European Environment for Scientific Software Installations (EESSI)

June 18th 2020

Kenneth Hoste (HPC-UGent)

whoami

- TODO

Kenneth Hoste (HPC-UGent)

kenneth.hoste@ugent.be - @boegel ( *) -*  *@kehoste ( )* 

Outline

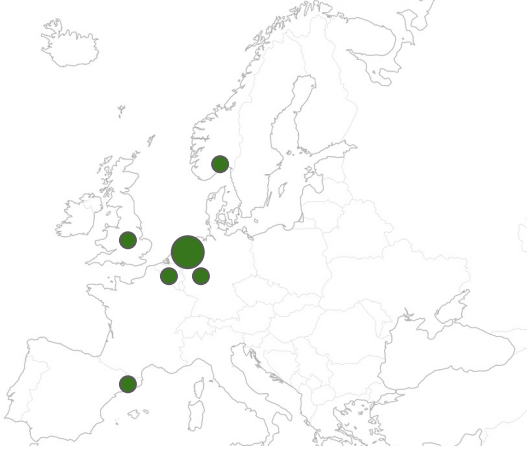
- Introduction (what, who, why)
- Brief history of the project
- Scope & goals
- High-level overview (how)
- Opportunities
- Current status
- Call for collaboration

Quick introduction

github.com/EESSI
eessi.github.io/docs

- European Environment for Scientific Software Installations
  (EESSI, pronounced as “easy”)
- collaboration between different European partners in HPC community
- goal is to build a common scientific software stack for HPC systems & beyond
- *heavily* inspired by Compute Canada software stack
- “grass roots” project, fueled by desire for collaborating on something useful
  (& having beers together)

Current EESSI partners



- Dutch institutes:
  - TU Delft: Robbert Eggermont, Koen Mulderij
  - University of Groningen: Bob Dröge, Henk-Jan Zilverberg
  - University of Twente: Geert Jan Laanstra
  - Vrije Universiteit Amsterdam: Peter Stol
  - SURF: Caspar van Leeuwen, Marco Verdicchio, Bas van der Vlies
- other European institutions:
  - HPC-UGent (Belgium): Kenneth Hoste
  - Jülich Supercomputing Centre (Germany): Alan O'Cais
  - University of Cambridge (United Kingdom): Mark Sharpley
  - University of Oslo (Norway): Thomas Röblitz
- companies:
  - Dell Technologies (Europe): Walther Blom, Jaco van Dijk
  - HPCNow! (Spain): Oriol Mula Valls

Motivation: changing HPC landscape

- Explosion of scientific software applications
  - bioinformatics, machine learning & AI, ...
  - fueled by recent developments
    - popularity of GitHub & similar platforms to share code
    - pressure to publish code along with research articles
    - deep learning hype
    - wider adoption of HPC across scientific domains (cfr. GPUs in bioinformatics)
- 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 contrast: *decrease* in available manpower in HPC support teams...

Brief history

- idea grew after visit of Dutch universities to Univ. of Cambridge via Dell
- desire to work together on shared software stack leveraging CernVM-FS
- several sites were already using EasyBuild
- discovered ComputeCanada 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, work on pilot setup

