EESSI

EUROPEAN ENVIRONMENT FOR
SCIENTIFIC SOFTWARE INSTALLATIONS

hpckp'20
Barcelona

June 18th 2020

#HPCKP20

Kenneth Hoste (HPC-UGent)

whoami

kenneth.hoste@ugent.be
@boegel
@kehoste

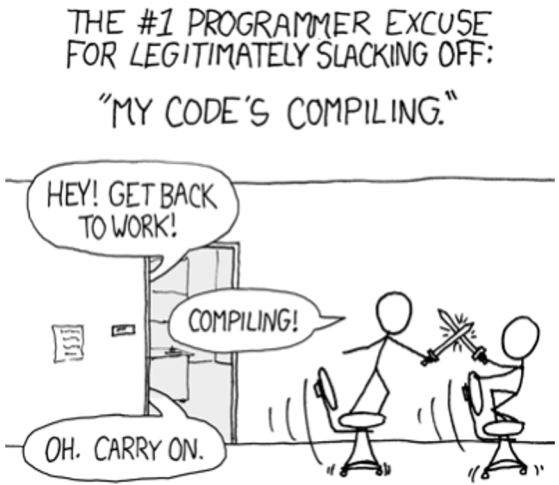
- Masters & PhD in Computer Science from Ghent University (Belgium)
- Joined HPC-UGent team in October 2010
- Main tasks: user support & training, software installations
- Slowly also became EasyBuild lead developer & release manager
- Likes family, beer, loud music, FOSS, helping people, dad jokes, stickers, ...
- Doesn't like C++, CMake, SCons, Bazel, TensorFlow, OpenFOAM, ...

Outline

- Introduction to EESSI (what, who, why)
- Brief history of the project
- Scope & goals
- High-level overview (how)
- Opportunities & challenges
- Current status
- Call for collaboration

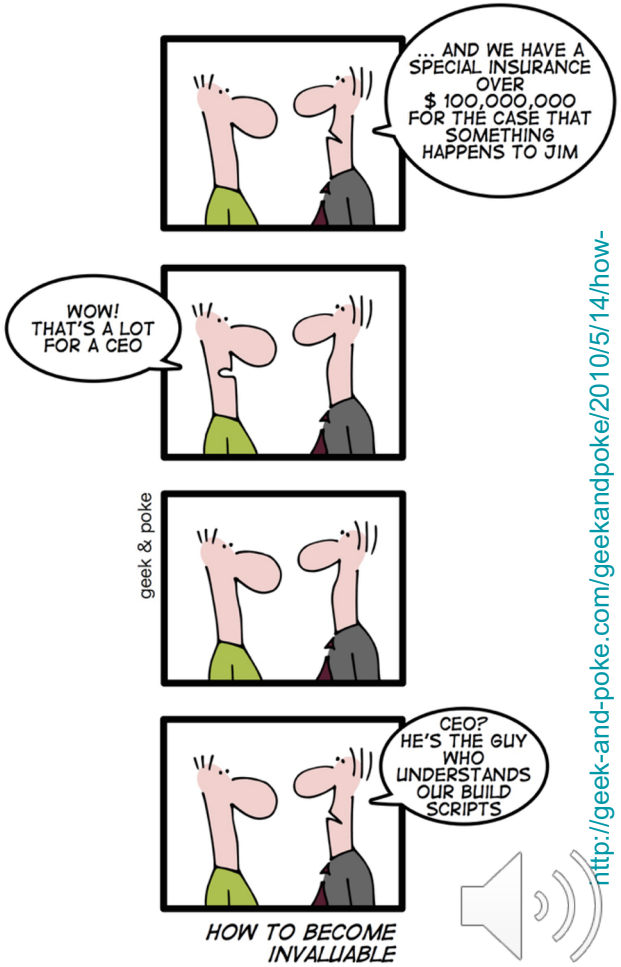
Getting scientific software installed

- Compilation from source (because of the ‘P’ in HPC)
- Things get messy fast, and it’s getting worse...



