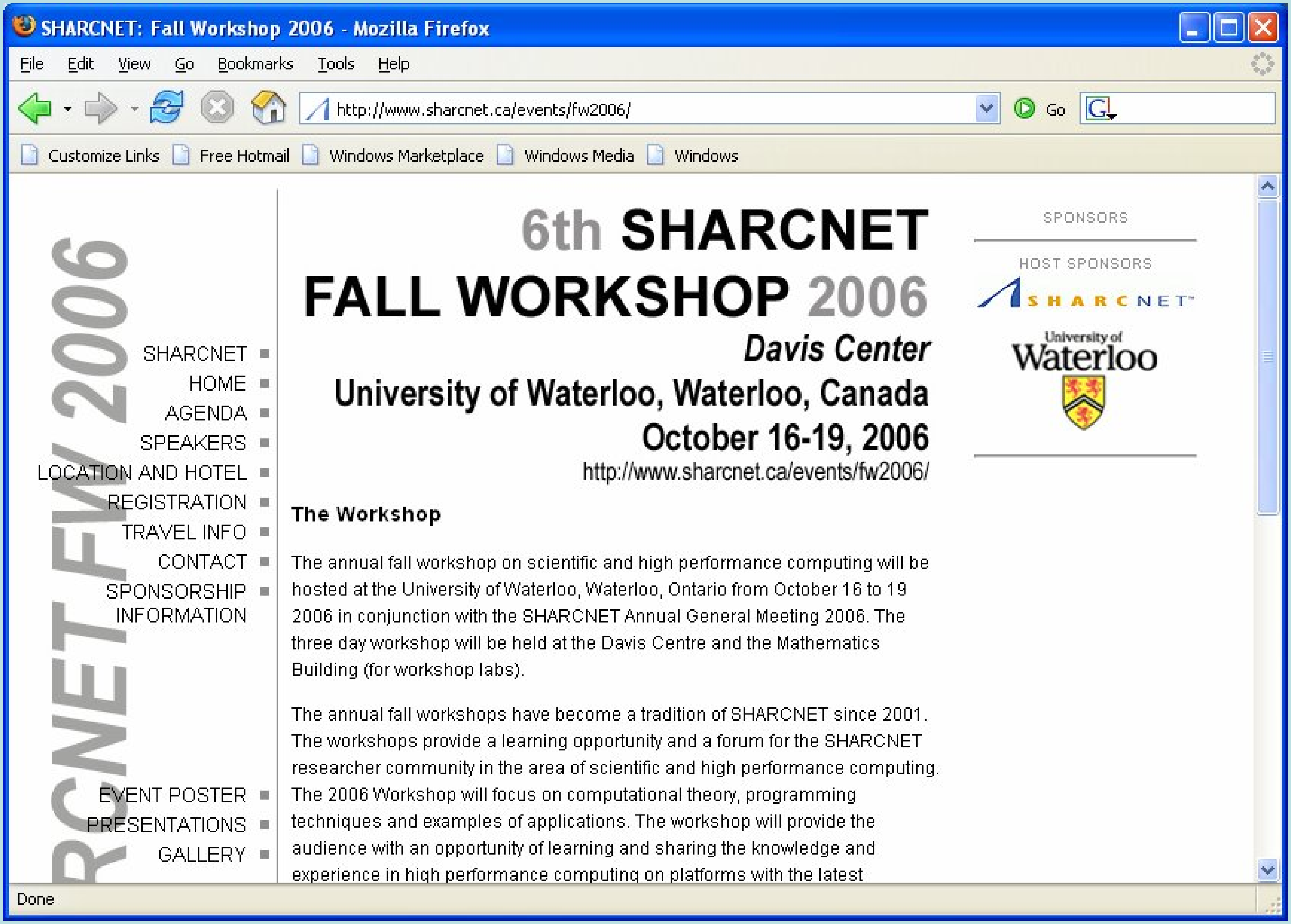
SHARCNET Seminars

September, 2006

Part I: Introduction to SHARCNET (~1 hr)

Part II: Using SHARCNET (Demo)

http://www.sharcnet.ca/events/fw2006/



6th SHARCNET FALL WORKSHOP 2006

Davis Center

University of Waterloo, Waterloo, Canada

October 16-19, 2006

http://www.sharcnet.ca/events/fw2006/

The Workshop

The annual fall workshop on scientific and high performance computing will be hosted at the University of Waterloo, Waterloo, Ontario from October 16 to 19 2006 in conjunction with the SHARCNET Annual General Meeting 2006. The three day workshop will be held at the Davis Centre and the Mathematics Building (for workshop labs).

The annual fall workshops have become a tradition of SHARCNET since 2001. The workshops provide a learning opportunity and a forum for the SHARCNET researcher community in the area of scientific and high performance computing. The 2006 Workshop will focus on computational theory, programming techniques and examples of applications. The workshop will provide the audience with an opportunity of learning and sharing the knowledge and experience in high performance computing on platforms with the latest

Part I: Introduction to SHARCNET

- SHARCNET
- Account
- Hardware
- Software
- File system
- Job Scheduling
- Support

What’s SHARCNET

http://www.sharcnet.ca

- **The SHARCNET Vision**

  To establish a world-leading, multi-university and college, interdisciplinary institute with an active academic-industry partnership, enabling forefront computational research in critical areas of science, engineering and business.

- **The SHARCNET Mission**

  SHARCNET exists to enable world-class computational research so as to accelerate the production of research results.

General Objectives:

- provide otherwise unattainable compute resources
- build common, seamless computing environment
- promote remote collaboration and researches

Academic and Affiliated Partners

The SHARCNET community consists of 16 academic institutions in Ontario

- **Founding members (June 2001)**
  The University of Western Ontario
  University of Guelph
  McMaster University
  Wilfrid Laurier University
  University of Windsor
  Fanshawe College
  Sheridan College

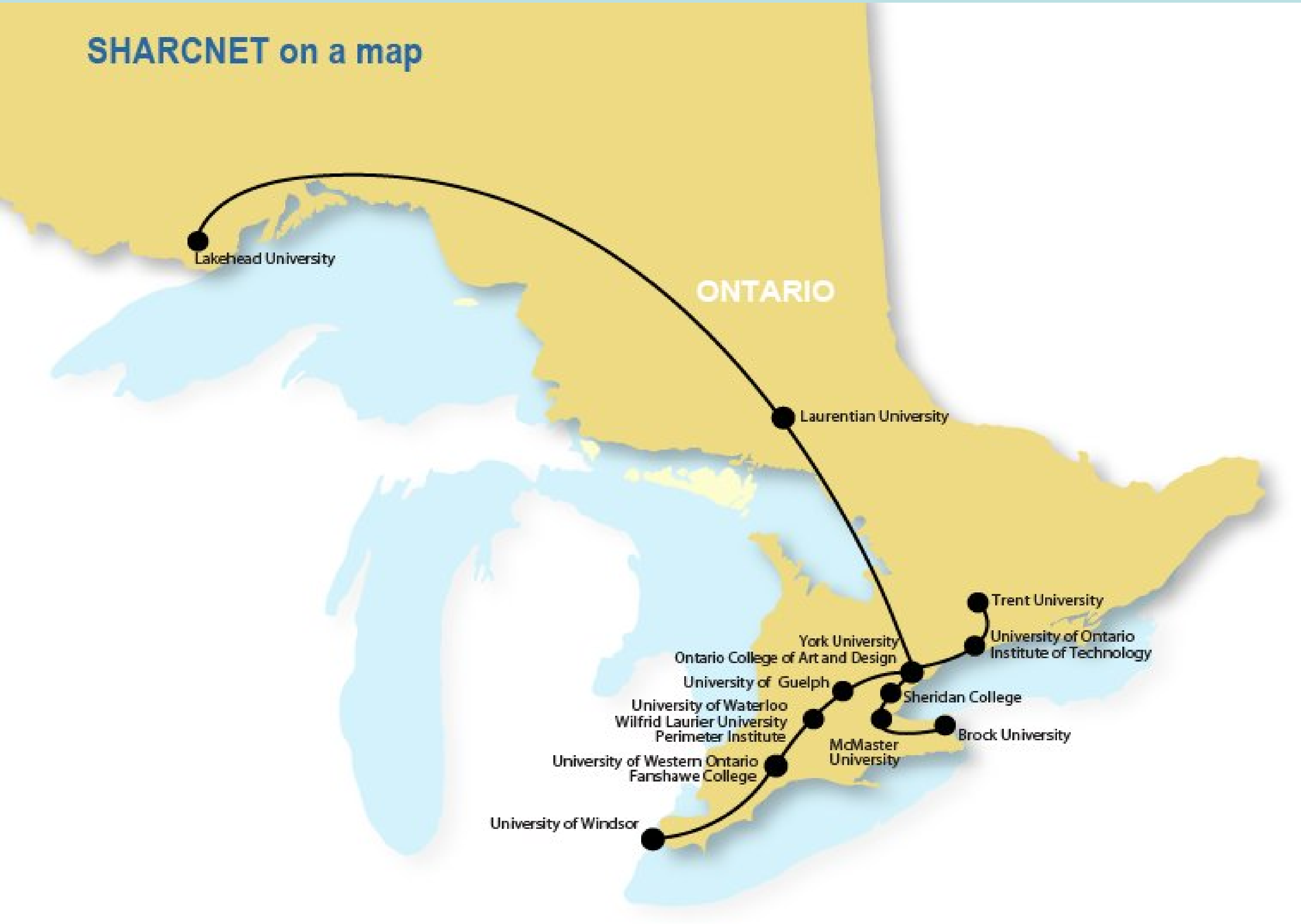
- **New Partners (June 2003)**
  University of Waterloo
  Brock University
  University of Ontario Institute of Technology
  York University

- **New Partners (Dec 2005)**
  Trent University
  Laurentian University
  Lakehead University

- **New Partners (March 2006)**
  Ontario College of Art and Design
  Perimeter Institute for Theoretical Physics

- **Affiliated Partners**
  Robarts Research Institute
  Fields Intitute for Mathematical Sciences

