Objectives

Compute Canada and other campus facilities:

- Overview
- How to access
- How to use

HPC in Canada

In January 2005, a national long-range plan (LRP) for HPC across Canada was proposed by c3.ca, which was at the time the national advocacy group for HPC.

This plan envisioned creating a sustained, world-class, physical and human infrastructure for computation-based research.

In July 2005, the Canadian Foundation for Innovation (CFI) announced a National Platforms Fund competition to fund the LRP for HPC.

In December 2006, CFI announced $60M over 3 years for HPC equipment to the newly formed *Compute Canada* (plus $18M in infrastructure operating and $10M from NSERC for personnel).

Compute Canada is now the over-arching structure that contains the regional consortia; c3.ca disbanded in late 2007.

WestGrid

Before Compute Canada, shared HPC infrastructure in Canada was divided among 7 regional consortia: WestGrid, SHARCNET, SciNet, HPCVL, RQCHP, CLUMEQ, and AceNet.

Very recently, RQCHP and CLUMEQ have amalgamated to form Calcul Quebec.

WestGrid presently consists of 14 partner institutions across the provinces of BC/AB/SK/MB.

Of these, there are 7 major partners (UVic, UBC, SFU, UofA, UofC, UofS, and UofM) that host the various pieces of shared infrastructure.

All Computer Canada partners have virtual collaboration facilities, and most have advanced visualization capabilities.

The shared infrastructure is mainly for supercomputing; Compute Canada is well-networked so that users can access resources regardless of where they are located.

The types of supercomputing facilities are (commodity) clusters, clusters with fast interconnect, and shared-memory systems.

There are also five GPGPU-based systems (two on WestGrid (one large and one small) and one on each of SHARCNET, SciNet, and RQCHP).

The UofS hosts a data storage centre, the primary storage facility for WestGrid and one of the largest in Canada, with over 3 PB of disk storage and 3 PB of tape storage.

WestGrid also has some licenses for popular software packages such as MATLAB, Gaussian (chemistry), and OpenFOAM (CFD).

Information on all the available software can be found at

http://www.westgrid.ca/support/software

UofS HPC Training Clusters

The shared national infrastructure is a fantastic resource for running HPC jobs, but it is not well-suited for training or code development.

To aid UofS researchers make use of the national HPC resources as well as to complement individual research clusters, ITS has made available two machines for

- training of HQP in theory and implementation of parallel programming and parallel programs
- parallel code development / testing / debugging for research (called "staging")

These machines are not intended to replace researcher clusters or Compute Canada resources.

You should have access to the training cluster machines by virtue of your enrollment in this course; login is with your nsid.

UofS Compute Cluster (`socrates`)

In May 2009, ITS commissioned a 37-node HPC cluster named `socrates` that has

- 1 head node (`socrates.usask.ca`)
- 8 capability nodes (`compute-0-0` to `compute-0-7`)
- 28 capacity nodes (`compute-0-8` to `compute-0-35`)

The designated use for `socrates` is for distributed-memory programs (1 Gigabit Ethernet interconnect).

Compilers available are `gcc`, `g77`, `gfortran`, `ifort`, and `icc` as well as the wrappers `mpicc` and `mpif77`.

The operating system is CentOS 6.2 with the ROCKS cluster management system.

MATLAB and Mathematica are also available.

Jobs are submitted through a batching system (TORQUE/Maui).

UofS Large-Memory System (`moneta`)

In September 2009, ITS commissioned a large-memory machine called `moneta` that has

- 4 Intel Xeon E7430 quad-core processors (16 cores),
- 256 GB RAM,
- 64-bit RedHat Enterprise Linux 5.4,
- 500 GB of scratch disk for storing intermediate data.

The designated use of `moneta` is for large shared-memory programs.

Compilers available are `gcc`, `g77`, and `gfortran`, all available in `/usr/bin`.

Software available includes MATLAB, Mathematica, Maple, and R.

There is no queueing manager on `moneta`.

UofS GPU-Accelerated Cluster (zeno)

zeno is a cluster with GPU accelerators and high-speed (InfiniBand) inter-node connections.

There are 8 compute nodes, each with 12 cores, 24GB of RAM, and a Tesla M2075 GPU processor.

To take advantage of the GPUs, a program on zeno must be compiled with the CUDA libraries.

