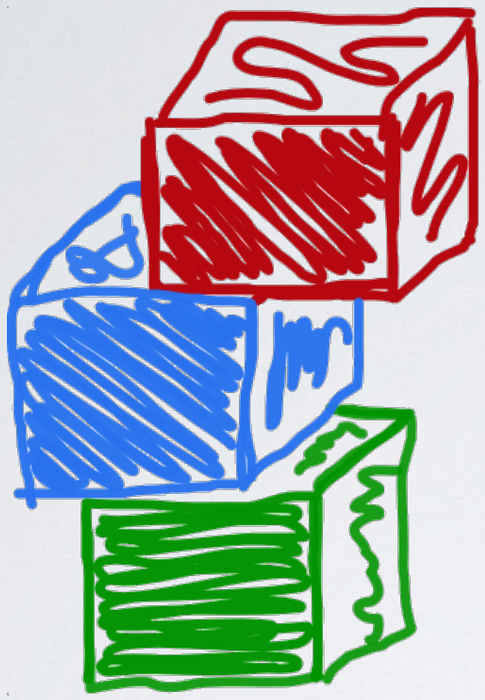


easybuild

Past, Present & Future

Kenneth Hoste (HPC-UGent) - kenneth.hoste@ugent.be

http://users.ugent.be/~kehoste/EasyBuild_past_present_future_201704.pdf

2nd EasyBuild User Meeting

Feb 8th 2017 - Jülich Supercomputing Centre

A bit of EasyBuild history...

2009
-
2012

- started as an HPC-UGent in-house project in 2009
- originally created by Stijn De Weirdt
- Kenneth Hoste inherited project end of 2010
- several interns have helped redesign the EasyBuild framework

A bit of EasyBuild history...

- first public release (v0.5) in April '12
- EasyBuild v1.0 released during SC'12 (Nov '12)
- used to be one large codebase;
  now split into framework/easyblocks/easyconfigs packages
- introduced concept of easyblocks & extensions
- started with (framework) test suite

A bit of EasyBuild history...

2009 - 2012	2012	2013
in-house development	first public release; v1.0	

- config via cfg file/environment/cmdline via `generaloption`
- introduction of EasyBuild bootstrap script
- support for:
  - more toolchains & toolchain components
  - Lmod as modules tool (+ the pure Tcl version of Tmod)
  - using a custom module naming scheme
  - `--dry-run / -D`

A bit of EasyBuild history...

2009 - 2012	2012	2013	2014
in-house development	first public release; v1.0	cfg, Lmod, bootstrap, MNS	

- GitHub integration (`--from-pr`, `--upload-test-report`)
- support for:
  - integration of `--try-*` with `--robot`
  - `--experimental`
  - the infamous easyconfig format v2 (which died a silent death)
  - hierarchical module naming schemes
  - hidden modules
- performance improvements & code cleanup
- documentation at http://easybuild.readthedocs.io/

A bit of EasyBuild history...

2009 - 2012	2012	2013	2014	2015
in-house development	first public release; v1.0	cfg, Lmod, bootstrap, MNS	GitHub, HMNS, `--try-* -r`	

- EasyBuild v2.0 (March '15)
  - no more automagic fallback to `ConfigureMake` easyblock
  - vsc-base as proper dependency (rather than including a copy)
- support for `--module-only`
- Cray support *(experimental)*
- module files in Lua syntax
- using 'external' modules
- GC3Pie as job backend
- `--include-*`
- packaging support via FPM
- `--extended-dry-run` / `-x`
- .yeb easyconfig *(experimental)*
- minimal toolchains *(experimental)*
- `--optarch=GENERIC`

A bit of EasyBuild history...

