Using the IBM Opteron 1350 at OSC

October 19-20, 2010

Table of Contents

- Hardware Overview
- The Linux Operating System
- User Environment and Storage

Hardware Overview

- Hardware introduction
- Login node configuration
- Compute node configuration
- External network connectivity

Hardware introduction

- Old configuration
  - 877 System x3455 compute nodes
  - 88 System x3755 compute nodes
  - 1 e1350 QS20 Blade Center
    - 4 Dual Cell
  - 4 System x3755 login nodes
- Expansion cluster
  - 650 System x3455 compute nodes
  - 10 System x3755 compute nodes
- All connected by 10 Gbps or 20 Gbps Infiniband

Hardware introduction

- What does this hardware mean to you?
  - Performance increased thirty-fold over the old OSC systems
  - More than 75 trillion floating point operations per second peak performance

Login node configuration

- 4 system x3755 login nodes
  - Quad socket, dual core 2.6 GHz Opterons
  - 32 GB RAM
  - 225 GB local disk space in **`/tmp`**

Compute node configuration

- 877 System x3455 compute nodes
  - Dual socket, dual core 2.6 GHz Opterons
  - 8 GB RAM
  - 48 GB local disk space in /tmp
- 88 System x3755 compute nodes
  - Quad socket, dual core 2.6 GHz Opterons
  - 64 GB (2 nodes), 32GB (16 nodes), 17GB (70 nodes) RAM
  - 1.8TB (10 nodes) or 218GB (76 nodes) local disk space in /tmp

Compute node configuration

- 650 System x3455 compute nodes
  - Dual socket, quad core 2.5 GHz Opterons
  - 24 GB RAM
  - 393 GB local disk space in /tmp
- 10 System x3755 compute nodes
  - Quad socket, quad core 2.4 GHz Opterons
  - 64 GB RAM
  - 188 GB local disk space in /tmp
- Infiniband:
  - 10 Gbit/s card on old nodes
  - 20 Gbit/s card on expansion nodes

	Dual Socket	Quad Socket

Dual Core	**Number of Cores:** 4
	Memory: 8 GB
	
	To request, specify:
	`1 ≤ N ≤ 512`
	`1 ≤ C ≤ 4`
	`Type=olddual`
	Example:
	`#PBS -l nodes=10:ppn=4:olddual`
	
	Number of Machines: 877
		Number of Cores: 8
		Memory: (70) 16 GB, (16) 32 GB, (2) 64 GB
		To request, specify:
		`N = 1`
		`1 ≤ C ≤ 8`
		`Type=oldquad`
		To request memory,
		`#PBS -l mem=16GB`
		Example:
		`#PBS -l nodes=1:ppn=8:oldquad`
		
		Number of Machines: 88

Quad Core	**Number of Cores:** 8
	Memory: 24 GB
	To request, specify:
	`1 ≤ N ≤ 256`
	`5 ≤ C ≤ 8`
	`Type=newdual`
	Example:
	`#PBS -l nodes=5:ppn=8:newdual`
	
	Number of Machines: 650
		Number of Cores: 16
		Memory: 64 GB
		To request, specify:
		`N = 1`
		`9 ≤ C ≤ 16`
		`Type=newquad`
		To request memory,
		`# PBS -1 mem = 32 GB`
		Example:
		`#PBS -l nodes=1:ppn=16:newquad`
		
		Number of Machines: 8

`#PBS -1 nodes=N:ppn=C:type`

External network connectivity

- Interactive logins using the **`ssh`** protocol from anywhere on the Internet are handled by the login node, **`glenn.osc.edu`** . The **`ssh`** protocol is used because it does not send clear-text passwords and uses encryption.
- Documentation: OSC Technical Information site: **`http://www.osc.edu/supercomputing`**

The Linux Operating System

- What is Linux?
- Linux features
- Why use Linux in a cluster environment
- Processes and threads in Linux

What is Linux?

