10 years of easybuild and the road ahead

Kenneth Hoste (HPC-UGent)

kenneth.hoste@ugent.be

20190614 - HPCKP'19 (Barcelona)

https://users.ugent.be/~kehoste/EasyBuild_20190614_10_years.pdf

https://easybuilders.github.io/easybuild

https://easybuild.readthedocs.io

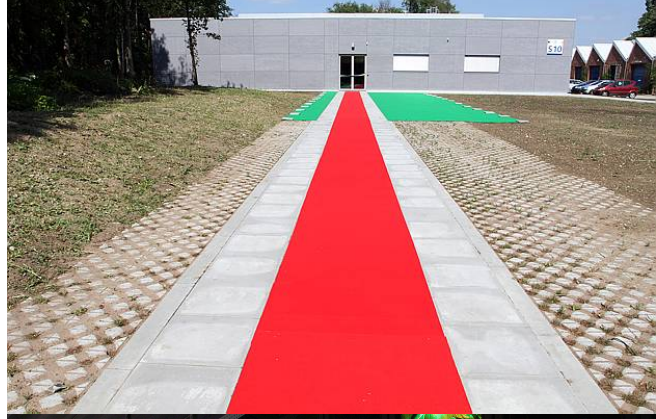
whoami

kenneth.hoste@ugent.be
@boegel *(GitHub, IRC, Slack)*
@kehoste *(Twitter)*

- Masters & PhD in Computer Science from Ghent University
- PhD topic: machine learning applied to software performance, compilers, ...
- joined HPC-UGent team in October 2010
- main tasks: user support & training, *software installations*
- slowly also became  **lead developer & release manager**
- likes family, beer, loud music, FOSS, helping people, dad jokes, stickers, ...
- doesn't like CMake, SCons, Bazel, TensorFlow, OpenFOAM, ...

HPC-UGent

GHENT UNIVERSITY



- part of central IT department of Ghent University (Belgium)
- centralised scientific computing services, training & support
- for researchers of UGent, industry & knowledge institutes
- 6 Tier-2 clusters (> 15k cores in total), ~2 PB shared storage
- 7+1 team members, > 3000 user accounts
- member of Flemish Supercomputer Centre (VSC)
  https://www.vscentrum.be

Vlaanderen
is computing

easybuild in a nutshell

https://easybuilders.github.io/easybuild - https://easybuild.readthedocs.io

- **framework for installing scientific software** *(built from source when possible)*
- strong focus on Linux & HPC systems (and hence also performance)
- **by default builds/optimises specifically for host architecture**
- implemented in Python 2, lead development by HPC-UGent
- available under GPLv2 license via PyPI, GitHub
- supports different compilers & MPI libraries, x86_64/ARM/POWER, ...
- currently supports 1,780 different software packages (+ extensions)
- welcoming and helpful worldwide community

terminology

http://easybuild.readthedocs.io/en/latest/Concepts_and_Terminology.html

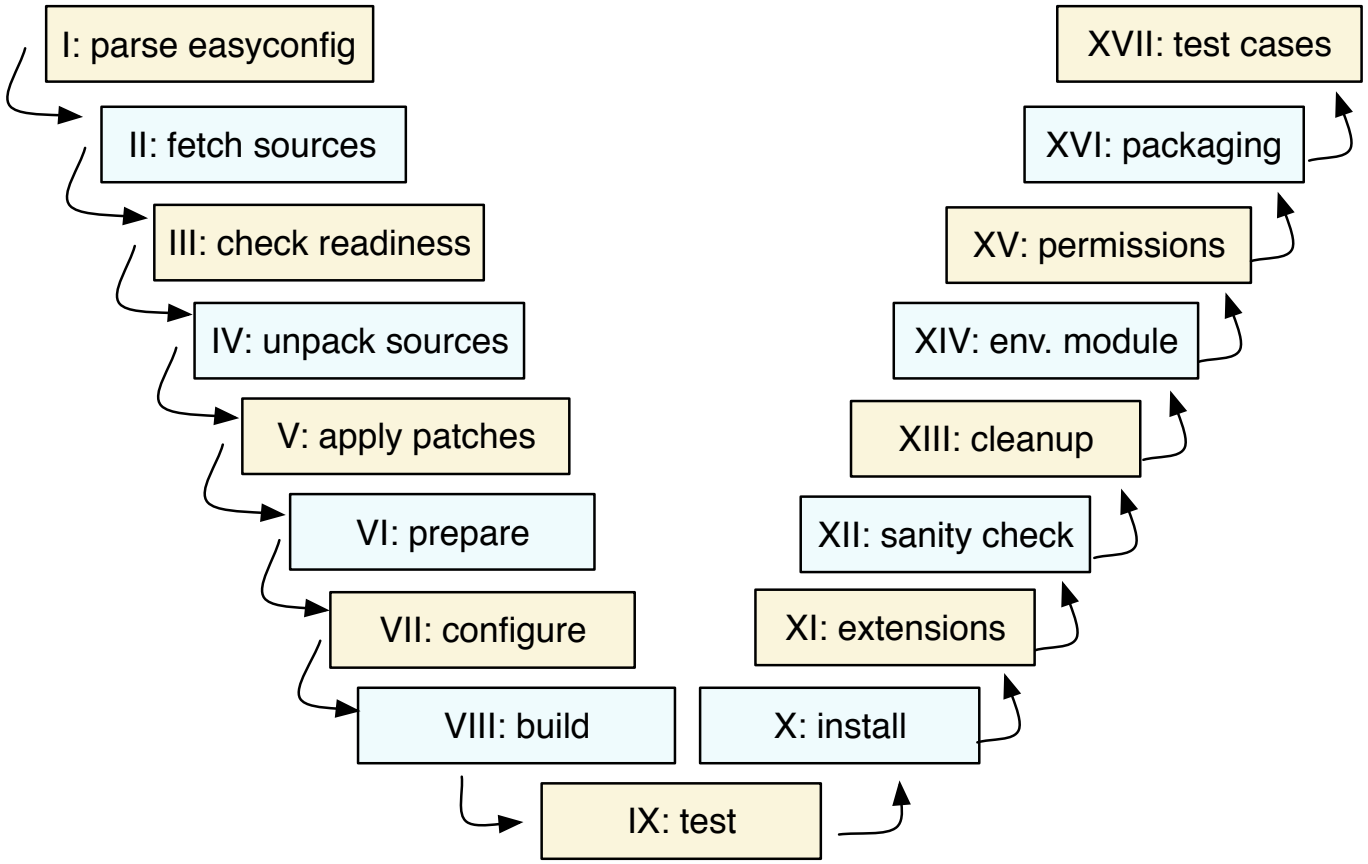
- **framework**
  - core of EasyBuild: Python modules & packages
  - provides supporting functionality for building/installing software, generating modules, ...
- **easyblock**
  - a Python module that serves as a build script, 'plugin' for the EasyBuild framework
  - implements a (generic or software-specific) build/install procedure
- **easyconfig file** (*.eb): build specification; software name/version, compiler toolchain, etc.
- **(compiler) toolchain***:* set of compilers + accompanying libraries (MPI, BLAS/LAPACK, ...)
- **extensions***:* additional packages for a particular applications (e.g., Python, R)

Installing TensorFlow from source with **one command**...


```
$ eb TensorFlow-1.13.1-foss-2018b-Python-3.6.6.eb
== temporary log file in case of crash /tmp/eb-GyvPHx/easybuild-U1TkEI.log
== processing EasyBuild easyconfig TensorFlow-1.13.1-foss-2018b-Python-3.6.6.eb
== building and installing TensorFlow/1.13.1-foss-2018b-Python-3.6.6...
== fetching files...
== creating build dir, resetting environment...
== unpacking...
== patching...
== preparing...
== configuring...
== building...
== testing...
== installing...
== taking care of extensions...
== postprocessing...
== sanity checking...
== cleaning up...
== creating module...
== permissions...
== packaging...
== COMPLETED: Installation ended successfully
== Results of the build can be found in the log file /opt/easybuild/software/Tensor...
== Build succeeded for 1 out of 1
== Temporary log file(s) /tmp/eb-GyvPHx/easybuild-U1TkEI.log* have been removed.
== Temporary directory /tmp/eb-GyvPHx has been removed.
```

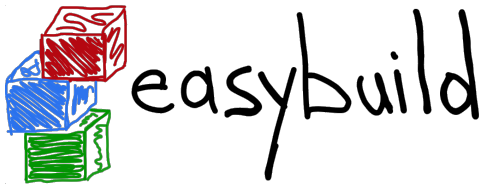
Step-wise installation procedure

EasyBuild performs a step-wise installation procedure for each software:



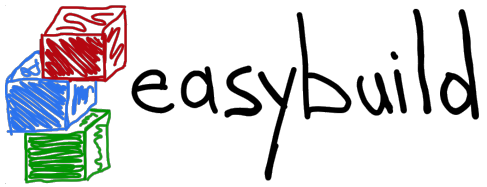
- download sources (best effort)
- set up build directory & environment
  - unpack sources (& apply patches)
  - load modules for toolchain & deps
  - define toolchain-related env vars (`$CC`, `$CFLAGS`, ...)
- configure, build, (test), install, (extensions)
- perform simple sanity check on installation
- generate environment module file

each step can be customised via easyconfig parameters or an easyblock



feature highlights (1)

- fully **autonomously** building and installing (scientific) software
  - automatic dependency resolution (via `--robot`)
  - automatic generation of environment module files (Tcl or Lua syntax)
- **no admin privileges required** (only write permission to installation target)
- thorough **logging** of executed build/install procedure
- **archiving** of easyconfigs, patches, easyblocks that were used
- highly **configurable**, via config files/environment/command line
- **dynamically extendable** with additional easyblocks, toolchains, etc.



feature highlights (2)

- support for **custom module naming schemes** (incl. hierarchical)
- **transparency** via support for 'dry run' installation & trace output
- **comprehensively tested**: lots of unit tests, frequent regression testing, ...
- actively developed, **frequent stable releases**
- **collaboration** between various HPC sites large & small
- integration with Torque/SLURM, FPM, Singularity, Docker, Cray PE, ...
- worldwide **community**

Supported software

https://easybuild.readthedocs.io/en/latest/version-specific/Supported_software.html

- latest EasyBuild (v3.9.2) supports installing **1,780 different software packages**
  - including CP2K, NAMD, NWChem, OpenFOAM, TensorFlow, WRF, ...
  - a lot of bioinformatics software is also supported out of the box
  - + >1,000 extensions: Python packages, R libraries, Perl modules, X11 libraries, ...
  - built from source when possible, optimised by host architecture by default
- diverse toolchain support:
  - compilers: GCC, Intel, Clang, PGI, IBM XL, Cray PE (GNU, Intel, CCE, PGI), CUDA
  - MPI libraries: OpenMPI, Intel MPI, MPICH, MPICH2, MVAPICH2, Cray MPI, ...
  - BLAS/LAPACK libraries: Intel MKL, OpenBLAS, ScaLAPACK, BLIS, Cray LibSci, ...

What  easybuild is *not*

- EasyBuild is ***not*** **YABT** (Yet Another Build Tool)

  it does *not* replace build tools like `cmake` or `make`; it wraps around them

- it is ***not*** **a replacement for package managers** (`yum`, `apt`, ...)

  it leverages some tools & libraries provided by the OS (glibc, OpenSSL, IB drivers, ...)

