Webinar series: Different aspects of EESSI

5 Mondays in a row May-June 2025

https://eessi.io/docs/training/2025/webinar-series-2025Q2

- ○ **Introduction to EESSI webinar/tutorial** (today)

- ○ Introduction to CernVM-FS (12 May)

- ○ Introduction to EasyBuild (19 May)

- ○ EESSI for CI/CD (26 May)

- ○ Using EESSI as the base for a system stack (2 June)

More info and registration →

What if you no longer have to install

*a **broad range of scientific software***

from scratch on every laptop, HPC cluster,

or cloud instance you use or maintain,

without compromising on performance?
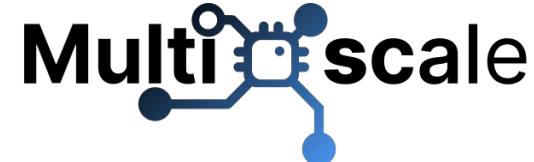
EESSI

EUROPEAN ENVIRONMENT FOR
SCIENTIFIC SOFTWARE INSTALLATIONS
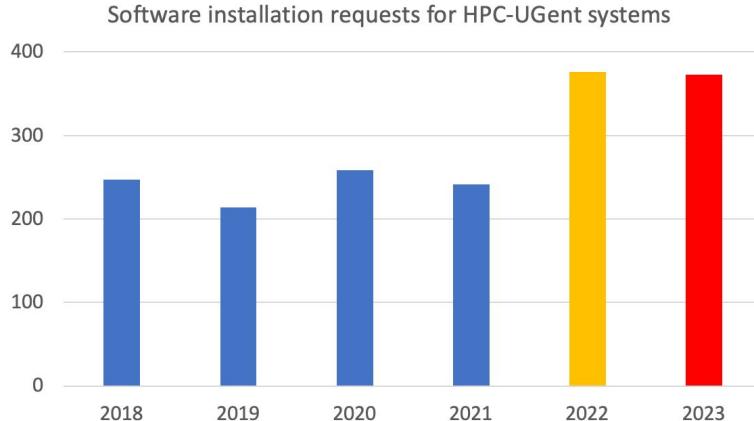
The changing landscape of scientific computing

- **Explosion of available scientific software** applications (bioinformatics, AI boom, …)

- Increasing interest in **cloud** for scientific computing (flexibility!)

- **Increasing variety in processor (micro)architectures** beyond Intel & AMD:
  Arm is ~~coming~~ already here (see Fugaku, JUPITER, …), RISC-V is coming (soon?)

- In strong contrast: available (wo)manpower **in HPC support teams is (still) limited**…

Software installation requests for HPC-UGent systems
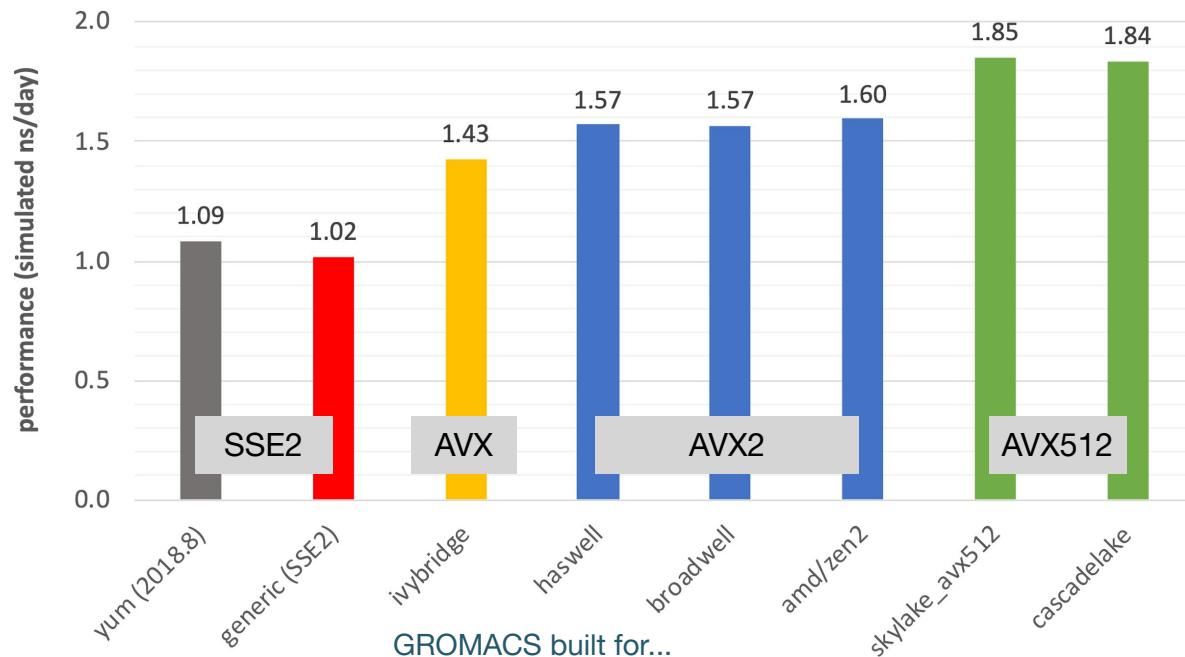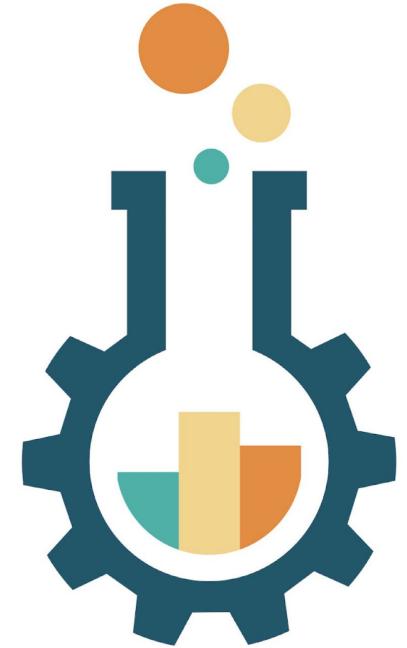
Major goals of EESSI

- Providing a truly **uniform software stack**

  - Use the (exact) same software environment everywhere

  - **Without sacrificing performance** for "mobility of compute"
    (like is typically done with containers/conda)

- **Avoid duplicate work** (for researchers, HPC support teams, sysadmins, ...)

  - Tools that automate software installation process
    (EasyBuild, Spack) are not sufficient anymore

  - Go beyond sharing build recipes => work towards a shared software stack

- Facilitate HPC training, development of (scientific) software, ...

Optimized scientific software installations

- Software should be optimized for the system it will run on (keep the P in HPC!)

- Impact on performance is often significant for scientific software!

- Example: GROMACS 2020.1 (PRACE benchmark, Test Case B)

- Metric: (simulated) ns/day, higher is better

- Test system: dual-socket Intel Xeon Gold 6420 (Cascade Lake, 2x18 cores)

- **Performance of different GROMACS binaries, on exact same hardware/OS**

Demo time!

