Introduction to SciNet

SciNet HPC Consortium
Compute Canada

August 26, 2011

Don't Panic

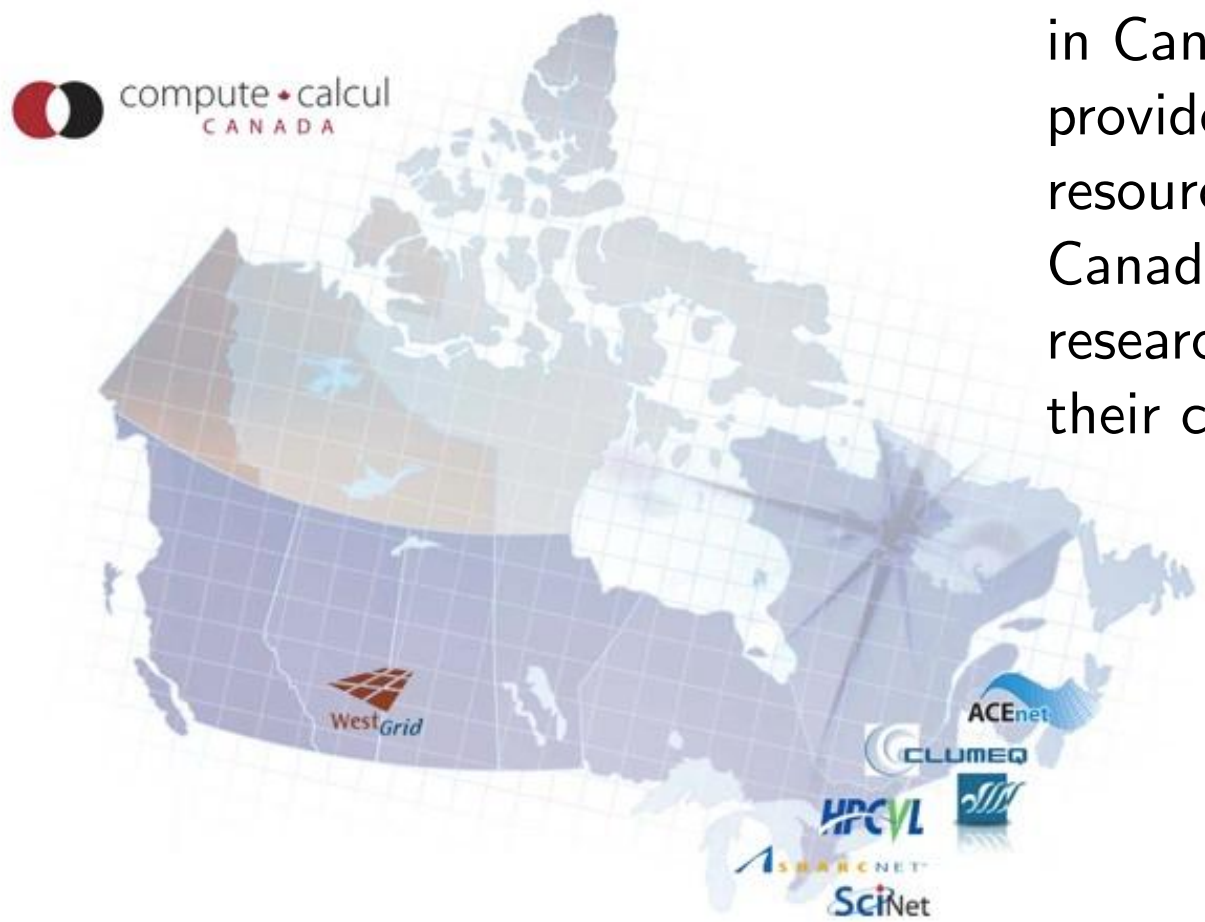
Outline

Part I

ABOUT SCINET

SciNet is ...

. . . a consortium for High-Performance Computing consisting of researchers at U. of T. and its associated hospitals.



One of 7 consortia in Canada that provide HPC resources to Canadian academic researchers and their collaborators.

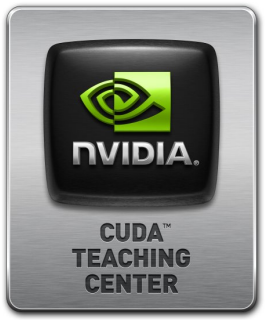
SciNet
compute • calcul
CANADA

SciNet is ...

. . . home to the 1st and 5th largest supercomputers in Canada.
(and some more)

SciNet is ...

- where you go for courses on a wide range of computational topics. Examples:
  - Intro to GPGPU with CUDA
  - Practical Parallel Programming Intensive
  - Intro to Scientific Programming with Modern FORTRAN
  - Intro to Scientific Programming with C++
  - Parallel I/O
  - Scientific Computing
- recognized by NVIDIA as both a CUDA research and teaching centre



- where you can meet with other users, and present your work, at the monthly SciNet User Group Meetings (SNUGs).

SciNet compute • calcul CANADA

SciNet is ...

. . . 4 technical analysts who can work directly with you to use our resources to produce good science.

- Jonathan Dursi
- Scott Northrup
- Ramses van Zon
- Daniel Gruner

+ 7 people that make sure everything runs smoothly.

- Jaime Pinto
- Joseph Chen
- Jason Chong
- Ching-Hsing Yu
- Neil Knecht
- Leslie Groer
- Chris Loken

+ Technical director Prof. Richard Peltier
+ Business manager Teresa Henriques
+ Project coordinator Jillian Dempsey

SciNet is ...

. . . 4 technical analysts who can work directly with you to use our resources to produce good science.

- Jonathan Dursi
- Scott Northrup
- Ramses van Zon
- Daniel Gruner

+ 7 people that make sure everything runs smoothly.

- Jaime Pinto
- Joseph Chen
- Jason Chong
- Ching-Hsing Yu
- Neil Knecht
- Leslie Groer
- Chris Loken

+ Technical director Prof. Richard Peltier
+ Business manager Teresa Henriques
+ Project coordinator Jillian Dempsey

SciNet is ...

. . . 4 technical analysts who can work directly with you to use our resources to produce good science.

- Jonathan Dursi
- Scott Northrup
- Ramses van Zon
- Daniel Gruner

+ 7 people that make sure everything runs smoothly.

