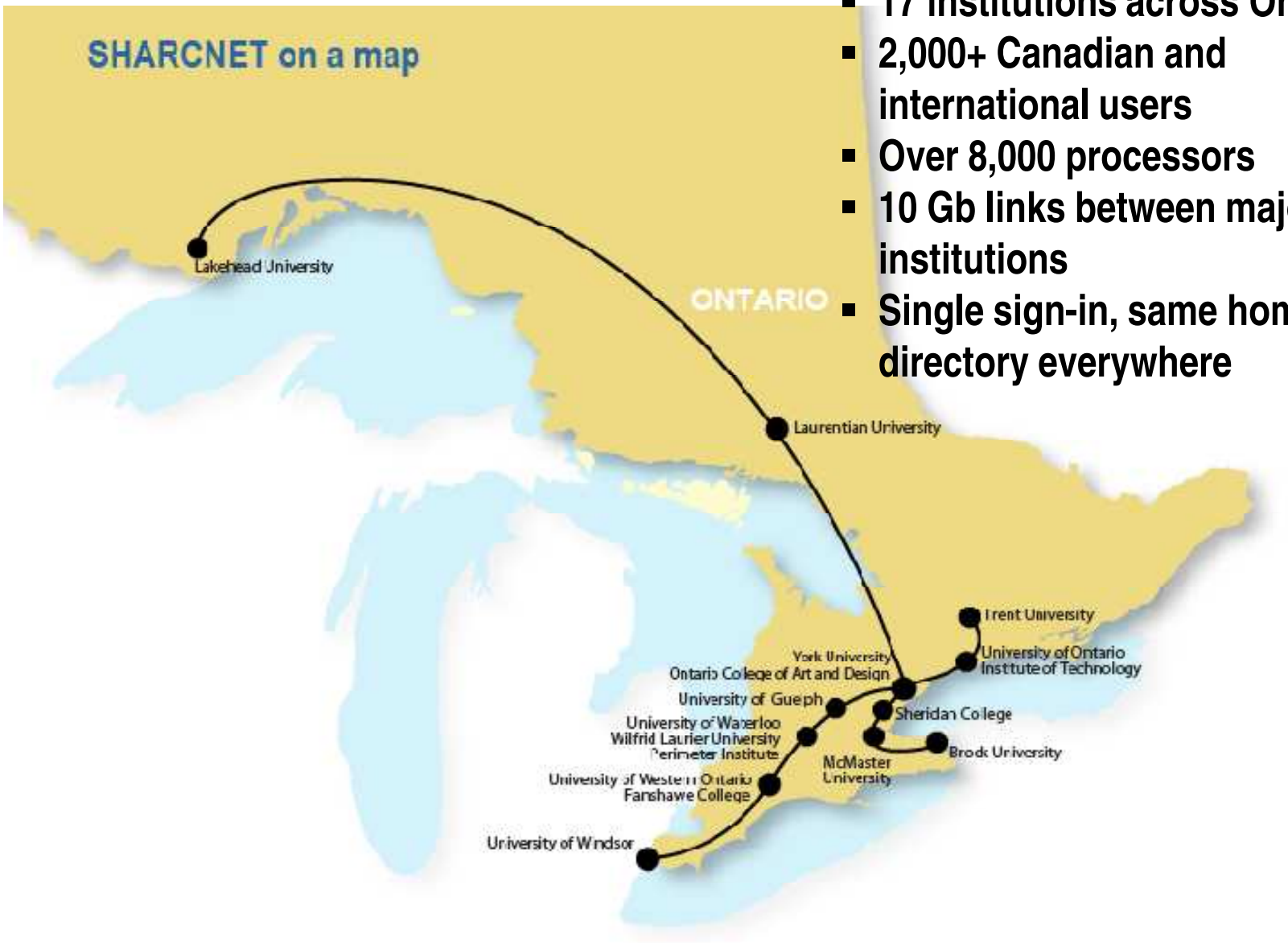
Using SHARCNET: An Introduction

Agenda:

- What is SHARCNET
- The HPC facilities
- People and support
- Getting an account
- SHARCNET essentials
- Working with the scheduler LSF

What is SHARCNET?