Demo: Using EESSI

```
/cvmfs/software.eessi.io/versions/2023.06/software
`-- linux
    |-- aarch64
    |   |-- generic
    |   |-- neoverse_n1
    |   `-- neoverse_v1
    `-- x86_64
        |-- amd
        |   |-- zen2
        |   `-- zen3
        |-- generic
        `-- intel
            |-- haswell
            `-- skylake_avx512
                |-- modules
                `-- software
```
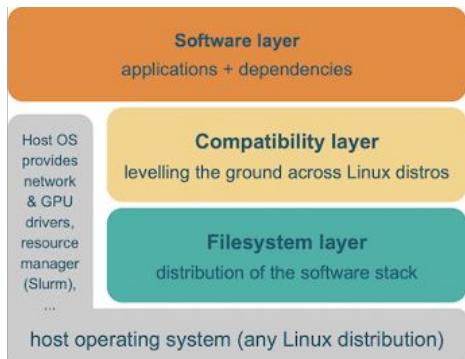
```
$ source /cvmfs/software.eessi.io/versions/2023.06/init/bash
Found EESSI pilot repo @
/cvmfs/software.eessi.io/versions/2023.06!
archdetect says x86_64/amd/zen3
Using x86_64/amd/zen3 as software subdirectory
...              Automatically detects CPU microarchitecture
Environment set up to use EESSI pilot software stack, have fun!

{EESSI 2023.06} $ module load R/4.3.2-gfbf-2023a

{EESSI 2023.06} $ which R
/cvmfs/software.eessi.io/versions/2023.06/software/linux/x86_64/
amd/zen3/software/R/4.3.2-gfbf-2023a/bin/R

{EESSI 2023.06} $ R --version
R version 4.3.2
```
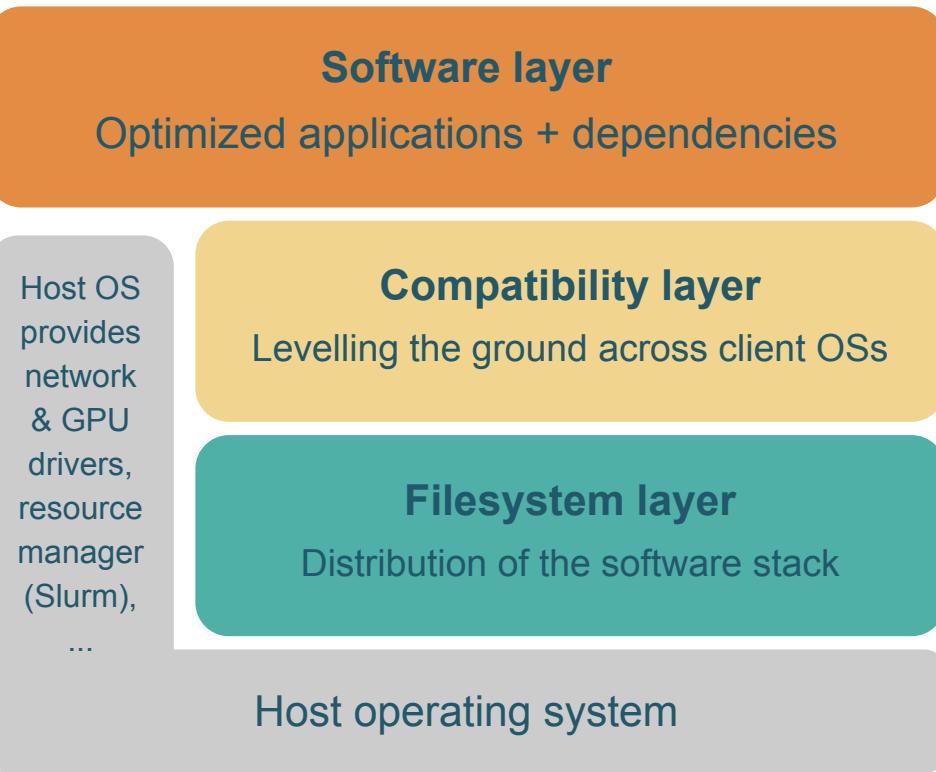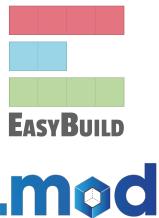
EESSI

# How does EESSI work?

- Software installations included in EESSI are:

  - Automatically **"streamed in" on demand** (via CernVM-FS)

  - Built to be **independent of the host operating system**

    "Containers without the containing"

  - **Optimized** for specific CPU generations + specific GPU types

- Initialization script **auto-detects** CPU + GPU of the system

High-level overview of EESSI
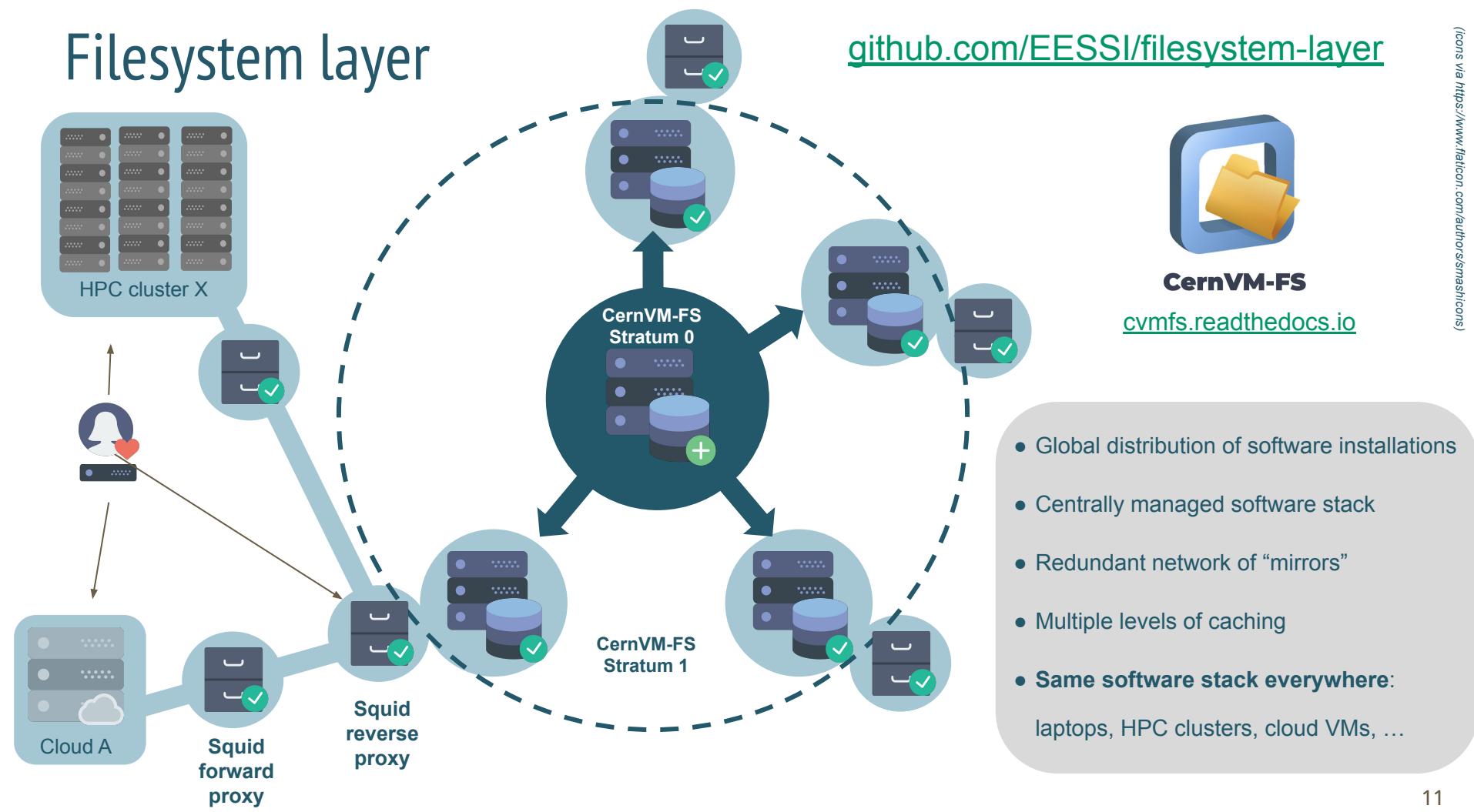
Filesystem layer

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

CernVM-FS

cvmfs.readthedocs.io

HPC cluster X

CernVM-FS
Stratum 0

CernVM-FS
Stratum 1

Cloud A

**Squid
forward
proxy**

**Squid
reverse
proxy**

- Global distribution of software installations

- Centrally managed software stack

- Redundant network of "mirrors"