- it is ***not*** **a magic solution** to all your (software installation) problems...

  you will still run into compiler errors (unless somebody has already taken care of it)

What  is

- a **uniform interface** that wraps around software installation procedures
- a huge **time-saver**, by automating tedious/boring/repetitive tasks
- a way to provide a **consistent software stack** to your users
- an **expert system** for software installation on HPC systems
- a **platform for collaboration** with HPC sites worldwide
- a way to **empower *users* to self-manage their software stack** on HPC systems
- a tool that can be leveraged for **building *optimised* container images**

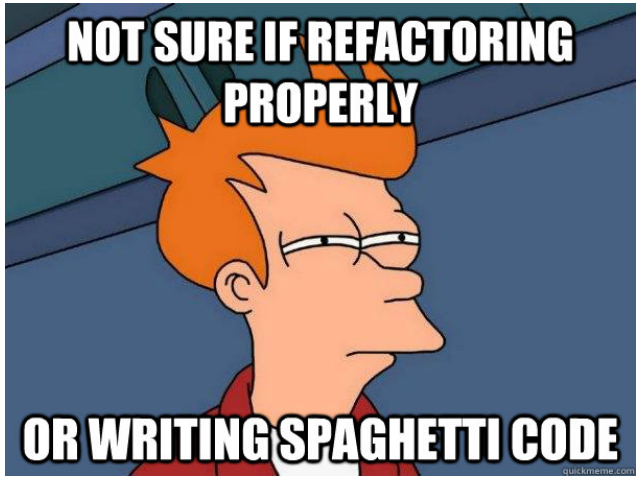
10 years of easybuild

5 eras:

- [summer 2009 - Apr'12]: in-house development at HPC-UGent
- [Apr'12 - Nov'12 (v0.x)]: public release, mailing list/Twitter/IRC, logo, first feedback
- [Nov'12- Feb'15 (v1.x)]: frequent stable releases, rise of the EasyBuild community
- [Mar'15 - Oct'16 (v2.x)]: maturing the project, spike in # supported software packages
- [Oct'16 - now (v3.x)]: support for RPATH/containers/hooks, maintainers team

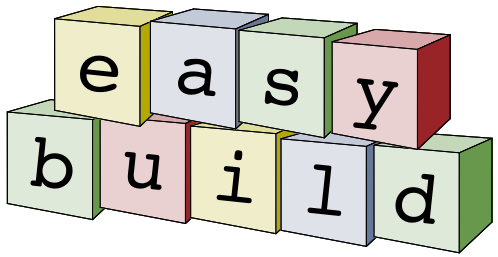
1st phase: in-house development at HPC-UGent

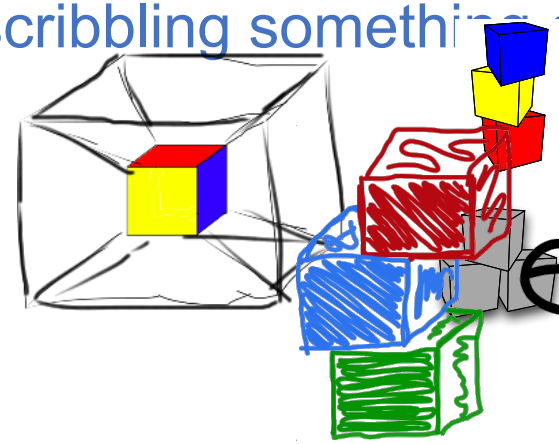
- started by Stijn De Weirdt (HPC-UGent tech lead) in summer 2009 (funny story...)
- main goal was to scratch own itch: deal with software install requests by users
- thrown into my lap towards end of 2010 to maintain & enhance
- took me a while to figure things out...
- significant contributions by summer interns
  - large refactoring efforts
  - introducing unit tests
  - implementation of various useful features like support for `eb --review-pr`
- gradually, the idea grew to make it publicly available to get some feedback...



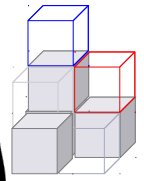
2nd phase: public release, EasyBuild 0.x

- first public release in April 2012 (paper at HEPIX'12 conference)
- initial feedback was very positive... and people actually started using it!
- changes & enhancements were made based on feedback
- we aimed to have a stable API by Supercomputing'12 (paper at PyHPC workshop)
- we felt like we needed a logo, but we weren't exactly graphical designers...
- so we ended up scribbling something on a whiteboard, and



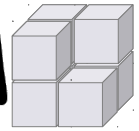


easybuild



EasyBuild

easybuild



EasyBuild

3rd phase: stable releases, rise of  community

- EasyBuild v1.0 was released Nov'12 (*during* SC12)
  - stable API & CLI (incl. 'eb' command)
  - 148 supported software packages...
- **frequent stable releases** since then (28 EasyBuild 1.x releases)
- start of (bi-weekly) EasyBuild conf calls in Nov'13
- EasyBuild hackatons all around Europe + Austin
- several *"Getting Scientific Software Installed"* BoF sessions at ISC & SC

Major developments in EasyBuild 1.x

Added support for:

- installing Python packages, R libraries, etc. as *extensions*
- configuring EasyBuild via cfg files, env. vars + CLI options
- `eb --dry-run`
- **support for using Lmod** & Tcl-only modules tool
- using a custom module naming scheme (incl. hierarchical MNS)
- **start of GitHub integration: `--from-pr` and `--upload-test-report`**
- verifying checksums for source files & patches

Using easyconfigs from a pull request

`eb --from-pr`

- no need to manually download easyconfigs to feed them to `eb`
- PR is (virtually) merged with current `develop` branch
- very useful for testing contributions!

```
$ eb --from-pr 12345
== temporary log file in case of crash /tmp/eb-kQRw7W/easybuild-HU4hce.log
== processing EasyBuild easyconfig /tmp/eb-kQRw7W/files_pr12345/s/scikit-
learn/scikit-learn-0.19.1-intel-2017b-Python-3.6.3.eb
== building and installing scikit-learn/0.19.1-intel-2017b-Python-3.6.3...
== fetching files...
== creating build dir, resetting environment...
== unpacking...
== patching...
== preparing...
== configuring...
== building...
== testing...
== installing...
```

Uploading test reports to a PR

`eb --upload-test-report`

- combined with **`--from-pr`** to test contribution and report results
- test report with info on host & environment is uploaded as a gist
- a comment is added in the PR to trigger a notification by GitHub

```
$ eb --from-pr 12345 --upload-test-report
...
== building and installing scikit-learn/0.19.1-intel-2017b-Python-3.6.3...
...
== COMPLETED: Installation ended successfully
...
Adding comment to easybuild-easyconfigs issue #12345: 'Test report by @boegel
**SUCCESS**
Build succeeded for 1 out of 1 (1 easyconfigs in this PR)
<hostname> - Linux CentOS 7.5, Intel Xeon CPU E5-2680 v3 @ 2.50GHz, Python 2.7.5
See https://gist.github.com/3c...58 for a full test report.'
== Test report uploaded to https://gist.github.com/3c...58 and mentioned in a
comment in easyconfigs PR#12345
== Build succeeded for 1 out of 1
```

Uploading test reports to a PR

```
eb --upload-test-report
```

boegel commented on 21 Nov 2017 — Owner

Test report by **@boegel**
SUCCESS
Build succeeded for 1 out of 1 (1 easyconfigs in this PR)
node2403.golett.os - Linux centos linux 7.4.1708, Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50GHz, Python 2.7.5
See https://gist.github.com/2ce867c657e05a8717978f3a73b9e019 for a full test report.

verdurin commented 3 days ago — Member

Test report by **@verdurin**
FAILED
Build succeeded for 0 out of 1 (1 easyconfigs in this PR)
rescomp2 - Linux centos 6.8, Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60GHz, Python 2.7.11
See https://gist.github.com/0dd5b4a3338a1350fe3b92b247bf84bb for a full test report.

Uploading test reports to a PR

```
eb --upload-test-report
```

EasyBuild test report for easyconfigs PR #5375

easybuild_test_report_easyconfigs_pr5375_20175921-UTC-10-59-58.md

Test report for https://github.com/easybuilders/easybuild-easyconfigs/pull/5375

Test result

Build succeeded for 1 out of 1 (1 easyconfigs in this PR)

Overview of tested easyconfigs (in order)

- **SUCCESS** *scikit-learn-0.19.1-intel-2017b-Python-3.6.3.eb*

Time info

- start: Tue, 21 Nov 2017 10:57:11 +0000 (UTC)
- end: Tue, 21 Nov 2017 10:59:58 +0000 (UTC)

Uploading test reports to a PR

```
eb --upload-test-report
```

EasyBuild test report for easyconfigs PR #5719

easybuild_test_report_easyconfigs_pr5719_20180124-UTC-18-01-27.md

Test report for https://github.com/easybuilders/easybuild-easyconfigs/pull/5719

Test result

Build succeeded for 0 out of 1 (1 easyconfigs in this PR)

Overview of tested easyconfigs (in order)

- **FAIL (build issue)** *RSEM-1.3.0-intel-2017b.eb* (partial log available at https://gist.github.com/7d9e48b400232b5fa0924e2bf28d240a)

a separate gist provides partial build log

Time info

- start: Wed, 24 Jan 2018 18:01:18 +0000 (UTC)

4th phase: maturing the project

- EasyBuild v2.0 was released in Mar'15
  - `vsc-base` as a proper dependency, Python 2.4 no longer supported,
    no more automatic fallback to `ConfigureMake` easyblock (was too confusing)
- **fast growth in supported software packages** (581 to 1,110 in just 1.5 years)
- **integration with Cray PE** (implemented together with Petar Forai)
- significant performance improvements
- switching from self-hosted Jenkins to Travis CI
- **common compiler toolchain (foss, intel) starting to catch on**
- emergence of Spack

Integration with Cray Programming Environment

https://easybuild.readthedocs.io/en/latest/Cray-support.html

- initial (experimental) integration with Cray PE implemented April'15 (v2.1.0)
- declared stable in Mar'16 (v2.7.0)
- custom `Cray*` toolchains leverage Cray-provided modules:

  `PrgEnv/*`, `cray-libsci`, `cray-mpich`, ...
- `--optarch` controls which `craype-*` module is used in build environment
- leveraging of Cray-provided "external" modules via provided metadata
- careful handling of Cray-provided modules in build environment & toolchains
- **used by CSCS since 2015 on Piz Daint & other Cray systems**

Community (common) toolchains

http://easybuild.readthedocs.io/en/latest/Common-toolchains.html

- **`intel` and `foss`[1] toolchains** are most commonly used in EasyBuild community
- helps to focus efforts of HPC sites using one or both of these toolchains
- updated twice a year, clear versioning scheme: `<year>{a,b}` (`2017b`, `2018a`, ...)
- latest version:
  - `foss/2019a`

    binutils 2.31.1, GCC 8.2, OpenMPI 3.1.3,

    OpenBLAS 0.3.5, FFTW 3.3.8
  - `intel/2019a`

    binutils 2.31.1 + GCC 8.2 as base

    Intel compilers 2019.1.144, Intel MPI 2018.4.274, Intel MKL 2019.1.144

(1) FOSS: Free and Open Source Software

Major developments in EasyBuild 2.x

Added support for:

- only (re-)installing module file via `eb --module-only`
- generating module files in Lua syntax
- easily extending EasyBuild via `--include-easyblocks` & co
- leveraging 'external' modules (which were not installed using EasyBuild)
- **quickly inspecting installation procedure via `eb -x`**
- `eb --install-latest-eb-release`

- significantly enhanced GitHub integration:
  - **creating/updating pull requests via `eb --new-pr`, `eb --update-pr`**
  - `eb --review-pr` (took a while to this feature merged...)

