and secure data transfer.
- Automatically tuning transfer settings, restarting interrupted transfers, and checking file integrity.



For more information:

<https://docs.computeCanada.ca/wiki/Globus/en>



Software Environment in HPC

- ❖ Local installation: `/home/$USER`; `/global/scratch/$USER`
Up to a user (experienced user); Analysts (User's consent).
- ❖ Central Installation: `/global/software/`; `/cvfms`; ...
Maintained by Analysts, Versions controlled via modules.
- ❖ Compilers and other applications:
Intel and GNU compilers; Tools; Libraries; Visualization; ...

Software Environment in HPC

User layer: Python packages, Perl and R modules, home made codes, ...

Easybuild layer: modules for Intel, PGI, OpenMPI, CUDA, MKL, high-level applications. Multiple architectures (sse3, avx, avx2, avx512)

`/cvmfs/soft.computecanada.ca/easybuild/{modules,software}/2017`

Nix layer: GNU libc, autotools, make, bash, cat, ls, awk, grep, etc.

`module nixpkgs/16.09 => $EBROOTNIXPKGS=`

`/cvmfs/soft.computecanada.ca/nix/var/nix/profiles/16.09`

Gray area: Slurm, Lustre client libraries, IB/OmniPath/InfiniPath client libraries (all dependencies of OpenMPI).

In Nix layer, but can be overridden using `PATH` & `LD_LIBRARY_PATH`.

OS: kernel, daemons, drivers, libcuda, anything privileged (e.g. the `sudo` command): always local. Some legally restricted software too (VASP).

Which Software to use?

- ❖ Home made programs, scripts and tools.

Up to a user, ...

- ❖ Free Software: GNU Public License.

Open Source, Binaries, ...

- ❖ Commercial Software:

Contact us with some details about the license, ...

We install the program and protect it with a UNIX group.

Software stack on new systems

- ❖ CVMFS: CERN Virtual Machine File System.

All software installed on cvmfs is mounted on new systems.

- ❖ RSNT: Research Support National Team.

➤ Install and Maintain the software stack on new systems.

➤ Documentation.

- ❖ Free software accessible via modules.

- ❖ Commercial software protected by POSIX groups and available only for the license holders.

Using Modules in HPC

❑ Why use modules?

- ❖ Control different versions of the same program.
- ❖ Avoid conflict between different versions and libraries.
- ❖ Set the right path to each program or library.

- ❖ Set some default modules:
 - Intel Compilers, GNU compilers, Other applications.
 - Open MPI, ...

Modules: some commands

- ❖ module **list**
- ❖ module **avail**
- ❖ module **load** soft/version; module **unload** soft/version
- ❖ module **show** soft/version; module **help** soft/version
- ❖ module **purge**
- ❖ module **use** ~/modulefiles; module **unuse** ~/modulefiles

Short demonstration on Grex and Cedar or Graham

Running and submitting jobs

- Connect to a cluster: `ssh`, Putty, MobaXterm, X2Go.
- Prepare your input: use an editor to change your files.
- Upload or Transfer your files (if prepared elsewhere):
`scp`; `stfp`; WinScp; FileZilla; Globus; ...
- Compile your own code (ask for support if needed):
Intel or GNU compilers; Libraries; Tools; ...
- Use existing modules and programs:
- Or ask support team to install new ones if necessary
- Prepare and Test your scripts and programs.
- Read about the scheduler: what is a scheduler?

Why do we need a scheduler?

❖ High Performance Computing:

- ✓ Multiple **users** and **different Allocations**: RAC; Default, ...
- ✓ Multiple **applications**.
- ✓ Multiple **jobs**: each job has a particular set of resources.

❖ What happened if there is no scheduler?

- **Over usage** for some resources.
- Abuse from users: others will never get a chance to run a job.
- Over-subscribing the resources.
- **Failure** of compute nodes.
- **Efficiency of the cluster**: some nodes can be over-loaded while others stay idle.

What a scheduler does?

- ❖ Put the jobs on the queue: one or more queues.
- ❖ Assign a priority to each job:
 - ✓ Resources asked for: **mem, procs, nodes, wall time, ...**
 - ✓ Recent usage: on Grex, **we consider the last three weeks.**
 - ✓ RAC or Default allocation.
- ❖ Assign a status to a job: **Q** (queue), **R** (running), **H** (hold)
- ❖ Run the job when the resources are available.
- ❖ Report some stats about the jobs: mem, ...
- ❖ Remove the job from the queue when it is done or exceeded the wall time.

Which scheduler to use?

- ❑ **GreX:** Torque and Moab (PBS scripts)
- ❑ **Cedar, Graham and Beluga:** SLURM
- ❖ There are other schedulers.
- ❖ Each system has one scheduler: consult the Guide Start
- ❖ Read the documentation about the scheduler before starting any job.
- Even if the schedulers are different, the idea is the same:
 - ✓ Submit and run jobs.
 - ✓ Each scheduler has its **own syntax**.