https://xkcd.com/303

https://xkcd.com/1654

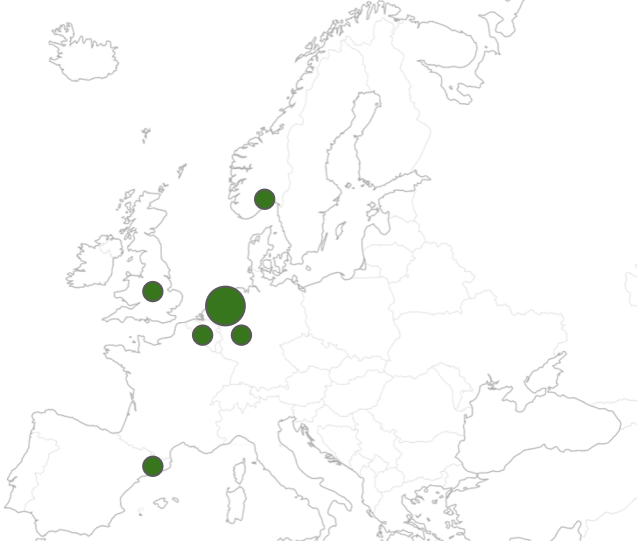


http://geek-and-poke.com/geekandpoke/2010/5/14/how-to-become-invaluable.html

Quick introduction

- European Environment for Scientific Software Installations (EESSI, pronounced as “easy”)
- Collaboration between different European partners in HPC community
- Goal: building a **common scientific software stack for HPC systems & beyond**
- Heavily inspired by Compute Canada software stack
- “Grass roots” project, fueled by a lack of time to do a proper job at installing scientific software and the desire for collaborating on something useful (+ having beers together)

Current EESSI partners



Dutch institutes:

TU Delft

TU/e Eindhoven University of Technology

University of Twente.

VU University Amsterdam

SURF (national institute)

university of groningen

Other European institutions:

Ghent University

Jülich Forschungszentrum

University of Cambridge

UiO : University of Oslo

Companies:

HPC Now!

DELL Technologies

Motivation: changing HPC landscape

- Explosion of scientific software applications
  - Bioinformatics, Machine learning & AI, ...
  - Fueled by recent shifts in scientific community
    - Popularity of GitHub & similar platforms to share code
    - Pressure to publish code along with research articles
    - Wider adoption of HPC across scientific domains (cf. GPUs in bioinformatics)
- Rise of interest in using cloud infrastructure
- Increasing variety in processor (micro)architectures
  - Wide variety of Intel microarchitectures + AMD, ARM, POWER, soon also RISC-V?
  - In addition, GPUs (NVIDIA, AMD, soon Intel?) and other accelerators
- In stark contrast: available manpower in HPC support teams...

EESSI history



- Idea grew after visit of Dutch universities to Univ. of Cambridge via Dell Technologies
- Desire to work together on shared software stack leveraging CernVM-FS
- Several sites were already using EasyBuild
- Discovered Compute Canada software stack via talk at PEARC'19
- Invitation to Kenneth Hoste to join next meeting @ TU Delft (Feb'20)
- Monthly online meetings since then to get organised

Inspiration for this project

- EESSI concept is heavily inspired by **Compute Canada software stack**
- Shared across 5 major national systems in Canada + a bunch of smaller ones
- 3 layers: CVMFS / ~~Nix~~ Gentoo Prefix / EasyBuild + Lmod
- See paper by Maxime Boissonneault & co at PEARC'19 (PDF available here)
  "Providing a Unified Software Environment for Canada's National Advanced Computing Centers"
- See also Maxime's talk at 5th EasyBuild User Meeting (slides - recorded talk) and the Compute Canada documentation

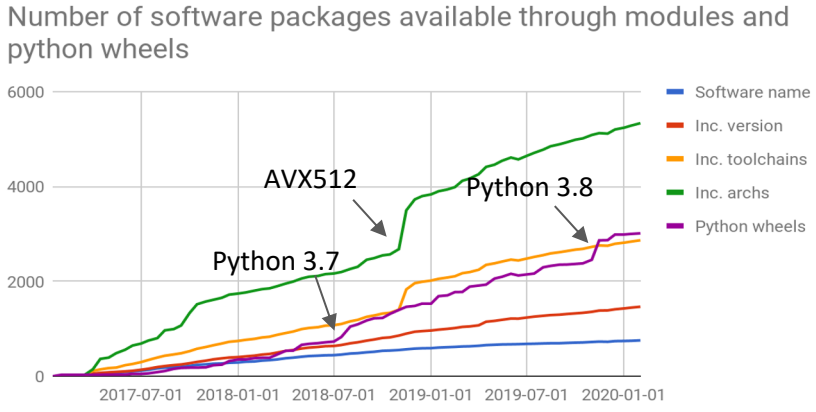
Compute Canada software stack

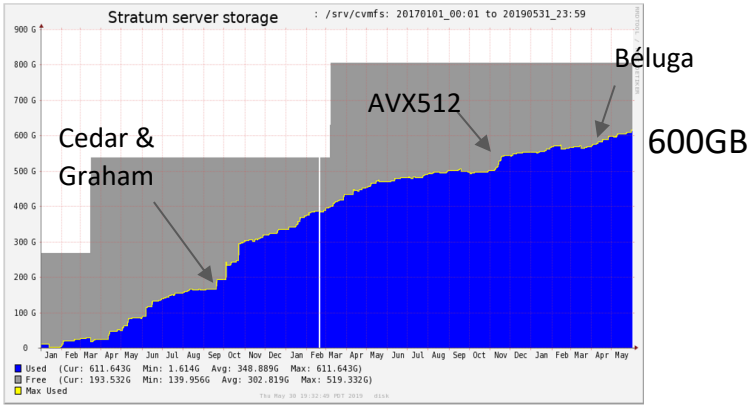
Available software

600+ scientific applications

4,000+ permutations of version/arch/toolchain

Type	Modules
AI	5
Bioinformatics	239
Chemistry	63
Data	19
Geo/Earth	23
Mathematics	82
MPI libraries	7
Physics	48
Various tools	176
Visualisation	28
Misc	38