Transparency of install procedure to be performed

https://easybuild.readthedocs.io/en/latest/Extended_dry_run.html

- **`eb --extended-dry-run`** (or '**`eb -x`**') reveals planned installation procedure
- *runs in a matter of seconds*
- shows commands that will be executed, build environment, generated module file, ...
- any errors that occur in used easyblock are ignored (but clearly reported)
- not 100% accurate since easyblock may require certain files to be present, etc.
- very useful when debugging easyblocks, instant feedback as a first pass
- implementation motivated by requests from the community
- helps to avoid impression that EasyBuild is a magic black box for installing software

Example output of --extended-dry-run (1/3)


```
$ eb WRF-3.8.0-intel-2016b-dmpar.eb -x

== temporary log file in case of crash /tmp/eb-Dh1wOp/easybuild-0bu9u9.log

== processing EasyBuild easyconfig /home/example/eb/easybuild-easyconfigs/easybuild/
easyconfigs/w/WRF/WRF-3.8.0-intel-2016b-dmpar.eb

...

*** DRY RUN using 'EB_WRF' easyblock (easybuild.easyblocks.wrf @ /home/example/eb/easybuild-
easyblocks/easybuild/easyblocks/w/wrf.py) ***

== building and installing WRF/3.8.0-intel-2016b-dmpar...
fetching files... [DRY RUN]

[fetch_step method]
Available download URLs for sources/patches:
  * http://www2.mmm.ucar.edu/wrf/src//$source
  * http://www.mmm.ucar.edu/wrf/src//$source

List of sources:
  * WRFV3.8.0.TAR.gz will be downloaded to /home/example/eb/sources/w/WRF/WRFV3.8.0.TAR.gz
```

Example output of `--extended-dry-run` (2/3)


```
$ eb WRF-3.8.0-intel-2016b-dmpar.eb -x
...

building... [DRY RUN]

[build_step method]
  running command "tcsh ./compile -j 4 wrf"
  (in /home/example/eb/software/WRF/3.8.0-intel-2016b-dmpar/WRF-3.8.0)
  running command "tcsh ./compile -j 4 em_real"
  (in /home/example/eb/software/WRF/3.8.0-intel-2016b-dmpar/WRF-3.8.0)
  running command "tcsh ./compile -j 4 em_b_wave"
  (in /home/example/eb/software/WRF/3.8.0-intel-2016b-dmpar/WRF-3.8.0)
...
[sanity_check_step method]
Sanity check paths - file ['files']
  * WRFV3/main/libwrflib.a
  * WRFV3/main/real.exe
  * WRFV3/main/wrf.exe
Sanity check paths - (non-empty) directory ['dirs']
  * WRFV3/main
  * WRFV3/run
Sanity check commands
  (none)
```

Example output of `--extended-dry-run` (3/3)

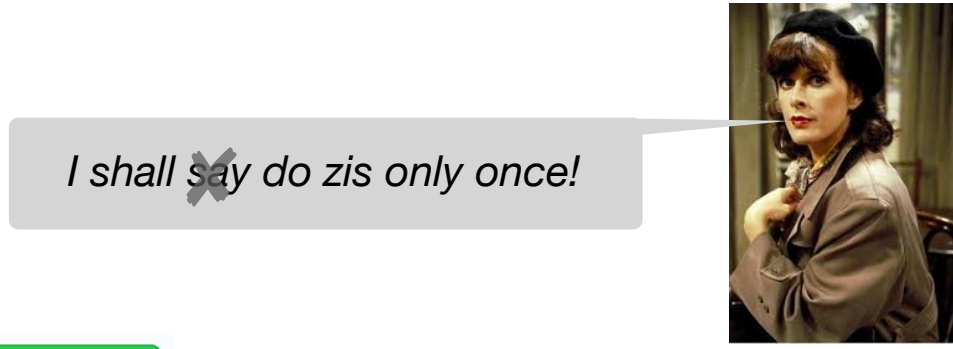

```
$ eb WRF-3.8.0-intel-2016b-dmpar.eb -x
...

[make_module_step method]
Generating module file /home/example/eb/modules/all/WRF/3.8.0-intel-2016b-dmpar,
with contents:

    #%Module
    proc ModulesHelp { } {
        puts stderr { The Weather Research and Forecasting (WRF) Model }
    }
    module-whatis {Description: WRF - Homepage: http://www.wrf-model.org}

    set root /home/example/eb/software/WRF/3.8.0-intel-2016b-dmpar

    conflict WRF

    if { ![ is-loaded intel/2016b ] } {
        module load intel/2016b
    }
    if { ![ is-loaded JasPer/1.900.1-intel-2016b ] } {
        module load JasPer/1.900.1-intel-2016b
    }
```

git can be a major hurdle for contributors

- Potential contributors may not be familiar (yet) with git...
- Do they ***have*** to learn git first before they can contribute?

So you want to contribute to easybuild ...

To contribute one or more easyconfig files,
you first need to prepare:



1) create a GitHub account via Sign up for GitHub

2) Fork https://github.com/easybuilders/easybuild-easyconfigs

3)
```
git clone git@github.com:YOU/easybuild-easyconfigs.git
```
(change this bit!)

Branch, stage, commit, push

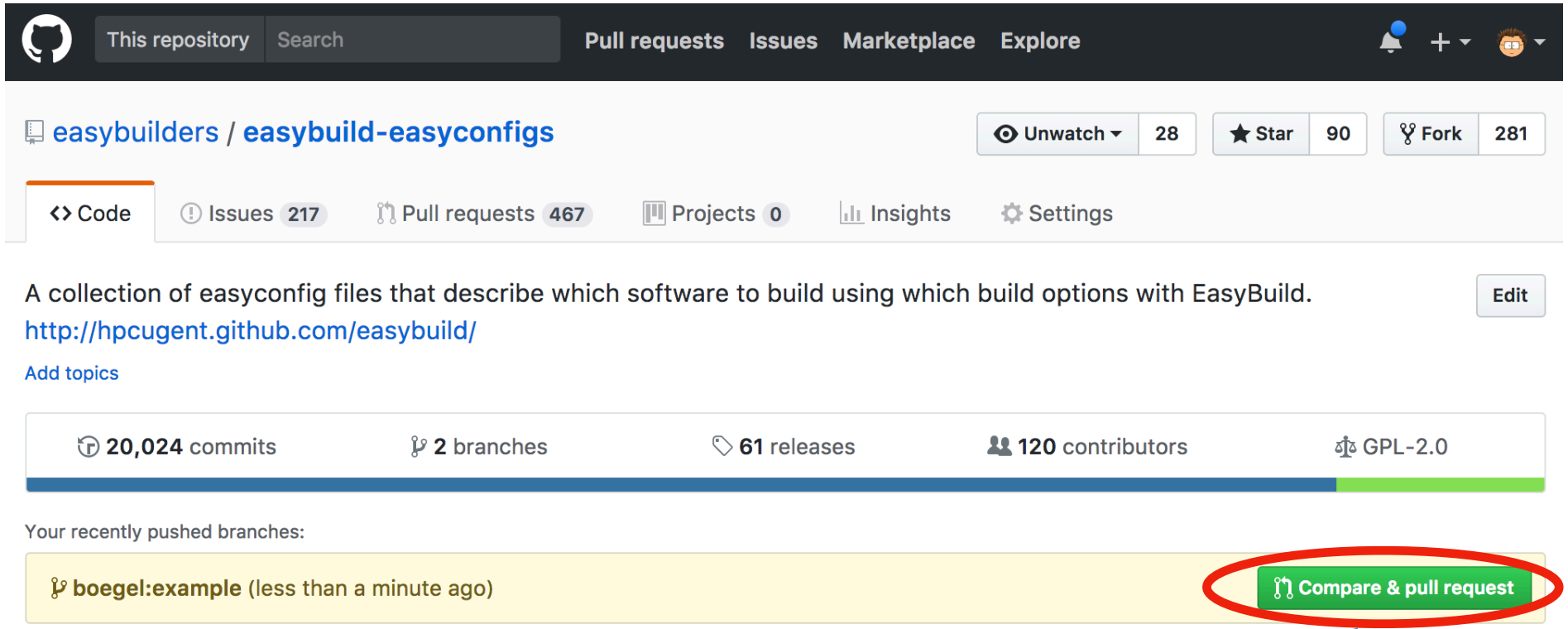
Once you have cloned the easyconfigs repository, you can really get to work...

```
# create and check out a new branch
$ git checkout -b funnybranchname

# stage the new/changed easyconfig file(s)
$ git add easybuild/easyconfigs/s/scikit-learn/*.eb

# commit the changes with a sensible commit message
$ git commit -m "update scikit-learn to 0.19.1"

# push your branch to your fork on GitHub
$ git push origin funnybranchname
```

(this may be different in your setup, see `git remote -v`)

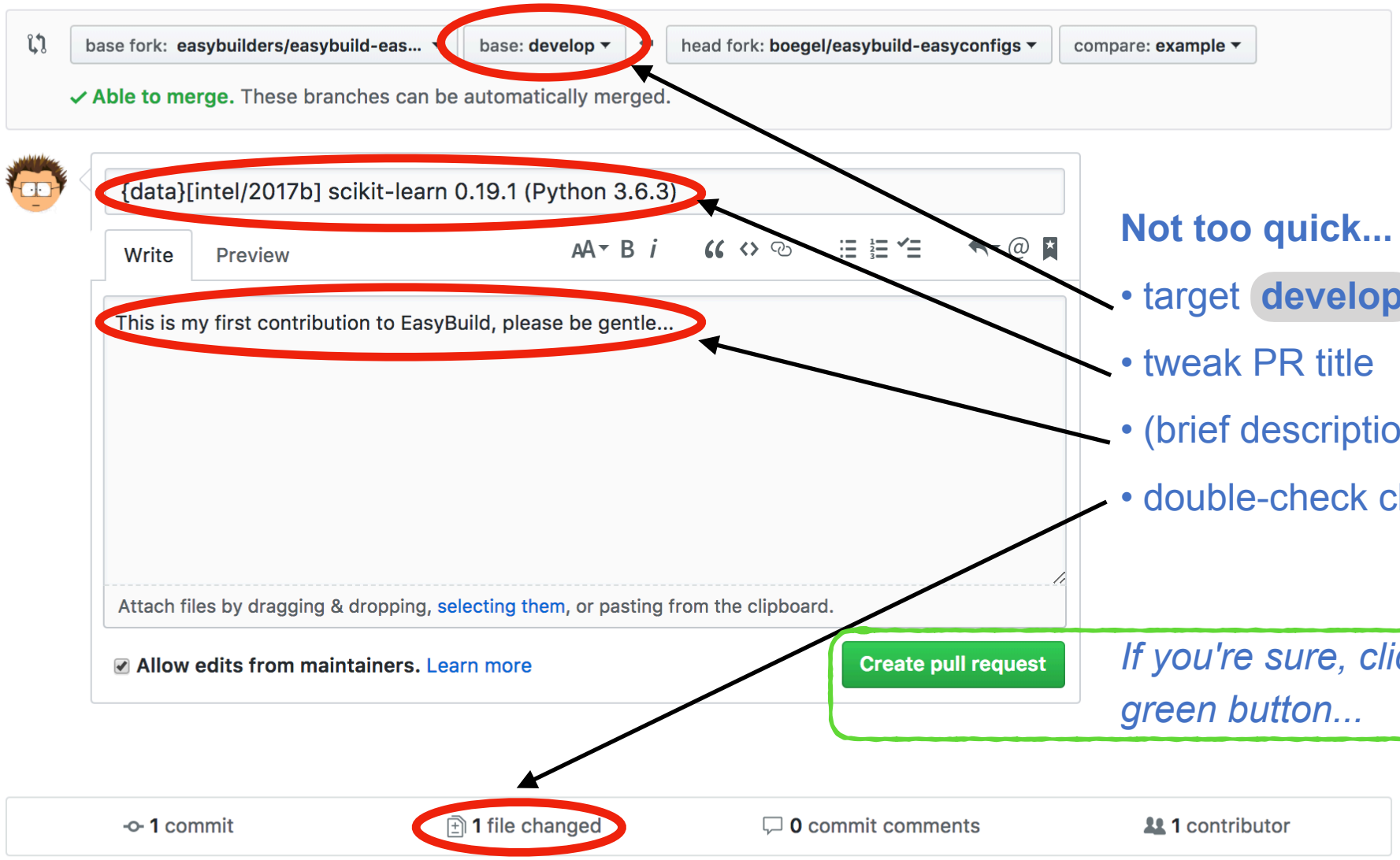
Clickety, clickety, click.