Part I. Introduction to SHARCNET



Industry and Government Partners

Private Sector

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

Government

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

SHARCNET Basics

- FREE to academic researchers

- 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

  Job runs in batch mode (scheduling system) with fairshare

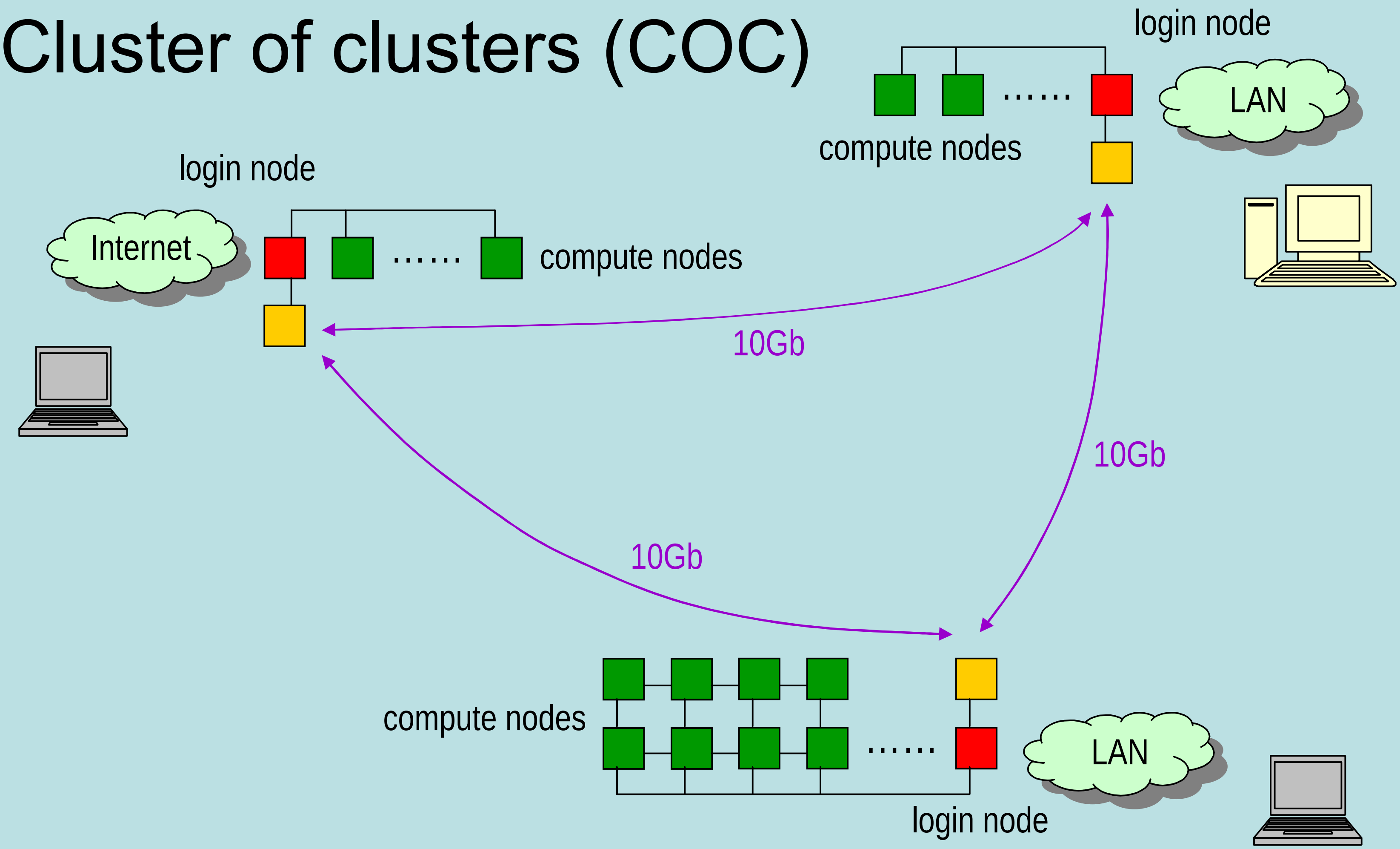
Getting An Account

- Apply for an account online (Web entries)
  1) from Help http://www.sharcnet.ca/Help/account.php
     go to 'Getting An Account' link
  2) from MySHARCNET https://www.sharcnet.ca/Portal/index.php
     go to 'New User' sublink
- A faculty's account (group account) will be approved by the site leader.
- If you are a student, a research fellow, you need to have a sponsor (e.g. your supervisor, coordinator). Account needs to be approved by the sponsor.
- If you are from a non SHARCNET institution and have no contact or affiliation with any of SHARCNET institution, your account needs to be approved by the Scientific Director of SHARCNET.
- You will have a **web account** that allows you to access SHARCNET related information and files, submit requests and manage your own profile using the same password.
- You will have access to all systems through the Internet via **SSH** login only
  ssh userid@name.sharcnet.ca

e.g., ssh syam@whale.sharcnet.ca

SHARCNET Environment

- Cluster of clusters (COC)



SHARCNET Facilities: Overview

- Computers: Clusters, SMP

  1) Architecture: Opteron, Alpha, Itanium/Xeon

  each with its own recommended compilers:

  Pathscale, PGI; Compaq; Intel

  2) Intended Use: HPC tasks specific

  Parallel (capability, utility, SMP, …), **Serial** (throughout, …)

- Visualization Clusters

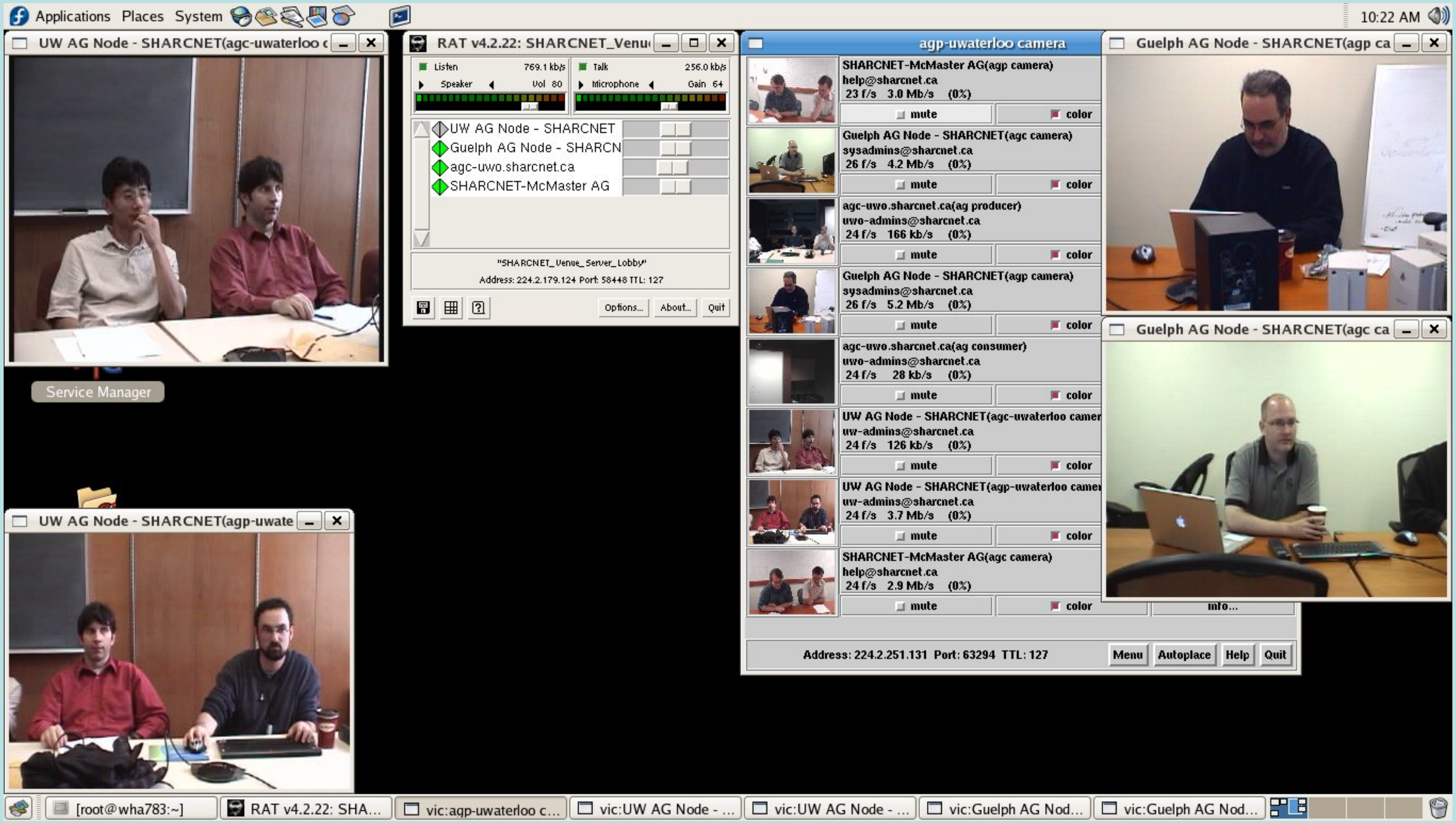
- Access Grid

  - multi-media video conferencing

  - cooperation, cross site workshop, etc

SHARCNET Access Grid

- AG rooms across all SHARCNET sites
- Currently on live: Western-Waterloo-Guelph-McMaster
- Usage: video conferecing/meeting, seminar/workshop, remote cooperate, etc



UW AG Node - SHARCNET
Guelph AG Node - SHARCN
agc-uwo.sharcnet.ca
SHARCNET-McMaster AG

"SHARCNET_Venue_Server_Lobby"
Address: 224.2.179.124 Port: 58448 TTL: 127

SHARCNET-McMaster AG(agp camera)
help@sharcnet.ca
23 f/s 3.0 Mb/s (0%)

Guelph AG Node - SHARCNET(agc camera)
sysadmins@sharcnet.ca
26 f/s 4.2 Mb/s (0%)

agc-uwo.sharcnet.ca(ag producer)
uwo-admins@sharcnet.ca
24 f/s 166 kb/s (0%)

Guelph AG Node - SHARCNET(agp camera)
sysadmins@sharcnet.ca
26 f/s 5.2 Mb/s (0%)

agc-uwo.sharcnet.ca(ag consumer)
uwo-admins@sharcnet.ca
24 f/s 28 kb/s (0%)

