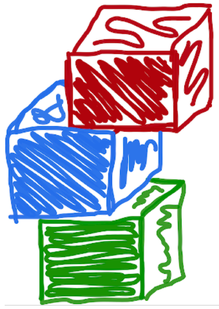


easybuild

EESSI

EUROPEAN ENVIRONMENT FOR SCIENTIFIC SOFTWARE INSTALLATIONS

Getting Scientific Software Installed From EasyBuild to EESSI

Kenneth Hoste (HPC-UGent)

24 August 2022 (virtual)

RISC2 Webinar Series | HPC System & Tools

whoami

kenneth.hoste@ugent.be

@boegel (GitHub)

@kehoste (Twitter)

EasyBuild Slack

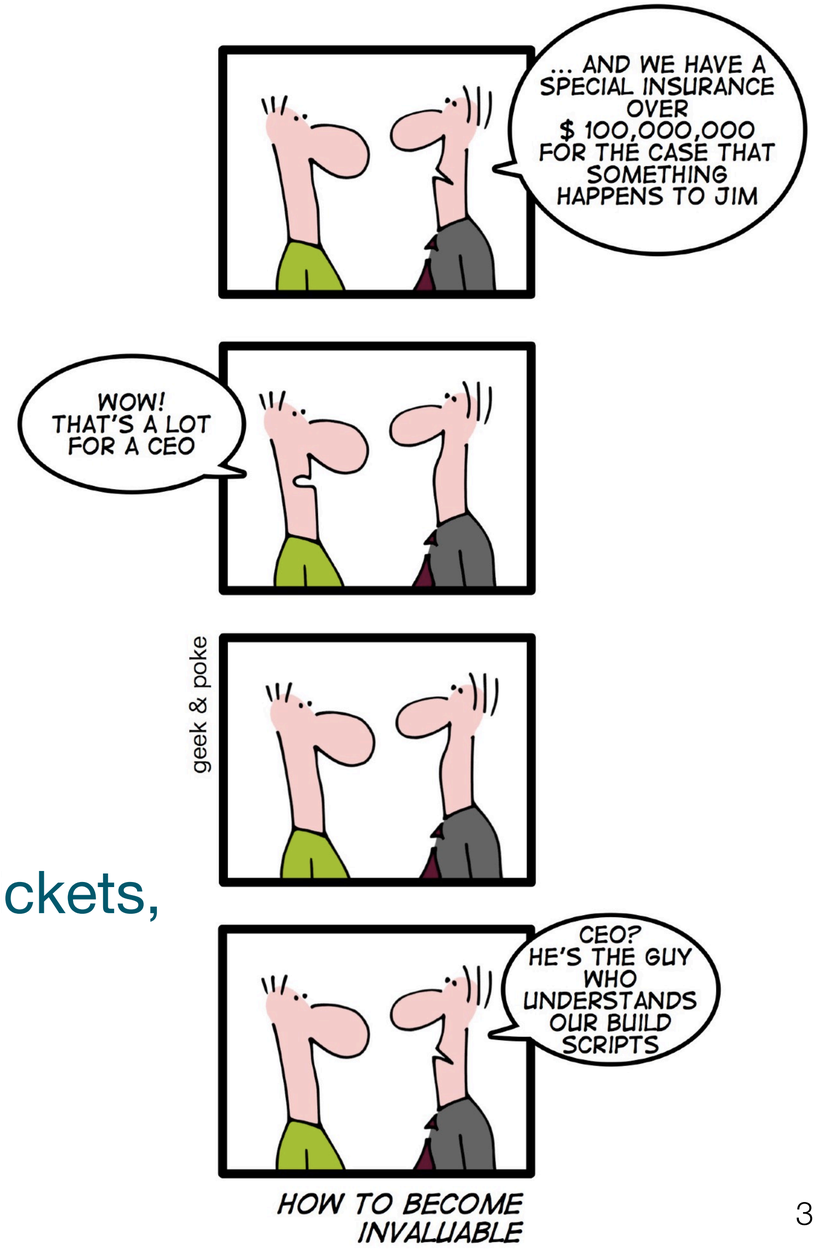
LinkedIn

- Computer scientist with Masters + PhD from Ghent University (BE)
- Joined HPC team at Ghent University in October 2010
- Main tasks: user support, incl. installation of scientific software
- Inherited maintenance of EasyBuild in 2011
- Slowly also became **EasyBuild lead developer & release manager** ...
- I like beer, loud music, FOSS (Free & Open Source Software), dad jokes
- I don't like compiler errors, CMake, SCons, Bazel, TensorFlow, OpenFOAM, ...

Getting Scientific Software Installed

Installation of scientific software is (still) a tremendous problem for HPC sites all around the world.

- Ideally built from source - performance is key!
- Tedious, time-consuming, frustrating,
  sometimes simply not worth the (manual) effort...
- **Huge burden on researchers & HPC support teams**
- Installation requests are ~15% of HPC-UGent helpdesk tickets,
  but they consume ***way*** more than 15% of our time...
- Very little collaboration among HPC sites (until recently)



http://geekandpoke.typepad.com/geekandpoke/2010/05/how-to-become-invaluable.html

Common issues with scientific software

Researchers focus on the *science* behind the software they implement,
and care little about software engineering, tools, build procedure, portability, ...

Scientists are (typically) not software developers or system administrators
(nor should they be!)

> *“If we would know what we are doing, it would not be called ‘research’.”*

This results in:

- use of non-standard build systems (or broken ones)
- "creative" software versioning (or no versions at all)
- dependency hell on steroids
- interactive installation scripts
- hardcoded parameters (compilers, libraries, paths, ...)
- poor/outdated/missing/incorrect documentation

in a nutshell

easybuild.io

docs.easybuild.io

youtube.com/c/easybuilders

twitter.com/easy_build

easybuild.io/tutorial

easybuild.io/eum

easybuild.io/tech-talks

- **Tool to get scientific software installed** (preferably from source)
- Focus on Linux & HPC systems, specific attention to ***performance*** of installed software
- Implemented in Python, supports both Python 2.7 and 3.5+
- Integrates with environment modules tool (Tcl-based Environment Modules + Lmod)
- **GPLv2 open source license**, available via GitHub and Python Package Index (PyPI)
- Supports different compiler toolchains + over 2,700 software packages (excl. extensions)
- Created by HPC-UGent team, now supported & developed by a **worldwide community**...