- Freely redistributable, open-source operating system
- Developed by programmers from all over the world
- Based on ideas espoused by UNIX and its variants
  - Not based directly on UNIX code
- Implements a superset of the POSIX and Open Group Single UNIX specifications for system interfaces

Linux features

- Freely distributable with full source code
- Runs on variety of platforms
- Multi-threaded, fully preemptive multitasking
- Implements most of the POSIX and Open Group Single UNIX system APIs
- Protocol and source compatibility with most other UNIX-like operating systems

Why use Linux in a cluster environment

- Widely available
- Inexpensive
- Easily modified and customized
- Compatible with most existing cluster software (MPI, batch systems, numerical libraries, etc.)
- Performs as well as or better than other operating systems on the same hardware for many technical computing applications

Processes and threads in Linux

- Historically: basic block of scheduling in UNIX—*process*
- UNIXes have added concept of multiple *threads* of execution within a single process
- Linux supports both processes and threads
- Linux's internal scheduler: tries to load-balance running processes and threads
  - Will be given full use of a processor as long as there are no more processes/threads than there are processors

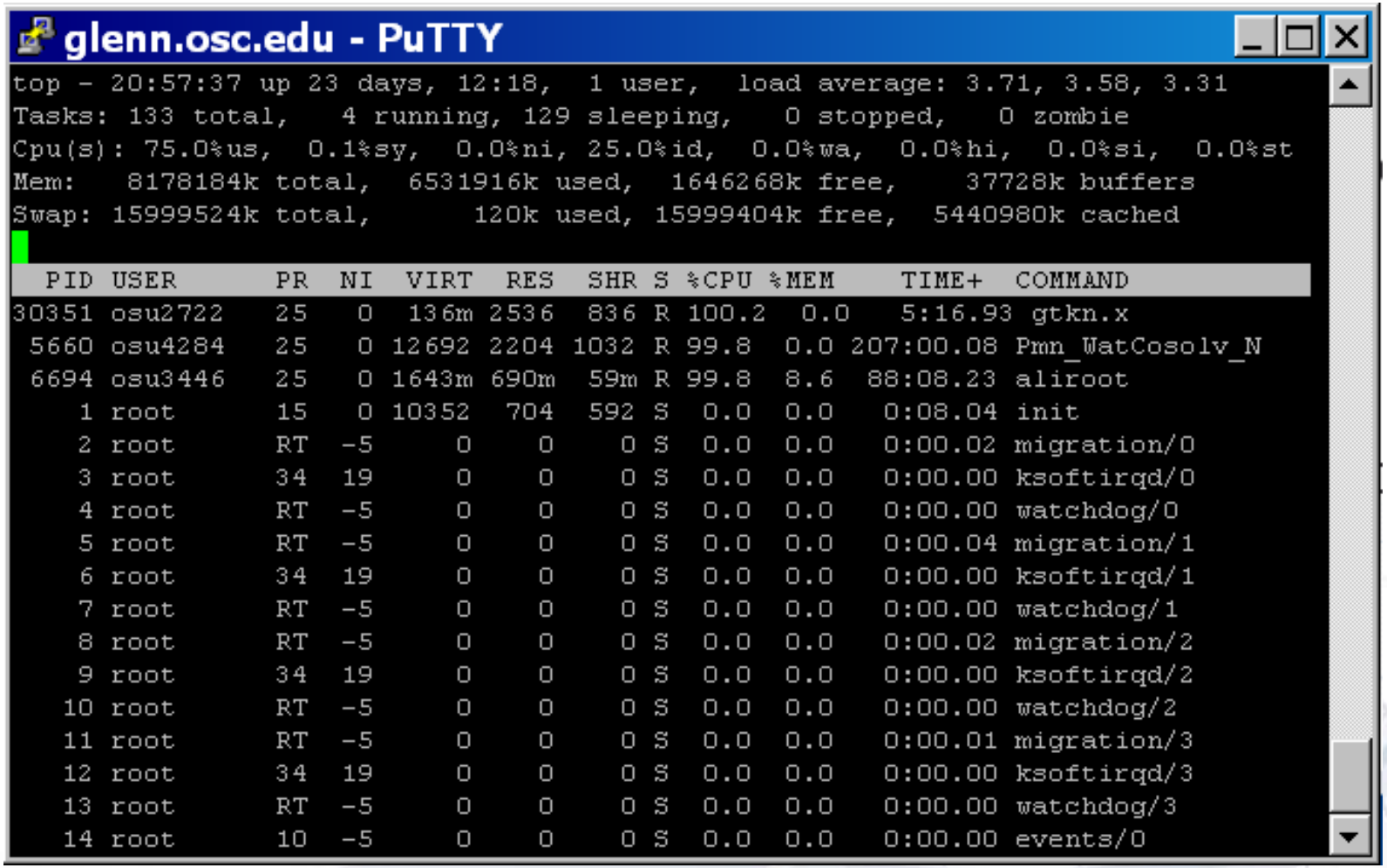
Processes and threads in Linux

- Process: a running program
- Elements of a process:
  - Memory (*text*, *data*)
  - Register contents
  - *Program Counter* (PC)
  - Process status
- Each process: unique *process id*
- Distinguish between *process* and *processor*

Processes and threads in Linux

- Process state codes
  - **S** sleeping (blocked), waiting for a resource
  - **R** running, actually doing work
  - **Z** terminated, but information still in process table
  - **T** stopped, can be restarted
- Sometimes processes *spin wait* or *busy wait*—eat CPU without doing anything useful
- Processes can be *switched out* to allow higher-priority process to run or to wait for something to happen, like I/O

Processes on a node



User Environment and Storage

- Accessing the IBM 1350 Opteron Cluster
- Modules
- Text editing
- System status
