EESSI & CernVM-FS meeting

August 17th 2020

Outline

- Introductions (who's who)
- EESSI in a nutshell
- CernVM-FS in EESSI
- Plan & current status
- Future work
- Questions

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)

Scope & goals

- Shared repository of (optimized!) 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

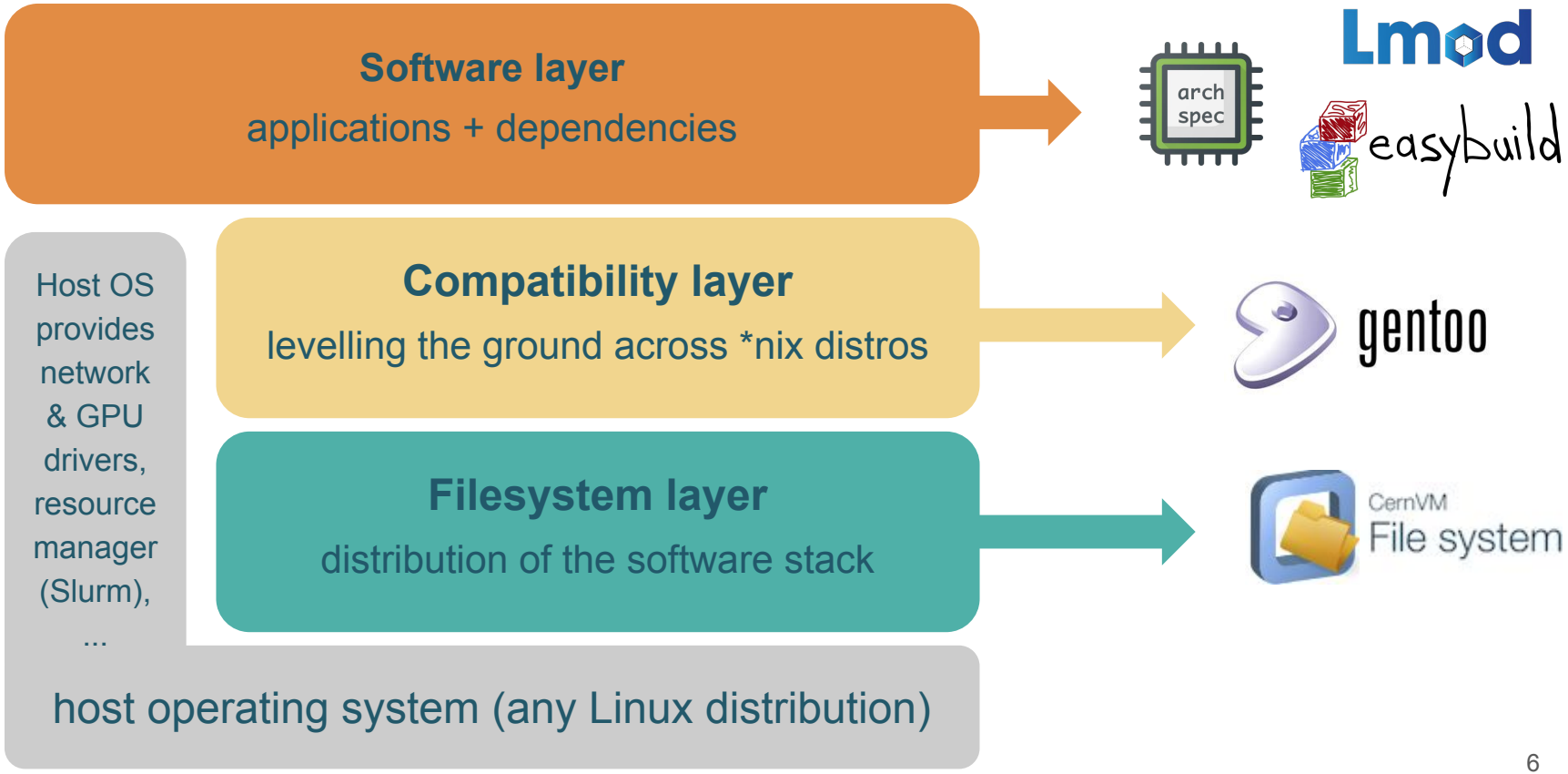
EESSI

EUROPEAN ENVIRONMENT FOR SCIENTIFIC SOFTWARE INSTALLATIONS

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: CernVM-FS / ~~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