Next, log on to GitHub and open a pull request...



You can use this useful button , if you're quick enough!

Almost there...



Not too quick...

- target **develop** branch
- tweak PR title
- (brief description)
- double-check changed files

If you're sure, click the shiny green button...

Are we done yet?

- once your pull request is open, your changes will be:
  - quickly tested by Travis CI (automagically)
  - reviewed by a maintainer
  - thoroughly tested by at least one person
- additional changes may be requested...
- which means you may have to go back to `git`...
- maybe even multiple times... (spread over multiple hours/days)

Fine-tuning your contribution

Do you still remember the funny branch name you used for that PR?

```
# check out your branch again
$ git checkout funnybranchname

# edit some files, as requested by the friendly reviewer
$ vim ...

# commit (all) your changes
$ git commit -am "fixed suggestions by reviewer"

# push your updated branch to GitHub to update the PR
$ git push origin funnybranchname
```

Too much overhead, manual steps...

"Isn't there an easier/more efficient way for doing this?!"

- need to come up with a sensible funny branch name first...
- 4 git commands to prepare the contribution
- at least 4 (well aimed) *mouse clicks* to open the pull request
- if requested, N x 3 more git commands for fine-tuning the PR
- + a bit of cleanup afterwards
  (delete branch both in local repository clone and on GitHub)

All of this assuming you:

- take into account the project policies
- don't mess up with `git` at any time...

Opening new pull requests

`eb --new-pr`

```
$ eb --new-pr sklearn.eb
== temporary log file in case of crash /tmp/eb-Iv07fs/easybuild-inS8Ny.log
== fetching branch 'develop' from
   https://github.com/easybuilders/easybuild-easyconfigs.git...
== copying easyconfigs to
   /tmp/eb-Iv07fs/git-working-dirhYsAA_/easybuild-easyconfigs...

Opening pull request
* target: easybuilders/easybuild-easyconfigs:develop
* from: boegel/easybuild-easyconfigs:20180125211924_new_pr_scikit-learn0191
* title: "{data}[intel/2017b] scikit-learn v0.19.1"
* description:
"""
(created using `eb --new-pr`)

"""
* overview of changes:
 .../s/scikit-learn/scikit-learn-0.19.1-intel-2017b-Python-3.6.3.eb | 31 ++++++++++++
 1 file changed, 31 insertions(+)

Opened pull request: https://github.com/easybuilders/easybuild-easyconfigs/pull/12345
== Temporary log file(s) /tmp/eb-Iv07fs/easybuild-inS8Ny.log* have been removed.
== Temporary directory /tmp/eb-Iv07fs has been removed.
```

one single 'eb' command
no 'git' commands, no clickety on GitHub

Opening new pull requests

`eb --new-pr`

```
$ eb --new-pr sklearn.eb
== temporary log file in case of crash /tmp/eb-Iv07fs/easybuild-inS8Ny.log
== fetching branch 'develop' from
   https://github.com/easybuilders/easybuild-easyconfigs.git...
== copying easyconfigs to
   /tmp/eb-Iv07fs/git-working-dirhYsAA_/easybuild-easyconfigs...

Opening pull request
* target: easybuilders/easybuild-easyconfigs:develop
* from: boegel/easybuild-easyconfigs:20180125211924_new_pr_scikit-learn0191
* title: "{data}[intel/2017b] scikit-learn v0.19.1"
* description:
"""
(created using `eb --new-pr`)

"""
* overview of changes:
 .../s/scikit-learn/scikit-learn-0.19.1-intel-2017b-Python-3.6.3.eb | 31 ++++++++++++
 1 file changed, 31 insertions(+)

Opened pull request: https://github.com/easybuilders/easybuild-easyconfigs/pull/12345
== Temporary log file(s) /tmp/eb-Iv07fs/easybuild-inS8Ny.log* have been removed.
== Temporary directory /tmp/eb-Iv07fs has been removed.
```

repository is ***cloned*** to temporary directory,
develop branch is ***checked out*** as a base

Opening new pull requests

`eb --new-pr`

```
$ eb --new-pr sklearn.eb
== temporary log file in case of crash /tmp/eb-Iv07fs/easybuild-inS8Ny.log
== fetching branch 'develop' from
   https://github.com/easybuilders/easybuild-easyconfigs.git...
== copying easyconfigs to
   /tmp/eb-Iv07fs/git-working-dirhYsAA_/easybuild-easyconfigs...

Opening pull request
* target: easybuilders/easybuild-easyconfigs:develop
* from: boegel/easybuild-easyconfigs:20180125211924_new_pr_scikit-learn0191
* title: "{data}[intel/2017b] scikit-learn v0.19.1"
* description:
"""
(created using `eb --new-pr`)

"""
* overview of changes:
 .../s/scikit-learn/scikit-learn-0.19.1-intel-2017b-Python-3.6.3.eb | 31 ++++++++++++
 1 file changed, 31 insertions(+)

Opened pull request: https://github.com/easybuilders/easybuild-easyconfigs/pull/12345
== Temporary log file(s) /tmp/eb-Iv07fs/easybuild-inS8Ny.log* have been removed.
== Temporary directory /tmp/eb-Iv07fs has been removed.
```

auto-derived branch name

GitHub repository to push to

a new branch is created & pushed
(you can specify the branch name yourself)

Opening new pull requests

`eb --new-pr`

```
$ eb --new-pr sklearn.eb
== temporary log file in case of crash /tmp/eb-Iv07fs/easybuild-inS8Ny.log
== fetching branch 'develop' from
   https://github.com/easybuilders/easybuild-easyconfigs.git...
== copying easyconfigs to
   /tmp/eb-Iv07fs/git-working-dirhYsAA_/easybuild-easyconfigs...

Opening pull request
* target: easybuilders/easybuild-easyconfigs:develop
* from: boegel/easybuild-easyconfigs:20180125211924_new_pr_scikit-learn0191
* title: "{data}[intel/2017b] scikit-learn v0.19.1"
* description:
"""
(created using `eb --new-pr`)

"""
* overview of changes:
 .../s/scikit-learn/scikit-learn-0.19.1-intel-2017b-Python-3.6.3.eb | 31 ++++++++++++
 1 file changed, 31 insertions(+)

Opened pull request: https://github.com/easybuilders/easybuild-easyconfigs/pull/12345
== Temporary log file(s) /tmp/eb-Iv07fs/easybuild-inS8Ny.log* have been removed.
== Temporary directory /tmp/eb-Iv07fs has been removed.
```

the easyconfig file is ***renamed***
and ***copied to the right location***

Opening new pull requests

`eb --new-pr`

```
$ eb --new-pr sklearn.eb
== temporary log file in case of crash /tmp/eb-Iv07fs/easybuild-inS8Ny.log
== fetching branch 'develop' from
   https://github.com/easybuilders/easybuild-easyconfigs.git...
== copying easyconfigs to
   /tmp/eb-Iv07fs/git-working-dirhYsAA_/easybuild-easyconfigs...

Opening pull request
* target: easybuilders/easybuild-easyconfigs:develop
* from: boegel/easybuild-easyconfigs:20180125211924_new_pr_scikit-learn0191
* title: "{data}[intel/2017b] scikit-learn v0.19.1"
* description:
"""
(created using `eb --new-pr`)

"""
* overview of changes:
 .../s/scikit-learn/scikit-learn-0.19.1-intel-2017b-Python-3.6.3.eb | 31 ++++++++++++
 1 file changed, 31 insertions(+)

Opened pull request: https://github.com/easybuilders/easybuild-easyconfigs/pull/12345
== Temporary log file(s) /tmp/eb-Iv07fs/easybuild-inS8Ny.log* have been removed.
== Temporary directory /tmp/eb-Iv07fs has been removed.
```

the pull request is prepared:
target develop branch, ***title & description***

Opening new pull requests

`eb --new-pr`

```
$ eb --new-pr sklearn.eb
== temporary log file in case of crash /tmp/eb-Iv07fs/easybuild-inS8Ny.log
== fetching branch 'develop' from
   https://github.com/easybuilders/easybuild-easyconfigs.git...
== copying easyconfigs to
   /tmp/eb-Iv07fs/git-working-dirhYsAA_/easybuild-easyconfigs...

Opening pull request
* target: easybuilders/easybuild-easyconfigs:develop
* from: boegel/easybuild-easyconfigs:20180125211924_new_pr_scikit-learn0191
* title: "{data}[intel/2017b] scikit-learn v0.19.1"
* description:
"""
(created using `eb --new-pr`)

"""
* overview of changes:
 .../s/scikit-learn/scikit-learn-0.19.1-intel-2017b-Python-3.6.3.eb | 31 ++++++++++
 1 file changed, 31 insertions(+)

Opened pull request: https://github.com/easybuilders/easybuild-easyconfigs/pull/12345
== Temporary log file(s) /tmp/eb-Iv07fs/easybuild-inS8Ny.log* have been removed.
== Temporary directory /tmp/eb-Iv07fs has been removed.
```

a high-level ***overview of the changes*** is shown (similar to output of '`git stat`')

