



# 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

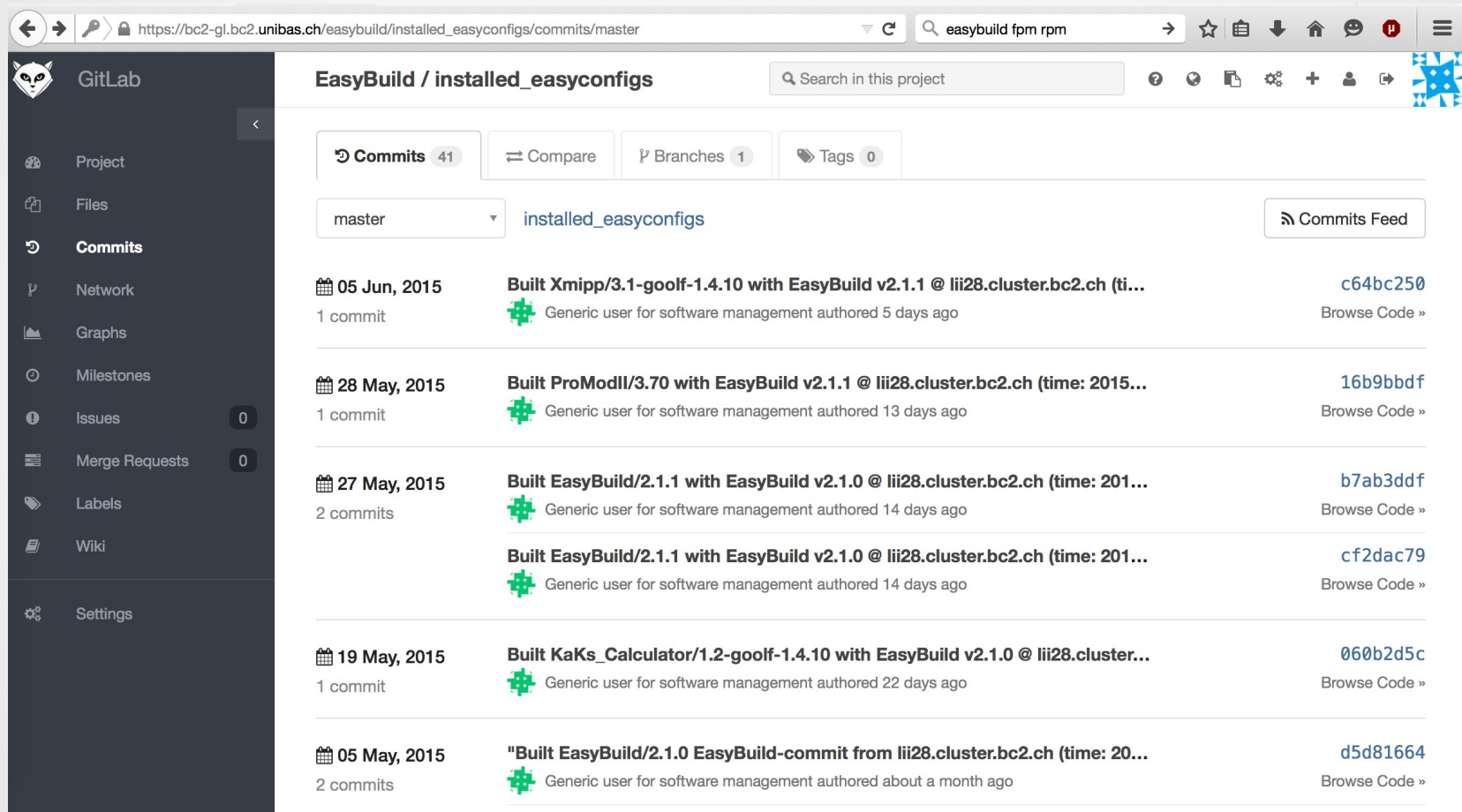
# Why EasyBuild?