What is SHARCNET? A High Performace Computing Consortium …



- 17 institutions across Ontario
- 2,000+ Canadian and international users
- Over 8,000 processors
- 10 Gb links between major institutions
- Single sign-in, same home directory everywhere

Industry and government partners

- Government
  - Canada Foundation for Innovation
  - Ontario Innovation Trust
  - Ontario R&D Challenge Fund
  - Optical Regional Advanced Network of Ontario (ORANO)

- Industry
  - Hewlett Packard
  - SGI
  - Quadrics Supercomputing World
  - Platform Computing
  - Nortel Networks
  - Bell Canada

Vision, mission and goals

- **The Vision**
  - *To become a world leading, academic high-performance computing consortium enabling forefront research and innovation*
- **The Mission**
  - *To promote and facilitate the use of high performance computational techniques among researchers in all fields and to create a new generation of computationally-aware individuals*
- **General Goals**
  - Provision of otherwise unattainable compute resources
  - Reduce time to science
  - Remote collaboration

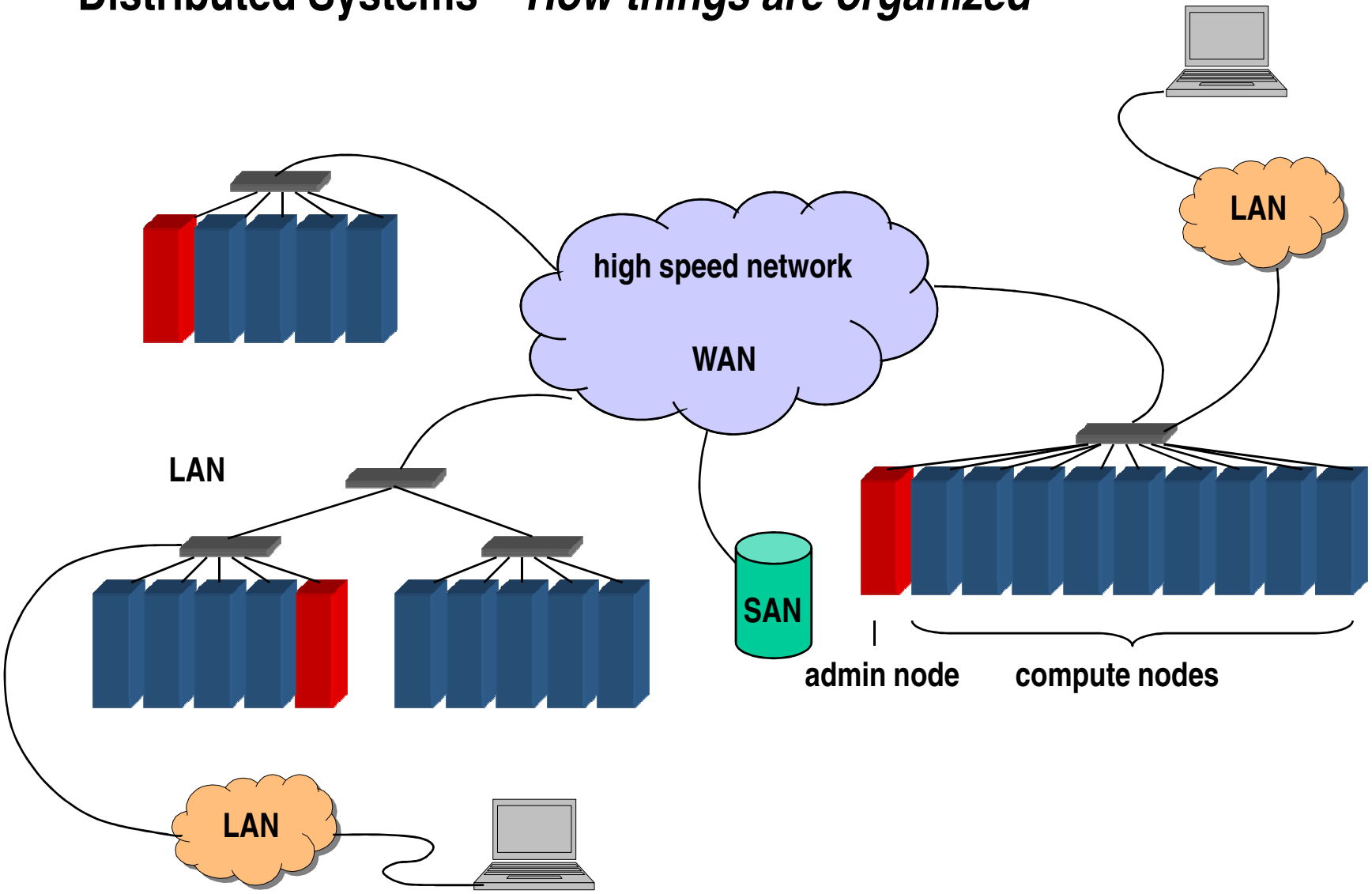
HPC Facilities

Computational resources

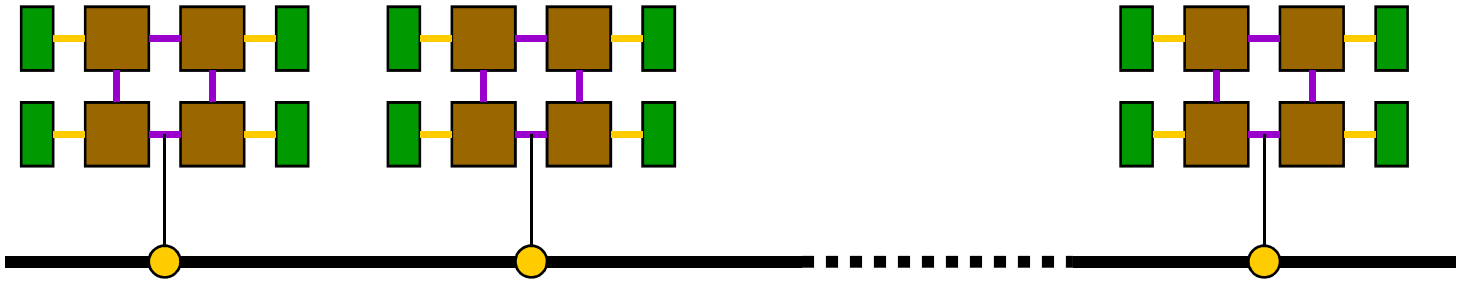
- **Computers**
  - Clusters (**distributed memory**): for parallel and serial programs
    - Very fast interconnect
    - Fast interconnect
    - Fast interconnect and SMP nodes
    - Serial farm
  - Symmetric multiprocessing (SMP) systems (**shared memory**): for concurrent, parallel, threaded applications.
- **Visualization Systems**
  - Visualization clusters (McMaster)
  - Visualization workstations + large LCD monitors at some institutions (local access encouraged).
- **Access Grid Rooms** (Multi-media)