2009 - 2012	2012	2013	2014	2015	2016
in-house development	first public release; v1.0	cfg, Lmod, bootstrap, MNS	GitHub, HMNS, `--try-* -r`	v2.0, Lua, -x GC3Pie, pkg `--module-only`, `--include-*`	

- `--new-pr`, `--update-pr`
- *stable* Cray support
- more performance improvements
- PGI as toolchain compiler
- unit tests at Travis CI
- `--list-software`
- stable `--minimal-toolchains`
- support use of `ccache`

- EasyBuild v3.0 (Nov'16)
  - Lmod/Lua/GC3Pie by default
  - robot considers subtoolchains
  - RPATH support *(experimental)*
  - archived easyconfigs
  - more frequent minor releases

EasyBuild v3.1.0

- latest stable feature release (Feb 3rd 2017)
  - support for `--check-style` to check 'code' style in easyconfig files
  - architecture-specific optimisation flags on per-compiler basis via `--optarch`
    http://easybuild.readthedocs.io/en/latest/Controlling_compiler_optimization_flags.html
  - improved toolchain support on ARM and POWER systems
    - FFTW easyblock that maximises use of processor features (by default)
    - updates to the CP2K and WPS easyblocks for the latest versions
    - toolchain definitions for the foss/2017a and intel/2017a common toolchains
      http://easybuild.readthedocs.io/en/latest/Common-toolchains.html
    - support for 16 new software packages, incl. QIIME (via conda), Tensorflow
    - & various (minor) bug fixes...
- more details: http://easybuild.readthedocs.io/en/latest/Release_notes.html

EasyBuild v3.1.0

JSC (implemented by) (contributions by)

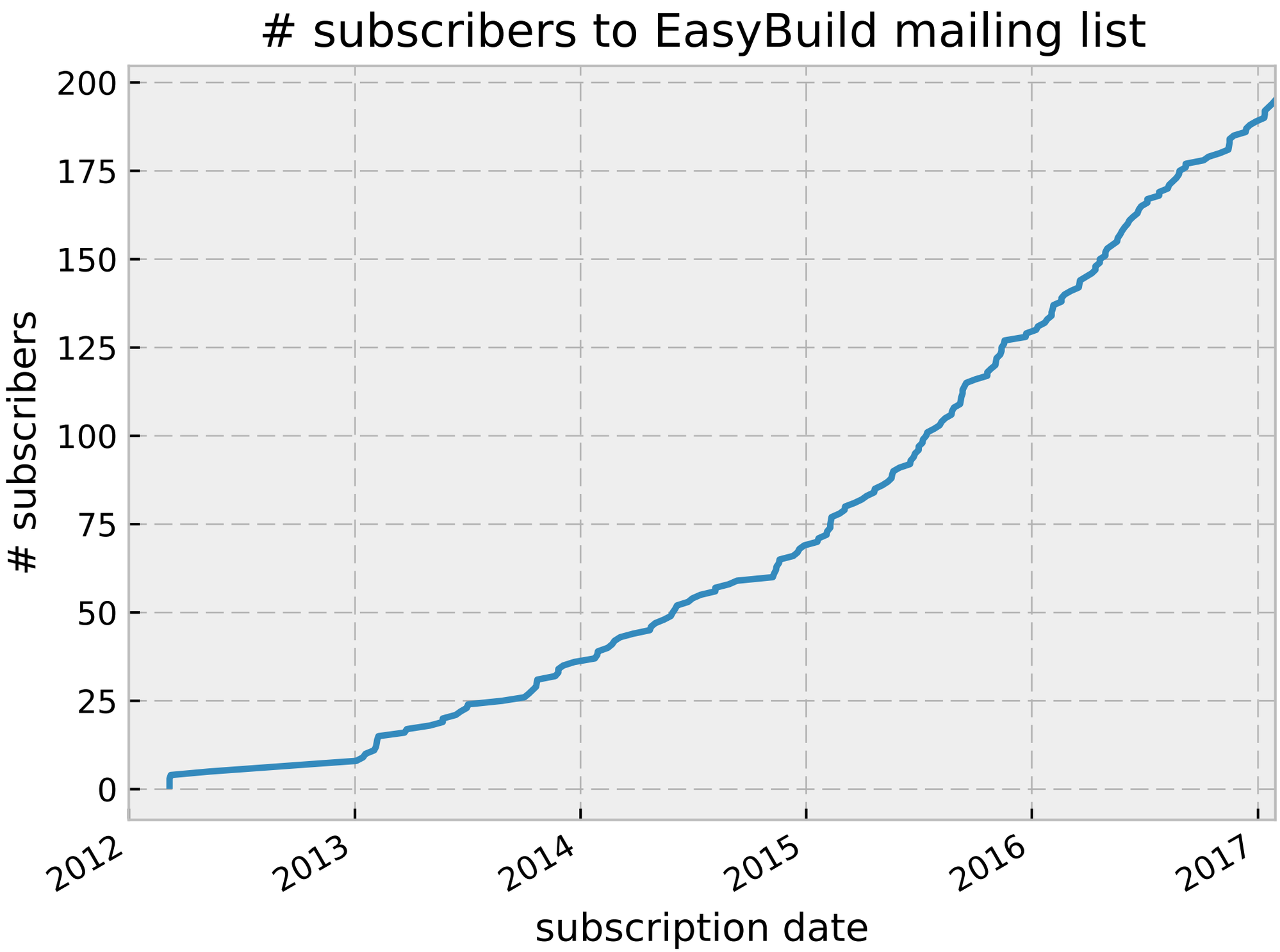
- latest stable feature release (Feb 3rd 2017)
  - support for `--check-style` to check 'code' style in easyconfig files — Damian
  - architecture-specific optimisation flags on per-compiler basis via `--optarch`
    http://easybuild.readthedocs.io/en/latest/Controlling_compiler_optimization_flags.html
  - improved toolchain support on ARM and POWER systems — Markus
  - FFTW easyblock that maximises use of processor features (by default) — Markus
  - updates to the CP2K and WPS easyblocks for the latest versions — Damian/Alan
  - toolchain definitions for the foss/2017a and intel/2017a common toolchains
    http://easybuild.readthedocs.io/en/latest/Common-toolchains.html — Damian
  - support for 16 new software packages, incl. QIIME (via conda), Tensorflow
  - & various (minor) bug fixes... — Alan, Markus/Alan
- more details: http://easybuild.readthedocs.io/en/latest/Release_notes.html

