

# Using modules & EasyBuild

Eng. Fotis Georgatos <[fotis@cern.ch](mailto:fotis@cern.ch)>

NTUA

19th January 2015, Nicosia

# Index

- ▶ Environment modules
- ▶ What & why of Environment Modules
- ▶ EasyBuild
- ▶ What & why of EasyBuild
- ▶ Example run / warm up hands-on
- ▶ Contributing back

# What are Environment Modules

From abstract of John L. Furlani talk at USENIX LISA V, 1991:

- ▶ The modules package **lessens the burden** of UNIX environment maintenance,
- ▶ while providing a mechanism for the **dynamic manipulation of application changes** as single entities.
- ▶ **Users not familiar with UNIX environment benefit** most from the single command interface.
- ▶ The Module package **assists system administrators with the documentation** and dissemination of information about new and changing applications.

Ref. **How To Save The Environment** by A. Zauner, FOSDEM'14:

[http://fosdem.org/2014/schedule/track/hpc\\_and\\_computational\\_science](http://fosdem.org/2014/schedule/track/hpc_and_computational_science)

## Yet why care about modules?

- ▶ **Allows multiple software versions to happily co-exist**
- ▶ Standard practice across HPC centers worldwide
- ▶ Ease of software administration, esp. for large platforms
- ▶ Ease of use, even more so with a large user base

⇒ High Performance Computing builds ARE complex: compilers, mpi stacks, toolchains, math-libs, optimisation, `--with-*`, interconnect, accelerators

# modules example commands

## module commands basics

```
$ module avail GCC/
--- /fhgfs/euclid/buildsets/eb141014/modules/all ---
GCC/4.6.3 GCC/4.6.4 GCC/4.7.2 GCC/4.7.3 GCC/4.8.1 GCC/4.8.2
$ module load GCC          ## loads default - here GCC/4.9.2
$ module list              ## list what is now loaded
$ module swap GCC/4.6.3    ## swap a version with another
$ module show GCC/4.7.3    ## show what the changes would be
```

⇒ There are more, try: `module unload/purge/help`

## modules concepts to retain

- ▶ "modulefiles" dynamically tune an environment set specific to a software collection
- ▶ `name/version` : This is the **identifier** of a package name
- ▶ may load/unload modulefiles if there is a dependency present
  - ▶ Typically, `$MODULEPATH` points where modulefiles will be searched
- ▶ examples include but are not limited to
  - ▶ `$PATH`
  - ▶ `$LD_LIBRARY_PATH`
  - ▶ `$MANPATH`, `$NLSPATH`
  - ▶ `$LM_LICENSE_FILE`, `$_JAVA_OPTIONS`,  
`$HYDRA_LAUNCHER_EXEC`

## why Lmod

- ▶ There are many Environment modules implementations to choose from:
  - ▶ Classical Tcl/C, Tcl only, Cmod (in C), **Lmod** (in Lua) etc
- ▶ Lmod, by Robert McLay of TACC, does what others do & more:
  - ▶ supports **hierarchical** namespaces (filter by given comp/mpi)
  - ▶ automation in modules swap (think: swap the compiler!)
  - ▶ features for conflict detection (mixing two MPI stacks?)
  - ▶ spider searching and keywords
  - ▶ colouring special builds (fi. GPU, Phi)
  - ▶ active community; the author rocks in responsiveness!

⇒ EasyBuild produces Tcl modulefiles, yet Lmod reads them fine!

## Any modules horror stories?

- ▶ Plenty ;-) Yet, most of the time: just exit the shell and retry!
- ▶ A notorious bug still plagues many HPC sites, worldwide:
  - ▶ environment-modules Tcl/C **version 3.2.9 has a tricky bug**
    - ▶ and most annoying bugs are of the **heizenbug** type, oh yes
  - ▶ 3.2.10 fixes it; RHEL repos have it and others too, just use it.
  - ▶ Who fixed it? Kenneth Hoste; hint: EasyBuild primary developer!
  - ▶ Do yourself a favor and also help others to avoid this bug!



# What is EasyBuild

- ▶ **flexible framework** for building/installing (scientific) software
- ▶ fully **automates** software builds, using Python as the vehicle
- ▶ **divert from the standard** configure / make / make install with custom procedures
- ▶ allows for easily **reproducing previous builds**

Ref: RTFM? Great documentation!

<http://easybuild.readthedocs.org>

## Yet why care about EasyBuild?

- ▶ keep the software build recipes/specifications **simple and human-readable**
- ▶ supports **co-existence of versions/builds** via dedicated installation prefix and module files
- ▶ enables **sharing** with the HPC community (win-win situation)
- ▶ automagic **dependency resolution**
- ▶ **retain logs** for traceability of the build processes
  
- ▶ **hierarchical namespaces for modules**

Ref: **Modern Scientific Software Management Using EasyBuild and Lmod**, SC'14/HUST workshop paper

# EasyBuild concepts

- ▶ EasyBuild framework: the **kernel** of the build magic lives here
- ▶ easyblocks: generic and specific software build patterns
- ▶ easyconfigs: simple build descriptions, in python-like language
- ▶ Toolchains: compilers & (mpi) libs set that function as unit
  - ▶ See next slide; Yes this is big deal, do check it out