Structure of a PBS script

```
#!/bin/bash
```

```
#PBS -S /bin/bash
```

Add the resources and some options

```
cd ${PBS_O_WORKDIR}
```

```
echo "Current working directory is `pwd`"
```

```
echo "Starting run at: `date`"
```

Load appropriate modules.

command line to run your program.

```
echo "Program finished with exit code $? at: `date`"
```

File:

test_job.pbs

To adjust for
each type of
job to submit

Submit the job using the command: **qsub** test_job.pbs

Most used PBS commands

PBS command	Description
#PBS -S /bin/bash	Set the shell that the job will be executed on compute node.
#PBS -l walltime=12:00:00	Wall time of 12 hours (max of 240 hours).
#PBS -l procs=1 #PBS -l nodes=1:ppn=1	Requests one core (serial job).
#PBS -l procs=48	Request 48 cores: MPI job (OpenMPI).
#PBS -l mem=4000mb	Total memory asked for.
#PBS -l pmem=2500mb	Memory per core (Total mem = pmem * Nc).
#PBS -l nodes=1:ppn=12	Multi-threading (OpenMP) programs.
#PBS -l nodes=4:ppn=12	4 nodes and 12 cores per node = 48 cores.

Most used PBS commands

PBS command	Description
#PBS -M <valid email>	Sets the email address for sending notifications about your job state.
#PBS -m abe	Sets the scheduling system to send you email: a: when the job is aborted by the batch system. B: when the job begins execution. e: when the job terminates.
#PBS -A <Acc. group>	Requests to use a specific accounting group
#PBS -o diffuse_job.out #PBS -e diffuse_job.err	standard error and output streams are directed to two separate files
#PBS -N JOB_NAME	Job name.

https://www.westgrid.ca/support/running_jobs

PBS environment variables

PBS command	Description
PBS_O_WORKDIR	Job's submission directory
PBS_O_HOME	Home directory of submitting user
PBS_JOBID	Unique PBS job id
PBS_NUM_NODES	Number of nodes allocated to the job
PBS_NUM_PPN	Number of procs per node allocated to the job
PBS_NODEFILE	File containing line delimited list on nodes allocated to the job
PBS_TASKNUM	Number of tasks requested
PBS_JOBNAME	User specified job name

https://www.westgrid.ca/support/running_jobs

How to monitor your jobs?

PBS command	Description
<code>qstat -a</code>	List all jobs on the system and their state.
<code>qstat -r</code>	List all running jobs on the system.
<code>qstat -u \$USER</code>	List all the jobs under user's account.
<code>showq -r</code>	Show all running jobs.
<code>showq -r -u \$USER</code>	Show all running jobs under user's account.
<code>qstat -f <job_id></code>	List detailed information on Job <job_id>
<code>checkjob <job_id></code>	List detailed information on Job <job_id>
<code>jobinfo -u \$USER</code>	Stats about CPU usage of the user's group.

Short demonstration on Grex

SLURM on new systems

```
#!/bin/bash
```

```
#SBATCH --account=def-some_user
```

Add the resources and some options

```
echo "Current working directory is `pwd`"  
echo "Starting run at: `date`"
```

Load appropriate modules.
command line to run your program.

```
echo "Program finished with exit code $? at: `date`"
```

File:
test_job.slurm

To adjust for
each type of
job to submit

Submit the job using the command: **sbatch** test_job.slurm

SLURM: some commands

SLURM command	Description
#SBATCH --account=def-some_user	Use the accounting group for jobs.
#SBATCH --time=0-00:05	Wall time in the format: DD-HH:MM
#SBATCH --ntasks=4	Requests 4 cores for MPI job
#SBATCH --nodes=2	Requests 2 whole nodes
#SBATCH --ntasks-per-node=32	32 (graham); 40 (beluga), 48 (cedar)
#SBATCH --mem=4000	Memory of 4GB for the job
#SBATCH --mem-per-cpu=2000M	Memory of 2GB per CPU
#SBATCH --cpus-per-task=4	Number of threads (OpenMP)
#SBATCH --job-name="JOB_NAME"	Job name.
#SBATCH --output=job_output.txt	Standard output.