Common toolchains

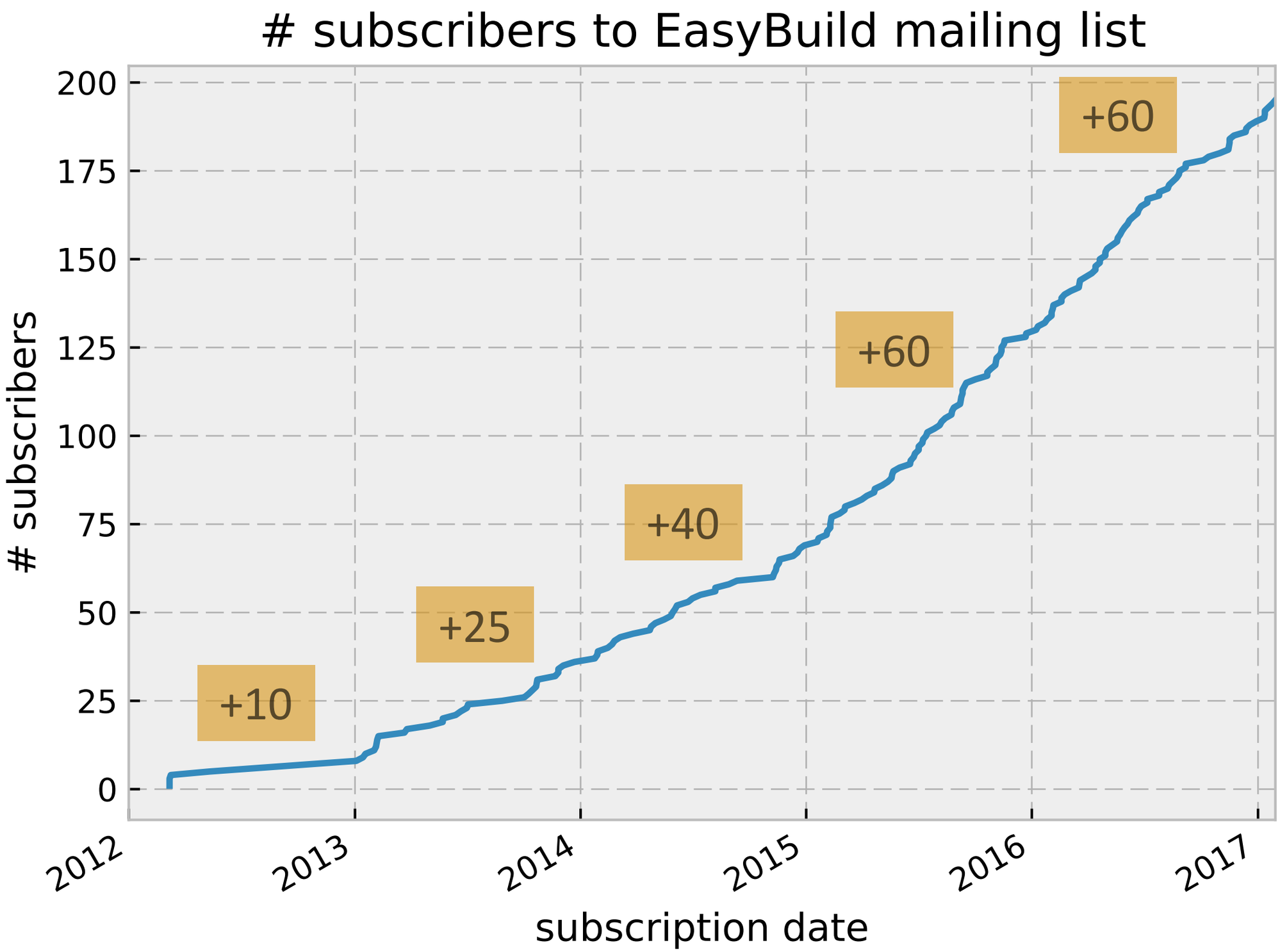
http://easybuild.readthedocs.io/en/latest/Common-toolchains.html

