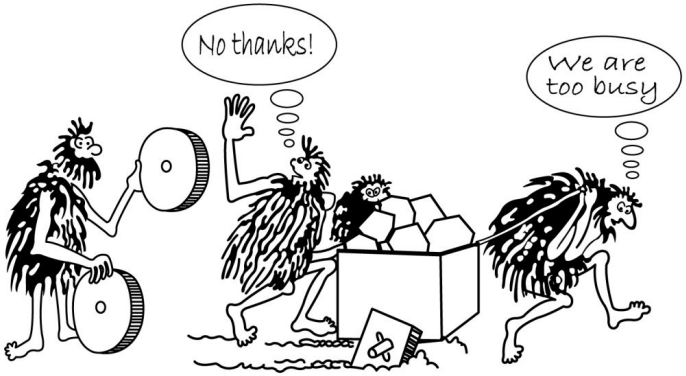
From zero to science in minutes

EESSI

EUROPEAN ENVIRONMENT FOR SCIENTIFIC SOFTWARE INSTALLATIONS

Multiscale

26th Geilo Winter School
Tue 20 Jan 2026



Thomas Röblitz - IT division - University of Bergen

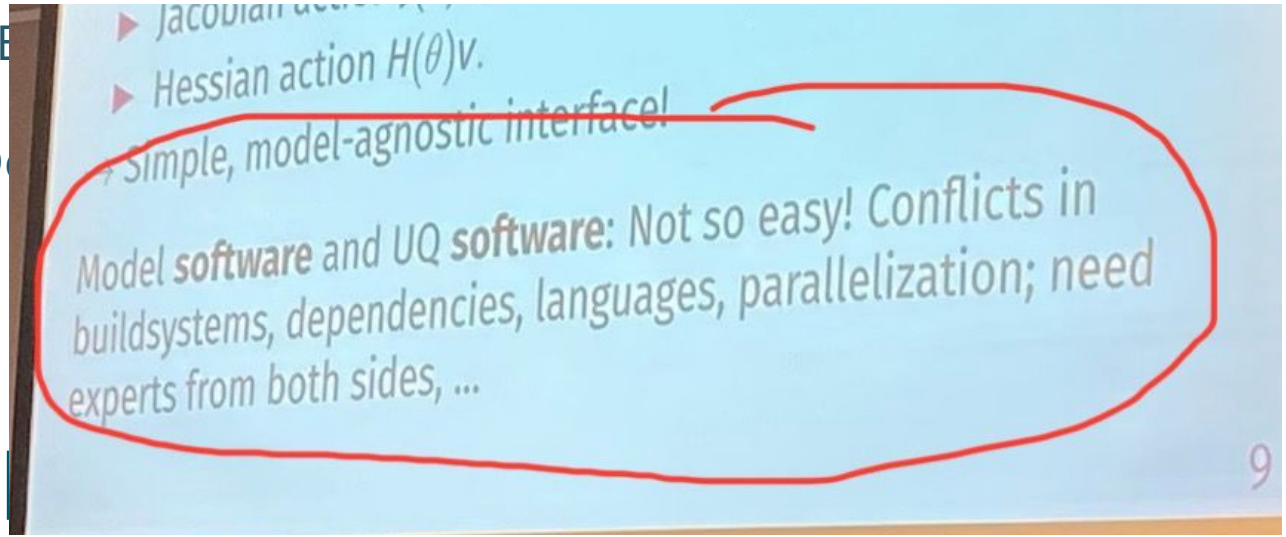
thomas.roblitz@uib.no

Goals for this lecture

- European Environment for Scientific Software Installations (EESSI)
- What it is - Why did we start it - Why should you care
- EESSI's design - How we build and test the EESSI software - Use cases
- Demos - Exercises

Goals for this lecture

- European Environment for Scientific Software Installations (EESSI)
- What it is - Why did we start it - Why should you care
- EE Use cases
- D



EUROPEAN ENVIRONMENT FOR SCIENTIFIC SOFTWARE INSTALLATIONS

Goals for this lecture

- European Environment for Scientific Software Installations (EESSI)
- What it is - Why did we start it - Why should you care
- EESSI's design - How we build the EESSI software - Use cases
- Demos - Exercises

