



Getting Started with CISL Facilities and Support

CESM Tutorial
12 August 2013

Rory Kelly
Consulting Services Group

CISL Mission for User Support

CISL will provide a balanced set of services to enable researchers to utilize community resources securely, easily, and effectively.

CISL Strategic Plan

CISL also supports special colloquia, workshops and computational campaigns giving users special privileges and access to facilities and services above normal service levels.



CISL Facilities Overview

- Computing Systems
 - Yellowstone
- Data Analysis and Visualization
 - Geyser and Caldera
- Data Archival
 - HPSS
 - GLADE
- Allocations and Security
- User support
- Training

Yellowstone



Yellowstone

IBM iDataPlex Cluster

- **Compute Nodes**
 - 4,518 Compute Nodes
 - 16 cores, 2.6 GHz Intel Sandy Bridge per node
 - 32 GB 1600 MHz DDR3 Memory per node
 - 72,288 Total Cores, 1.44 TB Total Memory
- **Interconnect**
 - Mellanox FDR Infiniband Network
 - Fat-tree topology
 - 13.6 GB/s bandwidth per node, 2.5 μ sec latency
 - 31.7 TB/s bisection bandwidth
- **GLADE GPFS Filesystem**
 - 10.9 PB of *useable storage*
 - 90 GB/s aggregate IO bandwidth
 - Shared across compute and DAV resources

Caldera and Geyser

Data Analysis and GPU Clusters

- **Caldera**
 - 16 visualization / compute nodes
 - 16 cores, 2.6 GHz Intel Sandy Bridge per node
 - 2 NVIDIA Tesla M2070-Q GPUs per node
 - 64 GB 1600 MHz DDR3 memory per node
- **Geyser**
 - 16 large memory nodes
 - 40 cores, 2.4 GHz Intel Westmere EX per node
 - 1 NVIDIA Quadro 6000 GPU per node
 - 1 TB 1600 MHz DDR3 memory per node
- **Pronghorn**
 - 8 cores, 2.6 GHz Intel Sandy Bridge per node
 - 2 Intel Xeon Phi 5110P coprocessors per node
 - 64 GB DDR3-1600 memory per node

GLADE Shared File System

GLADE is a high performance GPFS file system shared across CISL HPC and DAV resources

File space	Quota	Back Up	Purge Period	Description
/glade/u/home/username	10 GB	Yes	None	Home directories
/glade/scratch/username	10 TB	No	90 days	Temporary space for short term use
/glade/p/work/username	512 GB	No	None	Work space for longer term storage
/glade/p/project_code	N/A	No	None	Shared space for project allocations



Accessing Yellowstone with Yubikey

- **Use Secure Shell to log on with yubikey**
 - Terminal, Cygwin, PuTTY, etc
- **Using your Yubikey token**
 - When you log in to yellowstone
`ssh -X <username>@yellowstone.ucar.edu`
your screen displays a response:
`Token_Response:`
 - Enter your PIN number on the screen (**do not hit enter**), then touch the yubikey button. This will insert a new one-time password (OTP) and a return.
 - The yubikey is activated by the **warmth of your finger** not the pressure of pushing the button.

- **More information of Yubikey:**

<http://www2.cisl.ucar.edu/docs/yubikeys>





Shells

- **We support four shells on yellowstone**
 - tcsh
 - csh
 - bash
 - ksh
- **You can change your default shell using the System Account Manager (SAM)**
<https://sam.ucar.edu>
- **More information**
http://www2.cisl.ucar.edu/resources/yellowstone/quick_start

Supported Compilers

- **CISL supports the following compilers on yellowstone, geyser and caldera**
 - Intel (C, C++, Fortran)
 - icc, icpc, ifort, mpiicc, mpiicpc, mpiifort
 - Portland Group (C, C++*, Fortran)
 - pgcc, pgCC, pgfortran, pgf90, mpipcc, mpipf90
 - Pathscale (C, C++*, Fortran)
 - pathcc, pathCC, pathf95, mpiicc, mpief90
 - GNU (C, C++, Fortran)
 - gcc, g++, gfortran, mpigcc
- **More information**
http://www2.cisl.ucar.edu/resources/yellowstone/code_dev/compiling

Commonly Used Software

- **BLAS - Basic Linear Algebra Subroutines**
- **HDF5 - Hierarchical Data Format**
- **LAPACK and ScaLAPACK**
- **MKL - Math Kernel Library of general-purpose math routines**
- **GSL - GNU Scientific Library for C and C++ programmers**
- **NetCDF - Network Common Data**
- **PnetCDF - Parallel netCDF**
- **NCL - NCAR Command Language**
- **CDO - Climate Data Operators**
- **IDL - Interactive Data Language**
- **R - Statistical Computing Environment**
- **Python - Scripting Language**
- **Matlab - High Level Interactive Mathematical Environment**

Using Modules

Modules help manage the user software, including compilers, libraries and dependencies between them.

