Getting Started with Yellowstone

November 30, 2012

Si Liu
NCAR/CISL/OSD/USS
Consulting Services Group
Yellowstone Environment
NCAR Wyoming Supercomputing Center (NWSC)

- **High Performance Computing (Yellowstone)**
  - IBM iDataPlex Cluster with Intel 2.6-GHz Sandy Bridge EP with Advanced Vector Extensions (AVX)
  - 4,518 nodes, 72,288 cores
  - 1.504 PetaFlops, 29 Bluefire-equivalents, 1.26 PetaFlops HPL
  - 32 GigaBytes memory per node, 145 TeraBytes total memory
  - Mellanox FDR InfiniBand full fat-tree interconnect

- **Centralized Filesystems and Data Storage (GLADE)**
  - >90 GB/sec aggregate I/O bandwidth, GPFS filesystems
  - 10.9 PetaBytes initially -> 16.4 PetaBytes total usable capacity in 1Q2014

- **Data Analysis and Visualization (Geyser and Caldera)**
  - Geyser: Large Memory System with Intel Westmere EX processors
    - 16 IBM x3850 nodes, 640 Westmere-EX cores
    - 1TeraBytes memory, 1 nVIDIA Quadro 6000 GPU per node
  - Caldera: GPU-Computation/Vis System with Intel Sandy Bridge EP processors with AVX
    - 16 IBM x360 M4 nodes, 256 E5-2670 cores
    - 64 GigaBytes memory, 2 nVIDIA M2070Q GPUs per node
  - Knights Corner System with Intel Sandy Bridge EP processors with AVX (2013)
    - 16 Knights Corner nodes, 256 E5-2670 cores,

- **NCAR HPSS Data Archive**
  - 2 SL8500 Tape libraries (20k cartridge slots) @ NWSC
  - >100 PetaBytes capacity (with 5 TeraBytes cartridges, uncompressed)
  - 2 SL8500 Tape libraries (15k slots) @ Mesa Lab (current 14.5 PetaBytes archive)
Yellowstone
Yellowstone System Software

- LSF-HPC Batch Subsystem / Resource Manager
  - IBM has purchased Platform Computing.
- Red Hat Enterprise Linux (RHEL) Version 6
- IBM General Parallel File System (GPFS)
- Mellanox Universal Fabric Manager
- IBM xCAT cluster administration toolkit
- IBM Parallel Environment (PE)
Debugger and Performance Tools

- **Intel** Cluster Studio (Fortran, C, C++, performance & MPI libraries, trace collector & analyzer)
- **Intel** VTune Amplifier XE performance optimizer
- **PGI** CDK (Fortran, C, C++, pgdbg debugger, pgprof)
- **PGI** CDK GPU Version (Fortran, C, C++, pgdbg debugger, pgprof) for DAV systems only
- **PathScale EKOPath** (Fortran C, C++, PathDB debugger)
- **Rogue Wave** TotalView debugger

- HPM Toolkit
- IPM
- PAPI

- TAU
- Scalasca
- Threadspotter
Most Commonly Used Packages

- **BLAS** - Basic Linear Algebra Subroutines
- **HDF5** - Hierarchical Data Format
- **LAPACK and ScaLAPACK**
  - Extensive libraries of linear algebra subroutines
  - Implemented in the optimized Math Kernel Library
  - SCALAPACK: parallel, MPI-based version of LAPACK
- **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
Login Nodes

- **Login/Interactive**
  - 6 IBM x3650 M4 nodes; Intel Sandy Bridge EP processors with AVX
  - 16 cores and 128 GB memory per node
Login to Yellowstone

• **Secure Shell (SSH)**
  – Cygwin, Putty, Terminal, etc.
  
  ```
  ssh -X username@yellowstone.ucar.edu
  ssh -X -l username yellowstone.ucar.edu
  ```

• **Token_Response:**
  – Use your YubiKey token or CRYPTOCard keypad to generate it.
  – The YubiKey is activated by the warmth of your finger not the pressure in pushing the button.

```bash
siliu@cisl-harlem ~
$ ssh -X -l siliu yellowstone.ucar.edu
Token_Response:
Last Login: Wed Sep 19 14:54:51 2012 from 128.117.11.153
-bash-4.1$
```
Shells

- **Four Shells**
  - tcsh (default)
  - csh
  - bash
  - ksh

- **Change default shell at the Systems Accounting Manager (SAM)**
  - https://sam.ucar.edu/
## GLADE File Spaces on Yellowstone