EESSI in a nutshell

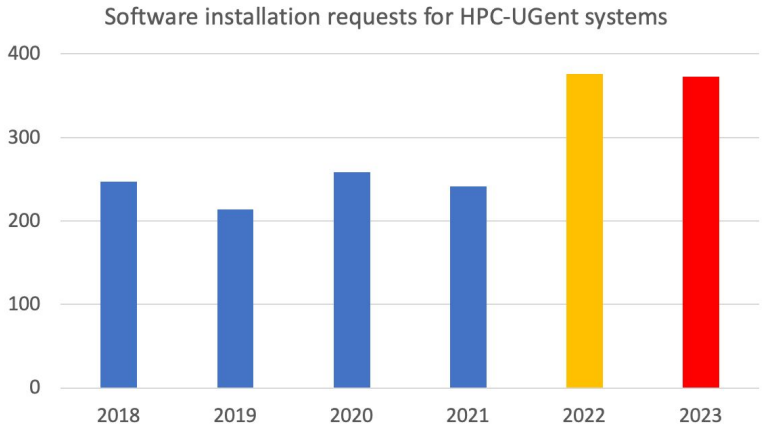
- **Shared repository of (optimized!) scientific software installations**
- Uniform way of providing software to users, regardless of the system they use!
- Should work on any Linux OS (+ WSL, macOS via Lima) and system architecture
- From laptops and personal workstations to HPC clusters and cloud
- Support for different CPU (micro)architectures, interconnects, GPUs, etc.
- **Focus on performance, automation, testing, collaboration**