Inspired by Compute Canada

- Heavily inspired by Compute Canada software stack,

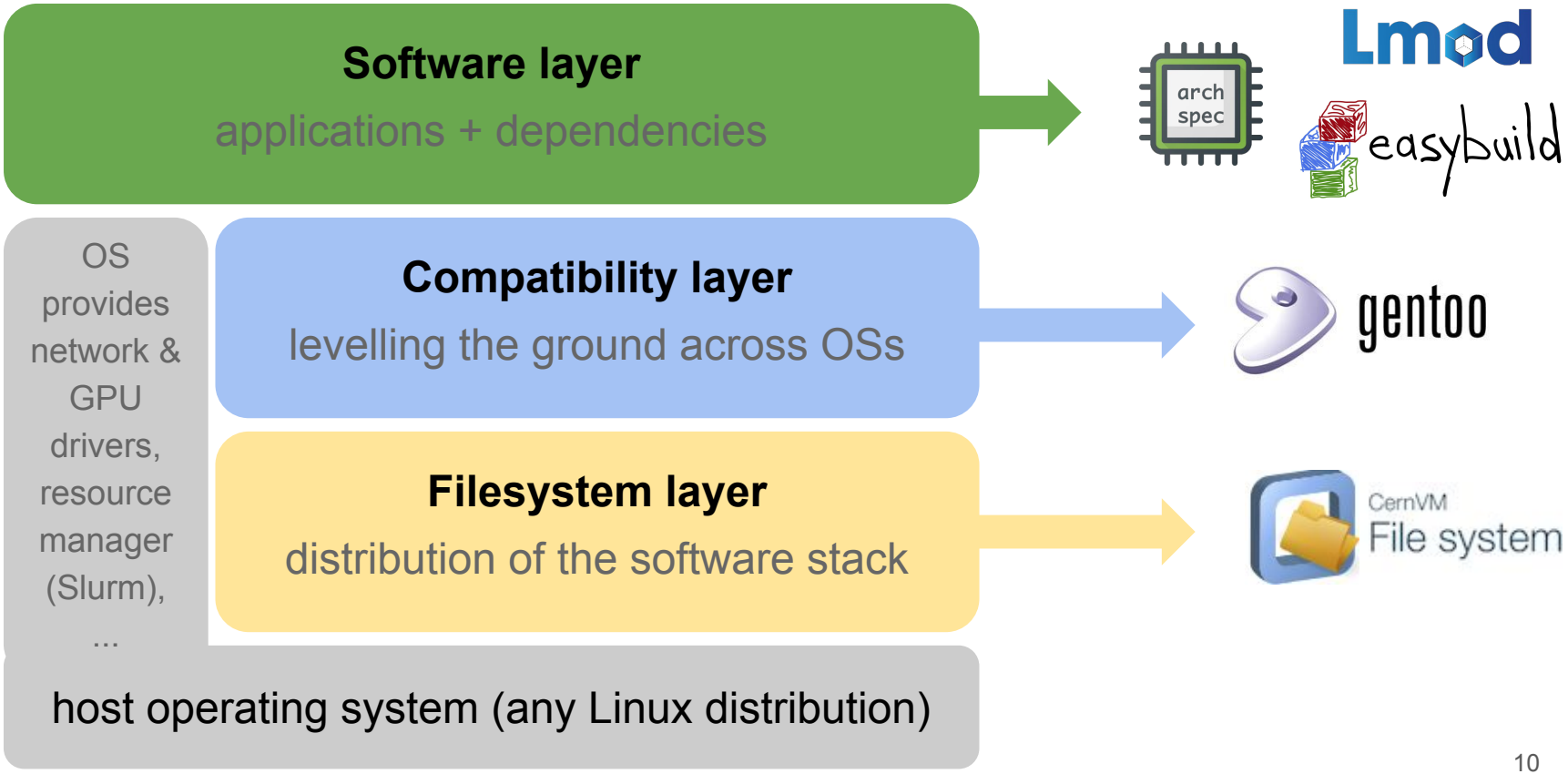
  see talk by Maxime Boissonneault at PEARC'19:

  "Providing a Unified Software Environment for Canada's National Advanced Computing Centers"

Scope & goals

- Shared repository of scientific software installations
- Reduce the time spent on installation of software for each site
- Uniform way of offering software to users
- Should work on any (common) Linux distribution and architecture
  - From laptop to HPC clusters and cloud infrastructures
  - Support different CPUs, interconnects, GPUs, etc
- Automated builds for all different, supported CPU architectures

