

# Introduction to SciNet

SciNet HPC Consortium  
Compute Canada

September 12, 2012

**Don't Panic**

# Part I

## ABOUT SCINET

# SciNet is ...

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



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



# SciNet is ...

... where you go for courses on a wide range of computational topics, e.g.

- Intro to SciNet
- Intro to Linux Shell
- Parallel programming courses
- Scientific Computing Course (for credit for physics/astro grads)
- Ontario HPC Summerschool
- ...





# SciNet is ...

- ... where you meet other users at monthly SciNet User Group meetings.  
Every 2nd Wednesday of the Month.  
1 or more TechTalks ([wiki.scinethpc.ca/wiki/index.php/SNUG\\_TechTalks](http://wiki.scinethpc.ca/wiki/index.php/SNUG_TechTalks))  
And pizza!
- ... where you go for SciNet Developer Seminars (starts this fall)

# SciNet people

There are 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

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

+ 7 people that make sure everything runs smoothly.

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

## Part II

### SCINET SYSTEMS

# Compute Resources at SciNet

## General Purpose Cluster (GPC)

- 3864 nodes with 8 Intel x86-64 cores @ 2.53Hz
- 30,912 cores
- 328 TFlops
- 16 GB RAM per node ( $\sim 14$ GB for user jobs)
- 16 threads per node
- Operating system: CentOS 6
- Interconnect: InfiniBand
- #16 on the June 2009 *TOP500* (Now at #66)
- #1 in Canada

[http://wiki.scinethpc.ca/wiki/index.php/GPC\\_Quickstart](http://wiki.scinethpc.ca/wiki/index.php/GPC_Quickstart)

# Other Compute Resources at SciNet

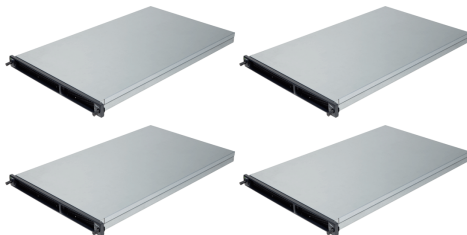
Tightly Coupled System (TCS)



Power 7 Linux Cluster (P7)



GPU Devel Nodes (ARC)



Soon: Blue Gene/Q



# Storage Resources at SciNet



## Disk space

- 1.4 PB of storage in 1790 drives
- Two controllers each delivering 4-5 GB/s (r/w)
- *Shared* file system GPFS on all systems
- Your files go in /home/g/group/user and /scratch/g/group/user.

## Storage space

- HPSS: Tape-backed storage expansion solution.

*Access by allocation*

<http://wiki.scinethpc.ca/wiki/index.php/HPSS>

# Storage Limits at SciNet

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

[http://wiki.scinethpc.ca/wiki/index.php/Data\\_Management](http://wiki.scinethpc.ca/wiki/index.php/Data_Management)

# Part III

## USING SCINET

# Get on the system

## 1. Access systems: [login.scinet.utoronto.ca](http://login.scinet.utoronto.ca)

*First ssh to login (not part of clusters):*

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

*The login nodes are gateways:*

- *only to be used for small data transfer*
- *and to proceed logging into one of the devel nodes.*

# Get on the system

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

*GPC:* gpc01, gpc02, gpc03, gpc04

*These are aliases for longer node names E.g.*

```
ssh gpc03 -X
```

*gets you to the gpc development node named gpc-f103n084.*

# Software and Libraries (1)

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

- Other than essentials, all software installed using module commands.
- sets 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/Modules6/Modules/versions-----  
3.2.8  3.2.9  
-----/scinet/gpc/Modules6/Modules/3.2.9/modulefiles-  
dot                modules                use.own  
module-cvs         use.deprecated  
module-info        use.experimental  
-----/scinet/gpc/Modules6/Modules/modulefiles-  
ImageMagick/6.6.7(default)  
R/2.13.1(default)  
R/2.14.1  
ROOT/5.30.03(default)  
ROOT/5.32.00  
Xlibraries/X11-64  
abyss/1.3.2  
adios/131-openmpi-gcc(default)  
amber/10.0.30  
antlr/2.7.7  
autoconf/2.68  
automake/1.11.2  
blast/2.2.23+  
...
```

[http://wiki.scinethpc.ca/wiki/index.php/Software\\_and\\_Libraries](http://wiki.scinethpc.ca/wiki/index.php/Software_and_Libraries)

# Software and Libraries (2)

|                                              |                                  |
|----------------------------------------------|----------------------------------|
| <code>module load &lt;module-name&gt;</code> | use particular software          |
| <code>module purge</code>                    | remove currently loaded modules  |
| <code>module avail</code>                    | list available software packages |
| <code>module list</code>                     | list loaded modules              |
| <code>module help &lt;module-name&gt;</code> | 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` → `intel/12.1.3`)

# Software and Libraries (3)

## Dependencies

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

### Example