Installing TensorFlow from source, with ease

easybuild.io

```
$ eb TensorFlow-2.7.1-foss-2021b-CUDA-11.4.1.eb
== temporary log file in case of crash /tmp/eb-GyvPHx/easybuild-U1TkEI.log
== processing EasyBuild easyconfig TensorFlow-2.7.1-foss-2021b-CUDA-11.4.1.eb
== building and installing TensorFlow/2.7.1-foss-2021b-CUDA-11.4.1...
== 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 /software/TensorFlow/2.7.1-…
== 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.
```

High-level feature summary (1/2)

easybuild.io

- **Fully autonomously building and installing (scientific) software**
- Automatic generation of environment module files (Tcl or Lua syntax)
- Thorough logging of executed installation procedure (for future reference)
- Archiving of easyconfigs ("build recipes") and patch files
- **Highly configurable**, via configuration files + environment variables + command line
- Actively developed, frequent stable releases (roughly every 6-8 weeks)

High-level feature summary (2/2)

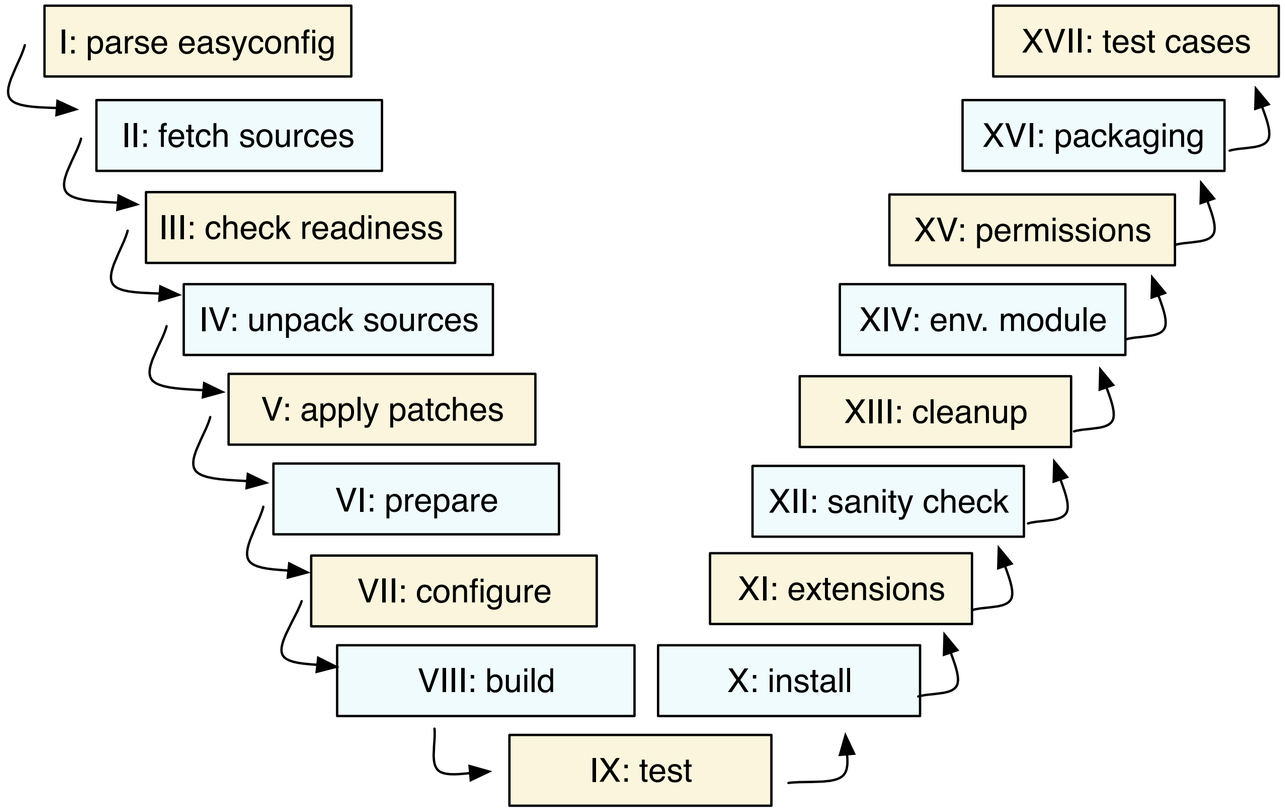
easybuild.io

- Support for using a custom module naming scheme (incl. hierarchical)
- **Dynamically extendable** with additional easyblocks, toolchains, etc.
- **Transparency** via support for 'dry run' installation & trace output
- **Customisable** through implementation of hooks (Python functions)
- Extensively tested: unit tests, thorough testing of easyconfig PRs, regression testing
- **Integration with GitHub**, Slurm, FPM, Docker, Singularity, Rich, ...
- **Worldwide collaboration** across HPC sites (JSC, CSCS, Compute Canada, LUMI, ...)

Step-wise installation procedure

easybuild.io

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 environment variables
- 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

10+ years of easybuild

easybuild.io

(April 2012)
First public release of EasyBuild
+ **start of EasyBuild community**

(Nov. 2012)
EasyBuild v1.0 (released *at* SC'12)

(mid-2009)
EasyBuild is created
by Stijn De Weirdt
(HPC-UGent tech lead)
as an in-house project

in-house development
(incl. redesigns, test suite, ...)

(Nov. 2014)
Proper documentation
easybuild.readthedocs.org

(March 2015)
EasyBuild v2.0
Backwards-incompatible changes

(2013 - 2019)
Getting Scientific Software Installed
SC/ISC Birds-of-a-Feather sessions

(2012 - 2016)
11 EasyBuild hackathons
across Europe (Univ. of Cyprus,
JSC, CSC.fi, CSCS, ...) + TACC

(Nov. 2016)
EasyBuild v3.0
Lmod as default modules tool
Stable GitHub integration

(2015)
Integration with Cray PE
EasyBuild @ cscs

(2016 - 2022)
7 EasyBuild User Meetings
(easybuild.io/eum**)**
@ Ghent, Jülich, Amsterdam,
Louvain-la-Neuve, Barcelona
+ 2x via Zoom/YouTube/Slack

(Sept. 2019)
EasyBuild v4.0
Compatible with Python 3
Only requires Python std. lib.

(Jan. 2021)
EasyBuild selected as
installation tool
for LUMI
(see EUM'22 talk)

(2020 - 2022)
EasyBuild tutorials at ISC
(easybuild.io/tutorial**)**
Online tutorial in June'20
Half-day tutorial @ ISC'21
Half-day tutorial @ ISC'22

ISC High Performance
The HPC Event.