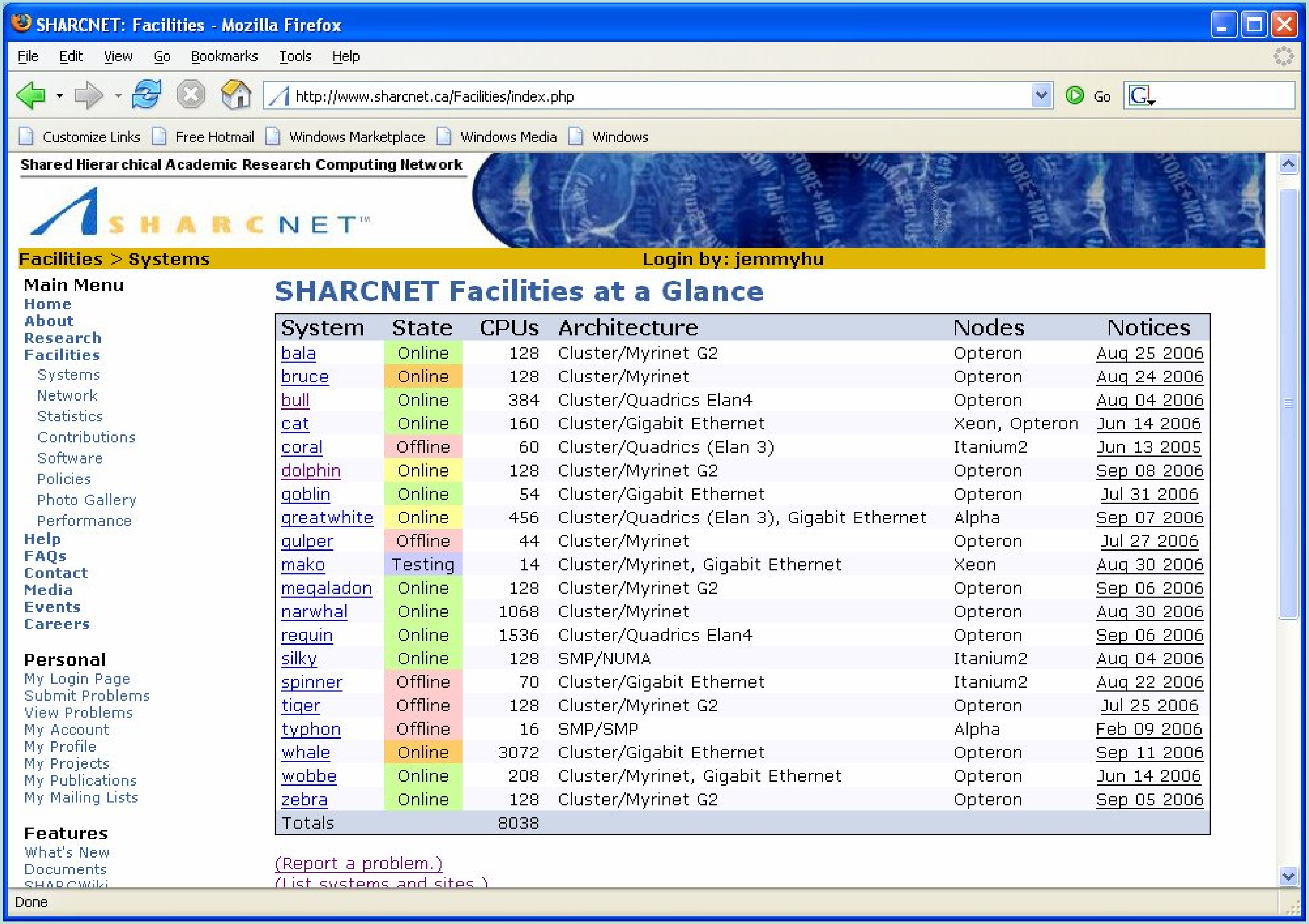
UW AG Node - SHARCNET(agc-uwaterloo camer
uw-admins@sharcnet.ca
24 f/s 126 kb/s (0%)

UW AG Node - SHARCNET(agp-uwaterloo camer
uw-admins@sharcnet.ca
24 f/s 3.7 Mb/s (0%)

SHARCNET-McMaster AG(agc camera)
help@sharcnet.ca
24 f/s 2.9 Mb/s (0%)

Address: 224.2.251.131 Port: 63294 TTL: 127

http://www.sharcnet.ca/Facilities/index.php



Facilities > Systems Login by: jemmyhu

SHARCNET Facilities at a Glance

System	State	CPUs	Architecture	Nodes	Notices
bala	Online	128	Cluster/Myrinet G2	Opteron	Aug 25 2006
bruce	Online	128	Cluster/Myrinet	Opteron	Aug 24 2006
bull	Online	384	Cluster/Quadrics Elan4	Opteron	Aug 04 2006
cat	Online	160	Cluster/Gigabit Ethernet	Xeon, Opteron	Jun 14 2006
coral	Offline	60	Cluster/Quadrics (Elan 3)	Itanium2	Jun 13 2005
dolphin	Online	128	Cluster/Myrinet G2	Opteron	Sep 08 2006
goblin	Online	54	Cluster/Gigabit Ethernet	Opteron	Jul 31 2006
greatwhite	Online	456	Cluster/Quadrics (Elan 3), Gigabit Ethernet	Alpha	Sep 07 2006
gulper	Offline	44	Cluster/Myrinet	Opteron	Jul 27 2006
mako	Testing	14	Cluster/Myrinet, Gigabit Ethernet	Xeon	Aug 30 2006
megaladon	Online	128	Cluster/Myrinet G2	Opteron	Sep 06 2006
narwhal	Online	1068	Cluster/Myrinet	Opteron	Aug 30 2006
requin	Online	1536	Cluster/Quadrics Elan4	Opteron	Sep 06 2006
silky	Online	128	SMP/NUMA	Itanium2	Aug 04 2006
spinner	Offline	70	Cluster/Gigabit Ethernet	Itanium2	Aug 22 2006
tiger	Offline	128	Cluster/Myrinet G2	Opteron	Jul 25 2006
typhon	Offline	16	SMP/SMP	Alpha	Feb 09 2006
whale	Online	3072	Cluster/Gigabit Ethernet	Opteron	Sep 11 2006
wobbe	Online	208	Cluster/Myrinet, Gigabit Ethernet	Opteron	Jun 14 2006
zebra	Online	128	Cluster/Myrinet G2	Opteron	Sep 05 2006
Totals		8038			

(Report a problem.)
(List systems and sites.)

SHARCNET Facilities: Intended use

SHARCNET Facilities are equally available to all users, but the principle is to use them efficiently. Users should select the right machine to submit their jobs. For example, parallel jobs are supposed to run on clusters with Quadrics/Myrinet interconnection.

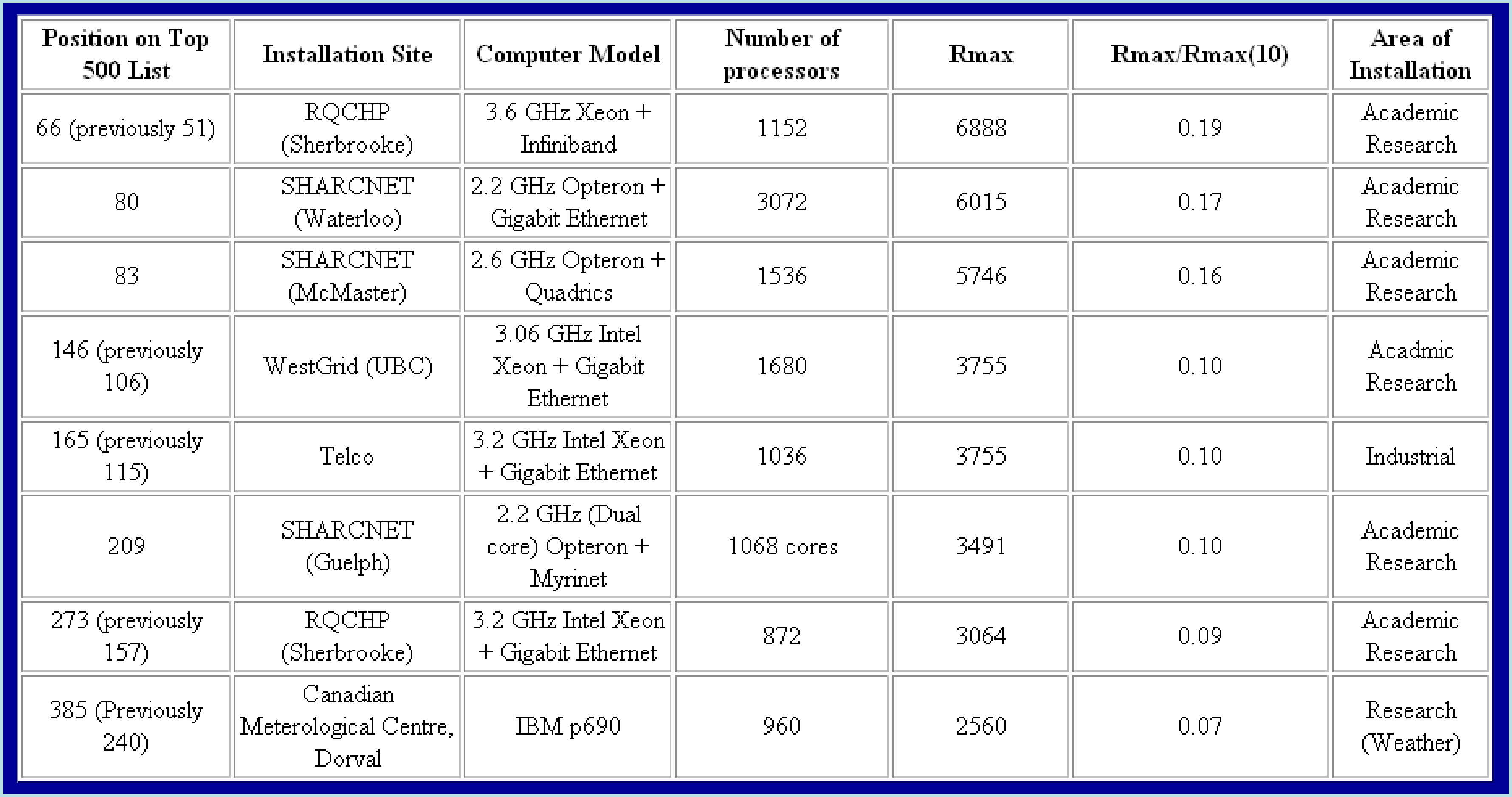
A guideline is

Cluster	CPUs	RAM /node	Storage	Interconnect	Intend Use
requin (Capability)	1536	8 GB	70 TB	Quadrics	Large scale MPI
narwhal (Utility)	1068	8 GB	70 TB	Myrinet	MPI, SMP
whale (Throughput)	3072	4 GB	70 TB	GigE	Serial
bull (SMP-friendly)	384	32 GB	70 TB	Quadrics	High RAM/BW MPI & SMP
silky (SMP)	128	256 GB	4 TB	NUMAlink	large memory SMP OpenMP/pthreads,
bala, bruce, dolphin, megaladon, tiger, zebra	128	8 GB	4 TB	Myrinet	Genaral purpose

Specifications: http://www.sharcnet.ca/Facilities/index.php

SHARCNET Position: Top 500

- top500.org lists the world's fastest supercomputers. This list shows Canada improving its position as technology refreshes for the SHARCNET consortium have been purchased.



