Introduction to SciNet

DON'T PANIC

SciNet

Outline

1. About SciNet
   - SciNet is . . .
   - How to get access
   - Useful sites
   - SciNet systems
2. Using SciNet
   - Software/Libraries
   - Compilers
   - Job submission
   - Final tips

PART 1 — 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 is ...

. . . home to the first and third largest supercomputers in 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
work directly with you t
our resources to produ
science.

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

– 7 people that make sure
verything 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 produ od 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

SciNet is ...

. . . 4 technical analysts w t make sure
work directly with you to s smoothly.
our resources to produ to
science. nen

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

ong

ng 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 produ od 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 produ 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 produ od 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

SciNet is ...

. . . 4 technical analysts who can work directly with you to use our resources to produ od science.

- Jonathan D
- Scott North
- Ramses va
- Daniel Gru

+ 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 produ od 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

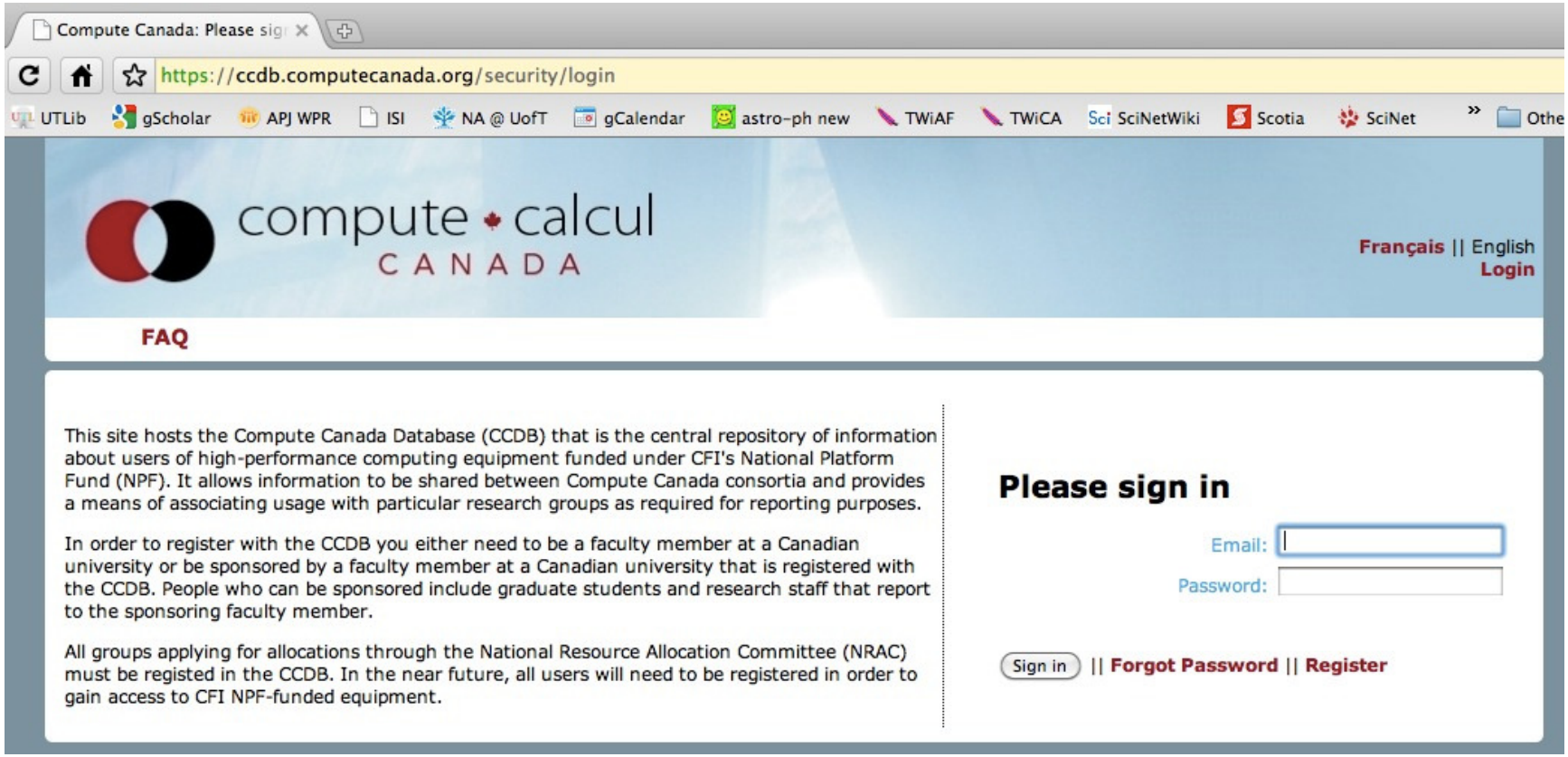
\+ Project coordinator Jillian Dempsey

How to get access

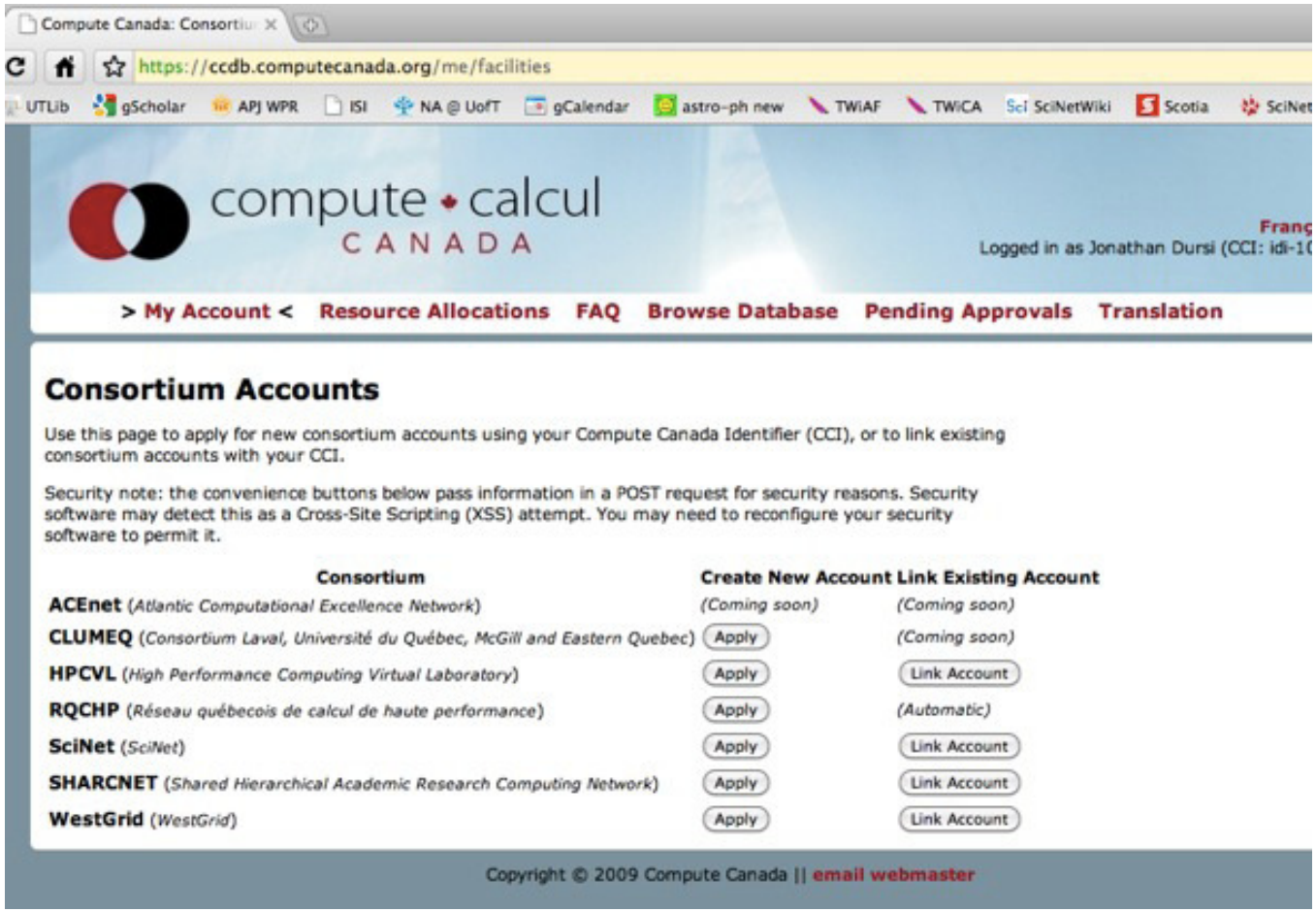
Any qualified researcher at a Canadian university can get a SciNet account through this two-step process:

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

How to get access



How to get access



How to get access

Getting more resources on SciNet

- Users who will be needing more than the default amount of resources must have their PI apply for it through the competitively awarded.
- Resource competition occurs in the fall of each year.
- Without such an allocation, a user may still use up to 32 GPC nodes at a times at low priority.