  - Video conferencing facilities, across site workshops, coast-to-coast seminars, etc.

Architecture

- Distributed Systems – *How things are organized*

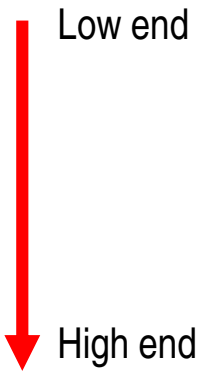


- **Distributed Systems – "Clusters"**
  - Simple (relatively) nodes connected by a network.
  - Each node can be a SMP machine, e.g. a four processor ("4-way") server.
  - Inter-process communication is explicit (e.g. message passing).
  - Much better scalability at the cost of communication overhead (performance).
  - Non-trivial to programmers!



Architecture

- **Interconnect Types**
  - 10/100 **Ethernet** – Cheap, seen in most labs, offices and home.
  - **Gigbit Ethernet** – Inexpensive, see in some labs, campus network.
  - **Myrinet** – mid to large clusters.
  - **Infiniband** – mid to large clusters.
  - **QsNet** – Capable, mission critical superclusters.



Suitable uses

- **Compute-Intensive Problems**
  - The resources are provided to enable HPC and are not intended as a replacement for a researcher's desktop or lab machines.
  - SHARCNET users can productively conduct HPC research on a variety of SHARCNET systems each optimally designed for specific HPC tasks
- **Academic HPC Research**
  - The research can be business-related, but must be done in collaboration with an academic researcher
- **Fairness Access**
  - Users have access to all systems
  - Clusters are designed for certain type of jobs
  - Job runs in batch mode (scheduling system) with fairshare

Support

- Webportal
- People
- Problem tracking
- Project consultation
- Education and training
- AccessGrid

Webportal

- SHARCNET's web site provides extensive information about deployed systems, software stack and help.

