HPC-CH Meeting
Software Management for HPC

Scientific Software Management at sciCORE

Pablo Escobar
Thursday, 11 June 2015

Tools we use

- Lmod
- https://www.tacc.utexas.edu/research-development/tacc-projects/lmod

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

- Automounter/Autofs
- http://en.wikipedia.org/wiki/Automounter

Why Lmod if modules works?

- Compatible with tcl module files (compatible with your current module files)
- CLI compatible with environment modules (transparent migration for users)
- Modules Cache
- Modules collections
- Support for hierarchical module tree
- Modules properties
- “ml“ wrapper
- More informative messages to users
- Support for more intuitive CLI: output to stdout, case-insensitive avail
- Modules usage tracking to syslog and mysql (released with Lmod 6.0 two days ago)
- various other enhancements: pushenv, families, sticky modules

Why Lmod if modules works?

But the main reasons why we choose Lmod are:

- Lmod is actively developed
- Lmod maintainer listens to feature requests and bug fixes
- Active collaboration between Lmod and Easybuild teams

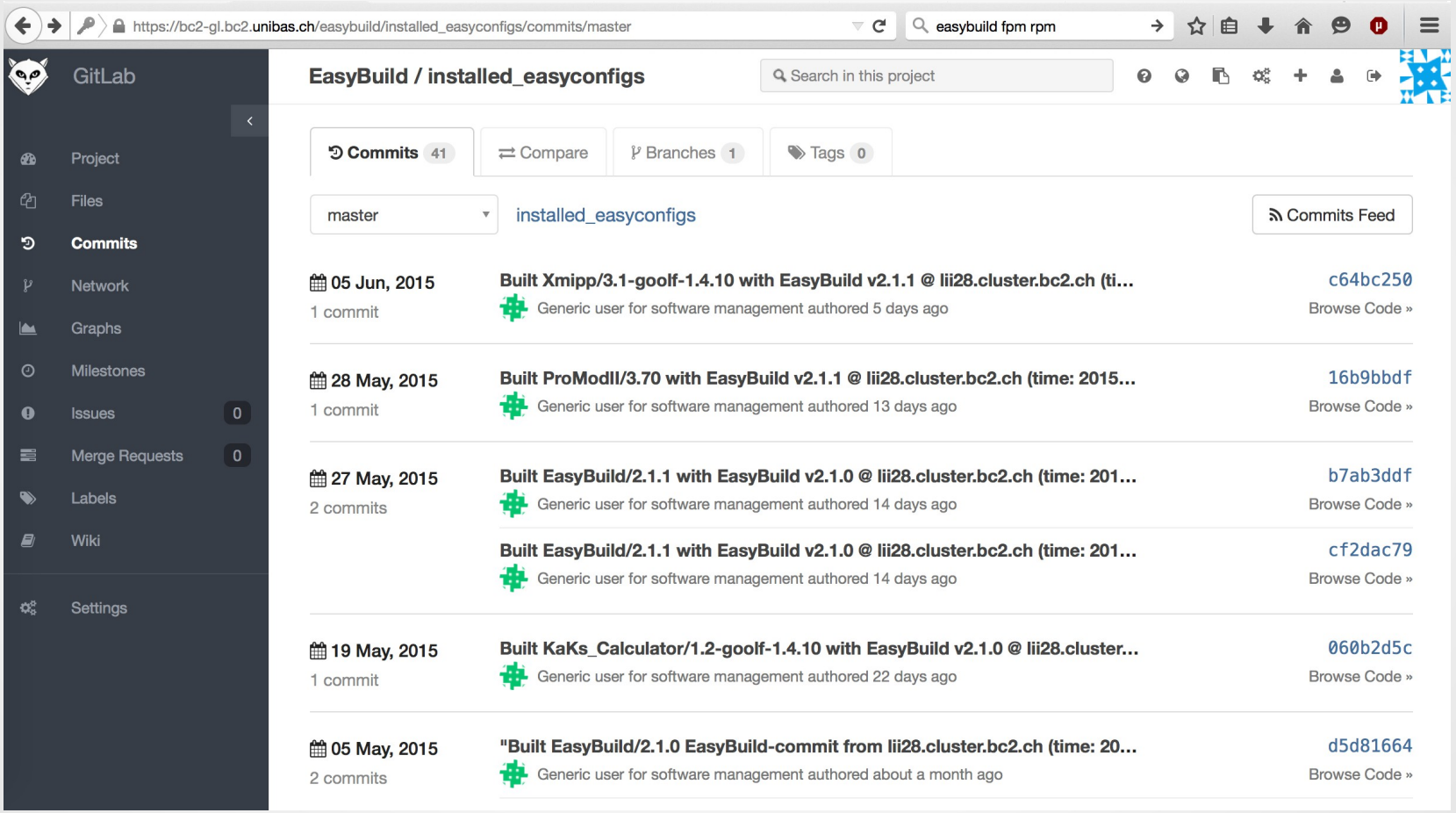
Why EasyBuild?

