Introduction to ACENET & Grid Engine

Accelerating Discovery with Computational Research

computecanada
regional partner

Slides at http://www.ace-net.ca/training/workshops-seminars/

Who am I?

Oliver Stueker

Computational Research Consultant

with ACENET since July 2015

Before that:

- Computational Chemist
- HPC user since ~ 2001

Oliver.Stueker@ace-net.ca

ACENET
accelerate discovery

computecanada
regional partner

What is ACENET?

We accelerate discovery in Atlantic Canada through leadership and innovation in Advanced Research Computing infrastructure, expertise and training.

What is Advanced Computing?
Computing that takes too long, or is too complex for a desktop/laptop.

- **Free of charge** for researchers and industry-researcher collaborations.
- Funding - federal and provincial governments, some member institutions.
- Regional partner of Compute Canada.

Consortium Partners

Computing Resources

A wide array of High Performance Computing and storage systems

Big Data and Data Analytics tools and environments

Leading edge GPU computing systems

High speed, secure file transfer with the Globus Portal

Extensive software library

GenAP computing platform to access and use genomic datasets

Compute Canada's Cloud computing and environment

Data storage systems more stable and secure than your desktop

Desktop and mobile videoconferencing

Collaboration rooms at all ACENET partner institutions to deliver and view large videoconferences.

Expertise

Determining computing resources needed

Designing, optimizing and troubleshooting computer code

Customizing tools

In-depth collaboration where needed

Group and individual training from novice to advanced

Accessible and responsive support staff

Installing, managing and maintaining advanced research computing equipment

Access to Digital Humanities expertise

Access to 3D Visualization expertise

What is High Performance Computing

Many CPUs working on one problem

- Many “serial” jobs running at once; or
- Many CPUs working in close coordination.
- Communication between CPUs is key!

Advantages

- Faster results
- Larger problems or more detailed simulations

Principal tool: Computer **cluster**

ACENET Computers

Four clusters

7000 CPU cores

20 TB RAM

500 TB disk storage

aggregated numbers

Plus tape

Plus software

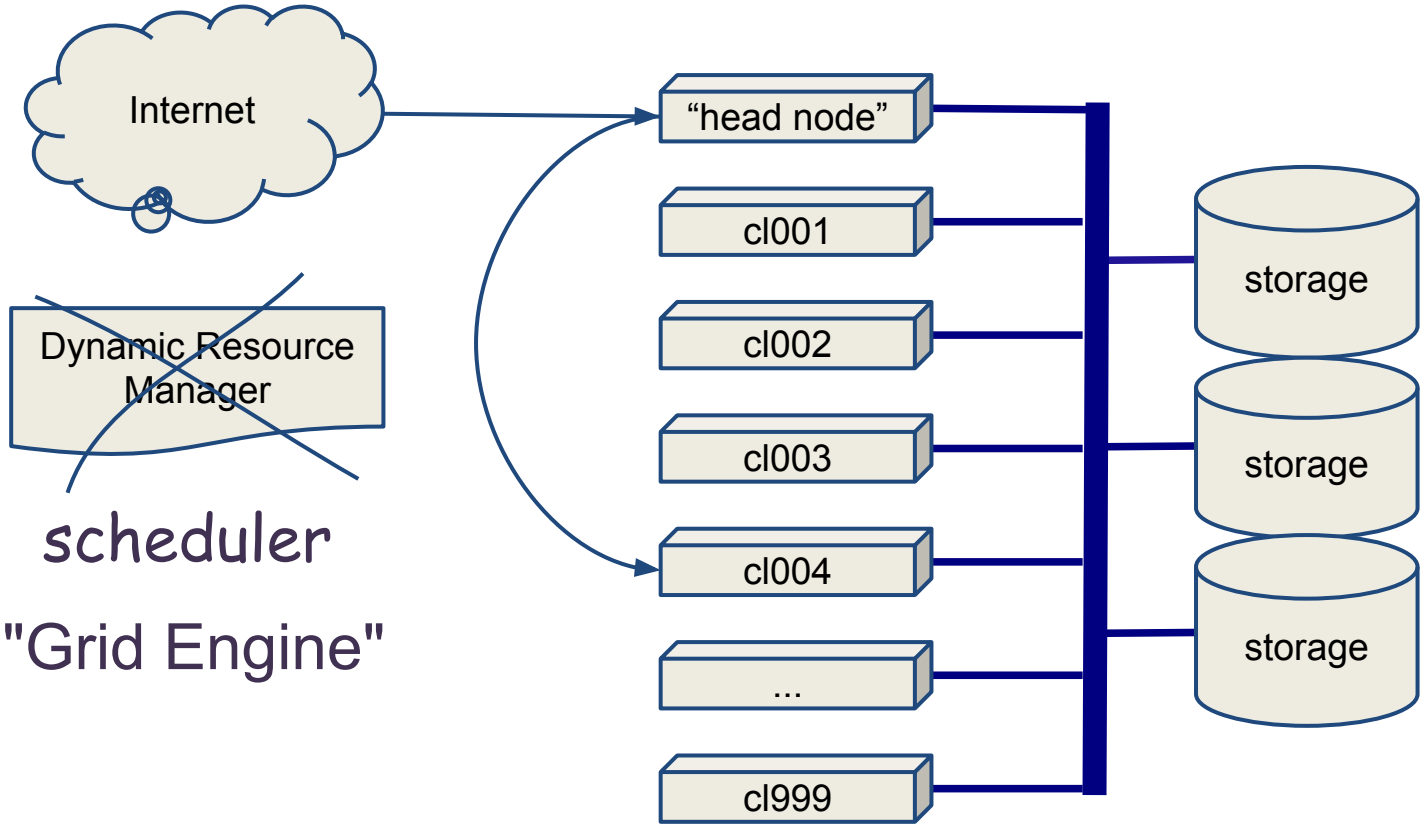
Remember: Shared resources!