Position on Top 500 List	Installation Site	Computer Model	Number of processors	Rmax	Rmax/Rmax(10)	Area of Installation
66 (previously 51)	RQCHP (Sherbrooke)	3.6 GHz Xeon + Infiniband	1152	6888	0.19	Academic Research
80	SHARCNET (Waterloo)	2.2 GHz Opteron + Gigabit Ethernet	3072	6015	0.17	Academic Research
83	SHARCNET (McMaster)	2.6 GHz Opteron + Quadrics	1536	5746	0.16	Academic Research
146 (previously 106)	WestGrid (UBC)	3.06 GHz Intel Xeon + Gigabit Ethernet	1680	3755	0.10	Acadmic Research
165 (previously 115)	Telco	3.2 GHz Intel Xeon + Gigabit Ethernet	1036	3755	0.10	Industrial
209	SHARCNET (Guelph)	2.2 GHz (Dual core) Opteron + Myrinet	1068 cores	3491	0.10	Academic Research
273 (previously 157)	RQCHP (Sherbrooke)	3.2 GHz Intel Xeon + Gigabit Ethernet	872	3064	0.09	Academic Research
385 (Previously 240)	Canadian Meterological Centre, Dorval	IBM p690	960	2560	0.07	Research (Weather)

Software Resources

- OS – Linux
  HP Opteron: HP XC3.0

- Compilers
  - Opteron: Pathscale (pathcc, pathCC, pathf90), PGI (pgcc, pgCC, pgf77/pgf90)
  - Alpha: compaq (ccc, cxx, fort)
  - Itanium/Xeon: Intel (icc, ifort)

- Scheduler
  - LSF, SQ

- Key parallel development support
  - MPI (HPMPI, MPICH, OPENMPI)
  - Multi-threading (pthreads, OpenMP)

Software Resources (continue)

- Libraries
  - ACML (AMD), CXML (Alpha), SCSL(SGI), ATLAS, GSL, ScaLAPACK, FFTW, PETSc, …

- Debugging/Profiling Tools
  - Debugging (DDT, gdb, …)
  - Profiling/Optimization (OPT, gprof,…)

- Application packages

  R, Blast, Gromacs, NWChem, Octave, …

- Commercial packages (cost share)

  - Gaussian: *Computational Chemistry Software*
  - Fluent: *Computational Fluid Dynamics Software*
  - Bring your license to SHARCNET (such as lumerical, …)

- Others. you ask/provide, we install

Details: http://www.sharcnet.ca/Facilities/software/softwarePage.php

File System Basics

Policy

- Same username/password across all systems, and webportal account.
- User self-management on webportal (site leader, sponsor, group member)
- Common home directory across SHARCNET (exceptions: wobbe, cat)
- SHARCNET-maintained software is in /opt/sharcnet
- /home backup

File system

pool	quota	expiry	purpose
/home	200 MB	none	sources, small config files
/work	none	none	active data files
/scratch	none	none	active data files
/tmp	160 GB	10 days	node-local scratch

- /scratch and /work are local to each cluster
  - not backed up
  - important: run jobs from /scratch or /work (performance!)

Job Scheduling (queue)

Same types of queues: mpi, threaded, serial, but priority is cluster specific.
For throughput (serial) clusters such as whale, we have

```
[jemmyhu@wha780 ~]$ 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
serial          80  Open:Active  -    -    -    -   1871     0  1871     0
threaded        80  Open:Active  -    -    -    -     64     0    64     0
mpi             40  Open:Active  -    -    -    -    840     0   840     0
[jemmyhu@wha780 ~]$
```

Test queue: preemptive, for debugging purpose only, limited cpu time (30 mins or 1 hr)

Job Scheduling (queue continue)

For systems with Quadrics/Myrinet interconnection, we have

```
[jemmyhu@nar317 ~]$ 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  -    -    -    -     16      0    16     0
mpi           80  Open:Active  -    -    -    -   1794   1128   650    16
serial        40  Open:Active  -    -    -    -    332      0   332     0
```

On bull (and greatwhite), we have

```
[jemmyhu@bl124 include]$ 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
gaussian     100  Open:Active   -   -    -     -     18      0    18     0
threaded      80  Open:Active   -   -    -     -     64     12    52     0
mpi           80  Open:Active   -   -    -     -    808    528   280     0
serial        40  Open:Active   -   -    -     -     22      0    22     0
```

Job Scheduling (queue commands – LSF, SQ)

- All jobs run on a batch mode (submitted to a queue)
  [No interactive job be allowed except on mako (devel cluster)]
- SQ commands (details follow on Part II):

- **sqsub**: submit a job to the system

```
sqsub –q mpi –n 8 –o logfile ./myCode
sqsub –q serial –o logfile ./myCode
```

- **sqjobs**: view job status

```
sqjobs –u all
sqjobs –u username (sqjobs)
```

- **sqkill**: kill a job

```
sqkill jobid
```

- **bhist**: view job history

```
bhist –l jobid
```

sqsub common options	
-q	job queue
-n	number of CPU
-i	redirect stdin
-o	redirect stdout

SHARCNET Support: Team

- HPC software specialist
  - Central resources for user support
  - software/development support
  - analysis of requirements
  - development/delivery of educational resources
  - research computing consultations

- System Administrators
  - user accounts
  - system software
  - hardware and software maintenance

http://www.sharcnet.ca/Help/index.php

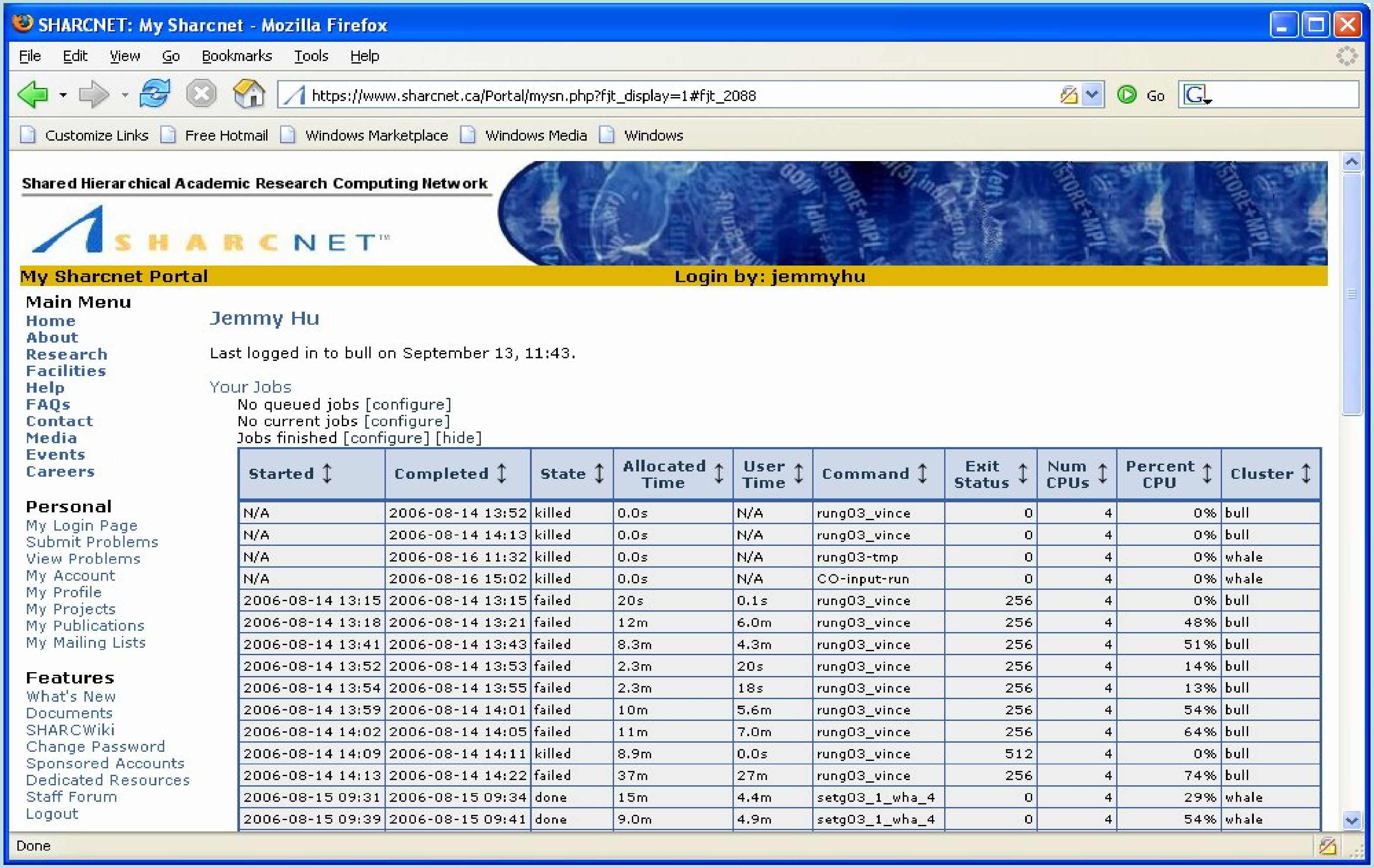
SHARCNET Support: Info on the Web

http://www.sharcnet.ca

- For general information: SHARCNET FAQ
  http://www.sharcnet.ca/Help/faq.php
- Up-to-date Facility information: facility page
  http://www.sharcnet.ca/Facilities/index.php
- Software page
  http://www.sharcnet.ca/Facilities/software/softwarePage.php
- Online problem tracking system (login needed)
  https://www.sharcnet.ca/Portal/problems/problem_search.php
- Supporting personal:
  http://www.sharcnet.ca/Help/index.php

