

easybuild

building software with ease

FOSS for Scientists devroom @ FOSDEM'13
February 2nd 2013

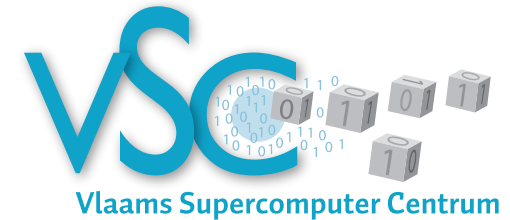
jens.timmerman@ugent.be

kenneth.hoste@ugent.be



Who are we?

- ▶ primary contact for High Performance Computing at Ghent University
- ▶ part of central IT department (DICT)
- ▶ member of Flemish Supercomputer Centre (VSC)



- ▶ HPC-UGent team consists of 7 FTEs
- ▶ 6 Tier2 systems, one Tier1 system (#163 in Top500)
528 workernodes, 8448 cores, 33TB memory, 152.3 TFlops
- ▶ system administration, user support & training, ...

Building scientific software is... fun!

Scientists focus on the functionality of their software, not on portability, build system, ...

Common **issues** with build procedures of scientific software:

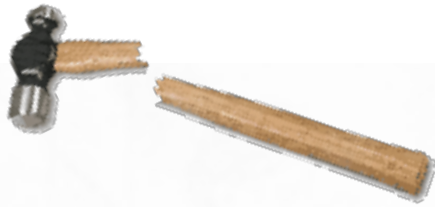
- ❑ **incomplete** build procedure
- ❑ requiring human **interaction**
- ❑ heavily **customized**
- ❑ use **hard-coded** settings
- ❑ poor and/or outdated **documentation**

Very **time-consuming** for user support teams!

**NEW VERSIONS NWCHEM, WRF
AND OPENFOAM REQUESTED**



**GOT NWCHEM TO BUILD
IN JUST 4 DAYS THIS TIME**



Current tools are lacking

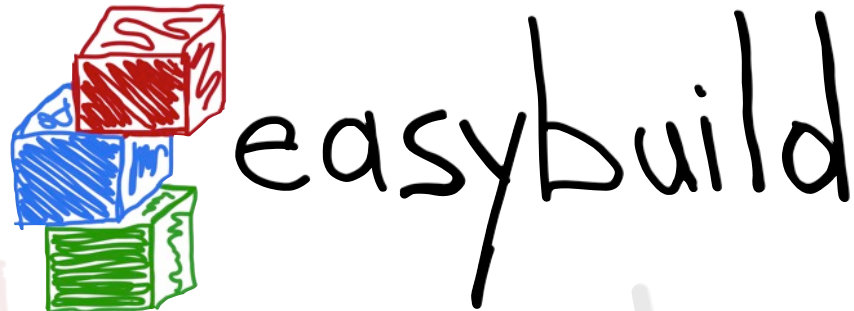
- ❏ building from **source** is preferred in an HPC environment
 - ❏ **performance** is critical, instruction selection is key (e.g. AVX)
- ❏ not a lot of packaged scientific software available (RPMs, ...)
 - ❏ requires **huge effort**, which is duplicated across distros
- ❏ existing build tools are
 - ❏ hard to **maintain** (e.g., bash scripts)
 - ❏ stand-alone, **no reuse** of previous effort
 - ❏ **OS-dependent** (HomeBrew, *Ports, ...)
 - ❏ **custom** to (groups of) software packages
e.g., Dorsal (DOLFIN), gmckpack (ALADIN)



Our build tool wish list

- ▶ **flexible** framework
- ▶ allows for **reproducible** builds
- ▶ supports **co-existence** of versions/builds
- ▶ **automated** builds and **dependency** resolution
- ▶ enables **sharing** of build procedure implementations

Building software with ease



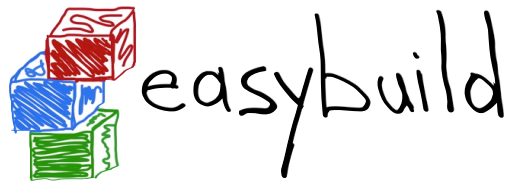
a software build and installation framework

- ❏ written in **Python**
- ❏ developed in-house (HPC-UGent) for 2.5 years
- ❏ **open-source (GPLv2)** since April 2012
- ❏ **stable API** since Nov. 2012 (v1.0.0)
- ❏ continuously enhanced and extended
- ❏ *<http://hpcugent.github.com/easybuild>*



What does EasyBuild need?