About 720 users

ACENET
accelerate discovery

computecanada
regional partner

What is a Cluster?



A minimal job script

job_script.sh

```
./myprogram </path/to/input >output
```

There are also ways to specify

- how many CPUs you want
- how much memory
- how much time

How Do I Get Connected?

Logging in:

ssh *Secure Shell*

Moving Data Around:

sftp *Secure File Transfer Protocol*

scp *Secure Copy*

Both provided with Mac OS X or Linux

```
ssh -X username@mahone.ace-net.ca
sftp username@placentia.ace-net.ca
```

Windows?

Try MobaXterm or PuTTY and WinSCP. Just google for them

Applications

Molecular	**Math&CS**	**Engineering**	**Earth & Ocean**	**Bio**
Gaussian	Sage	ANSYS	CDO	AbySS
Gromacs	GAP	Fluent	NCO	Bowtie
Q-Chem	DiVinE	CFX	GMT	rapsearch2
VASP	...	OpenFOAM	MM5	MrBayes
NAMD	**Astro**	…	...	Migrate-n
WebMO	StarLab			PhyML
VMD	IRAF			PLINK
...	...			...

Software requests to: *support@ace-net.ca*

ACENET
accelerate discovery

computecanada
regional partner

Software Development

Languages	Compiler Suites	Parallel APIs	Libraries	Tools
C/C++	GCC	Open MPI	LAPACK	TotalView
Fortran	Intel	OpenMP	FFTW	dbx, gdb
Perl	PGI	...	GSL	Valgrind
Python	SunStudio		ACML	make
R			netCDF	cvs, svn
Java			HDF	git
...			Boost	...
			...	

Software requests to: *support@ace-net.ca*

ACENET
accelerate discovery

computecanada
regional partner

Available Compilers At ACENET

C, C++, and Fortran

Portland Group (PGI)

- pgcc, pgCC, pgf77, pgf90, pgf95

GNU Compiler Suite

- gcc, g++, gcc4, g++4, gfortran, g77

Intel Compiler Suite

- icc, icpc, ifort

SunStudio

- cc, CC, f77, f90, f95

See www.acceleratediscovery.ca/wiki/Programming_Tools

Environment Modules

```
$ module list
Currently Loaded Modulefiles:
 1) pgi/12.10        2) openmpi/pgi/1.2.9

$ gcc --version
gcc (GCC) 4.4.7 ...

$ module avail gcc
gcc/4.4.7(default)  gcc/4.6.4  gcc/4.8.3

$ module load gcc/4.8.3
$ gcc --version
gcc (GCC) 4.8.3 ...        ← changed!
$ module unload gcc
```

MPI

Open MPI is default implementation

Use modules to vary the underlying compiler, e.g.

```
$ module unload openmpi
$ module load gcc openmpi/gcc
```

See www.acceleratediscovery.ca/wiki/Open_MPI

Parallel Job Processing

```
$ cat jobscript
#$ -cwd
#$ -j yes
#$ -l h_rt=48:00:00
#$ -l h_vmem=2G
#$ -pe ompi* 32
module purge
module load gcc openmpi/gcc
mpirun ./application
```

Notice that process count and host list is passed automatically from Grid Engine to Open MPI.