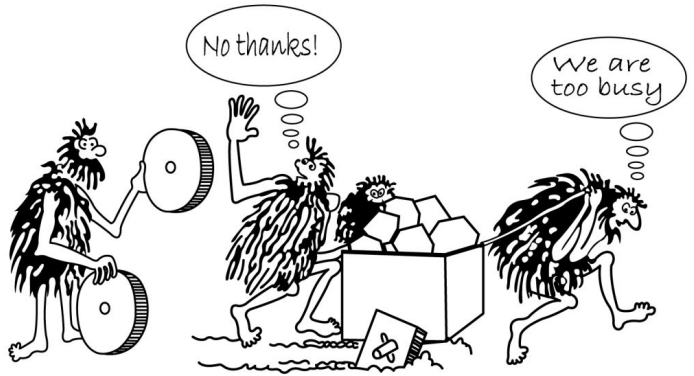
https://eessi.io

https://eessi.io/docs

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**…

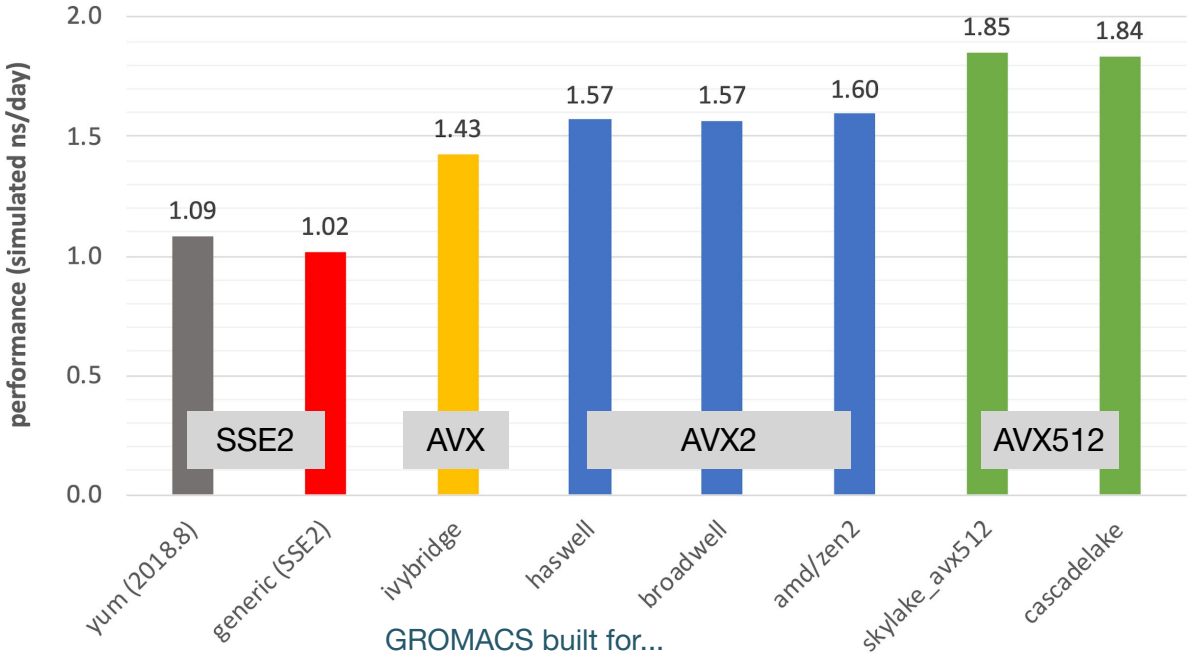




Optimized scientific software installations

- 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!

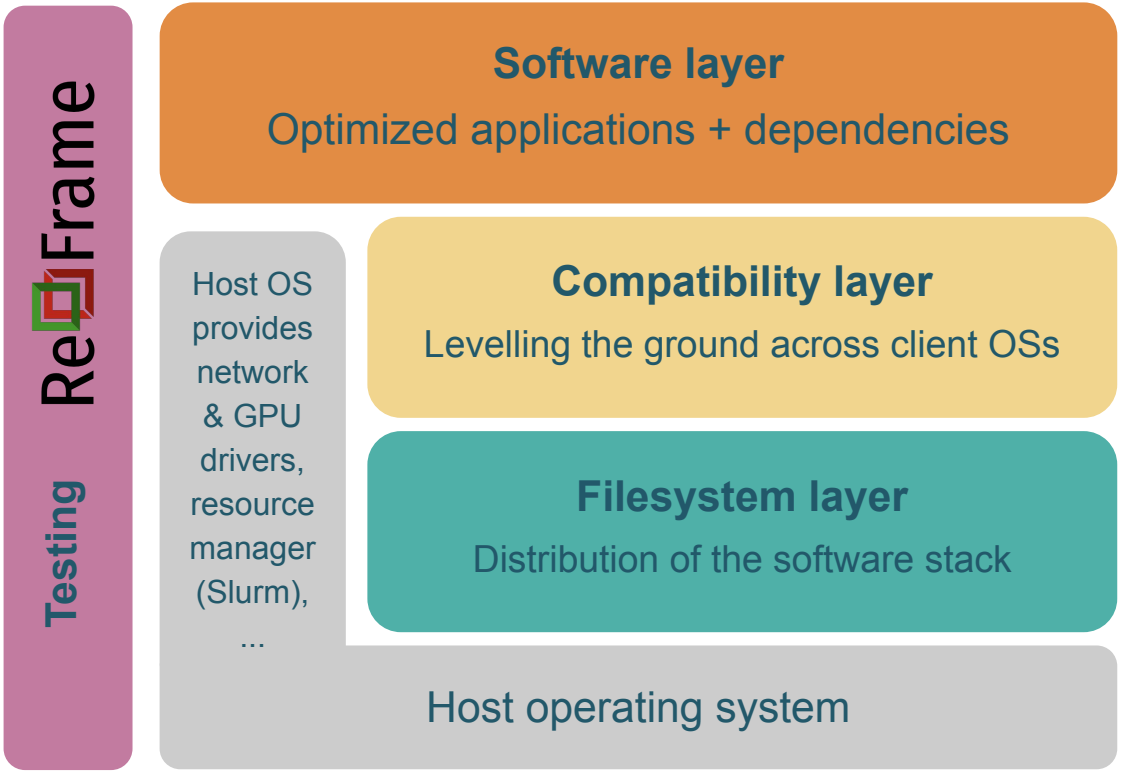
- Example: GROMACS 2020.1 (PRACE benchmark, Test Case B)
- Metric: (simulated) ns/day, higher is better
- Test system: dual-socket Intel Xeon Gold 6420 (Cascade Lake, 2x18 cores)
- **Performance of different GROMACS binaries, on exact same hardware/OS**