- Jaime Pinto
- Joseph Chen
- Jason Chong
- Ching-Hsing Yu
- Neil Knecht
- Leslie Groer
- Chris Loken

+ Technical director Prof. Richard Peltier
+ Business manager Teresa Henriques
+ Project coordinator Jillian Dempsey

SciNet compute • calcul CANADA

SciNet is ...

. . . 4 technical analysts who can work directly with you to use our resources to produce good science.



- Jonathan Dursi
- Scott Northrup
- Ramses van Zon
- Daniel Gruner

+ 7 people that make sure everything runs smoothly.

- Jaime Pinto
- Joseph Chen
- Jason Chong
- Ching-Hsing Yu
- Neil Knecht
- Leslie Groer
- Chris Loken

+ Technical director Prof. Richard Peltier
+ Business manager Teresa Henriques
+ Project coordinator Jillian Dempsey

SciNet is ...

. . . 4 technical analysts who can work directly with you to use our resources to produce good science.




- Jonathan Dursi
- Scott Northrup
- Ramses van Zon
- Daniel Gruner

+ 7 people that make sure everything runs smoothly.

- Jaime Pinto
- Joseph Chen
- Jason Chong
- Ching-Hsing Yu
- Neil Knecht
- Leslie Groer
- Chris Loken

+ Technical director Prof. Richard Peltier
+ Business manager Teresa Henriques
+ Project coordinator Jillian Dempsey

How to get an account

Any qualified researcher at a Canadian university can get a SciNet account through this two-step process:

1. Register for a Compute Canada Database (CCDB) account
2. Non-faculty need a sponsor (supervisor's CCRI number), who has to have a SciNet account already.
3. Login to CCDB and apply for a SciNet account (click *Apply* beside SciNet on the *Consortium Accounts* page)

How to get an account

Default account

- Allows to simultaneously run 32 jobs of 48 hours wall-time.
- Use of maximally 256 cores.
- So in a year you could use over 2 million core-hours.
- 10GB of storage (plus a bunch of temporary scratch storage)

How to get more resources on SciNet

- Users who will be needing more than the default amount of resources (compute cycles and/or storage) must have their PI apply for it through the competitively awarded.
- Resource competition occurs in the fall of each year.
- Having an allocation also increases your priority in the queue.

Part II

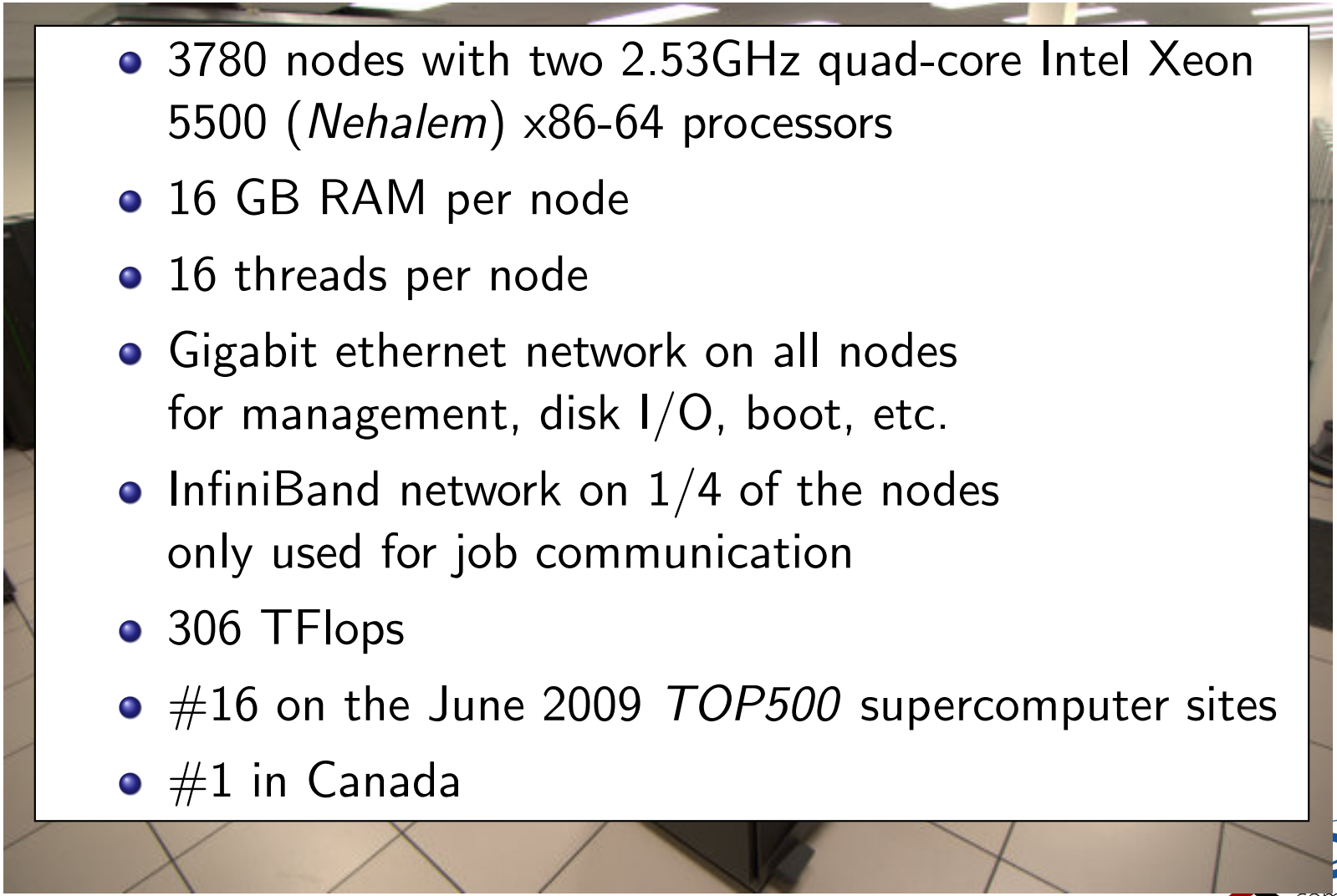
SCINET SYSTEMS

Resources at SciNet

General Purpose Cluster (GPC)

Resources at SciNet

General Purpose Cluster (GPC)