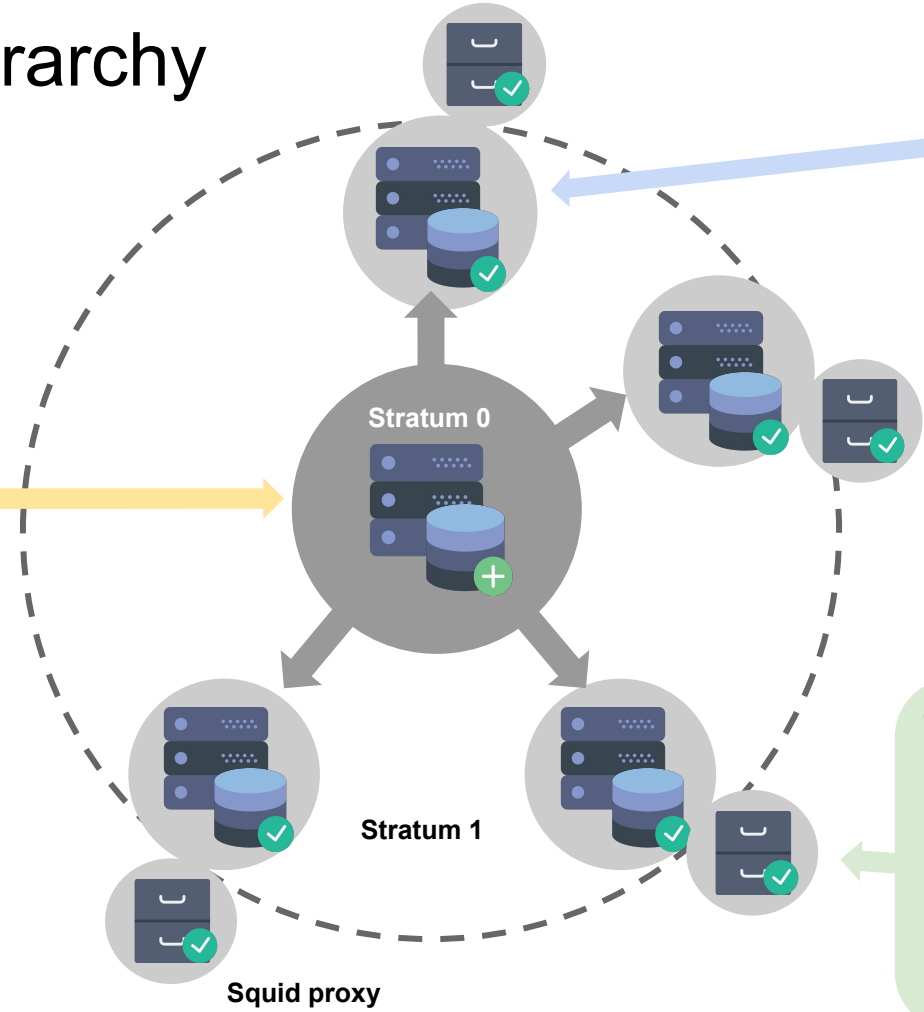
High-level overview of the EESSI project



Filesystem layer

- CernVM-FS, a.k.a. CVMFS (https://cvmfs.readthedocs.io)
- “scalable, reliable and low-maintenance software distribution service”
- distributed filesystem in different levels: Stratum 0, Stratum 1, clients
- created by CERN in 2008 to distribute LHC software stack worldwide
- POSIX **read-only** file system in user space via FUSE kernel module
- specifically built for distributing software (lots of small files, binaries, …)
- HTTP only, aggressive caching, scales to >100M files and >100k clients

CVMFS hierarchy



Stratum 0

- **one** central server
- only one
- hosts CVMFS “repositories”
- new software is added here
- usually restricted access