# Toolchains? Which toolchains?

- ▶ goolf
  - ▶ popular Open Source Software toolchain, incl. **GCC**, **OpenMPI**, **OpenBLAS** w. **LAPACK**, **FFTW**
- ▶ goolfc
  - ▶ like goolf, yet with **CUDA** embedded (accelerated builds!)
- ▶ goalf
  - ▶ like goolf but without OpenBLAS, replaced by **ATLAS**
- ▶ ictce
  - ▶ popular Intel toolchain, incl icc/fort, Intel mpi, Intel MKL
- ▶ cgmvolff
  - ▶ includes **Clang/C** and **GCC/fortran**, **MVAPICH**, **OpenBLAS** w. **LAPACK**, **FFTW**
- ▶ dummy
  - ▶ OS-generic toolchain, i.e., assume the distribution provides it

Ref:

[http://easybuild.readthedocs.org/en/latest/eb\\_list\\_toolchains.html](http://easybuild.readthedocs.org/en/latest/eb_list_toolchains.html)

## Suggested initial steps

- ▶ See documentation website: <http://easybuild.readthedocs.org>
- ▶ Ensure: GCC (any version) AND Python (2.4 or greater)
- ▶ Ensure: env-modules (3.2.10 or greater) OR Lmod (fi. 5.8.0)
- ▶ Find EasyBuild wiki and the list of supported apps
- ▶ `eb -version` check you are up to speed
- ▶ `eb -H` list available options - READ THIS
- ▶ If latest version needed, read and perform:  
**Bootstrapping-EasyBuild**

## Common informational options

- ▶ `eb -list-toolchains` what does each toolchain variant contain?
- ▶ `eb -list-easyblocks` lists easyblock types
- ▶ `eb -avail-easyconfig-constants` as says
- ▶ `eb -a` available easyconfig parameters will be listed
- ▶ `eb -search /R-` or `eb -S /R-` search available easyconfigs
- ▶ `eb ictce-5.5.0.eb -dry-run -r` or `eb ictce-5.5.0.eb -Dr`

⇒ The latter is essential to get an overview of intended installs!

# EasyBuild example run

## modules/EasyBuild example runs

```
[hpcadu01@euclid ~]$ module avail EasyBuild
--- /fhgfs/euclid/buildsets/eb141014/modules/all ---
EasyBuild/1.10.0      EasyBuild/1.15.2 (D)
[hpcadu01@euclid ~]$ module load EasyBuild
[hpcadu01@euclid ~]$ eb --version
[hpcadu01@euclid ~]$ eb -S hwloc/
[hpcadu01@euclid ~]$ eb --try-software-name=hwloc \
  --try-software-version=1.10.1rc1 --try-toolchain=GCC,4.8.2
[hpcadu01@euclid ~]$ module load hwloc/1.10.1rc1-GCC-4.8.2
[hpcadu01@euclid ~]$ which lstopo
[hpcadu01@euclid ~]$ lstopo -t.txt | less -RS ## bonus command
## Use arrow keys, and enjoy the view ;-)
[hpcadu01@euclid ~]$ EASYBUILD_SOURCEPATH=/fhgfs/sources/ \
time eb PRACE-20130605-golf-1.4.10.eb -r
```

# Typical Workflow

- ▶ Use `-S` to search available ready easyconfigs for GROMACS
  - ▶ `eb -S gromacs`
- ▶ Use `-Dr` to see what will be build on this CUDA variant:
  - ▶ `eb GROMACS-4.6.5-goolfc-2.6.10-hybrid.eb -Dr`
- ▶ Execute the build with:
  - ▶ `eb GROMACS-4.6.5-goolfc-2.6.10-hybrid.eb -r`

⇒ This happens when you are somewhat lucky, because somebody has already automated all the build steps! Or, it will behave right for you the 2nd time, once you have got the habit of providing fully reproducible builds and sharing with others ;-)



# Contributing back

Which type of technologies help in common HPC objectives and where/how to contribute to it?

- ▶ Use modules - any technology variety that suites you is OK
  - ▶ Document and share how modules work or do not work for you
- ▶ Use EasyBuild - the best and only alternative for its task ;-)
  - ▶ Try to provide easyconfigs/easyblocks and issue PRs on github
- ▶ Contribute documentation and/or scripts back
  - ▶ Solution is relevant when it's applicable across 2 sites/systems
  - ▶ Issue **Pull Requests** on **github.com** as soon as you have something ready to share!

## Current challenges & ongoing work; Audience feedback?

- ▶ Overall quality advancements, better tests:
  - ▶ validate installations across several HPC sites
  - ▶ implement benchmarks
  - ▶ require domain specific knowledge
- ▶ rpath vs `$LD_LIBRARY_PATH`
- ▶ Sources being removed from the web (fi. bioconductor)
- ▶ Others?

# References

- ▶ modules Homepage:
  - ▶ [http://en.wikipedia.org/wiki/Environment\\_Modules\\_\(software\)](http://en.wikipedia.org/wiki/Environment_Modules_(software))
- ▶ EasyBuild Homepage:
  - ▶ <http://hpcugent.github.io/easybuild/>
- ▶ EasyBuild Documentation:
  - ▶ <http://easybuild.readthedocs.org/en/latest/>

└ Thank you

└ Thank you

# Thank you