```
gpc-f103n084$ module purge
```

```
gpc-f103n084$ module load python
```

```
python/2.7.2(11):ERROR:151: Module 'python/2.7.2' depends on one of  
the module(s) 'gcc/4.7.0 gcc/4.6.1 gcc/4.4.6'
```

```
python/2.6.2(11):ERROR:102: Tcl command execution failed:  prereq gcc
```

```
gpc-f103n084$ module load gcc python
```

```
python/2.7.2(11):ERROR:151: Module 'python/2.7.2' depends on one of  
the module(s) 'intel/12.1.5 intel/12.1.3 intel/12.1.2 intel/12.1'
```

```
python/2.6.2(11):ERROR:102: Tcl command execution failed:  prereq intel
```

```
gpc-f103n084$ module load gcc intel python
```

```
gpc-f103n084$ module list
```

Currently Loaded Modulefiles:

1) gcc/4.6.1

2) intel/12.1.3

3) python/2.7.2

# Software and Libraries (4)

## Commercial Software?

- SciNet has an extremely large and broad user base ( $\sim 1000$  users)
- Only commercial software that can benefit everyone:  
Compilers, math libraries and debugger.
- No Matlab, Gaussian, IDL, ...  
(but Octave)
- Can help you to install software for which you have a license.

# Compiling on SciNet (1): GPC

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

```
ssh gpc04 [-X]
```

or

```
gpcdev [-X]
```

- We recommend Intel compilers, which are

```
icc, icpc, ifort
```

for C, C++, and Fortran, respectively (from the module `intel`)

- Optimize 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 OpenMPI, in module `openmpi` (v1.4.4)
  - 2 Intel MPI, in module `intelmpl` (v4.0.3)

# Compiling on SciNet (2): library modules

- To compile code that uses that a library from a module, add

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

- To link, add

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

# Compiling on SciNet (3): library modules

## Example

```
scinet04$ ssh gpc03
gpc-f103n084$ module list
No Modulefiles Currently Loaded.
gpc-f103n084$ pwd
/home/s/scinet/rzon
gpc-f103n084$ ls
main.c morecode.c
gpc-f103n084$ module load intel gsl
gpc-f103n084$ module list
Currently Loaded Modulefiles:
1) intel/12.1.3 2) gsl/1.13-intel
gpc-f103n084$ icc -O3 -xHost -o main.o main.c
gpc-f103n084$ icc -O3 -xHost -I${SCINET_GSL_INC} -o morecode.o morecode.c
gpc-f103n084$ icc -o main morecode.o main.o -L${SCINET_GSL_LIB} -lgsl
-lgslcblas
gpc-f103n084$ ./main
```

# Submitting jobs

- 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. You need to use all 8 cores on the node!
- Best to run from the scratch directory (**home=read-only**)
- Copy essential results out after runs have finished.

# Submitting jobs

## 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    | min.time | max.time | max jobs | max cores |
|----------|----------|----------|----------|-----------|
| batch    | 15m      | 48h      | 32/1000  | 256/8000  |
| debug    |          | 2h/30m   | 1        | 16/64     |
| largemem | 15m      | 48h      | 1        | 16        |

# GPC queues

- Submit to these queues from a GPC devel node with

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

- Common options (usually in script):

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

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

-q: specifies the queue, e.g.

```
-q batch
```

```
-q debug
```

```
-q largemem
```

-I specifies that you want an interactive session.

-X specifies that you want X forwarded.

# GPC job script example

```
#!/bin/bash

#PBS -l nodes=1:ppn=8

#PBS -l walltime=1:00:00

#PBS -N simple-omp-job

cd $PBS_O_WORKDIR

export OMP_NUM_THREADS=8

./openmp_example > output
```

```
$ qsub scriptname.pbs
```

# GPC queues

- HyperThreading: Appears as if there are 16 processors rather than 8 per node. For OpenMP applications this is the default unless `OMP_NUM_THREADS` is set. For MPI, try `-np 16`.  
*Always request ppn=8, even with hyperthreading.*
- 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`
- 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.

# Example 1 (GPC)