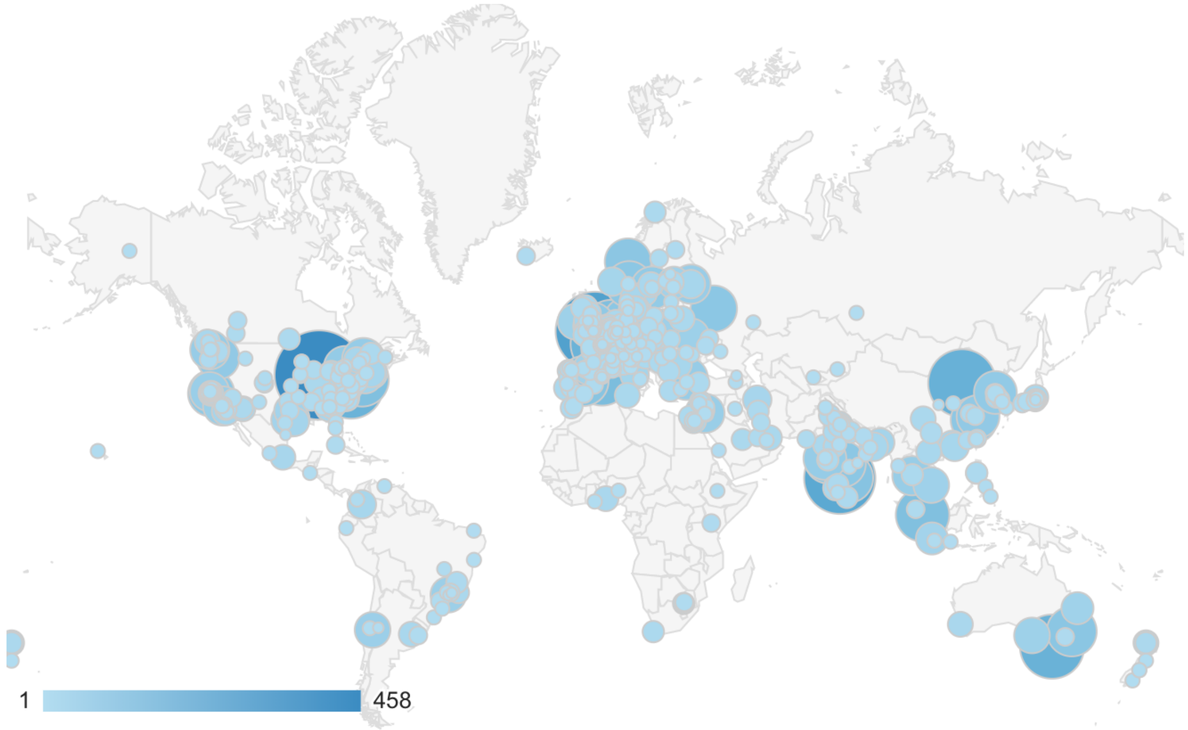
EasyBuild tutorial

easybuild
easybuild.io

- Extensive introductory tutorial on EasyBuild is available at **easybuild.io/tutorial**
- Was presented at ISC'20, **ISC'21 (virtual - recording available)**, and ISC'22 conferences
- Covers terminology, installation and configuration, basic usage, troubleshooting, …
- **Includes hands-on exercises** (and solutions, don't peek!)
- More extended version was presented for LUMI User Support Team (recording available)
  - Also covers implementing easyblocks, customising EasyBuild via hooks, …

The EasyBuild community

easybuild.io



- EasyBuild is used by HPC sites all around the world!
- Map shows from where and how often EasyBuild documentation was accessed (in 1 year)
- Worldwide team of EasyBuild maintainers: Canada, Europe + UK, Singapore, …

The EasyBuild community

easybuild.io

5th EasyBuild User Meeting in Barcelona (Jan 2020)

- **EasyBuild User Meeting (EUM)** since 2016: mix of presentations, discussions, tutorials, …
- Topics include both EasyBuild itself, and related topics (other projects, HPC trends, emerging topics, …)
- **Remote participation possible**: live-streamed (and recorded) presentations, live Q&A, …
- All presentations of previous user meetings are available on the EasyBuild YouTube channel
- More information via https://easybuild.io/eum - EUM'23 will be planned soon!

The EasyBuild community

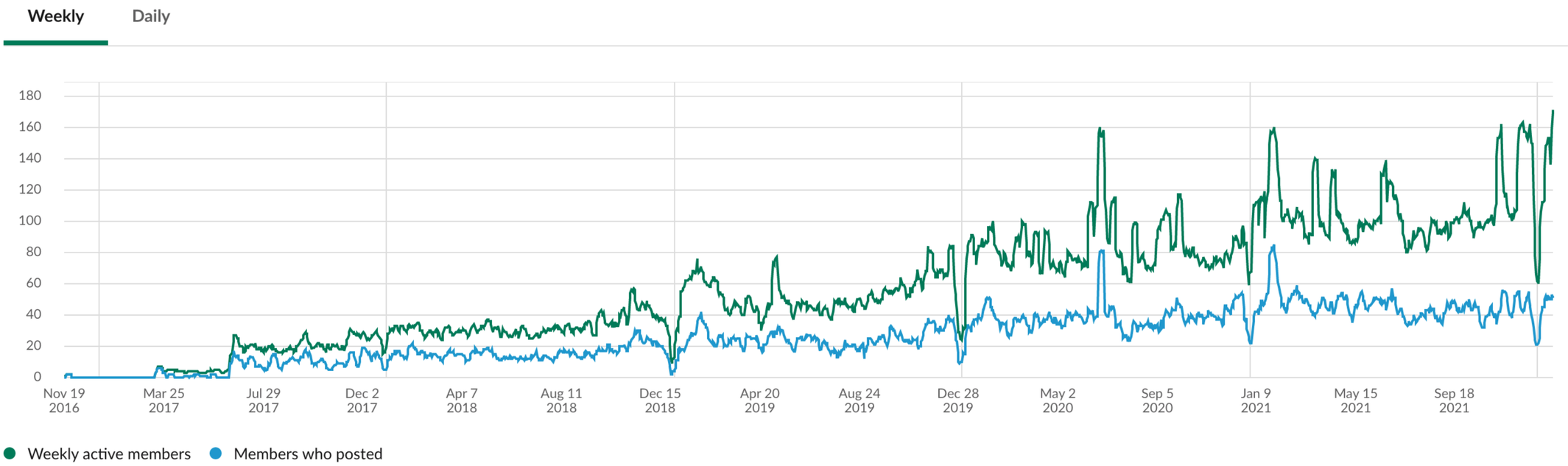
easybuild.io

HPC-UGent site visit by Cuban HPC system administrators (June 2022)

(see also: "The Birth of HPC Cuba" presentation at FOSDEM'17)

- Dedicated sessions (tutorials, updates on recent developments, …) on a regular basis
- Both by us (HPC-UGent) and other enthusiastic EasyBuild users/contributors/maintainers

