Introduction to Modules at CHPC

Zhiyu (Drew) Li & Anita Orendt
Research Consulting & Faculty Engagement
Center for High Performance Computing
{zhiyu.li; anita.orendt}@utah.edu
Overview of Talk

- Why Modules
- Where to find more information
- How to setup to use modules
- Module basics
- Demonstration
What modules do

• Modules are a way of managing the user’s shell environment in an interactive session or a batch job
Why Modules

• Modules lets users dynamically change the Shell Environment – including easily adding and removing directories needed for a given task from system Environment Variables, eg $PATH, without needing to log out and back in

• Easy to switch between version of a package or application – again without having to start a new session

• Useful when packages have conflicts in their environment settings
Module Documentation at CHPC

- https://www.chpc.utah.edu/documentation/software/modules.php
- https://www.chpc.utah.edu/documentation/software/modules-advanced.php
- Video -- https://youtu.be/Cu6C5INLDAY

We make use of TACC’s LMOD

- https://www.tacc.utexas.edu/research-development/tacc-projects/lmod
- LUA based
All accounts automatically use modules –

- This is done via the login scripts (dot files) CHPC provides in home directories, even if you have older versions
- **DO NOT** make changes in the `.tcshrc` and `.bashrc`
- Use the `.custom.csh`.custom.sh to customize environment variables and/or pre-load modules for programs regularly used in ssh sessions
- Use `.aliases` file to create aliases (but not environment variables); if this file exists it will be sourced during login
  - `alias c='clear'`
- You may reset login scripts (dot files) if necessary
  - CHPC official version: /uufs/chpc.utah.edu/sys/modulefiles/templates
Basic Module commands

- **module** - shows the list of module commands
- **module load <name>** - loads module name (shortcut: **ml <name>**)
- **module unload <name>** - unloads module name (**ml -<name>**) 
- **module avail** - shows a list of "available" modules (**ml av**)
- **module list** - shows a list of loaded modules (**ml**)
- **module help** - prints help for the module command
- **module help <name>** - prints help for module
- **module show <name>** - prints the module file (**lua**)
- **module purge** - unload all modules
- **module reset system** – resets to system default
- **module swap <name1> <name2>** - swaps between two modules
- **module spider <string>** - shows all modules that have string in name
Explore CHPC Modules

- **module list (ml)** – see what are currently loaded
  
  ```
  [class99@linuxclass:~]$ module list
  Currently Loaded Modules:
  1) chpc/1.0 (S)
  
  Where:
  S: Module is Sticky, requires --force to unload or purge
  ```

- **Naming Convention** `<name>/<version>`

- **module load `<name>/<version>`** -- load a module
  
  ```
  [class99@linuxclass:~]$ module load matlab/R2022a
  [class99@linuxclass:~]$ module list
  Currently Loaded Modules:
  1) chpc/1.0 (S) 2) matlab/R2022a
  
  Where:
  S: Module is Sticky, requires --force to unload or purge
  ```

- `which matlab`

  ```
  /uufs/chpc.utah.edu/sys/installdir/matlab/R2022a/bin/matlab
  ```
Explore CHPC Modules (cont.)

- **module purge** – unload all modules

  ```bash
  [class99@linuxclass:~]$ module purge
  [class99@linuxclass:~]$ module list
  Currently Loaded Modules:
  1) chpc/1.0 (S)
  ```

- **Load a default module** (not specify the version)

  ```bash
  [class99@linuxclass:~]$ module load matlab
  [class99@linuxclass:~]$ module list
  Currently Loaded Modules:
  1) chpc/1.0 (S) 2) matlab/R2023b
  ```

  Where:
  S: Module is Sticky, requires --force to unload or purge

  ```bash
  [class99@linuxclass:~]$ which matlab
  /uuufs/chpc.utah.edu/sys/installdir/matlab/R2023b/bin/matlab
  ```
Module Avail command

- module avail (ml av) – what modules are currently loadable
  - Not an exhaust list of CHPC modules
  - Show Loadable modules at the moment
    - Standalone
    - Dependent modules are already loaded in current environment
  - List may change when other modules get loaded or unloaded
  - Marks default (D), already loaded (L), gpu specific (g) and aliases

```plaintext
masurca/3.3.1
mathematica/12.3.1
mathematica/13.3.1
mathematica/14.0.0 (D)
matlab/R2021b
matlab/R2022a
matlab/R2022b
matlab/R2023a
matlab/R2023b (L,D)
maven/3.3.3
maxquant/1.6.10.43
medaka/1.5.0 (g)
medaka/1.6.1 (g)
medaka/1.7.2 (g,D)
medea/3.7.2
mega2/6.0.0
megadetector/4.1
megalodon/2.3.3
mercurial/3.6
merlin/1.1.2
mesa/r15.14.0
```
CHPC Module Organization

• Core
  – Contains modules for applications independent of both the compiler and MPI implementation

• Compiler
  – Contains modules for applications dependent on a compiler (& version) but not on a MPI implementation

• MPI
  – Contains modules for applications dependent on both a compiler and a MPI implementation

*Modules themselves are named by application name/version*
Module Avail command (cont.)