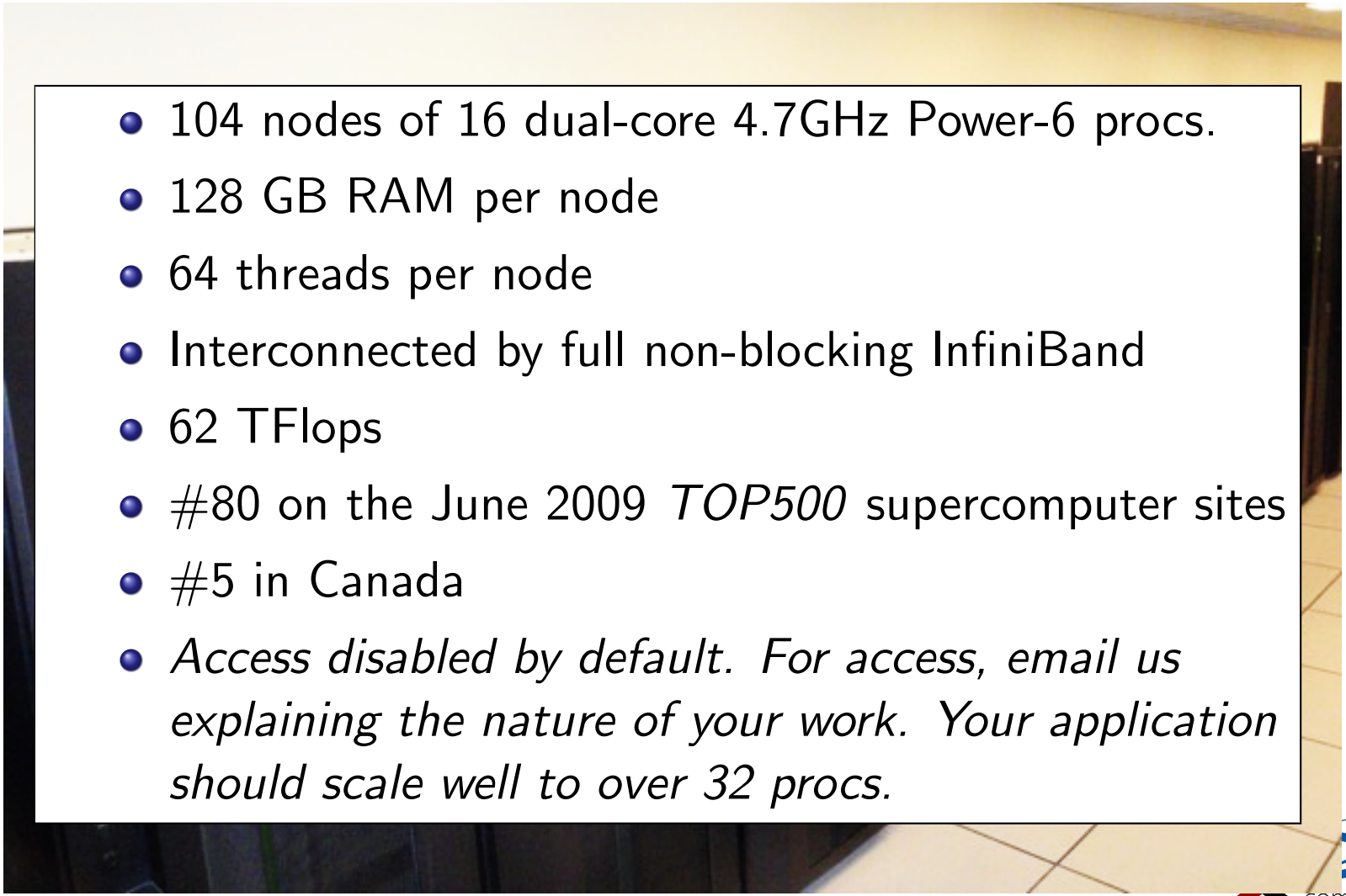
- 3780 nodes with two 2.53GHz quad-core Intel Xeon 5500 (*Nehalem*) x86-64 processors
- 16 GB RAM per node
- 16 threads per node
- Gigabit ethernet network on all nodes for management, disk I/O, boot, etc.
- InfiniBand network on 1/4 of the nodes only used for job communication
- 306 TFlops
- #16 on the June 2009 *TOP500* supercomputer sites
- #1 in Canada

Resources at SciNet

Tightly Coupled System (TCS)

Resources at SciNet

Tightly Coupled System (TCS)



- 104 nodes of 16 dual-core 4.7GHz Power-6 procs.
- 128 GB RAM per node
- 64 threads per node
- Interconnected by full non-blocking InfiniBand
- 62 TFlops
- #80 on the June 2009 *TOP500* supercomputer sites
- #5 in Canada
- *Access disabled by default. For access, email us explaining the nature of your work. Your application should scale well to over 32 procs.*

Resources at SciNet

Power 7 linux cluster (P7)