- Two major new clusters with Skylake CPUs
- Built new modules with AVX512 for most packages
- High deduplication
- Further details

(slide by Maxime Boissonneault, EUM20)

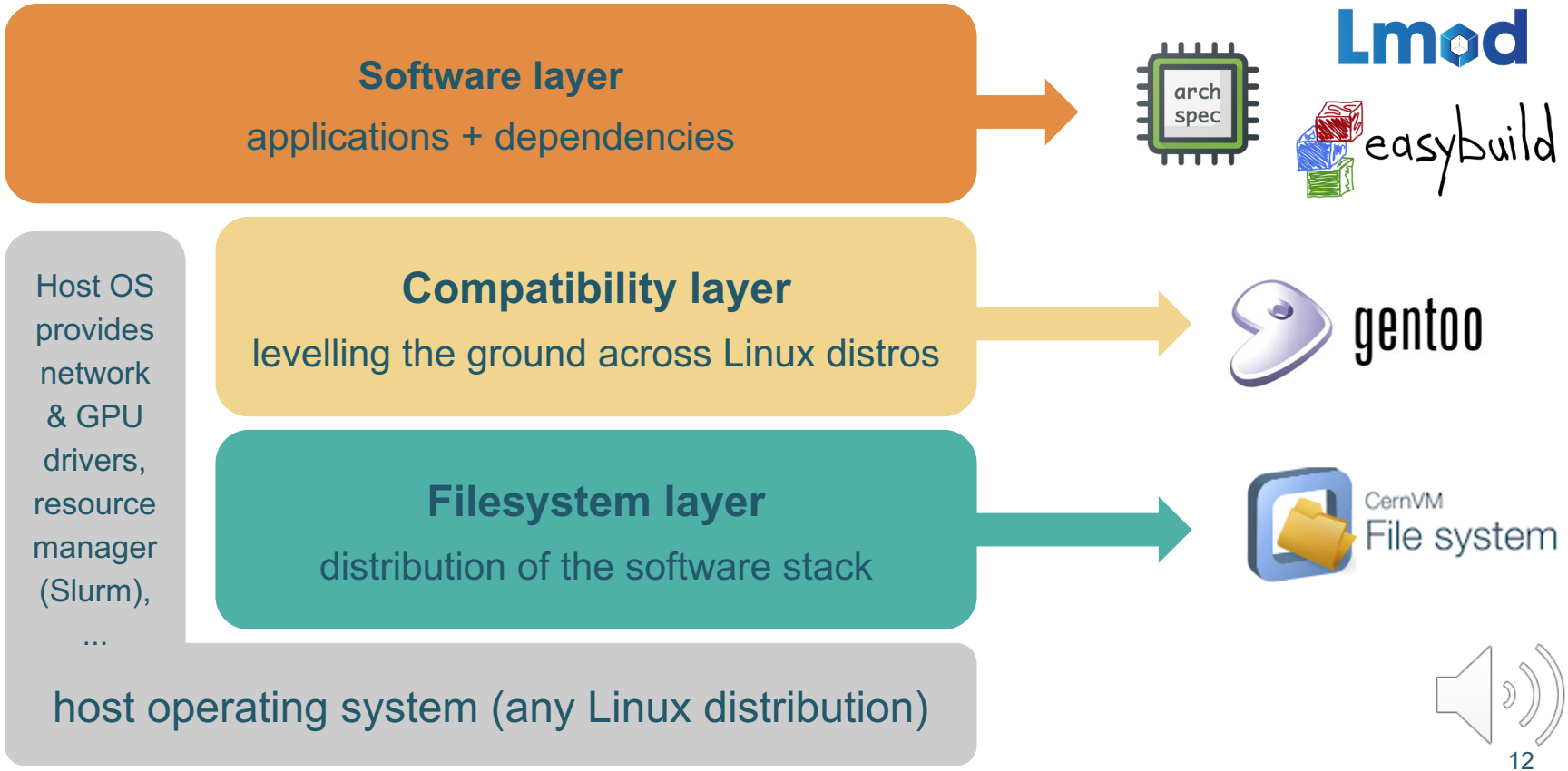
Scope & goals

- Shared repository of scientific software installations
- Avoid duplicate work across HPC sites
- Uniform way of offering software to users, regardless of system they use
- Should work on any (common) Linux distribution and system architecture
  - From laptops and personal workstations to HPC clusters and cloud
  - Support for different CPUs, interconnects, GPUs, etc
- Focus on **performance**, automation, testing, collaboration