<table>
<thead>
<tr>
<th>File space</th>
<th>Quota</th>
<th>Backup</th>
<th>Purge policy</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home: /glade/u/home/username</td>
<td>10 GB</td>
<td>Yes</td>
<td>Not purged</td>
<td>User home directory</td>
</tr>
<tr>
<td>Scratch: /glade/scratch/username</td>
<td>10 TB</td>
<td>No</td>
<td>90-day since last access</td>
<td>Temporary computational space</td>
</tr>
<tr>
<td>Work: /glade/p/work/username</td>
<td>512 GB</td>
<td>No</td>
<td>Not purged</td>
<td>User work space</td>
</tr>
<tr>
<td>Project: /glade/p/[project_code]</td>
<td>N/A</td>
<td>No</td>
<td>Not purged</td>
<td>Project space allocations (via allocation request)</td>
</tr>
</tbody>
</table>

- The 90 day retention period might be changed when necessary.
- GLADE spaces described above are shared by Yellowstone, Geyser, and Caldera.
Migrating to Yellowstone GLADE (1)

- Bluefire users are responsible for migrating any files they need from the Mesa Lab GLADE file system or Bluefire's /ptmp file system to the appropriate Yellowstone GLADE file space!

- How to copy files from Bluefire to Yellowstone
  - Logon to mirage system
    
    `ssh mirage[0,1,2].ucar.edu`
  
  - Copy files from Mesa Lab(Bluefire) GLADE to Yellowstone GLADE
    
    `cp /glade/scratch/siliu/oldfile /glade/nwsc_scratch/siliu/newfile`
## Yellowstone GLADE mounted on Mirage

<table>
<thead>
<tr>
<th>File system</th>
<th>On Yellowstone</th>
<th>On Mirage system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yellowstone home space</td>
<td>/glade/u/home/username</td>
<td>/glade/u/home/username</td>
</tr>
<tr>
<td>Yellowstone scratch space</td>
<td>/glade/scratch/username</td>
<td>/glade/nwsc_scratch/username</td>
</tr>
<tr>
<td>Yellowstone work space</td>
<td>/glade/p/work/username</td>
<td>/glade/p/work/username</td>
</tr>
<tr>
<td>Yellowstone project space</td>
<td>/glade/p/[project_code]</td>
<td>/glade/p/[project_code]</td>
</tr>
<tr>
<td>Bluefire /ptmp/username</td>
<td>N/A</td>
<td>/gpfs/ptmp/username</td>
</tr>
</tbody>
</table>
Migrating to Yellowstone GLADE (2)

- SCP on Bluefire, Mirage, or Yellowstone is optional.
  - On bluefire
    `scp /glade/home/siliu/oldfile siliu@yellowstone.ucar.edu:/glade/u/home/siliu/newfile`
  - On Yellowstone
    `scp siliu@bluefire.ucar.edu:/glade/home/siliu/oldfile /glade/u/home/siliu/newfile`

- Globus Online and Gridftp will be available soon.
Accessing HPSS system

- Use High Performance Storage System (HPSS) for long-term data storage

- Please do NOT use HPSS as temporary staging area for migrating files.

- CISL HSI guide
  - https://www2.cisl.ucar.edu/docs/hpss/hsi

- CISL HTAR guide
  - https://www2.cisl.ucar.edu/docs/hpss/htar
# Queues and Charges

- **Core-hours charges**
  
  \[
  \text{wall-clock hours} \times \text{nodes used} \times \text{cores per node} \times \text{queue factor}
  \]

- **Check on computing and storage charges through SAM**

<table>
<thead>
<tr>
<th>Queue</th>
<th>Wall clock</th>
<th>Job size (# of processors)</th>
<th>Priority</th>
<th>Queue factor</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capability</td>
<td>12 hours</td>
<td>16,384-65,536</td>
<td>2</td>
<td>1.0</td>
<td>Noon Fri to 6 a.m. Mon after ASD</td>
</tr>
<tr>
<td>Regular</td>
<td>12 hours</td>
<td>16-16,384</td>
<td>2</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>Premium</td>
<td>12 hours</td>
<td>16-16,384</td>
<td>1</td>
<td>1.5</td>
<td></td>
</tr>
<tr>
<td>Economy</td>
<td>12 hours</td>
<td>16-16,384</td>
<td>3</td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>Small</td>
<td>2 hours</td>
<td>16-4,096</td>
<td>1.5</td>
<td>1.0</td>
<td>8 a.m. to 5 p.m.</td>
</tr>
<tr>
<td>Standby</td>
<td>12 hours</td>
<td>16-16,384</td>
<td>4</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>HPSS</td>
<td>12 hours</td>
<td>16-16,384</td>
<td>1</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>
Compilers