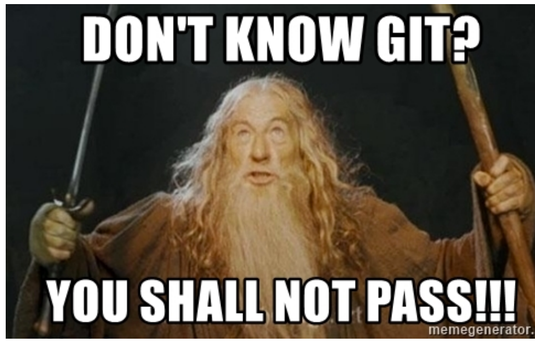
- **Get in touch with the EasyBuild community via the Slack channel!**
- Since 2017, close to 650 members, steadily growing, around the clock support
- **Join via easybuild.io/join-slack**
- Or join the EasyBuild mailing list (more info here)



GitHub integration

easybuild.io

- Interest in EasyBuild community to contribute back started increasing fast in 2015
- Resulted in more pull requests (by more contributors), of wildly varying style & quality
- **Big burden on EasyBuild maintainers** to process incoming contributions (review + test)
- In addition, lots of wannabe contributors were struggling with the Git contribution workflow…
- Late 2015, **GitHub integration features were implemented in EasyBuild**
- Makes it trivial to test open pull requests, create/update pull requests, etc. straight from the EasyBuild command line!
- More info: docs.easybuild.io/en/latest/Integration_with_GitHub.html



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

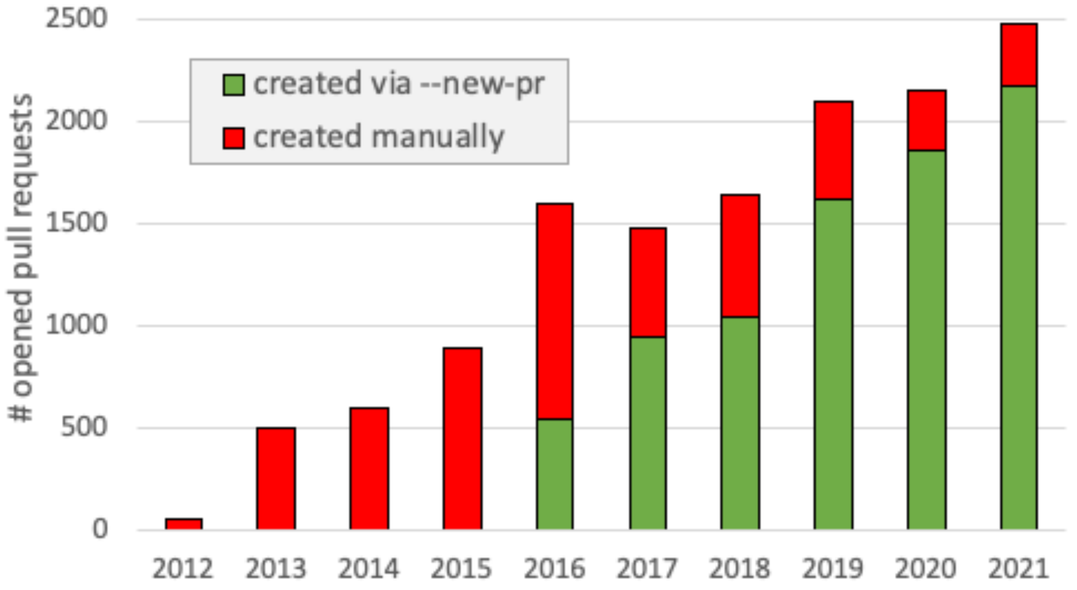
saves a lot of time!

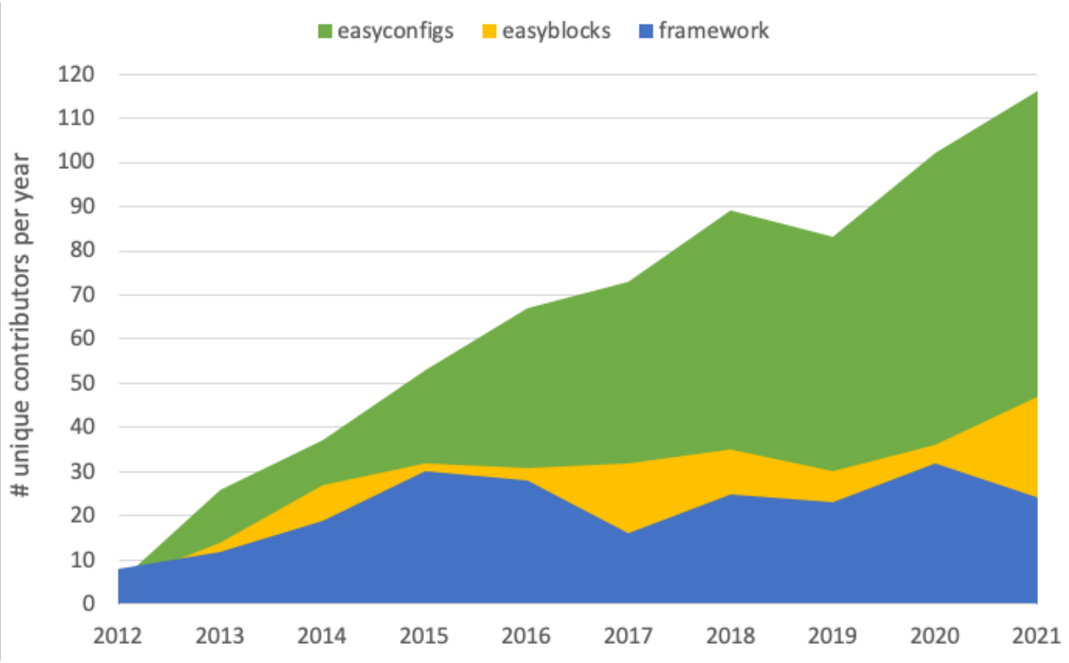
```
eb --new-pr sklearn.eb
```

Community contributions

easybuild.io

- **Growing number of community contributions**: close to 2,500 easyconfig PRs per year!
- Over 100 unique contributors per year (and steadily increasing)
- Significant impact of GitHub integration features like `eb --new-pr`