Resources at SciNet

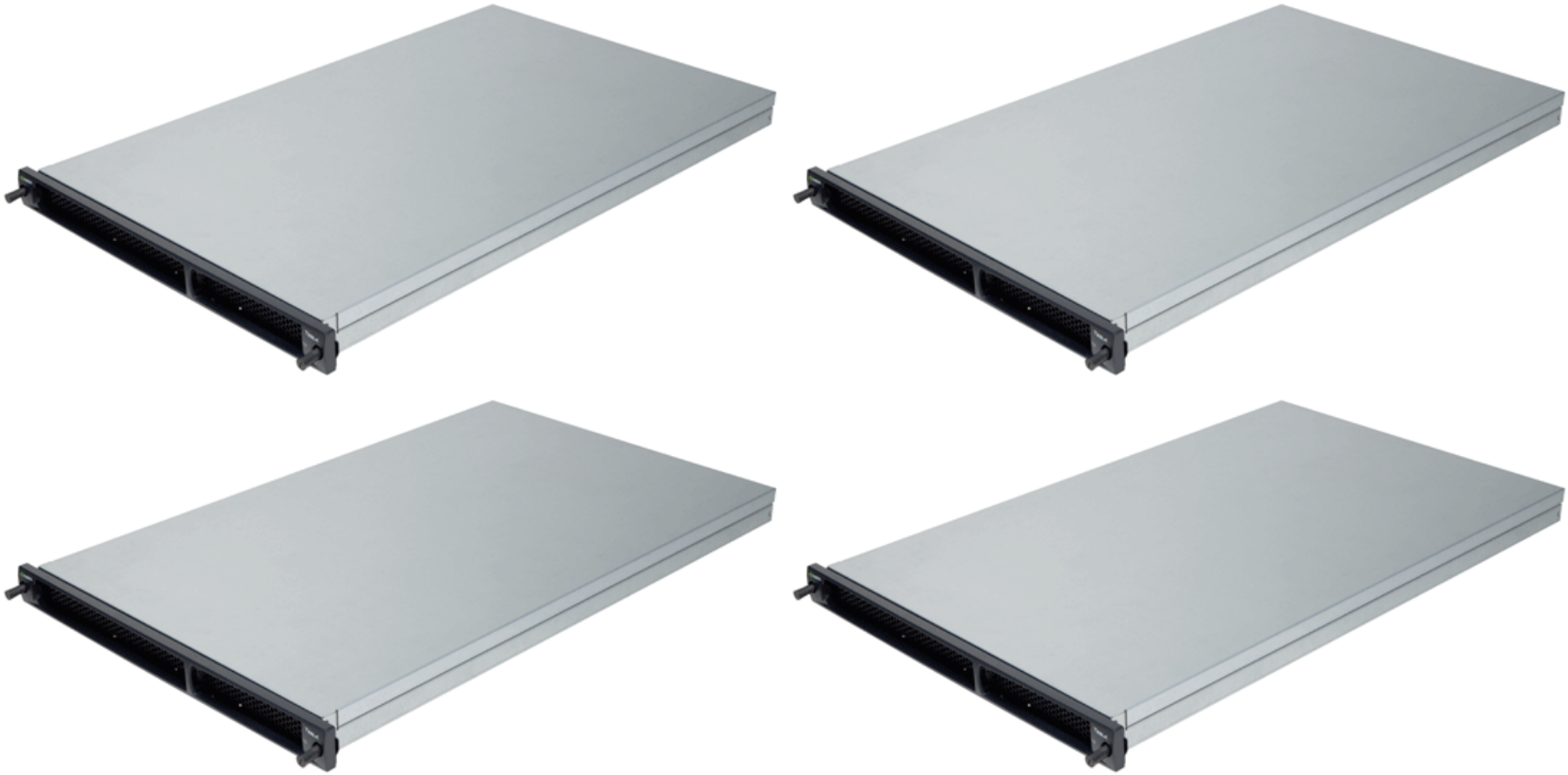
Power 7 linux cluster (P7)

- 5 (will be 8) nodes with 4 8-core 3.3GHz Power7s.
- 128 GB RAM per node
- 128 threads per node
- Interconnected by InfiniBand
- 4.2 (will be 6.7) TFlops peak
- Accessable to TCS users

SciNet compute • calcul CANADA

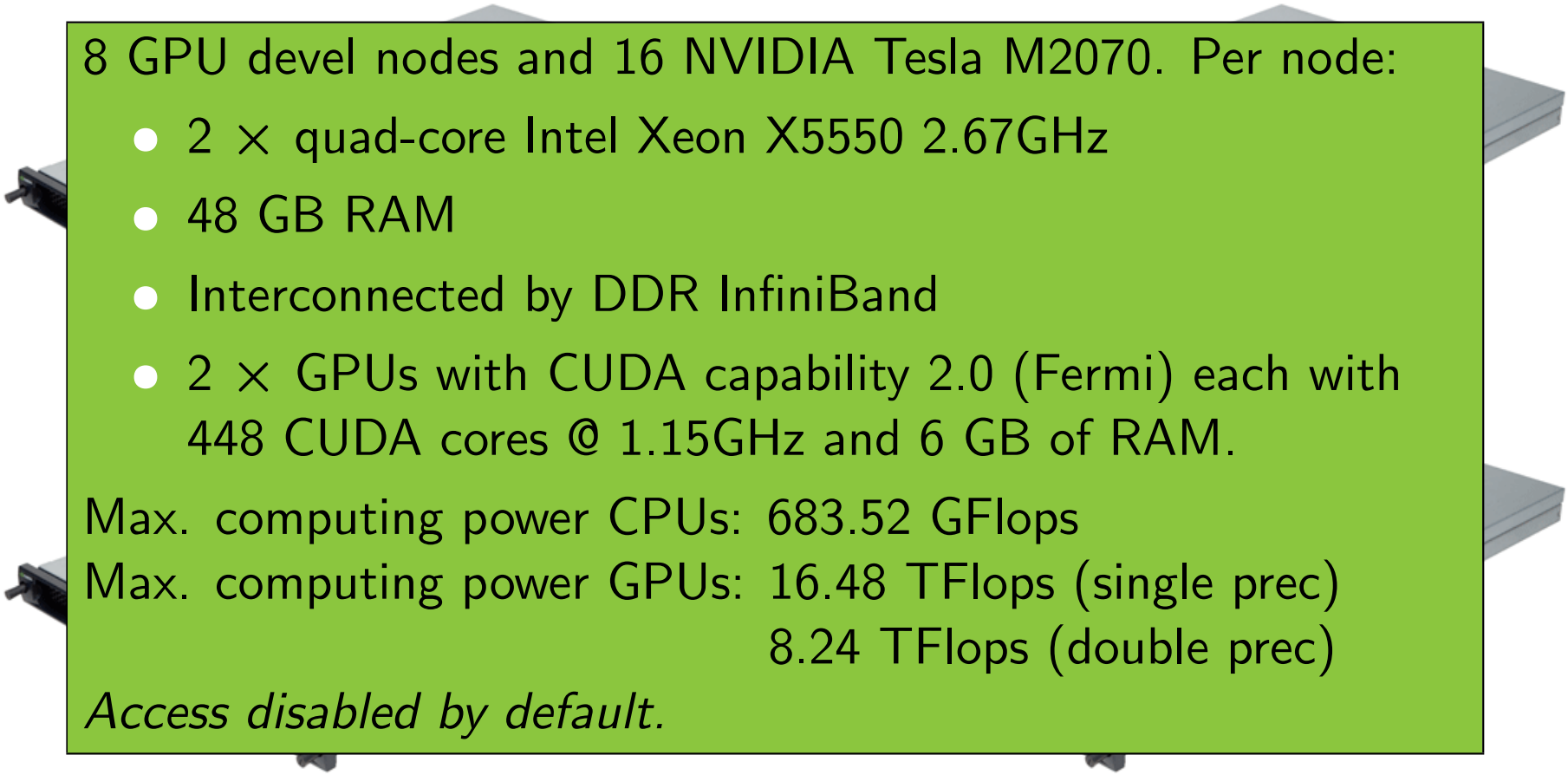
Resources at SciNet

Accelerator Research Cluster (ARC)



Resources at SciNet

Accelerator Research Cluster (ARC)



8 GPU devel nodes and 16 NVIDIA Tesla M2070. Per node:

- 2 × quad-core Intel Xeon X5550 2.67GHz
- 48 GB RAM
- Interconnected by DDR InfiniBand
- 2 × GPUs with CUDA capability 2.0 (Fermi) each with 448 CUDA cores @ 1.15GHz and 6 GB of RAM.