- Multiple levels of caching

- **Same software stack everywhere**:

  laptops, HPC clusters, cloud VMs, …

11
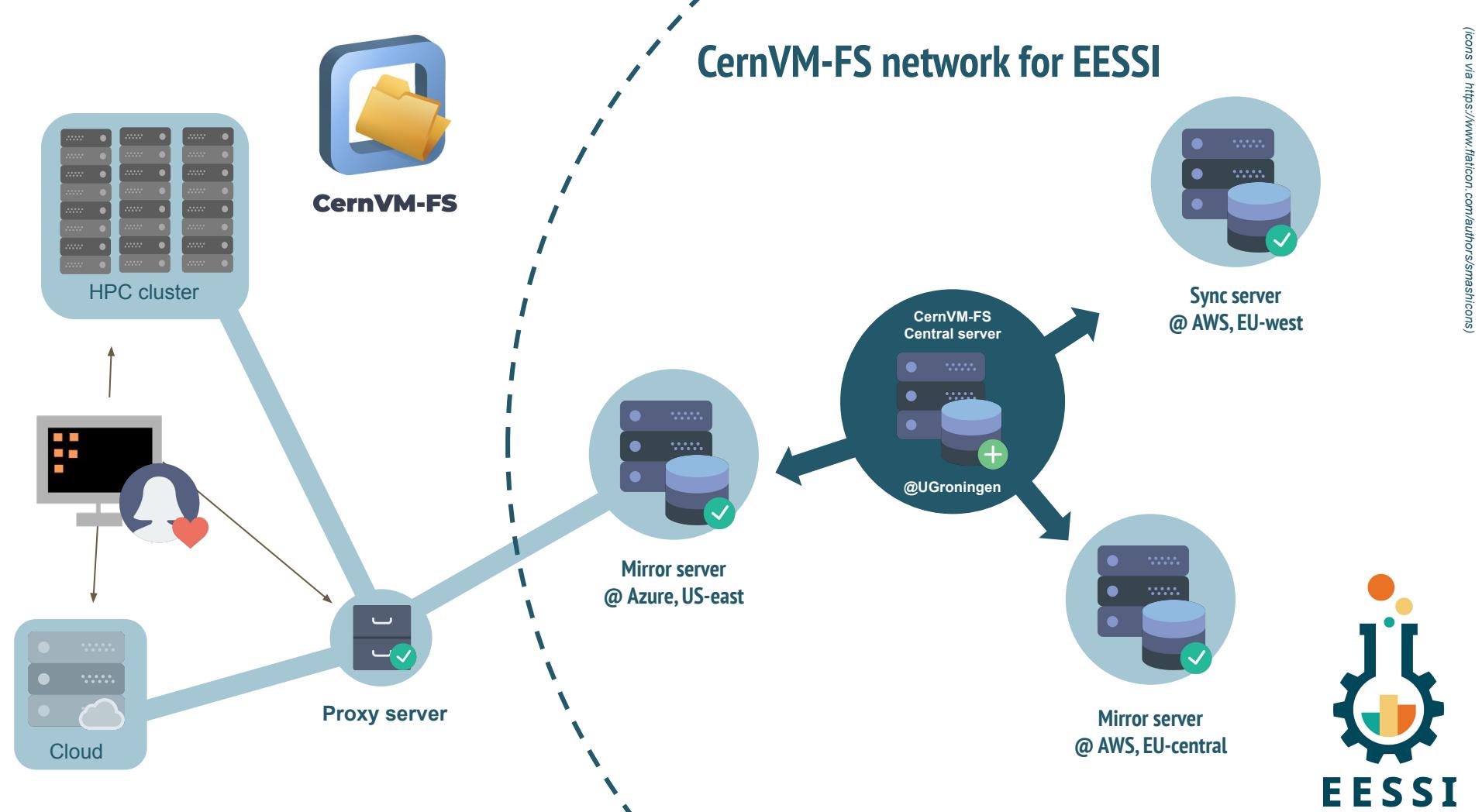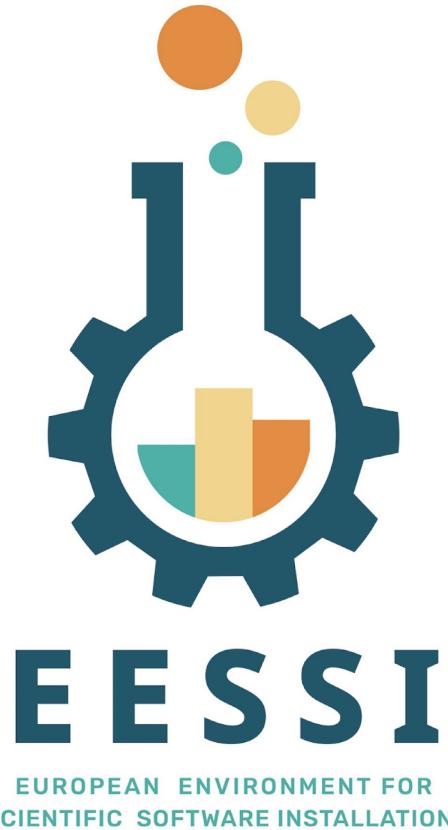
CernVM-FS network for EESSI

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

CernVM-FS

HPC cluster

Cloud

Proxy server

Mirror server
@ Azure, US-east

CernVM-FS
Central server

@UGroningen

Sync server
@ AWS, EU-west

Mirror server
@ AWS, EU-central

EESSI

CernVM-FS webinar

Next Monday!
12 May 2025, 13:30-15:30 CEST

https://eessi.io/docs/training/2025/webinar-series-2025Q2

13
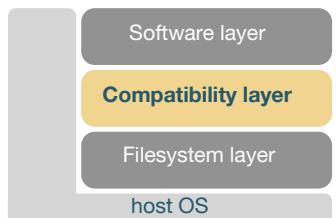
Compatibility layer

- Gentoo Prefix installation (in `/cvmfs/.../compat/<os>/<arch>/`)

- **Set of Linux tools & libraries installed in non-standard location**

- Limited to low-level stuff, incl. glibc (no Linux kernel or drivers)

- Similar to the OS layer in container images

- Only targets a supported **processor family** (`aarch64`, `x86_64`, `riscv64`)

- **Levels the ground for different client operating systems** (Linux distributions)

- Currently in production repository:

  `/cvmfs/software.eessi.io/versions/2023.06/compat/linux/aarch64`

  `/cvmfs/software.eessi.io/versions/2023.06/compat/linux/x86_64`

powered by

gentoo

Software layer
Compatibility layer
Filesystem layer
host OS

14
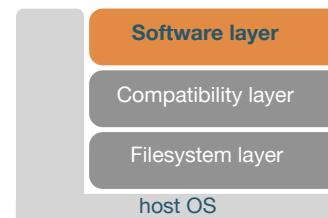
Software layer

- Provides **installations of scientific software** applications & libraries (incl. deps)

- Optimized for specific CPU microarchitectures (AMD Zen3, ...)

  - Separate subdirectory/tree for each (in `/cvmfs/.../software/...`)

- Support for specific generation of **(NVIDIA) GPUs** via `/accel/` subdirectories

- **Leverages libraries** (like glibc) **from compatibility layer** (*not* from host OS)

