

easybuild

Into a new decade

Kenneth Hoste (HPC-UGent - VSC)

kenneth.hoste@ugent.be

2022-11-14

Agenda

- **Looking back at the previous decade**
 - A brief history of EasyBuild
 - Whirlwind tour of major EasyBuild features that were developed
 - The EasyBuild community
 - Lessons learned
- **Looking forward**
 - Current ongoing efforts
 - What's coming (soon)
 - Exciting opportunities ahead
- **A nice surprise at the end...**

10+ years of EasyBuild



<https://time.graphics/line/160902>

EasyBuild has gone through 6 “eras” so far:

- *summer 2009 - Apr'12: in-house development at HPC-UGent*
- *Apr'12 - Nov'12 (v0.x): public release, mailing list/Twitter/IRC, logo, first users & 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 - Sept'19 (v3.x): support for RPATH/hooks/containers, EasyBuild **maintainers** team*
- *Sept'19 - now (v4.x): compatibility with **Python 3**, only Python stdlib + env. modules tool required*

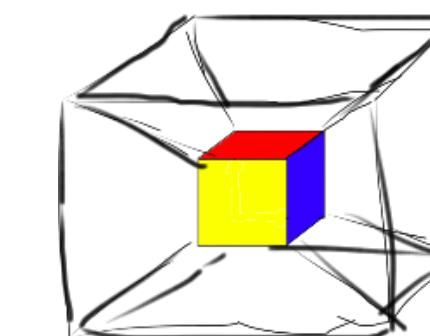
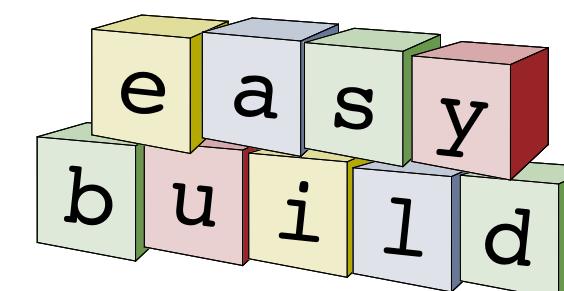
The dawn of EasyBuild

- **EasyBuild was created to scratch our own itch** at HPC-UGent in 2009 by Stijn De Weirdt
 - Lots of software installation requests for first central HPC cluster at Ghent University
 - Nobody seemed to have a good way to deal with this burden (other than manpower)
- We didn't really have experience with developing (open source) software...
- **Students interns were a *big* help** to make EasyBuild ready to release publicly:
cleaning up, refactoring, redesigning the codebase (making Pylint score **positive**),
kickstarting a suite of (unit) tests, fixing bugs, developing features, ...
- Initial development was for Scientific Linux 5,
incl. Python 2.4, Environment Modules 3.2.7



Should we make EasyBuild available publicly?

- We started to wonder if we should make release **EasyBuild as open source software...**
- Our hope was to get feedback from the HPC community, maybe we were missing something?
- Of course, we needed a logo first...

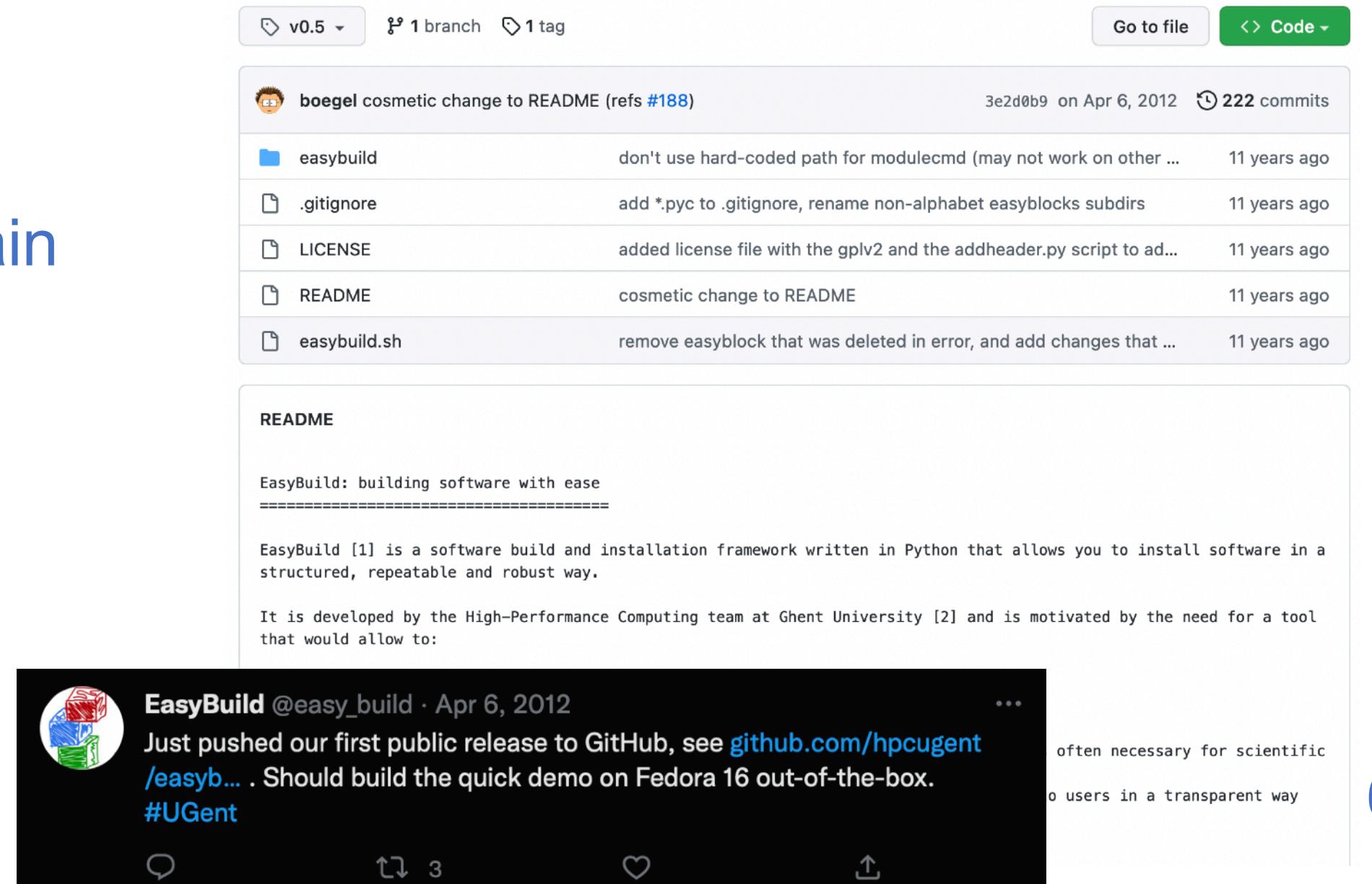


- Inspired by some Belgian beer, we ended up picking our now notorious logo design:



Releasing EasyBuild into the wild

- In 2012, we felt the codebase was decent enough to **make it publicly available (GPLv2)**
- **easybuild GitHub repository** was created on 8 March 2012
 - In-house development until then was done in a private SVN repository
 - Shortly after, we split things up into 4 separate repositories (framework, easyblocks, easyconfigs, docs)
- **Initial EasyBuild “release” (v0.5 version tag) on 6 April 2012**
 - No eb command yet (introduced in July 2012 by student intern)
 - Crude documentation via GitHub wiki
 - Only provided support for building HPL with goalf toolchain (GCC, OpenMPI, ATLAS, (Sca)LAPACK, FFTW)
 - ~6 kLoC, 23 easyconfig files + 9 easyblocks included



Promoting EasyBuild

- First public talk (23 April 2012): *EasyBuild: building software with ease* by Jens Timmerman at HEPiX Spring 2012 workshop (slides still available!)

EasyBuild: Building Software With Ease.

Jens Timmerman Stijn De Weirdt Kenneth Hoste

Department ICT
Ghent University
easybuild@lists.ugent.be

HEPiX Spring 2012 Workshop



Not all end-users can build the software they use.

- Compilation requires lots of expertise and specific knowledge.
 - Not all documentation is created equal.
 - Not all software has a straightforward build procedure.
- Collaboration and sharing the knowledge is problematic at best.