- compiler toolchains we try to promote in the community
- more people/sites using them leads to more saved time on average
  - focus efforts on easyconfigs that others can just pick up and use
  - maximise benefit from collaboration
- two 'flavors':
  - `foss` (Free & Open Source Software)
    - (binutils +) GCC, OpenMPI, OpenBLAS + LAPACK, FFTW (currently)
  - `intel`
    - (binutils + GCC +) Intel compilers, Intel MPI, Intel MKL
- 6-month update cycle (Dec-Jan, Jun-Jul)
- since July 2014 (`foss/2014b`, intel/2014b)
- latest: `foss/2017a`, intel/2017a

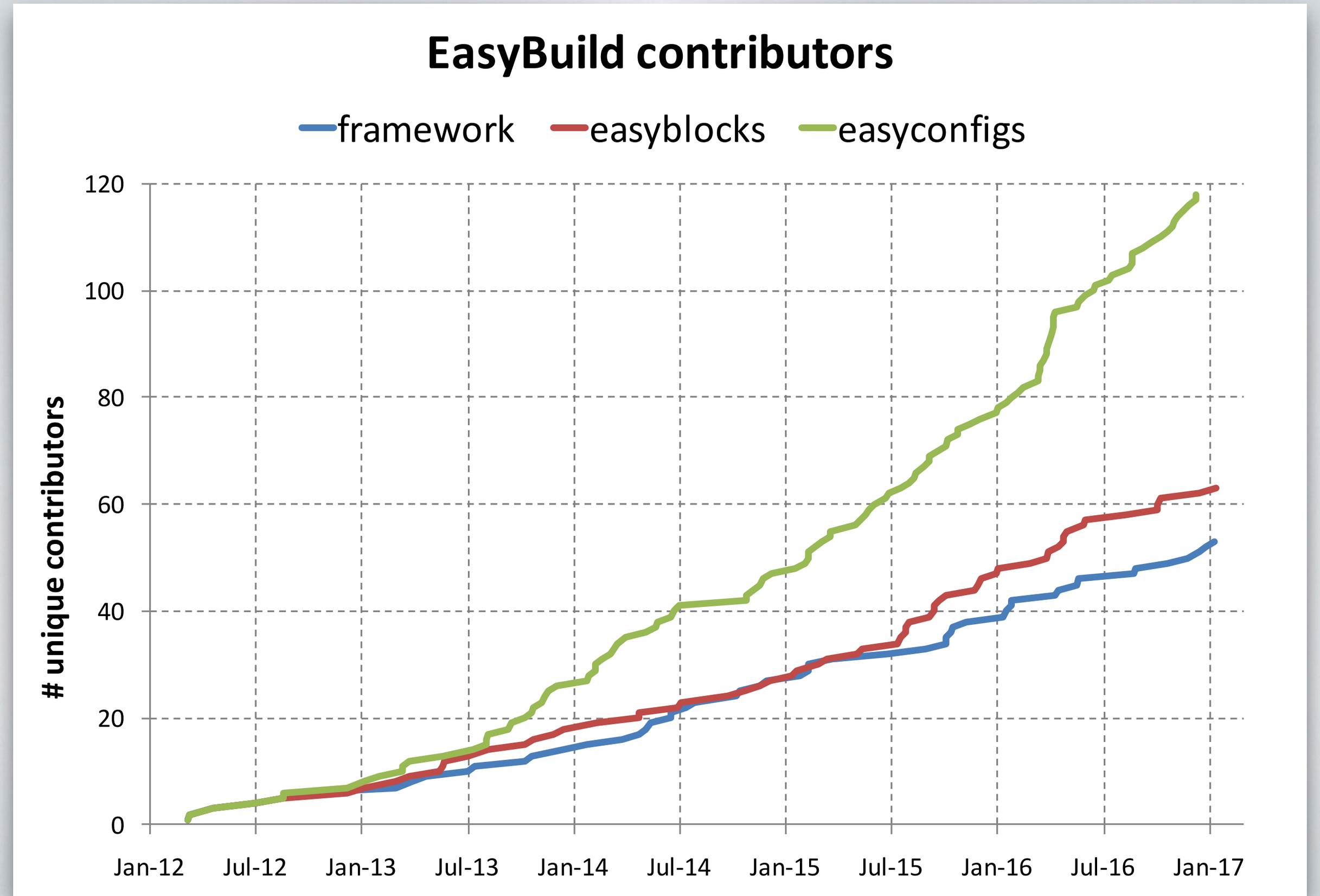
EasyBuild community growth



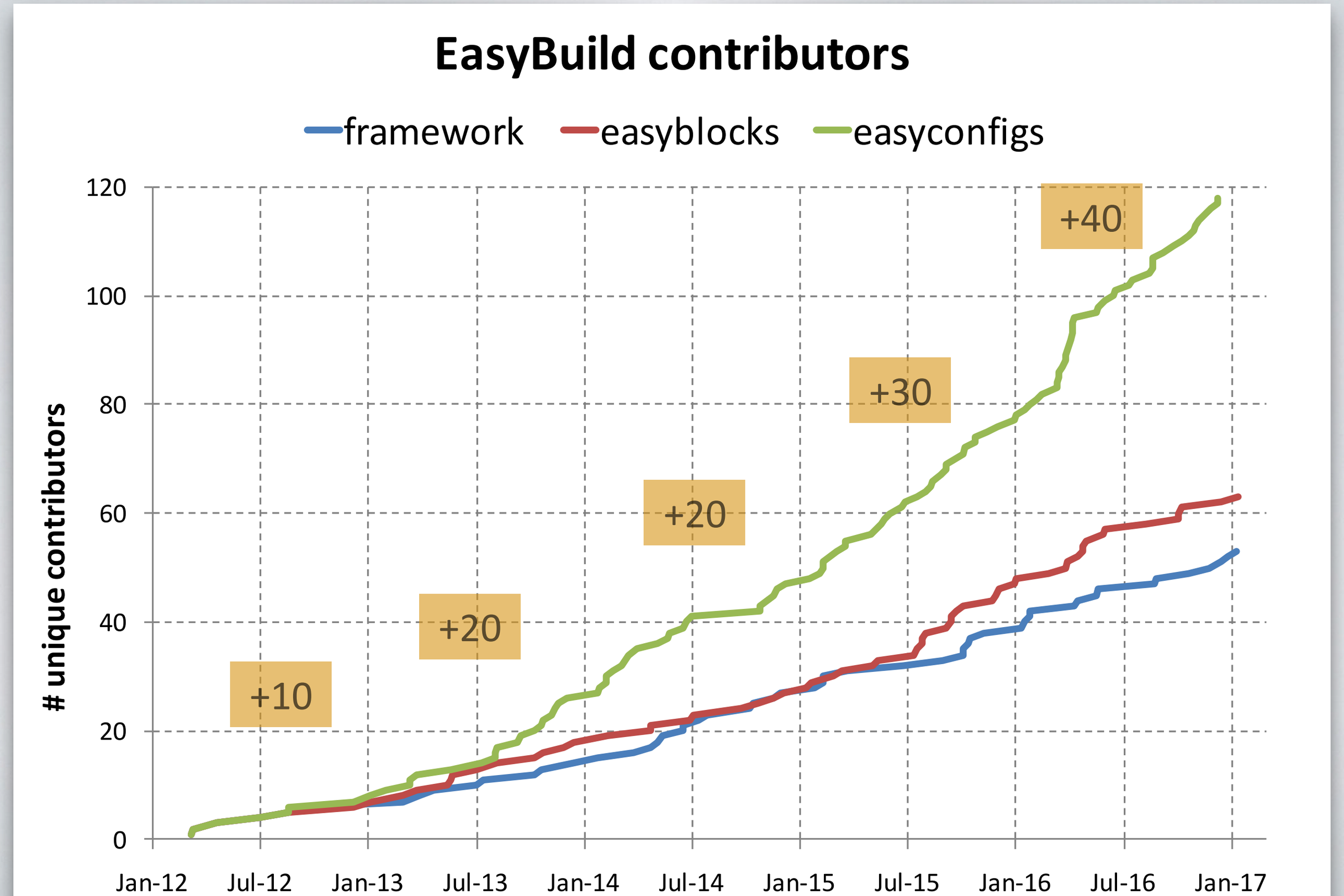
EasyBuild community growth



EasyBuild community growth



EasyBuild community growth



RPATH support

- still under `--experimental` in EasyBuild v3.1.0
- close to being stable
  - well tested by Pablo
  - some minor open issues (https://github.com/hpcugent/easybuild-framework/issues/1992)
    - should not use `$ORIGIN`, hardcode installation prefix instead
- optionally wrap installed compilers to make users also use RPATH
- some software requires a custom RPATH sanity check
- some software requires extra work to ensure RPATH linking
- support for specifying link dependencies would be useful
- supporting both RPATH and RUNPATH should be quite easy