Where to find most system information

Where to find HOWTO's

See what other people are doing

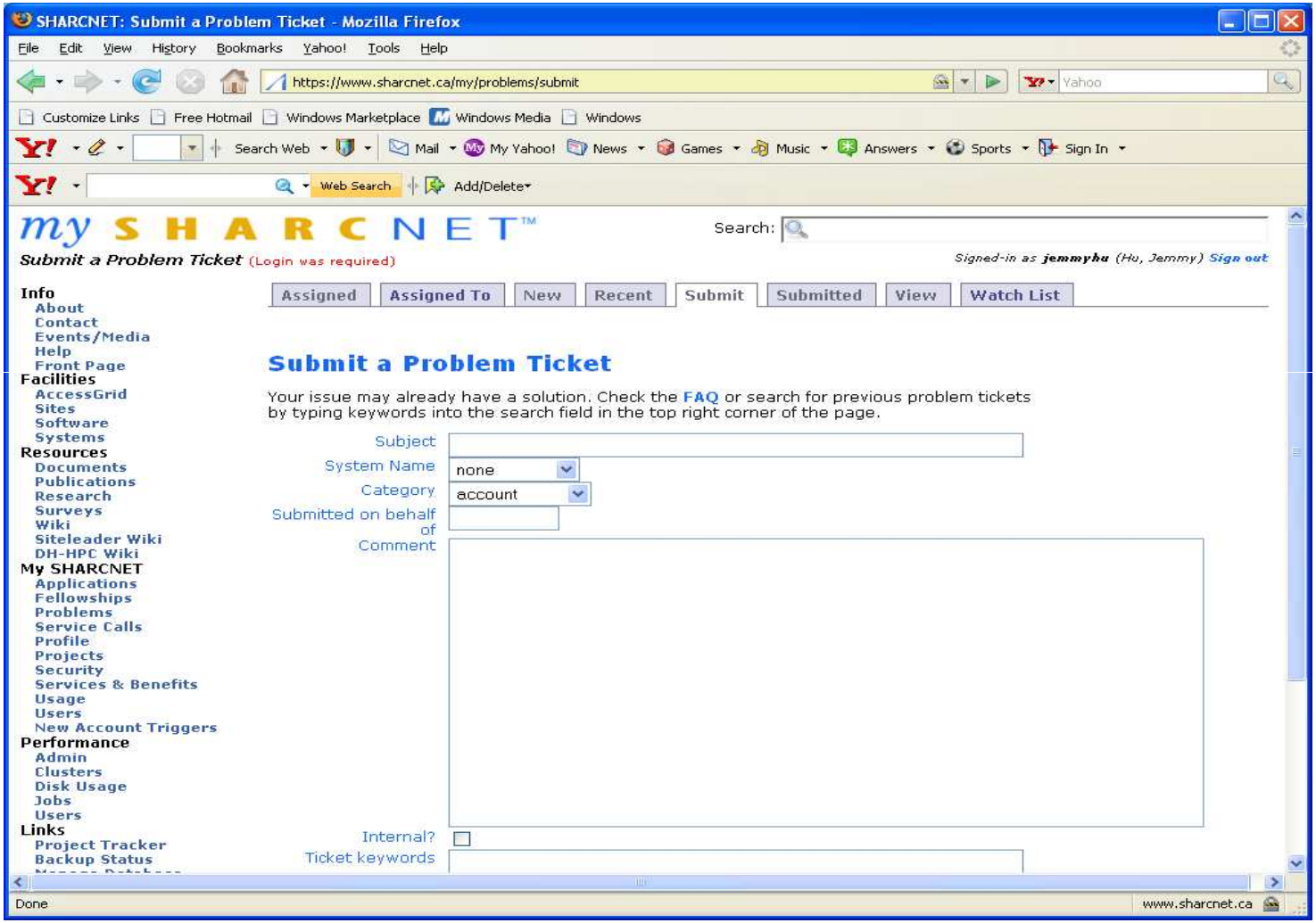
Support people

- **HPC Consultants**
  - A point of contact, central resource.
  - Analysis of requirements.
  - Development support, performance analysis.
  - Training and education.
  - Project, technical computing consultations.

- **System Administrators**
  - User accounts.
  - System software.
  - Hardware and software maintenance.

Problem tracking

- Use **Problem Tracking** in the web portal.
  https://www.sharcnet.ca/my/problems/submit



Project support

- Project consultation;
- Dedicated resource access;
- Dedicated programming support

Education and training

- SHARCNET offers different forms of education and training
  - Weekly online seminar: New users introduction and research topics;
  - **Workshops** and **on-site training**;
  - Summer school – week long, intensive courses on high performance and technical computing;
  - Credit courses at undergraduate and graduate level.

AccessGrid and collaboration

- **Using AccessGrid – Bringing People Face-to-face**
  - Video conferencing.
  - Online, multi-site seminars.
  - Remote collaboration.

Getting An Account and Help

Getting an account

- **Apply for An Account Online**
  - Apply online at http://www.sharcnet.ca/.
  - Students, postdoc, visiting fellows must have a **sponsor** who has an account.
  - *SPECIAL ARRANGEMENT.*
- **Account Approval:**
  - Faculty accounts are approved by the **site leader.**
  - Students/postdoc/fellows require a faculty **sponsor**, who shall approve such accounts.
  - Non-SHARCNET institution accounts are approved by the **Scientific Director.**
- **Web Account**
  - Your user name and password allow you to access information/files, submit requests and manage your own profile through the ***web portal***.