- **Linux / OS X**
 - used daily on Scientific Linux 5.x/6.x (Red Hat-based)
 - also tested on Fedora, Debian, Ubuntu, CentOS, SLES, ...
 - a couple of known issues on OS X
 - no Windows support (and none planned for now)
- **Python 2.4** or more recent 2.x
- **environment modules** (lmod support planned)
- system C/C++ compiler to bootstrap a GCC toolchain



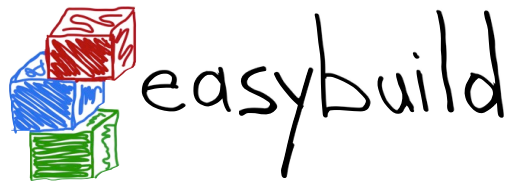
'Quick' demo for the impatient

```
$ easy_install --user easybuild
```

```
$ eb HPL-2.0-goalf-1.1.0-no-OFED.eb --robot
```




- ▣ downloads all required sources (best effort)
- ▣ constructs *goalf* toolchain, and builds HPL with it
goalf: GCC, OpenMPI, ATLAS, LAPACK, FFTW, ScaLAPACK, BLACS
- ▣ default: source/build/install dir in `$HOME/.local/easybuild`

note: we need a better *quick* demo



Terminology



framework

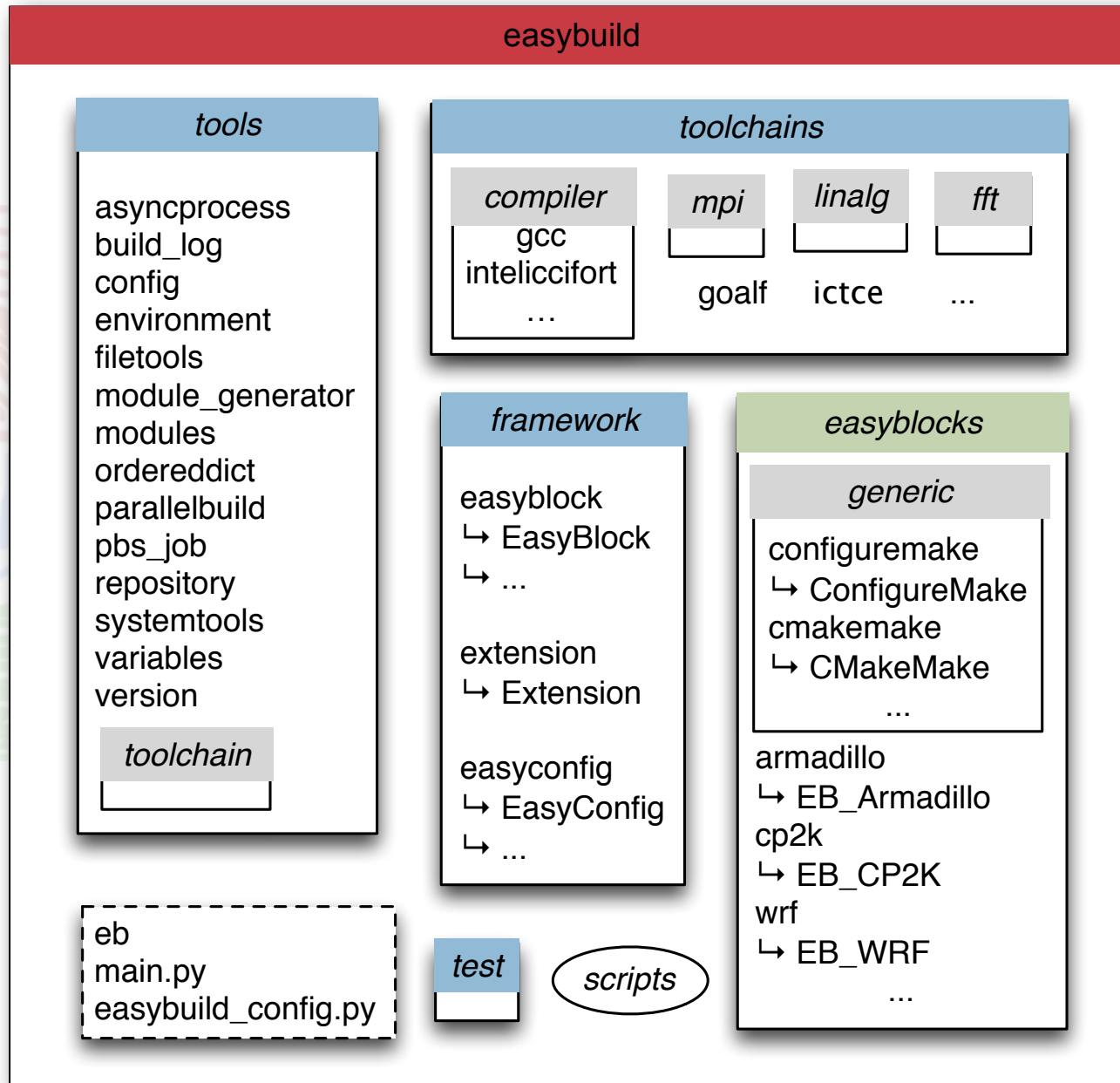
-  Python packages forming the heart of EasyBuild
-  provides (loads of) supporting functionality
-  very modular design w.r.t. toolchains and easyblocks

