WWU MÜNSTER

Bundesministerium für Bildung und Forschung

Multiscale

EuroHPC Joint Undertaking

EESSI -going scientific software installations

EUROPEAN ENVIRONMENT FOR SCIENTIFIC SOFTWARE INSTALLATIONS

Alexander Puck Neuwirth [1]

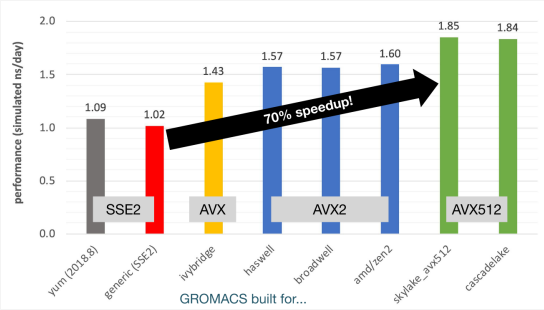
[1] alexander.neuwirth@uni-muenster.de

Motivation

Reduce administration effort (HPC):

- More software and version
- More users
- Different hardware
  - CPUs: AMD, Intel, ARM, ...
  - GPUs: NVIDIA, AMD, ...
- Scientific software is complex

Performance:



- Using latest instructions sets like AVX-512 is crucial
- Saves resources

Scope & Goals

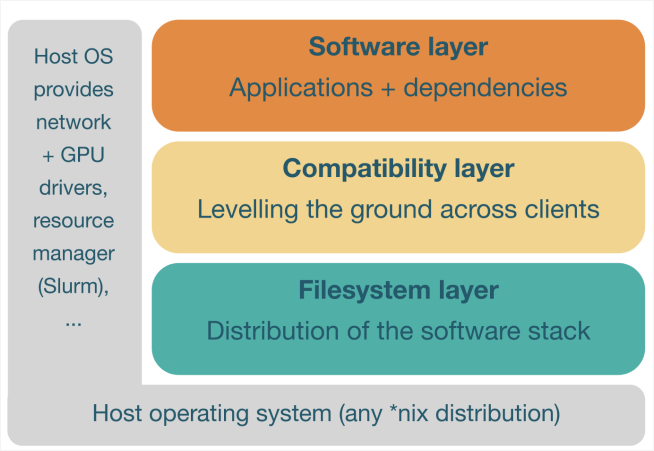
European Environment for Scientific Software Installations[2]

- Collaboration in HPC community to provide an **optimized software stack** across HPCs
  $\Longrightarrow$ less duplication of effort, maintenance
- Uniform user experience on HPC, HTC, cloud, GPU and personal systems
- Streaming of software, avoid long download and compile times
- Better automation, testing, reproducibility, continous integration and benchmarking
- Focus on **performance** and **portability**

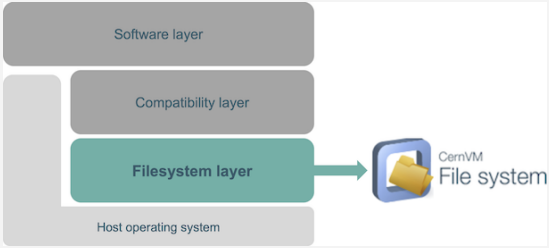
Heavily inspired by Compute Canada/Digital Research Alliance of Canada

[2] Dröge, B. et al. Software: Practice and Experience **53,** 176–210. ISSN: 1097-024X. http://dx.doi.org/10.1002/spe.3075 (Feb. 2022).

Layered structure



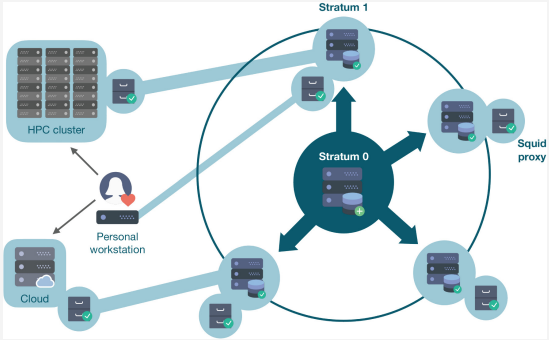
Filesystem layer



- Distributed file system optimized for scalable software delivery
- Highly reliable, scalable, atomic udpates
- Clients mount **read-only** POSIX file system via FUSE with HTTP transport