Figure: <http://geekandpoke.typepad.com/>

Current state

- In use for over 3 years, we build all end-user software with it
- Framework cleaned up and released as open source begin April 2012.
 - <https://github.com/hpcugent/easybuild>
- Very well tested under Scientific Linux (SL) 5 and 6.
- Framework also tested under:
 - Fedora 16
 - Ubuntu 10.04 and 11.10
 - Suse
 - Arch Linux
 - OS X 10.7.3



List of currently supported software packages (>250)

ABAQUS	ABINIT	ABYSS	ADF	ALLPATHS-LG	ARPACK	ATLAS	AmberTools	Armadillo	AutoDock-Vina	Autoconf	Aztec	BEAGLE	BLACS	BLAST	BWA	BamTools	BiSearch	BiGIFT	Biopython	Blas	Blitz++	Boost	Bowtie	CAMFR	CAPHE	CCFin	CD-HIT	CFITSIO	CLHEP	ClooG-PPL	CMakel	COMSOL	CP2K	CPLEX	CPMD	Cabal	Charm++	ClustalW	ClustalW2	Cufflinks	DIANA	DOLFIN	Denoiser	EMAN	EPD	ETSFIO	F77SPPLIT	FASTX	FFTW	FIAT	FLAME	FLUENT	FastTree	Ferret	Flume	GAMESS-US	GATE	GCC	GDAL	GDB	GEANT4	GEOS	GHC	GLPK	GMP	GRASS	GROMACS	GSL	Gambit	Gaussian	GenomeAnalysisTK	GotobLAS	HBase	HDFS	HOD	HPCC	Hadoop	Harminv	Hive	Hoard	IMOD	IPM	IPython	ISIS	Insight	Instant	IronPython	JUnit	JasPer	Java	JumboMem	LAPACK	LMF	LS-DYNA	Libint	M4	MATLAB	MEMS	METIS	MPB	MPFR	MPITB	MUMPS	MVAPICH2	Maple	Maven	Meep	Mono	Mpc	MrBayes	MyMediaLite	MySQL	NAMD	NCL	NCO	NWChem	OPT++	ORCA	Octave	Open64	OpenCV	OpenFOAM	OpenMPI	PCRE	PEST	PETSc	PHENIX	PHP	PPL	PROJ	PVM	ParMETIS	Parallel-netCDF	Path64	Perl	PostgreSQL	Primer3	PyNAST	PyPar	Python	QIIME	QuEST	Quantum-ESPRESSO	R	RAPTOR	RDP-Classifer	ROOT	ROSETTA	Reval	Ryy	Ruby	SAMtools	SAS	SCOTCH	SCons	SGA	SHELXL	SOA	Pdenovo	SPARSKIT	SPIDER	SPRNG	SWIG	Sage	ScALAPACK	ScientificPython	SeqTrimNEXT	Silico	SuiteSparse	Szip	Tcl	Tk	TopHat	Treefinder	Trinity	UCLUST	UFC	UFL	UNAFold	VASP	Viper	WIEN2k	WRF	Wanner90	Xmipp	Y12M	YASMIN	ant	bcMPI	boost-numeric-bindings	bzzip2	cURL	cdbtools	fastahack	ffmpeg	g2l2n	galib	ginkml	gngfl	gnuplot	gogfl	gomkl	google-sparsehash	gtk-sharp	gzip	h5utils	iSight	icc	ictce	ifort	imkl	impi	ipp	itac	jobs	libctb	libtextutils	libmatheval	libpng	libsmm	libxml2	libxml2	likwid	maui	mhash	microbiomeutl	mpiBLAST	mpiGraph	netCDF	orc	picard	ploticus	pyTables	pysam	python-meep	quupdate	readline	schroedinger	tbb	unixODBC	x264	yasm	zlib
--------	--------	-------	-----	-------------	--------	-------	------------	-----------	---------------	----------	-------	--------	-------	-------	-----	----------	----------	--------	-----------	------	---------	-------	--------	-------	-------	-------	--------	---------	-------	-----------	--------	--------	------	-------	------	-------	---------	----------	-----------	-----------	-------	--------	----------	------	-----	--------	-----------	-------	------	------	-------	--------	----------	--------	-------	-----------	------	-----	------	-----	--------	------	-----	------	-----	-------	---------	-----	--------	----------	------------------	----------	-------	------	-----	------	--------	---------	------	-------	------	-----	---------	------	---------	---------	------------	-------	--------	------	----------	--------	-----	---------	--------	----	--------	------	-------	-----	------	-------	-------	----------	-------	-------	------	------	-----	---------	-------------	-------	------	-----	-----	--------	-------	------	--------	--------	--------	----------	---------	------	------	-------	--------	-----	-----	------	-----	----------	-----------------	--------	------	------------	---------	--------	-------	--------	-------	-------	------------------	---	--------	---------------	------	---------	-------	-----	------	----------	-----	--------	-------	-----	--------	-----	---------	----------	--------	-------	------	------	-----------	------------------	-------------	--------	-------------	------	-----	----	--------	------------	---------	--------	-----	-----	---------	------	-------	--------	-----	----------	-------	------	--------	-----	-------	------------------------	--------	------	----------	-----------	--------	-------	-------	--------	-------	---------	-------	-------	-------------------	-----------	------	---------	--------	-----	-------	-------	------	------	-----	------	------	--------	--------------	-------------	--------	--------	---------	---------	--------	------	-------	---------------	----------	----------	--------	-----	--------	----------	----------	-------	-------------	----------	----------	--------------	-----	----------	------	------	------

J. Timmerman (HPC UGent) EasyBuild HEPiX Spring 2012 20 / 21

Summary

- EasyBuild allows to easily reproduce software builds/installations and install multiple versions.
- Feedback is very welcome, contributions even more:
 - easybuild@lists.ugent.be
 - <https://github.com/hpcugent/easybuild>
- Outlook
 - We are currently cleaning up easyblocks and releasing them on github regularly.



J. Timmerman (HPC UGent) EasyBuild HEPiX Spring 2012 21 / 21

- First feedback via EasyBuild IRC channel (May 2012):
“so you've badly reimaged prefix portage. Great!”

- EasyBuild mailing list was made public in May 2012
 - First subscriber outside of HPC-UGent team: Fotis Georgatos (with @cern.ch email)
 - Now exactly 300 subscribers (but mostly replaced by EasyBuild Slack with over 670 members)

EasyBuild v1.0

- EasyBuild v1.0 was released on 13 Nov 2012 via GitHub + PyPI
 - Literally hours before it was going to be presented at the PyHPC workshop at Supercomputing'12 (see [paper](#) + [slides](#))
 - Main point was to stabilising the API of the EasyBuild framework
 - Support for 148 software packages (~21 kLoC, 76 easyblocks, 339 easyconfigs)
- Overall feedback was very positive, apparently we had hit a “hole in the market”
- This motivated us to continue with the public development of EasyBuild...

Dr. T
@aterrel

hpcugent.github.com/easybuild/ EasyBuild making installing supercomputing software easy for the masses. Coolest thing I'll take home from #sc12 so far

4:56 PM · Nov 12, 2012 · Twitter for Android

6 Retweets 2 Likes

Two commands from #SC12 PyHPC: easy_install easybuild; eb <your fav package>

5:13 PM · Nov 16, 2012 · Twitter for Android

easybuild Scientific software

Scientists (generally) spend their time and effort in developing (and testing) their code, not in maintaining it.

Build procedures for scientific software are often:

- **incomplete**: e.g., no actual *install* step, only build-in-src-dir
- **non-standard**: e.g., requiring human interaction
- **customized**: custom scripts for configuration, building, ... instead of configure, cmake, make, etc.
- **hard-coded**: no configure option for libraries, compiler commands and flags, ...

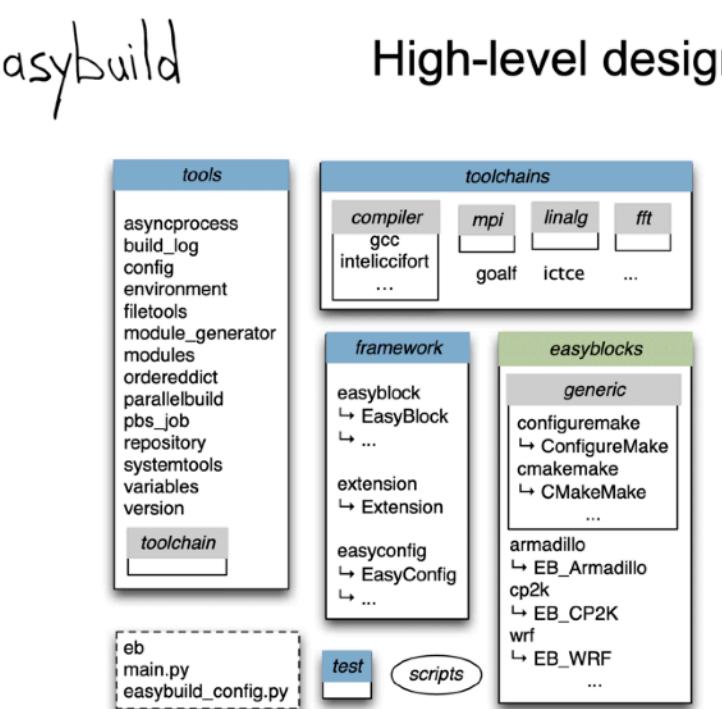
Very time-consuming for HPC user support teams!

easybuild Use case: WRF

Weather Research and Forecasting Model (WRF)

complex(ish) dependency graph

dependencies also feature nasty build procedures



easybuild High-level design

EasyBuild is a software build and installation framework written in Python.

open-source (GPLv2), available via PyPi and GitHub

It provides:

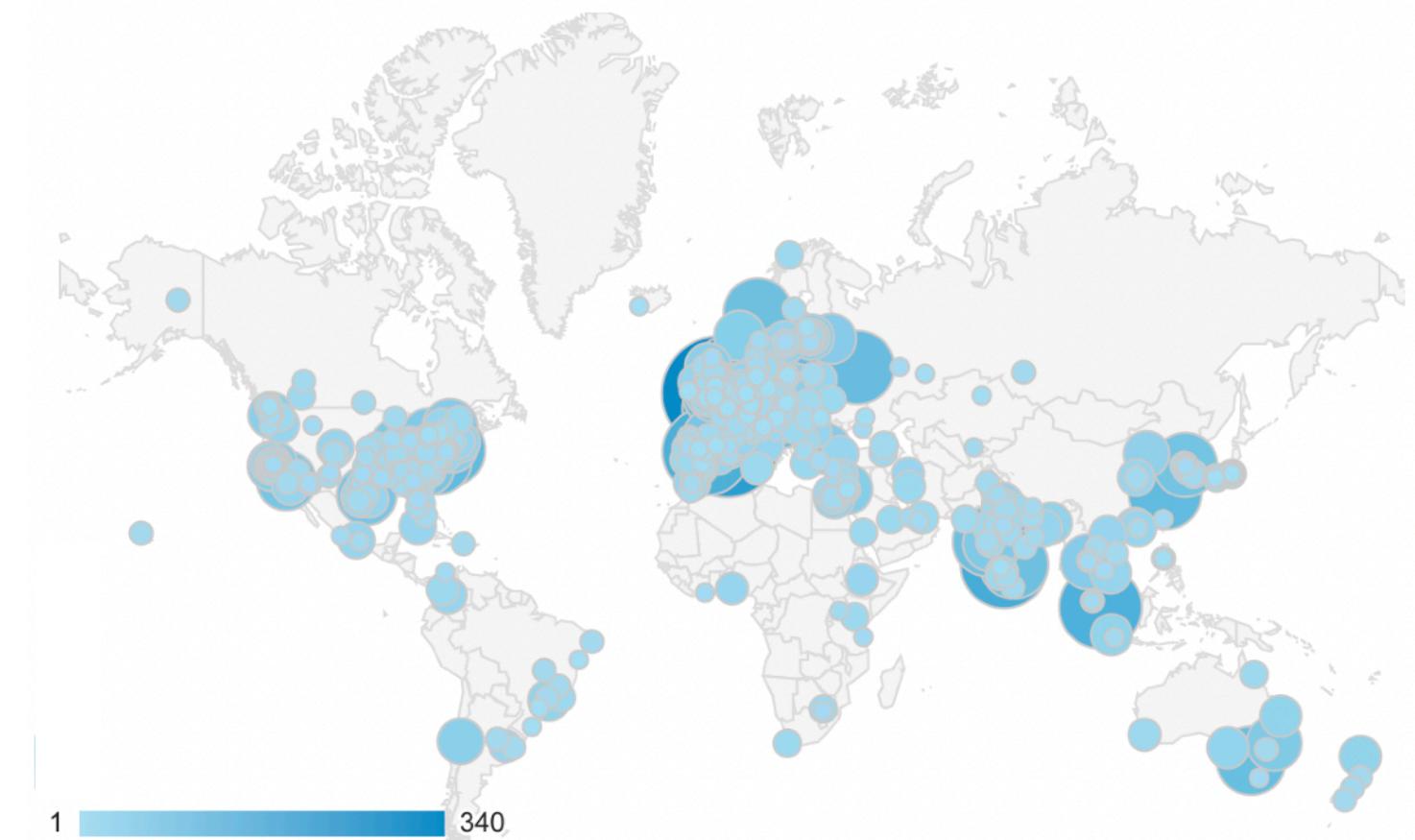
- a robust framework for implementing build procedures
- lots of supporting functionality
 - extracting, patching, executing shell commands, creating module files, ...
- modular support for compilers, libraries (MPI, BLAS/LAPACK, ...)
- modular support for custom software build procedures

easybuild Building software with ease

- developed in-house for over 3.5 years
- available on GitHub (GPLv2) since April 2012
- v1.0.0 (stable API) just released (Nov. 13th 2012)
- support for GCC and Intel compilers, ATLAS, Intel MKL, ...
- custom easyblocks available for 77 software packages
 - more being ported from our legacy version in coming weeks/months
- 338 example easyconfigs for 148 different software packages
- used in Scientific Linux (SL) 5/6 day-to-day
- Univ. of Luxembourg uses it on Debian, with great success

EasyBuild community

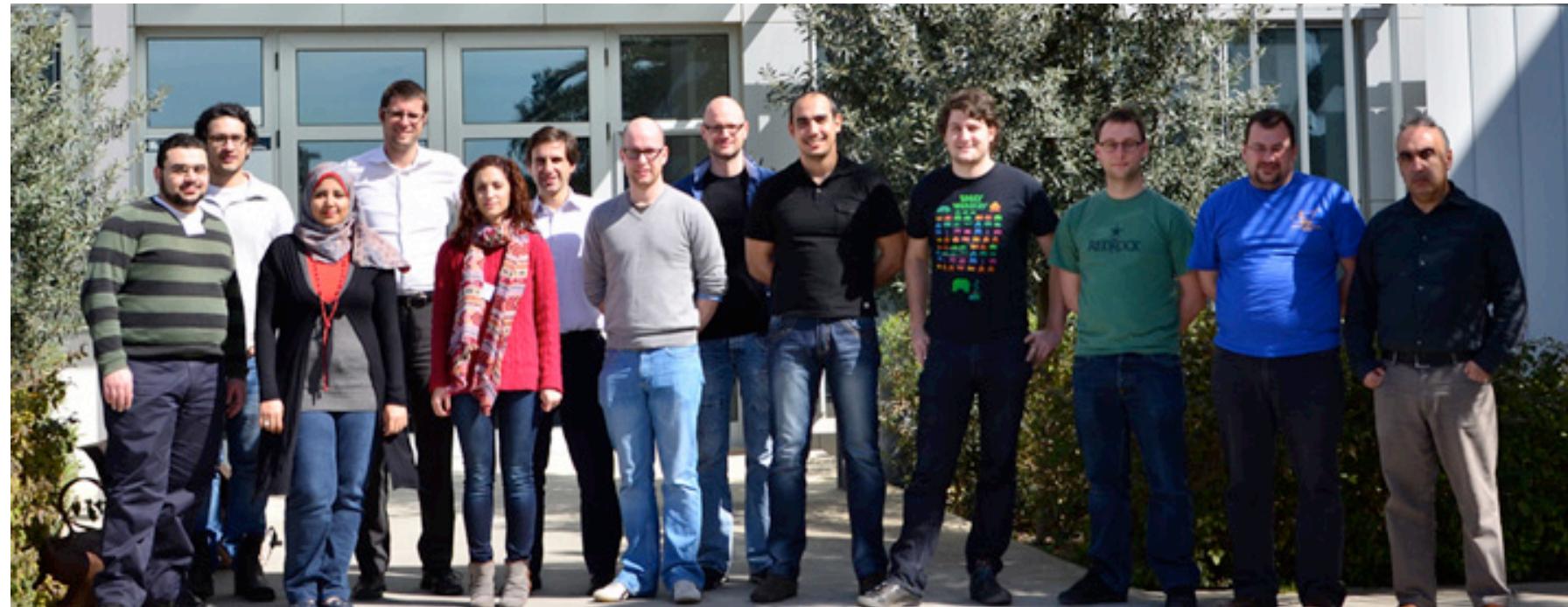
- Relatively quickly, a **community** emerged around EasyBuild
- Many HPC sites were starting to use EasyBuild, and also actively joined development!
- We started organising **hackathons**: brainstorming & discussions, extending/enhancing EasyBuild, ...
- **Bi-weekly conference calls** since Nov'13: discuss development, answer questions, ...
- **Getting Scientific Software Installed BoF sessions** at ISC and SC (2013-2015, 2018-2019)
- **HPC devrooms at FOSDEM** (2014-2022)
- **User Meetings** since 2016 - easybuild.io/eum
- **User Surveys** since 2017 - easybuild.io/user_survey
- **Tutorials** at ISC & beyond (2020-2022) - easybuild.io/tutorial



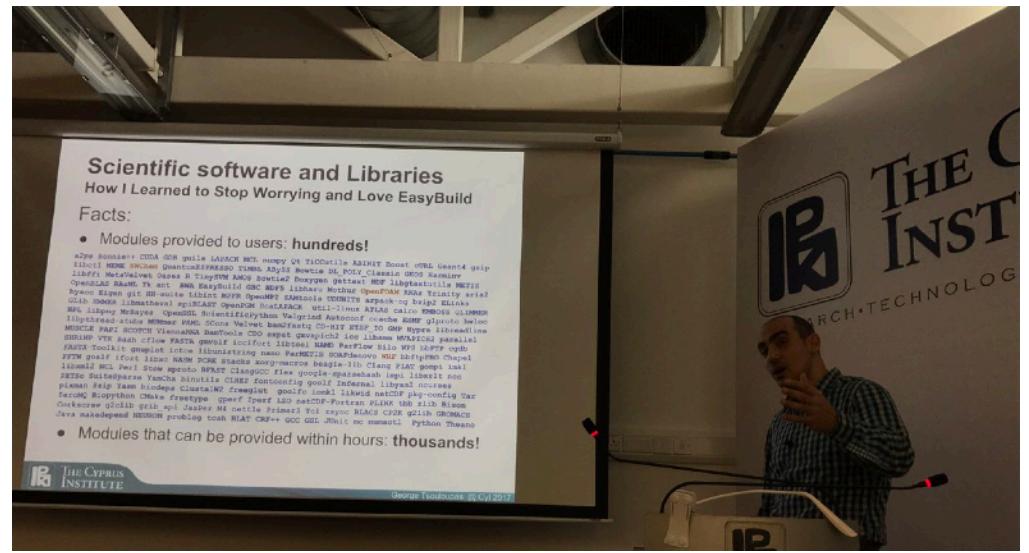
Cities from where EasyBuild docs were visited
in the last year, currently about
1,000 - 1,500 page views per week