Max. computing power CPUs: 683.52 GFlops
Max. computing power GPUs: 16.48 TFlops (single prec)
8.24 TFlops (double prec)

Access disabled by default.

SciNet compute • calcul CANADA

Resources at SciNet

Disk space

- 1790 1TB disk drives, for a total of 1.4 PB of storage
- Two DCS9900 couplets, each delivering 4-5GB/s (r/w)
- Single *shared* file system GPFS on both TCS and GPC
- I/O goes over Gb/s ethernet network on the GPC, and over the infiniband network on the TCS
- Your files go in /home/user and /scratch/user.

Storage space

- HPSS: Tape-backed storage expansion solution.

location	quota	block-size	time-limit	backup	devel	comp
/home	10GB	256kB	perpetual	yes	rw	ro
/scratch	20TB/1M	4MB	3 months	no	rw	rw

Part III

USING SCINET

Get on the system

1. Access systems: login.scinet.utoronto.ca

First ssh to login (not part of clusters):

```
ssh -l <username> login.scinet.utoronto.ca [-Y]
```

The login nodes are gateways, they are not part of any of the clusters and they are only to be used for small data transfer and to proceed logging into one of the devel nodes.

2. Go to the right cluster: ssh to the devel nodes

GPC:	`gpc01, gpc02, gpc03, gpc04`
ARC:	`arc01`
TCS:	`tcs01, tcs02`
P7:	`p701`

SciNet compute • calcul CANADA

Software and Libraries

Once you log into devel nodes, what software is already installed?

- Other than essentials, all software installed as modules.
- modules set environment variables `LD_LIBRARY_PATH`, `PATH`, ...)
- Allows multiple, conflicting versions of package to be available.
- More on *Software and Libraries* page of wiki.

```
gpc-f103n084-$ module avail

------ /scinet/gpc/Modules/version_indepen
3.2.6

------ /scinet/gpc/Modules/3.2.6/modulefiles
dot                 modules             use.own
module-cvs          use.deprecated
module-info         use.experimental

------- /scinet/gpc/Modules/modulefiles --
ROOT/5.26.00
Xlibraries/X11-32
Xlibraries/X11-64(default)
amber10/amber10
autoconf/autoconf-2.64
blast/2.2.23+
cmake/2.8.0
...
```

http://wiki.scinet.utoronto.ca

Software and Libraries

`module load <module-name>`	use particular software
`module purge`	remove currently loaded modules
`module avail`	list available software packages
`module list`	list loaded modules
`module help <module-name>`	describe a module

- Load frequently used modules in `.bashrc` in home directory.
- Load run-specific modules inside the job script.
- Short name gives default (e.g. `intel` $\longrightarrow$ `intel/intel-v11.1.072`)
- To compile code that uses that a library from a module, add

```
-I${SCINET_[shortmodulename]_INC}
```

- To link, add

```
-L${SCINET_[shortmodulename]_LIB}
```

Dependencies

- Modules sometimes require other modules to be loaded first.
- Module will let you know if you didn't.
- For example:

```
gpc-f103n084-$ module purge

gpc-f103n084-$ module load python
python/2.6.2(11):ERROR:151:  Module 'python/2.6.2' depends on one of
the module(s) 'gcc/4.4.0'
python/2.6.2(11):ERROR:102:  Tcl command execution failed:  prereq gcc/4.4.0
gpc-f103n084-$ module load gcc python

gpc-f103n084-$
```

Commercial Software?

- SciNet has an extremely large and broad user base (∼900 users)

⇒ *Cannot buy everyone's favourite commercial software package.*

- Only commercial software we have installed is software that can benefit everyone:
  - GPC and ARC: Intel compilers, MKL (both in module intel)
  - TCS and P7: IBM compilers, ESSL
  - GPC, ARC, TCS and P7: DDT debugger from Allinea
- No Matlab, Gaussian, IDL, ...
- Can work with you to get commercial software that you have license for installed.

Compiling on SciNet systems

GPC compilers

- From `login.scinet.utoronto.ca`, ssh to one of the four devel nodes.

  ```
  ssh gpc04 [-Y]
  ```

- We recommend Intel compilers, which are

  ```
  icc, icpc, ifort
  ```

  for C, C++, and Fortran, respectively (from the module `intel`)
- Optimize your code for the GPC machine using of at least

  ```
  -O3 -xhost
  ```

- Add `-openmp` to the command line for OpenMP
- Compile MPI code with `mpif77/mpif90/mpicc/mpicxx`.
  1. Open MPI, in module `openmpi` (v1.4.1) default
  2. Intel MPI, in module `intelmpi` (v4.0.0)

SciNet compute • calcul CANADA

Compiling on SciNet systems

TCS compilers

- ssh to a devel node

  `ssh tcs01` *or* `ssh tcs02`

- Use IBM compilers: `xlc,xlC,xlf` for C, C++, and Fortran (`module load vacpp xlf` for the latest version)
- For OpenMP, use `xlc_r,xlC_r,xlf_r`.
- For MPI, `mpcc,mpCC,mpxlf` are the mpi wrappers.
- Suggested compiler flags:

  `-q64 -O3 -qhot -qarch=auto -qtune=auto`

  supplemented by `-qsmp=omp -O4` for OpenMP programs.
- On the link line we suggest using

  `-q64 -bdatapsize:64k -bstackpsize:64k`

  also supplemented by `-qsmp=omp -O4` for OpenMP programs.