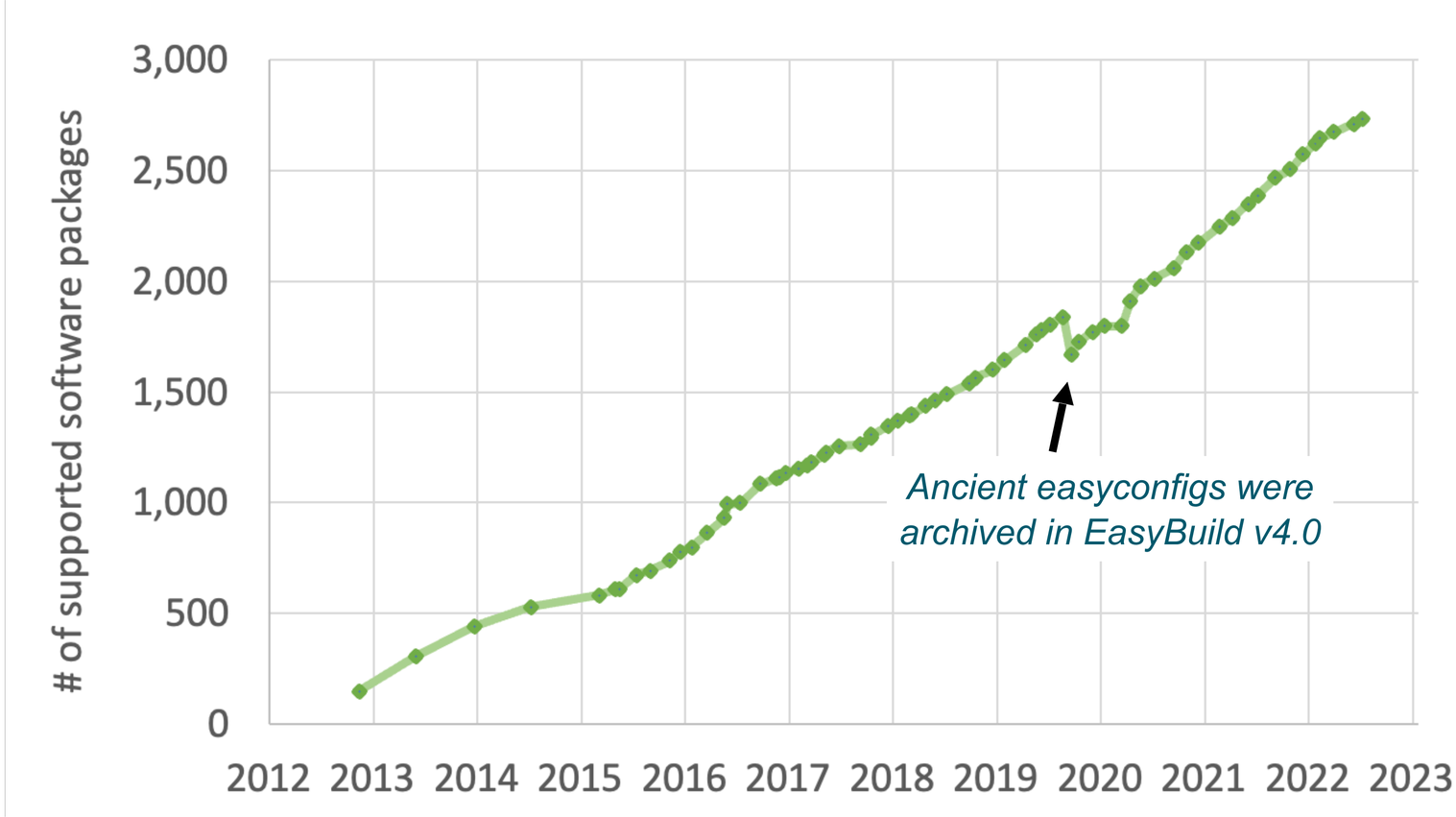




Community contributions

easybuild.io

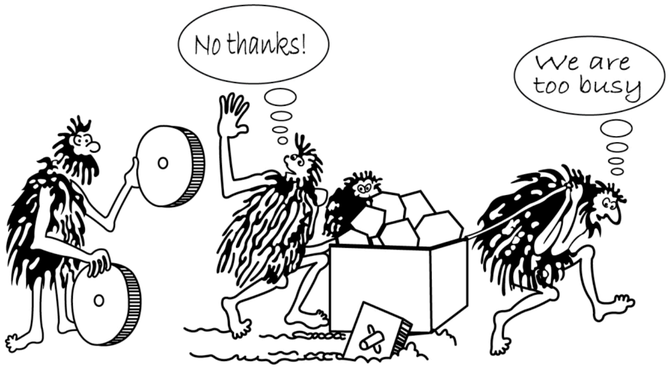
Evolution of number of supported software packages in EasyBuild (excl. extensions)



Over 2,700 different software supported by EasyBuild v4.6.0 (plus ~2,000 extensions)

Stable upward trend, no sign of slowing down...

Landscape of computational science is changing



- **Explosion of available scientific software applications**
  - Broader scope in terms of scientific domains (bioinformatics, AI, ...)
  - Fueled by recent shifts in scientific community
    - Pressure to publish code along with research articles + popularity of GitHub
    - Wider adoption of HPC across scientific domains (GPUs in bioinformatics, ...)
- **Increasing interest in using cloud infrastructure** (both private and commercial)
- **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, ...)
- **In strong contrast: available manpower in HPC support teams...**

It is time to take the next step...

- EasyBuild helps a lot with managing scientific software stacks, but it's not enough anymore...
- **HPC sites *are still* losing way too much time in getting scientific software installed**
  - Installation requests for new software often require significant investment & expertise
  - Problems occur due to site-specific differences in system setup & configuration
  - Small details (OS packages, disk partitioning, ...) can cause trouble
- **Situation is getting *worse*, not better**: more software, increasing variety in hardware, ...
- **There is a lot of potential for more extensive collaboration beyond installing software**
  - Ensuring correctness of installations w.r.t. scientific results they produce
  - Evaluating performance of provided installations + performance monitoring
  - ReFrame developed by CSCS is definitely a step in the right direction!

Introducing EESSI...

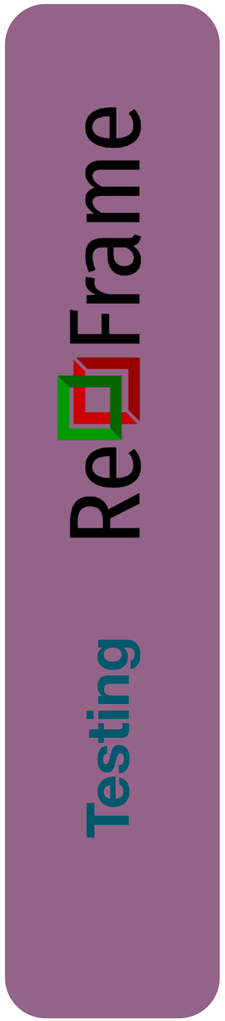
- *European Environment for Scientific Software Installations*
- Collaborative project, by and for the computational science community
- **Shared central stack of (optimized) scientific software installations**
- Uniform way of providing software to users, regardless of system they use
- Should work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

EESSI
EUROPEAN ENVIRONMENT FOR
SCIENTIFIC SOFTWARE INSTALLATIONS

www.eessi-hpc.org

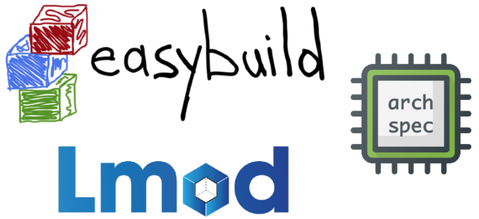
eessi.github.io/docs (try out pilot repository!)