High-level overview of the EESSI project



CernVM-FS in EESSI

- Transport layer for all software installations + Gentoo Prefix (and custom tools)
- One tree per architecture (CPU, GPU)
    - Client automatically looks in the right tree based on its architecture
- Config repository for distributing configuration files
- Different repositories for testing and production (and licensed software?)
- Use GEO API for finding the closest Stratum 1

EESSI directory structure


```
.
└── cvmfs/
    ├── cvmfs-config.eessi-hpc.org
    ├── test.eessi-hpc.org
    └── prod.eessi-hpc.org
```

EESSI directory structure


```
.
└── cvmfs/
    ├── cvmfs-config.eessi-hpc.org
    ├── test.eessi-hpc.org
    └── prod.eessi-hpc.org/
        └── 2020.06/
            ├── compat/
            │   ├── aarch64
            │   ├── ppc64le
            │   └── x86_64
            └── software/
                ├── aarch64
                ├── ppc64le
                └── x86_64
```

EESSI directory structure


```
.
└── cvmfs/
    ├── cvmfs-config.eessi-hpc.org
    ├── test.eessi-hpc.org
    └── prod.eessi-hpc.org/
        └── 2020.06/
            ├── compat/
            │   ├── aarch64
            │   ├── ppc64le
            │   └── x86_64/
            │       ├── bin
            │       ├── etc
            │       ├── lib64
            │       ├── sbin
            │       └── usr
            └── software/
                ├── aarch64
                ├── ppc64le
                └── x86_64
```

EESSI directory structure


```
.
└── cvmfs/
    ├── cvmfs-config.eessi-hpc.org
    ├── test.eessi-hpc.org
    └── prod.eessi-hpc.org/
        └── 2020.06/
            ├── compat
            └── software/
                ├── aarch64
                ├── ppc64le
                └── x86_64/
                    ├── amd/
                    │   └── zen2
                    └── intel/
                        ├── haswell
                        └── skylake/
                            ├── modules
                            └── software/
                                ├── GROMACS/
                                │   ├── 2019.3-fosscuda-2019b
                                │   └── 2020.1-foss-2020a-Python-3.8.2
                                └── TensorFlow/
                                    └── 2.2.0-fosscuda-2019b-Python-3.7.4
```

CernVM-FS in EESSI: current status

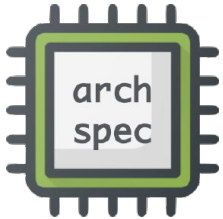
- Ansible role and playbooks for deploying all components
  - See https://github.com/EESSI/filesystem-layer
  - Based on: https://galaxy.ansible.com/galaxyproject/cvmfs
- Pilot CernVM-FS repository for playing around and testing
  - Currently only contains Prefix installation with Lmod + EasyBuild
- One Stratum 1 at the moment
  - Add more (geographically distributed) servers in the near future

Current focus: complete pilot setup

- Working towards a functioning pilot version
  - Limited scope in terms of architectures, software, client OS
  - Initial version of documentation
  - Find roadblocks (and overcome then)
  - At the moment mostly focusing on getting some software installed
- Working on documentation, website, Github repos, Twitter, …
- Scale up to a more production-ready service (see next slide)

Future work

- Scale up by adding more CernVM-FS infrastructure (Stratum 1, proxies)
- Optimize CernVM-FS settings
- Test and production repos
- Improve and extend documentation
- Support more architectures (CPUs, GPUs, interconnects) and client OSes
    - Use archspec to determine the right tree in the repository



Questions

- What kind of optimizations are recommended for such repositories?
  - Generate catalogs: automatically? cvmfsdirtab?
  - Single dir at top level
- Use https?
- What's the best way to allow multiple users to install software in the right tree?
  - Different build nodes with different architectures
  - Build process should find deps in and install the software to /cvmfs
  - Use gateway/publishing mechanism (possibly inside a Docker container)?

Questions

- Good idea to use DNS entries to have all Stratum 0/1 servers available as cvmfs-s[0,1]-[sitename].eessi-hpc.org, and use these in the configuration files?
- Experiences/recommendations for installing software stack in non-default location, in particular w.r.t. compiler (--with-sysroot in GCC) and avoid picking up libraries from host OS when building?