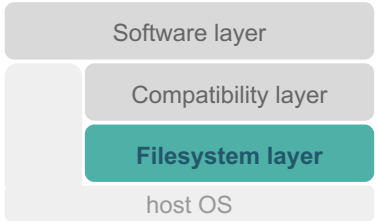
EESSI

EUROPEAN ENVIRONMENT FOR SCIENTIFIC SOFTWARE INSTALLATIONS

High-level overview of the EESSI project



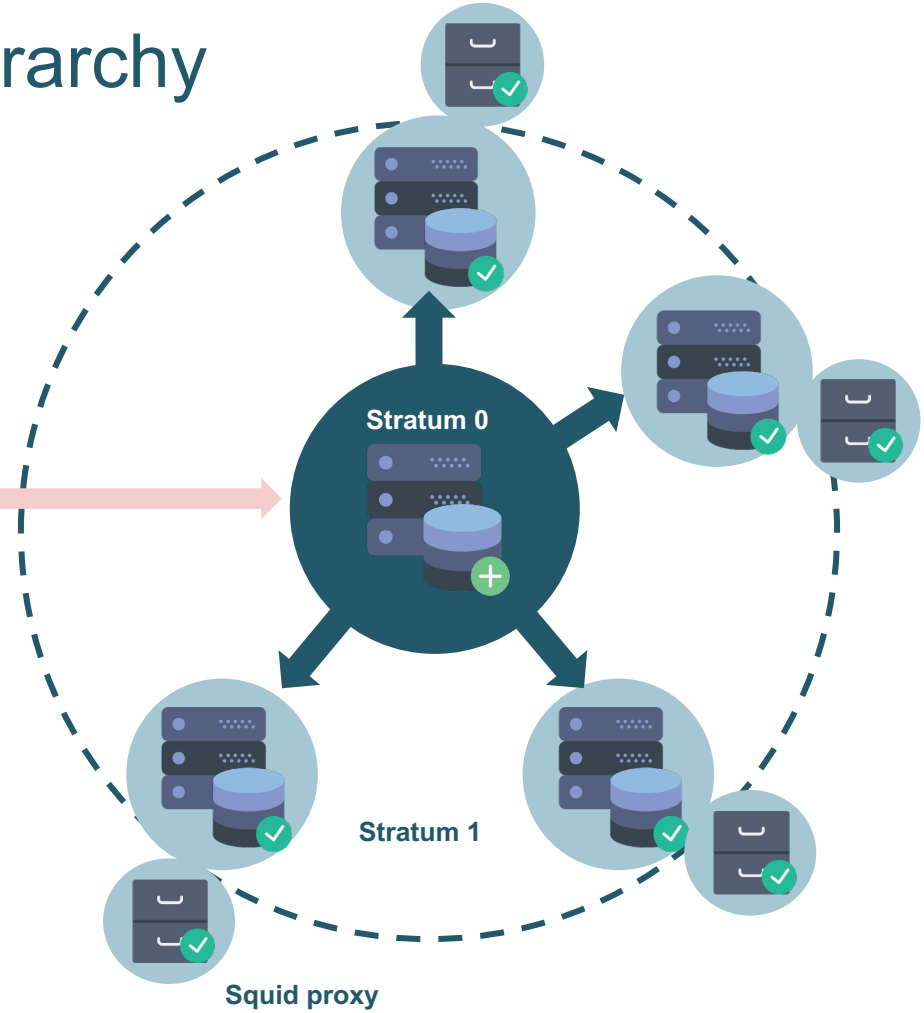
Filesystem layer



- CernVM-FS, a.k.a. CVMFS (https://cvmfs.readthedocs.io)
- “Scalable, reliable and low-maintenance software distribution service”
- Distributed filesystem in different tiers: Stratum 0, Stratum 1, proxies, clients
- Created by CERN in 2008 to distribute LHC software stack worldwide
- POSIX *read-only* filesystem in user space via FUSE kernel module
- Specifically built for distributing software (lots of small files, access pattern, ...)
- HTTP only, aggressive caching, scales to >100M files and >100k clients