EasyBuild hackathons

- Early adopter(s) of EasyBuild were keen to get actively involved with the project
- 1st EasyBuild hackathon was organised in Ghent in August 2012, attended by HPC-UGent team + Fotis Georgatos (Univ. of Luxembourg)
- 10 more hackathons followed in 2012-2016, all across Europe + Austin, Texas @ TACC
- Notes & slides are still available via the [EasyBuild wiki](#)



EasyBuild promotion & events



"EasyBuild saved my life"
George Tsouloupas (Cyl)
PRACE Spring School (2017)



EasyBuild coffee mugs



EasyBuild cake (SC'17)



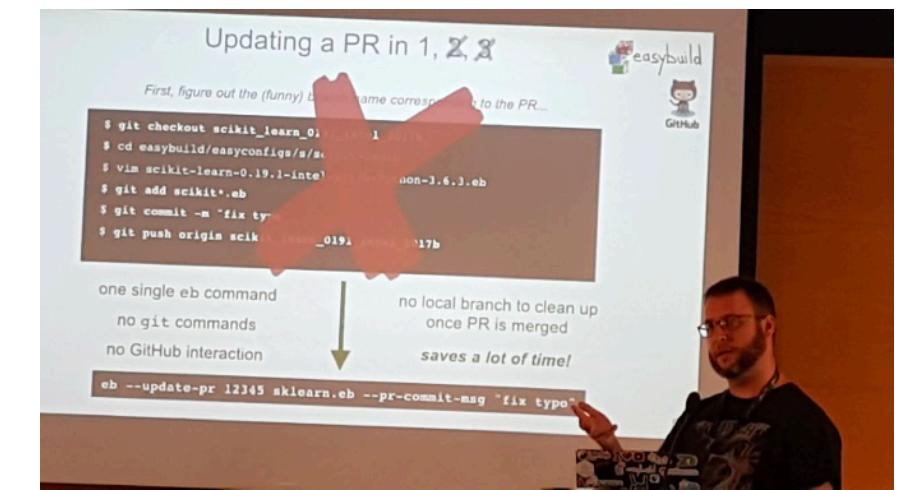
EasyBuild workshop in
Birmingham (UK) - 2018



EasyBuild maintainers @ FOSDEM'20



Cuban HPC team visiting Ghent (Jun'22)

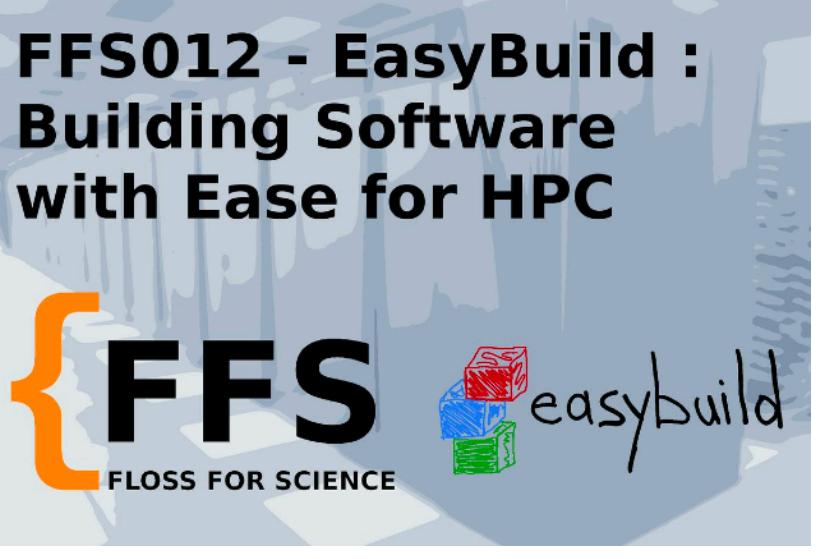


EasyBuild @ HPCKP'19

EasyBuild
718 Tweets

EasyBuild
@easy_build
EasyBuild: building software with ease, a Python framework for building and installing (scientific) software
easybuild.io Joined March 2012
62 Following 666 Followers

 @easy_build

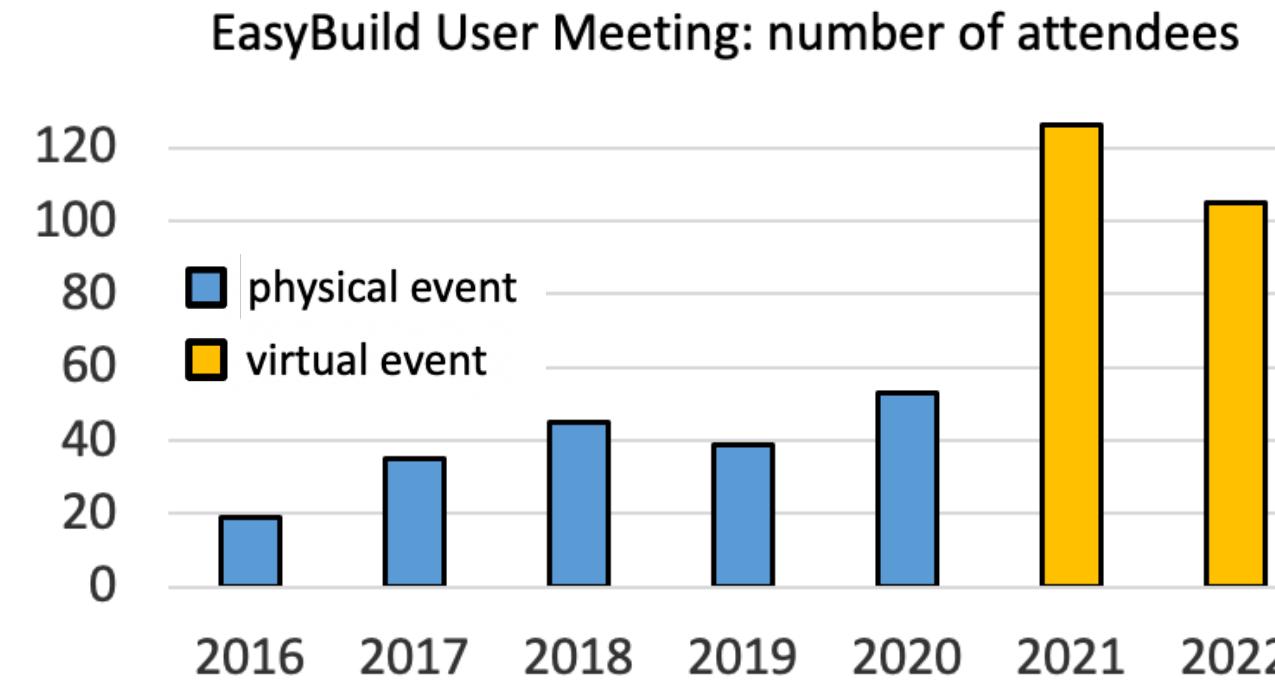


EasyBuild in FFS podcast
(Dec'18)

EasyBuild User Meetings

easybuild.io/eum

- Since 2016 the hackathons have evolved into **user meetings**
- Less focus on brainstorming and developing, more on usage of EasyBuild & other related projects
- **Very well attended**, last physical user meeting (Jan'20) had over 50 attendees!
- **All talks were live streamed and recorded**, available via the [EasyBuild YouTube channel](#)
- EUM'21 and EUM'22 were fully virtual due to COVID-19 pandemic
- **We intend to have a physical EasyBuild user meeting again in 2023!** (likely later, April-May?)