One account, access both systems and web portal

- **Login to Systems**
  - Siteless login – single username/password for all systems.
  - Same user home directory across all systems.
  - Systems are designed and deployed for different purposes:
    - **parallel applications**,
    - **serial applications**, e.g. large number of **serial case runs**,
    - **threaded applications** that make use of shared memory by threads, etc
- **Login to Web**
  - Discovery resources.
  - See statistics.
  - Users can change password on the web.
  - Report and keep track of problems.

Where to look for Information

- **Help** on the web site.
- **FAQs** are on the web. Go to SHARCNET web site
- **Weekly Online Seminars** on every Monday
- **Education Online** – slides, examples from past workshops are also available on the web on the **Help** page.
- System status is available on the web on the **Facilities** page.
- **RSS** feeds.

How to get help

- Use **Problem Tracking** in the web portal.

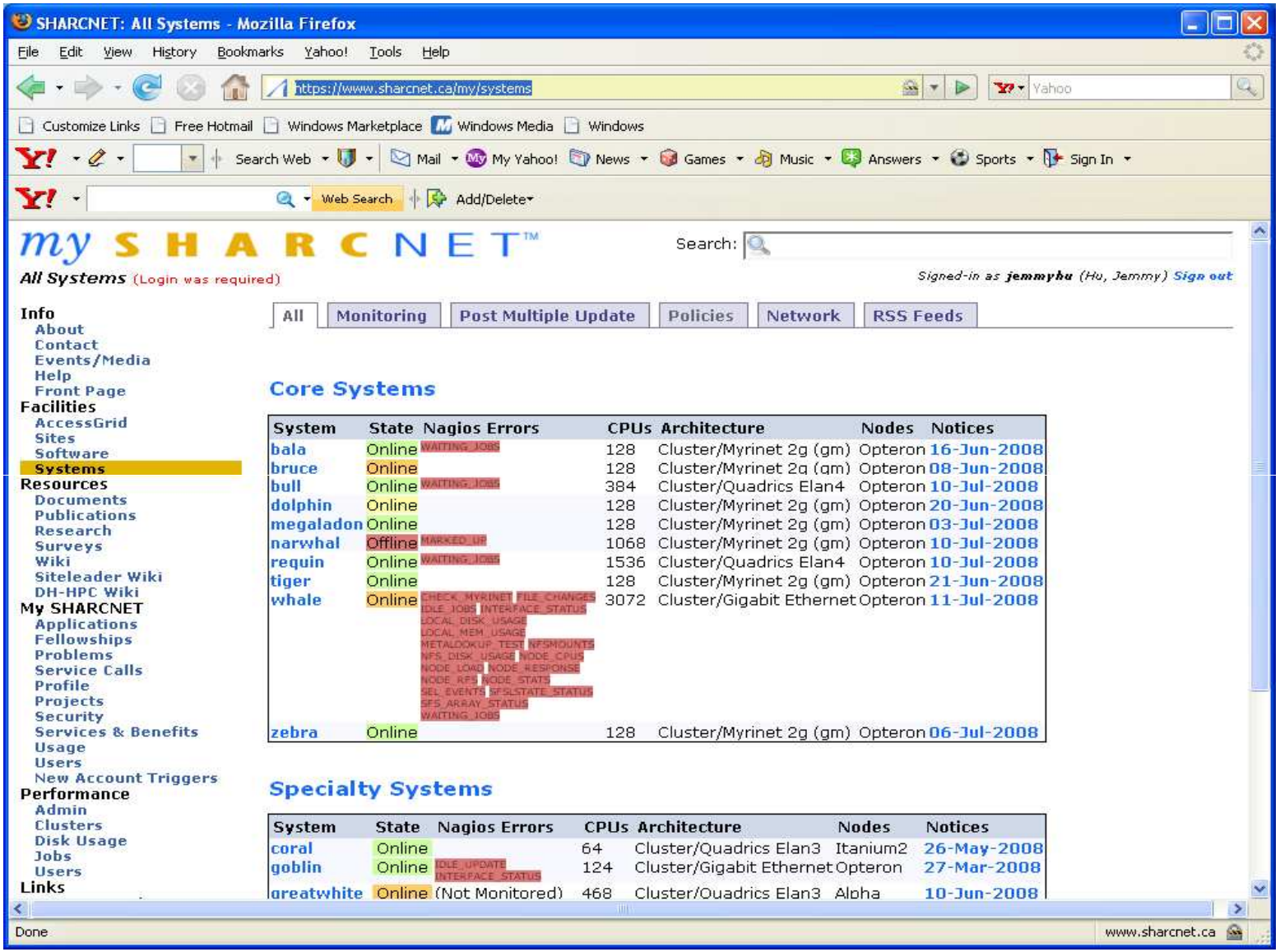
- E-mail.
- Phone Call.

