Getting Started with CISL Facilities and Support

CESM Tutorial
6 August 2018

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 Facilities Overview

- Computing Systems
  - Cheyenne
- Data Storage and Archival
  - Glade
  - HPSS
- Data Analysis and Visualization
  - Geyser and Caldera
- Allocations
- Additional Training Resources
- Contacting User support

Cheyenne

SGI ICE XA Cluster

CISL Computational & Information Systems Laboratory

NCAR NSF

Cheyenne
SGI ICE XA Cluster

- Batch Compute Nodes
  - 18-core, 2.3-GHz Intel "Broadwell" CPUs
  - 4,032 dual-socket nodes
  - 145,152 total cores, 5.34 PFLOPs peak
  - 313 TB total memory (3,164 64-GB and 864 128-GB nodes)
- High-Performance Interconnect
  - Mellanox EDR InfiniBand
  - 9-D enhanced hypercube topology
  - 100-Gbps link bandwidth — 0.5 μs latency
  - 36 TB/s bisection bandwidth
- Login Nodes
  - 6 nodes with 2.3-GHz Intel Xeon E5-2697v4 processors
  - 36 cores & 256 GB memory per node

Accessing Cheyenne

- Use SSH to log in with your yubikey

  ITerm2, Terminal, Cygwin, PuTTY, etc

- Using your Yubikey token

  log in to cheyenne

  ```
  ssh -X <username>@cheyenne.ucar.edu
  ```

  your screen will display 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 on Yubikey:

  `http://www2.cisl.ucar.edu/docs/yubikeys`

Supported Shells

- We support two shells on cheyenne
  - tcsh
  - bash
- Your shell can be changed in SAM

  https://sam.ucar.edu
- More information

https://www2.cisl.ucar.edu/resources/computational-systems/cheyenne/quick-start-cheyenne

Supported Compilers

- CISL supports three compilers on cheyenne
- Intel (C, C++, Fortran)

  icc, icpc, ifort, mpicc, mpicxx, mpif90
- Portland Group (C, C++, Fortran)

  pgcc, pgc++, pgfortran, mpicc, mpicxx, mpif90
- GNU (C, C++, Fortran)

  gcc, g++, gfortran, mpicc, mpicxx, mpif90
- More information:

https://www2.cisl.ucar.edu/resources/computational-systems/cheyenne/quick-start-cheyenne#compiling

Commonly Used Software

BLAS - Basic Linear Algebra Subroutines

LAPACK and ScaLAPACK

MKL - Math Kernel Library of general-purpose math routines

NetCDF - Network Common Data

PnetCDF – Parallel netCDF

HDF5 - Hierarchical Data Format

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

- User software on cheyenne is managed with modules
- **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

- **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:

https://www2.cisl.ucar.edu/resources/computational-systems/cheyenne/user-environment/environment-modules

Job Control With PBS

- Job submission (qsub)
  ```
  qsub script
  ```
- Job Monitoring (qstat)
  ```
  qstat
  qstat -f
  ```
- Job Removal (qdel)
  ```
  qdel jobid
  ```

More Info:

https://www2.cisl.ucar.edu/resources/computational-systems/cheyenne/user-environment/environment-modules

Example PBS Job Script

```
#!/bin/bash -l
#PBS -l select=16:mpiprocs=18:ompthreads=2
#PBS -l place=scatter:excl
#PBS -l walltime=0:45:00
#PBS -N compute_01
#PBS -j oe
#PBS -o batch.sh.out
#PBS -A SCSG0001
#PBS -q special

mpiexec_mpt -n 288 omplace -tm intel17 ./program.exe
```

GLADE File System

GLADE is a parallel file system shared across all CISL computers

File space	Quota	Back Up	Purge Period	Description
/glade/u/home/username	25 GB	Yes	None	Home directories
/glade/scratch/username	10 TB	No	60 days	Temporary space for short term use
/glade/work/username	1 TB	No	None	Work space for longer term storage
/glade/flash/username	N/A	No	2 weeks	Available by request

HPSS

HPSS Introduction

- High Performance Storage System (320+ 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
- https://www2.cisl.ucar.edu/resources/storage-and-file-systems/hpss