Compiling on SciNet systems

TCS compilers and P7 compilers

- ssh to a devel node

  `ssh tcs01` *or* `ssh tcs02`

- Use IBM compilers: `xlc,xlC,xlf` for C, C++, and Fortran (`module load vacpp xlf` for the latest version)
- For OpenMP, use `xlc_r,xlC_r,xlf_r`.
- For MPI, `mpcc,mpCC,mpxlf` are the mpi wrappers.
- Suggested compiler flags:

  `-q64 -O3 -qhot -qarch=auto -qtune=auto`

  supplemented by `-qsmp=omp -O4` for OpenMP programs.
- On the link line we suggest using

  `-q64 -bdatapsize:64k -bstackpsize:64k`

  also supplemented by `-qsmp=omp -O4` for OpenMP programs.

Compiling on SciNet systems

TCS compilers and P7 compilers

- ssh to a devel node

  `ssh tcs01` *or* `ssh tcs02` , *or* `ssh p701`

- Use IBM compilers: `xlc,xlC,xlf` for C, C++, and Fortran (`module load vacpp xlf` for the latest version)
- For OpenMP, use `xlc_r,xlC_r,xlf_r`.
- For MPI, `mpcc,mpCC,mpxlf` are the mpi wrappers.
- Suggested compiler flags:

  `-q64 -O3 -qhot -qarch=auto -qtune=auto`

  supplemented by `-qsmp=omp -O4` for OpenMP programs.
- On the link line we suggest using

  `-q64 -bdatapsize:64k -bstackpsize:64k`

  also supplemented by `-qsmp=omp -O4` for OpenMP programs.

SciNet compute • calcul CANADA

Compiling on SciNet systems

TCS compilers and P7 compilers

- ssh to a devel node

  `ssh tcs01` *or* `ssh tcs02` , *or* `ssh p701`

- Use IBM compilers: `xlc,xlC,xlf` for C, C++, and Fortran (`module load vacpp xlf` for the latest version)
- For OpenMP, use `xlc_r,xlC_r,xlf_r`.
- For MPI, `mpcc,mpCC,mpxlf` are the mpi wrappers.
- Suggested compiler flags:

  `-q64 -O3 -qhot -qarch=auto -qtune=auto`

  supplemented by `-qsmp=omp -O4` for OpenMP programs.
- On the link line we suggest using

  `-q64 -bdatapsize:64k -bstackpsize:64k`

  also supplemented by `-qsmp=omp -O4` for OpenMP programs.

SciNet compute • calcul CANADA

Compiling on SciNet systems

ARC compilation

- From `login.scinet.utoronto.ca`, ssh to devel node

  ```
  ssh arc01
  ```

- The NVIDIA cuda compiler is available (3.2 and 4.0).
  You can `module load cuda/3.2` or `cuda/4.0`.
  The compiler is called `nvcc`.
- Optimize for the Tesla GPUs using the compiler flags

  ```
  -arch=sm_13 -O3
  ```

- As of version 3.0, OpenCL is included in the CUDA Toolkit so loading the CUDA module is all that is required.
- Debuggers: cuda-gdb or ddt (`module load ddt`)

Submitting jobs

SciNet=shared resource

- To run a job, you must submit to a batch queue.
- You submit jobs from a devel node in the form of a script
- Scheduling is by node!
- Best to run from the scratch directory (home=read-only)
- Copy essential results out after your runs have finished.

Limits

- Group based limits:
  possible for your colleagues to exhaust group limits
- Talk to us first to run massively parallel jobs ($>$ 2048 cores)
- While their resources last, jobs will run at a higher priority than others for groups with an allocation.

GPC queues

queue	time(hrs)	max jobs	max cores
batch	48	32/1000	256/8000 (512/16000 threads)
debug	2/0.5	1	16/64 (32/128 threads)
largemem	48	1	16 (32 threads)

- Submit to these queues with

```
qsub [options] <script>
```

- Common options:

  `-l`: specifies requested nodes and time, e.g.

```
-l nodes=1:ppn=8,walltime=1:00:00
```

```
-l nodes=2:ib:ppn=8,walltime=1:00:00
```

  `-q`: specifies the queue, e.g.

```
-q largemem
```

```
-q debug
```

  `-I` specifies that you want an interactive session.

- **Mandatory** number of nodes may be specified in script.

GPC queues

- GPC HyperThreading: Appears as if there are 16 processors rather than 8 per node. For OpenMP application, try setting `OMP_NUM_THREADS=16`. For MPI, try `-np 16`.
- *Always first test if this is beneficial and feasible!*
- Once the job is incorporated into the queue (this takes a minute), you can use: `showq` to show the queue, and job-specific commands such as `showstart, checkjob, canceljob`
- There is no separate queue for inifiniband nodes, just use `:ib`
- Always request ppn=8, even with hyperthreading.
- The largemem queue is exceptional, in that it provides access to two nodes (only) that have 16 processors and 128GB of ram.
- There is no queue for serial jobs, so if you have serial jobs, YOU will have to bunch together 8 of them to use the node's full power. GNU Parallel can help you with that.

SciNet compute • calcul CANADA

ARC queue

time(hrs)	max jobs	max cores
48	32	64cpu / 7168 gpu

- Submit to these queues with

```
qsub [options] <script>
```

- Common options:

  `-l`: specifies requested nodes and time, e.g.

```
-l nodes=1:gpus=2:ppn=8,walltime=1:00:00
```

  `-I` specifies that you want an interactive session.

- See submitted jobs with `qstat`.
- Limits on this system may change.

TCS/P7 queues

queue	time(hrs)	max jobs	max cores
verylong	48	2/25	64/800 (128/1600 threads)

Submitting is done with

```
llsubmit <script>
```

and `llq` shows the queue.

- The POWER processors have Simultaneous Multi Threading. Similar to HyperThreading.
- Once your job is in the queue, you can use `llq` to show the queue, and job-specific commands such as `llcancel, llhold, ...`
- *Do not run serial jobs on the TCS!*
- To make your jobs start sooner, reduce the `wall_clock_limit` to be closer to the estimated run time (perhaps adding about 10 % to be sure). Shorter jobs are scheduled sooner than longer ones.