Understanding the Grid Engine

Job Submission and Scheduling at ACENET

Slides at http://www.ace-net.ca/training/workshops-seminars/

computecanada
regional partner

Objectives

Run jobs: **`qsub`**

Resources: time, memory, *etc.*

Monitoring: **`qstat, showq, qsum`**

Troubleshooting: **`qacct, jobmem,`** ...

How and why it works

qsub and qstat

```
$ qsub script.sh
Your job 2491 ("script.sh") has been submitted

$ qstat
job-ID  prior   name       user         state submit/start
at          queue           slots ja-task-ID
-----------------------------------------------------------
  2491 0.60500  script.sh  rdickson     r     03/10/2008
11:58:04   short.q@cl012      1

$ qstat
$
```

Normal states: **qw** = waiting, **r** = running

No reply? Job is finished.

The Job Script

```
$ cat script.sh
#$ -cwd
#$ -j y
#$ -l h_rt=1:0:0
module purge
module load gcc r
R --no-save -q < my_program.R
$ qsub script.sh
Your job 2491 (“script.sh") has been
submitted
```

Required parameter: h_rt

You must supply a run time estimate

h_rt = “hard run-time”

- job will be killed when h_rt reached
- leave a margin for error!

Syntax:

```
#$ -l h_rt=hh:mm:ss
#$ -l h_rt=seconds
$ qsub -l h_rt=hh:mm:ss other args...
```

qsub flags ≈ script directives

Directives in script...

```
#$ -l h_rt=24:0:0
#$ -cwd
./myprog <file.in >file.out
```

...equivalent to arguments to qsub

```
qsub -l h_rt=24:0:0 -cwd script
```

Command line overrides script

Asking for CPU cores

```
#$ -pe parallel_env  n_slots
```

```
#$ -pe ompi* 4
```
for example

The **parallel environment** is usually one of

`ompi*`	*distributed memory*
`openmp`	*shared memory*
`gaussian`	*computational chemistry*

Asking for memory: h_vmem

`#$ -l h_vmem=`*`bytes`*

- "hard limit on virtual memory"
- allocation *per slot for parallel jobs*
- if running procs exhaust physical memory, performance drops by x100 !
- see defaults at http://www.acceleratediscovery.ca/wiki/Memory_Management
- amount available varies by host
  4G per slot is typical

Options: Output Control

#$ -cwd

run script from current working directory,
i.e. directory from which submitted

#$ -o *stdout_file*
#$ -e *stderr_file*

redirect stdout and stderr from ‘script.sh.o1235’

#$ -j y

join stdout and stderr streams into one file

Basic Scheduling Algorithm

Don't interrupt running jobs

Assign priorities to waiting jobs

When resources come free:

- Run highest priority job that can be satisfied by available resources
- Iterate until all resources are committed or all jobs have been checked

Wait for new resources

Priority “Fairshare”