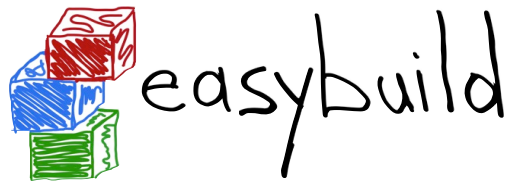
easyblock

-  Python module providing an implementation of a build procedure
-  generic or software-specific

easyconfig file (.eb)

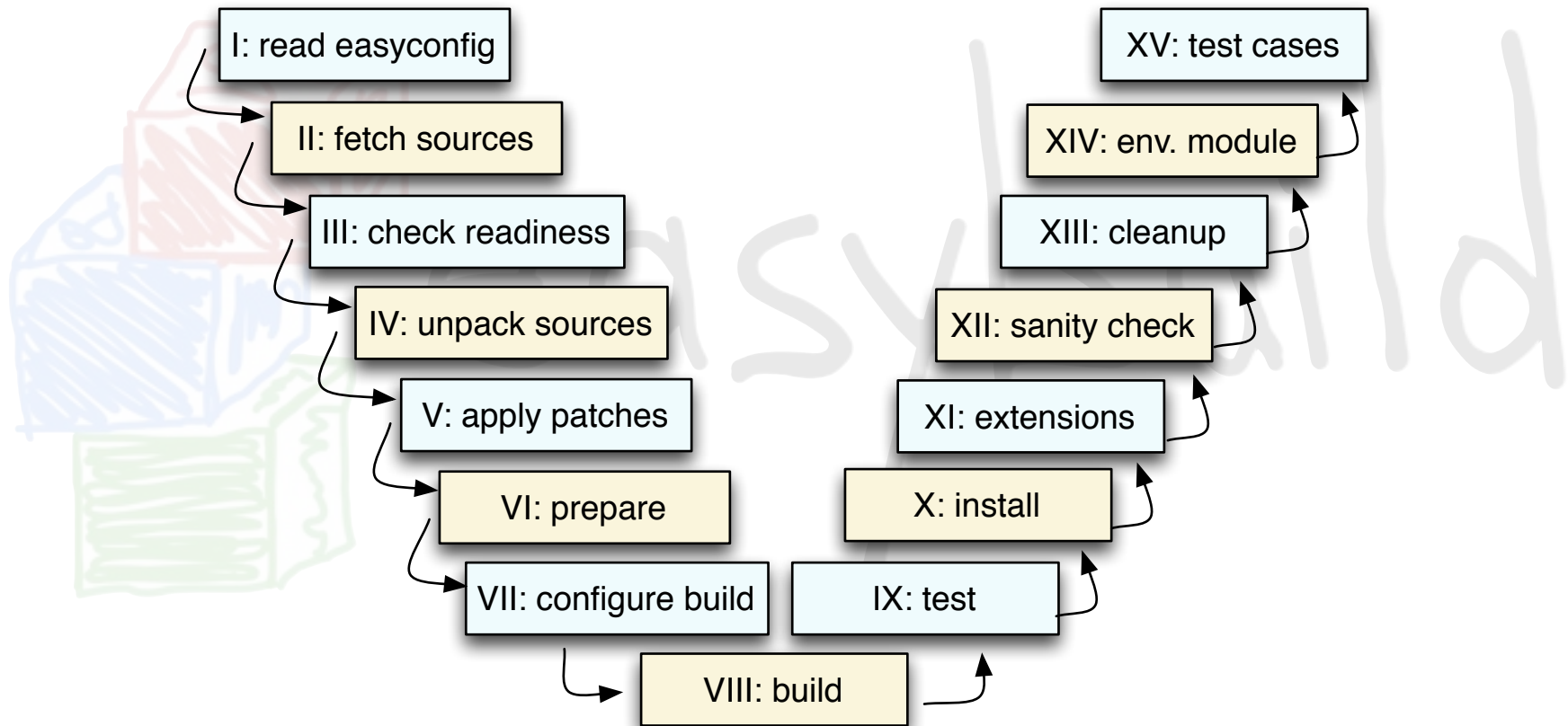
-  build specification: name, version, toolchain, build options, ...
-  simple text files, Python syntax