Example (GPC)

```
#!/bin/bash
#PBS -l nodes=8:ppn=8,walltime=1:00:00
#PBS -N myInformativeJobName
cd $PBS_O_WORKDIR
#assume module load openmpi in .bashrc
mpirun -np 64 ./mycode > out
```

```
$ cd /scratch/$USER
$ qsub myjob.pbs
$ qstat
    Job id              Name             User Time Use S Queue
    ----------------- -------------- ---- -------- - --------
    2961983.gpc-sched myInformat...  rzon        0 Q batch
```

Example (GPC)

```
#!/bin/bash
#PBS -l nodes=8:ppn=8,walltime=1:00:00
#PBS -N myInformativeJobName
cd $PBS_O_WORKDIR
#assume module load openmpi in .bashrc
mpirun -np 64 ./mycode > out
```

```
$ cd /scratch/$USER
$ qsub myjob.pbs
$ qstat
    Job id              Name             User Time Use S Queue
    ----------------- -------------- ---- -------- - --------
    2961983.gpc-sched myInformat...  rzon        0 R batch_eth
```

Example (GPC)

```
#!/bin/bash
#PBS -l nodes=8:ppn=8,walltime=1:00:00
#PBS -N myInformativeJobName
cd $PBS_O_WORKDIR
#assume module load openmpi in .bashrc
mpirun -np 64 ./mycode > out
```

```
$ cd /scratch/$USER
$ qsub myjob.pbs
$ qstat
    Job id              Name             User Time Use S Queue
    ----------------- -------------- ---- -------- - --------
    2961983.gpc-sched myInformat...  rzon .......  C batch_eth
```

Example (GPC)

```
#!/bin/bash
#PBS -l nodes=8:ppn=8,walltime=1:00:00
#PBS -N myInformativeJobName
cd $PBS_O_WORKDIR
#assume module load openmpi in .bashrc
mpirun -np 64 ./mycode > out
```

```
$ cd /scratch/$USER
$ qsub myjob.pbs
$ qstat
    Job id              Name             User Time Use S Queue
    ----------------- -------------- ---- -------- - --------
    2961983.gpc-sched myInformat...  rzon .......  C batch_eth

$ ls
    myInformativeJobName.e2961983    mycode      out
    myInformativeJobName.o2961983
```

Part IV

DATA MANAGEMENT

File system

SciNet $\neq$ 4000 $\times$ your pc

- Compute nodes do not contain hard drives!
- The available disk space, /home and /scratch, all part of the GPFS file system which runs over the network.
- GPFS is a high-performance file system which provides rapid reads and writes to large data sets in parallel from many nodes.
- It performs quite poorly at accessing data sets which consist of many, small files.
- Don't keep many small files on the system.
  They waste space, and are slower to access, read and write.

I/O strategies

- Do not read and write lots of small amounts of data to disk.
  Reading data in from one 4MB file can be enormously faster than from 100 40KB files.
- Write your data out in binary. Faster and takes less space.
- Each process writing to a file of its own is not scalable.
  A directory gets locked by the first process accessing it, so the other processes have to wait for it.
- Consider using MPI-IO (part of the MPI-2 standard), (parallel) NetCDF or HDF5.
- If you must read and write a lot to disk, use ramdisk if possible.
  The ramdisk can be accessed using /dev/shm/ and is currently set to 8GB max.
- Copy back from ramdisk at end of run.

SciNet compute • calcul CANADA

Moving large data

Moving less than 10GB through the login nodes

- Only login nodes visible from outside SciNet (1Gb/s link).
- Use scp or rsync.
- but datamover1 node is faster.

Moving more than 10GB through the datamover1 node

- Should be done from the datamover1 node (10Gb/s link).
- From any SciNet node, ssh to datamover1.
- Transfers must originate from datamover1.
  Cannot copy files from the outside world to datamover1.
- Your machine must be reachable from the outside.

Moving data to HPSS

- HPSS is a tape-based storage solution.
- Available to groups with a special allocation.
- SNUG TechTalk on Sept. 14 on HPSS.

SciNet compute • calcul CANADA

Final tips

- Test your job's requirements and scaling behaviour.
  Start runs on a small scale and work your way up to larger scales.
- Accurately specify the walltime when you submit a job.
- Avoid reading and writing lots of small amounts of data to disk.
- Do not create lots of files.
- Do not submit single serial jobs.
- Do not keep lots of files in your directory (use `tar`).
- Read the SciNet User Guide

SciNet compute • calcul CANADA

Useful web sites

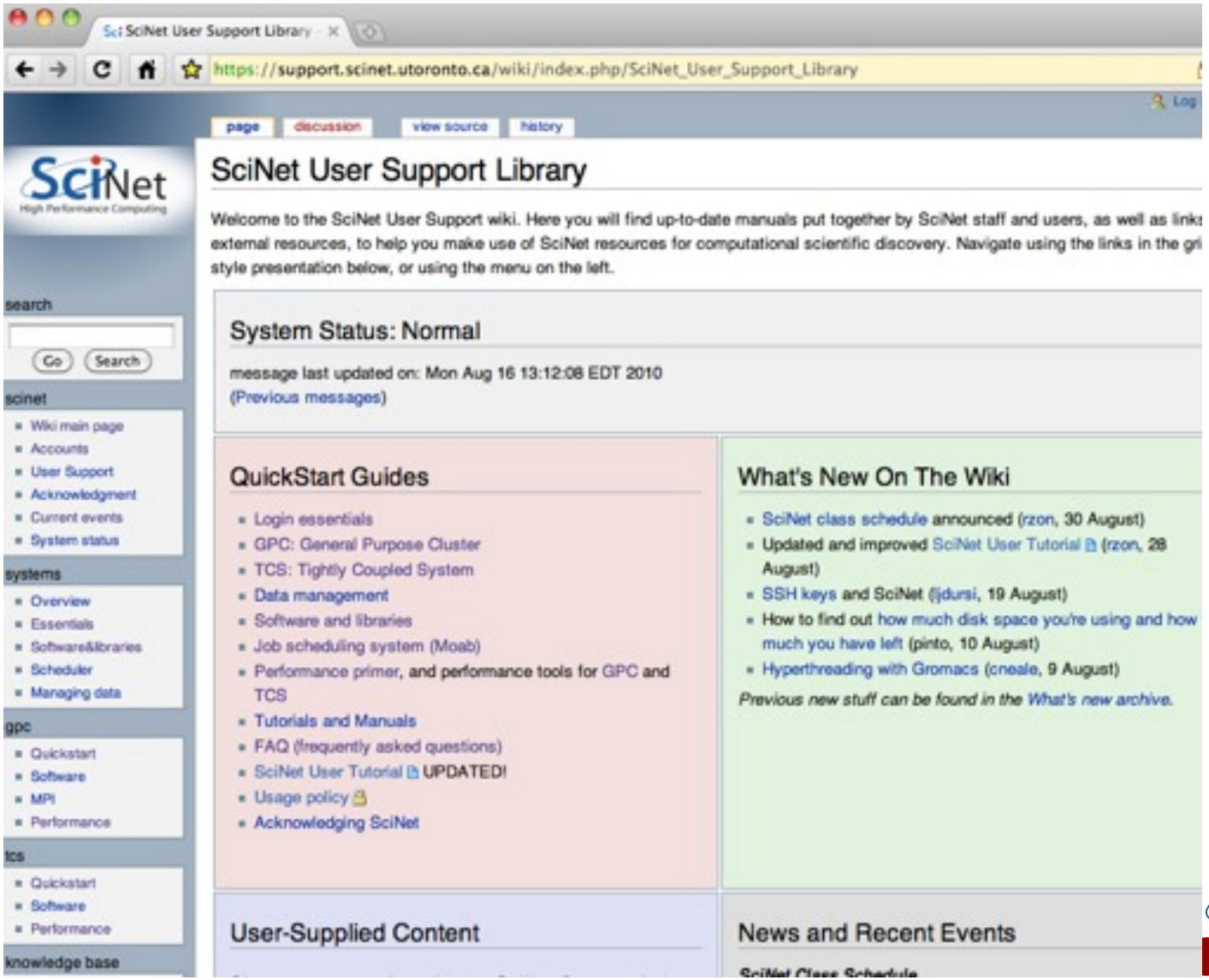
Portal: https://portal.scinet.utoronto.ca