Now what?

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

SciNet usage reports
Change password, default allocation, maillist subscriptions

Now what?

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

SciNet usage reports
Change password, default allocation, maillist subscriptions

Now what?

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

SciNet usage reports
Change password, default allocation, maillist subscriptions

Wiki: https://support.scinet.utoronto.ca/wiki

More information can be found at the wiki.

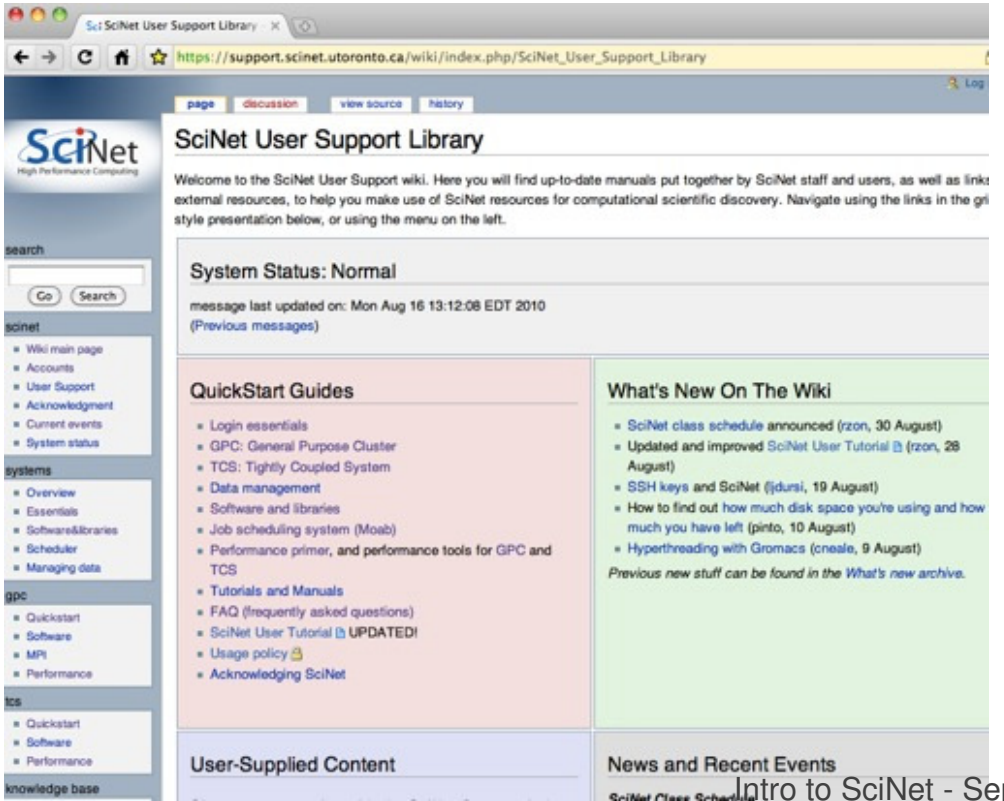
Now what?

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

SciNet usage reports
Change password, default allocation, maillist subscriptions

Wiki: https://support.scinet.utoronto.ca/wiki

More information can be found at the wiki.



Now what?

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

SciNet usage reports
Change password, default allocation, maillist subscriptions

Wiki: https://support.scinet.utoronto.ca/wiki

More information can be found at the wiki.

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

To learn about SciNet courses and sign up.

Now what?

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

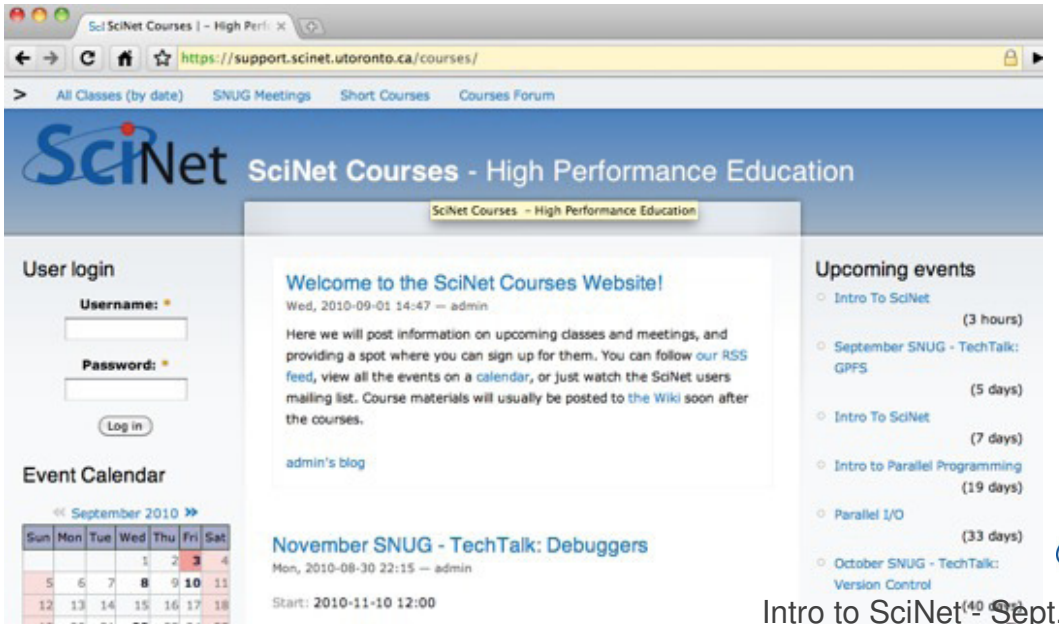
SciNet usage reports
Change password, default allocation, maillist subscriptions