The CUDA 4.2 environment is available on zeno.

zeno can stage for systems like parallel.westgrid.ca, WestGrid's main GPU cluster.

The module command controls which environments are loaded at each shell invocation for the default installed versions of mpi on the cluster.

The initadd command adds the appropriate modules:

```
module initadd openmpi/1.6 nvidia/cuda
```

UofS WestGrid Collaboration and Visualization Facility (AG 2D71)

Commissioned in September, 2009, this facility is designed to support advanced visualization and remote research collaborations.

Collaboration technologies include AccessGrid / H.323 videoconferencing and teleconferencing.

Facility allows for effective collaboration between a few researchers or 20+ people for remote presentations, e.g., WestGrid and Coast2Coast Seminar Series.

A CyViz dual-projector system and dedicated computer will enable stereo 3D visualization of research data. Remote visualization from visualization clusters located at other institutions is also easily supported.

Equipment to support this collaboration technology includes multiple dedicated servers, 4 video cameras, 3 high-resolution projectors, an 18-foot custom screen, echo cancellation audio system, and wireless microphones and speaker phone.

Access to Compute Canada resources

You are eligible for a Compute Canada account if you are associated with an approved research project.

In general, any academic researcher from a Canadian research institution with significant HPC research requirements may apply for a Compute Canada account. A project description is required.

Students require sponsorship from a faculty supervisor, i.e., by joining an approved project.

There is a single point from which requests for Compute Canada accounts are generated and approved (e.g., see the WestGrid website).

Identical accounts are then "automatically" created on the various Compute Canada clusters.

Once an account is created, users can then log in, transfer files, etc., to any Compute Canada machine using a standard protocol such as `ssh`, `scp`, etc., as they would with any other UNIX workstation.

Use of Compute Canada resources

Each major partner in Compute Canada has a *Site Lead*, a technically oriented person who oversees the operation and maintenance of the shared infrastructure and provides a local point of contact.

At the UofS, the Site Lead is Jason Hlady.

Jason is available to help with anything from finding more about Compute Canada resources to setting up and using and account to helping your programs run more efficiently (or at all!) on Compute Canada.

`jason.hlady@usask.ca`

All Compute Canada systems use a UNIX variant or Linux operating system.

As mentioned, work such as editing, compiling, testing, and debugging code may be done interactively on Compute Canada machines, but this is not a recommended practice; use the UofS HPC training resources `socrates`, `moneta`, and `zeno` instead.

The majority of the Compute Canada computing resources are available only for batch-oriented production computing.

In other words, users must use a UNIX shell scripting language to write *job scripts* to run their programs.

Job scripts are submitted to the batch-job handling system (or queue) for assignment to a machine. The results are reported to the user upon job completion.

There is often a significant time lag between job submission and assignment, so this is an extremely inefficient way to (for example) debug code.

Every user is given a default allocations to Compute Canada resources (access to CPUs and disk space).

An active user not requiring large memory or processor requirements would have access to 20–80 processors (depending on the machine) on a fairly regular basis.

Researchers desiring more than their default allocation for their work must submit a request for more resources to the *Resource Allocation Committee* (RAC).

Running batch jobs

The system software that handles batch jobs consists of two pieces: a resource manager (TORQUE) and a scheduler (Moab).

Batch job scripts are UNIX shell scripts (basically text files of commands for the UNIX shell to interpret, similar to what you could execute by typing directly at a keyboard) containing special comment lines that contain *TORQUE directives*.

TORQUE evolved from software called Portable Batch System (PBS).

So TORQUE directive lines begin with #PBS, some environment variables contain "PBS", and the script files typically have a .pbs suffix (although not required).

Note: There are small differences in the batch job scripts, particularly for parallel jobs, among the various Compute Canada systems! See specific instructions for individual machines.

Example: Job script `diffuse.pbs` for a serial job on glacier to run a program named `diffuse`.