High-level overview of EESSI



Software layer

optimized installations of applications + dependencies



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

Compatibility layer

levelling the ground across Linux distros

("containers without the containing")

Filesystem layer

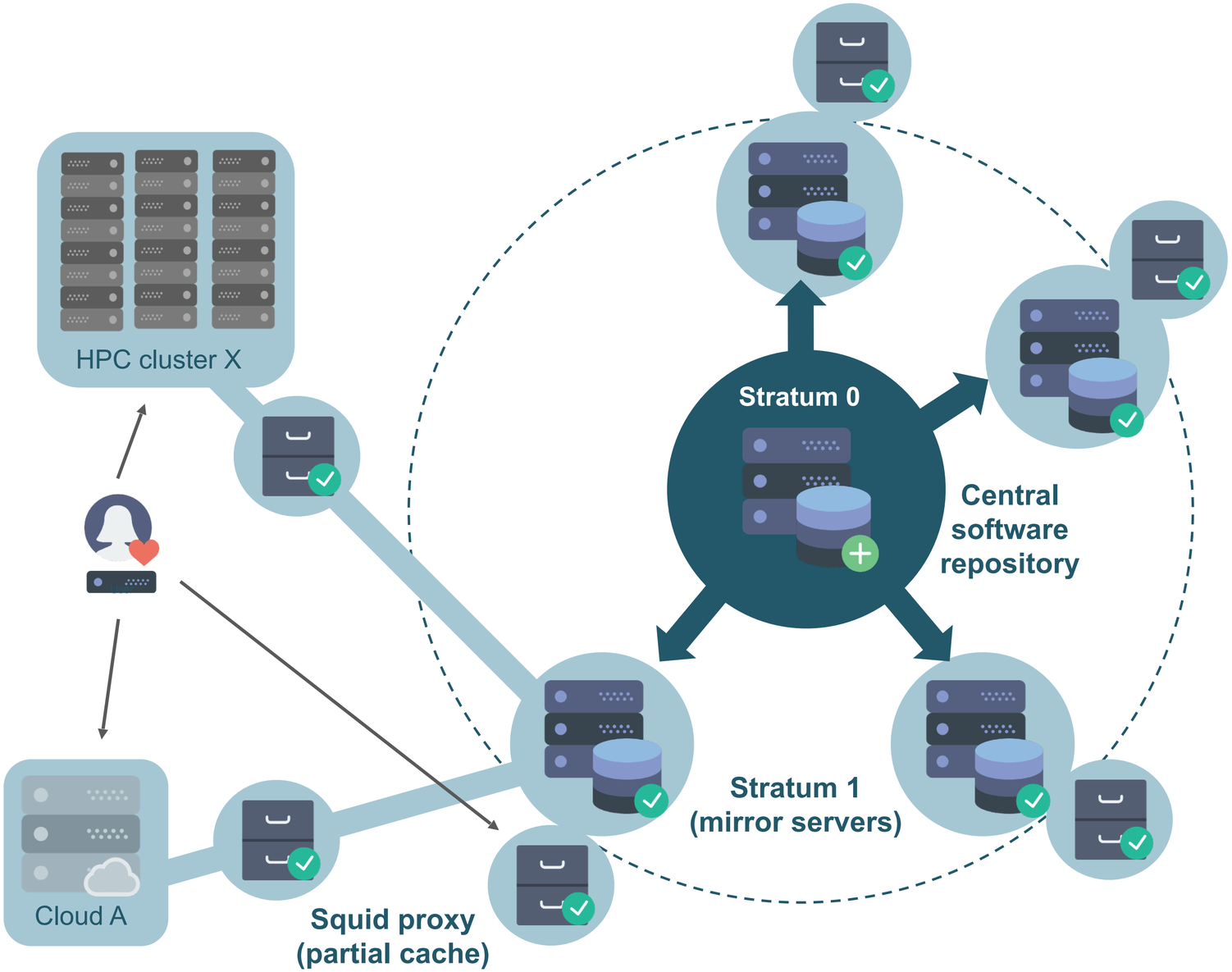
global distribution of the software stack

host operating system (any Linux distribution)

Heavily inspired by

software stack

Software distribution via CernVM-FS



CernVM-FS

https://cvmfs.readthedocs.io

- Global distribution of software *installations*
- Software is downloaded *on-demand* over HTTP
- Transparent for end users, feels like local software
- Added software is *automatically* picked up
- Redundant network of “mirror” servers
- Multiple levels of caching (start-up performance)
- **Same software stack everywhere**:
  laptops, workstations, HPC clusters, cloud VMs, …

(icons via https://www.flaticon.com/authors/smashicons)

From zero to science in 3 steps

- Step 1: Get access to EESSI 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

```
# Assumption: EESSI is accessible (CernVM-FS is installed + EESSI configuration is 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 2021.12] $ module load GROMACS
[EESSI pilot 2021.12] $ gmx mdrun ...
```

Use cases

- A **uniform software stack** across HPC clusters, clouds, servers and laptops
- Can be leveraged in **continuous integration** (CI) environments for software testing
- Significantly facilitates setting up **infrastructure for HPC training**
    - Using throwaway clusters in the cloud (Azure, AWS, …)
    - Participants can easily get access to same software environment "at home"
- **Enhanced collaboration** with software developers and application experts
    - Work towards "vetting" of scientific software installations included in EESSI
    - Relieve developers from burden of getting their software (properly) installed
- **Community software** and **portable workflows**

Current status of EESSI

status.eessi-infra.org

- Started as an idea in Feb'20, effort kickstarted shortly after
- Monthly online meetings since mid 2020
- Currently an unfunded "side project", actively pursuing dedicated project funding...
- **Work-in-progress, proof-of-concept pilot repository available** - eessi.github.io/docs/pilot
  - Supports `x86_64` (Intel, AMD), `aarch64` (Arm 64-bit), `ppc64le` (POWER)
  - Included software: Bioconductor, GROMACS, OpenFOAM, R, TensorFlow, ...
  - Targets: Intel Haswell + Skylake, AMD Rome + Milan, AWS Graviton2+3, POWER9 (partial)
- Significant amount of sponsored credits provided by both AWS and Azure
- Actively developing monitoring, tooling, test suite, GPU support, contribution workflow, etc.

EESSI paper (open access)

doi.org/10.1002/spe.3075

Detailed overview of motivation, project goals, design, challenges & limitations, use cases, etc.