Opening new pull requests

`eb --new-pr`

```
$ eb --new-pr sklearn.eb
== temporary log file in case of crash /tmp/eb-Iv07fs/easybuild-inS8Ny.log
== fetching branch 'develop' from
   https://github.com/easybuilders/easybuild-easyconfigs.git...
== copying easyconfigs to
   /tmp/eb-Iv07fs/git-working-dirhYsAA_/easybuild-easyconfigs...

Opening pull request
* target: easybuilders/easybuild-easyconfigs:develop
* from: boegel/easybuild-easyconfigs:20180125211924_new_pr_scikit-learn0191
* title: "{data}[intel/2017b] scikit-learn v0.19.1"
* description:
"""
(created using `eb --new-pr`)

"""
* overview of changes:
 .../s/scikit-learn/scikit-learn-0.19.1-intel-2017b-Python-3.6.3.eb | 31 ++++++++++++
 1 file changed, 31 insertions(+)

Opened pull request: https://github.com/easybuilders/easybuild-easyconfigs/pull/12345
== Temporary log file(s) /tmp/eb-Iv07fs/easybuild-inS8Ny.log* have been removed.
== Temporary directory /tmp/eb-Iv07fs has been removed.
```

the ***pull request is opened automatically***, using the GitHub API & token

Opening a pull request is now... easy!


```
$ mv sklearn.eb scikit-learn-0.19.1-intel-2017b-Python-3.6.3.eb
$ mv scikit*.eb easybuild/easyconfigs/s/scikit-learn
$ git checkout develop && git pull upstream develop
$ git checkout -b scikit_learn_0191_intel_2017b
$ git add easybuild/easyconfigs/s/scikit-learn
$ git commit -m "{data}[intel/2017b] scikit-learn v0.19.1"
$ git push origin scikit_learn_0191_intel_2017b
```

+ log into GitHub to actually open the pull request (clickety, clickety...)

one single `eb` command

no `git` commands

no GitHub interaction

metadata is automatically derived from easyconfig

saves a lot of time!

```
eb --new-pr sklearn.eb
```

Updating existing pull requests

`eb --update-pr`

to update a PR: provide PR#, updated files and sensible commit message

```
$ eb --update-pr 12345 sklearn.eb --pr-commit-msg "minor style fixes"
== temporary log file in case of crash /tmp/eb-EKe1WF/easybuild-w1Ycat.log
== Determined branch name corresponding to easybuilders/easybuild-easyconfigs
PR #12345: 20180125211924_new_pr_scikit-learn0191
== fetching branch '20180125211924_new_pr_scikit-learn0191' from https://
github.com/boegel/easybuild-easyconfigs.git...
== copying easyconfigs to /tmp/eb-EKe1WF/git-working-dirbIzGHn/easybuild-
easyconfigs...
Overview of changes:
 .../s/scikit-learn/scikit-learn-0.19.1-intel-2017b-Python-3.6.3.eb | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

Updated easybuilders/easybuild-easyconfigs PR #12345 by pushing to branch
boegel/20180125211924_new_pr_scikit-learn0191
== Temporary log file(s) /tmp/eb-EKe1WF/easybuild-w1Ycat.log* have been removed.
== Temporary directory /tmp/eb-EKe1WF has been removed.
```

Updating existing pull requests

`eb --update-pr`

branch that corresponds with PR is determined *automatically*

```
$ eb --update-pr 12345 sklearn.eb --pr-commit-msg "minor style fixes"
== temporary log file in case of crash /tmp/eb-EKe1WF/easybuild-w1Ycat.log
== Determined branch name corresponding to easybuilders/easybuild-easyconfigs
PR #12345: 20180125211924_new_pr_scikit-learn0191
== fetching branch '20180125211924_new_pr_scikit-learn0191' from https://
github.com/boegel/easybuild-easyconfigs.git...
== copying easyconfigs to /tmp/eb-EKe1WF/git-working-dirbIzGHn/easybuild-
easyconfigs...
Overview of changes:
 .../s/scikit-learn/scikit-learn-0.19.1-intel-2017b-Python-3.6.3.eb | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

Updated easybuilders/easybuild-easyconfigs PR #12345 by pushing to branch
boegel/20180125211924_new_pr_scikit-learn0191
== Temporary log file(s) /tmp/eb-EKe1WF/easybuild-w1Ycat.log* have been removed.
== Temporary directory /tmp/eb-EKe1WF has been removed.
```

Updating existing pull requests

`eb --update-pr`

PR branch is downloaded to temporary directory,
updated files are copied

```
$ eb --update-pr 12345 sklearn.eb --pr-commit-msg "minor style fixes"
== temporary log file in case of crash /tmp/eb-EKe1WF/easybuild-w1Ycat.log
== Determined branch name corresponding to easybuilders/easybuild-easyconfigs
PR #12345: 20180125211924_new_pr_scikit-learn0191
== fetching branch '20180125211924_new_pr_scikit-learn0191' from https://
github.com/boegel/easybuild-easyconfigs.git...
== copying easyconfigs to /tmp/eb-EKe1WF/git-working-dirbIzGHn/easybuild-
easyconfigs...
Overview of changes:
 .../s/scikit-learn/scikit-learn-0.19.1-intel-2017b-Python-3.6.3.eb | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

Updated easybuilders/easybuild-easyconfigs PR #12345 by pushing to branch
boegel/20180125211924_new_pr_scikit-learn0191
== Temporary log file(s) /tmp/eb-EKe1WF/easybuild-w1Ycat.log* have been removed.
== Temporary directory /tmp/eb-EKe1WF has been removed.
```

Updating existing pull requests

`eb --update-pr`

updated files are renamed and copied to right location

```
$ eb --update-pr 12345 sklearn.eb --pr-commit-msg "minor style fixes"
== temporary log file in case of crash /tmp/eb-EKe1WF/easybuild-w1Ycat.log
== Determined branch name corresponding to easybuilders/easybuild-easyconfigs
PR #12345: 20180125211924_new_pr_scikit-learn0191
== fetching branch '20180125211924_new_pr_scikit-learn0191' from https://
github.com/boegel/easybuild-easyconfigs.git...
== copying easyconfigs to /tmp/eb-EKe1WF/git-working-dirbIzGHn/easybuild-
easyconfigs...
Overview of changes:
 .../s/scikit-learn/scikit-learn-0.19.1-intel-2017b-Python-3.6.3.eb  2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

Updated easybuilders/easybuild-easyconfigs PR #12345 by pushing to branch
boegel/20180125211924_new_pr_scikit-learn0191
== Temporary log file(s) /tmp/eb-EKe1WF/easybuild-w1Ycat.log* have been removed.
== Temporary directory /tmp/eb-EKe1WF has been removed.
```

Updating existing pull requests

`eb --update-pr`

changes are committed, high-level overview is shown (cfr. '`git stat`')

```
$ eb --update-pr 12345 sklearn.eb --pr-commit-msg "minor style fixes"
== temporary log file in case of crash /tmp/eb-EKe1WF/easybuild-w1Ycat.log
== Determined branch name corresponding to easybuilders/easybuild-easyconfigs
PR #12345: 20180125211924_new_pr_scikit-learn0191
== fetching branch '20180125211924_new_pr_scikit-learn0191' from https://
github.com/boegel/easybuild-easyconfigs.git...
== copying easyconfigs to /tmp/eb-EKe1WF/git-working-dirbIzGHn/easybuild-
easyconfigs...
Overview of changes:
 .../s/scikit-learn/scikit-learn-0.19.1-intel-2017b-Python-3.6.3.eb | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

Updated easybuilders/easybuild-easyconfigs PR #12345 by pushing to branch
boegel/20180125211924_new_pr_scikit-learn0191
== Temporary log file(s) /tmp/eb-EKe1WF/easybuild-w1Ycat.log* have been removed.
== Temporary directory /tmp/eb-EKe1WF has been removed.
```

Updating existing pull requests

`eb --update-pr`

modified branch is pushed to GitHub to update the PR

```
$ eb --update-pr 12345 sklearn.eb --pr-commit-msg "minor style fixes"
== temporary log file in case of crash /tmp/eb-EKe1WF/easybuild-w1Ycat.log
== Determined branch name corresponding to easybuilders/easybuild-easyconfigs
PR #12345: 20180125211924_new_pr_scikit-learn0191
== fetching branch '20180125211924_new_pr_scikit-learn0191' from https://
github.com/boegel/easybuild-easyconfigs.git...
== copying easyconfigs to /tmp/eb-EKe1WF/git-working-dirbIzGHn/easybuild-
easyconfigs...
Overview of changes:
 .../s/scikit-learn/scikit-learn-0.19.1-intel-2017b-Python-3.6.3.eb | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

Updated easybuilders/easybuild-easyconfigs PR #12345 by pushing to branch
boegel/20180125211924_new_pr_scikit-learn0191
== Temporary log file(s) /tmp/eb-EKe1WF/easybuild-w1Ycat.log* have been removed.
== Temporary directory /tmp/eb-EKe1WF has been removed.
```

Updating a PR in 1, ~~2, 3~~

First, figure out the (funny) branch name corresponding to the PR...

```
$ git checkout scikit_learn_0191_intel_2017b
$ cd easybuild/easyconfigs/s/scikit-learn
$ vim scikit-learn-0.19.1-intel-2017b-Python-3.6.3.eb
$ git add scikit*.eb
$ git commit -m "fix typo"
$ git push origin scikit_learn_0191_intel_2017b
```

one single `eb` command

no `git` commands

no GitHub interaction

no local branch to clean up once PR is merged

saves a lot of time!

```
eb --update-pr 12345 sklearn.eb --pr-commit-msg "fix typo"
```

5th phase: extending functionality + team of maintainers

- EasyBuild v3.0 was released in Nov'16
  - Lmod + Lua syntax by default
  - subtoolchain-aware `--robot`
  - stable GitHub integration
- **yearly EasyBuild User Meetings**
- move from github.com/hpcugent to github.com/easybuilders
- switch to SHA256 checksums (now required in easyconfig pull requests)
- **team of maintainers** + maintainer-of-the-week rotating (double) role
- creating of slack channel + YouTube channel *(https://tiny.cc/easybuild_youtube)*

Yearly easybuild User Meetings

group picture
4th EasyBuild User Meeting
@ UC Louvain (Feb 2019)

- EasyBuild community has been growing rapidly the last couple of years
- used by **hundreds of HPC sites and companies worldwide, incl. JSC, CSCS, SURFsara, Pfizer, ...**
- very welcoming & supportive to newcomers