SciNet usage reports
Change password, default allocation, maillist subscriptions

Exception: gpgpu mail list:
https://support.scinet.utoronto.ca/mailman/listinfo/scinet-gpgpu

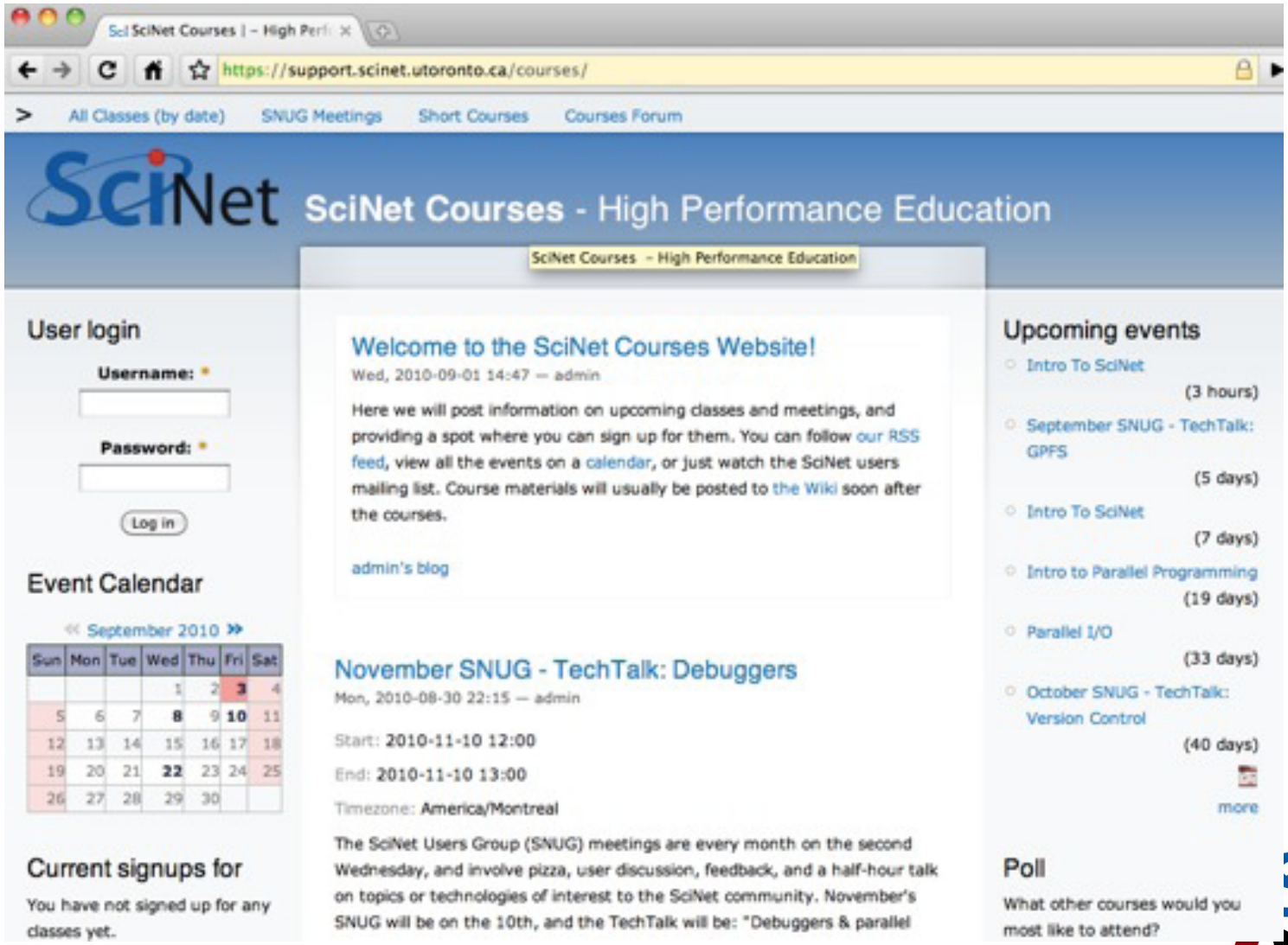
Useful web sites

Wiki: http://wiki.scinet.utoronto.ca



Useful web sites

Courses: https://support.scinet.utoronto.ca/courses



Links

Wiki: http://wiki.scinet.utoronto.ca

Courses: https://support.scinet.utoronto.ca/courses

Portal: https://portal.scinet.utoronto.ca

Technical support: support@scinet.utoronto.ca