Wiki: https://support.scinet.utoronto.ca/wiki

More information can be found at the wiki.

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

To learn about SciNet courses and sign up.



Now what?

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

SciNet usage reports
Change password, default allocation, maillist subscriptions

Wiki: https://support.scinet.utoronto.ca/wiki

More information can be found at the wiki.

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

To learn about SciNet courses and sign up.

Technical support: support@scinet.utoronto.ca

For further questions, mail our system administrators and technical analysts.

Now what?

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

SciNet usage reports
Change password, default allocation, maillist subscriptions

Wiki: https://support.scinet.utoronto.ca/wiki

More information can be found at the wiki.

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

To learn about SciNet courses and sign up.

Technical support: support@scinet.utoronto.ca

For further questions, mail our system administrators and technical analysts.

Login: login.scinet.utoronto.ca

Access SciNet systems: `ssh` to login (not part of clusters)

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

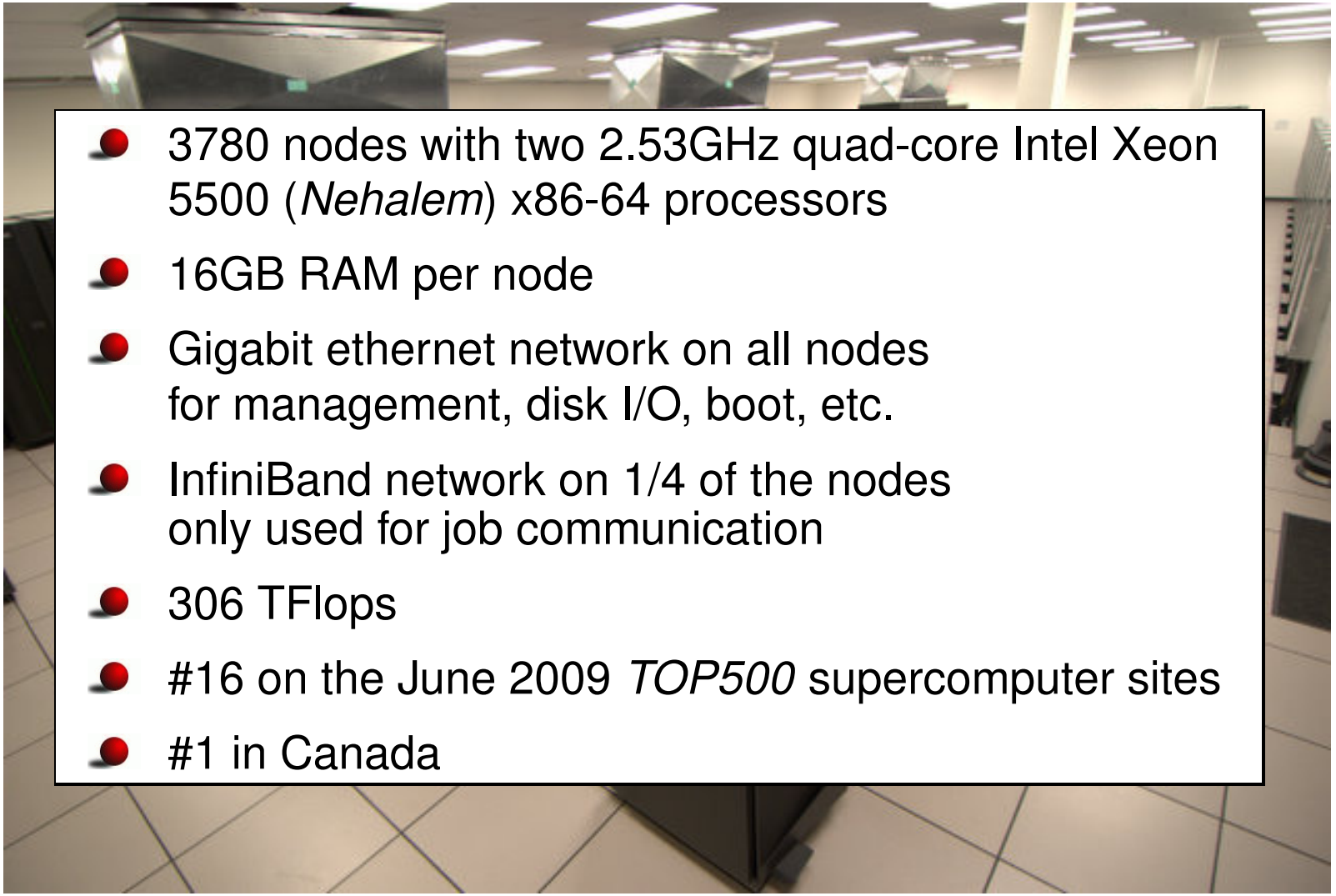
Overview of the SciNet Systems

Systems

General Purpose Cluster (GPC)

Systems

General Purpose Cluster (GPC)



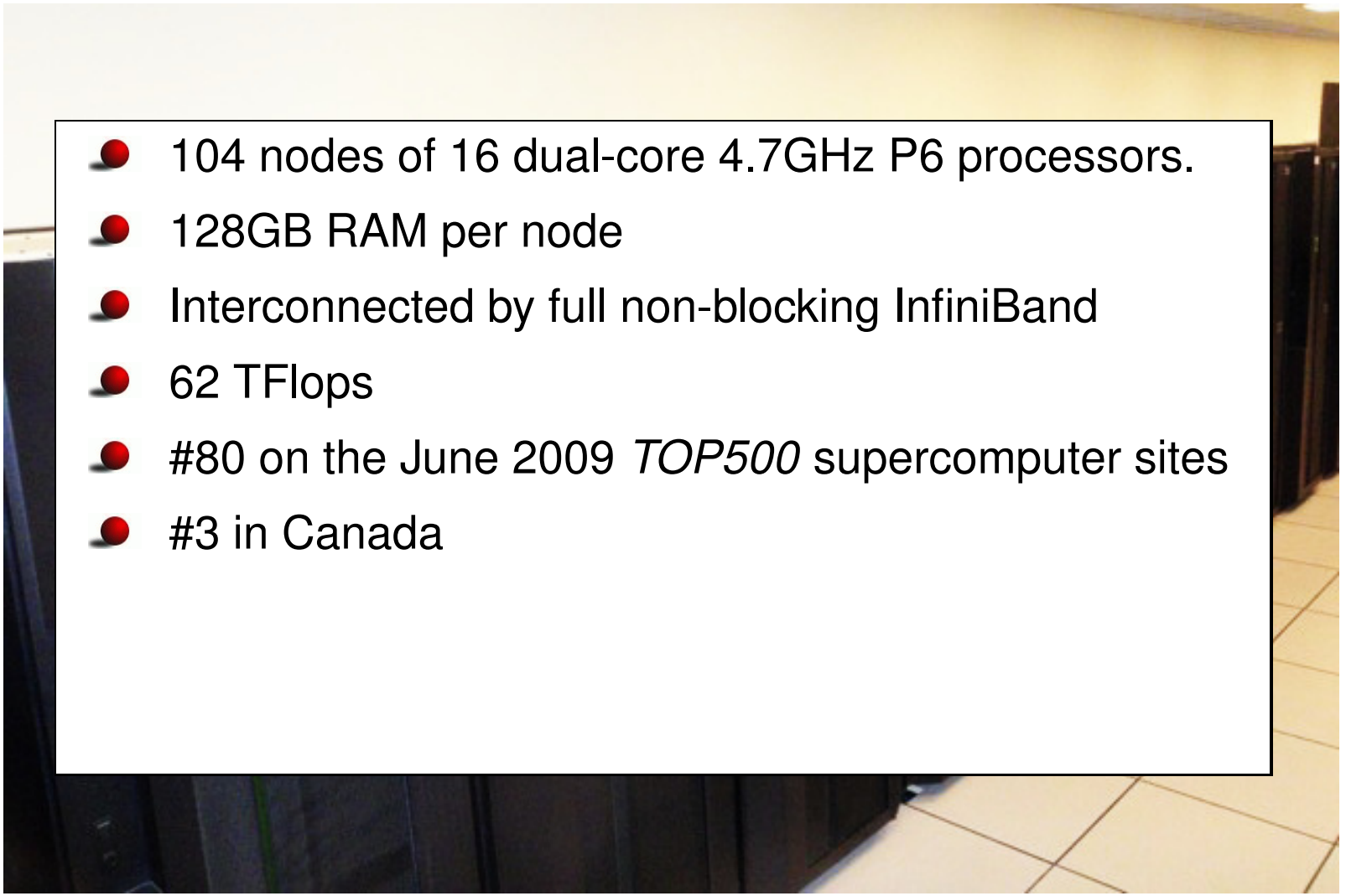
- 3780 nodes with two 2.53GHz quad-core Intel Xeon 5500 (*Nehalem*) x86-64 processors
- 16GB RAM per node
- Gigabit ethernet network on all nodes for management, disk I/O, boot, etc.
- InfiniBand network on 1/4 of the nodes only used for job communication
- 306 TFlops
- #16 on the June 2009 *TOP500*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