```
gpc-f101n084-$ module load intel openmpi
gpc-f101n084-$ mpif90 -O3 -xhost mycode.f90 -o mycode
gpc-f101n084-$ mkdir $SCRATCH/example1
gpc-f101n084-$ cp mycode $SCRATCH/example1
gpc-f101n084-$ cd $SCRATCH/example1
gpc-f101n084-$ cat > myjob.pbs
#!/bin/bash
#PBS -l nodes=8:ppn=8,walltime=1:00:00
#PBS -N JobName
cd $PBS_O_WORKDIR
module load intel openmpi
mpirun -np 64 ./mycode > out
gpc-f101n084-$ qsub myjob.pbs
2961983.gpc-sched
gpc-f101n084-$ qstat (or checkjob 2961983, or showq -u $USER)
  Job id              Name              User Time Use S Queue
  -----
  2961983.gpc-sched JobName              rzon              0 Q batch
gpc-f101n084-$ ls
JobName.e2961983  JobName.o2961983  mycode  myjob.pbs
out
```

# Example 2 (GPC)

```
gpc-f101n084-$ module load intel
gpc-f101n084-$ ifort -O3 -xhost mycode.f90 -o mycode
gpc-f101n084-$ mkdir $SCRATCH/example2
gpc-f101n084-$ cp mycode $SCRATCH/example2
gpc-f101n084-$ cd $SCRATCH/example2
gpc-f101n084-$ cat > joblist.txt
    mkdir run1; cd run1; ../mycode 1 > out
    mkdir run2; cd run2; ../mycode 2 > out
    mkdir run3; cd run3; ../mycode 3 > out
    ...
    mkdir run64; cd run64; ../mycode 64 > out
gpc-f101n084-$ cat > myjob.pbs
    #!/bin/bash
    #PBS -l nodes=1:ppn=8,walltime=24:00:00
    #PBS -N ASerialJob
    cd $PBS_O_WORKDIR
    module load intel gnu-parallel
    parallel -j 8 < joblist.txt
gpc-f101n084-$ qsub myjob.pbs
    2961985.gpc-sched
gpc-f101n084-$ ls
    ASerialJob.e2961985  ASerialJob.o2961985  joblist.txt  mycode
    myjob.pbs            run1/                run2/        run3/
```

# Part IV

## DATA MANAGEMENT

# File system

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

- Each compute nodes has 0 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.
- Performs poorly 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 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.
- 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 11GB max.
- Copy back from ramdisk at end of run.

# 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 > 5TB.

# 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  
[http://wiki.scinethpc.ca/wiki/images/5/54/SciNet\\_Tutorial.pdf](http://wiki.scinethpc.ca/wiki/images/5/54/SciNet_Tutorial.pdf)

# Useful web sites

[www.scinethpc.ca](http://www.scinethpc.ca)

SciNet | Science At Scale - Mozilla Firefox

www.scinethpc.ca

# SciNet

Home SciNet for Me In The News SciNet Blogs About SciNet Events Other SciNet Links Search ...

## SciNet and the Discovery of the Higgs Boson

“SciNet is absolutely central to make anything out of what happens,” Teuscher [a University of Toronto ATLAS Researcher] said in this [Toronto Star](#) article. Want to learn more about computation and the Higgs? This PC Advisor article has a very good overview of the massive data challenges that the worlds largest scientific experiment faces, and [...]

SciNet is Canada's largest supercomputer centre, providing Canadian researchers with computational resources and expertise necessary to perform their research on scales not previously possible in Canada. SciNet powers work from the biomedical sciences and aerospace engineering to astrophysics and climate science. SciNet is part of Compute/Calcul Canada, a national infrastructure for supercomputing-powered innovation composed of seven consortia, and is funded by the Canada Foundation for Innovation, NSERC, the Government of Ontario, and the University of Toronto.

Find out more [about us](#), how to [contact us](#), or even how to start using SciNet resources.

If you're [a researcher](#), [an educator](#), or [work in industry](#), check out what SciNet can do for you!

SciNet is a member of Compute/Calcul Canada

 **compute • calcul**  
CANADA

 **ciNet**  
compute • calcul  
CANADA

# Useful web sites: Wiki

wiki.scinethpc.ca

SciNetWiki - Mozilla Firefox

wiki.scinethpc.ca/wiki/index.php/SciNet\_User\_Support\_Library

142.150.198.54 Talk for this IP address Log in

SciNet

compute + calcul CANADA

Other SciNet Websites