- Installed with EasyBuild, incl. environment module files

- Lmod environment modules tool is used to access installations

- **Best subdirectory for host is selected automatically** via archdetect

EESSI

powered by

EASYBUILD.io

Lmod

arch spec

| Software layer |
| Compatibility layer |
| Filesystem layer |
| host OS |

15
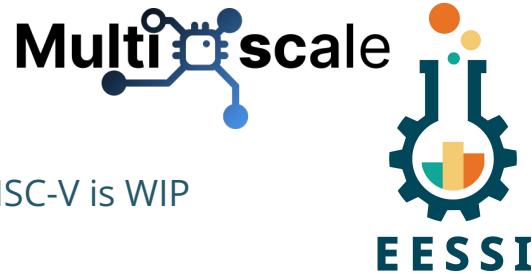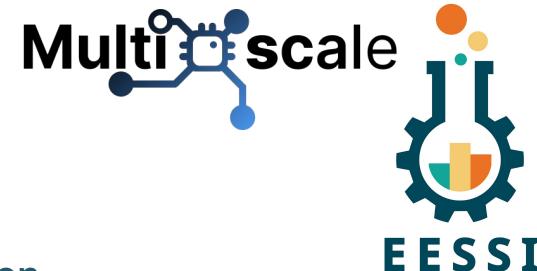
Supported system architectures
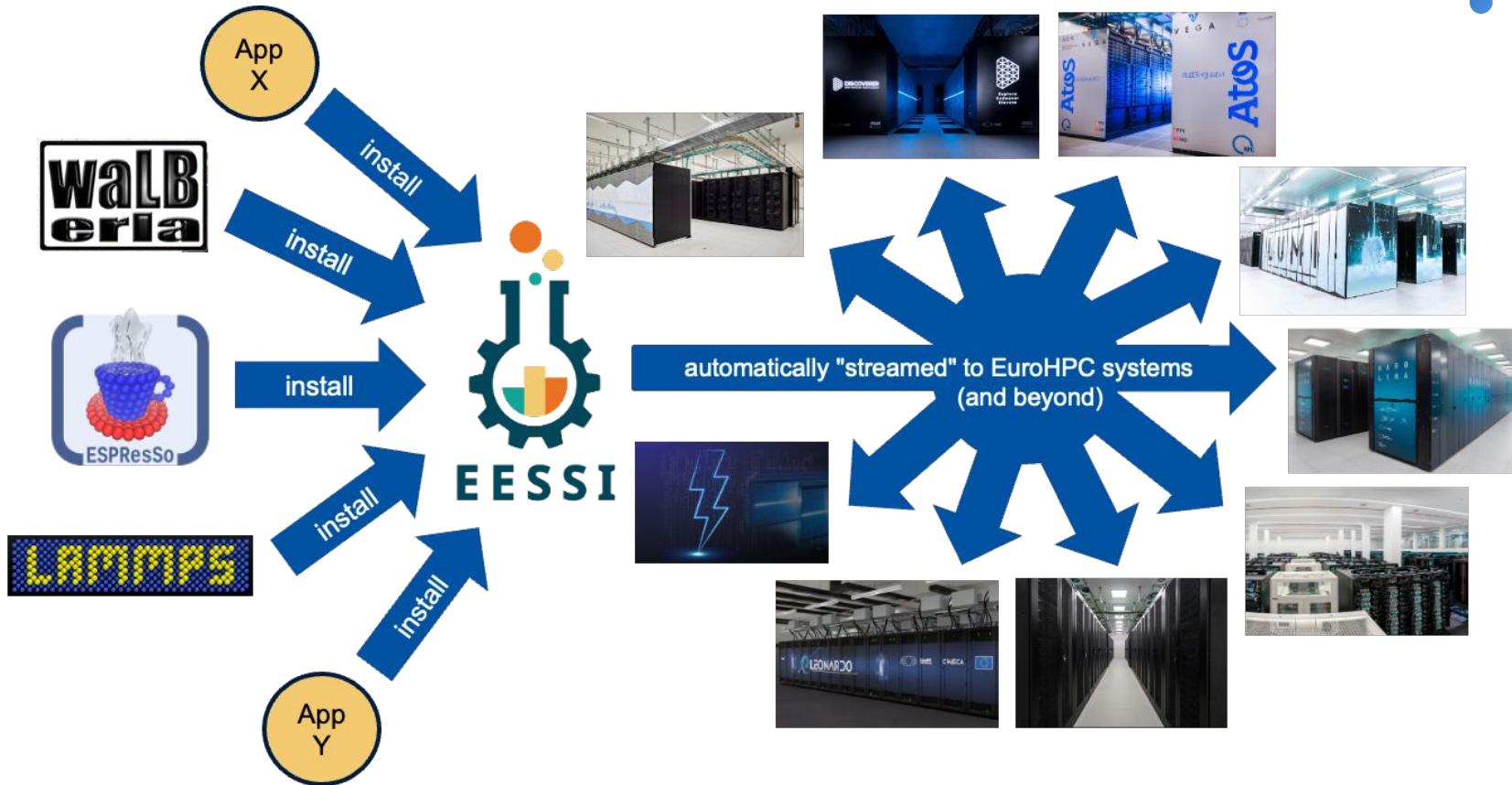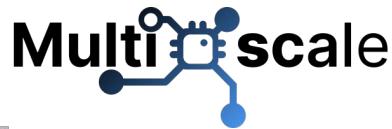
- Different generations of `x86_64` (Intel, AMD) and Arm 64-bit CPUs; RISC-V is WIP
  - Including A64FX (Deucalion, WIP) & NVIDIA Grace (JUPITER, WIP)
  - Also works on laptops, in virtual machines in the cloud, on Raspberry Pi boards, etc.

- Different accelerators: **NVIDIA GPUs** (today) + **AMD GPUs** (soon)
  - Available combination: AMD Rome (Zen2) + NVIDIA A100 (cc80), AMD Milan (zen3) + NVIDIA A100 (cc80) and AMD Genoa (zen4) + NVIDIA H100 (cc90) only software installations for AMD Rome (Zen2) + NVIDIA A100 are available

- **Various interconnects** like Infiniband, via "fat" MPI libraries
  - Support for injecting a vendor-provided MPI library is available

- Goal is to support system architecture of **all** (current & future) **EuroHPC systems**

On which systems is EESSI already available?
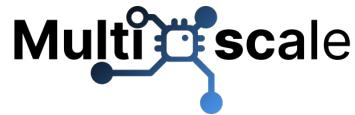
- EuroHPC JU systems:

  - Native installation (via CernVM-FS) on **Vega** + **Karolina + Deucalion**

  - Semi native installation (via rsync) on **MareNostrum5**

  - EESSI can be used via `cvmfsexec` tool on Deucalion, Discoverer, MeluXina ([see blog post](#))

  - Native installation on **MeluXina**, **Lumi** is a work-in-progress

  - JSC has expressed significant interest to make EESSI available on **JUPITER**

- EESSI is already available on various other European systems (and beyond)

  - Snellius @ SURF, EMBL, Univ. of Stuttgart, VSC sites in Belgium, Sigma2 in Norway, etc.

- **Overview of (known) systems that have EESSI available at [eessi.io/docs/systems](#)**

EESSI as a shared software stack

automatically "streamed" to EuroHPC systems (and beyond)

NVIDIA GPU support in EESSI
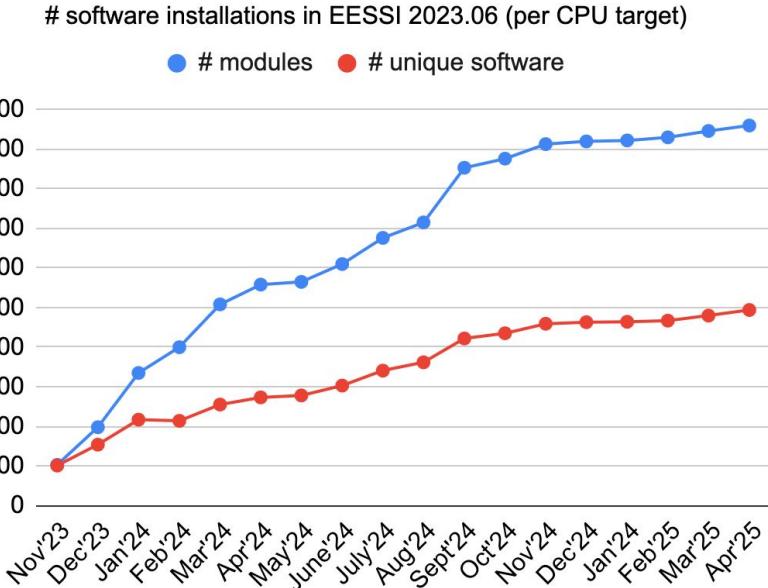
- Initial support for CUDA software is in place in EESSI version 2023.06

- Detailed documentation available at eessi.io/docs/gpu

- Problems we had to deal with:

    1) We don't know where the **NVIDIA GPU driver libraries** are in host OS…

    2) We **can not redistribute the full CUDA installation** due to EULA, only runtime libraries…

- In EESSI, we provide scripts to deal with both these problems:

    1) `link_nvidia_host_libraries.sh` script to link GPU driver libraries provided by OS "into" EESSI; (requires write access to (target of) `/cvmfs/software.eessi.io/host_injections`)

    2) `install_cuda_host_injections.sh` script to **install full CUDA installation** to subdirectory of (target of) `/cvmfs/software.eessi.io/host_injections` (and unbreak symlinks in CUDA in EESSI)

- **Available CUDA software in EESSI:** CUDA-Samples, GROMACS, ESPResSo, LAMMPS, NCCL, OSU Micro-Benchmarks

- More CPU/GPU combos and software (PyTorch, TensorFlow, AlphaFold, …) coming soon…

- Testing of the software we install complicates the process of adding GPU software

Overview of available software

Currently ~960 software installations available
per CPU target via software.eessi.io CernVM-FS repository;
increasing every week

- Almost 500 different software packages

- Excl. extensions: Python packages, R libraries

- Including ESPResSo, GROMACS, LAMMPS,
  OpenFOAM, PyTorch, R, QuantumESPRESSO,
  TensorFlow, waLBerla, WRF, …

- eessi.io/docs/available_software/overview

- Using recent compiler toolchains: currently
  focusing on `foss/2023a` and `foss/2023b`

software installations in EESSI 2023.06 (per CPU target)