Step-wise install procedure

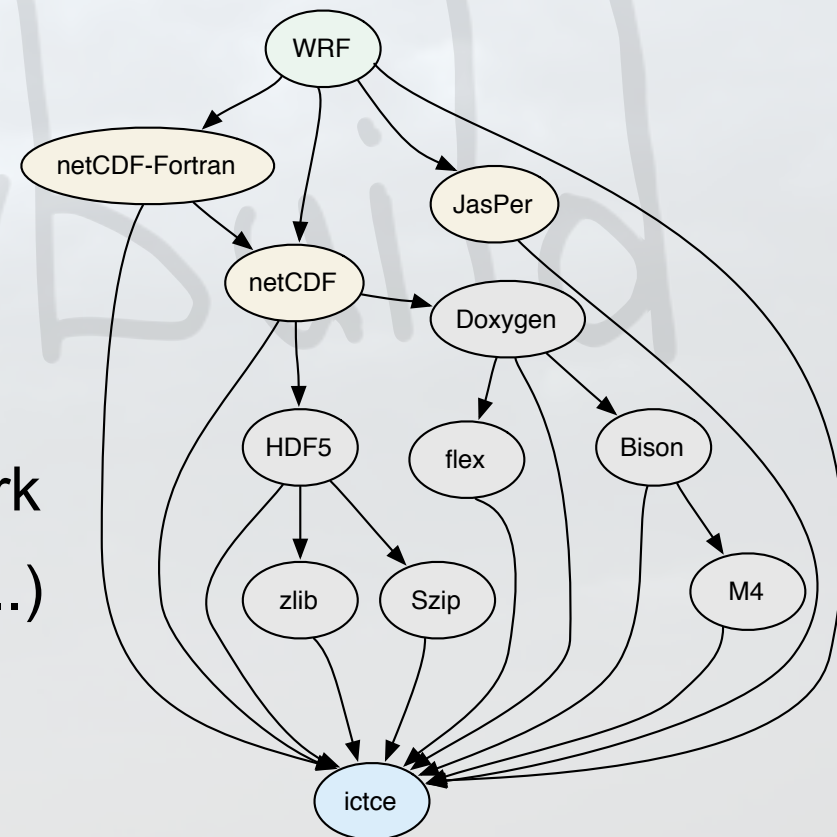
build and install procedure as implemented by EasyBuild