- Third party applications
- Storage

Accessing the IBM 1350 Opteron Cluster

- Connections to OSC machines are via **ssh** only
  - Linux: Use Secure Shell protocol: at prompt, enter
    ssh *userid@glenn.osc.edu*
  - Windows: **ssh** Software Needed
    - Both commercial and free versions are available
- Logging off
  - At prompt, enter "**exit**"
- Graphics
  - Standard on Unix systems
    - Need extra software for PC or Mac
  - Many performance analysis tools have an X-based GUI along with command line interfaces
  - **ssh** from a Unix/Linux machine *should* automatically configure X-Display forwarding

Accessing the IBM 1350 Opteron Cluster

- More on X-Display from IBM 1350 Opteron Cluster
  - Can run virtually any X client program on login node displayed to remote workstation
    - Prefer that you use this **only** for programs that can't be run any other way
  - Note: running remotely displayed **`xterm`** session requires much I/O bandwidth
    - Doesn't really benefit you more than **`ssh`**
  - Remote X-Display in interactive batch jobs (more in debugging MPI programs section)
    - Also supported in **`ssh`** sessions

Modules

- Modules interface
  - Allow multiple versions of software to coexist
  - Allow you to add or remove software from your environment without having to manually modify environment variables

Modules

- What modules do you have loaded?
  - At the prompt, type: **`module list`**
- What modules are available?
  - At the prompt, type: **`module avail`**
- Multiple versions of the same software
  - **`module avail matlab`**
- How to add a software module to your environment
  - At the prompt, type: **`module load`** ***`modulename`***
- How to remove a software package from your environment
  - At the prompt, type: **`module unload`** ***`modulename`***

Modules

- Modules and the UNIX shell
  - How modules work
    - Modify environment variables like **$PATH** and **$MANPATH** within your shell
      - Can be done at prompt or in **.profile** or **.cshrc**
    - Do NOT explicitly set **$PATH** in your **.profile** or **.cshrc**
    - DO append directories to the existing **$PATH**
      - Type: **setenv PATH $HOME/bin:$PATH** (csh)
      - Type: **export PATH=$HOME/bin:$PATH** (ksh)