Our contact info is listed on the **Contact** page

SHARCNET Essentials

- Computing Systems
- Moving, editing files
- Compiling programmes
- Software and libraries
- Running programmes in batch mode with LSF

Computing Systems: https://www.sharcnet.ca/my/systems



Facilities: Intended use (cont'd)

Cluster	CPUs	RAM /node	Storage	Interconnect	Intended Use
requin (Capability)	1536	8 GB	70 TB	Quadrics	Resource intensive MPI (fine grained, large memory)
narwhal (Utility)	1068	8 GB	70 TB	Myrinet (GM)	MPI, small-scale threaded
whale (Throughput)	3072	4 GB	70 TB	GigE	Serial
bull (SMP-friendly)	384	32 GB	70 TB	Quadrics	MPI, small-scale threaded and large memory applications
silky (SMP)	128	256 GB	4 TB	NUMAlink	Threaded and large memory applications
bala, bruce, dolphin, megaladon, tiger, zebra	128	8 GB	4 TB	Myrinet (GM)	General purpose
mako	16	2GB	200GB	Myrinet (MX)	Development/testbed

How one typically uses SHARCNET

- Login to a system via SSH, you see familiar UNIX environment.
- Edit source code and/or change the input data/configuration file(s).
- Compile source code.
- Submit a program (or many) to batch queuing system.
- Check results later.

How one works – a UNIX example

1. Copy files to SHARCNET systesm

   scp projects.tar.gz bull.sharcnet.ca:

2. Login to remote system bull, with X connection

   ssh bull.sharcnet.ca –X –l bge

3. Edit your files using your favourite editor, e.g. emacs, vi

   tar zxvf projects.tar.gz

   cd projects/elec_price

   vi price_main.cc

4. Compile your code

   c++ price_main.cc fun1.cc fun2.cc … fun5.cc –o price

5. Test drive your code

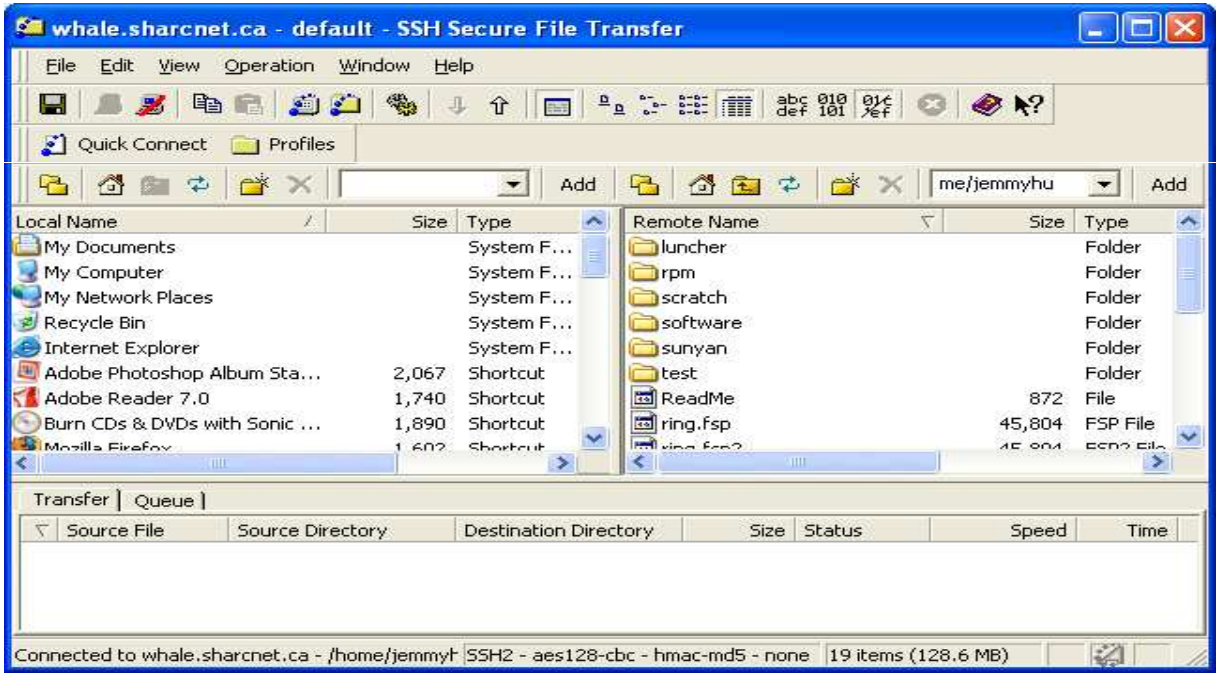
   sqsub –q serial –t –o price.log ./price