- module load gcc/8.5.0 → module av

```
---- /uufs/chpc.utah.edu/sys/modulefiles/spack/linux-rocky8-x86_64/Compiler/linux-rocky8-nehalem/gcc/8.5.0 ----
    boost/1.77.0           mpich/4.1.2
    cantera-pokitt/develop mvapich2/2.3.6   (D)
    cdo/2.0.5              mvapich2/2.3.7
    fftw/2.1.5             netcdf-c/4.8.1
    fftw/3.3.10            netcdf-cxx/4.2
    gdal/3.3.2             netcdf-fortran/4.5.3
    geant4/10.7.3          openblas/0.3.18
    geant4/11.0.3          openkim-models/2021-01-28
    gsl/2.7                openmpi/4.1.1   (L)
    hdf4/4.2.15            openmpi/4.1.3-gpu (g)
    hdf5/1.10.7            openmpi/4.1.3
    hoomd-blue/2.5.0-gpu-mkl (g) openmpi/4.1.4   (D)
    hoomd-blue/2.5.0-gpu-obl (g,D) openmpi/4.1.5-gpu (g)
    hysplit/5.2.3          openmpi/4.1.5
    intelmpi/2019.10.317   openmpi/4.1.6-gpu (g)
    intel-oneapi-mkl/2021.4.0 openmpi/4.1.6
    intel-oneapi-mkl/2022.0.2 py-numpy/1.19.5-mkl
    intel-oneapi-mkl/2022.2.1 py-numpy/1.19.5-obl
    intel-oneapi-mpi/2021.1.1 (D) py-numpy/1.19.5
    intel-oneapi-mpi/2021.2.0 py-numpy/1.21.3-p38
```

- module load openmpi/4.1.1 → module av

```
---- /uufs/chpc.utah.edu/sys/modulefiles/spack/linux-rocky8-x86_64/MPI/linux-rocky8-nehalem/gcc/8.5.0/openmpi/4.1.1 ----
    hdf5/1.8.22    hypre/2.23.0    parallel-netcdf/1.12.2
    hdf5/1.10.7    lammmps/20220107  petsc/3.16.4
    hyphy/2.5.41    netcdf-c/4.8.1   w14mpi/3.5.0
```
Module Avail command (cont.)

- module load hdf5/1.10.7 → which h5diff

```sh
[class99@linuxclass:~]$ module load hdf5/1.10.7
[class99@linuxclass:~]$ module list
Currently Loaded Modules:
  1) chpc/1.0 (S)  2) gcc/8.5.0  3) openmpi/4.1.1  4) hdf5/1.10.7

Where:
  S: Module is Sticky, requires --force to unload or purge
```

```sh
[class99@linuxclass:~]$ which h5diff
/uufs/chpc.utah.edu/sys/spack/linux-rocky8-nehalem/gcc-8.5.0/hdf5-1.10.7-fcupy
pb7a7hytf7lwba7sgwx6hkenyv/bin/h5diff
```

- Note: For a specific application (<name>/<version>), there might be multiple modules installed by different compilers and/or mpi implementations --- For reproducibility, take notes on dependency modules as well as the target module.
Module Spider command

• Most of the time, we want to do direct searches
  – Is a module installed on CHPC?
  – What versions do you have?
  – How do I load it? (what dependencies needed?)