2nd EUM @ JSC (2017)



5th EasyBuild User Meeting in Barcelona (Jan'20)

Major EasyBuild features

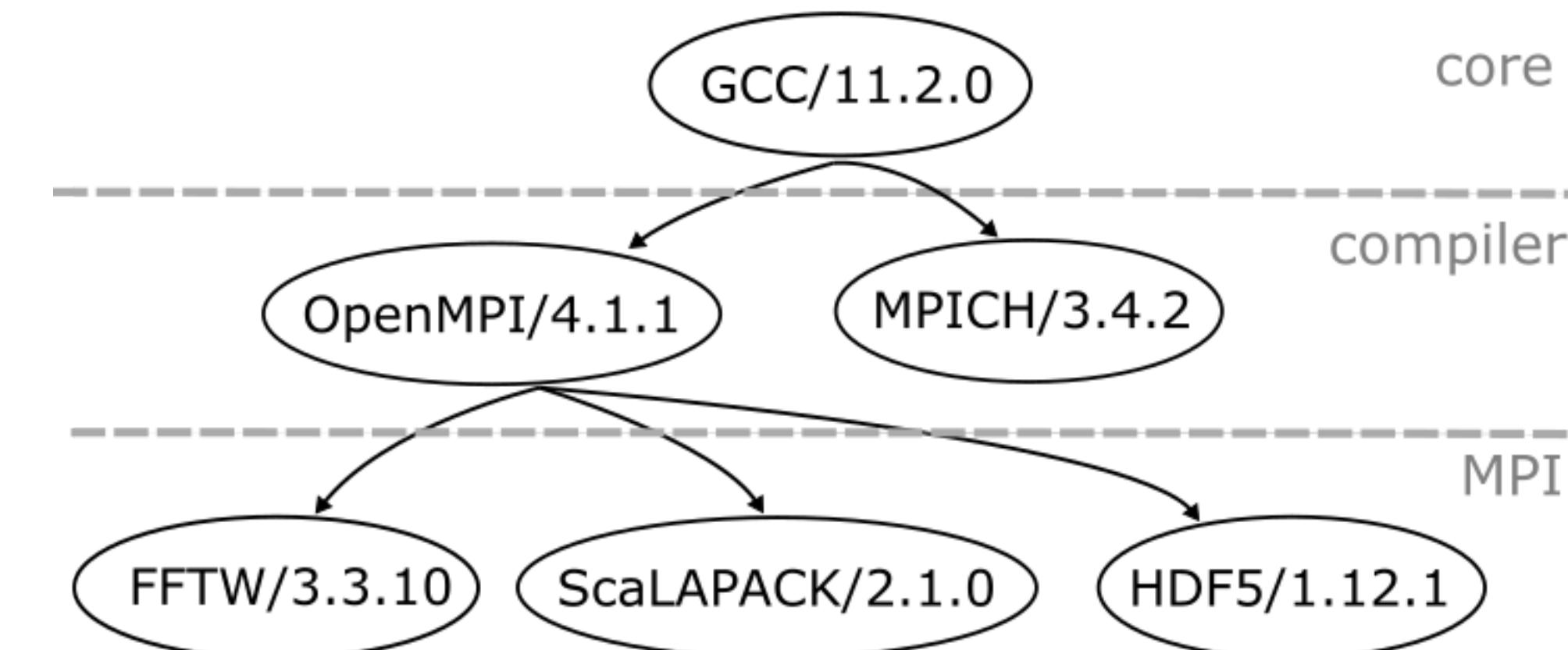
docs.easybuild.io - easybuild.io/tutorial

- Over the years, various useful features have been developed in EasyBuild
- Often driven by, or contributed by, the EasyBuild community
- Examples include:
 - Support for using a **hierarchical module naming scheme**
 - Including and verifying (SHA256) **checksums** for source files and patches
 - **Common toolchains** to focus the effort of the community: `foss`, `intel`
 - **Integration with GitHub** to significantly facilitate contribution and review process
 - **Trace output** and **extended dry run** (`eb -x`) makes EasyBuild less of a black box
 - Distributing the installation of a stack of software by **submitting (Slurm) jobs**
 - Fine-grained **hooks to customise EasyBuild** to adhere to site-specific policies
 - Integration with Rich for **colourful output + progress bars**

Hierarchical module naming schemes

easybuilders.github.io/easybuild-tutorial/2022-isc22/module_naming_schemes

- Since July 2014, EasyBuild can be configured to use a **hierarchical module naming scheme**
 - `eb --module-naming-scheme HierarchicalMNS ...`
- A module hierarchy typically consists of 3 levels (but doesn't need to be):
 - A core level for modules installed with the system compiler (like GCC)
 - A compiler level for modules installed with a particular (EasyBuild-installed) compiler
 - An MPI level for modules that are installed on top of a specific compiler + MPI library
- Benefits for end users:
 - Short module names
 - Only compatible modules can be loaded
- Using Lmod is highly recommended!



Contributing to EasyBuild is... easy!

docs.easybuild.io/en/latest/Integration_with_GitHub.html

```
$ 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

Trace output (eb --trace)

docs.easybuild.io/en/latest/Tracing_progress.html

Get a more detailed view on what EasyBuild is doing while the installation is running

```
$ eb TensorFlow-2.7.1-foss-2021b.eb --trace
...
== preparing...
>> loading toolchain module: foss/2021b
>> loading modules for build dependencies:
>> * Bazel/3.7.2-GCCcore-11.2.0
>> ...
>> loading modules for (runtime) dependencies:
>> * Python/3.9.6-GCCcore-11.2.0
>> * h5py/3.6.0-foss-2021b
>> ...
>> defining build environment for foss/2021b toolchain
...
== installing extension TensorFlow 2.7.1 (32/32)...
...
>> running command:
    [started at: 2022-10-06 14:12:38]
    [output logged in /tmp/eb-pRHwkc/easybuild-run_cmd-SOINRV.log]
    bazel ... --jobs=6 //tensorflow/tools/pip_package:build_pip_package
>> command completed: exit 0, ran in 00h41m22s
...
```

Extended dry run mode (eb -x)

docs.easybuild.io/en/latest/Extended_dry_run.html

Get a quick overview of how EasyBuild *would* install something, in a matter of seconds

```
$ eb WRF-4.3-foss-2021a-dmpar.eb -x
...
building... [DRY RUN]

[build_step method]
running command "tcsh ./compile -j 4 wrf"
(in /home/example/eb/software/WRF/4.3-foss-2021a-dmpar/WRF-4.3)
running command "tcsh ./compile -j 4 em_real"
(in /home/example/eb/software/WRF/4.3-foss-2021a-dmpar/WRF-4.3)
running command "tcsh ./compile -j 4 em_b_wave"
(in /home/example/eb/software/WRF/4.3-foss-2021a-dmpar/WRF-4.3)
...
[sanity_check_step method]
Sanity check paths - file ['files']
* WRF-4.3/main/libwrflib.a
* WRF-4.3/main/real.exe
* WRF-4.3/main/wrf.exe
Sanity check paths - (non-empty) directory ['dirs']
* WRF-4.3/main
* WRF-4.3/run
Sanity check commands
(none)
```

Site customisation via hooks

docs.easybuild.io/en/latest/Hooks.html

- EasyBuild behaviour can be customised to adhere to site policies using hooks
- A *_`_hook` Python function is called (if defined) at specific points when installing software
- These functions can run additional checks or actions, or manipulate internal data structures
- Different types of hooks: start/end hook, parse hook, pre/post-step hooks, ...
- Simple example: customising configure options for Open MPI via a pre-configure hook:

```
def pre_configure_hook(self, *args, **kwargs):  
  