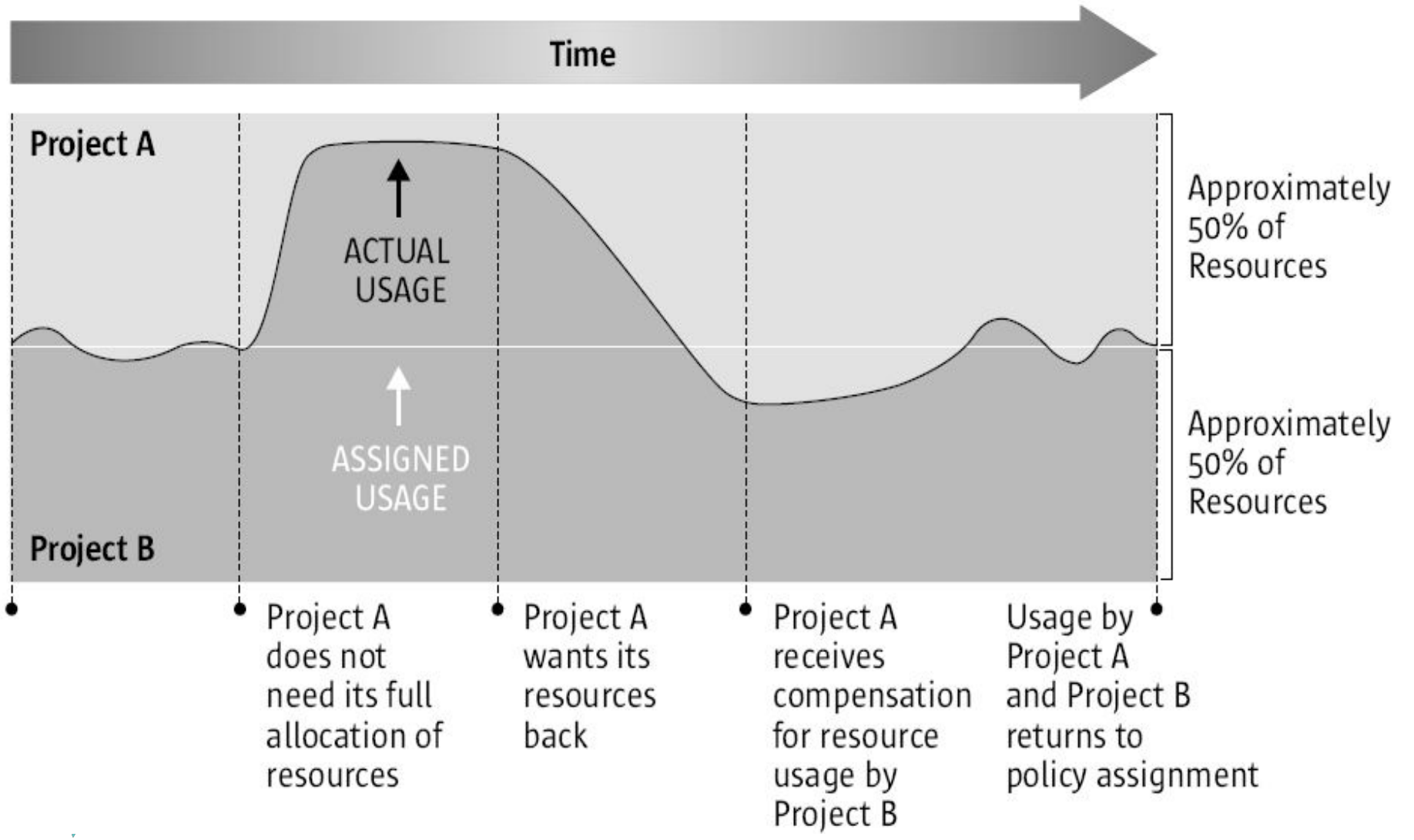
Each project (research group) starts with equal rights

Using the system uses up priority

- Time-decaying record of usage kept per project
- one month half-life

Projects with low recent usage get higher priority

Fairshare



http://gridengine.info/articles/2005/09/30/pretty-pictures-explain-functional-vs-sharetree-scheduling

Reading Showq

$ showq

```
============================== Running jobs ==============================
  JOBID   USERNAME   STATE   SLOTS    ----- ENDS BY -----         QUEUE@HOST
3259638   cmcinnis       r       1    05/15/2015 07:04:31      short.q@cl100
3259639   cmcinnis       r       1    05/15/2015 07:05:46      short.q@cl084
3259640   cmcinnis       r       1    05/15/2015 07:06:16      short.q@cl094
3259186   michaeld       r      32 R  05/15/2015 09:07:02     medium.q@cl012
3259468      phred       r      16 R  05/15/2015 13:11:31      short.q@cl011
3254059   ylorcand       r      64 R  05/16/2015 01:38:17       long.q@cl097
3259670     rbrand       r       1    05/16/2015 04:42:31      short.q@cl010
============================== Pending jobs ==============================
  JOBID   USERNAME   STATE   SLOTS   ---- SUBMITTED ----   PRIORITY  TIME REQ
3259599    crunchy      qw     100    05/13/2015 23:14:46   0.5406   600:00:00
3254061   ylorcand      qw      64 R  05/01/2015 06:30:46   0.5300    48:00:00
3254060   aanderso      qw      64 R  05/01/2015 09:35:45   0.5125    48:00:00
```

What is a Queue?

Technical term in Grid Engine

Abstract container for *running* jobs

If h_rt ≤ queue limit, job can go in queue (container)

Queues (containers) have finite size

e.g. long.q at mahone has 24 slots

Waiting list ≠ queue in Grid Engine

Not strictly a group of compute hosts either

e.g. both sub.q and tarasov.q
include host cl050@placentia.ace-net.ca

What Queues Are There?

short.q	*up to 48 hours*	*~70% of CPUs*
medium.q	*up to 2 weeks (336 hours)*	*~20% of CPUs*
long.q	*up to 1 month (720 hours)*	*~10% of CPUs*
test.q	*up to 1 hour + flag "test=true"*	*1 node*

Plus Some Restricted & Special purpose queues

Where To Go For Help

Online:

- *command* --help
- man *command*

ACEnet wiki

- http://ace-net.ca/wiki/ACEnet

Email support:

- support@ace-net.ca
- Please supply as much info as possible

www.ace-net.ca/wiki/Ask_Support