Wiley Online Library

Journal of: Software: Practice and Experience

RESEARCH ARTICLE | Open Access

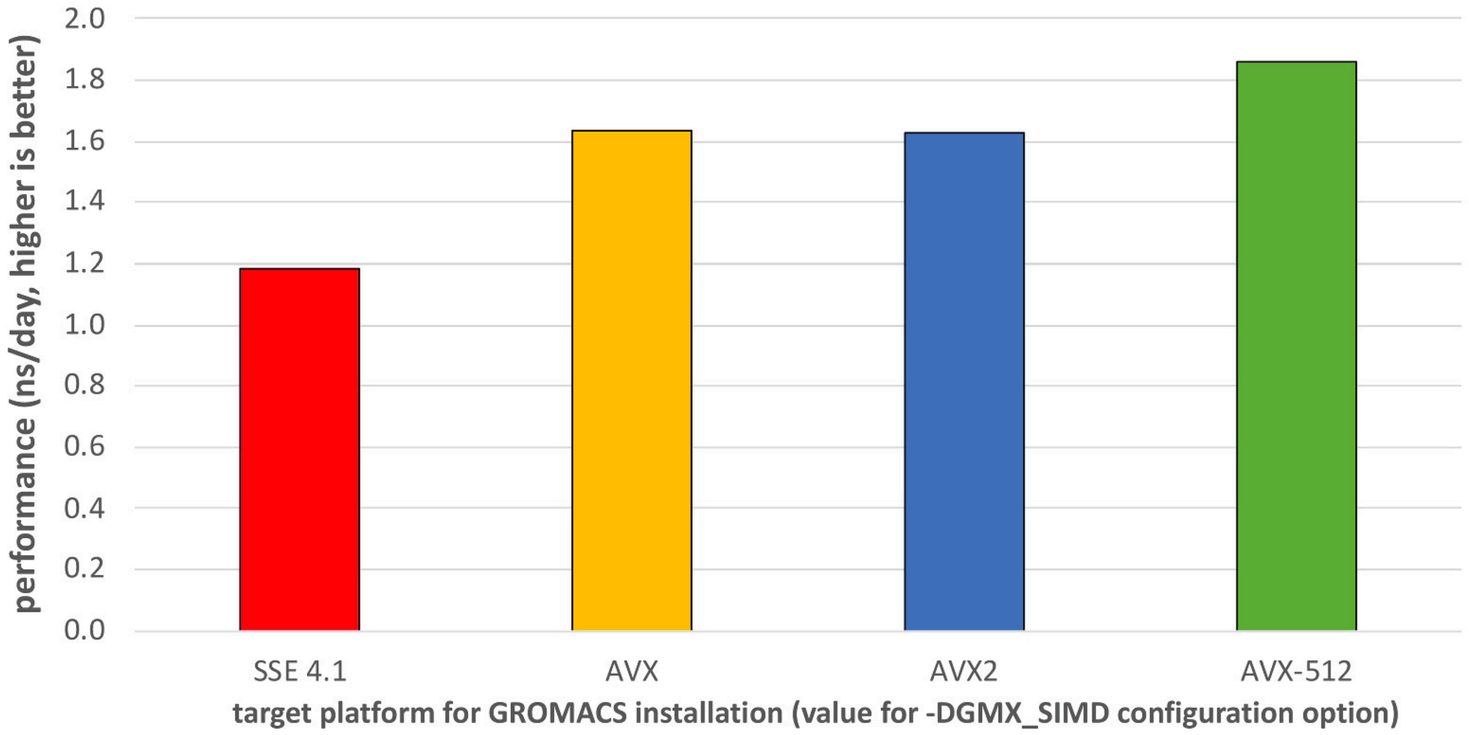
EESSI: A cross-platform ready-to-use optimised scientific software stack

Bob Dröge (Univ. of Groningen), Victor Holanda Rusu (CSCS), Kenneth Hoste (HPC-UGent), Caspar van Leeuwen (SURF), Alan O'Cais (JSC), Thomas Röblitz (Univ. of Bergen)

First published: 16 February 2022

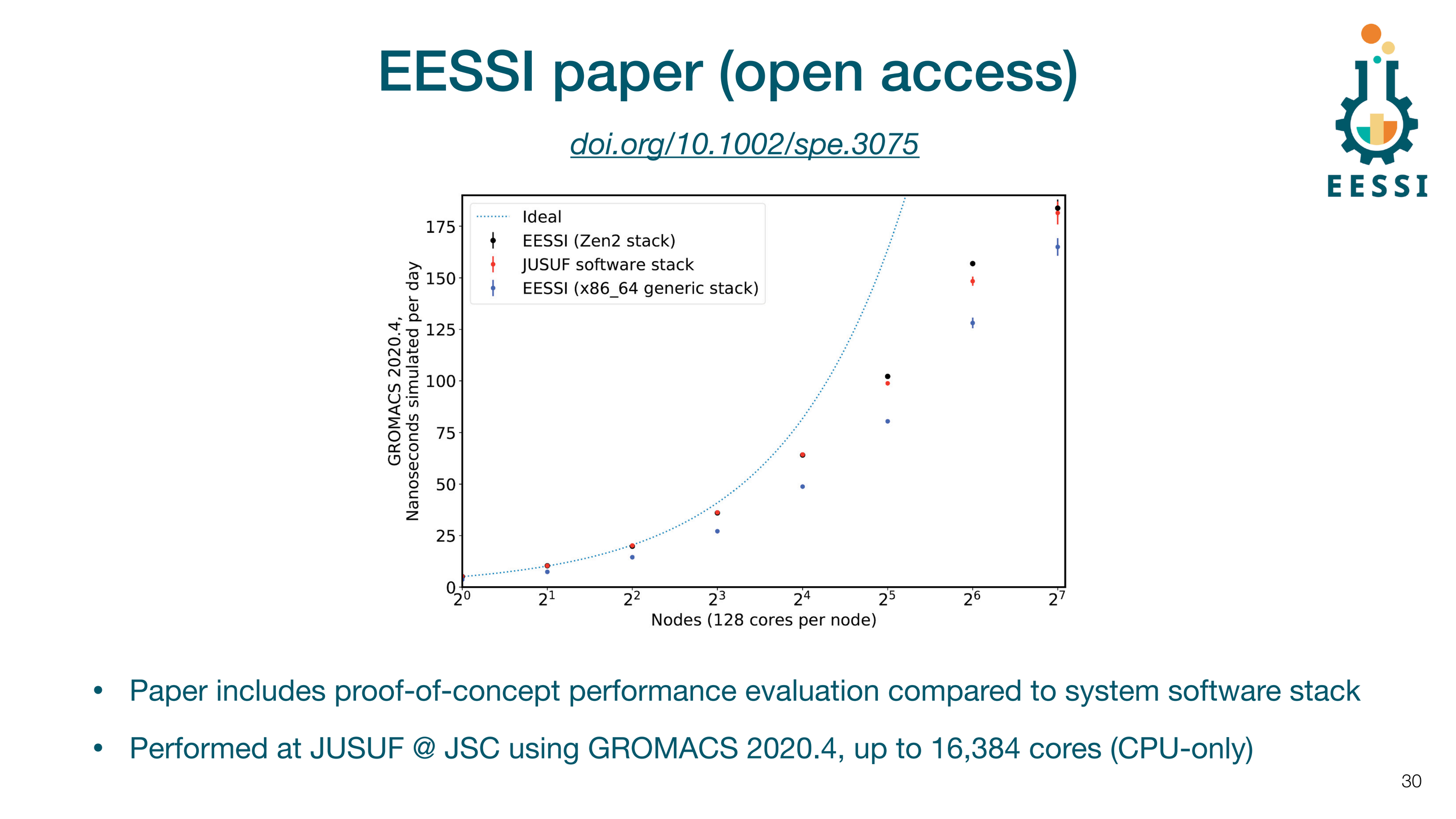
EESSI paper (open access)