Text editing

- Front end node has **`vi`** editor installed
  - To verify, at the prompt, type: **`which vi`**
- Popular editor, **`emacs`**, also available
  - To verify, at the prompt, type: **`which emacs`**
- To review **`vi`** and **`emacs`** usage, use a browser search engine, a commercial reference book

System status

- Useful batch queue commands
  - **`qstat`**: show status of PBS batch jobs
  - **`showstart`**: shows an **estimated** time job will start
  - **`qpeek`**: shows status of running job
  - **`OSCusage`**: shows account balance of user's project

System status

- **OSCusage**
  - Interface to OSC's local accounting database
  - RU usage for your project on specified date or range of dates
  - At prompt, type: **OSCusage**
  - Example (partial output):

```
          Usage Statistics for project PZS0002
               for 12/30/2007 to 12/30/2007

                    RU Balance: -11512.30382

  Username  Dates                                RUs  Status

   oschelp  12/30/2007 to 12/30/2007        0.00000  ACTIVE
     alanc  12/30/2007 to 12/30/2007        0.00000  ACTIVE
       srb  12/30/2007 to 12/30/2007       73.11018  ACTIVE
      jimg  12/30/2007 to 12/30/2007        0.00000  ACTIVE
   woodall  12/30/2007 to 12/30/2007        0.00000  ACTIVE
```

System status

- **OSCusage**
  - Default — usage of all members of project group listed
  - Option — use **-q** to see only your own statistics
  - At prompt, type: **OSCusage -q**

```
        Usage Statistics for project PZS0002
            for 12/30/2007 to 12/30/2007

                RU Balance: -11512.30382

Username  Dates                                RUs  Status

 woodall  12/30/2007 to 12/30/2007      0.00000  ACTIVE
 ============   PZS0002 TOTAL    0.00000
```

System status

- **`OSCusage`**
  - For date or range of dates
    - Specify start and end dates
    - Format: **`MM/DD/YYYY`**
    - End date used only if you want more than one day
  - Detailed list
    - Use **`–v`** (verbose) flag
    - How much was charged for CPU usage on each of OSC's machines

Third party applications

- Statewide Software Licensed software
  - Altair Hyperworks
    - high-performance, comprehensive toolbox of CAE software for engineering design and simulation
  - Totalview Debugger
    - performance tuning and debugging tools
  - Intel Compilers, Tools, Libraries
    - an array of software development products from Intel
  - Partek
    - an integrated, scalable software environment capable of analysis and transformation of millions of records or millions of variables

Third party applications

- **Chemistry** (asterisk indicates license must be submitted before software access is enabled)
  - *Amber
  - *Gaussian03
  - *Gaussian09
  - GROMACS
  - LAMMPS
  - MacroModel®
  - NAMD
  - *Turbomole
- **Bioinformatics**
  - Bioperl
  - BLAST
  - Clustalw
  - MrBayes

Third party applications

- **Structural Mechanics** (asterisk indicates license must be submitted before software access is enabled; ₵ indicates software is available through OSC's Statewide Software License Distribution)
  - *ABAQUS
  - ₵Altair HyperWorks
  - *ANSYS
  - *LSDYNA

Third party applications

- **Fluid Dynamics** (asterisk indicates license must be submitted before software access is enabled; ₵ indicates software is available through OSC's Statewide Software License Distribution)
  - *Fluent
  - ₵GridPro

Third party applications

- **Mathematics/Statistics**
  - MATLAB
  - Octave
  - R
  - Stata
  - FFTW
  - ScaLAPACK
  - sprng & sprng2
  - ACML
  - Intel MKL

Third party applications

- **General programming software**
  - HDF5
  - Intel compilers
  - NetCDF
  - PBS (TORQUE)
  - PGI compilers
  - Gnu Compiler and debugger

Third party applications

- **Visualization software** (◆ indicates license is available for download)
  - GNUplot
  - VTK
  - ◆VolSuite