Systems

Tightly Coupled System (TCS)

Systems

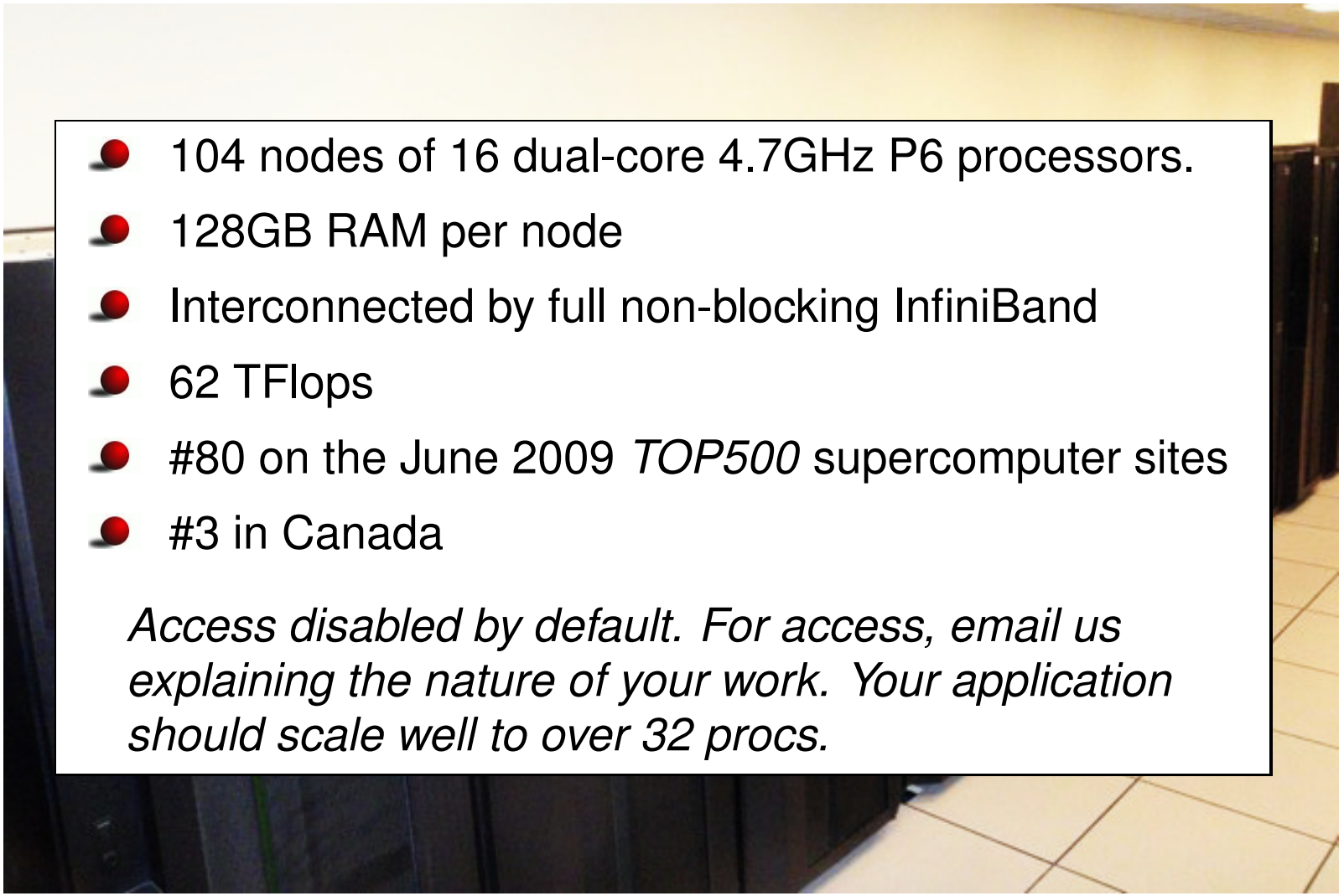
Tightly Coupled System (TCS)