    if self.name == 'OpenMPI' and '--with-verbs' in self.cfg['configopts']:   
        self.log.info("[pre-configure hook] Replacing --with-verbs with --without-verbs")  
        self.cfg['configopts'] = self.cfg['configopts'].replace('--with-verbs', '--without-verbs')
```

- Extensively used by some EasyBuild sites (cfr. [EUM'22 talk on heterogeneous MPI stack](#))

Rich output, progress bars

docs.easybuild.io/en/latest/Progress_bars.html

- **Multi-level progress bars** to show progress on downloading of files, installation steps, extensions, easyconfigs being installed, ...
- Requires having the  `rich_` Python package installed (which requires Python 3)
`pip3 install rich`
- Supported since EasyBuild 4.5.0, controlled via `--output-style` configuration option

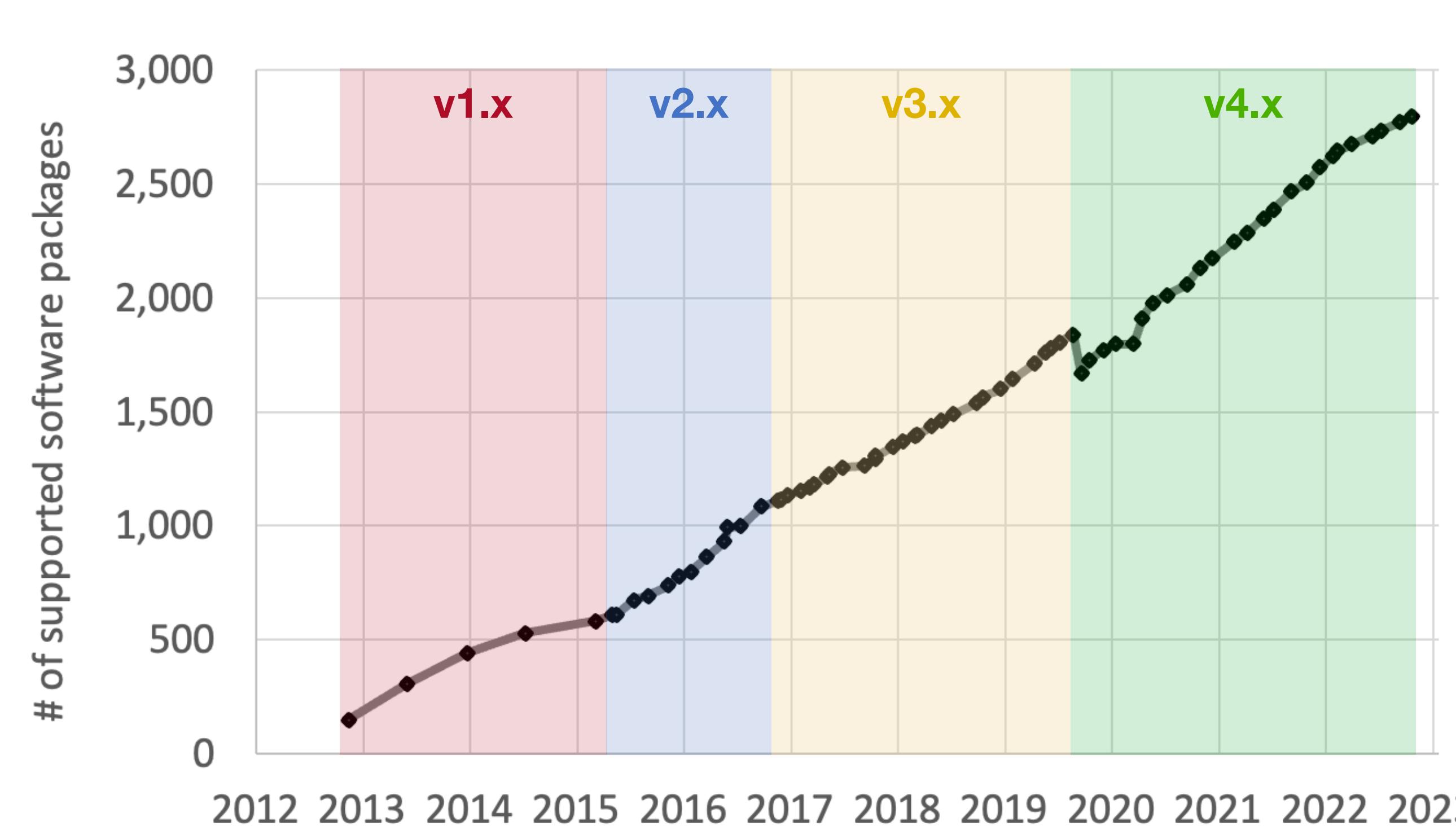
```
== fetching files...
Downloading gcc-11.2.0.tar.gz ━━━━━━━━━━━━ 52.4/143.2 MB 5.5 MB/s 0:00:17
Fetching files: 10% (1/10) ━━━━━━━━━━ 0:00:10 (gcc-11.2.0.tar.gz)
```