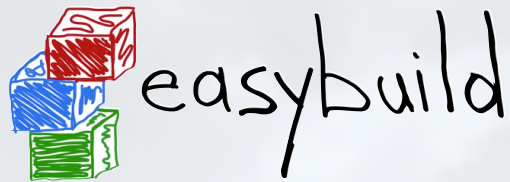


most of these steps can be customized if required

*building and installing **WRF** (Weather Research and Forecasting Model)*

- ▶ <http://www.wrf-model.org>
- ▶ complex(ish) **dependency graph**
- ▶ very **non-standard build procedure**
 - ▶ interactive `configure` script (!)
 - ▶ resulting `configure.wrf` needs work (hardcoding, tweaking of options, ...)
- ▶ `compile` script (wraps around `make`)
- ▶ no actual installation step



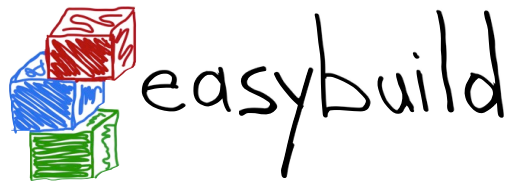


Example use case (2/2)

*building and installing **WRF** (Weather Research and Forecasting Model)*

- ▶ easyblock that comes with EasyBuild implements build procedure
 - ▶ running `configure` script **autonomously**
 - ▶ **building** with `compile` and **patching** `configure.wrf`
 - ▶ **testing** build with standard included tests/benchmarks
- ▶ various example `easyconfig` files available
 - different versions, toolchains, build options, ...
- ▶ building and installing WRF becomes child's play, for example:

```
eb --software=WRF,3.4 --toolchain-name=ictce --robot
```



Features

■ **logging and archiving**

- entire build process is logged thoroughly, logs stored in install dir
- easyconfig file used for build is archived (file/svn/git repo)

■ **automatic dependency resolution**

- software stack be built with a single command, using `--robot`

■ **running interactive installers autonomously**

- by passing a Q&A Python dictionary to the `run_cmd_qa` function

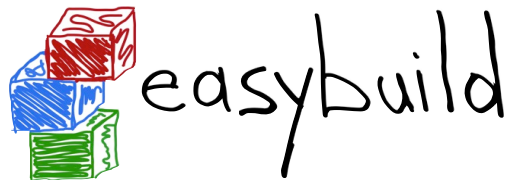
■ **building software in parallel**

- e.g., on a (PBS) cluster, by using `--job`

■ **comprehensive testing: unit tests, regression testing**

Our build tool wish list

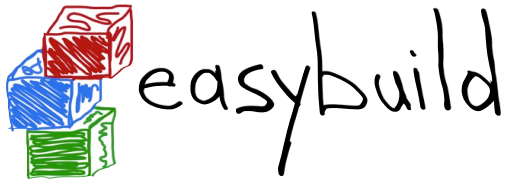
- ▶ **flexible** framework
 - ⇒ **additional support plugs in easily**
- ▶ allows for **reproducible** builds
 - ⇒ **easyblocks + robust framework = reproducibility**
- ▶ supports **co-existence** of versions/builds
 - ⇒ **installs in custom prefix, generates env. modules**
- ▶ **automated** builds and **dependency** resolution
 - ⇒ **'robot' feature for handling dependency hells**
- ▶ enables **sharing** of build procedure implementations
 - ⇒ **easyblocks can be shared or contributed back**



List of supported software (v1.1.0)

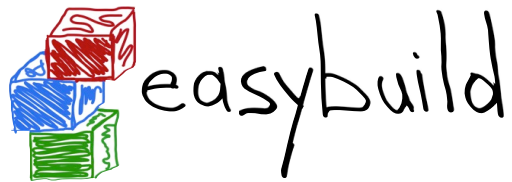
198 different software packages (488 example easyconfigs)

