Version DJ2012-01
18 Oct 2012

Lab 1

Programming Environment,
File Systems, Modules, Batch System

LAB I: Program Environments

- login to the Dell Linux Cluster (Lonestar) using your TACC Portal Account username:

```
$ ssh -X <username>@lonestar.tacc.utexas.edu
```

- Change to your $WORK directory:

```
$ cdw
$ pwd
```

- Untar the file lab1.tar file (in ~train00) into your directory:

```
$ tar xvf ~train00/lab1.tar
```

- Change into the newly created lab1 directory:

```
$ cd lab1
$ ls
```

Programming Environment

- Try out a few commands, for example…

```
pwd
ls -al  #both chars are lower case "L", not number 1
env
env | grep -i tacc
man ls    #press space and "q"
du -sch *
echo $WORK
history
```

Modules

- List the arguments available in the module command
  ```
  $ module
  ```
- List the modules that are presently loaded
  ```
  $ module list
  ```
- List the modules that are available
  ```
  $ module avail
  ```

- Determine which mpicc is being used and switch compiler/MPI stacks
  ```
  $ which mpicc
  $ module swap intel gcc
  $ which mpicc
  $ module swap gcc intel
  $ which mpicc
  ```

Module spider

- "module avail" tells what packages are available with current compiler and mpi implementation
- "module spider" lists all packages independent of compiler or mpi stack
- Try:

```
$ module spider
$ module spider petsc
$ module spider petsc/3.1-cxx
```

More Modules

- Test Compiler Family

```
$ module reset
$ module load gcc
```

→ expect an error

```
$ module swap intel gcc
```

→ expect message re reloaded modules

```
$ echo $TACC_FAMILY_COMPILER
```

Creating your own default setup

- Create your own initial modules for login

```
$ module reset
$ module load git mkl
$ module setdefault
```

- To reset your environment

```
$ module reset
$ module setdefault
```

SGE Batch

- **cd** to **$WORK/lab1/batch**.
- Compile the simple "hello world" fortran or C MPI code

```
$ mpif90 -O3  mpihello.f90 -o mpihello
    OR
$ mpicc  -O3  mpihello.c -o mpihello
```

- Look over the "job" script, and submit the program to SGE

```
$ qsub job
```

- Watch the status of your job (you will have to be quick to see something!)

```
$ watch showq -u -l
```
(Ctrl-C to quit watching)

- Now, put a "sleep 60" statement in the jobs script, resubmit it, & delete the job. (If you don't yet speak vi or emacs, consider giving nano a try by typing "**nano job**".)

```
$ qsub job    (observe the returned jobid , or get it from qstat)
$ qdel jobid
```