SHARCNET Support: web tools

My Account



Shared Hierarchical Academic Research Computing Network

SHARCNET™

My Sharcnet Portal **Login by: jemmyhu**

Main Menu
Home
About
Research
Facilities
Help
FAQs
Contact
Media
Events
Careers

Personal
My Login Page
Submit Problems
View Problems
My Account
My Profile
My Projects
My Publications
My Mailing Lists

Features
What's New
Documents
SHARCWiki
Change Password
Sponsored Accounts
Dedicated Resources
Staff Forum
Logout

Jemmy Hu

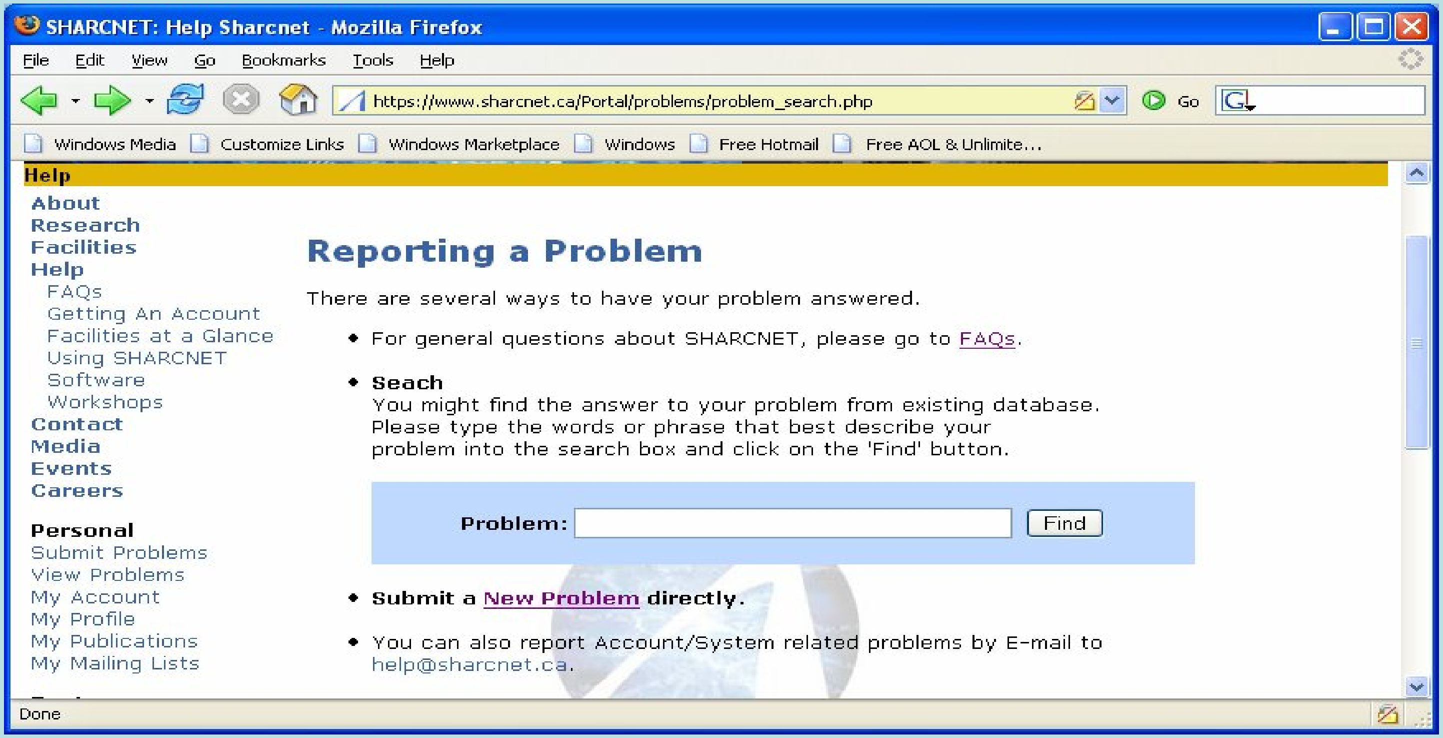
Last logged in to bull on September 13, 11:43.

Your Jobs
No queued jobs [configure]
No current jobs [configure]
Jobs finished [configure] [hide]

Started ↕	Completed ↕	State ↕	Allocated Time ↕	User Time ↕	Command ↕	Exit Status ↕	Num CPUs ↕	Percent CPU ↕	Cluster ↕
N/A	2006-08-14 13:52	killed	0.0s	N/A	rung03_vince	0	4	0%	bull
N/A	2006-08-14 14:13	killed	0.0s	N/A	rung03_vince	0	4	0%	bull
N/A	2006-08-16 11:32	killed	0.0s	N/A	rung03-tmp	0	4	0%	whale
N/A	2006-08-16 15:02	killed	0.0s	N/A	CO-input-run	0	4	0%	whale
2006-08-14 13:15	2006-08-14 13:15	failed	20s	0.1s	rung03_vince	256	4	0%	bull
2006-08-14 13:18	2006-08-14 13:21	failed	12m	6.0m	rung03_vince	256	4	48%	bull
2006-08-14 13:41	2006-08-14 13:43	failed	8.3m	4.3m	rung03_vince	256	4	51%	bull
2006-08-14 13:52	2006-08-14 13:53	failed	2.3m	20s	rung03_vince	256	4	14%	bull
2006-08-14 13:54	2006-08-14 13:55	failed	2.3m	18s	rung03_vince	256	4	13%	bull
2006-08-14 13:59	2006-08-14 14:01	failed	10m	5.6m	rung03_vince	256	4	54%	bull
2006-08-14 14:02	2006-08-14 14:05	failed	11m	7.0m	rung03_vince	256	4	64%	bull
2006-08-14 14:09	2006-08-14 14:11	killed	8.9m	0.0s	rung03_vince	512	4	0%	bull
2006-08-14 14:13	2006-08-14 14:22	failed	37m	27m	rung03_vince	256	4	74%	bull
2006-08-15 09:31	2006-08-15 09:34	done	15m	4.4m	setg03_1_wha_4	0	4	29%	whale
2006-08-15 09:39	2006-08-15 09:41	done	9.0m	4.9m	setg03_1_wha_4	0	4	54%	whale

SHARCNET Support: Problem Reporting

- Using online SHARCNET Problem Tracking System to submit any questions/problems.
  - Problems will be handled with appropriately by supporting staffs.
  - A record of the question together with the comments and final resolution will be tracked.
  - User view: submit, search, view, comment, reopen



Reporting a Problem

There are several ways to have your problem answered.

- For general questions about SHARCNET, please go to FAQs.
- **Seach**
  You might find the answer to your problem from existing database. Please type the words or phrase that best describe your problem into the search box and click on the 'Find' button.

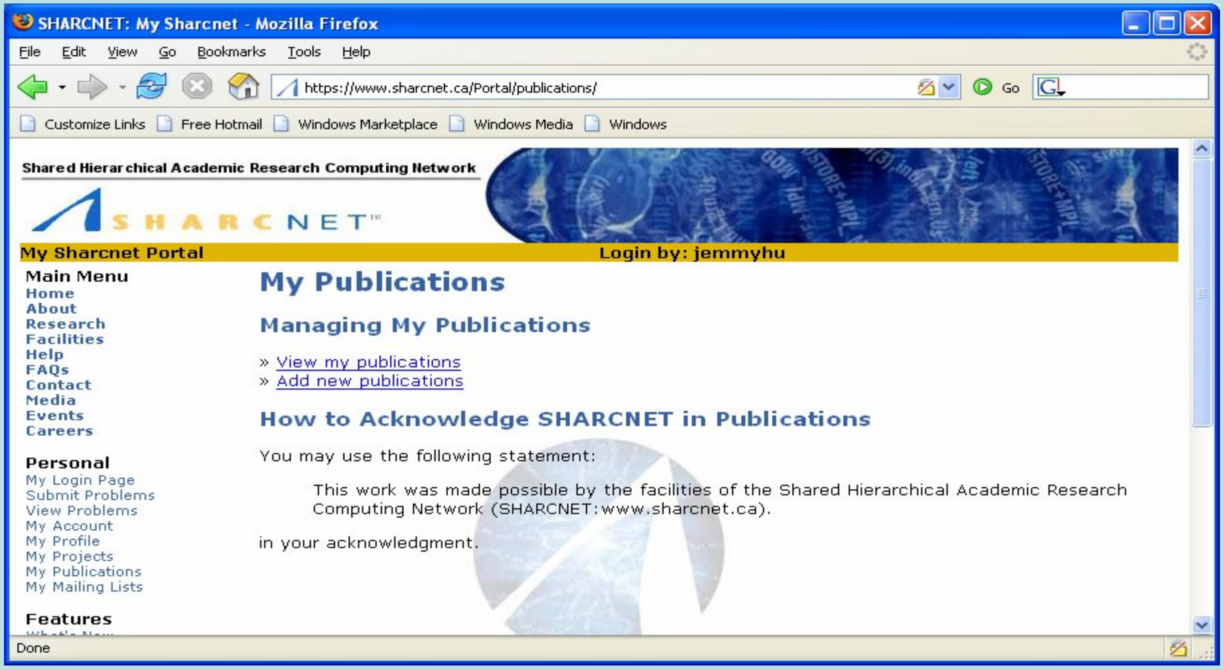
  Problem: ________ Find

- **Submit a New Problem directly.**
- You can also report Account/System related problems by E-mail to help@sharcnet.ca.

- Email to help@sharcnet.ca
- Email or phone supporting staff: http://www.sharcnet.ca/Help/index.php

User co-operation

- Efficient use of SHARCNET resources
- Contribute hardware/software
- Acknowledge SHARCNET



My Publications

Managing My Publications

» View my publications
» Add new publications

How to Acknowledge SHARCNET in Publications

You may use the following statement:

> This work was made possible by the facilities of the Shared Hierarchical Academic Research Computing Network (SHARCNET:www.sharcnet.ca).

in your acknowledgment.

End of Part I

Questions?

