Working In HPC Environments With Python

Sean Patrick Santos

CESM Software Engineering Group

Yellowstone Users Seminar
Motivation

Current CESM batch script

```bash
#!/bin/tcsh -f
#BSUB -n 900
#BSUB -R "span[ptile=15]"
#BSUB -q small
#BSUB -N
#BSUB -a poe
#BSUB -x
#BSUB -o cesm.stdout.%J
#BSUB -e cesm.stderr.%J
#BSUB -J FC5_A_test
#BSUB -W 1:00
#BSUB -P P93300606
set maxthrds = 2
source /glade/apps/opt/lmod/lmod/init/csh
module load perlmods
```
Motivation

Current CESM batch script

```bash
#!/bin/tcsh -f
#BSUB -n 900
#BSUB -R "span[ptile=15]"
#BSUB -q small
#BSUB -N
#BSUB -a poe
#BSUB -x
#BSUB -o cesm.stdout.%J
#BSUB -e cesm.stderr.%J
#BSUB -J FC5_A_test
#BSUB -W 1:00
#BSUB -P P93300606
set maxthrd = 2
source /glade/apps/opt/lmod/lmod/init/csh
module load perlmods
```
Strategy
Strategy

- Python or Perl can be used for LSF batch scripts (and on other systems too).
Strategy

- Python or Perl can be used for LSF batch scripts (and on other systems too).
- Module systems also provide Python bindings!
Strategy

- Python or Perl can be used for LSF batch scripts (and on other systems too).
- Module systems also provide Python bindings!
- Just `exec` the file and go.
Strategy

- Python or Perl can be used for LSF batch scripts (and on other systems too).
- Module systems also provide Python bindings!
- Just `exec` the file and go.
- Yellowstone has
  ```
  lmod/init/env_modules_python.py
  ```
  (LMod).
Strategy

- Python or Perl can be used for LSF batch scripts (and on other systems too).
- Module systems also provide Python bindings!
- Just `exec` the file and go.
- Yellowstone has `lmod/init/env_modules_python.py` (LMod).
- Other systems have a similar file, or a binding named `python.py`.
Strategy

- Python or Perl can be used for LSF batch scripts (and on other systems too).
- Module systems also provide Python bindings!
- Just `exec` the file and go.
- Yellowstone has `lmod/init/env_modules_python.py` (LMod).
- Other systems have a similar file, or a binding named `python.py`.
- Defines a function called `module` for interaction with module systems.
Module Interface Objects
Module Interface Objects

- CESM unit tests have a Python utility script.
Module Interface Objects

- CESM unit tests have a Python utility script.
- Wants to use the same environment as CESM system tests.
Module Interface Objects

- CESM unit tests have a Python utility script.
- Wants to use the same environment as CESM system tests.
- Wants to use a more OOP interface, not building strings!
Module Interface Objects (Example)

Using environment.py

```python
import environment as ENV
mod = ENV.ModuleInterface()
mod.python_init(
    "/glade/apps/opt/lmod/.../env_modules_python.py")
mod.purge()
mod.load("ncarenv/1.0")
mod.load("ncarbinlibs/1.0")
if compiler == "intel":
    mod.load("intel/14.0.2")
elif compiler == "pgi":
    mod.load("pgi/13.9")
mod.load("ncarcompilers/1.0")
mod.load("cmake/2.8.10.2")
```
Machine Environment Config Files
Machine Environment Config Files

- Can do better than this!
Machine Environment Config Files

- Can do better than this!
- Want to separate machines (configuration) files from executable scripts as completely as possible.
Can do better than this!

Want to separate machines (configuration) files from executable scripts as completely as possible.

Next example is from CESM scripts and system tests: Perl instead of Python, written by Jay Shollenerberger in CSEG.
New XML Config Files (WIP)

```xml
<config_env>
  <machine mach="yellowstone">
    <module_system>module</module_system>
    <init_path lang="perl">
      /glade/.../init/perl
    </init_path>
    <cmd_path lang="perl">
      /glade/.../libexec/lmod
    </cmd_path>
  </machine>
</config_env>
```
New XML Config Files (WIP) Continued

```xml
<block compiler="intel">
  <module>load compiler/intel/13.1.2</module>
</block>
<block>
  <env name="OMP_STACKSIZE">256M</env>
</block>
</machine>
</config_env>
```
ModuleLoader.pm

# Use module entry.
my $modcmd = $module->textContent();
$self->moduleCmd($modcmd);

# Use environment entry.
my $varName = $env->getAttribute('name');
my $varValue = $env->textContent();
$ENV{$varName} = $varValue;
Availability
Availability

- `environment.py` is currently in CESM beta tags. It should work for Yellowstone, but other platforms are WIP.
Availability

- `environment.py` is currently in CESM beta tags. It should work for Yellowstone, but other platforms are WIP.

- Ultimate locations of CESM Python scripts still undecided; may end up on internal server, GitHub, and/or project space on Yellowstone.
Availability

- `environment.py` is currently in CESM beta tags. It should work for Yellowstone, but other platforms are WIP.

- Ultimate locations of CESM Python scripts still undecided; may end up on internal server, GitHub, and/or project space on Yellowstone.

- `environment.py` (and unit tests for it!) are at `/glade/p/cesmda/cseg/tools/python_environment`
Availability

- `environment.py` is currently in CESM beta tags. It should work for Yellowstone, but other platforms are WIP.

- Ultimate locations of CESM Python scripts still undecided; may end up on internal server, GitHub, and/or project space on Yellowstone.

- `environment.py` (and unit tests for it!) are at `/glade/p/cesmda/cseg/tools/python_environment`

- CESM scripts with XML environment configuration coming soon.