6. Run your code (production runs)

   sqsub –q seiral –o price.log1 ./price

File transfer from/to your desktop

- UNIX
  - User **scp** or **sftp**
- Windows
  - User **putty**, or
  - **SSH Secure File Transfer/Shell**



Compiling programmes

- SHARCNET provides a unified compiling environment that chooses the right underlying compiler, options and libraries for you! Use them always unless you know better.

Command	Language	Extension	Example
cc	C	c	**cc** code.c –o code.exe
CC, c++, cxx	C++	.C, .cc, .cpp, .cxx, c++	**CC** code.cpp –o code.exe
f77	Fortran 77	.f, .F	**f77** Fcode.f –o Fcode.exe
f90/f95	Fortran	.f90, .f95, .F90, F95	**f90** Fcode.f90 –o Fcode.exe
mpicc	C	c	**mpicc** mpicode.c –o mpicode.exe
mpiCC	C++	C++	**mpiCC** mpicode.cc –o mpicode.exe
mpif77	Fortran 77	f77	**mpif77** mpicode.f –o mpicode.exe
mpif90/mpif95	Fortran	f90/f95	**mpif90** mpicode.f90 –o mpicode.exe

Development toolkit

- **Libraries**
  - ACML, ATLAS, CXML, ScaLAPACK, **MKL (Intel), MLIB (hp), IMSL** (VNI), **NAG**, etc.
  - ScaLAPACK, FFTW, PETSc and many others.
  - *…requested by users.*
- **Application Packages**
  - Blast, Gromacs, NWChem, Octave, R, …
- **Commercial Packages**
  - **Gaussian** (several places, site license required).
  - **Fluent.**
  - **MATLAB.**
- **Development tools**
  - Debugging: **DDT**, gdb, …
  - Profiling/Optimization: OPT, Altix toolkit (SGI), gprof,

Go to web site at: http://www.sharcnet.ca/Help/

File system basics

- **Policy**
  - Same username/password across all systems, and web account.
  - Common home directory across SHARCNET (exceptions: wobbe, cat)
  - Common SHARCNET wide software are in /opt/sharcnet
  - /home backup
- **File system**

pool	quota	expiry	purpose
/home	200 MB	none	Source, small configuration files, backed up regularly.
/work	200 GB	none	Active data files (longer time storage, NOT backed up). Local to system.
/scratch	None	2 weeks	Active data files, binaries (2 weeks, NOT backed up). Local to system.
/tmp	160 GB	10 days	Node-local scratch

  - **Important:** run jobs on /scratch or /work

Archiving your data

- **Saving your files in archive**

  archive --put *tar_file_name* $file_1$ [, $file_2$[, …, $file_n$]]

- **Poking archive**

  archive --list [*directory* | *file*]

- **Retrieving your files from archive**

  archive --get [*directory* | *file*]

- **Deleting archived files**

  archive --remove [*directory* | *file*]

Working with LSF/SQ

Commonly used SQ commands

bqueues – list available queues.

sqsub – submit a program ("job") to a specific queue.

sqjobs – list the status of submitted jobs.

sqkill – kill a program by job ID.

bhist – list jobs history

Running jobs

- **Log on to desired system**
  - Ensure files are in /work/username
  - Do not run jobs out of /home --- it can be very slow
  - Jobs are submitted using the **sq** commands (a wrapper, e.g. around LSF)

    sqsub –q *queue_name* [*options*] *your_program* [*your_args*]

- **Choosing the right queue**

Job Type	Queue Name	CPUs
Parallel	**mpi**	>2
Parallel	**threaded** (system dependent)	4 (greatwhite, bull, etc.), 2 (requin, wobbe, cat) 128 (silky)
Serial	**serial**	1
Test runs	**test**, must be used with one of the above queues	

  - You might see some other queues not listed such as staff, gaussian, etc. They special queues with restrictions, not open to all users.