```
Installing 'reshape2' extension (116/990) ━━━━━━━━━━ 0:26:14
... Installing R/4.1.0-foss-2021a: taking care of extensions (9 out of 17 steps done) ━━━━━━━━━━ 0:33:48
0:36:40 2 out of 3 easyconfigs done: CMake/3.20.1-GCCcore-10.3.0 (OK), bzip2/1.0.8-GCCcore-10.3.0 (OK)
```

Software supported by EasyBuild

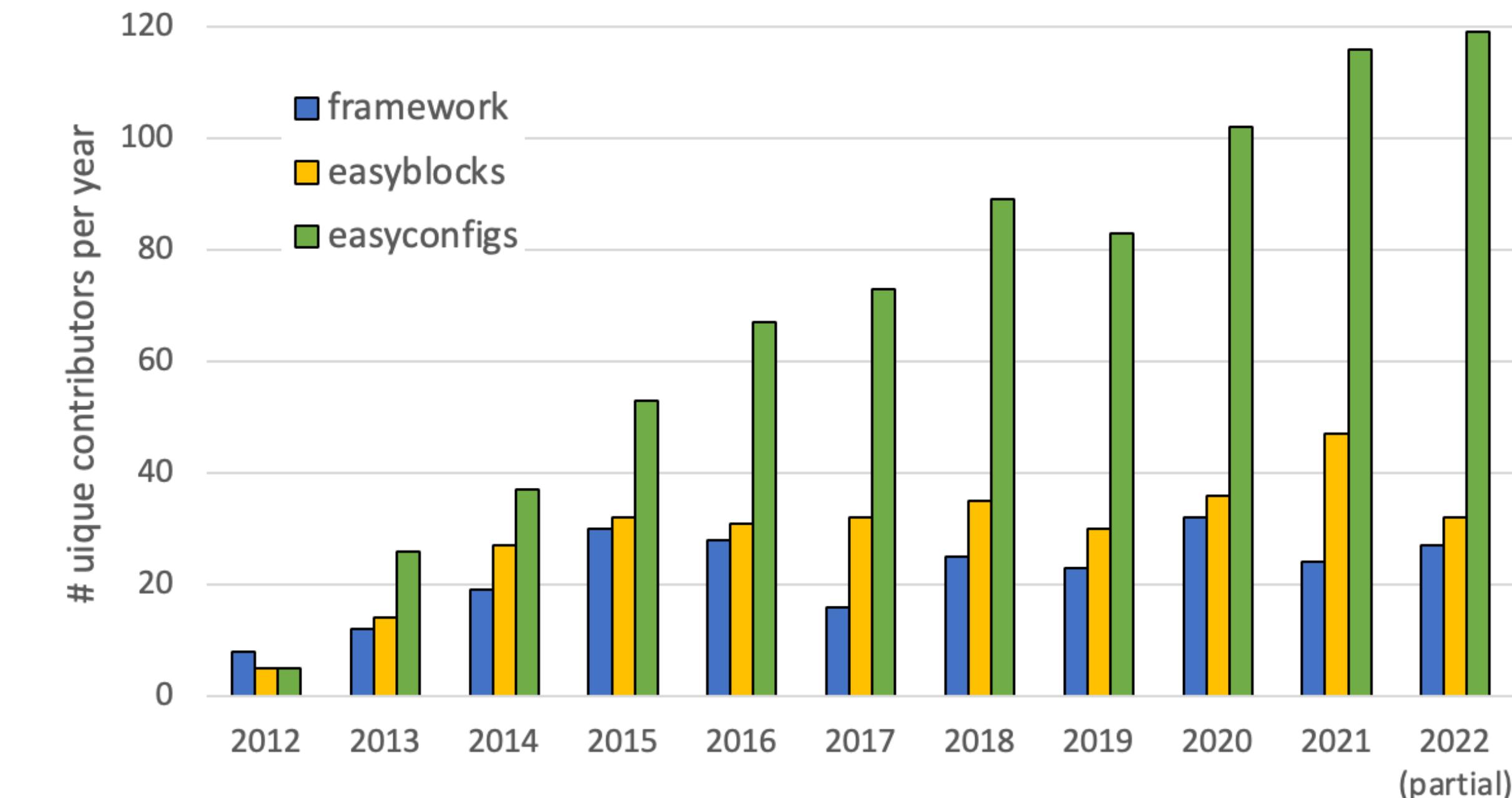
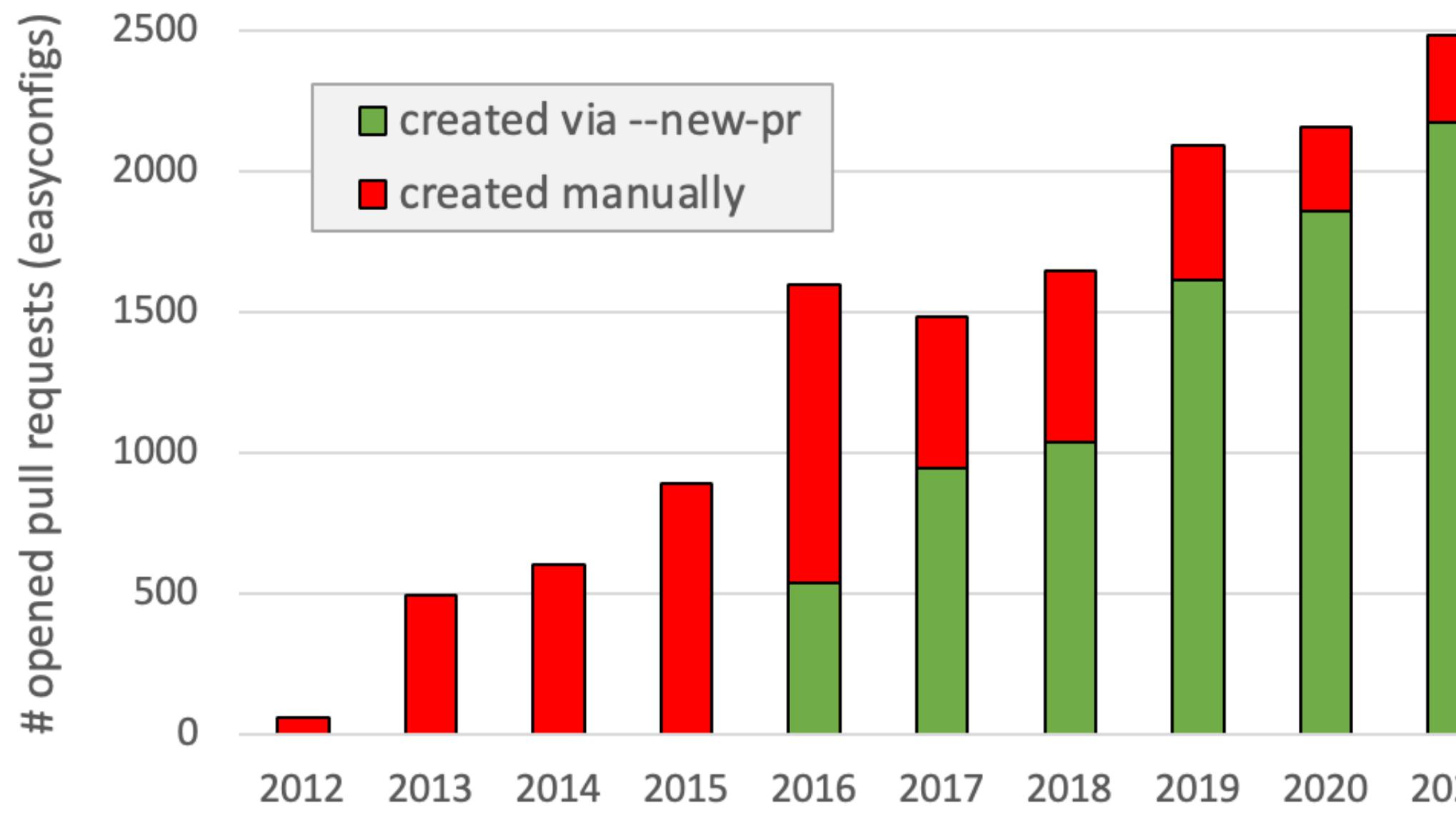
docs.easybuild.io/en/latest/version-specific/Supported_software.html

- Over the years, there is a **steady increase** in the number of supported software packages
- **Steeper upward slope in recent years:** ~400 extra software per year (excl. extensions)
- **We'll soon break the threshold of 3,000 supported software packages** (plus > 2,000 extensions)
- Small decrease in 2019 due to archiving of ancient easyconfigs in EasyBuild v4.0



EasyBuild contributors and contributions

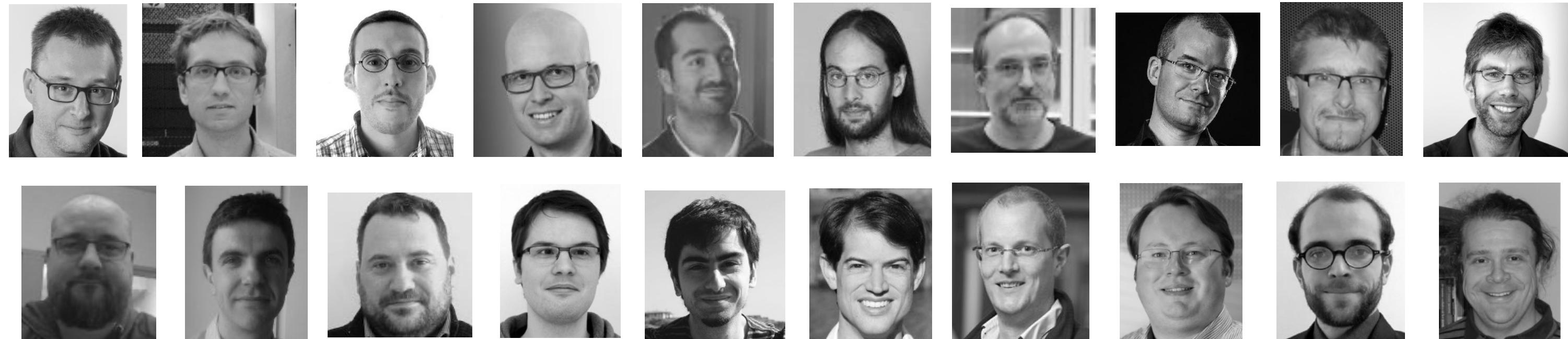
- **Growing number of community contributions:** close to 2,500 easyconfig PRs per year!
- **Close to 25,000 PRs in total:** framework: ~4,100 - easyblocks: ~2,800 - easyconfigs: > 16,600
- **Over 375 unique contributors in total** so far, across the 3 EasyBuild main repositories
- **Well over 100 unique contributors per year** (and still increasing)
- Significant impact of GitHub integration features like `eb --new-pr` (more PRs, structure, ...)



EasyBuild maintainers

docs.easybuild.io/en/latest/Maintainers.html

- Since 2017, there are multiple EasyBuild maintainers outside of the HPC-UGent team
- Currently, there are 21 EasyBuild maintainers (some are dormant or less active)
- We have a rotating maintainer-of-the-week role for handling incoming issues and PRs
- 1st EasyBuild maintainer summit (virtual) on 28-29 Sept'21 (see [notes on EasyBuild wiki](#))
- **EasyBuild would not be possible without the voluntary work done by the maintainers!**
- **Please thank an EasyBuild maintainer** if you get the chance: buy them a coffee, a beer, ...



Lessons learned

- If you decide to release open source software: be careful, you may start a community...
- **Word-of-mouth** is the best form of advertisement: give talks, tell your friends/colleagues, ...
- Even after a decade of giving talks, **some people will not know about your project yet...**
- You may have the best project ever, **you will still have to “sell” it to people**
- The best way to convince people about your project is to **let them use it first-hand**
- **Getting people to understand** what your project does and doesn’t do is ***hard***
- **New features are easily overlooked**, so point them out at every opportunity
- **Value every contribution**, regardless how “experienced” the contributor is
- **Don’t take your own experience and skills for granted**, be patient with people

Into a new decade

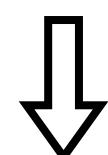
- EasyBuild has grown tremendously in the last decade
- We hope it will be even more successful in the next 10 years...
- Long-term planning, like setting out a clear road map (that we stick to), is difficult
- Short-term:
 - Complete porting of EasyBuild documentation to MarkDown + MkDocs
 - EasyBuild v4.7.0 (incl. 2022b common toolchains, enhanced support for easy stack files, ...)
 - EasyBuild v5.0 (incl. some backwards-incompatible changes)
 - 8th EasyBuild User Meeting (April-May 2023?)
- Longer term:
 - Improved error reporting when installations fail
 - Making EESSI production-ready

Revamping the EasyBuild documentation

github.com/easybuilders/easybuild-docs

- We are actively porting the existing EasyBuild documentation to MarkDown + MkDocs
- Current format (.rst + rendering with Sphinx) is holding back maintenance + contributions
- Primary focus is on porting to new format, later also review of contents - any help is welcome!
- Live local preview of documentation while editing via `mkdocs serve`
- Significantly better and faster searching functionality
- Also moving documentation sources to a dedicated repository: [easybuilders/easybuild-docs](https://github.com/easybuilders/easybuild-docs)
- We will try and make sure that existing `docs.easybuild.io` links do not get broken

```
.. _pip:  
  
Using pip to Install EasyBuild  
  
-----  
  
Since EasyBuild is released as a Python package on PyPI (https://pypi.org/project/easybuild) you can install it using ``pip``, the most commonly used tool for installing Python packages.  
  
Install EasyBuild with:  
  
pip install easybuild
```



```
## Using pip to Install EasyBuild {: #pip }  
  
Since EasyBuild is released as a [Python package on PyPI] (https://pypi.org/project/easybuild) you can install it using `pip`, the most commonly used tool for installing Python packages.  
  