● # modules   ● # unique software
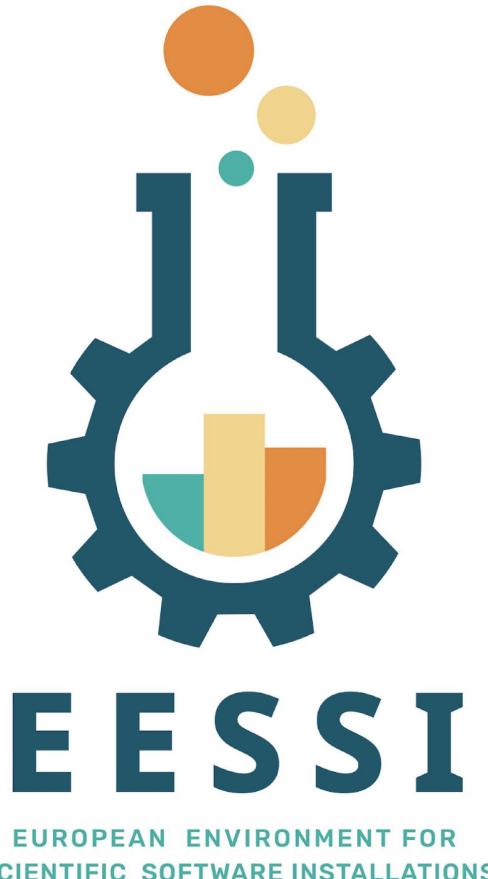
Current status of EESSI

- Production CernVM-FS repository `software.eessi.io` available since Nov'23

- Ansible playbooks, scripts, docs available at https://github.com/eessi

- Target CPU microarchitectures (see also
  https://eessi.io/docs/software_layer/cpu_targets):

  ```
  {aarch64,x86_64}/generic
  intel/{haswell, skylake_avx512}, amd/{zen2,zen3,zen4},
  aarch64/{neoverse_n1,neoverse_v1,a64fx}
  ```

- **NVIDIA GPU support in place**, limited set of GPU software installed

- **Supported by Azure and AWS**: sponsored credits to develop necessary infrastructure

Hands-on live demo

Getting access to EESSI

Using EESSI

Richard

Native installation of CernVM-FS

- For a single system, it's sufficient to install and configure CernVM-FS client

- For an HPC cluster, a bit more work is needed to:
  - Enhance the reliability of the access to EESSI
  - Improve startup performance of software

- It is recommended to:
  - Have a full copy of the EESSI repositories in your local network, by setting up a private CernVM-FS Stratum-1 "mirror" server
  - Have one or more proxy servers, to offload the Stratum-1 server(s)

- See also https://multixscale.github.io/cvmfs-tutorial-hpc-best-practices

Next Monday!
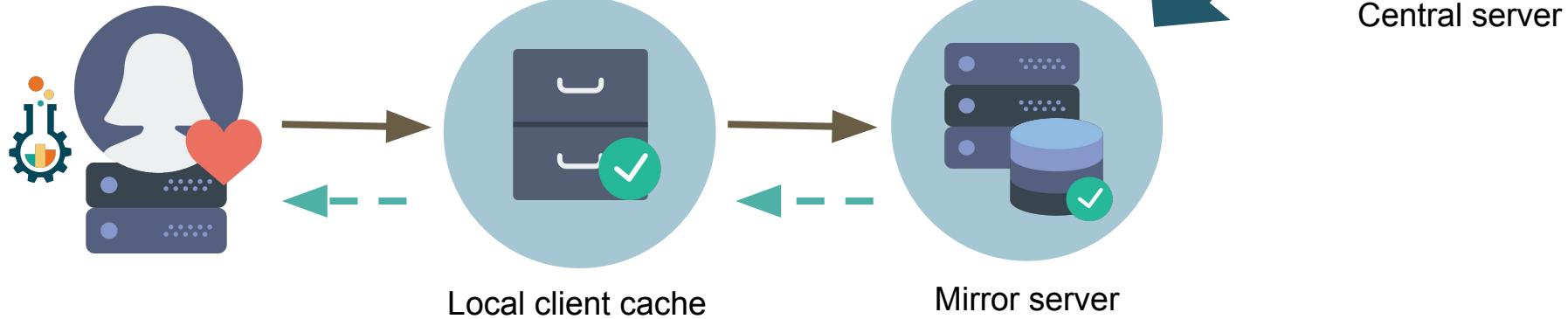
12 May 2025, 13:30-15:30 CEST

CernVM-FS
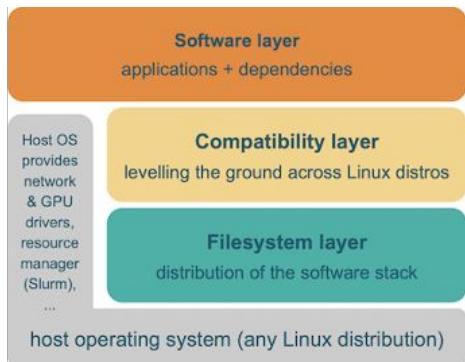
The EESSI User Experience

```
$ source /cvmfs/software.eessi.io/versions/2023.06/init/bash
{EESSI 2023.06} $ module load GROMACS/2024.1-foss-2023b
{EESSI 2023.06} $ gmx mdrun ...
```


Central server

Local client cache

Mirror server

EESSI provides **on-demand streaming** of (scientific) software (like music, TV-series, …)

Richard

How does EESSI work again?

- Software installations included in EESSI are:

  - Automatically **"streamed in" on demand** (via CernVM-FS)

  - Built to be **independent of the host operating system**

    "Containers without the containing"

  - **Optimized** for specific CPU generations + specific GPU types

- Initialization script **auto-detects** CPU + GPU of the system

Software layer
applications + dependencies

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

Compatibility layer
levelling the ground across Linux distros

Filesystem layer
distribution of the software stack

host operating system (any Linux distribution)

Native installation of CernVM-FS

```
# Native installation
# Installation commands for RHEL-based distros
# like CentOS, Rocky Linux, Almalinux, Fedora, …

# install CernVM-FS
sudo yum install -y

https://ecsft.cern.ch/dist/cvmfs/cvmfs-release/cvmfs-release-latest.noarch.rpm
sudo yum install -y cvmfs

# create client configuration file for CernVM-FS
# (no proxy, 10GB local CernVM-FS client cache))
sudo bash -c "echo 'CVMFS_CLIENT_PROFILE="single"' > /etc/cvmfs/default.local"
sudo bash -c "echo 'CVMFS_QUOTA_LIMIT=10000' >> /etc/cvmfs/default.local"

# Make sure that EESSI CernVM-FS repository is accessible
sudo cvmfs_config setup
```

See docs for alternative ways of installing CernVM-FS natively, via a VM on a personal computer
eessi.io/docs/getting_access/eessi_wsl - eessi.io/docs/getting_access/eessi_limactl

Richard

26

EESSI demo

Case 1: Native installation + configuration of CernVM-FS to access EESSI

o   On x86_64 (AMD CPUs) virtual machine in AWS

o   Install & configure CernVM-FS (requires root access)

o   Use case: laptops, virtual machines, or any system with administrative privileges

Case 2: Accessing EESSI via container that includes CernVM-FS

o   On aarch64 (Arm CPUs) virtual machine in AWS

o   Run `eessi_container.sh` script (no root access required)

o   Requires Apptainer 1.0.0 (or newer) or Singularity 3.7.x , Git

o   Use case: HPC systems where Apptainer is available but no native installation of CernVM-FS

Richard

Example: Running LAMMPS in a Slurm job script


```
#!/bin/bash
#SBATCH --job-name="EESSI_Demo_LAMMPS_lj"
#SBATCH --ntasks=4
#SBATCH --ntasks-per-node=4
#SBATCH --cpus-per-task=1
#SBATCH --output=EESSI_demo.out
#SBATCH --error=EESSI_demo.err
#SBATCH --time=0:30:0
#SBATCH --partition=cpu_rome

# CPU microarchitecture is automatically detected by EESSI init script
source /cvmfs/software.eessi.io/versions/2023.06/init/bash
module load LAMMPS/29Aug2024-foss-2023b-kokkos
mkdir /tmp/$USER && cd /tmp/$USER
curl -o in.lj https://raw.githubusercontent.com/lammps/lammps/refs/heads/develop/bench/in.lj
export OMP_NUM_THREADS=1
mpirun -np 4 lmp -in in.lj
rm -r /tmp/$USER
```