CVMFS hierarchy

CernVM File system

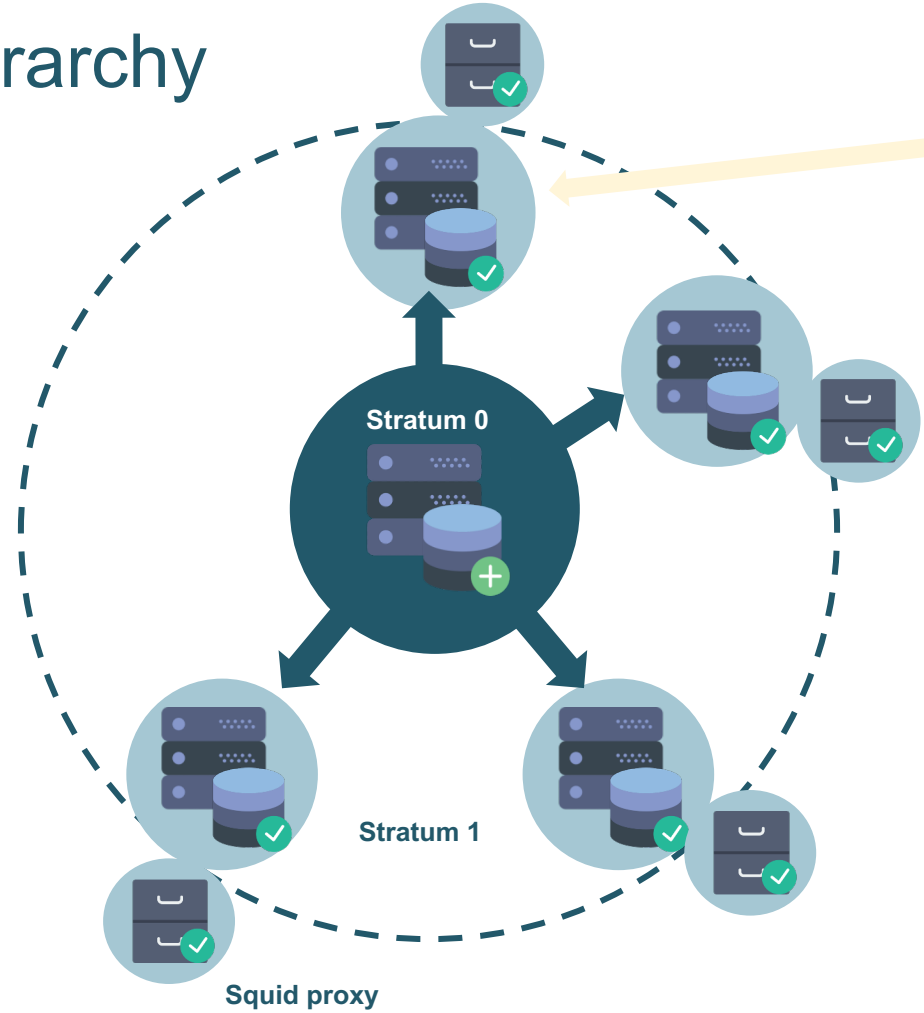


Stratum 0

- Central server
- **Only one**
- Hosts CVMFS “repositories”
- New software is added here
- Usually restricted access