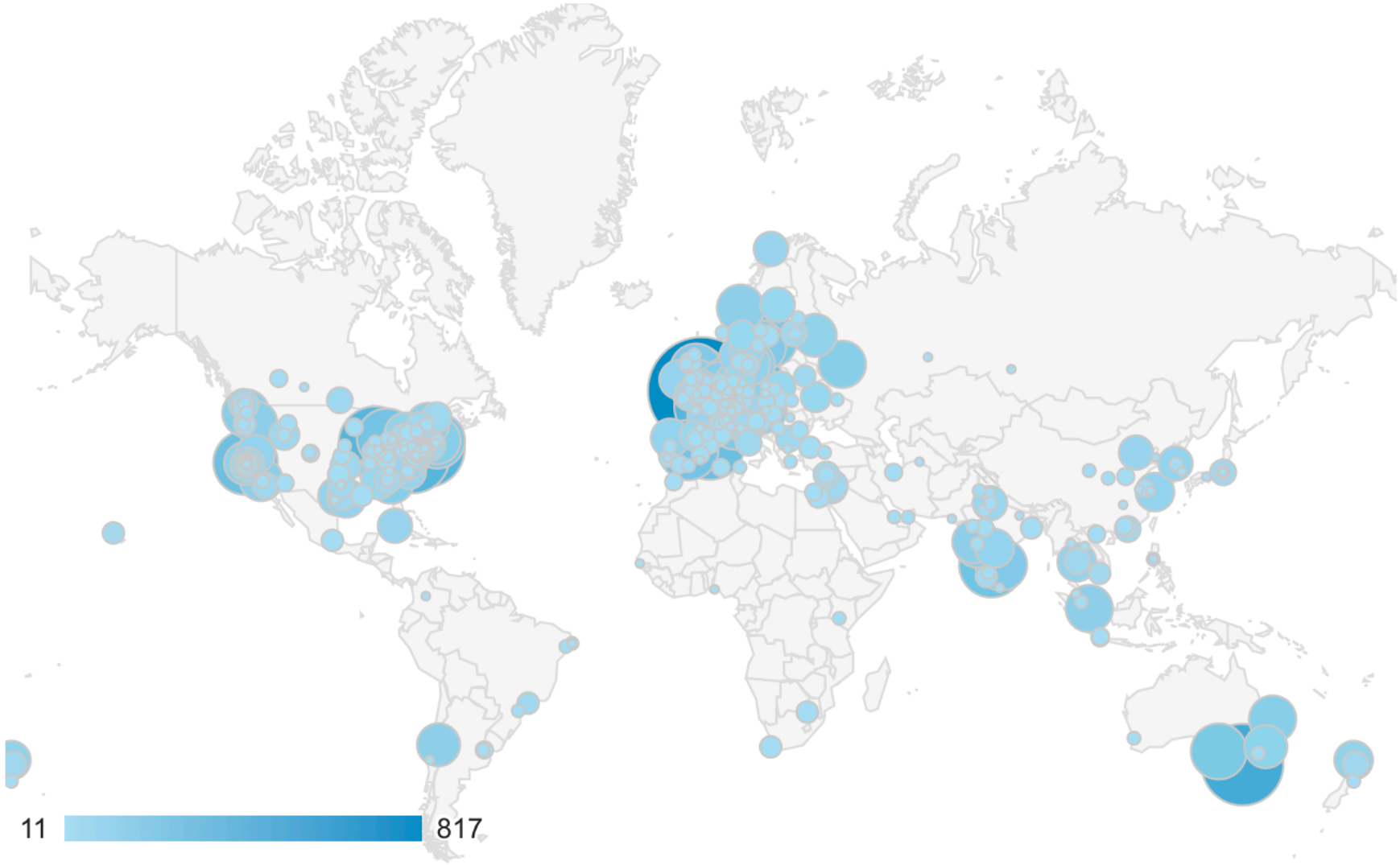
mailing list: https://lists.ugent.be/wws/info/easybuild
Slack channel: easybuild.slack.com
IRC: #easybuild on FreeNode

EasyBuild documentation visits

EasyBuild documentation at https://easybuild.readthedocs.io hitting **~500 weekly visitors**

(source: Google Analytics)

The sun never sets on EasyBuild...



cities from where https://easybuild.readthedocs.io was visited at least 10 times during the last year

(source: Google Analytics)

Maintainers, maintainers, maintainers, ...

easybuild maintainers

Damian Alvarez - `@damianam`
(Jülich Supercomputing Centre)

Miguel Dias Costa - `@migueldiascosta`
(National University of Singapore)

Pablo Escobar - `@pescobar`
(sciCORE, University of Basel)

Kenneth Hoste - `@boegel`
(HPC-UGent)

Adam Huffman - `@verdurin`
(Big Data Institute, University of Oxford)

Samuel Moors - `@smoors`
(Free University of Brussels)

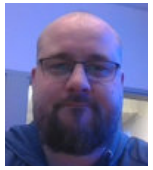
Alan O'Cais - `@ocaisa`
(Jülich Supercomputing Centre)

Bart Oldeman - `@bartoldeman`
(ComputeCanada)

Ward Poelmans - `@wpoely86`
(Free University of Brussels)

Åke Sandgren - `@akesandgren`
(Umeå University, Sweden)

Davide Vanzo - `@vanzod`
(Vanderbilt University)



Mikael Öhman - `@micketeer`
(Chalmers University of Technology)

easybuild slack (+ IRC) channel

- `#general` on *easybuild.slack.com* is linked with `#easybuild` on IRC (freenode)
- ~130 members on Slack channel (+ about 15 (also) on IRC)
- steady growth in activity: 30-40 weekly active users
- **self-invite via *https://easybuild-slack.herokuapp.com* !**

Major developments in EasyBuild 3.x

Added support for:

- **tracing installation procedure via `eb --trace`**
- adding (SHA256) checksums via `eb --inject-checksums`
- previewing pull requests via `eb --preview-pr`
- only downloading sources via `eb --fetch`
- **hooks to customise EasyBuild in a maintainable way**
- distributing installations by submitting them (directly) to Slurm via `--job`
- **generating Singularity/Docker container recipes/images** (experimental)
- reporting missing dependencies via `eb --missing`

Support for tracing the installation procedure

https://easybuild.readthedocs.io/en/latest/Tracing_progress.html

`eb --trace` shows more details while EasyBuild is performing installation(s)

- exact location of build and install directories
- list of source files & patches
- modules being loaded
- commands being executed
- pointer to temporary log file with output of commands
- timing information for each command (start time + how long it took)
- results of sanity check

Example output of `eb --trace`

```
$ eb TensorFlow-1.13.1-foss-2018b-Python-3.6.6.eb --trace
...
== preparing...
  >> loading toolchain module: foss/2018b
  >> loading modules for build dependencies:
  >>  * Bazel/0.20.0-GCCcore-7.3.0
  >>  * protobuf/3.6.1-GCCcore-7.3.0
  >> loading modules for (runtime) dependencies:
  >>  * Python/3.6.6-foss-2018b
  >>  * wheel/0.31.1-foss-2018b-Python-3.6.6
  >>  * h5py/2.8.0-foss-2018b-Python-3.6.6
  >> defining build environment for foss/2018b toolchain
...
== installing extension TensorFlow 1.13.1 (13/13)...
...
  >> running command:
        [started at: 2019-05-12 14:12:38]
        [output logged in /tmp/eb-pRHwkc/easybuild-run_cmd-SOINRV.log]
        bazel ... --jobs=6 --config=mkl //tensorflow/tools/pip_package:build_pip_package
  >> command completed: exit 0, ran in 00h41m22s
...
```

Generating container recipes/images using 

https://easybuild.readthedocs.io/en/latest/Containers.html

Container recipes can be *generated* by using `eb --containerize` ,
which is a lot more convenient than having to puzzle them together manually...

Support for Singularity containers was significantly improved in EasyBuild v3.9.2,
but it is still an **experimental feature**

- requires configuring EasyBuild with `--experimental` ,
  or setting the `$EASYBUILD_EXPERIMENTAL` environment variable
- this functionality may change in future EasyBuild releases (not stable yet)
- **feedback, bug reports, suggestions for enhancements are very welcome!**

Container support: the basics

https://easybuild.readthedocs.io/en/latest/Containers.html

experimental feature

- `eb --containerize` will generate a **container recipe**
  (rather than actually performing any installations)
- Singularity is used by default, Docker also supported
- if `--container-build-image` is specified, **container image** will be built too
  (required `sudo` privileges!)
- **container configuration** can be controlled via `--container-config`
- container recipes & images are created in the "container path"
  (see output of `eb --show-config`)

Example: TensorFlow in CentOS 7 container

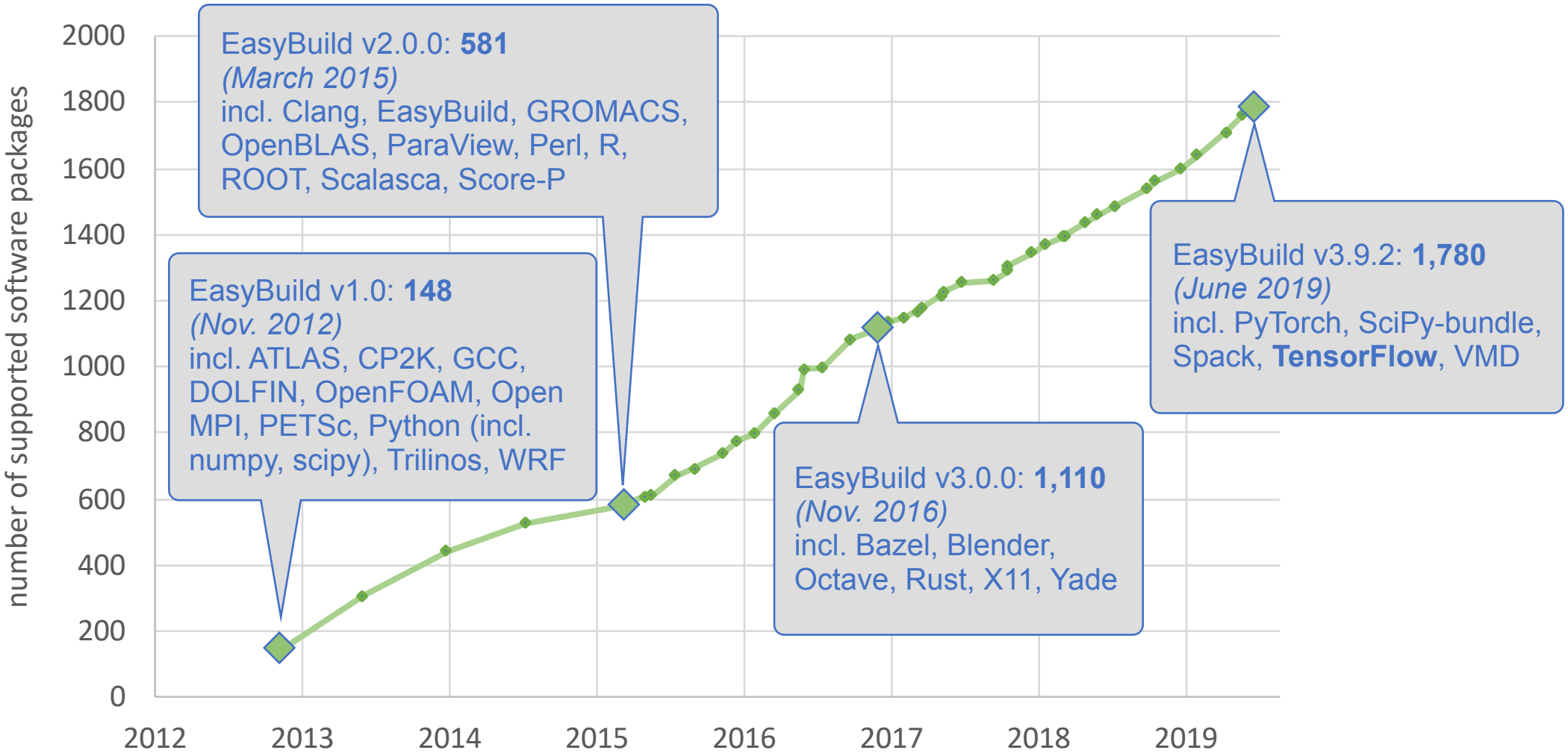
https://easybuild.readthedocs.io/en/latest/Containers.html

experimental feature

```
# configure EasyBuild to allow using experimental features
export EASYBUILD_EXPERIMENTAL=1
# specify container configuration (CentOS 7 from scratch)
export EASYBUILD_CONTAINER_CONFIG="bootstrap=yum,osversion=7"
# instruct EasyBuild to create container recipe that installs
# TensorFlow (incl. all dependencies)
eb --containerize TensorFlow-1.13.1-foss-2019a-Python-3.7.2.eb
# build container image
sudo singularity build TF.sif $HOME/containers/*TensorFlow*
```