EESSI

CernVM-FS

Richard

# What if the software you need is not provided by EESSI yet ?
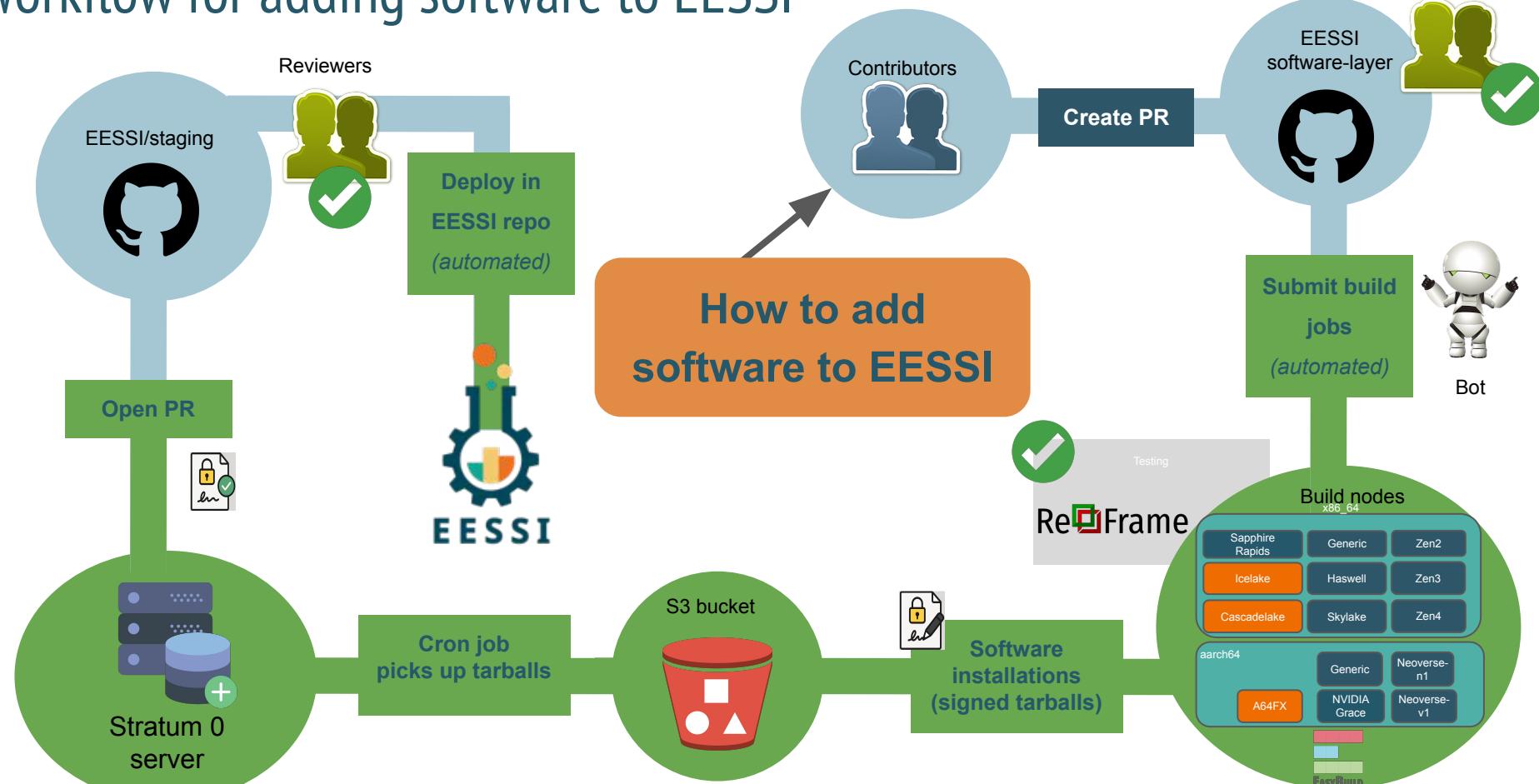
- Build software locally on top of EESSI with EasyBuild:

  - https://www.eessi.io/docs/using_eessi/building_on_eessi

- Contribute to the EESSI software stack and make the software available for others!

  - Contributions to the EESSI software stack are welcome:
    https://eessi.io/docs/adding_software/overview

  - The contribution policy is to provide guidelines for adding software to EESSI:
    https://www.eessi.io/docs/adding_software/contribution_policy

    For example: software that can be redistributed (open source), …

Richard

Workflow for adding software to EESSI

eessi.io/docs/software_layer/adding_software

Software testing is an important part of EESSI

- Smoke test: sanity check commands are run by EasyBuild

  to check that installed software is not horribly broken while using EasyBuild

- Regression testing via EESSI test suite: eessi.io/docs/test-suite

  - Collection of portable tests for software available in EESSI

  - Running on selected (single node) tests when building new
    software for EESSI (before deployment)

  - Periodically (daily/weekly) on about multiple different systems

  - Can also be used for other software stacks (that are built with EasyBuild)

  - Periodic runs of EESSI test suite help to catch performance regressions

Richard

Software testing is an important part of EESSI

- Example: failing tests in GROMACS test suite when installing it in EESSI

  - [Fixed](#) in GROMACS 2024.3

  - See https://gitlab.com/eessi/support/-/issues/47

  - Filesystem race in GROMACS test suite when running tests concurrently

  - **Bug in Arm SVE support**, leading to (very) wrong results for several tests

    - See https://gitlab.com/gromacs/gromacs/-/issues/5057

   - Works fine on A64FX (512-bit SVE), but problem on Graviton 3 + NVIDIA Grace!

Richard
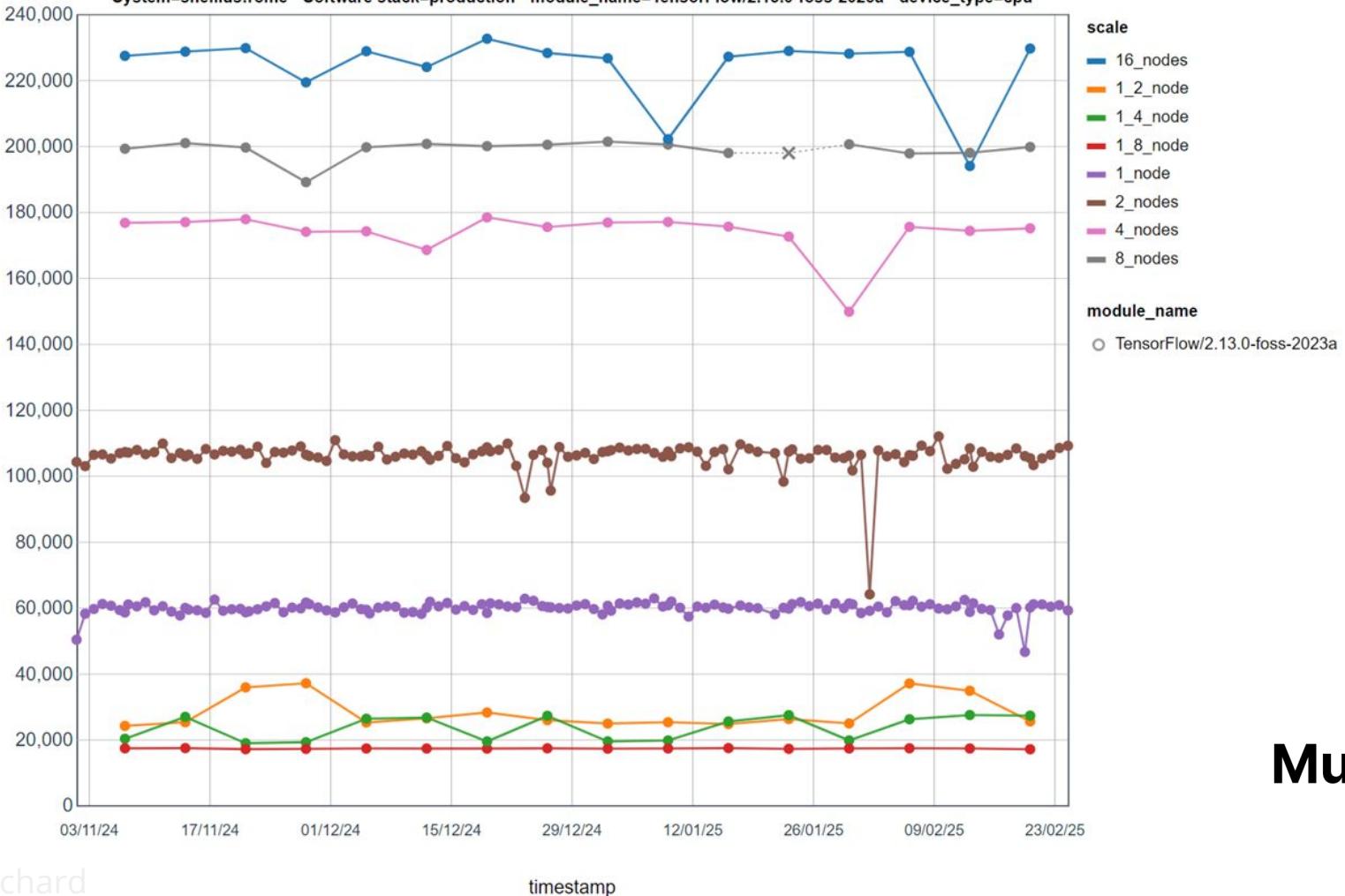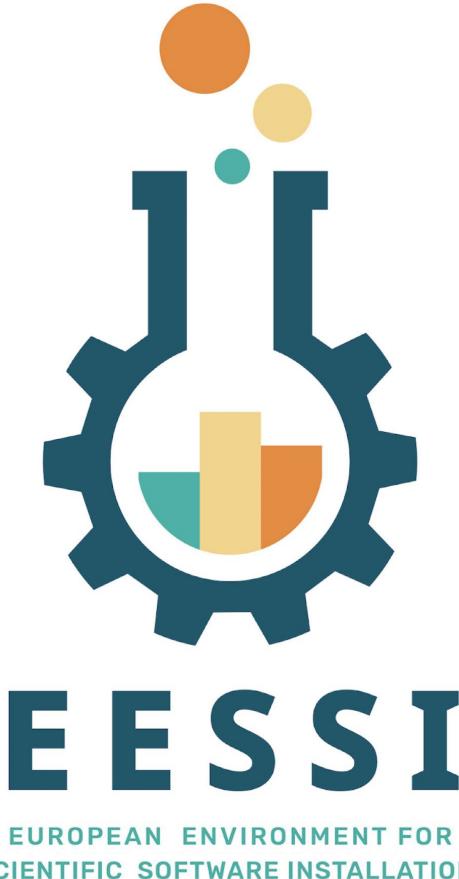
EESSI dashboard