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

CVMFS hierarchy



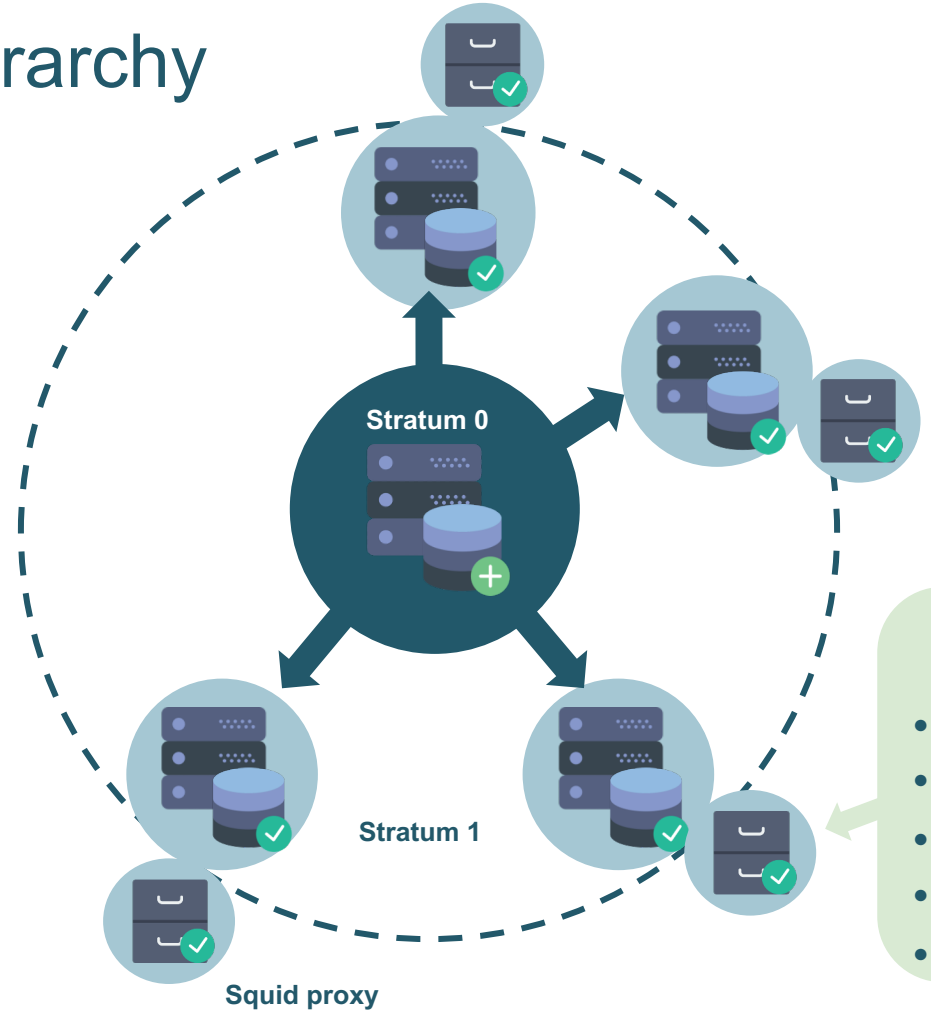
Stratum 1

- CVMFS “replica”
- Full copy of repositories (mirror)
- **Read-only**
- Multiple ones
- Geographically distributed
- Standard web server
- Reduce load on Stratum 0
- Improve reliability
- Clients *may* connect here

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

CVMFS hierarchy

CernVM File system

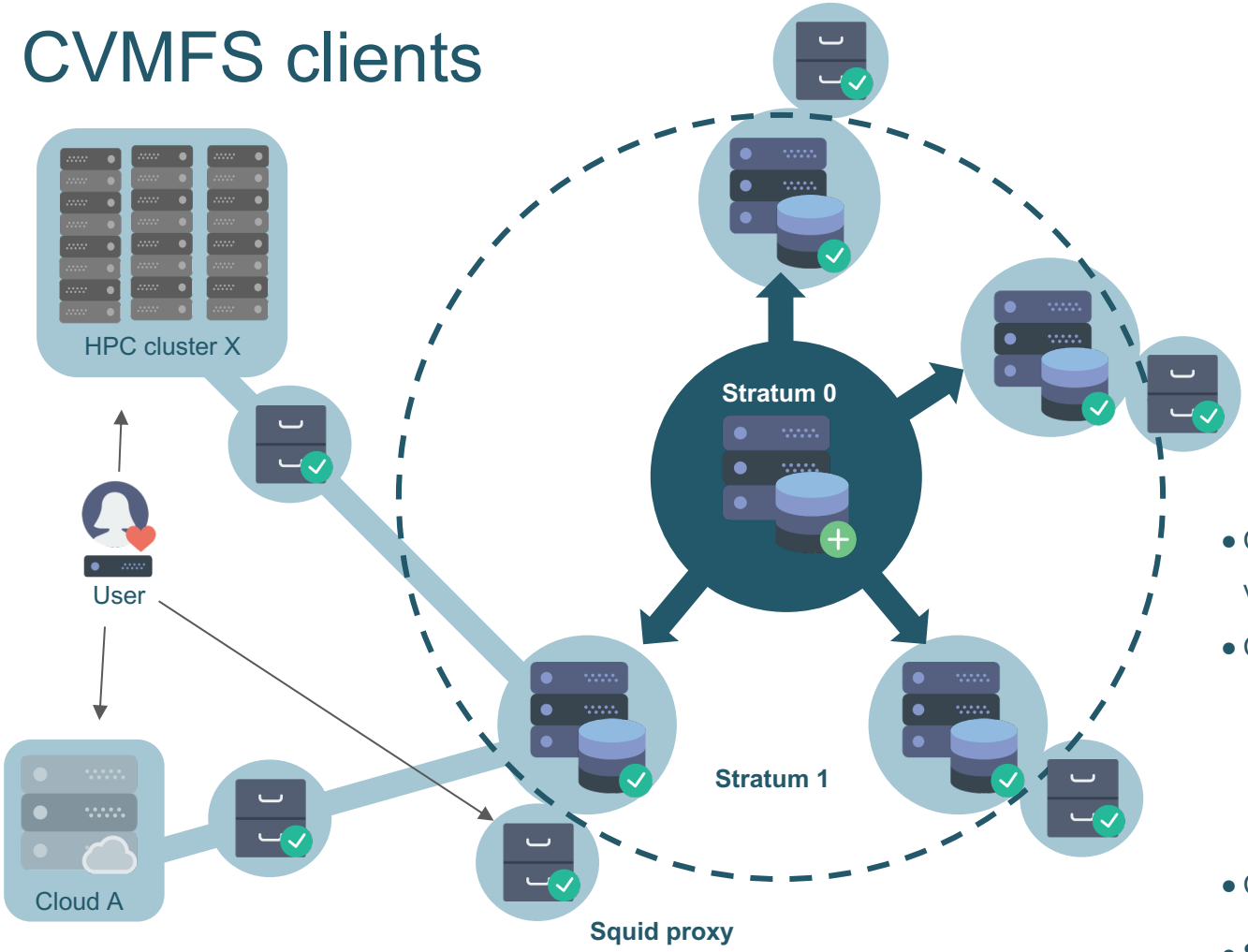


Squid proxy

- Reverse proxy for Stratum 1 servers
- Caching to improve client I/O performance
- Reduce load on Stratum 1 servers
- Can be used for load balancing
- Clients usually connect to on… these