doi.org/10.1002/spe.3075



- Motivation for EESSI: keeping the 'P' in HPC (also when using cloud or containers)
- Impact of target instruction set for GROMACS: 57% speedup on CPUs that support AVX-512
- When using containers, performance is often sacrificed for "mobility of compute"...

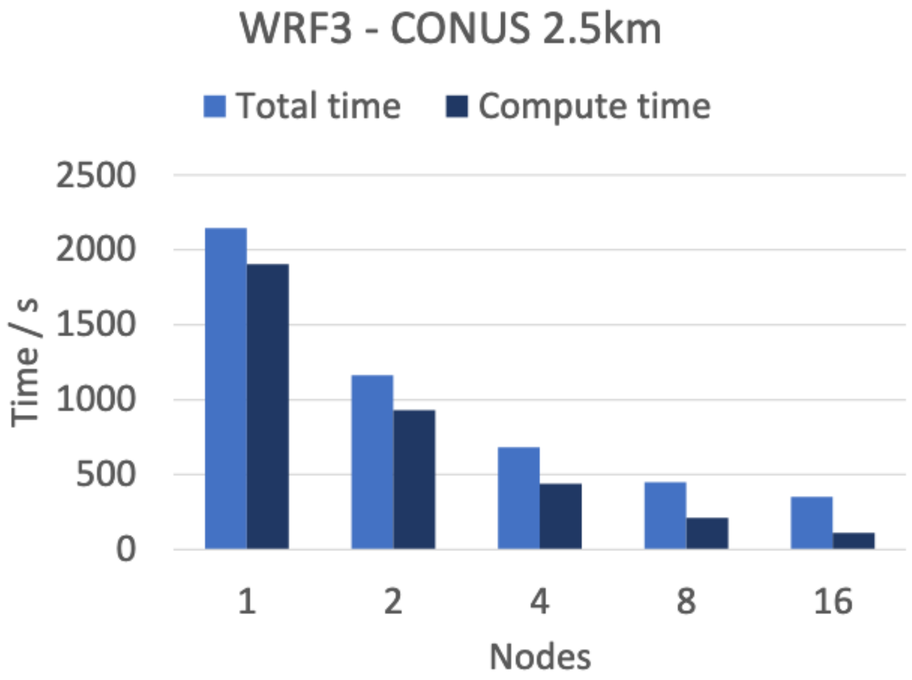
EESSI paper (open access)

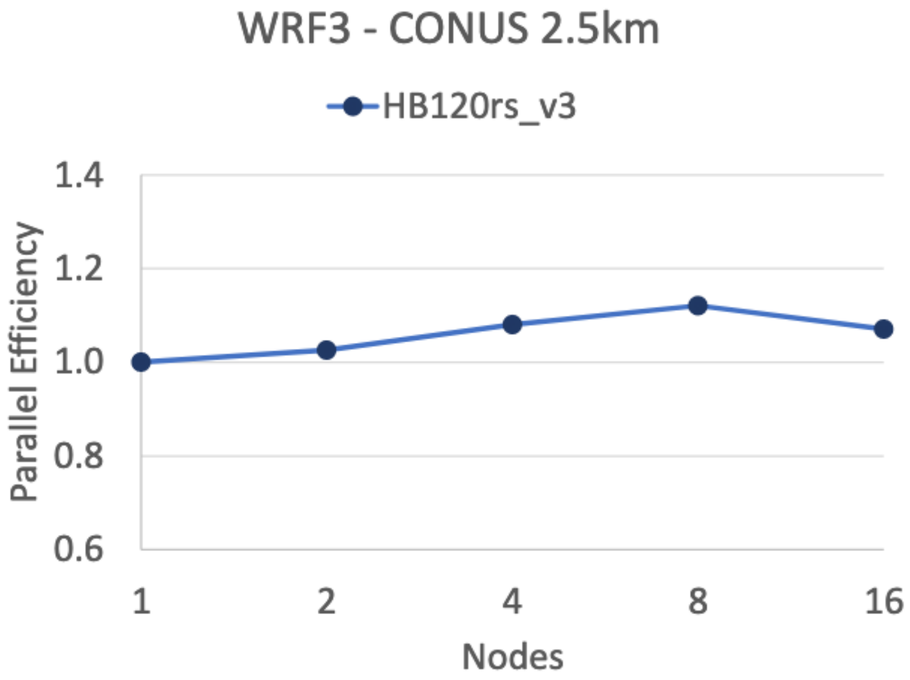


doi.org/10.1002/spe.3075

- Paper includes proof-of-concept performance evaluation compared to system software stack
- Performed at JUSUF @ JSC using GROMACS 2020.4, up to 16,384 cores (CPU-only)

Running WRF in Azure via EESSI





- Performance experiment by Hugo Meiland & Davide Vanzo (Microsoft Azure),
  for their EUM'22 talk "Leveraging EESSI for WRF simulations at scale on Azure HPC"
- WRF v3.9.1.1 with `foss/2020a` on `HB120rs_v3` (120 cores, AMD EPYC 7V13, HDR Infiniband)
- Essentially using EESSI pilot repository as is, no changes needed to leverage fast interconnect!

EESSI

EUROPEAN ENVIRONMENT FOR SCIENTIFIC SOFTWARE INSTALLATIONS

eessi-hpc.org

eessi.github.io/docs

github.com/EESSI

Join our mailing list & Slack channel

eessi-hpc.org/join

Twitter: @eessi_hpc

Status page: status.eessi-infra.org

Paper: doi.org/10.1002/spe.3075

EESSI-related talks at EUM'22:

Getting started - WRF in Azure - Adding software

Monthly online meetings, open to anyone interested

(first Thursday, 14:00 CE(S)T)