ABINIT ABySS ACML **ALADIN** AMOS ant Armadillo ASE ATLAS Autoconf BEAGLE BFAST
BiSearch Bison BLACS BLAST Boost Bowtie2 BWA byacc bzip2 CGAL Chapel CLHEP
ClustalW2 CMake **CP2K** CPLEX Cufflinks cURL CVXOPT Cython Docutils DOLFIN
Doxygen ECore Eigen ESPResSo expat FASTX-Toolkit FFC FFTW FIAT flex freetype FSL
g2clib g2lib GATE GCC Geant4 GEOS GHC git glproto gmacml GMP gmvapich2 goalf
gompi google-sparsehash GPAW Greenlet grib_api GSL guile h5py h5utils Harminv HDF
HDF5 HMMER HPL hwloc Hypre icc iccifort ictce ifort imkl impi Infernal Instant iomkl ipp itac
JasPer Java Jinja2 JUnit LAPACK libctl libdrm libffi libgtextutils libibmad libibumad libibverbs
Libint libmatheval libpciaccess libpng libpthread-stubs libreadline libsmm libtool libunistring
libxcb libxml2 libxslt M4 makedepend Maple matplotlib MCL MDP Meep MEME Mercurial
Mesa MetaVelvet METIS MPFR mpiBLAST MrBayes MTL4 MUMmer MVAPICH2 NCBI-
Toolkit **NCL** ncurses netCDF netCDF-Fortran NEURON numpy **NWChem** Oger **OpenFOAM**
OpenMPI OpenPGM OpenSSL ORCA PAPI ParMETIS Pasha paycheck PCRE **PETSc**
petsc4py pkg-config Primer3 Python python-meep PyZMQ **QuantumESPRESSO** R RNAz
ROOT SAMtools ScaLAPACK ScientificPython scipy SCOOP SCOTCH setuptools Shapely
SHRiMP SLEPc SOAPdenovo Sphinx SuiteSparse SWIG Szip tbb Tcl Theano Tk Tophat
Tornado Trilinos Trinity UFC UFL util-linux Velvet ViennaRNA Viper VSC-tools **WIEN2k**
wiki2beamer **WPS WRF** xcb-proto XCrySDen XML xorg-macros xproto ZeroMQ zlib



Current status

- **EasyBuild v1.1.0** released January 27 2013
 - planned monthly releases (v1.x.0), bugfix releases as needed
 - various features pending:
 - **more flexibility**, e.g., module naming scheme, lmod support
 - bring **documentation** wiki up-to-date
 - generate **packages** for supported software (RPMs, .deb, ...)
 - support for **more software** and additional **compiler toolchains**
 - **small community**, growing steadily
 - UGent + VSC partners
 - University of Luxembourg
 - Gregor Mendel Institute (Austria)
 - ...



Call for contribution

❏ **feedback**

- ❏ give it a spin, let us know how it turns out
- ❏ what features do you require that are missing?

❏ **report problems**

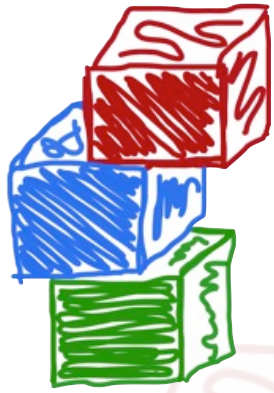
- ❏ via mail, GitHub issue tracker, IRC, ...

❏ **help verify the correctness of easyblocks/easyconfigs**

❏ **contribute back**

- ❏ features in framework, support for additional compilers/libraries
- ❏ easyblocks and/or easyconfig files (new software, updates, ...)

Let's build a community to tackle this problem together!



easybuild

building software with ease

website: <http://hpcugent.github.com/easybuild>

GitHub: [https://github.com/hpcugent/easybuild\[-framework|-easyblocks|-easyconfigs\]](https://github.com/hpcugent/easybuild[-framework|-easyblocks|-easyconfigs])

PyPi: [http://pypi.python.org/pypi/easybuild\[-framework|-easyblocks|-easyconfigs\]](http://pypi.python.org/pypi/easybuild[-framework|-easyblocks|-easyconfigs])

mailing list: easybuild@lists.ugent.be

Twitter: [@easy_build](https://twitter.com/easy_build)

IRC: [#easybuild](https://freenode.net) on freenode.net

