Introduction to x86 Cluster Environment

Brett D. Estrade
estrabd@lsu.edu
Objectives

- Get familiar with the shell and “environment”
- Understand the environment under the queue systems
- Learn to manipulate and control your environment
- Become familiar with the “SoftEnv” tool
- Realize that cluster environments are really no different than regular *nix environments.
Clusters

- Available:
  - eric.loni.org ... (LSU); uses CCT/SuperMike Credentials
  - oliver.loni.org ... (ULL); uses LONI Credentials

- In progress/testing
  - louie.loni.org ... (Tulane); in testing phase
  - tezpur ... (LSU HPC); no idea
Cluster Environments

• Every effort is made to maintain consistency across all clusters.

• All clusters use “SoftEnv” to manage the many available tools and libraries:
  - compilers (intel, pgi)
  - MPI implementations (MVAPICH 1&2, MPICH 1, etc)
  - Libraries (PETc, NetCDF, LINPACK, BLAS, etc)
Cluster Interaction

• One must login via a terminal interface to a head node:

  ssh username@eric.loni.org

• One there, one must interact using the login shell directly.

• It is possible to get GUI apps to run from the head node and appear on your desktop, but this is an advanced feature that can be address if time permits ... or in a more advanced tutorial.
What is the *shell*?

- The shell is a high level interface to the operating system for users.
- This is “the prompt” that you get when you login.
- Different shells are preferred different users, but they all provide the same access to the underlying OS.

```
/bin/bash  /bin/tcsh  /bin/ksh
```

How does it work?

- User issues shell command
- The shell sends a lower level command to the OS
- The OS executes the command and returns any results back to the user via the shell
Features of the *shell*

- All modern, common shells provide typical language constructs (comparison, flow control, etc).
- A set of shell commands and constructs can be saved into a text file and be run as a program; these are called *shell scripts*.
- Shells can track global variables that are referred to as their “environment”.
What is the environment?

- The shell's environment is used to store useful system information;
- This information is stored as “environmental variables”
- Some variables are set when you first login;
- Other variables can be customized by the user using a specific set of files contained in their HOME directory
What is the environment?

Global Environment

- PATH=/usr/bin:/etc:/usr/sbin:/usr/ucb:/usr/bin/X11:/sbin:/usr/java14/jre/bin:/usr/java14/bin:/usr/ucb:/sbin:/usr/sbin:/lib/instl:/usr/sbin:rsct/bin:/csm install/bin:/opt/csm/hpsnm/tools/bin:/usr/local/mgmt:/usr/lpp/LoadL/full/bin:/usr/local/bin:/usr/opt/ifor/ls/os/aix/bin:
- LOGIN=estrabd
- PWD=/home/estrabd
- HOME=/home/estrabd
- LOGNAME=estrabd

Local Environment

- /bin/bash
- /bin/tcsh
- /bin/ksh

Operating System
Viewing the environment

• View the entire environment

  $ env | more  # common to all shells

• View a specific variable

  $ printenv VARNAME  # common to all shells
  # or...

  $ echo ${VARNAME}  # common to all shells
Notable environmental variables

- **HOME**
  - Your home directory

- **PATH**
  - List of colon delimited paths that should be searched for executables.

- **LD_LIBRARY_PATH, LIBPATH** (on IBM AIX)
  - Colon delimited list of paths to look for libraries (replaces -L when linking C or Fortran programs); Note: AIX prefers, **LIBPATH**

- **LD_INCLUDE_PATH**
  - Colon delimited list of paths to look for include files (-I when linking C or Fortran programs).
Not all shells are created equally

- Beware that different shells behave differently and have different commands for similar functionality

- The 2 common families
  - Bourne Shell: *bash, ksh, zsh*
  - C-Shell: *csh, tcsh*

- Bourne
  - Good as a login shell and as a basis for a program

- C-Shell
  - Good only as a login shell; **avoid using it to program**
Manipulating the environment

• Creating or modifying a global environmental variable:

  $ export VARNAME='value'  # bourne shell

  $ setenv VARNAME 'value'  # c-shell
PATH Variables

- use to store multiple paths used for automatically searching for executables, libraries, etc
- extra care must be taken when modifying these variables because in most cases you should simply add to the existing value

```bash
$ export PATH="/my/new/path:${PATH}" # bourne shell

$ setenv PATH "'/my/new/path:${PATH}" # c-shell
```
Automating environment set up

- When one logs in interactively, the system default environment is set using the following files:
  - `/etc/profile`
    - for users of bash
  - `/etc/csh.cshrc`
    - for users of c-shell
- These files automatically set up the default path and vital user system variables
Customizing the default environment

• /bin/bash users may create the following files in their home directory
  - ~/.bash_profile # run first after /etc/profile if exists**
  - ~/.bashrc # run after ~/.bash_profile if exists
  - ~/.profile # run after ~/.bashrc if exists

• /bin/bash users may also create a file that is executed when one logs out
  - ~/.bash_logout

** ~/.bash_profile must not produce any standard out, since it will break commands such as rsync.
Customizing the default environment

• /bin/tcsh users may create the following files in their home directory

  ~/.cshrc  # run first after /etc/csh.cshrc if exists;
Managing the environment with SoftEnv

- SoftEnv is a tool that allows one to manage the environment amid the variety of software tools and libraries offered in an HPC environment.

- Often times, these software packages might conflict with one another, so SoftEnv provides a fairly straightforward way to easily ensure that your environment is set up properly.
Using SoftEnv

- SoftEnv is automatically set up for users of all shells.
- By default, only the basic set of paths are provided; the user must customize the desired packages using his ~/.soft file.
User Environment and PBS

- Currently, SoftEnv is available under the batch environment; this means that your "~/.soft" file applies when you submit a job to the queue.

- To include the current environment in a PBS batch job, be sure the following directive is placed in the queue script:

  #PBS -V

- The "-V" option may also be passed in as an argument to "qsub" when submitting a queue script:

  %qsub -V my_batch_script.pbs
Interactive PBS

• Currently, SoftEnv is available under the batch environment; this means that your “~/.soft” file applies when you submit a job to the queue.

• To include the current environment in a PBS batch job, be sure the following directive is placed in the queue script:

  #PBS -V

• The “-V” option may also be passed in as an argument to “qsub” when submitting a queue script:

  %qsub -V my_batch_script.pbs

• And for to get an interactive PBS session with the current environment:

  %qsub -I -V -lnodes=2:ppn=4 -lwalltime=00:10:00
The ~/.soft file

- Used to tell SoftEnv what values to include in the environment
- The initial ~/.soft in everyone's home directory looks like the following:

```
# This is the .soft file.
# It is used to customize your environment by setting up environment
# variables such as PATH and MANPATH.
# To learn what can be in this file, use 'man softenv'.
#
@default
```
Adding packages to ~/.soft

• Determine what is available with the “softenv” command
  
  $ softenv

• Add the line, “+package_name” to the ~/.soft file

  @default
  +package1
  +package2
  ....

• To effect the changes, use the “resoft” command
Lab 1

1) Login to {eric,oliver,louie}.loni.org

2) Determine what shell you are using with “ps”
   % ps

3) View your environment with “env”
   % env | more

4) Note to what the variable, “PATH”, is set
   % printenv PATH

5) Set a global variable, “MYNEWVAR”
   % export MYNEWVAR=123 #bash
   # setenv MYNEWVAR 123 #tcsh
Lab 2

1) Run “env” under pbs
   % pbs -I -V -lnodes=1:ppn=4 -lwalltime=00:10:00
   ...# let things initialize
   % env

2) Modify ~/.bashrc or ~/.cshrc, depending on your shell; log out and login again; determine if your changes have affected the environment
Additional resources

- https://docs.loni.org
- http://www.loni.org/help
- http://www.hpc.lsu.edu/help
- sys-help@loni.org
Free Windows Clients

- puTTY (interactive terminal)
- WinSCP (file transfer)
- OpenSSH via Cygwin (also facilitates X11 forwarding)
Local Resources

- http://docs.loni.org
- http://www.hpc.lsu.edu/help/
- Help:
  - sys-help@loni.org
  - otrs@loni.org