- More applications can be found at Software page:

http://www.osc.edu/supercomputing/software/software.shtml

Storage

- OSC's Mass Storage Environment
  - Home directories -- **/home**
    - 500GB of storage
    - 1000000 files
    - If a user's account is over the limit, he/she will be unable to create new files.
  - At each login, your quota and usage information will be displayed above the command line prompt:

```
As of 2010 Jul 15 04:02 userid yzhang on /nfs/06 used 28GB of quota 500GB
and 41374 files of quota 1000000 files

As of 2010 Jul 16  04:02 project/group G-3040 on /nfs/proj01 used 27GB of
quota 0GB and 573105 files of quota 0 files
```

Storage

- OSC's Mass Storage Environment
  - Project space
    - If more than 500GB or 1000000 files are required, users can request additional storage.
    - Send a request to jimg@osc.edu with the following information:
      - Userid
      - Project number
      - Estimated amount of disk space required beyond the 500 GB of home directory storage
      - Estimated length of time the extra disk space will be required
      - Brief justification for the additional storage
    - This project storage is separate from the home directory quota and monitored through the allocations evaluation process.

Storage

- File management
  - May want to compress large, unused files
    - Use **`compress, gzip, or bzip`** commands
      - **`gzip`** tends to do a better job
  - Use **`sftp`** command to transfer files between the login node (**`glenn.osc.edu`**) and your local workstation or PC
    - All cluster nodes mount home directories from the mass storage environment; any files you transfer to your home directory on **`glenn`** will also be accessible from the other OSC systems
    - Secure remote copy command, **`scp`**, also available
      - Transfer files to interactive mode

Storage

- OSC's Mass Storage Environment
  - Local scratch space -- **`/tmp`**
    - Local **`ext3`** file system on each node
      - 48 GB on each old x3455 node
      - 393 GB on each new x3455 node
      - 1.6 TB (10 nodes) or 217 GB (76 nodes) on each old x3755 node
      - 188 GB on each new x3755 node
    - 315 TB total
    - **NOT SHARED!**
    - **NOT BACKED UP!**

Storage

- Batch-managed temporary directories
  - Every job run by the system
    - Unique local temporary directory created for it, named in the environment variable **`$TMPDIR`**
    - Stored in **`/tmp`** on each compute node assigned to the job
    - **Not shared!**

Storage

- **`pbsdcp`**: Staging data in and out of **`$TMPDIR`**
  - **`$TMPDIR`** (and **`/tmp`**)
    - not shared between nodes
    - cannot use a simple **`cp`** command to copy files into it