Part II: Practice Issues for how to use SHARCNET

Demo

- Login & File Transfer
- Compiler (default, options),
- Software/Library Usage
- Run Jobs
- LSF Email
- Problem Reporting
- Publication
- Parallel Programming

Login & File Transfer

- Login: only by SSH
  - domain name of SHARCNET systems:
    - external: machine.sharcnet.ca
    - internal: machine.sharcnet
  - Linux: ssh username@machine.sharcnet.ca
  - Windows: ssh available as part of Cygwin tools (http://www.cygwin.com), (http://ftp.ssh.com/pub/ssh/SSHSecureShellClient-3.2.9.exe) and many graphical clients available (PuTTY, etc.)
- File Transfer:
  - Linux: scp filename machine.sharcnet.ca:/scratch/username
  - Windows: ssh file transfer, e.g.:

Compilers

AMD Opteron clusters:

PathScale C/C++, Fortran compilers (pathcc, pathCC, pathf90) [default]

Portland Group (PGI) C/C++, Fortran compilers (pgcc, pgCC, pgf77/pgf90)

Alpha Cluster (greatwhite)

Compaq C/C++, Fortran compilers (ccc, cxx, fort)

Intel Itanium/Xeon: Intel compilers (silky)

Intel C/C++, Fortran compilers (icc, ifort)

(mako, spinner, coral)

GNU C/C++, GNU Fortran compilers (gcc, g77, gfortran)

Compilers: **SHARCNET compile script**

SHARCNET provides a “compile” script (use as “generic” compiler)

- tries to do what it “should” - optimizations, compiler, etc.
- recommended unless you know better

Command	Language	Extension	Example
cc	c	c	cc –o code.exe code.c
CC	C++	.C, .cc, .cpp, .cxx, c++	cc –o code.exe code.cpp
c++	C++		
cxx	C++		
f77	Fortran 77	.f, .F	f77 –o Fcode.exe Fcode.f
f90/f95	F90/F95	.f90, .f95, .F90, F95	f90 –o Fcode.exe Fcode.f90
mpicc	c	c	mpicc –o mpicode.exe mpicode.c
mpiCC	C++	C++	mpiCC –o mpicode.exe mpicode.cc
mpif77	Fortran 77	f77	mpif77 –o mpicode.exe mpicode.f
mpif90/mpif95	Fortran90/95	F90/f95	mpif90 –o mpicode.exe mpicode.f90

For OpenMP code: f90 –openmp –o openmpcode openmpcode.f90

Compilers: Commonly used options

There are minor differences between compilers (see man page for details).

e.g. -PathScale:

-c Do not link, generate object file only.

-o *file* Write output to specified *file* instead of default.

-I*dir* Add *dir* to search path for include files.

-l*library* Search the library named liblibrary.a or liblibrary.so such as –lmpi, -lacml

-L*path* add path to search path for the loader.

-g[N] Specify the level of debugging support produced by the compiler

- **-g0** No debugging information for symbolic debugging is produced. This is the default.
- **-g2** Produce additional debugging information for symbolic debugging.

-O[n] Optimization level n=0 to 3. Default is **-O2**.

- **-O0** Turns off all optimizations.
- **-O1** Turns on local optimizations that can be done quickly.
- **-O2** Turns on extensive optimization. This is the default
- **-O3** Turns on aggressive optimization. This includes but is not limited to turning on the Loop Nest Optimizer

-Ofast A single method for turning on a collection of optimizations.-Ofast equivalent to -O3 -ipa -OPT:Ofast -fno-math-errno

-pg Generate extra code to profile information suitable for the analysis program pathprof

-Wall Enable most warning messages.

Compilers: Examples

- **Basic:**

  ```
  pathcc -o executable source.c
  ```

- **Add optimizations:**

  ```
  pathcc -Ofast -o executable source.c
  ```

- **Specify additional include paths:**

  ```
  pathcc -I/work/project/include -c source.c
  ```

- **Link to ACML library (requires both -l and -L)**

  ```
  pathf90 -L/opt/sharcnet/acml/current/pathscale64/lib
                      -o executable source.c -lacml
  ```

- **SHARCNET defaults** (i.e. compile script):
  - Pathscale: -Ofast   Compaq: -fast   Intel: -O2 -xW -tpp

- ***Make*** is recommended for all compilation

Software/Library Usage

List and detailed usage on the web:
http://www.sharcnet.ca/Facilities/software/softwarePage.php

Basic Math Library: architecture specific

ACML (AMD),
CXML/CPML (Alpha),
MKL(Intel),
SCSL(SGI)
Atlas, ScaLapack, GSL

Likely included in the library path to make the usage simple
Fortran: simply add compile option –lacml, etc

```
f90 –lacml –o myfcode myfcode.f90
```

If head file needed (c/c++), add compile option –Idir

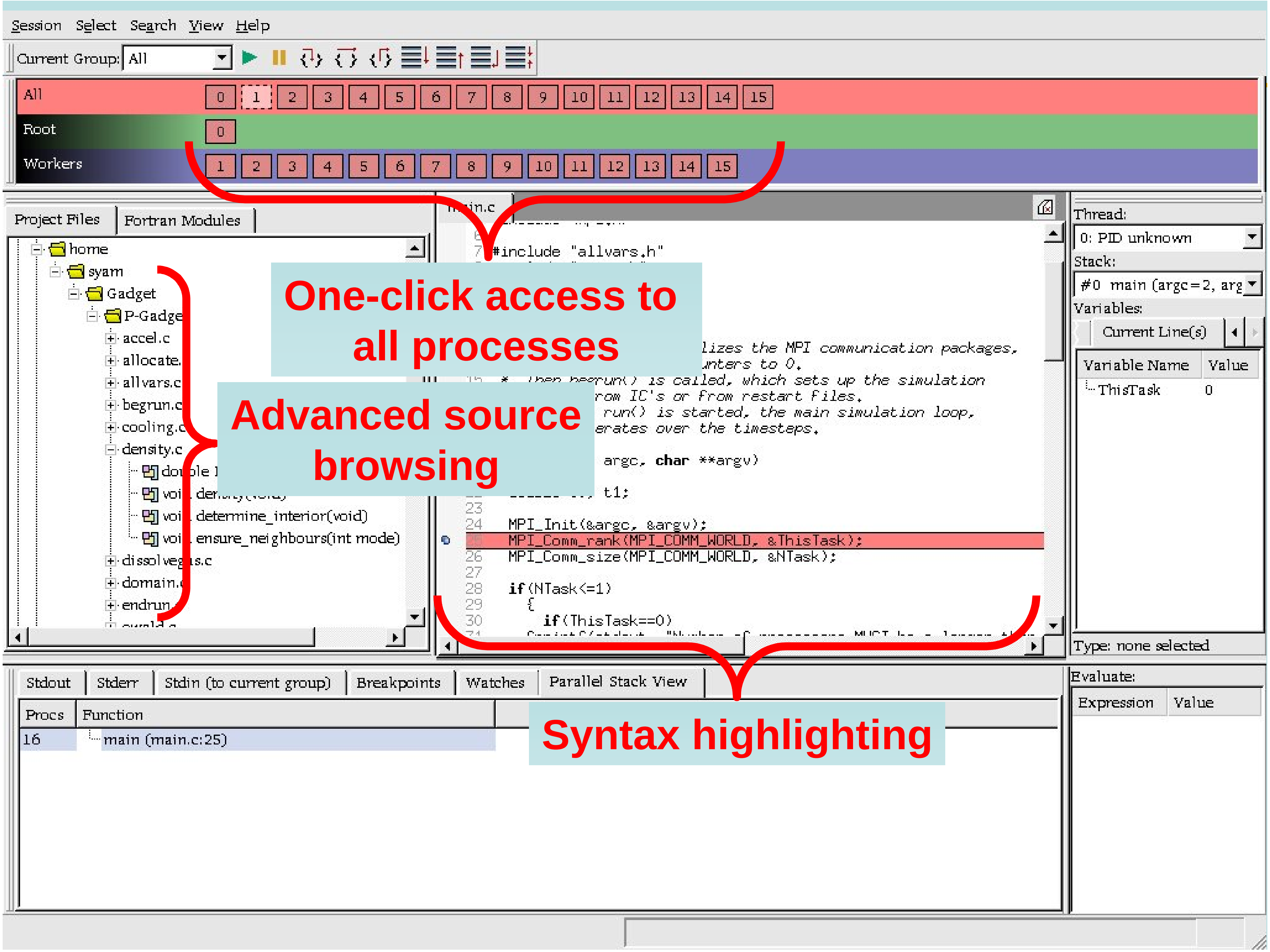
```
cc -I/opt/sharcnet/acml/current/pathscale64/include –lacml –o myCcode myCcode.c
```

DDT – the Distributed Debugging Tool

- A powerful debugger for parallel programs with an advanced GUI. Works best with MPI programs, but can also be used for threaded and serial jobs.
- Works with C, C++, and many flavours of Fortran (77, 90, 95).
- Installed on requin, narwhal, bull, and six PoP clusters.
- To use DDT, type

  ```
  ddt program [arguments]
  ```

  then choose the number of processes, and press “Submit”.
- DDT itself invokes the scheduler using the test queue. The debugging session starts almost immediately, but has a one hour time limit.



```
#include "allvars.h"

                     lizes the MPI communication packages,
                     unters to 0.
 *   Then begrun() is called, which sets up the simulation
                     rom IC's or from restart files.
                     run() is started, the main simulation loop,
                     erates over the timesteps.

                     argc, char **argv)

                     t1;

  MPI_Init(&argc, &argv);
  MPI_Comm_rank(MPI_COMM_WORLD, &ThisTask);
  MPI_Comm_size(MPI_COMM_WORLD, &NTask);

  if(NTask<=1)
    {
      if(ThisTask==0)
```

Thread: 0: PID unknown

Stack: #0 main (argc=2, arg

Variables:

Variable Name	Value
ThisTask	0

Procs	Function
16	main (main.c:25)

One-click access to all processes

Advanced source browsing

Syntax highlighting

Run Jobs Basics

- **Log on to the desired cluster/machine**
  - ensure files are in /scratch/username (or /work)
  - do not run jobs out of /home --- it can be very slow
  - jobs are submitted using the SQ system (wrapper, e.g. around LSF)

- **Job Category - Queue**

Job Type	**Queue**	**CPUs (default=1)**
Parallel	mpi	>2
Parallel	threaded (Machine dependent)	2 (requin, wobbe, cat) 2-4 (most clusters) >2 (silky)
Serial	serial	1

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** - displays historical information about jobs

sqsub (submit job)

```
sqsub [-o ofile][-i ifile][-t][-q queue][-n ncpus] command...
```

- **Commonly used options:**

  -q queue   queue name (serial, threaded, mpi; default serial)
  -n ncpus   require n cpus (default 1)
  -i ifile   job reads inputs from ifile (no default)
  -o ofile   job output goes to ofile (default: terminal unless batch)
  -o efile   job errors go to efile (default: ofile)
  -t or --test test mode: short but immediate