• `module spider <string>`
  - show all versions

  ```bash
  [class99@linuxclass:~]$ module spider hdf5
  hdf5:

  Versions:
  hdf5/1.8.19
  hdf5/1.8.22
  hdf5/1.10.7
  hdf5/1.12.2
  hdf5/1.14.1-2
  Other possible modules matches: phdf5
  ```

• `module spider <name>/<version>`
  - show how to load a specific module

  ```bash
  [class99@linuxclass:~]$ module spider hdf5/1.10.7
  hdf5: hdf5/1.10.7

  You will need to load all module(s) on any one of the lines below unless the "hdf5/1.10.7" module is available to load.
  ```
  ```bash
  gcc/11.2.0  openmpi/4.1.6
  gcc/11.2.0-cpu openmpi/4.1.6
  gcc/11.2.0-gpu openmpi/4.1.6
  gcc/8.5.0
  gcc/8.5.0 intel-oneapi-mpi/2021.4.0
  gcc/8.5.0 openmpi/4.1.1
  intel-oneapi-compilers/2021.4.0
  intel-oneapi-compilers/2021.4.0 intel-oneapi-mpi/2021.4.0
  intel-oneapi-compilers/2021.4.0 openmpi/4.1.1
  intel/2018.5.274
  nvhpc/21.5
  nvhpc/21.5-nompi
  nvhpc/21.7
  ```
Module Show command

- Format `module show <module-name>/<version>`
- Shows you the content of the module file (lua)
- This is useful if there is information on running the program included in the module
- Only works if module is available, i.e., you have modules that it depends on loaded

```
$ module show matlab/R2023b

/uufs/chpc.utah.edu/sys/modulefiles/CHPC-r8/Core/matlab/R2023b.lua:

help({[This module sets the variables for Matlab R2023b ]})
setenv("MATLAB_ROOT","/uufs/chpc.utah.edu/sys/installdir/matlab/R2023b")
setenv("MLM_LICENSE_FILE","27000@ls1.chpc.utah.edu,27000@ls2.chpc.utah.edu,27000@ls3.chpc.utah.edu")
prepend_path("PATH","/uufs/chpc.utah.edu/sys/installdir/matlab/R2023b/bin")
whatis("MATLAB_R2023b")
whatis("http://www.mathworks.com")
whatis("Installed on 09/21/2023")
family("matlab")
```
Under the hood

• Module changes Environment Variables (system & application)

```
$ module list

Currently Loaded Modules:
  1) chpc/1.0 (S)

  Where:
  S: Module is Sticky, requires --force to unload or purge

$ which matlab
/usr/bin/which: no matlab in (/uufs/chpc.utah.edu/sys/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin)

$ echo $PATH
/uufs/chpc.utah.edu/sys/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin

$ module load matlab

$ echo $PATH
/uufs/chpc.utah.edu/sys/installdir/matlab/R2023b/bin:/uufs/chpc.utah.edu/sys/installdir/matlab/R2023b/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin

$ which matlab
/uufs/chpc.utah.edu/sys/installdir/matlab/R2023b/bin/matlab
```

Other Information

• We also define module “families”; can only have one module in a family loaded at one time
  – Used for python, compilers, mpi, R
  – For example, if you have intel loaded, and load any gcc it will unload intel

• Parallel versions of boost, HDF5 have separate modules
  – hdf5 for module for serial build, phdf5 for module for parallel build
  – boost for serial, pboost for parallel
Default, aliases, and hidden modules

- For some applications have a default module – one that is installed if you do not provide a specific version
  - Typically, but not always, the latest version is specified to be the default
- For some modules, especially those with long version names, there is also an alias defined
- We have depreciated older installations and their modules so some modules have been hidden
  - Many depreciated modules were from old OS, so some may not work on current OS
  - `module --show_hidden avail`
Rocky8 changes

- For intel compilers
  - intel-oneapi-compilers
- For intel mpi
  - intel-oneapi-mpi
- For intel mkl
  - intel-oneapi-mkl
- For netcdf
  - netcdf-c, netcdf-cxx, netcdf-fortran
- For pgi – now nvhpc
- For gcc – default version is gcc/8.5.0

https://chpc.utah.edu/documentation/software/rockylinux8-osupdate.php
Rocky8 changes

• For python
  – No /usr/bin/python, instead there is /usr/bin/python2 (2.7.18), /usr/bin/python3 (3.6.8)
    • For these we have created modules python/2.7.18 and python/3.6.8 so that these can be used with ‘python’
  – CHPC installed 3.10.3 is default when using ml python

• For R
  – 4.2.2 is default
  – Can still use the containerized builds of R
    • R/4.1.2-basic
    • R/4.1.2-bioconductor
    • R/4.1.2-geospatial
Getting Help

• CHPC website
  – www.chpc.utah.edu
    • Getting started guide, cluster usage guides, software manual pages, CHPC policies

• Service-Now issue/incident tracking system
  – Help Desk: helpdesk@chpc.utah.edu

• We use chpc-hpc-users@lists.utah.edu for sending messages to users