- 104 nodes of 16 dual-core 4.7GHz P6 processors.
- 128GB RAM per node
- Interconnected by full non-blocking InfiniBand
- 62 TFlops
- #80 on the June 2009 *TOP500* supercomputer sites
- #3 in Canada

Systems

Tightly Coupled System (TCS)



- 104 nodes of 16 dual-core 4.7GHz P6 processors.
- 128GB RAM per node
- Interconnected by full non-blocking InfiniBand
- 62 TFlops
- #80 on the June 2009 *TOP500* supercomputer sites
- #3 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.

Systems

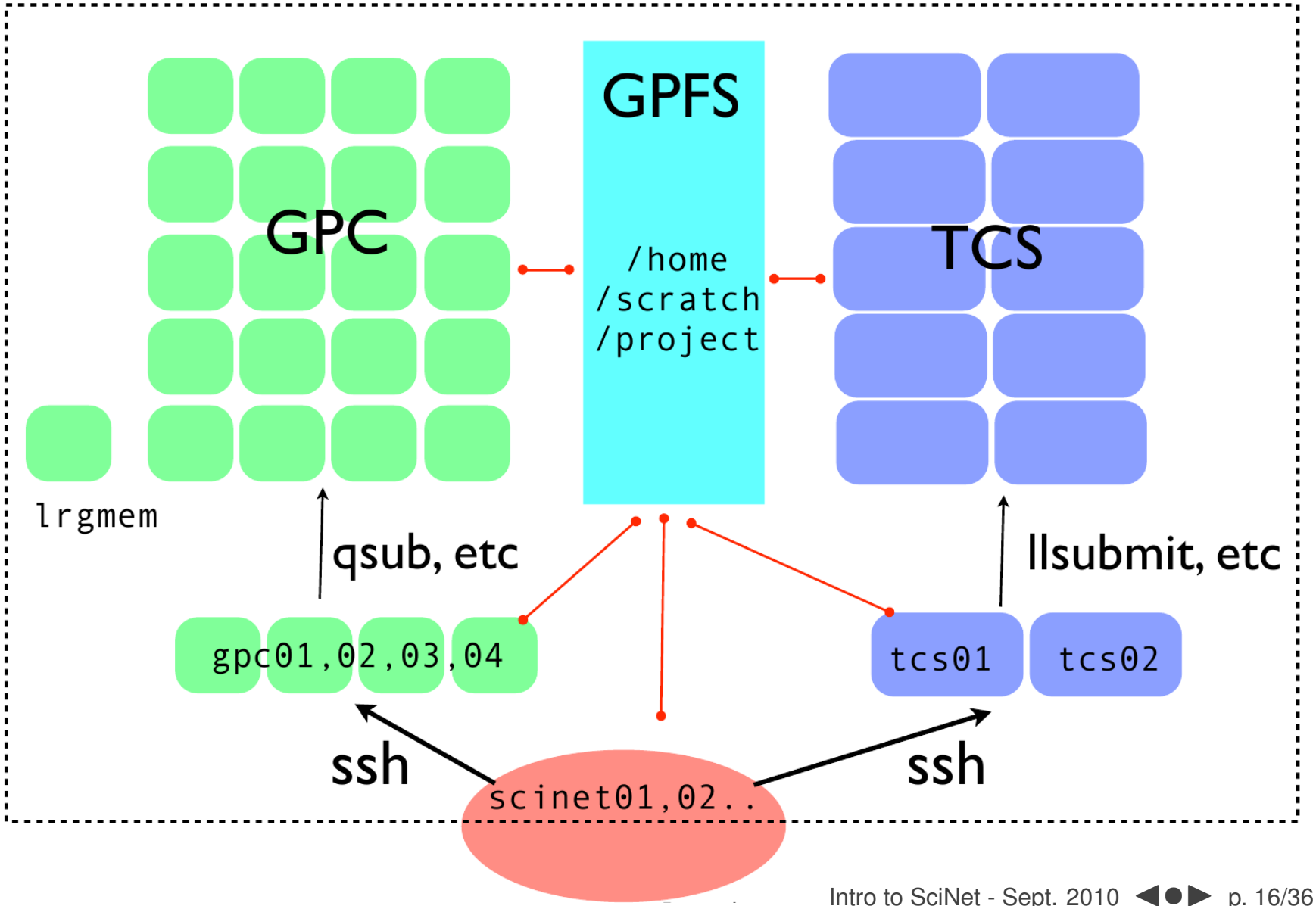
Disk space

- 1790 1TB disk drives, for a total of 1.4 PB of storage
- Two DCS9900 couplets, each delivering 4-5GB/s read/write access to the drives
- Single *GPFS* filesystem on both TCS and GPC
- I/O goes over Gb ethernet network on the GPC, and over the infiniband network on the TCS

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

SciNet

Systems


GPFS
GPC
/home
/scratch
/project
TCS
lrgmem
qsub, etc
llsubmit, etc
gpc01,02,03,04
tcs01
tcs02
ssh
ssh
scinet01,02..

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.

Systems

File system

- 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), 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.

PART 2 — USING SCINET

Software and Libraries

Modules

- Once you log into devel nodes, what's already installed?
- Other than essentials, all software installed as modules.
- modules set environment variables (LD_LIBRARY_PATH, PATH, ...) to include the appropriate package.
- Allows multiple versions of package to be available.
- More on *Software and Libraries* page of wiki.

```
gpc-f103n084-$ module avail

------ /scinet/gpc/Modules/version_inde
3.2.6

------ /scinet/gpc/Modules/3.2.6/module
dot                 modules           use.c
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
...
```

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.
- Short name prefered (`intel` not `intel/intel-v11.1.056`)
- To compile code that uses that package add

  `-I${SCINET_[shortmodulename]_INC}`
- To link, add

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

SciNet

Software and Libraries

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/gcc-4.4.0'
python/2.6.2(11):ERROR:102:  Tcl
command execution failed:  prereq gcc/
gcc-4.4.0
gpc-f103n084-$ module load gcc python

gpc-f103n084-$
```

Software and Libraries

Commercial Software?

- We have an extremely large and broad user base
- We cannot buy everyone's favourite commercial software package.
- Only commercial software we have installed is software that can benefit everyone:
  - GPC: intel compilers, MKL (both in module intel)
  - TCS: ibm compilers, ESSL
- No matlab, gaussian, idl, . . .
- Can work with you to get commercial software that you have license for installed.

Compiling

- **WARNING:**
  The login machines are not the same architecture as either the GPC or TCS nodes, so you should not compile programs on the login machines.