SciNet website  
SciNet courses  
SciNet portal

Systems

GPC cluster  
TCS cluster  
GPU cluster  
P7 cluster  
HPSS storage

Knowledge Base

SciNet essentials  
Tutorials/Manuals  
FAQ  
Software  
Moab scheduler  
Data management  
GPC MPI versions

Wiki

Recent changes  
Random page  
Wiki help

Toolbox

Page Discussion Read View source View history Go Search

## SciNet User Support Library

System Status: **UP**

Tue Jun 26 18:49:54 EDT 2012 ([Previous messages](#))

### QuickStart Guides

- [SciNet User Tutorial](#)
- [Tutorials and Manuals](#)
- [Essentials](#)
- [GPC: General Purpose Cluster](#)
- [TCS: Tightly Coupled System](#)
- [GPU nodes of the Accelerator Research Cluster](#)
- [P7: Power 7 Linux cluster](#)
- [Software and libraries](#)
- [Data management](#)
- [FAQ \(frequently asked questions\)](#)
- [Acknowledging SciNet](#)

### What's New On The Wiki

- [GPU Cluster](#) now integrated into GPC scheduler (Jul 5)
- [PGI compilers](#) supporting OpenACC and Cuda Fortran installed on the [GPU Cluster](#) (Jul 4)
- [PGI compiler manuals](#) added to the [Tutorials and Manuals](#) page (Jul 4)
- [Ontario Summerschool on High Performance Computing Central](#) slides and code (Jun 29)
- [TechTalk Remote Development](#) slides (Jun 13)

Previous new stuff can be found in the [What's new archive](#).

### User-Supplied Content

**Share your expertise with the SciNet Community!**

- [Running specific codes on SciNet machines](#)
- [Tips for performance or convenience on SciNet machines](#)
- [Running serial batch jobs \(including GNU parallel\)](#)
- [Using the ramdisk on SciNet's GPC](#)
- [User space perl modules](#)

### News and Recent Events

- Jun 7, 2012: SciNet Shutdown, at least till 10PM.
- Jun 13, 2012: [June SNUG](#) 📢 TechTalk by Ramses van Zon: [Remote Development on SciNet](#)
- Jun 25-28, 2012: [Ontario Summerschool for High Performance Computing - Central](#), hosted by SciNet 📢
- Jul 9, 2012, [Intro to SciNet](#) 📢

# Useful web sites: Courses

<https://support.scinet.utoronto.ca/courses>

**SciNet Courses - High Performance Education**

Navigation: All Classes (by date) | SNUG Meetings | Short Courses | Past Courses | Courses Forum

**Intro to SciNet**  
Short Course  
Monday, July 9, 2012 - 12:00 — Monday, July 9, 2012 - 13:30

A class of approximately 90 minutes where you will learn how to use the systems. Experienced users may still pick up some valuable pointers during these sessions.

You may find the SciNet User Tutorial useful: [http://wiki.scinethpc.ca/wiki/images/5/54/SciNet\\_Tutorial.pdf](http://wiki.scinethpc.ca/wiki/images/5/54/SciNet_Tutorial.pdf)

Location: SciNet offices at 256 McCaul Street, 2nd Floor.

Sign up here: [signup form](#).

**September SNUG**  
SNUG meeting  
Wednesday, September 12, 2012 - 12:00 — Wednesday, September 12, 2012 - 13:00

The SciNet Users Group (SNUG) meetings are every month on the second Wednesday, and involve pizza, user discussion, feedback, and a half-hour talk on topics or technologies of interest to the SciNet community.

Location: SciNet offices at 256 McCaul Street, 2nd Floor.

**Upcoming events**

- [Intro to SciNet](#) (3 days)
- [September SNUG](#) (68 days)
- [October SNUG](#) (96 days)
- [November SNUG](#) (131 days)
- [December SNUG](#) (159 days)

[more](#)

**Poll**

What time of year would be best for our week-long Practical Parallel Programming intensive?  
Summer (As currently)

73%

Fall

**Event Calendar**  
« July 2012 »

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
| 1   | 2   | 3   | 4   | 5   | 6   | 7   |
| 8   | 9   | 10  | 11  | 12  | 13  | 14  |
| 15  | 16  | 17  | 18  | 19  | 20  | 21  |
| 22  | 23  | 24  | 25  | 26  | 27  | 28  |
| 29  | 30  | 31  |     |     |     |     |

**Current signups for rzon**

You have not signed up for any classes yet.

**Other Resources**

# Useful web sites: Portal

<https://portal.scinet.utoronto.ca>

*SciNet usage reports*

*Change password, default allocation, maillist subscriptions*

