Uni.lu HPC School 2018

PS3b: Software Build and Customization using Easybuild

Uni.lu High Performance Computing (HPC) Team

S. Peter

University of Luxembourg (UL), Luxembourg

http://hpc.uni.lu

Latest versions available on Github:

UL HPC tutorials:	https://github.com/ULHPC/tutorials
UL HPC School:	http://hpc.uni.lu/hpc-school/
PS3b tutorial sources:	ulhpc-tutorials.rtfd.io/en/latest/tools/easybuild/

Summary

1 **Introduction**

2 Software/Modules Management

Main Objectives of this Session

- **Discover** Environment Modules and Lmod — Part 1
- **Installation** of EasyBuild — Part 2 (a)
- Understanding **local vs. global** installation — Part 2 (b)
- **Build your own software** on top of the provided software set — Part 2 (c)
  - ↪ local installation of HPL
  - ↪ using **existing** easyconfigs
- **Write your own easyconfig** file — Part 2 (d)
  - ↪ build the latest version of CMake

Summary

1 Introduction

2 Software/Modules Management

Software/Modules Management

https://hpc.uni.lu/users/software/

- Based on Environment Modules / LMod
  - ↪ convenient way to dynamically change the users environment `$PATH`
  - ↪ permits to easily load software through `module` command
- Currently on UL HPC:
  - ↪ > **163 software packages**, in *multiple* versions, within **18 categ.**
  - ↪ reworked software set for `iris` cluster and now deployed everywhere
    - ✓ RESIF v2.0, allowing [real] semantic versioning of released builds
  - ↪ hierarchical organization **Ex**: `toolchain/{foss,intel}`

```
$> module avail                                   # List available modules
```

```
$> module load <category>/<software>[/<version>]
```

Software/Modules Management

- Key `module` variable: `$MODULEPATH` / where to look for modules
  ↪ altered with `module use <path>`. **Ex**:

```
export EASYBUILD_PREFIX=$HOME/.local/easybuild
export LOCAL_MODULES=$EASYBUILD_PREFIX/modules/all
module use $LOCAL_MODULES
```

Software/Modules Management

- Key module variable: $MODULEPATH / where to look for modules
  ↪ altered with module use <path>. **Ex**:

```
export EASYBUILD_PREFIX=$HOME/.local/easybuild
export LOCAL_MODULES=$EASYBUILD_PREFIX/modules/all
module use $LOCAL_MODULES
```

Main modules commands:

Command	Description
`module avail`	Lists all the modules which are available to be loaded
`module spider <pattern>`	Search for among available modules **(Lmod only)**
`module load <mod1> [mod2...]`	Load a module
`module unload <module>`	Unload a module
`module list`	List loaded modules
`module purge`	Unload all modules (purge)
`module display <module>`	Display what a module does
`module use <path>`	Prepend the directory to the MODULEPATH environment variable
`module unuse <path>`	Remove the directory from the MODULEPATH environment variable

Software/Modules Management

http://hpcugent.github.io/easybuild/

- Easybuild: open-source framework to (automatically) build scientific software
- **Why?**: *"Could you please install this software on the cluster?"*
  - ↪ Scientific software is often **difficult** to build
    - ✓ non-standard build tools / incomplete build procedures
    - ✓ hardcoded parameters and/or poor/outdated documentation
  - ↪ EasyBuild helps to facilitate this task
    - ✓ **consistent** software **build and installation** framework
    - ✓ includes testing step that helps validate builds
    - ✓ **automatically generates LMod modulefiles**

```
$> module use $LOCAL_MODULES
$> module load tools/EasyBuild
$> eb -S HPL    # Search for recipes for HPL software
$> eb HPL-2.2-intel-2017a.eb # Install HPL 2.2 w. Intel toolchain
```

Software/Modules Management

- **RESIF**: Revolutionary EasyBuild-based Software Installation Framework
  - ↪ Automatic Management of **software sets**
  - ↪ Fully automates software builds and supports all available toolchains
  - ↪ Clean (hierarchical) modules layout to facilitate its usage
  - ↪ **Versioning** of software set builds
  - ↪ Easyconfig files from multiple sources
  - ↪ Define options and software in easy to read **yaml** files
  - ↪ Targeted at ULHPC sysadmins use case to build many softwares in one go

Available software sets

- Gaia
  - **default**: available by default
  - **bioinfo**: load with

```
module use $RESIF_ROOTINSTALL/bioinfo/modules/all
```

- Iris
  - **default**: available by default
  - **bioinfo**: load with

```
module use /opt/apps/resif/data/stable/bioinfo/modules/all
```

Policies

- **We provide** software that is used by many users on the cluster
- What the **users** should install themselves:
  - ↪ Python packages
  - ↪ R packages
  - ↪ Perl modules
  - ↪ software only used by 1 or 2 persons

DIY

We provide several how-tos and tutorials for installing software as a user on the ULHPC website and in our HPC school tutorials:

- Installation with **EasyBuild**
- Installation from source with `configure` and `make`
- Installation of Python packages and usage of virtual environments
- Installation of Perl modules and how to set up a local library

Or just download precompiled binaries (see Bioinformatics tutorial) ;-)

Thank you for your attention...

Questions?

http://hpc.uni.lu

High Performance Computing @ uni.lu

Prof. Pascal Bouvry
Dr. Sebastien Varrette
Valentin Plugaru
Sarah Peter
Hyacinthe Cartiaux
Clement Parisot

University of Luxembourg, Belval Campus
Maison du Nombre, 4th floor
2, avenue de l'Université
L-4365 Esch-sur-Alzette
mail: hpc@uni.lu

1. **Introduction**
2. **Software/Modules Management**