- The EESSI dashboard offers a centralized interface for visualizing results generated by the EESSI test suite. It supports a range of use cases, including:

  - System monitoring, providing insight into the health and stability of platforms where EESSI is deployed

  - Cross-system performance comparison

  - Aids in regression source identification

- Open to publish data from other systems, contact: support@eessi.io

https://dashboard.eessi.io

Richard

EESSI / TensorFlow (study "default", 01/11/2024 - 24/02/2025)
System=snellius:rome • Software stack=production • module_name=TensorFlow/2.13.0-foss-2023a • device_type=cpu

34

Richard

EESSI use cases

Richard
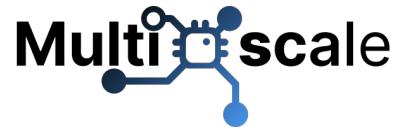
Use cases enabled by EESSI

- A **uniform software stack** across HPC clusters, clouds, laptops

- Enable **portable workflows**

- Significantly facilitates setting up infrastructure for **HPC training**

- Plans to integrate it into Open OnDemand (OoD), in collaboration with OoD developers

  - EESSI provides the scientific software, OoD provides an easy way to access it

- Can be leveraged in **continuous integration (CI)** environments

- Enhanced **collaboration with software developers** and application experts

Also discussed in our open-access paper, available via doi.org/10.1002/spe.3075

Leveraging EESSI in CI environments

- EESSI can be used in CI environments like:

- EESSI can provide:

  - Different compilers to test your software with

  - Required dependencies for your software

  - Additional tools like ReFrame, performance analysis tools, ...

- Other than CernVM-FS to get access to EESSI, no software installations required!

  - Everything that is actually needed is pulled in on-demand by CernVM-FS

- Significantly facilitates also running CI tests in other contexts (laptop, HPC, ...)

https://www.eessi.io/docs/using_eessi/eessi_in_ci

Richard

Collaboration with software developers + experts

- A central software stack by/for the community opens new doors…

- We can **work with software developers/experts** to verify the installation

  - Check how installation is configured and built

  - Help to verify whether software is functional for different use cases

  - Show us how to do extensive testing of their software

  - Evaluate performance of the software, enable performance monitoring

  - *"Approved by developers"* stamp for major applications included in EESSI

- Relieve software developers from burden of getting their software installed

  - Remove need to provide pre-built binary packages?

- Developers can also leverage EESSI themselves: dependencies, CI, …

Richard

Deploying pre-release versions of scientific software in EESSI
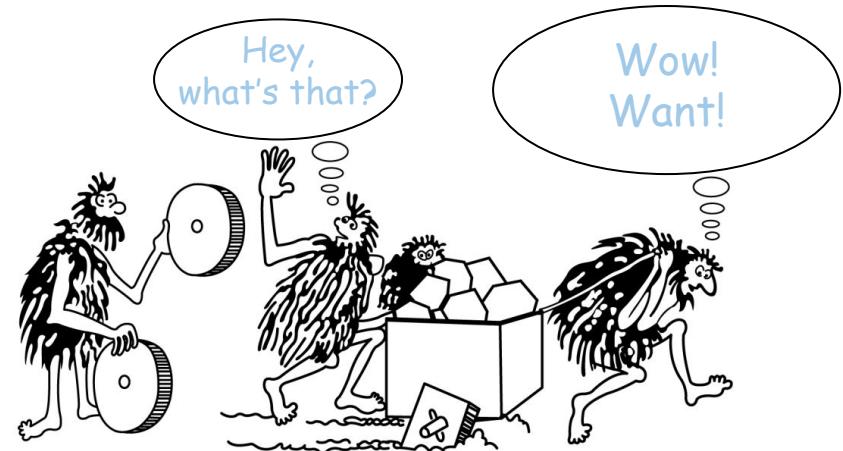
- **dev.eessi.io CernVM-FS repository**
    - Available on Vega
    - https://eessi.io/docs/repositories/dev.eessi.io
    - More information Coming soon
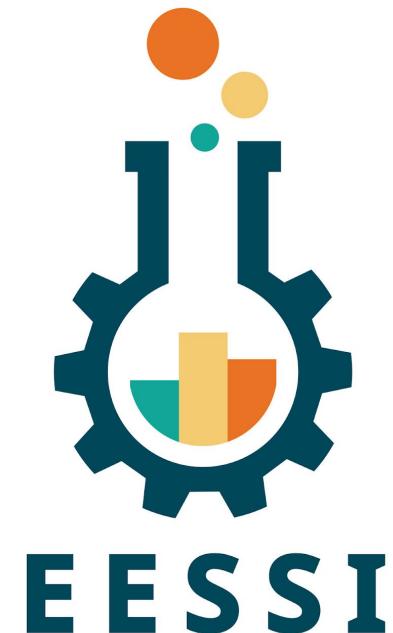
```
/cvmfs/dev.eessi.io/versions/2023.06/software
`-- linux
    `-- x86_64
        |-- amd
        |   |-- zen2
                |-- modules
                    |-- all
                        |-- ESPResSo
                        |   |-- 4.2.2-foss-2023a-2ba17de6096933275abec0550981d9122e4e5f28.lua
                        |-- GROMACS
                        |   |-- 2024.3-foss-2023b-d0f934abfd1394621c40858a2c2dd9123451df4e.lua
                        |-- LAMMPS
                        |   |-- 570c9d190fee556c62e5bd0a9c6797c4dffcc271-foss-2023a-kokkos-dev_OBMD.lua
                        |-- ...
```