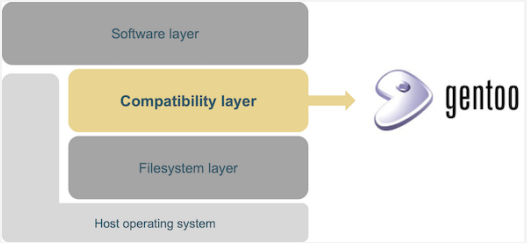
  $\Longrightarrow$ Bypass firewalls ...
- EESSI for now does only distributes free open source software since its **public**ly available

CERN Virtual Machine File System



- Used in high-energy physics community to **globally distribute software**
- Clients pull files on demand, with aggressive caching
- Transparent deduplication, chunking and compression
- Corollary: unpacked container images
  - Faster start up times, since only the necessary files are pulled
  - Compression even across layers/images

Compatibility layer



Currently supported microarchitectures:

- aarch64/generic
- aarch64/neoverse_n1
- aarch64/neoverse_v1
- x86_64/amd/zen2
- x86_64/amd/zen3
- x86_64/intel/haswell
- x86_64/intel/skylake_avx512
- x86_64/generic

- Linux installation in a non standard location (here read-only CVMFS)
- Levels the ground between operating systems
- Does not need to be fully optimized, only used to build the software stack
- Each family of processors (x86_64, aarch64) needs a thin gentoo-prefix
- Can be security patched (glibc, zlib, ...)

Gentoo Prefix[3]

Gentoo:

- **Source-based** distribution of Linux
- Highly **customizable** allowing performance and efficiency optimizations
- Supports **many architectures**

[3] Xu, B. et al. EPJ Web of Conferences **245** (eds Doglioni, C. et al.) 05036. ISSN: 2100-014X. http://dx.doi.org/10.1051/epjconf/202024505036 (2020).

Gentoo Prefix[3]

Gentoo:

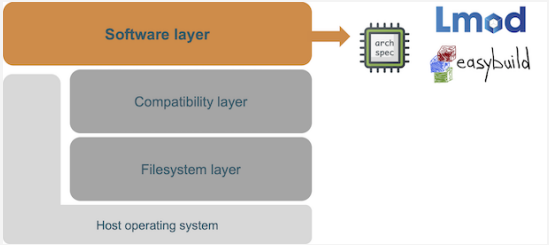
- **Source-based** distribution of Linux
- Highly **customizable** allowing performance and efficiency optimizations
- Supports **many architectures**

Prefix (from `./configure --prefix="..."`):

- Minimalistic, self-contained Gentoo environment
- Gentoo Prefix installs within an offset, or prefix, allowing installation in alternative locations and preventing conflicts.
- Operates **unprivileged without root access or chroot**
- Currently, Gentoo Prefix is successfully utilized on various systems including macOS, Linux, Windows (WSL, Cygwin) and Solaris 11.
- Can be used as a package manager for a non-Gentoo system

[3] Xu, B. et al. EPJ Web of Conferences **245** (eds Doglioni, C. et al.) 05036. ISSN: 2100-014X. http://dx.doi.org/10.1051/epjconf/202024505036 (2020).

Software layer



- Optimized for the target microarchitecture, built and tested automatically
- No dependencies on host libraries

archspec: Selects the best suited part of the software stack for a particular host system architecture

Lmod: Environment modules tool widely adopted in HPCs

EasyBuild: Framework for building and installing optimized software (typically HPC)

Demo: CVMFS installation

CVMFS installation for debian (others similar), else apptainer/singularity available

```
## Install CVMFS
# CERN repo key
$ sudo wget https://cvmrepo.web.cern.ch/cvmrepo/apt/cernvm.gpg -O \
  /etc/apt/trusted.gpg.d/cernvm.gpg
# CERN repo
$ sudo apt-add-repository "deb http://cvmrepo.web.cern.ch/cvmrepo/apt/\
   bookworm-prod main"
# fetch
$ sudo apt-get update
# install
$ sudo apt-get install -y cvmfs
## Tune CVMFS settings in /etc/cvmfs/default.local then
$ sudo cvmfs_config setup
```