Examples (Demo)

```
sqsub –q mpi –n 4 –o hello-mpi.log ./hello-mpi
sqsub –q threaded –n 4 –o hello-openmp.log ./hello-openmp
sqsub –q serial –o hello.log ./hello
```

Submit to ‘**test**’ queue

```
sqsub –q mpi –t –n 4 –o hello-mpi.log ./hello-mpi
```

Show page

```
[jemmyhu@wha780 examples]$ pwd
/scratch/jemmyhu/examples

[jemmyhu@wha780 examples]$ sqsub -q mpi -t -n 4 -o hello_mpi_c.log ./hello_mpi_c
Job <134177> is submitted to queue <test>.

[jemmyhu@wha780 examples]$ sqjobs
 jobid queue state ncpus nodes time command
------ ----- ----- ----- ----- ---- -------------
134177  test     R     4  wha2   2s ./hello_mpi_c
2972 CPUs total, 196 idle, 2776 busy; 2020 jobs running; 1 suspended, 232 queued.

[jemmyhu@wha780 examples]$ 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   -    -    -    -      4      0     4     0
serial        80  Open:Active   -    -    -    -   2228    232  1996     0
threaded      80  Open:Active   -    -    -    -     16      0    16     0
mpi           40  Open:Active   -    -    -    -    960      0   760   200
[jemmyhu@wha780 examples]$
```

sqjobs (view job status)

sqjobs [-r][-q][-z][-v][-u user][-n][--summary][jobid...]

options:

-a or --all	show all jobs: all users and all states
-r	show running jobs
-q	show queued jobs
-z	show suspended/preempted jobs
-u user	show jobs for the given user
-n, --none	show one-line summary of cluster
--summary	show a line-per-user summary of all jobs
-h or --help	show usage
--man	show man page
jobid...	one or more jobids to examine

Show page

```
[jemmyhu@wha780 examples]$ sqjobs
[jemmyhu@wha780 examples]$ sqjobs -u jemmyhu
[jemmyhu@wha780 examples]$ sqjobs -u tao
 jobid queue state ncpus      nodes  time command
------ ----- ----- ----- ---------- ----- -------
133327   mpi     R     8 wha[31,33] 3439s ./prog
2972 CPUs total, 0 idle, 2972 busy; 2020 jobs running; 0 suspended, 275 queued.
[jemmyhu@wha780 examples]$ sqjobs -n
2972 CPUs total, 0 idle, 2972 busy; 2020 jobs running; 0 suspended, 275 queued.
[jemmyhu@wha780 examples]$ sqjobs --summary
    user     q          r          z          ncpus      njobs
--------     ---        ---        -          -----      -----
ikotsire      0         728        0          728        728
 racinej      0         1          0          624          1
     jdc      0         462        0          462        462
  garicb      0         320        0          320        320
   ricky    253         313        0          313        566
jeffchen     10          62        0           62         72
 mjchase      0          40        0           40         40
  maryam      1           4        0           16          5
   ……
```

sqkill (kill a job)

sqkill jobid [jobids...]

```
[jemmyhu@wha780 examples]$ sqjobs
[jemmyhu@wha780 examples]$ sqsub -q mpi -t -n 4 -o hello_mpi_c.log ./hello_mpi_c
Job <134227> is submitted to queue <test>.
[jemmyhu@wha780 examples]$ sqjobs
 jobid queue state ncpus nodes time command
------ ----- ----- ----- ----- ---- -------------
134227  test     Q     4          3s ./hello_mpi_c
2972 CPUs total, 0 idle, 2972 busy; 2020 jobs running; 0 suspended, 269 queued.
[jemmyhu@wha780 examples]$ sqkill 134227
Job <134227> is being terminated
[jemmyhu@wha780 examples]$ sqjobs
[jemmyhu@wha780 examples]$
```

May need to wait for a few seconds for the system to kill a job

bhist (display job history info)

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

OPTIONS

-a Displays information about both finished and unfinished jobs. This option overrides -d, -p, -s, and -r.

-b Brief format. Displays the information in a brief format. If used with the -s option, shows the reason why each job was suspended.

-d Only displays information about finished jobs.

-l Long format. Displays additional information.

-u user_name Displays information about jobs submitted by the specified user

Part II: Practice Issues for how to use SHARCNET

```
[jemmyhu@wha780 examples]$ bhist -u jemmyhu
No matching job found
[jemmyhu@wha780 examples]$ bhist -a
Summary of time in seconds spent in various states:
JOBID   USER    JOB_NAME  PEND   PSUSP  RUN  USUSP  SSUSP  UNKWN  TOTAL
134177  jemmyhu *o_mpi_c  8      0      37   0      0      0      45
134227  jemmyhu *o_mpi_c  10     0      10   0      0      0      20
[jemmyhu@wha780 examples]$ bhist -l 134177
Job <134177>, User <jemmyhu>, Project <jemmyhu>, Job Group </jemmyhu/jemmyhu>,
                     Command </opt/hpmpi/bin/mpirun -srun -o hello_mpi_c.log ./
                     hello_mpi_c>
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/jemmyhu>, Execution CWD
                     </scratch/jemmyhu/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
```

LSF Email (pass)

- Default (off?)
- option to turn on LSF email in sqsub

  -m

Problem Reporting: web showcase

- Access problem, check facility page (state, notice)
  http://www.sharcnet.ca/Facilities/index.php

- Submit problem online
  https://www.sharcnet.ca/Portal/problems/problem_search.php

- View/comment your own problems
  https://www.sharcnet.ca/Portal/problems/problem_get.php

Submit Publication

- To add a new application

  https://www.sharcnet.ca/Portal/publications/publication_add.php

- To view publications

  https://www.sharcnet.ca/Portal/publications/publication_update.php

High Performance Computing

- Definition is nebulous
  - resource (processing) intensive computation
  - computing where the need for speed is compelling
    - computing nearer the limit of what is feasible
  - parallel computing (this is too strict)

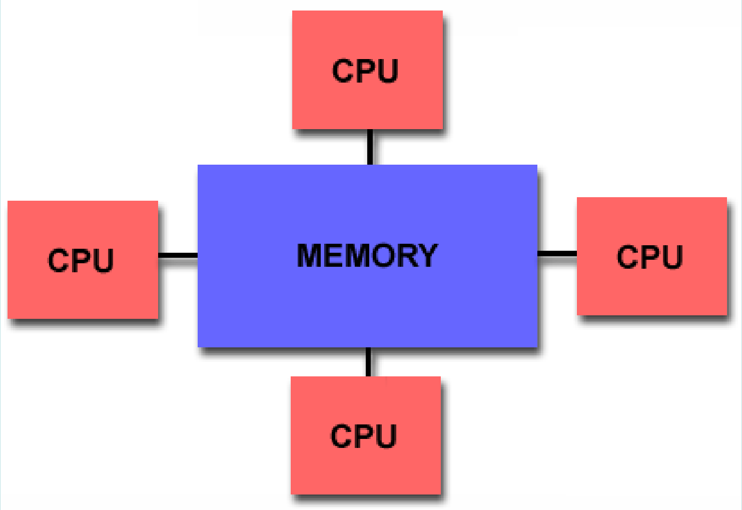
- In reality, HPC is concerned with varied issues involving:
  - hardware
    - pipelining, instruction sets, multi-processors, inter-connects
  - algorithms
    - efficiency, techniques for concurrency
  - software
    - compilers (optimization/parallelization), libraries

Problems Faced

- Hardware
  - in order to facilitate processors working together they must be able to communicate
  - interconnect hardware is complex
    - sharing memory is easy to say, harder to realize as system scales
    - communication over any kind of network is still painfully slow compared to bus speed --- overhead can be significant
- Software
  - parallel algorithms are actually fairly well understood
  - the realization of algorithms in software is non-trivial
  - compilers
    - automated parallelism is difficult
  - design
    - portability and power are typically at odds with each other

Parallel Computer Memory Architectures: Shared Memory

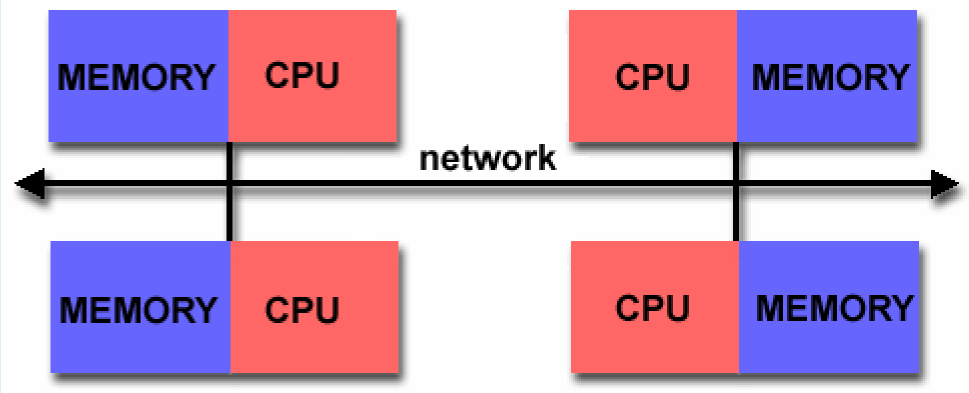
- Shared memory parallel computers vary widely, but generally have in common the ability for all processors to access all memory as global address space.



- Multiple processors can operate independently but share the same memory resources.
- Changes in a memory location effected by one processor are visible to all other processors.
- Shared memory machines can be divided into two main classes based upon memory access times: ***UMA*** and ***NUMA***.

Parallel Computer Memory Architectures:
Distributed Memory

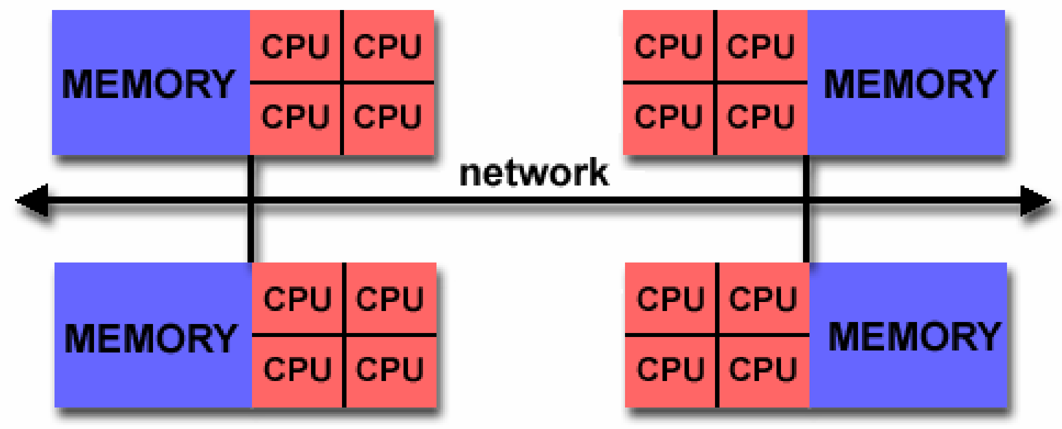
- A communication network to connect inter-processor memory
- Processors have their own local memory. Memory addresses in one processor do not map to another processor, so there is no concept of global address space across all processors.
- When a processor needs access to data in another processor, it is usually the task of the programmer to explicitly define how and when data is communicated. Synchronization between tasks is likewise the programmer's responsibility.
- The network "fabric" used for data transfer varies widely, though it can be as simple as Ethernet.