EESSI in a nutshell

- **On-demand streaming** of **optimized** scientific software installations

- **Works on any Linux distribution** thanks to EESSI compat layer

- **Uniform software stack** across various systems: laptop, HPC, cloud, …

- Community-oriented: **let's tackle the challenges we see together!**

EUROPEAN ENVIRONMENT FOR
SCIENTIFIC SOFTWARE INSTALLATIONS

Richard

# EESSI won an HPCWire Reader's Choice award!
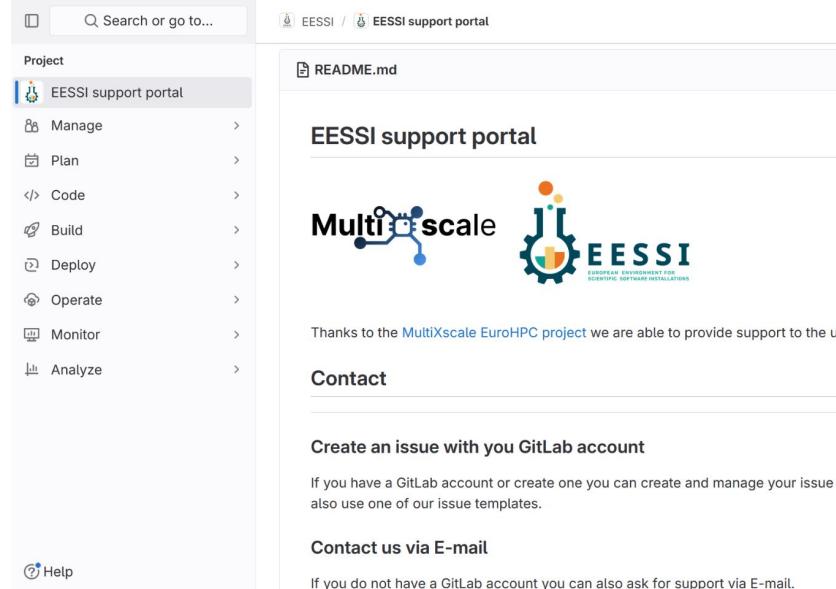
eessi.io/docs/blog/2024/11/18/hpcwire-readers-choice-awards-2024-for-eessi

Support for installing, using, contributing to EESSI

- **Via GitLab, or via email:** `support@eessi.io`

- Report problems

- Ask questions

- Request additional software

- Get help with contributing to EESSI

- Suggest enhancements, additional features, …

- Confidential tickets possible (security issues, …)

Dedicated support team, thanks to EuroHPC Centre-of-Excellence Multi scale

Richard

Tutorial *"Best Practices for CernVM-FS in HPC"*
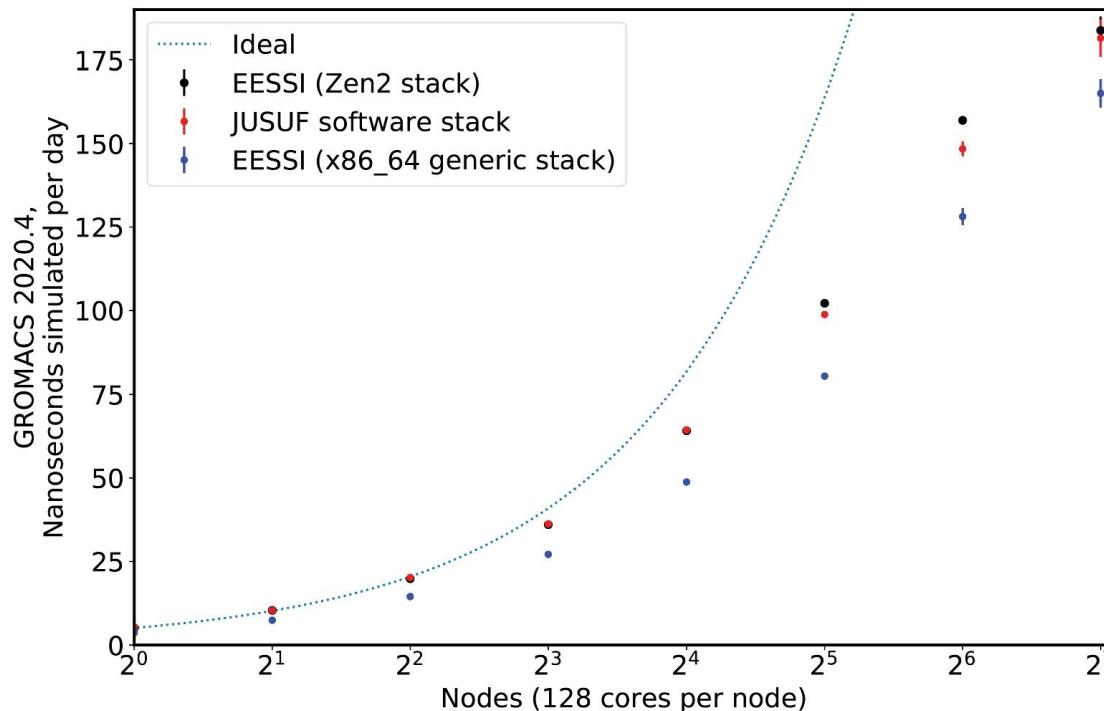
- multixscale.github.io/cvmfs-tutorial-hpc-best-practices

- Held online on 4 Dec 2023 (~3 hours), recorded & available on YouTube

- Over 200 registrations, ~125 attending the meeting

- Lecture + hands-on demos

- Topics:
  - Introduction to CernVM-FS + EESSI
  - Configuring CernVM-FS: client, Stratum 1 mirror server, proxy server
  - Troubleshooting problems
  - Benchmarking of start-up performance w/ TensorFlow

  - Shorter version of this is presented next week,
    see https://eessi.io/docs/training/2025/webinar-series-2025Q2

Richard

EESSI paper (open access, Feb'22)
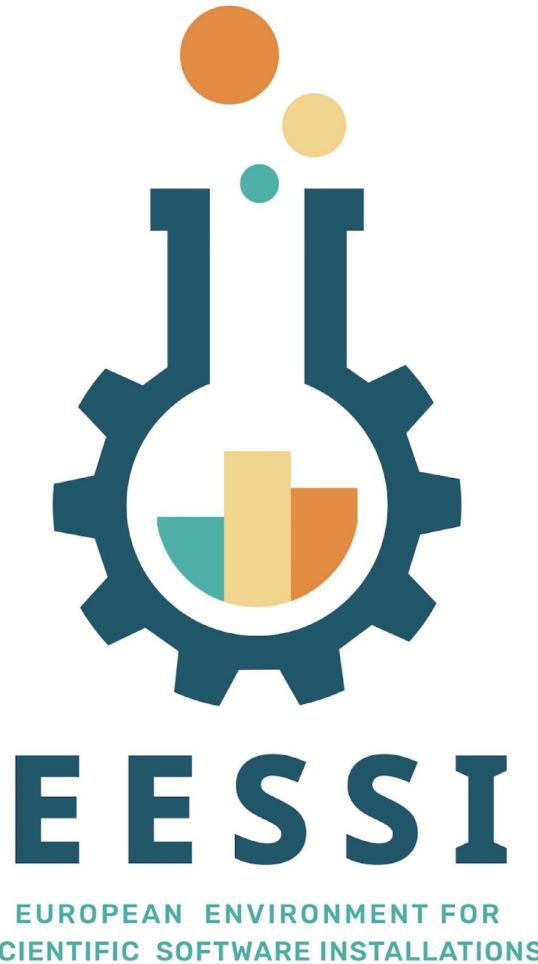
Paper includes proof-of-concept performance evaluation compared to system software stack,

performed at JUSUF @ JSC using GROMACS 2020.4, up to 16,384 cores (CPU-only)

Richard

Website: eessi.io

GitHub: github.com/eessi

Documentation: eessi.io/docs

Blog: eessi.io/docs/blog

Join the EESSI Slack

YouTube channel: youtube.com/@eessi_community

Paper (open access): doi.org/10.1002/spe.3075

EESSI support portal: gitlab.com/eessi/support

Bi-monthly online meetings (1st Thu, odd months, 2pm CE(S)T)

Richard

Web page: **multixscale.eu**

Facebook: **MultiXscale**

Twitter: **@MultiXscale**

LinkedIn: **MultiXscale**

BlueSky: **MultiXscale**

Webinar series: Different aspects of EESSI

5 Mondays in a row May-June 2025

https://eessi.io/docs/training/2025/webinar-series-2025Q2

- **Introduction to EESSI webinar/tutorial** (today)

- Introduction to CernVM-FS (12 May)

- Introduction to EasyBuild (19 May)

- EESSI for CI/CD (26 May)

- Using EESSI as the base for a system stack (2 June)

More info and registration →