Stratum 1

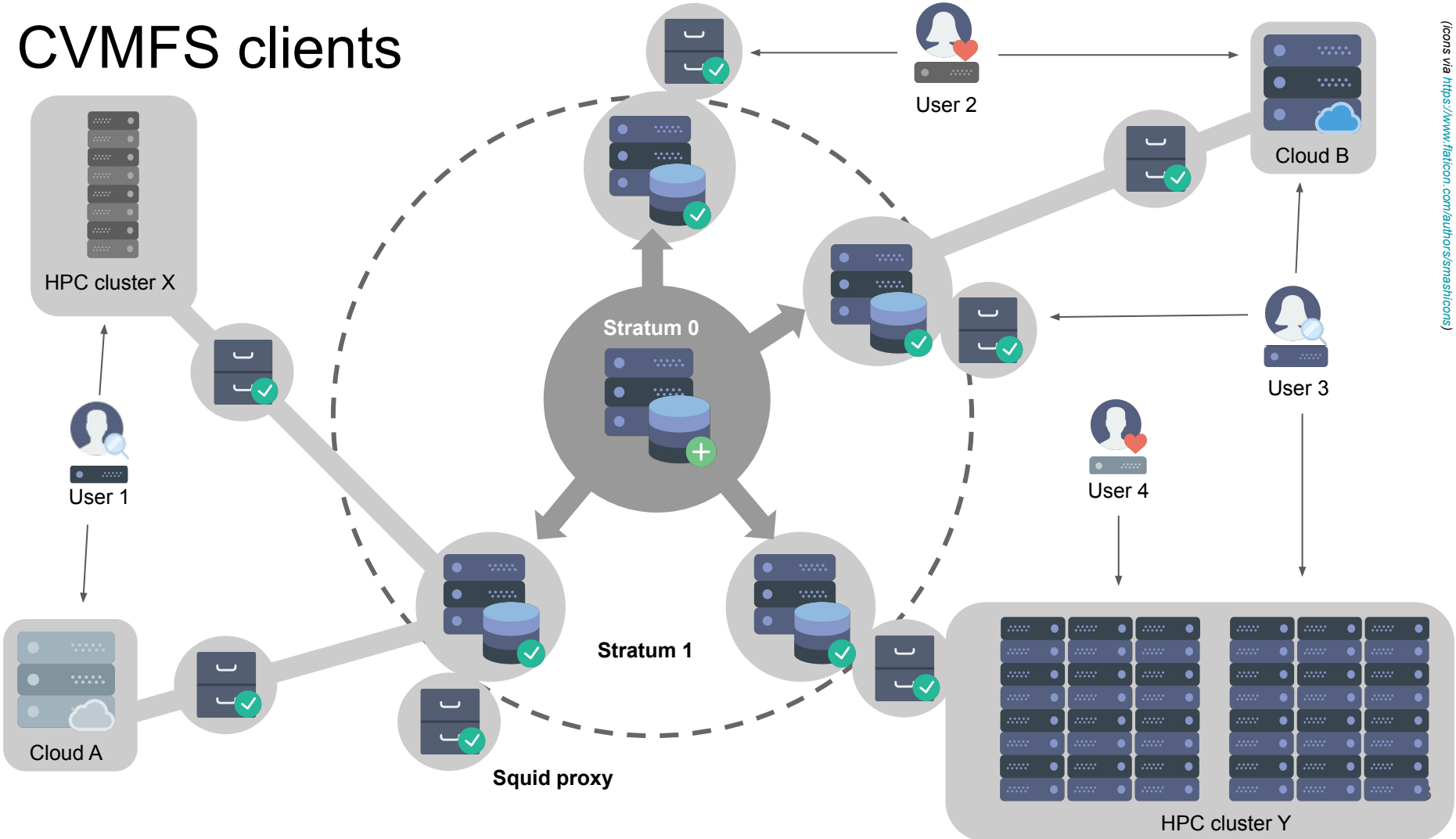
- CVMFS “replica”
- full copy of repositories (mirror)
- **read-only**
- standard web server
- reduce load on Stratum 0
- improve reliability
- geographically distributed
- clients *may* connect here

Squid proxy

- reverse proxy for Stratum 1
- caching to improve client I/O performance
- reduce load on Stratum 1
- can be used for load balancing
- clients usually connect to one of these

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

CVMFS clients



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

Compatibility layer

- Gentoo Prefix (https://wiki.gentoo.org/wiki/Project:Prefix)
- Gentoo Linux packages installed in non-standard location (a “prefix”)
  - using Gentoo's package manager Portage
- Low-level operating system libraries, down to libc
- Ensures compatibility of installed applications with client's Linux distro

Software layer

- STUFF GOES HERE

github.com/easybuilders

easybuild.readthedocs.io

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

youtube.com/c/easybuilders

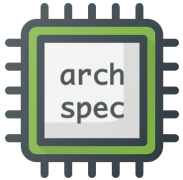
- framework for **installing scientific software on HPC systems**
- originally created by HPC team at Ghent University (Belgium) in 2009
- **stable since 2012**, frequent releases (every 6-8 weeks), extensively tested
- installations are **optimized for processor architecture of host** by default
- supports **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)

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)
- 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

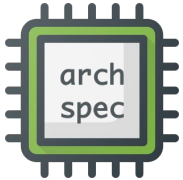
archspec.readthedocs.io

license: Apache 2.0 or MIT

- 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 available to leverage this data: `pip install archspec`
- **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?”

Opportunities

- collaboration across HPC community (centres, vendors, consultancy companies, …)
- working together with developers of scientific software to vet the installations
- let users seamlessly hope between sites: same software available everywhere!

Current status

- Lots of effort being put into the different layers
- Working towards a functioning pilot setup
- Automated roll out of the CVMFS file system layer using Ansible (based on Galaxy repo)
- Testing with Gentoo Prefix underway
- Designing the software layer: brainstorm done, actual work to be started
- Setting up the documentation

Roadmap

- STUFF GOES HERE

Pilot setup

- OpenFOAM, TensorFlow, GROMACS, some bioinformatics pipeline

Call for collaboration

- ACTIVE
- STUFF GOES HERE
- suggestions for nice locations to have beers