Parallel Computer Memory Architectures:

Hybrid Distributed-Shared Memory

- Employ both shared and distributed memory architectures
- The shared memory component is usually a cache coherent SMP machine. Processors on a given SMP can address that machine's memory as global.
- The distributed memory component is the networking of multiple SMPs. SMPs know only about their own memory - not the memory on another SMP. Therefore, network communications are required to move data from one SMP to another.
- Current trends seem to indicate that this type of memory architecture will continue to prevail and increase at the high end of computing for the foreseeable future.
- Advantages and Disadvantages: whatever is common to both shared and distributed memory architectures.



Parallel Computing

- The general idea is if one processor is good, many processors will be better
- Parallel programming is not generally trivial
  - tools for automated parallelism are either highly specialized or absent
- Many issues need to be considered, many of which don’t have an analog in serial computing
  - data vs. task parallelism
  - problem structure
  - parallel granularity

Message Passing Interface (MPI)

- Library providing message passing support for parallel/distributed applications
  - not a language: collection of subroutines (Fortran), functions/macros (C)
  - explicit communication between processes
- Advantages
  - standardized
  - scalability generally good
  - memory is local to a process (debugging/performance)
- Disadvantages
  - more complex than implicit techniques
  - communication overhead

MPI Programming Basics

- **Basic functionality, the ability to:**
  - Start Processes
  - Send Messages
  - Receive Messages
  - Synchronize
- **Core Functions**
  - MPI_Init()
  - MPI_Finalize()
  - MPI_Comm_rank()
  - MPI_Comm_size()
  - MPI_Send()
  - MPI_Recv()
  - MPI_BCAST()
  - MPI_REDUCE()

MPI library

- Include MPI header file
  - C
    - #include “mpi.h”
  - Fortran
    - include “mpif.h”

- Compile with MPI library
  - mpicc
  - mpif90

Example: MPI *Hello, world!*

```
#include <stdio.h>
#include “mpi.h”

int main(int argc, char *argv[])
{
   int rank, size;

   MPI_Init(&argc, &argv);       /* starts MPI */
   MPI_Comm_rank(MPI_COMM_WORLD, &rank); /* get current process id */
   MPI_Comm_size(MPI_COMM_WORLD, &size);    /* get number of processes */

   printf(“Hello, world! from process %d of %d\n”, rank, size);

   MPI_Finalize();        /* end of mpi*/

   return(0);
}
```

Compile and Run Result

- **Compile**

```
mpicc –o hello_world_c hello_world.c
```

- **Submit a mpi job**

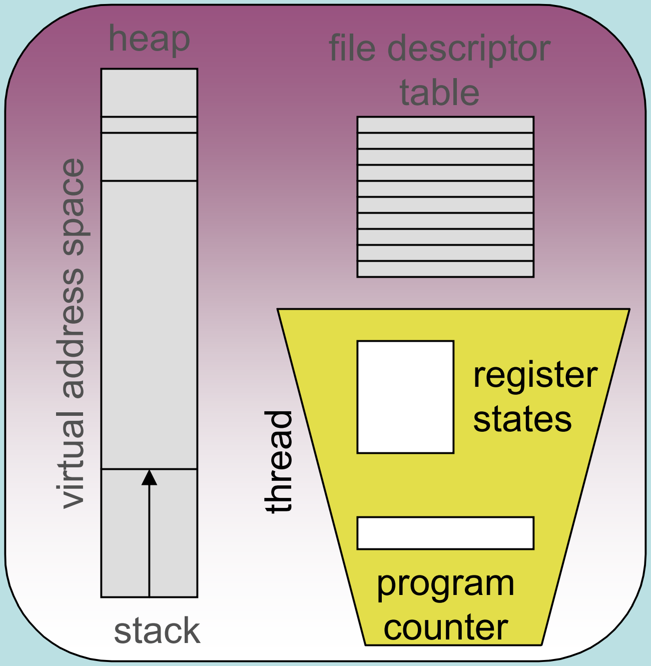
```
sqsub –q mpi –n 4 –o hello.log ./hello_world_c
```

- **Run Result (use 4 cpus)**

```
Hello, world! from process 0 of 4
Hello, world! from process 1 of 4
Hello, world! from process 2 of 4
Hello, world! from process 3 of 4
```

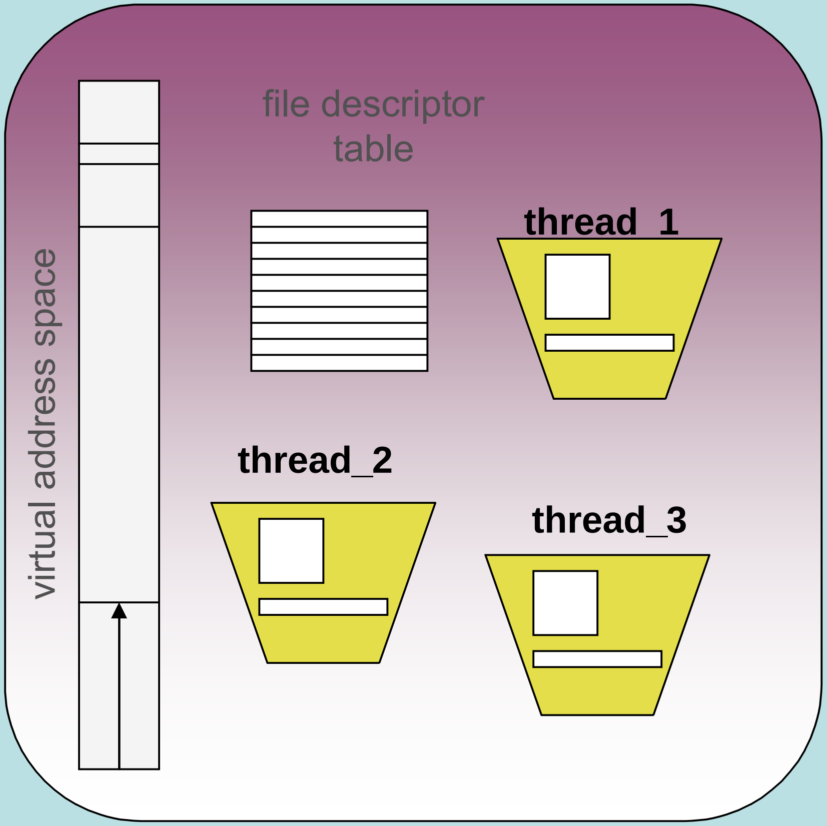
Threads

- DEF’N: a thread is a sequence of executable code *within* a process
- A serial process can be seen, at its simplest, as a single thread (a single “thread of control”)
  - represented by the program counter
  - sequence (increment PC), iteration/conditional branch (set value of PC)
- In terms of record-keeping, only a small subset of a process is relevant when considering a thread
  - register states; program counter



Conceptual View of a Thread (simplified)

Multi-threading



- Distribute work by defining multiple threads to do the work
  - e.g. OpenMP, pthreads
- Advantages
  - all process resources are implicitly shared (memory, file descriptors, etc.)
  - overhead incurred to manage multiple threads is relatively low
  - looks much like serial code
- Disadvantages
  - all data being implicitly shared creates a world of hammers, and your code is the thumb
  - exclusive access, contention, etc.

OpenMP Basics

- Using **compiler directives**, **library routines** and **environment variables** to automatically generate threaded (or multi-process) code that can run in a concurrent or parallel environment.
- A proposed industry standard. API for shared memory programming.
- Available for C/C++, FORTRAN, *appearing as macros*, through compiler support

- C: Include omp header file

  #include “omp.h”
- compile with –openmp flag

  f90 –openmp –o hello_openmp hello_openmp.f

OpenMP Example: Hello World

```
      PROGRAM HELLO
      INTEGER ID, NTHRDS
      INTEGER OMP_GET_THREAD_NUM, OMP_GET_NUM_THREADS
!$OMP PARALLEL PRIVATE(ID)
      ID = OMP_GET_THREAD_NUM()
      PRINT *, 'HELLO WORLD FROM THREAD', ID
!$OMP BARRIER
      IF ( ID .EQ. 0 ) THEN
        NTHRDS = OMP_GET_NUM_THREADS()
        PRINT *, 'THERE ARE', NTHRDS, 'THREADS'
      END IF
!$OMP END PARALLEL
      END
```

Compile and Run Result

- **Compile**

```
f90 –openmp  –o  hello_openmp_f  hello_world_openmp.f
```

- **Submit job**

```
sqsub –q threaded –n 4 –o hello_openmp.log ./hello_openmp_f
```

- **Run Results** (use 4 cpus)

```
HELLO WORLD FROM THREAD 2
HELLO WORLD FROM THREAD 0
HELLO WORLD FROM THREAD 3
HELLO WORLD FROM THREAD 1
THERE ARE 4 THREADS
```

Pthreads Basics

- Include Pthread header file
  - #include “pthread.h”
- Compile with Pthreads support/library
  - cc -pthread …
    - compiler vendors may differ in their usage to support pthreads (link a library, etc.)
    - when in doubt, consult the man page for the compiler in question

Example: “Hello, world!”

```
#include <stdio.h>
#include “pthread.h”

void output (int *);

int main(int argc, char *argv[])
{
   int id;
   pthread_t thread[atoi(argv[1])];

   for (id = 0; id < atoi(argv[1]); id++)
      pthread_create(&thread[id], NULL, (void *)output, (void *)&id);

   return(0);
}

void output(int *thread_num)
{
   printf(“Hello, world! from thread %d\n”, *thread_num);
}
```

Thank You.

Questions?