- Fully automates software builds and module files generation
- Easily reproduce previous builds (we can reinstall/rebuild our ~300 installed modules in a new cluster or to a different install path with just one command line)
- Keep the software build recipes/specifications simple and human-readable
- Enables sharing with the HPC community
- Automatic sanity checks (after installation EasyBuild will check that binaries and libraries are in place)

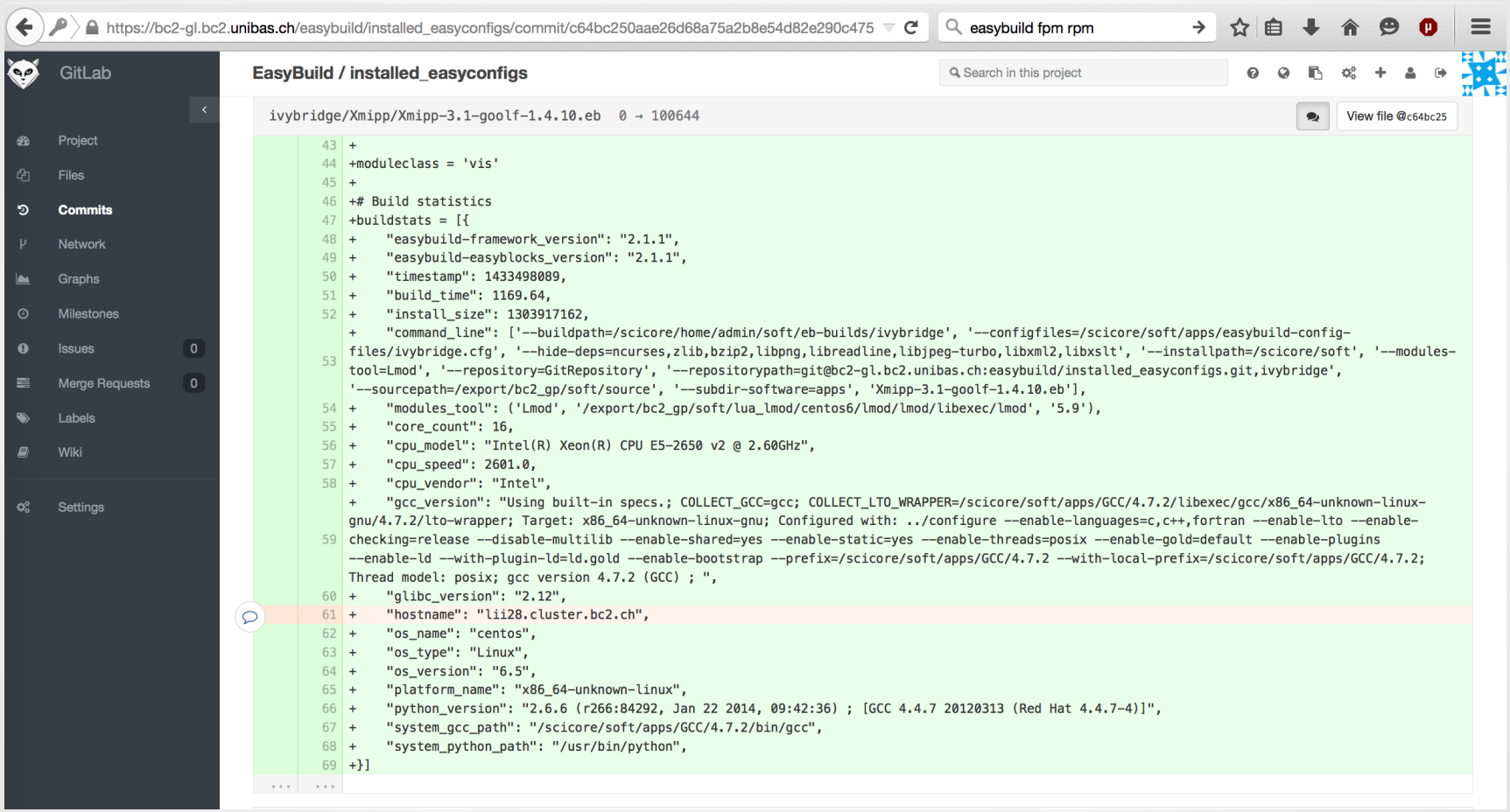
Why EasyBuild?

- Automatic dependency resolution via --robot
- Retain logs for traceability of the build processes
- Git integration so you can trace any change in the installed build recipes
- You can use it to automatize ANY install procedure (autoconf, make, cmake, python modules, perl packages, custom install scripts…)
- Most installation methods are already supported (for our ~300 apps we have just ~10 custom easyblocks)

EasyBuild: Git integration



EasyBuild: Git integration



Automounter / Autofs

- We compile each application for each of our cpu types (enabling compiler optimization)
- That means, we keep few identical software stacks. One software stack per cpu type (sandybridge, ivybridge..etc)
- With automounter we keep the software path the same in every machine but depending on the cpu type automounter points to the optimized software stack for the machine's cpu type
- Depending on the machine were the job runs the software stack optimized for that cpu type will be used without user interaction

What's next?

- Build software by submitting jobs to the cluster?
  - Next EasyBuild release will include GC3Pie integration to submit builds to the cluster (Riccardo Murri will go on details about this in his talk)

    https://github.com/hpcugent/easybuild-framework/pull/1008

- Generate RPMs with EasyBuild?
  - Next EasyBuild release will include experimental FPM integration to generate rpm or deb packages

    https://github.com/jordansissel/fpm

    http://rjeschmi-eb-draft.readthedocs.org/en/latest/Creating_binary_packages.html

    https://github.com/hpcugent/easybuild-framework/pull/1224

- Docker?

Coming EasyBuild/Lmod events

- Before next SC'15 a easybuild hackathon will take place in TACC (Austin, Texas)
  https://github.com/hpcugent/easybuild/wiki/10th-EasyBuild-hackathon#agenda

- If you are interested in learning Easybuild/Lmod directly from main developers please join :)

THANKS