- 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)

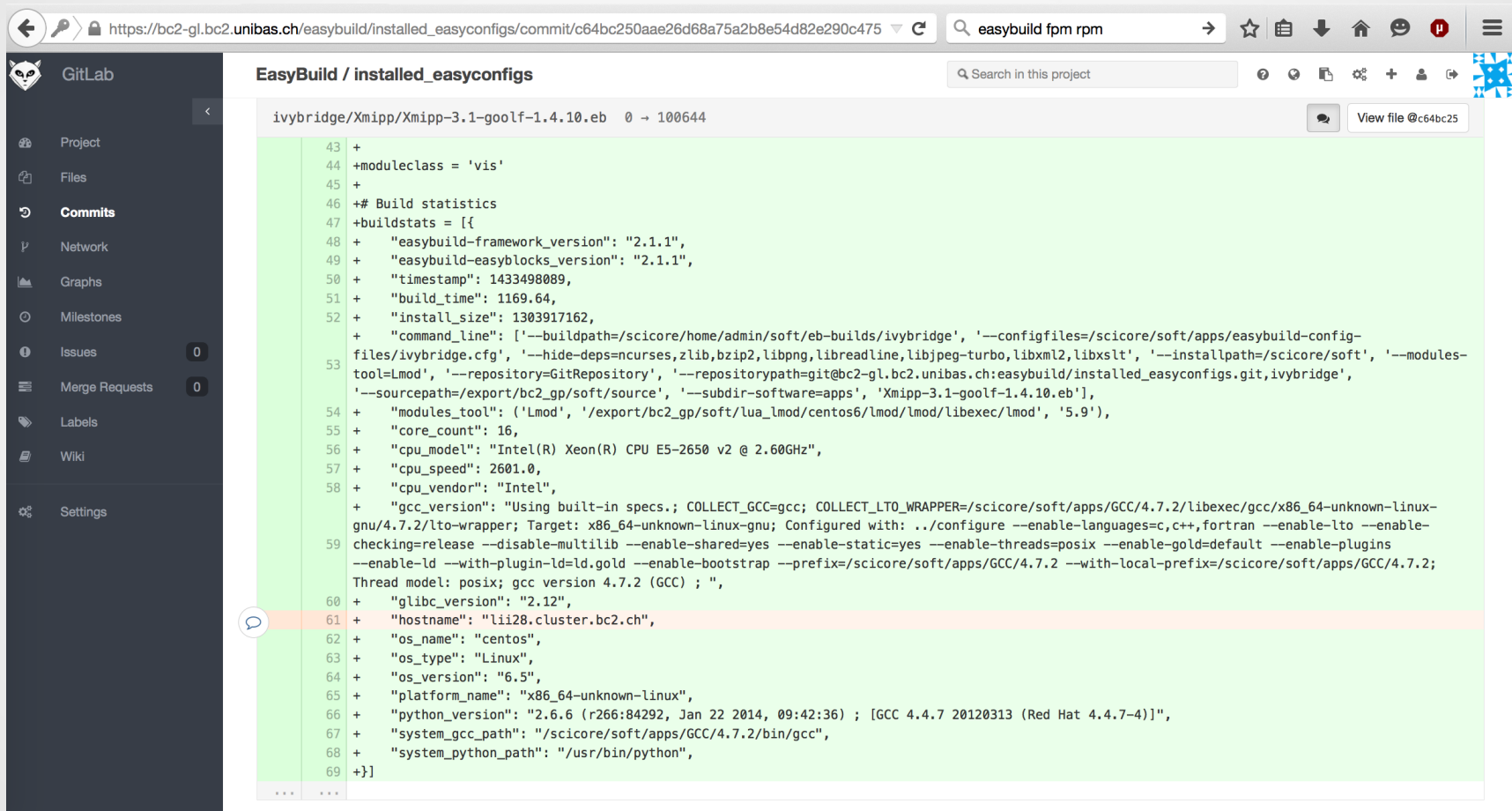
# EasyBuild: Git integration



The screenshot shows a GitLab repository page for 'EasyBuild / installed\_easyconfigs'. The browser address bar shows the URL: https://bc2-gl.bc2.unibas.ch/easybuild/installed\_easyconfigs/commits/master. The page features a dark sidebar on the left with navigation options: Project, Files, Commits, Network, Graphs, Milestones, Issues (0), Merge Requests (0), Labels, Wiki, and Settings. The main content area displays a list of commits for the 'master' branch. At the top, there are tabs for 'Commits 41', 'Compare', 'Branches 1', and 'Tags 0'. Below these, a dropdown menu shows 'master' and 'installed\_easyconfigs'. A 'Commits Feed' button is also present. The commit list includes the following entries:

Date	Commit Message	Author	Hash	Actions
05 Jun, 2015	Built Xmipp/3.1-goolf-1.4.10 with EasyBuild v2.1.1 @ lii28.cluster.bc2.ch (ti...	Generic user for software management	c64bc250	Browse Code »
28 May, 2015	Built ProModII/3.70 with EasyBuild v2.1.1 @ lii28.cluster.bc2.ch (time: 2015...	Generic user for software management	16b9bbdf	Browse Code »
27 May, 2015	Built EasyBuild/2.1.1 with EasyBuild v2.1.0 @ lii28.cluster.bc2.ch (time: 201...	Generic user for software management	b7ab3ddf	Browse Code »
	Built EasyBuild/2.1.1 with EasyBuild v2.1.0 @ lii28.cluster.bc2.ch (time: 201...	Generic user for software management	cf2dac79	Browse Code »
19 May, 2015	Built KaKs_Calculator/1.2-goolf-1.4.10 with EasyBuild v2.1.0 @ lii28.cluster...	Generic user for software management	060b2d5c	Browse Code »
05 May, 2015	"Built EasyBuild/2.1.0 EasyBuild-commit from lii28.cluster.bc2.ch (time: 20...	Generic user for software management	d5d81664	Browse Code »

# EasyBuild: Git integration



The screenshot displays a web browser window with the URL `https://bc2-gl.bc2.unibas.ch/easybuild/installed_easyconfigs/commit/c64bc250aae26d68a75a2b8e54d82e290c475`. The page title is "EasyBuild / installed\_easyconfigs". The main content area shows the text of a file, with line 61 highlighted in orange. The file content is a JSON-like structure representing build statistics and system information.

```
43 +
44 +moduleclass = 'vis'
45 +
46 +# Build statistics
47 +buildstats = [{
48 +   "easybuild-framework_version": "2.1.1",
49 +   "easybuild-easyblocks_version": "2.1.1",
50 +   "timestamp": 1433498089,
51 +   "build_time": 1169.64,
52 +   "install_size": 1303917162,
53 +   "command_line": ["--buildpath=/scicore/home/admin/soft/eb-builds/ivybridge", '--configfiles=/scicore/soft/apps/easybuild-config-files/ivybridge.cfg', '--hide-deps=ncurses,zlib,bzip2,libpng,libreadline,libjpeg-turbo,libxml2,libxslt', '--installpath=/scicore/soft', '--modules-tool=Lmod', '--repository=GitRepository', '--repositorypath=git@bc2-gl.bc2.unibas.ch:easybuild/installed_easyconfigs.git,ivybridge', '--sourcepath=/export/bc2_gp/soft/source', '--subdir-software=apps', 'Xmipp-3.1-goolf-1.4.10.eb'],
54 +   "modules_tool": ('Lmod', '/export/bc2_gp/soft/luamod/centos6/lmod/lmod/libexec/lmod', '5.9'),
55 +   "core_count": 16,
56 +   "cpu_model": "Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60GHz",
57 +   "cpu_speed": 2601.0,
58 +   "cpu_vendor": "Intel",
59 +   "gcc_version": "Using built-in specs.; COLLECT_GCC=gcc; COLLECT_LTO_WRAPPER=/scicore/soft/apps/GCC/4.7.2/libexec/gcc/x86_64-unknown-linux-gnu/4.7.2/lto-wrapper; Target: x86_64-unknown-linux-gnu; Configured with: ../configure --enable-languages=c,c++,fortran --enable-lto --enable-checking=release --disable-multilib --enable-shared=yes --enable-static=yes --enable-threads=posix --enable-gold=default --enable-plugins --enable-lt --with-plugin-lt=ld.gold --enable-bootstrap --prefix=/scicore/soft/apps/GCC/4.7.2 --with-local-prefix=/scicore/soft/apps/GCC/4.7.2; Thread model: posix; gcc version 4.7.2 (GCC) ; ",
60 +   "glibc_version": "2.12",
61 +   "hostname": "lii28.cluster.bc2.ch",
62 +   "os_name": "centos",
63 +   "os_type": "Linux",
64 +   "os_version": "6.5",
65 +   "platform_name": "x86_64-unknown-linux",
66 +   "python_version": "2.6.6 (r266:84292, Jan 22 2014, 09:42:36) ; [GCC 4.4.7 20120313 (Red Hat 4.4.7-4)]",
67 +   "system_gcc_path": "/scicore/soft/apps/GCC/4.7.2/bin/gcc",
68 +   "system_python_path": "/usr/bin/python",
69 +}]
```



# 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 where 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](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