- **Intel (default)**
  - ifort, icc, icpc
  - mpif90, mpicc, mpicxx (mpiifort, mpiicc, mpiicpc)

- **PGI**
  - pgfortran, pgcc, pgCC
  - mpif90, mpicc, mpicxx (mpipf90, mpipcc, mpiCC*)

- **PathScale**
  - pathf90, pathcc, pathCC
  - mpif90, mpicc, mpicxx (mpiif90, mpiicc, mpiCC*)

- **GNU**
  - gfortran, gcc, g++
  - mpif90, mpicc, mpicxx (mpifort, mpicc, mpiCC)

- **CUDA**

- **OpenCL**

*The MPI versions of the C++ PGI and PathScale compilers are not yet available.*
Yellowstone Environment Module

- **CISL uses environment modules to help you configure your environment.**
  - Load binaries, header files, libraries, and man-paths into your current environment
  - Handle the module hierarchy of different compilers and different versions of packages conveniently
- **LMOD (Developed by Robert McLay, TACC)**
  - Lua based module system
  - Change the users' environment through modulefiles dynamically
- **NCAR environment variables**
  - `INC_NCAR, LIB_NCAR, and COMP_NCAR`
- **NCAR environment modules**
  - `ncarbinlibs`
  - `ncarenv`
  - `ncarcompilers`
NCAR Compiler Wrappers

- “ncarcompilers” module will be loaded automatically.
  - Make link/compile commands simpler
- Call the compiler wrapper as
  - `ifort -o myprog.exe myprogram.f90`
- What actually happens
  - `ifort -o myprog.exe $COMP_NCAR $INC_NCAR myprogram.f90 $LIB_NCAR`

INC_NCAR= -l/glade/apps/include -l/glade/apps/opt/netcdf/4.2/intel/12.1.4/include
LIB_NCAR= -Wl,-path,/ncar/opt/intel/12.1.0.233/composer_xe_2011_sp1.10.319/compiler/lib/ia32
-Wl,rpath,/ncar/opt/intel/12.1.0.233/composer_xe_2011_sp1.10.319/compiler/lib/intel64
-lnetcdf_c++4 -lnetcdff -lnetcdf -Wl,-rpath,/glade/apps/opt/netcdf/4.2/intel/12.1.4/lib

This rpath setting reduces the likelihood of runtime errors!
Module Usage (1)

- **List available modules**
  - `module avail` or `module av`

**Module Hierarchy on Yellowstone**

```
----------------------------------------------------
/glade/apps/opt/modulefiles/compilers
----------------------------------------------------
gnu/4.4.6 (default)  gnu/4.7.0  gnu/4.7.2  intel/12.1.4 (default)  pgi/11.5  pgi/12.5 (default)
gnu/4.6.3  gnu/4.7.1  intel/11.1.073  pathscale/4.0.12.1 (default)  pgi/12.1
----------------------------------------------------
```

```
---------------------------------------------------------
/glade/apps/opt/modulefiles/idep
---------------------------------------------------------
antlr/2.7.7  cdo/1.5.6.1 (default)  gsl/1.15  ncarbinlibs/0.0  nco/4.2.0  python/2.7.3
cdo/1.5.5  fftw/3.3.2  hwloc/1.5  ncarenv/0.0  ncview/2.1.1
---------------------------------------------------------
```

```
--------------------------------------------------------
/glade/apps/opt/modulefiles/cdep/intel
--------------------------------------------------------
hdf5-mpi/1.8.9  hdf5/1.8.9  impi/4.0.3.008  mkl/10.3  ncarcompilers/1.0  netcdf-mpi/4.2
netcdf/4.2  pnetcdf/1.3.0
```

Use "module spider" to find all possible modules.
Use "module keyword key1 key2 ..." to search for all possible modules matching any of the "keys".
Module Usage (2)

- **List loaded modules**
  - *module list* or *module li*

  Currently Loaded Modules:
  1) intel/12.1.4  2) ncarcompilers/1.0  3) netcdf/4.2
  4) ncarenv/0.0  5) ncarbinlibs/0.0

- **The following ncar modules are loaded by default.**
  - ncarcompilers
  - ncarenv
  - ncarbinlibs
Module Usage (3)

• Print whatis information about module
  – `module whatis netcdf`

  netcdf/4.2 : NetCDF version 4.2 compiled with GNU compilers

• Print help message
  – `module help netcdf`

----------------------------------- Module Specific Help for "netcdf/4.2"-----------------------------------

NetCDF (Network Common Data Form) is a set of software libraries and self-describing, machine independent data formats that support the creation, access, and sharing of array-oriented scientific data.