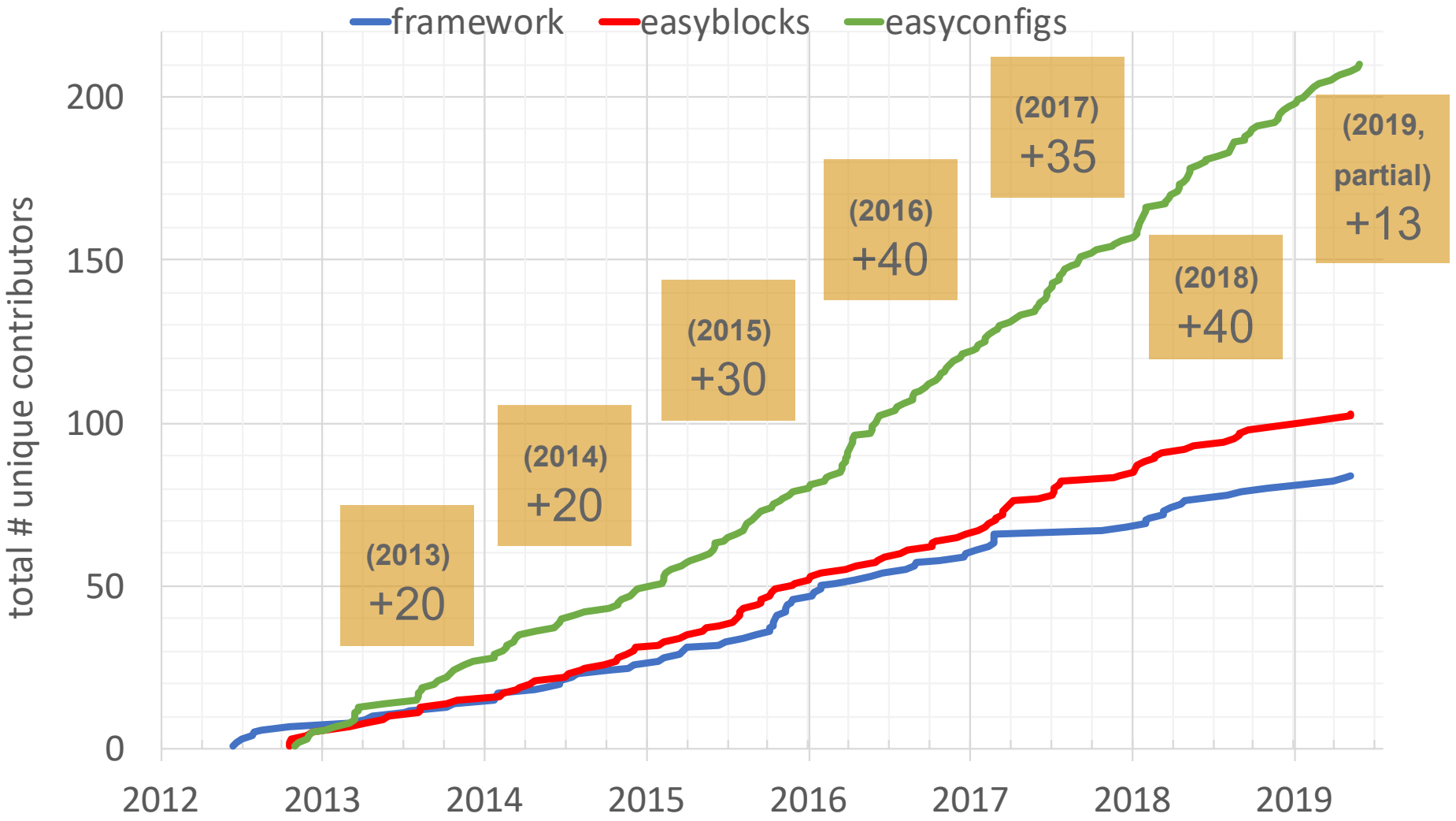
Evolution of supported software

steady growth in # supported software packages (excl. extensions)



Contributors, contributors, contributors...

Unique contributors (total, per repository)

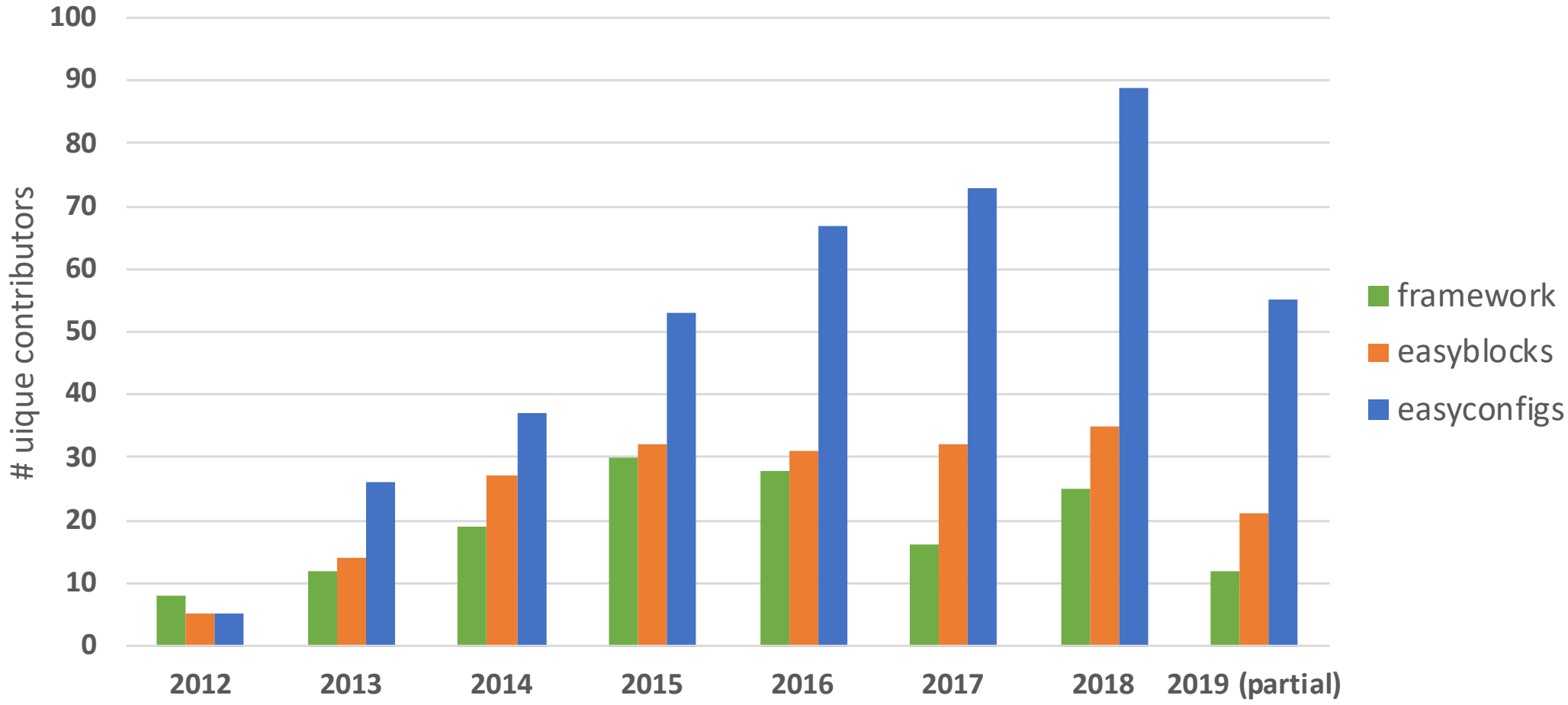


> 200 unique contributors for easyconfigs, > 100 for easyblocks, ~85 for framework

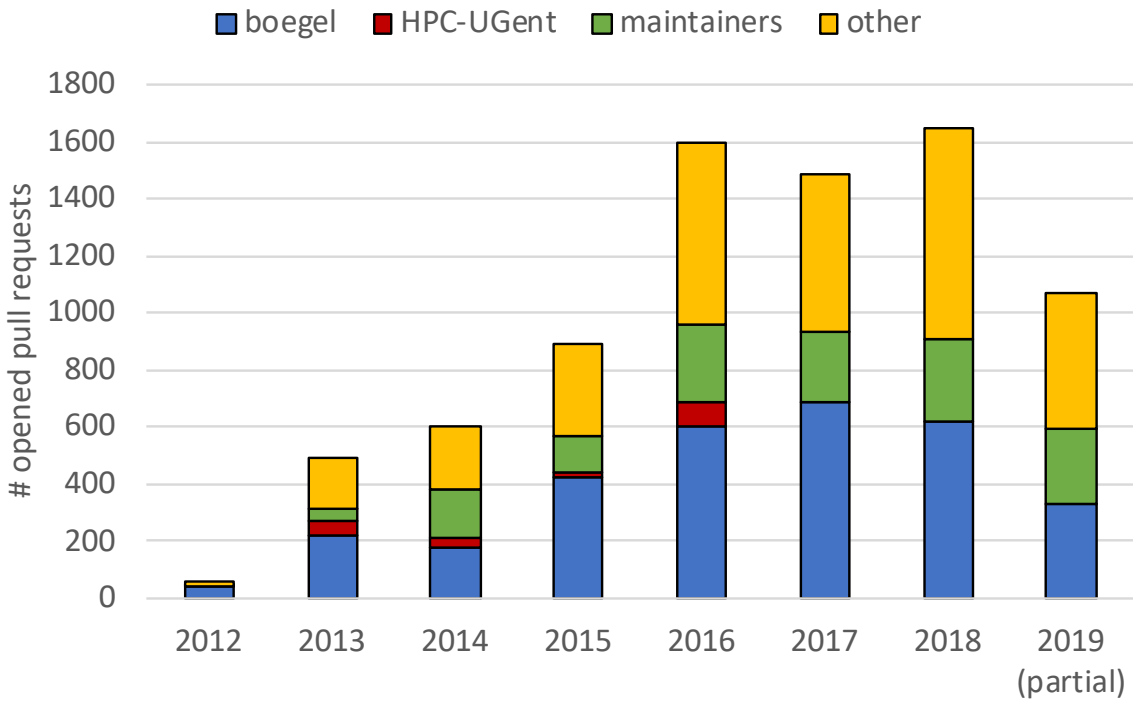
Unique contributors (per repository, per year)

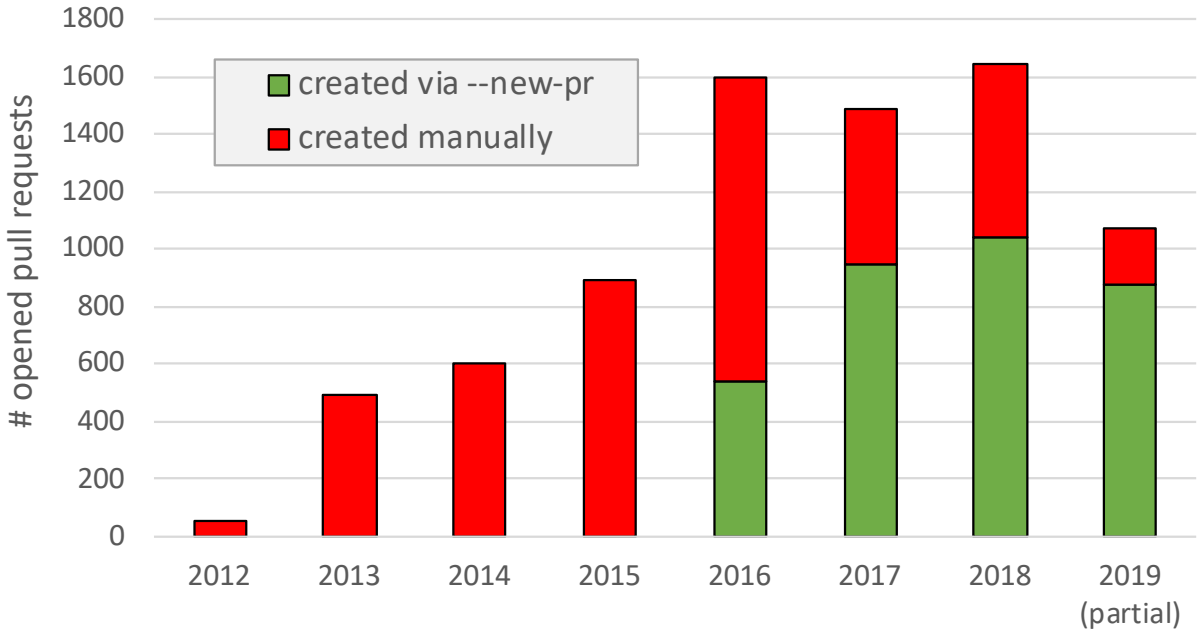
Group of easyconfigs contributors is growing every year (reaching 100 in 2019?).