Demo: Easy to use

On a system with CVMFS installed

```
## CVMFS fetches directory on ls
$ ls /cvmfs/software.eessi.io/versions/
2023.06
$ source /cvmfs/software.eessi.io/versions/2023.06/init/bash
Found EESSI repo @ /cvmfs/software.eessi.io/versions/2023.06!
archdetect says x86_64/intel/haswell
Using x86_64/intel/haswell as software subdirectory.
Environment set up to use EESSI (2023.06), have fun!
{EESSI 2023.06} USER@PC:~$ module avail
[...] # many modules

{EESSI 2023.06} USER@PC:~$ command -v rivet
{EESSI 2023.06} USER@PC:~$ module load Rivet
{EESSI 2023.06} USER@PC:~$ command -v rivet-mkhtml
/cvmfs/software.eessi.io/versions/2023.06/software/linux/x86_64/intel/\
haswell/software/Rivet/3.1.9-gompi-2023a-HepMC3-3.2.6/bin/rivet
```

Examples: EESSI + direnv

- `direnv` automatically sets and unsets environment variables on a directory level
- Hooked in `.bashrc`: `eval "$(direnv hook bash)"`

`.envrc` file in root of repo:

```
myproject $ cat .envrc
source /cvmfs/software.eessi.io/versions/2023.06/init/bash
module purge
module load Nextflow
module load TensorFlow
```

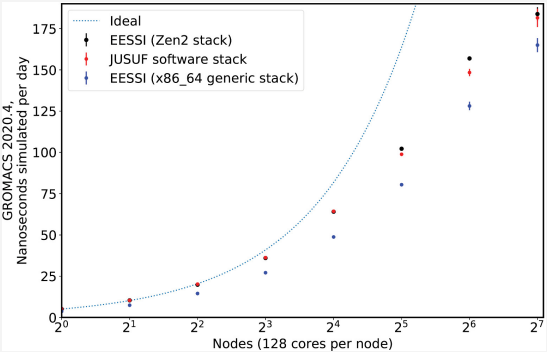
- Activates the modules on `cd` into the directory, after `direnv allow`ing once
- More convenient and less storage/networking required than containers.
- All developers have a consistent environment

Examples: GitHub Actions

```yaml
name: Tensorflow example https://github.com/EESSI/github-action-eessi
on: [push, pull_request]
jobs:
    build:
        runs-on: ubuntu-latest
        steps:
        - uses: actions/checkout@v3
        - uses: eessi/github-action-eessi@v3
          with:
            eessi_stack_version: '2023.06'
        - name: Test EESSI
          shell: bash
          run: |
            python -c 'import tensorflow; print(tensorflow.__version__)'
            module load CMake
```

- Loads from `.envrc`, simplifies devop and pipeline constuction
- Independent of underlaying image/container/OS

Performance Evaluation



Program: GROMACS, molecular dynamics for simulations of proteins, lipids, and nucleic acids

CPU: AMD EPYC with HDR100 interconnect

MPI: 12 GB/s OpenMPI bandwidth

Conclusion

EESSI enables a variety of use cases:

- A uniform software stack across HPC clusters, clouds, servers, and laptops
- Can be leveraged in continuous integration (CI) environments
- Significantly simplifies setting up infrastructure for HPC (+training)
- Enhanced collaboration with software developers and application experts
- Enable portable workflows

Thank you!

EESSI is powered by FOSS

EESSI

easybuild

- Installation tool for scientific software
- Optimises for build host (by default)
- Supports over 2,000 software pkgs

https://easybuilders.github.io/easybuild

archspec

- Python library
- Detect processor type
- Check compatibility with host CPU

https://github.com/archspec

Lmod

- Environment modules tool (written in Lua)
- Intuitive access to software installations
- Multiple software versions side-by-side

https://lmod.readthedocs.io

gentoo

https://wiki.gentoo.org/wiki/Project:Prefix

- Linux distribution, installs from source
- Prefix project: install packages in <prefix>
- Supports x86_64, Arm64, POWER, ... & Linux, macOS

ReFrame

- Regression testing framework for HPC
- Verify correctness
- Check performance
- Tests are implemented in Python

https://reframe-hpc.rtfd.io

CernVM-FS

- Software distribution service
- Scalable, read-only, globally distributed filesystem
- Mount filesystem over HTTP

https://cernvm.cern.ch/fs

ANSIBLE

- Automation
- Configuration mgmt

Singularity

- Build isolation
- Easy access for clients

CLUSTER IN THE CLOUD

HashiCorp Terraform

- Creating cloud clusters / instances on demand

4

[4] Dröge, B. en. 2021. https://av.tib.eu/media/53664.