Storage

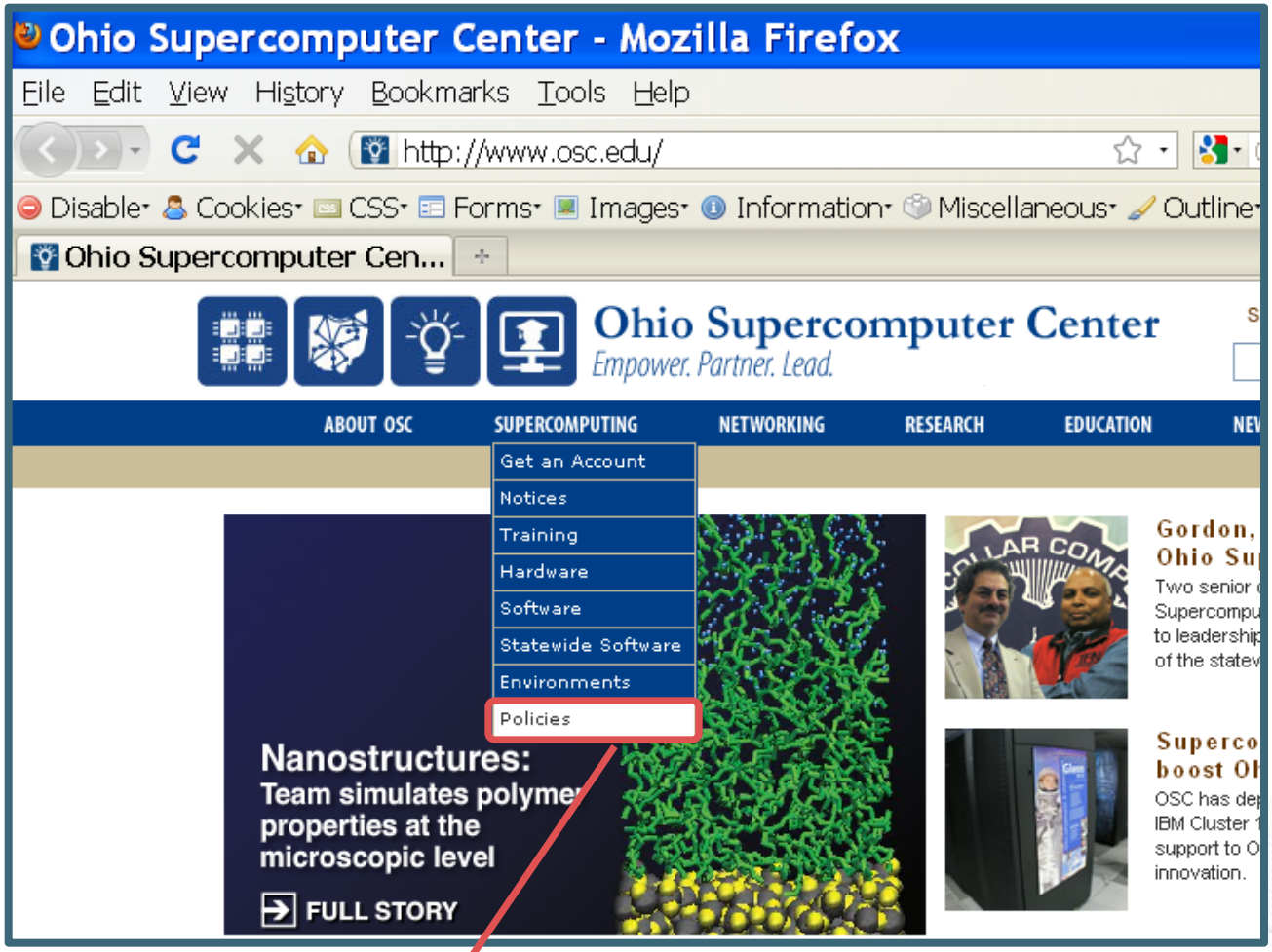
- Batch-managed temporary directories
  - **pbsdcp**: Staging data in and out of **$TMPDIR**
  - distributed copy command called **pbsdcp**
    - General format:
      - **pbsdcp [flags]** ***file1*** **[more files]** ***targetdir***
    - Scatter mode (default):
      - **pbsdcp $HOME/*inputfile* $TMPDIR**
  - Gather mode:
    - **pbsdcp –g $TMPDIR/'*' $HOME/output**
    - Enclose wildcards (*****, **?**, etc.) in single quotes in gather mode!
  - Other flags:
    - **-p** – preserve modification times
    - **-r** – recursive

Storage

- Managing file placement in jobs
  - **NEVER DO HEAVY I/O IN YOUR HOME DIRECTORY!**
    - Home directories are for long-term storage, not scratch files
    - One user's heavy I/O load can affect responsiveness for all users on that file system

Storage

- Managing file placement in jobs
  - Preferred method for I/O-intensive jobs
    - Stage input files from home directory into **`$TMPDIR`**
    - Execute calculations in **`$TMPDIR`**
    - Stage results files in **`$TMPDIR`** back to home directory



OSC Policies

OSC Policies

- OSC-1, OSC Data Lifecycle Management Policy
  - Use of home directory, project directory and **`$TMPDIR`**
  - Storage and file quotas
  - Backup and recovery

OSC Policies

- OSC-11, OSC User Management Policy
  - Who can get an account
  - Charges for accounts
  - Types of accounts
  - Account restrictions
  - Account resource units
  - Inappropriate system use