Helpful module commands

- **module av**
 - *lists available modules*
- **module list**
 - *show the modules currently loaded*
- **module load/unload <module-name>**
 - *load module module-name into the environment*
- **module swap <module 1> <module 2>**
 - *swap module 1 for module 2*
- **module help**
 - *display help on module commands*
- **module help <module-name>**
 - *display help specific to module-name*



Using Modules

Helpful module commands (continued)

- `module whatis <module-name>`
 - *short info on module-name*
- `module save <set-name>`
 - *save currently loaded module set as set-name*
- `module restore <set-name>`
 - *reload all modules in saved set set-name*
- `module purge`
 - *remove all loaded modules from environment*
- `module reset`
 - *reset module environment*

More Info:

<http://www2.cisl.ucar.edu/resources/software/modules>





- ## Controlling Jobs With LSF
- **Job submission (bsub)**
 - `bsub < script`
 - `bsub -U CESM_WS < script`
 - Submit a job against reservation CESM_WS
 - **Job Monitoring (bjobs and bhist)**
 - `bjobs`
 - `bjobs -u all`
 - `bjobs -q regular`
 - `bhist`
 - `bhist -n 0 jobid`
 - **Job Removal (bkill)**
 - `bkill jobid`



Example LSF Job Script

```

#!/bin/csh
# LSF batch script to run an MPI application
#
#BSUB -P SCSG0001      # project code
#BSUB -W 00:20        # wall-clock time (hrs:mins)
#BSUB -n 64           # number of MPI tasks in job
#BSUB -R "span[ptile=16]" # run 16 MPI tasks per node
#BSUB -J myjob        # job name
#BSUB -o myjob.%J.out # output file name %J is job ID
#BSUB -e myjob.%J.err # error file name %J is job ID
#BSUB -q regular      # queue

#run the executable
mpirun.lsf ./myjob.exe
    
```

For more info:

http://www2.cisl.ucar.edu/resources/yellowstone/using_resources/runningjobs

HPSS



HPSS Introduction

- High Performance Storage System (100+ PB of storage)
- Hierarchical Storage Interface (HSI) is the primary interface for data transfer to/from HPSS along with metadata access and data management.
- HPSS Tape Archiver (HTAR) is used to package files on your file system to a single archive file and then send it to HPSS.
- HPSS is used for long term archiving of files, not for short term temporary storage.

Hierarchical Storage Interface (HSI)

- **POSIX like interface**
- **Different ways to invoke HSI**
 - Command line invocation
 - hsi** cmd
 - hsi** cget hpssfile (from your default dir on HPSS)
 - hsi** cput myfile (to your default dir on HPSS)
 - Open an HSI session
 - hsi** to start a session; end, exit, quit to stop session.
restricted shell-like environment
 - **hsi** "in cmdfile"
File of commands scripted in "cmdfile"
- **Navigating HPSS while in HSI session**
 - pwd , cd, ls, cdls
- **More info**
<http://www2.cisl.ucar.edu/docs/hpss/hsi>



Data Transfer

- **Writing data – cput command**
 - `[HSI]/home/user1> cput file.01`
 - `[HSI]/home/user1> cput file.01 : new.hpss.file`
- **Reading data – cget command**
 - `[HSI]/home/user1-> cget file.01`
 - `[HSI]/home/user1-> cget file.01 : hpss.file`

Data Analysis and Visualization

- **Data Analysis and Visualization**
 - High-end servers available 7 x 24 for interactive data analysis, data-post processing, and visualization
- **Data Sharing**
 - Shared data access within the lab
 - Access to the NCAR Archival Systems and NCAR Data Sets
- **Remote Visualization**
 - Access to visual computing platforms from the convenience of your office using tcp/ip based remote image delivery service
- **Visualization Consulting**
 - Consult with CISL staff on your visualization problems





User Support

- **CISL Homepage:**
 - <http://www2.cisl.ucar.edu/>
- **CISL Consulting Services**
 - NCAR Mesa Lab Area 55, Floor 1B
- **CISL HELP**
 - Call (303)497-2400
 - Email to cislhelp@ucar.edu
 - Submit an extraview ticket