Using sq commands

- **Submitting a serial jobs**

  sqsub -q serial --test –r 10h ./simula

  sqsub -q serial –r 10.0d -o simula.out ./simula

  sqsub -q serial –r 10m -o simula.out -i simula.in ./simula

- **Submitting a parallel jobs**

  sqsub -q mpi --test -n 24 –r 10h ./simula

  sqsub -q mpi -n 24 –r 10m -o simula.out ./simula

  sqsub -q mpi -n 24 –r 10d -o simula.out -i simula.in ./simula

- **Submitting a parallel job with request on process distribution**

  sqsub -q mpi --test –r 20m -n 24 ./simula

  sqsub -q mpi -n 24 -N 6 –r 10h -o simula.out ./simula

  sqsub -q mpi -n 24 -N 24 –r 2.0d -o simula.out -i simula.in ./simula

  Across 24 nodes

Running jobs

- **sqsub –help**

  display command options

Running jobs

- **bqueues** - list queues

```
[…@nar316 ~]$ bqueues
QUEUE_NAME      PRIO STATUS          MAX JL/U JL/P JL/H NJOBS  PEND   RUN  SUSP
staff            150  Open:Active       -    -    -    -     0     0     0     0
test             100  Open:Active       -    -    -    -     0     0     0     0
threaded          80  Open:Active       -    -    -    -     6     0     6     0
mpi               80  Open:Active       -    -    -    -  1972   914   994    64
serial            40  Open:Active       -    -    -    -    25     0    10    15
```

Running jobs

- **sqjobs** – list the status of submitted jobs.

```
[…@nar316 ~]$ sqjobs
 jobid user queue state ncpus time command
------ ---- ----- ----- ----- ---- -------------
136671  bge   mpi     R    20   0s ./my_mpi_prog
1060 CPUs total, 30 idle, 1030 busy; 43 jobs running; 16 suspended, 12 queued.
```

Running jobs

- **sqkill** – kill a job in the queue that you want to stop.

```
[…@nar316 ~]$ sqjobs
 jobid user queue state ncpus time command
------ ---- ----- ----- ----- ---- -------------
136672  bge   mpi     Q     1   0s ./my_mpi_prog
1060 CPUs total, 50 idle, 1010 busy; 42 jobs running; 16 suspended, 13 queued.


[…@nar316 ~]$ sqkill 136672
Job <136672> is being terminated
```

Running jobs

- **bhist** – list jobs history

bhist [-a | -d | -p | -r | -s] [-b | -w] [-l] [-t] ...

Options:

	-a	displays finished and unfinished jobs (over-rides -d, -p, -s and -r)
	-b	brief format; if used with -s option, shows reason why jobs were suspended
	-d	only display finished jobs
	-l	long format; displays additional information
	-u user	display jobs submitted by specified user

Running jobs

- **bhist** – A snapshot of command output

```
nar317:~/pub/exercises% bhist -a
Summary of time in seconds spent in various states:
JOBID   USER    JOB_NAME  PEND    PSUSP   RUN   USUSP   SSUSP   UNKWN   TOTAL
134177  dbm     *o_mpi_c  8       0       37    0       0       0       45
134227  dbm     *o_mpi_c  10      0       10    0       0       0       20
nar317:~/pub/exercises% bhist -l 134177
Job <134177>, User <dbm>, Project <dbm>, Job Group </dbm/dbm>,
                     Command </opt/hpmpi/bin/mpirun -srun -o mpi_hello.log ./
                     mpi_hello>
Fri Sep 15 13:06:08: Submitted from host <wha780>, to Queue <test>, CWD <$HOME/
                     scratch/examples>, Notify when job ends, 4 Processors Requ
                     ested, Requested Resources <type=any>;
Fri Sep 15 13:06:16: Dispatched to 4 Hosts/Processors <4*lsfhost.localdomain>;
Fri Sep 15 13:06:16: slurm_id=318135;ncpus=4;slurm_alloc=wha2;
Fri Sep 15 13:06:16: Starting (Pid 29769);
Fri Sep 15 13:06:17: Running with execution home </home/dbm>, Execution CWD
                     </scratch/dbm/examples>, Execution Pid <29769>;
Fri Sep 15 13:06:53: Done successfully. The CPU time used is 0.3 seconds;
Fri Sep 15 13:06:57: Post job process done successfully;
Summary of time in seconds spent in various states by  Fri Sep 15 13:06:57
  PEND     PSUSP    RUN      USUSP    SSUSP    UNKWN    TOTAL
  8               0                37               0                0
        0        45
```

The End.

- Questions?