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

CVMFS clients

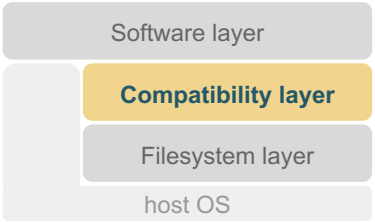


CernVM File system

- Clients access software stack via (squid proxy of) a Stratum 1
- Clients include:
  - Laptops
  - Personal workstations
  - HPC clusters
  - Cloud instances
- Clients also use local filesy[illegible] cache
- **Same software stack everywhere!**

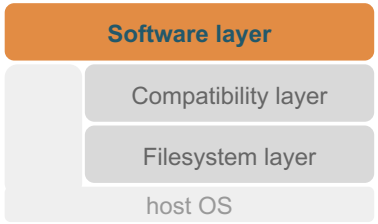
(icons via https://www.flaticon.com/authors/smashicons)

Compatibility layer



- Gentoo Prefix (https://wiki.gentoo.org/wiki/Project:Prefix)
- Consists of Gentoo Linux packages installed in non-standard location (a “prefix”)
  - Using Gentoo's package manager Portage
- Long history, originally to run Gentoo on macOS systems
- Supports x86_64, ARM, POWER (?) architectures and Linux + macOS clients
- In EESSI: only used for a limited set of low-level libraries and tools, down to glibc
- Required to ensure compatibility of software stack with client operating syste

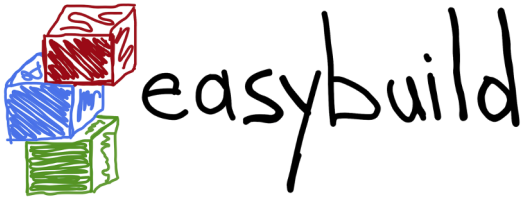
Software layer



- Top layer provides actual scientific software installations
- Installed using EasyBuild, optimized for a specific system architecture (CPU, GPU)
- Environment modules are provided to “load” the software so it’s ready for use
- This software leverages tools & libraries in Compatibility layer (not from host OS)
- Kernel modules and drivers (network, GPU) are still provided by host OS
- Best suited part of software stack can be selected automatically for host (via archspec)

github.com/easybuilders

easybuild.readthedocs.io



easybuild.slack.com
(join via easybuild-slack.herokuapp.com)

youtube.com/c/easybuilders

- Framework for installing scientific software on HPC systems
- Originally created by HPC-UGent in 2009, public and stable since 2012
- Frequent new releases, extensively tested, GPLv2 license
- Installations are **optimized for processor architecture of host** by default
- Supports almost 2000 different software packages (excl. extensions),
  including LAMMPS, NWChem, OpenFOAM, PETSc, PyTorch, R, TensorFlow, Trilinos, …
  (see https://easybuild.rtfd.io/en/latest/version-specific/Supported_software.html)
- Frequent hackathons & meetups, incl. 5 EasyBuild User Meetings (2016 - 2020)
- > 250 unique contributors (~100 yearly contributors)

github.com/TACC/Lmod

lmod.readthedocs.io

TACC
TEXAS ADVANCED COMPUTING CENTER

- A modern environment modules tool, implemented in Lua
- Developed by Robert McLay (TACC), MIT license
- Quick adoption in HPC community in recent years
- Simplifies managing of your environment: `module load`, `module avail`, ...
- Compatible with module files written in both Lua and Tcl
- Designed specifically to provide good support for hierarchical module tree
- Lots of addition features: module cache, collections, properties, sticky modules, tracing, ...
- Default modules tool in EasyBuild

A library to reason about system architecture aspects

github.com/archspec

archspec.readthedocs.io

- Originally part of Spack, now a standalone library
- Currently focused on CPU microarchitectures,
  intention is to extend the concept to network, GPUs, …
- JSON file with “database” of CPU microarchitectures (Intel, AMD, ARM, POWER)
- **Fine-grained human-readable labels**, a.k.a. codenames (haswell, skylake, …)
-  Python library is available to leverage this data
- **Query instruction set support & compatibility between microarchitectures, ...**
- Cross-project collaboration between Kenneth Hoste (EasyBuild)
  and Todd Gamblin + Massimiliano Culpo (Spack)

What can archspec be used for?