- Compile on devel nodes:

  ```
  gpc01, gpc02, gpc03, gpc04
  tcs01, tcs02
  ```

  Also good for short, small scale tests.
- Test your job's requirements and scaling behaviour before submitting a large scale computation.
- Devel nodes are used by *everyone* who needs to use the SciNet systems, so be considerate.

Compiling

GPC compilation

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

  ```
  ssh gpc04
  ```

- We recommend Intel compilers, which are `icc`, `icpc`, and `ifort` for C, C++, and Fortran, from the module `intel`.
- Optimize your code for the GPC machine using of at least the following compiler flags

  ```
  -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)

Compiling

TCS compilers

- ssh to tcs devel nodes
  `ssh tcs01` or `ssh tcs02`
- Use the IBM compilers on the TCS devel nodes. These are `xlc,xlC,xlf` for C, C++, and Fortran.
- For OpenMP code, use `xlc_r,xlC_r,xlf_r`.
- For MPI code, `mpcc,mpCC,mpxlf` are the mpi wrappers.

- We strongly suggest the compiler flags

  ```
  -q64 -O3 -qhot -qarch=pwr6 -qtune=pwr6
  ```

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

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

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

SciNet

Submitting jobs

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

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

Submitting jobs

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)

You 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=1: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; no script needed

Mandatory number of nodes may be specified in script.

Submitting jobs

- 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, 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. You request these through the option `: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.

SciNet

Submitting jobs

TCS queue

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 POWER6 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.

Submitting jobs

Example for 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
```

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

Submitting jobs

Example for 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
```

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

Submitting jobs

Example for 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
```

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

Submitting jobs

Example for 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
```

```
$ 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
```

Final tips

- Use the right compilers and compile with optimization.
- 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 wiki at support.scinet.utoronto.ca/wiki
- Take SciNet classes: support.scinet.utoronto.ca/courses
- Email to support@scinet.utoronto.ca with any SciNet questions or problems.

SciNet