Most significant known issues

- installing EasyBuild is (still) too painful
  - requiring setuptools is a PITA (and it's getting worse)
  - optional dependencies should be easier to install along with EasyBuild
  - may be fixable by creating a singe EasyBuild package for each release
- no good way yet to ensure we catch all required dependencies
  - contributed easyconfigs should be tested in isolation from the OS...
- lack of good support for site customisations
- maintenance burden of easyconfig files
  - fat .yeb easyconfigs should help significantly with this
- keeping up with contributions is hard, we need more manpower/automation
- anything else?

Most significant known issues (1/2)

- installing EasyBuild is (still) too painful
  - requiring setuptools is a PITA (and it's getting worse)
  - optional deps should be easier to install along with EasyBuild
    - may be fixable by creating a singe package for each release
- no good way yet to ensure we catch all required dependencies
  - contributed easyconfigs should be tested in isolation from OS...

Most significant known issues (2/2)

- lack of good support for site customisations
- maintenance burden of easyconfig files
  - fat .yeb easyconfigs should help significantly with this
- keeping up with contributions is hard,

  we need more manpower/automation
- anything else?

EasyBuild vs Spack

https://spack.io/

- "flexible package manager for supercomputers"
- quite similar to EasyBuild: also Python2, high-level structure is comparable
- has seen tremendous growth in 2016
- very good documentation (cfr. SC16 tutorial)
- powerful command line to juggle with dependencies
- not 100% stable yet, but getting there fast (next release should be v1.0)
- main target use case is different from EasyBuild (imho)
  - mostly intended for developers of large scientific software applications
  - requires more software development expertise
    - Python
    - Git
    - packaging experience

Future work (1/2)

- stable RPATH support
- better support for site customisation
  - currently expected to maintain Git repo with tweaked easyconfigs
  - we should have an easier/more integration mechanism
    - revive .ebp idea?
- support for fat easyconfigs in YAML syntax (.yeb)
  - step-wise: multiple toolchains
  - later also multiple software versions, etc.
  - goes way beyond just parsing of easyconfigs...

Future work (2/2)

- EasyBuild & Singularity
  - running EasyBuild in isolation from the OS
  - using EasyBuild to create Singularity images
  - testing EasyBuild on multiple operating systems (regardless of host OS)
- EasyBuild tutorial at SC17?
  - opportunity to improve EasyBuild documentation