- Collect host info (via `/proc/cpuinfo` on Linux, `sysctl` command on macOS)
- Determine codename of host CPU via archspec CLI ⟶

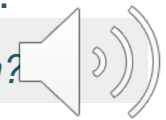
```
$ archspec cpu
skylake
```

- Check **compatibility** of host with given microarchitecture:

  "Can the current host run binaries that were compiled for Intel Haswell?"

- Query **capabilities** of microarchitectures: *"Does Intel Haswell support AVX-512?"*
- Query **compiler flags**: *"How do I compile for Intel Broadwell with icc?"*
- **Partial ordering** of microarchitectures, picking best match from list of options:

  "Given AVX, AVX2 and AVX-512 binaries, what's the best match on a Broadwell system?"



Relevance to EESSI

Archspec is helpful in the context of this project for:

- **Determining the software installation prefix to use in EasyBuild**

  Example: `/cvmfs/pilot.eessi-hpc.org/2020/x86_64/intel/`**`haswell`**

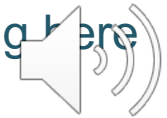
- **Automatically picking the best suited part of the software stack for a host**

  Example:

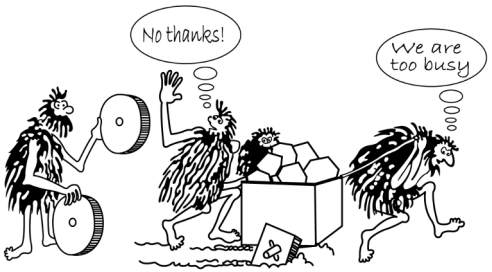
  - Host CPU is Intel Broadwell
  - Installations are available for Sandy Bridge (AVX), Haswell (AVX2), Skylake (AVX512)
  - Best match is software installed for Haswell (compatible & best performance)

Testing the software installations

- Just getting the software installed is not enough,
  we also need to ensure it actually *works*
- Both the *correctness* and *performance* of installations should be tested
- For this, we plan to use ReFrame (https://reframe-hpc.readthedocs.io),
  a software regression testing framework for HPC systems developed by CSCS
- For end-user applications in the software stack, we want to include test scripts
- Working closely together with the software developers could be interesting here



Opportunities



EESSI
EUROPEAN ENVIRONMENT FOR SCIENTIFIC SOFTWARE INSTALLATIONS

- Collaboration across HPC community
  (centres, vendors, consultancy companies, software developers, …)
- Let users seamlessly hop between sites: same software available everywhere!
- Leveraging each others work w.r.t optimized software installations more efficiently
- Thoroughly tested scientific software installations for the HPC community
- Easy benchmarking: should I use AVX2 or AVX512 binaries on my Skylake system?
- Work together with developers of scientific software to vet the installations

Challenges

- Finding time and manpower to get this set up properly
- Network interconnect (MPI & co), accelerators (GPUs), ...
- CPU architectures other than x86_64, like ARM, POWER, RISC-V, …
- Dealing with licensed software, like Intel tools, MATLAB, ANSYS, ...
- Integration with resource managers (Slurm) and vendor provided software (Cray PE)
- Convincing site admins to adopt EESSI (loss of control, NIH syndrome)
- Answer to *“Will X actually work?”* is usually *“Yes, see what Compute Canada d*

Current status

- Monthly (online) meetings to discuss progress
- Actively working towards a functioning pilot setup
- Making progress in all 3 layers:
  - Automated rollout of the CVMFS filesystem layer using Ansible is working
  - Testing with Gentoo Prefix underway
  - Designing the software layer: brainstorm done, actual work to be started
- Setting up documentation, website, GitHub repositories, ...

Pilot setup

- To figure out workflow and who will work on what
- Try to weed out roadblocks in current plan and identify oversights
- First round of documentation (adopting EESSI, for end users, etc.)
- Limited scope:
  - Small set of CPU architectures + one or two types of GPUs
  - Stick to EasyBuild's default module naming scheme (EasyBuildMNS) for now
  - Only Linux clients (no macOS or Windows via WSL for now)
  - Small software stack: OpenFOAM, TensorFlow, GROMACS, some bioinformatics pipeline
- **NOT FOR PRODUCTION USE!**

Call for collaboration

- Interested in the project? **Reach out to us!**
- We're currently looking for:
  - Candidate ***active*** contributors to help out with the pilot setup
  - *European* members of the HPC community interested in joining the effort
  - Experience with installing scientific software, EasyBuild, Git, GitHub is helpful
- We also welcome suggestions for nice locations to have beers...

More information

Website: https://www.eessi-hpc.org

GitHub organisation: https://github.com/EESSI

Documentation: https://eessi.github.io/docs

Twitter: @eessi_hpc

EESSI

EUROPEAN ENVIRONMENT FOR
SCIENTIFIC SOFTWARE INSTALLATIONS

kenneth.hoste@ugent.be