Monitor your jobs with SLURM

-
- `squeue -u $USER [-t RUNNING] [-t PENDING]` # list all current jobs.
- `squeue -p partitionName` # list all jobs in a partition.
- `sinfo` # view information about Slurm partitions.
- `sacct -j jobID --format=JobID,MaxRSS,Elapsed` # resources used by completed job.
- `sacct -u $USER --format=JobID,JobName,AveCPU,MaxRSS,MaxVMSize,Elapsed`
- `scontrol show job jobID` # produce a very detailed report for the job.
- `sprio [-j jobID1,jobID2] [-u $USER]` # list job priority information.
- `sshare -U --user $USER` # show usage info for user.
- `sinfo --states=idle` # show idle node(s) on cluster.
- `scancel [-t PENDING] [-u $USER] [jobID]` # kill/cancel jobs.

Torque vs SLURM: serial job

File: serial_job.pbs

```
#!/bin/bash
#PBS -S /bin/bash
#PBS -l procs=1
#PBS -l mem=2500mb
#PBS -l walltime=24:30:00

cd $PBS_O_WORKDIR
echo "Starting run at: `date`"
module load lammps/11aug17
Imp_grex < in.lammps
echo "Program finished at: `date`"
```

qsub serial_job.pbs

File: serial_job.slurm

```
#!/bin/bash

#SBATCH --ntasks=1
#SBATCH --mem=2500M
#SBATCH --time=1-00:30

echo "Starting run at: `date`"
module load lammps/20170331
Imp < in.lammps
echo "Program finished at: `date`"
```

sbatch serial_job.slurm

Torque vs SLURM: OpenMP job

File: omp_job.pbs

```
#!/bin/bash
#PBS -S /bin/bash
#PBS -l nodes=1:ppn=8
#PBS -l pmem=2500mb
#PBS -l walltime=24:30:00
cd $PBS_O_WORKDIR
echo "Starting run at: `date`"
export OMP_NUM_THREADS=8
module load lammps/11aug17
Imp_grex < in.lammps
echo "Program finished at: `date`"
```

qsub omp_job.pbs

File: omp_job.slurm

```
#!/bin/bash

#SBATCH --cpus-per-task=8
#SBATCH --mem-per-cpu=2500M
#SBATCH --time=1-00:30

echo "Starting run at: `date`"
export OMP_NUM_THREADS=8
module load lammps/20170331
Imp < in.lammps
echo "Program finished at: `date`"
```

sbatch omp_job.slurm

Torque vs SLURM: MPI job

File: `mpi_job.pbs`

```
#!/bin/bash
#PBS -S /bin/bash
#PBS -l procs=16
#PBS -l pmem=2500mb
#PBS -l walltime=12:30:00

cd $PBS_O_WORKDIR
echo "Starting run at: `date`"
module load lammps/11aug17
mpiexec lmp_grex < in.lammps
echo "Program finished at: `date`"
```

qsub `mpi_job.pbs`

File: `mpi_job.slurm`

```
#!/bin/bash

#SBATCH -ntasks=16
#SBATCH --mem-per-cpu=2500M
#SBATCH --time=0-12:30

echo "Starting run at: `date`"
module load lammps/20170331
srun lmp < in.lammps
echo "Program finished at: `date`"
```

sbatch `mpi_job.slurm`

Torque vs SLURM: GPU job

File: gpu_job.pbs

```
#!/bin/bash
#PBS -S /bin/bash
#PBS -l node=1:ppn=12:gpus=3
#PBS -l mem=2500mb
#PBS -l walltime=24:30:00
cd $PBS_O_WORKDIR
echo "Starting run at: `date`"
module load cuda
module load [some module]
gpu_program
echo "Program finished at: `date`"
```

qsub gpu_job.pbs

File: gpu_job.slurm

```
#!/bin/bash
#SBATCH --nodes=3
#SBATCH --gres=gpu:1
#SBATCH --mem=4000M
#SBATCH --time=1-00:30

echo "Starting run at: `date`"
module load cuda
module load [some module]
gpu_program
echo "Program finished at: `date`"
```

sbatch gpu_job.slurm

Useful Links

- ❖ WestGrid: <https://www.westgrid.ca/>
- ❖ Grex: <https://www.westgrid.ca/support/quickstart/Grex>
- ❖ WG Start Guide: https://www.westgrid.ca/support/quickstart/new_users
- ❖ Software on WG: <https://www.westgrid.ca/support/software>
- ❖ Compute Canada:
https://docs.compute canada.ca/wiki/Compute_Canada_Documentation
- ❖ CCDB: <https://ccdb.compute canada.ca/security/login>
- ❖ CC Software: https://docs.compute canada.ca/wiki/Available_software
- ❖ Running Jobs: https://docs.compute canada.ca/wiki/Running_jobs
- ❖ PuTTY: <http://www.putty.org/>
- ❖ MobaXterm: <https://mobaxterm.mobatek.net/>
- ❖ X2Go: <https://wiki.x2go.org/doku.php>
- ❖ Help and support on CC: support@compute canada.ca
- ❖ WG training material: <https://westgrid.github.io/trainingMaterials/>