Stable group of easyblocks contributors, fluctuating a bit for framework.



Overview of contributions (easyconfigs)





- explosive growth in easyconfig PRs since 2016 (partially thanks to `--new-pr`)
- stabilised in last couple of years (?), ~1,600 PRs in 2016-2018
- growing ratio of PRs from contributors that are not maintainers (~45% in 2018)
- growing ratio of easyconfig PRs created via `eb --new-pr` (>80% so far in 2019)

EasyBuild 4.0

https://github.com/easybuilders/easybuild/issues/447

Ongoing effort, see '4.x' branches in EasyBuild repositories on GitHub!

- ✓ **support for running EasyBuild on top of Python 3.5** (or more recent Python 3)
- ✓ no more required dependencies (vsc-base has been ingested, setuptools no longer needed)
- ✓ deprecated 'dummy' toolchain, replaced with 'system' toolchain (mostly about renaming)
  - custom easyblock for OpenMPI with more sensible default configuration
  - take the opportunity the change some defaults & make some minor breaking changes
- WIP switch to using `pip` by default for installing Python packages (?)
- WIP (maybe) stable integration with Singularity & Docker
- **ETA: summer 2019** (maybe EasyBuild 4.0-alpha early July?)

Status report on support for using Python 3

- **porting effort to allow running EasyBuild with Python 3 mostly done**
- support for using Python 2.6 or Python 2.7 is retained
- see `4.x` branches in `easybuild-*` repositories (kept in sync with `develop`)
- current status:
  - relevant code from vsc-install + vsc-base ingested into easybuild-framework
  - all unit tests pass in framework, easyblocks, easyconfigs repositories
  - *can build and install TensorFlow from scratch using '`eb`' running on top of Python 3*
- more extensive testing is required to declare success...
- estimated effort so far: 50-60 hours

see `python3` label: https://github.com/easybuilders/easybuild-framework/labels/python3

Future work: more extensive/organised testing

- current pre-release regression testing is (way) too ad-hoc
  - rebuild *all* included easyconfig files at HPC-UGent infrastructure
  - weed out fluke failures from important problems that should block the release
  - no public dashboard
- hope is to set up a **shared testing infrastructure** for maintainers
  - leverage Singularity to testing in different OSs (CentOS, Debian, SLES, ...)
  - use QEMU to test on specific processor architectures (Haswell, Skylake, ARM, POWER?)
  - minimal vs fat OS configurations
  - use AWS? (some technical issues there + funding needed to cover costs)
  - allow other (approved) contributors to trigger test builds for their contributions?
  - use ReFrame to run EasyBuild regression tests
    - waiting for good support for specifying dependencies between tests

vs

Spack

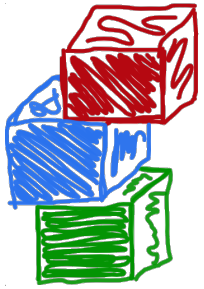
https://spack.rtfd.io

- EasyBuild: GPLv2 license; Spack: MIT/Apache 2.0 license
- **no stable releases yet for Spack (< 1.0), EasyBuild is stable since 2012**
- ~ on par w.r.t. amount of supported software (but differences w.r.t. which software)
- (originally) targeted for different use cases: HPC support teams vs developers
- fixed dependency/toolchain versions in EasyBuild vs flexible CLI in Spack
- running EasyBuild on top of Python 3 is not supported yet (but coming really soon!)
- macOS support in EasyBuild is limited (no toolchains/testing for macOS)
- both projects are backed by an active & supportive community!

For a more detailed comparison,
see https://archive.fosdem.org/2018/schedule/event/installing_software_for_scientists

Lessons learned

- be careful with starting a project, you may start a community...
- no two projects are equal, even if they may seem (very) similar
- time is a b*tch that takes no prisoners
  - real work vs development vs community vs release management vs giving talks vs family & personal life...
- being able to work on a project “on the job” is a MAJOR help
- don’t take skills for granted, be patient/welcoming to new/infrequent contributors
- it takes time for people to pick up on new functionality (even if it's awesome)
- you may have created the best thing ever, you still have to "sell" it...



easybuild

Questions?

Kenneth Hoste (HPC-UGent)

kenneth.hoste@ugent.be

20190614 - HPCKP'19 (Barcelona)

https://users.ugent.be/~kehoste/EasyBuild_20190614_10_years.pdf

https://easybuilders.github.io/easybuild

https://easybuild.readthedocs.io

Extra slides

(only used in case of time/interest)

Using a custom module naming scheme

- a couple of different module naming schemes are included in EasyBuild
  - see `--avail-module-naming-schemes`
  - specify active module naming scheme via `--module-naming-scheme`
  - default: `EasyBuildMNS` (`<name>/<version>-<toolchain>-<versionsuffix>`)
- **you can implement your own module naming scheme relatively easily**
  - specify how to compose module name using provided metadata
  - via Python module that defines custom derivative class of `ModuleNamingScheme`
  - make EasyBuild aware of it via `--include-module-naming-schemes`
- decouple naming of install dirs vs modules via `--fixed-installdir-naming-scheme`

Flat module naming scheme

- all modules are always available for loading
- long(er) module names
- '`module avail`' may be overwhelming for users
- too easy to load incompatible modules together

legend

(not available)

(available)

(loaded)

GCC/5.3.0

GCC/6.1.0

OpenMPI/1.10.2-GCC-5.3.0

OpenMPI/2.1.0-GCC-5.3.0

OpenMPI/1.10.3-GCC-6.1.0

OpenMPI/2.1.0-GCC-6.1.0

FFTW/3.3.4-gompi-2016.04

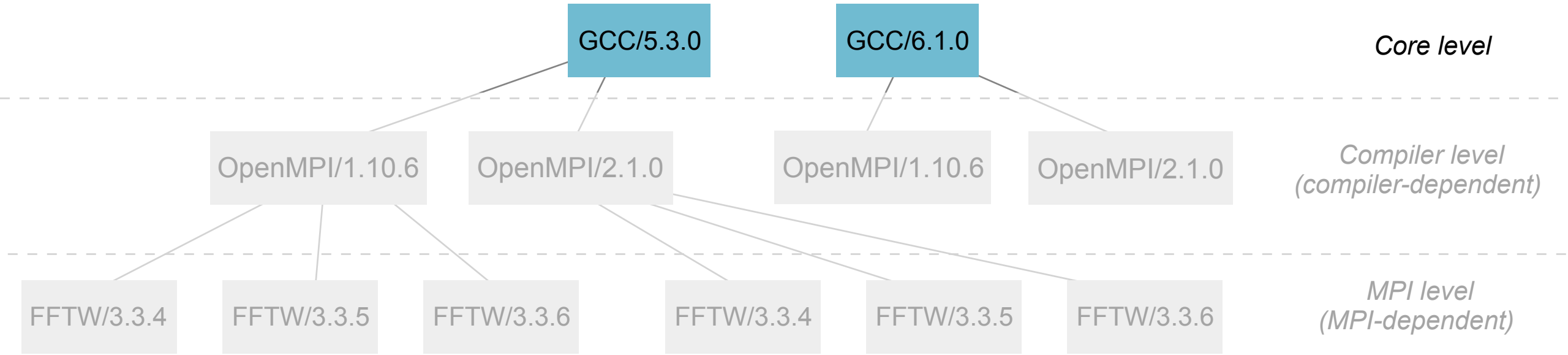
FFTW/3.3.6-gompi-2016.04

FFTW/3.3.4-gompi-2016.07

FFTW/3.3.6-gompi-2016.07

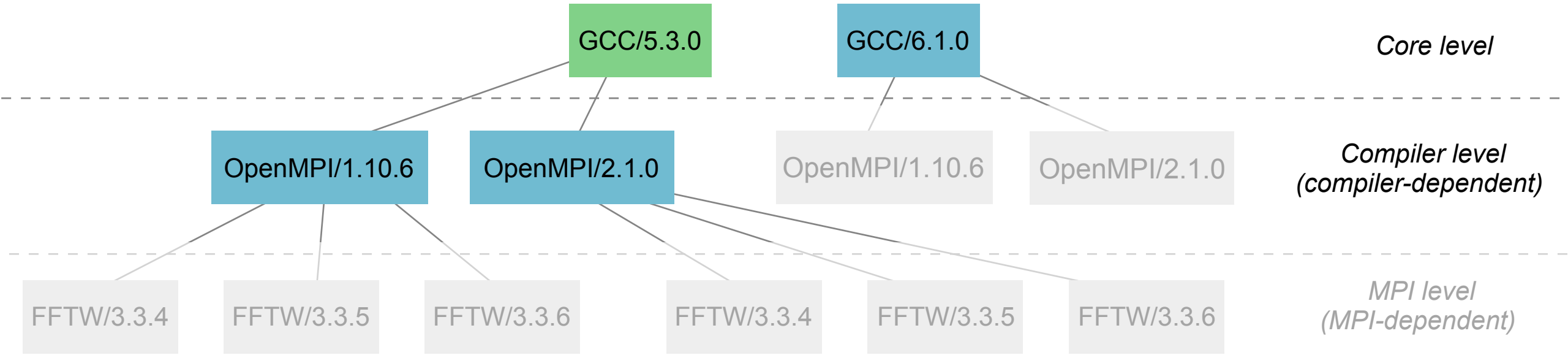
Hierarchical module naming scheme (1)

- modules are organised in a tree-like fashion
- initially, only 'core' modules are available for loading
- typically 3 hierarchy levels: core, compiler-dependent, MPI-dependent
- other modules are only visible via '`module spider`'



Hierarchical module naming scheme (2)

- Core modules may extend `$MODULEPATH` with an additional location
- loading a Core module may make more modules available
- in this example, loading a GCC module makes OpenMPI modules available



Hierarchical module naming scheme (3)

- even more modules may be made available by loading other modules
- for example, loading an OpenMPI modules reveals MPI-dependent modules
- EasyBuild can organise modules in hierarchy for you!