This module loads NetCDF 4.2 package of Fortran, C, and C++. It will place the NetCDF executables (e.g., nc-config) in your path, and also update the environment variable "NETCDF" to the NetCDF installation directory.

Loading this module helps you automatically link the related library during compiling and linking steps.

To search manual pages for NetCDF commands, please type the command "apropos netcdf".
To read manual pages for NetCDF commands, please type the command "man <command_name>".

......
Module Usage (4)

• Add module(s)
  – `module load netcdf`
  – `module add nco`

• Unload modfile1 and load modfile2
  – `module sw intel pgi`
  – Switching compilers causes dependent modules to be unloaded/loaded based on the module hierarchy.

Due to MODULEPATH changes the following modules have been reloaded:
  1) netcdf  2) ncarcompilers

• Remove module(s)
  – `module rm nco`
  – `module unload netcdf`
Module Usage (5)

Customize your own module environment

- See a list of your customized environments
  - `module list default` (or `module ld`)

- Set up additional custom environments module
  - `module sd mypgi`

- Use one of the custom environments
  - `module gd mypgi`
Module Usage (6)

Other useful module commands:

- **List all possible modules**
  - `module spider`

- **Unload all modules**
  - `module purge`

- **Do a module purge and load system defaults**
  - `module reset`

- **Print help message**
  - `module help`
Module examples

- **Compile/link with netcdf library**
  - Compile your program directly
  - NetCDF is loaded by default
  - `module load netcdf`

- **Compile/link with BLAS, ScaLAPACK, ESSL, or MKL**
  - "`module load mkl`" and compile as usual

- **Debug**
  - "`module load debug`" and compile as usual
  - "`module load totalview`" and debug with totalview
Parallel Jobs on Yellowstone

- **IBM Parallel Environment (PE)**
  - The Parallel Operating Environment (POE)
  - Message Passing Interface (MPI)
  - A parallel debugger (pdb) for debugging parallel programs
  - IBM High Performance Computing Toolkit

- **Intel MPI**
  - Under Intel compiler
  - *module load impi*
  - Link with Intel MPI library
  - Launch with IBM PE
Running Jobs (1)

#!/bin/csh
#
# LSF batch script to run an MPI application
#
#BSUB -P UUOM0001 # project code
#BSUB -W 01:30 # 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 replaced by Job ID
#BSUB -e myjob.%J.err # error file name, %J replaced by Job ID
#BSUB -q regular # queue

#run the executable
mpirun.lsf ./myjob.exe
Running Jobs (2)

- Submit jobs
  - `bsub < myscript`

- Monitor jobs
  - `bjobs`
  - `bjobs -q queue_name`
  - `bjobs -u user_name`

- Display stdout/stderr of unfinished jobs
  - `bpeek jobid`

- Kill jobs
  - `bkill jobid`
Yellowstone's First Two Months

- Yellowstone is still a work in progress.
- Most of the system will be dedicated to supporting the 11 Accelerated Scientific Discovery (ASD) projects.
- General users in first two months
  - Time for porting, testing, and preparing for the transition from Bluefire
  - Have access to 648 nodes (10,368 cores)
  - Can only submit jobs to the economy queue
  - These restrictions and policies will be changed after completing evaluation of the overall system's stability and some planned performance tests.
- Geyser and Caldera
  - Basic environment
  - Check our subsequent email
- Access to HPSS is available(hsi, htar)
Looking for Help

• **Need extra packages or libraries**
  – Package or library name
  – Version number
  – Links to the download page (if possible)

• **Meet problem when compiling or running programs**
  – The modules you loaded during your work
  – The command you used
  – The error messages you got
  – The crashed job id (if possible)
User Support Team

- **CISL and Yellowstone documentation:**
  - http://www2.cisl.ucar.edu/resources/yellowstone
  - http://www2.cisl.ucar.edu/resources/yellowstone/quick_start

- **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