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 your laptop
- Visualization Consulting

  Consult with CISL staff on your visualization problems
- More info

https://www2.cisl.ucar.edu/resources/computational-systems/geyser-and-caldera

Data Analysis and Visualization

- Geyser
  - 16 large memory nodes
  - 40 cores, 2.4 GHz Intel Westmere EX per node
  - 1 NVIDIA Quadro K5000 GPU per node
  - 1 TB 1600 MHz DDR3 memory per node
- Caldera
  - 30 visualization / compute nodes
  - 16 cores, 2.6 GHz Intel Sandy Bridge per node
  - 2 NVIDIA Tesla K20X GPUs per node (some nodes)
  - 64 GB 1600 MHz DDR3 memory per node

Accessing DAV Nodes

- Interactive access to DAV nodes use the command

  `execdav`

- The execdav command has these optional arguments:
  - -a project_code (defaults to value of DAV_PROJECT)
  - -t time (minutes:seconds or hours:minutes:seconds; defaults to 6 hours)
  - -n number_of_cores (defaults to 1 core: -n 1)
  - -m nG (use this if you want to specify the amount of memory you need to use on the node, from 1 to 900 gigabytes: -m 300G, for example; if you do not specify memory per node, the default memory available is 1.87G per core that you request.)
  - -g gpu_type (gpu_type can be k20, k5000, any or none; defaults to none)

Accessing DAV Nodes
Batch Jobs

- Batch jobs on DAV nodes use the Slurm scheduler
- Basic Slurm Commands:
- Submit a job

```
sbatch script_name
```

- Check job status

```
squeue -u $USER
scontrol show job [job number]
```

- Delete a job

```
scancel [job number]
```

Accessing DAV Nodes

```
#!/bin/tcsh
#SBATCH -J job_name
#SBATCH -n 8
#SBATCH --ntasks-per-node=4
#SBATCH --mem=8G
#SBATCH -t 00:60:00
#SBATCH -A project_code
#SBATCH -p dav
#SBATCH -e job_name.err.%J
#SBATCH -o job_name.out.%J

setenv TMPDIR /glade/scratch/$USER/temp
mkdir -p $TMPDIR

module purge
module load gnu/7.3.0 ncarenv ncarcompilers openmpi

srun ./mpihello
```

Allocations and Account Info

- For accounting info and allocations log on to the Systems Account Manager (SAM)

  https://sam.ucar.edu

  - Change shells
  - Change default Unix group
  - Change HPSS default project
  - See active projects, charges, and balances

- More Info

https://www2.cisl.ucar.edu/user-support/systems-accounting-manager

Additional Training

- Consulting Group provides classes in topics of interest to yellowstone users
  - Fortran, C, C++
  - Python
  - NCL, IDL, Matlab
  - Parallel Computing with MPI and OpenMP
  - Visualization
  - Linux and UNIX Commands and System Use
- Classes are also archived for web viewing

  https://www2.cisl.ucar.edu/training

Contacting 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

Questions?

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

  On HPSS file system: pwd, cd, ls, mkdir, cdls

  On GLADE file system: lpwd, lcd, lls, lmkdir, lcdls
- More info

  https://www2.cisl.ucar.edu/resources/storage-and-file-systems/hpss

Data Transfer Individual Files

- 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
- Can also use wildcards for cget/cput
  - [HSI]/home/user1-> cget *.nc

Data Transfer: HTAR for File Collections

- Use HTAR for large numbers of files or file trees

```
htar -c –f ex_dir.tar ex_dir
```

- To list contents of an HTAR archive

```
htar -t –f ex_dir.tar
```

- Can add/retrieve files to/from an HTAR archive without downloading the whole archive
- More Info:

https://www2.cisl.ucar.edu/resources/storage-and-file-systems/hpss/using-htar-transfer-files

HPSS Help

- To get help with an HSI command that takes at least one argument, just type the command name without arguments

```
[HSI]/home/user1-> cput
```

- To get help with an HSI Command that takes no arguments, type the command followed by -?

```
[HSI]/home/user1-> ls -?
```