Install EasyBuild with:  
  
``` shell  
pip install easybuild
```
```

EasyBuild v4.6.2 documentation (release 20221021.0) >

 easybuild

Quick search Go

Table of Contents

- What is EasyBuild?
- Concepts and terminology
- Typical workflow example: building and installing WRF

Installing EasyBuild

- Requirements
- Using pip to Install EasyBuild

EasyBuild - building software with ease

Search

Installing EasyBuild

EasyBuild is Python software, so there are a couple of ways to install it.

We recommend installing EasyBuild using `pip`. This method is described at [Using pip to Install EasyBuild](#).

It is also possible to install EasyBuild as a module. To do this, use the 3-step procedure outlined at [EasyBuild with EasyBuild](#).

Do take into account the required and optional dependencies (see [Requirements and Dependencies](#)).

Notes on other ways of installing EasyBuild are available under section [Alternative installation methods](#).

Contents

- [Installing EasyBuild](#)
 - Requirements

EasyBuild - building software with ease

Home

What is EasyBuild?

Terminology

Installation

Requirements

Using pip to Install EasyBuild

Sanity check

Updating an existing EasyBuild installation

Additional pip install options

EasyBuild is Python software, so there are a couple of ways to install it.

We recommend installing EasyBuild using `pip`. This method is described at [Using pip to Install EasyBuild](#).

It is also possible to install EasyBuild as a module. To do this, use the 3-step procedure outlined at [EasyBuild with EasyBuild](#).

Do take into account the required and optional dependencies (see [Requirements and Dependencies](#)).

Notes on other ways of installing EasyBuild are available under section [Alternative installation methods](#).

EasyBuild v5.0

easybuilders/easybuild/wiki/Breaking-changes-being-considered-for-EasyBuild-v5.0

- We will soon start working towards the release of EasyBuild v5.0
- Planned backwards-incompatible changes:
 - **Drop support for running EasyBuild on top of Python 2.7** (EOL since 1 Jan 2020)
 - **Archive ancient easyconfigs** (using GCC < 8.0 or a common toolchain < 2019a)
 - **Changed defaults in PythonPackage easyblock: use_pip + sanity_pip_check by default**
- Major changes (which may already be implemented prior to EasyBuild v5.0):
 - **New run function for running shell commands**, with cleaner API (and better error reporting)
 - **Vendorizing of LooseVersion** (since distutils will be removed from Python stdlib in Python 3.12)
- Several other ideas, to be discussed among EasyBuild maintainers
- **Additional suggestions and ideas welcome:** open an issue, discuss on Slack or mailing list, ...)

Challenges ahead

The landscape of computational science is changing (fast)...

- **Explosion of available scientific software applications**
 - Broader scope in terms of scientific domains (bioinformatics, AI, ...)
 - Fuelled by recent shifts in scientific community + pressure to publish code
 - Wider adoption of HPC across scientific domains (GPUs in bioinformatics, ...)
- **Increasing interest in using cloud infrastructure** (both private and commercial)
 - Hard to beat the provided flexibility and scale
 - Should not imply sacrificing performance for mobility of compute (see containers, conda, ...)
- **Increasing variety in processor (micro)architectures**
 - Intel + AMD, ARM, POWER, soon also RISC-V (cfr. European Processor Initiative, ...)
 - In addition, GPUs (NVIDIA, AMD, soon Intel?) and other accelerators (TPUs, IPUs, ...)
- **Shift towards Clang-based compilers** (Intel, AMD, ...)

Beyond EasyBuild with EESSI

- European Environment for Scientific Software Installations
- Collaborative project, by and for the computational science community
- Shared central stack of (optimised) scientific software installations
- Uniform way of providing software to users, regardless of system they use
- Should work regardless of OS and system architecture (HPC, cloud, ...)
- Focus on performance, automation, testing, collaboration
- Funded effort via EuroHPC project *MultiXscale* (to start in 2023)



EESSI
EUROPEAN ENVIRONMENT FOR
SCIENTIFIC SOFTWARE INSTALLATIONS

www.eessi-hpc.org
eessi.github.io/docs (try out the pilot repository!)
paper: doi.org/10.1002/spe.3075

High-level overview of EESSI

ReFrame

Testing

Host OS
provides
network &
GPU drivers,
resource
manager
(Slurm), ...

host operating system (any Linux distribution)



Software layer
optimized installations of applications + dependencies

Compatibility layer
levelling the ground across Linux distros
(“*containers without the containing*”)

Filesystem layer
global distribution of the software stack



Heavily inspired by

Digital Research
Alliance of Canada
software stack

From zero to science in 3 steps

- Step 1: Get access to EESSI CernVM-FS repository, either through:
 - system-wide CernVM-FS installation (requires admin privileges)
 - container with CernVM-FS + EESSI configuration pre-installed
- Step 2: Set up environment: source EESSI init script
- Step 3: Load module(s) and run!



eessi.github.io/docs/pilot

github.com/eessi/eessi-demo

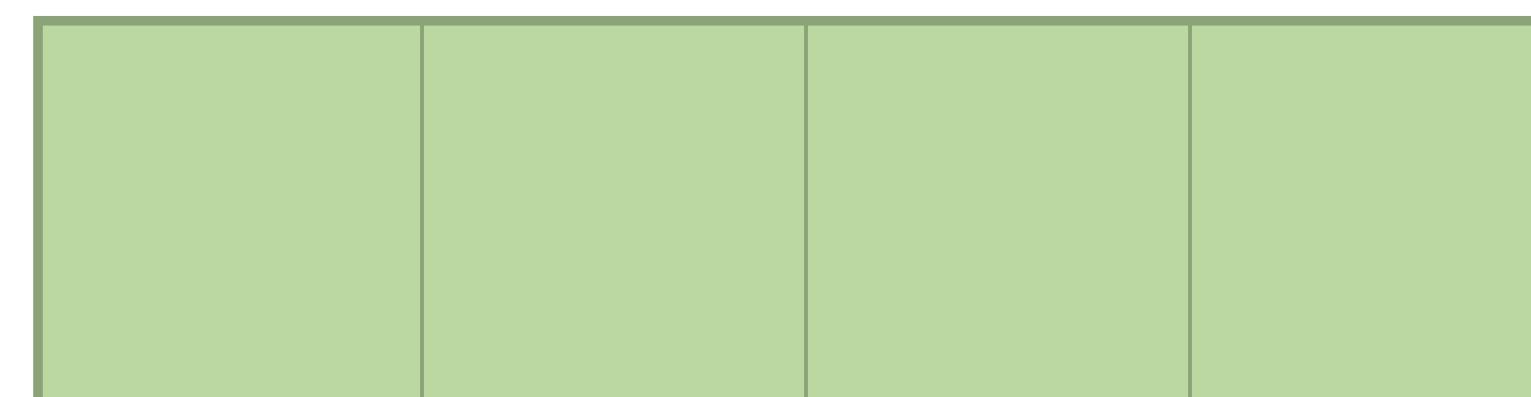
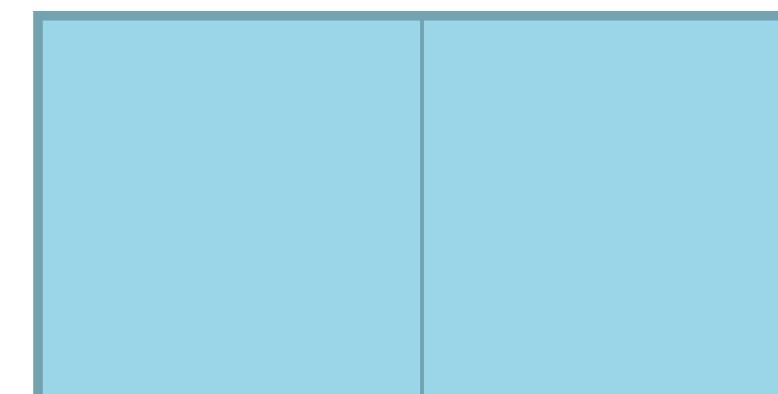
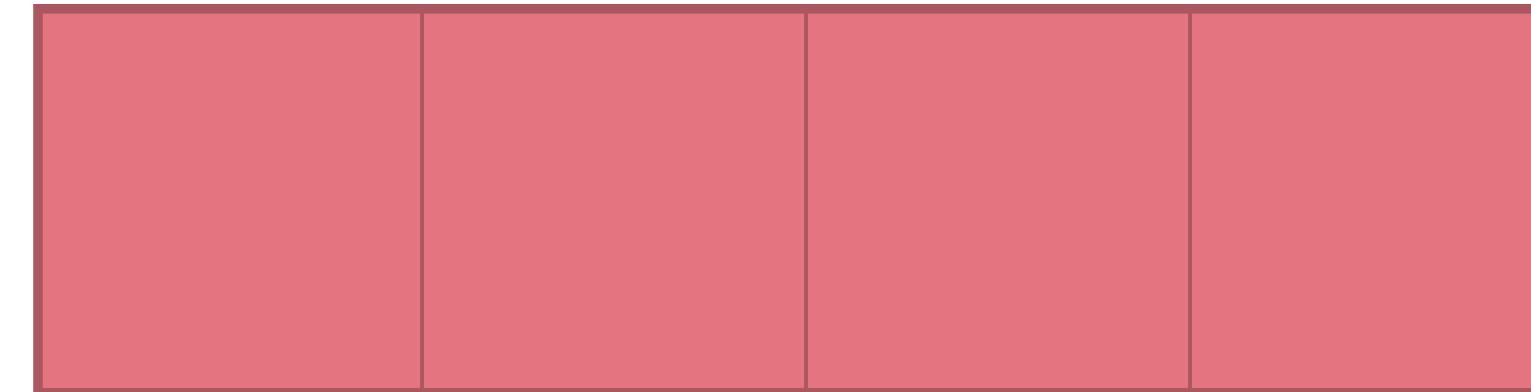
```
# Assumed status: EESSI is accessible (CernVM-FS installed + EESSI configuration in place)
```

```
# Step 2: set up environment (CPU architecture is detected automatically)  
$ source /cvmfs/pilot.eessi-hpc.org/latest/init/bash
```

```
# Step 3: load module(s) to activate software (check with 'module avail'), and run!  
[EESSI pilot 2021.06] $ module load GROMACS  
[EESSI pilot 2021.06] $ gmx mdrun ...
```

And now for a small surprise...

- After (over) a decade of EasyBuild, it is time for something new...
- A particular aspect of EasyBuild is due for a refresh...
- Something that has served us well until now, but could be improved a lot...
- An evolution of what was there before...
- To make things a bit more professional...
- It may cause some confusion at first...
- But it will grow on you eventually...



EASYBUILD.io
building software with ease

easybuild.io - docs.easybuild.io - easybuild.io/tutorial

easybuild.io/join-slack - youtube.com/c/easybuilders - twitter.com/easy_build