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

# Script for running serial program, diffuse, on glacier

cd $PBS_O_WORKDIR
echo "Current working directory is ‘pwd‘"

echo "Starting run at: ‘date‘"
./diffuse
echo "Job finished with exit code $? at: ‘date‘"
```

To submit the script `diffuse.pbs` to the batch job handling system, use the `qsub` command:

```
qsub diffuse.pbs
```

If a job will require more than the default memory or time (typically 3 hours) allocation, additional arguments may be added to the `qsub` command.

If `diffuse` is a parallel program, the number of nodes on which it is to run must be specified, e.g.,

```
qsub -l walltime=72:00:00, mem=1500mb, nodes=4 diffuse.pbs
```

When `qsub` processes the job, it assigns it a `jobid` and places the job in a queue to await execution.

The status of all the jobs on the system can be displayed using

```
showq
```

To show just the jobs associated with your user name, use

```
showq -u username
```

To delete a job from the queue (or kill a running job), use `qdel` with the `jobid` assigned from qsub:

```
qdel jobid
```

It is wise for programs to periodically save output to a file so you can see how they are doing (and restart from that point if necessary). This is called checkpointing.

Sometimes, e.g., if you need to confirm how much memory your job is using, you may have to send e-mail to `support@westgrid.ca` to request that an administrator check on the job for you.

Other useful commands: <command> *job.id*

`qstat`: examine the status of a job

`qalter`: alter a job (specify attributes)

`qhold`: put a job on hold

`qorder`: exchange order of two jobs (specify jobids)

`qrls`: release a job

`qsig`: send a signal to a job (specify signal)

Other PBS directives

```
# Set the name of the job (up to 15 characters,
# no blank spaces, start with alphanumeric character)
#PBS -N JobName

# make pbs interpret the script as a bash script
#PBS -S /bin/bash

# specify filenames for standard output and error streams
# By default, standard output and error streams are sent
# to files in the current working directory with names:
#     job_name.osequence_number  <-  output stream
#     job_name.esequence_number  <-  error stream
# where job_name is the name of the job and sequence_number
# is the job number assigned when the job is submitted.

#PBS -o stdout_file
#PBS -e stderr_file
```

```
# Specify the maximum cpu and wall clock time.
# cput  =
# pcput =
# The wall clock time should take queue waiting time into
# account.  Format:   hhhh:mm:ss   hours:minutes:seconds
# Be sure to specify a reasonable value here.
# If the job does not finish by the time reached,
# the job is terminated.

#PBS -l     cput=6:00:00
#PBS -l    pcput=1:00:00
#PBS -l walltime=6:00:00

# Specify the maximum amount of physical memory required.
# kb for kilobytes, mb for megabytes, gb for gigabytes.
# mem  = max amount of physical memory used by all processes
# pmem = max amount of physical memory used by any process
# vmem and pvmem are analogues for virtual memory
# Take care in setting this value.  Setting it too large
# can result in the job waiting in the queue for sufficient
# resources to become available.

#PBS -l mem=512mb
#PBS -l pmem=128mb
```

```
# PBS can send email messages to you about the
# status of your job.  Specify a string of
# either the single character "n" (no mail), or one or more
# of the characters "a" (send mail when job is aborted),
# "b" (send mail when job begins), and "e" (send mail when
# job terminates).  The default is "a" if not specified.
# You should also specify the email address to which the
# message should be send via the -M option.

#PBS -m abe
#PBS -M user_email_address

# Specify the number of nodes requested and the
# number of processors per node.

#PBS -l nodes=1:ppn=1

# Define the interval when job will be checkpointed
# in terms of an integer number of minutes of CPU time.

#PBS -c c=2
```

There is further help available for using PBS on WestGrid via the command `man pbs`.

User responsibilities

Compute Canada systems are *shared production HPC environments*; i.e., they are meant for running programs, not for developing code or learning how to use software.

Although some support is available, in practice users should know enough UNIX to transfer files, submit and monitor batch jobs, monitor disk usage, etc.

Users are expected to use Compute Canada resources responsibly!

Users should be able to estimate memory requirements (both RAM and disk) and run times for their jobs.

Code is expected to be optimized through appropriate choice of algorithm, compiler flags, and/or optimized numerical libraries.

Code is also expected to checkpoint.

How to use the UofS HPC resources

You should have accounts on `socrates`, `moneta` and `zeno` by virtue of being enrolled in this course.

For security reasons, these machines are on the UofS private network and so cannot be accessed directly from off campus; i.e., users can only connect to these machines from another machine in the `usask` domain (even if only through a virtual private network (VPN)).

Login is done using your UofS NSID and password, e.g.,

```
ssh abc123@socrates.usask.ca
```

Help is available by e-mailing `hpc_consult@usask.ca`.

More details are available through the University of Saskatchewan website.

Summary

- The HPC landscape in Canada and at the UofS
- Using HPC resources: from theory to practice