Major goals of EESSI

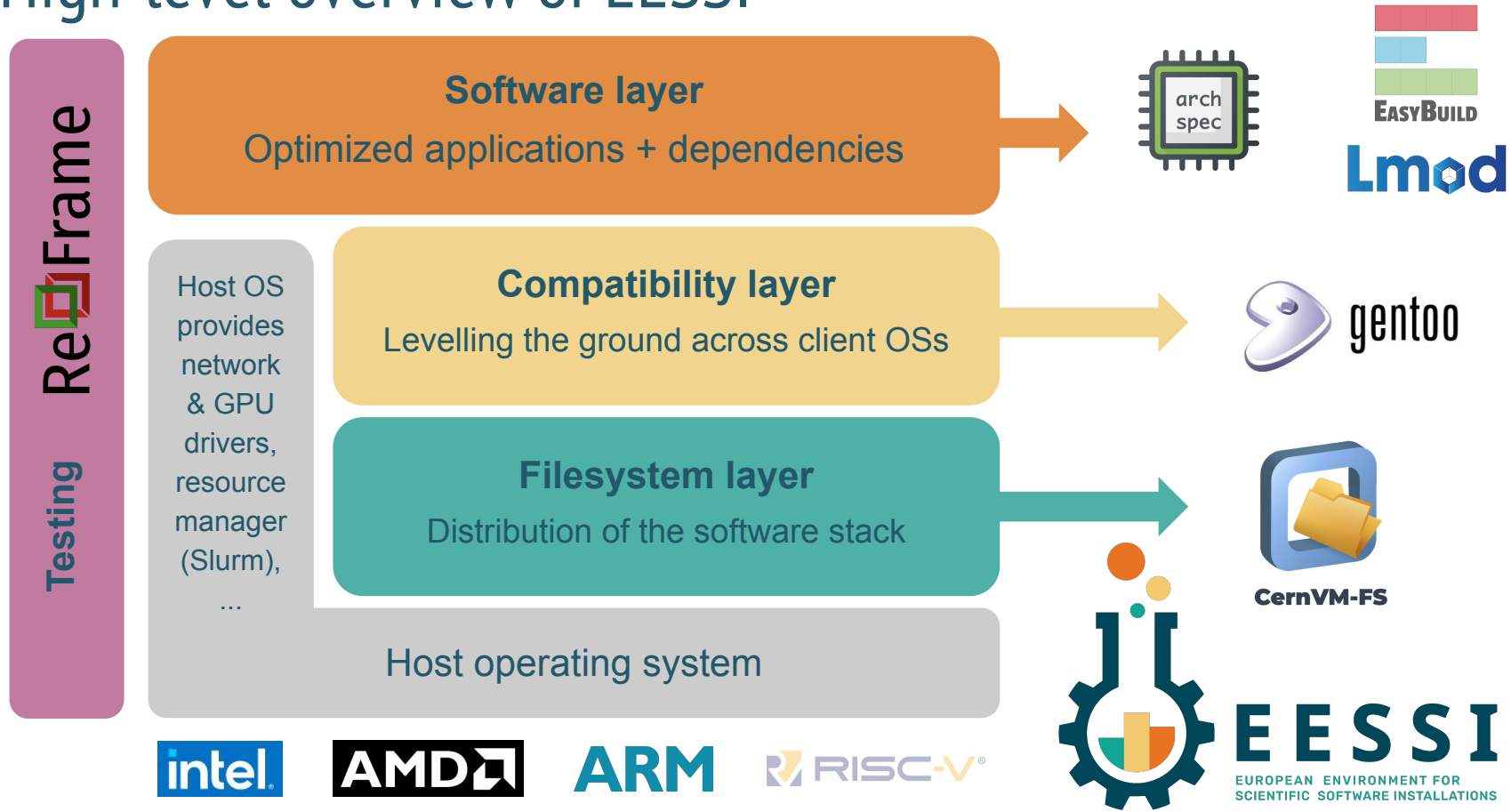
- **Avoid duplicate work** (for researchers, HPC support teams, sysadmins, …)
  - Tools that automate software installation process (EasyBuild, Spack) are not sufficient anymore
  - Go beyond sharing build recipes => work towards a shared software stack
- Providing a truly **uniform software stack**
  - Use the (exact) same software environment everywhere
  - **Without sacrificing performance** for “mobility of compute” (like is typically done with containers/conda)
- Facilitate HPC training, development of (scientific) software, …

High-level overview of EESSI

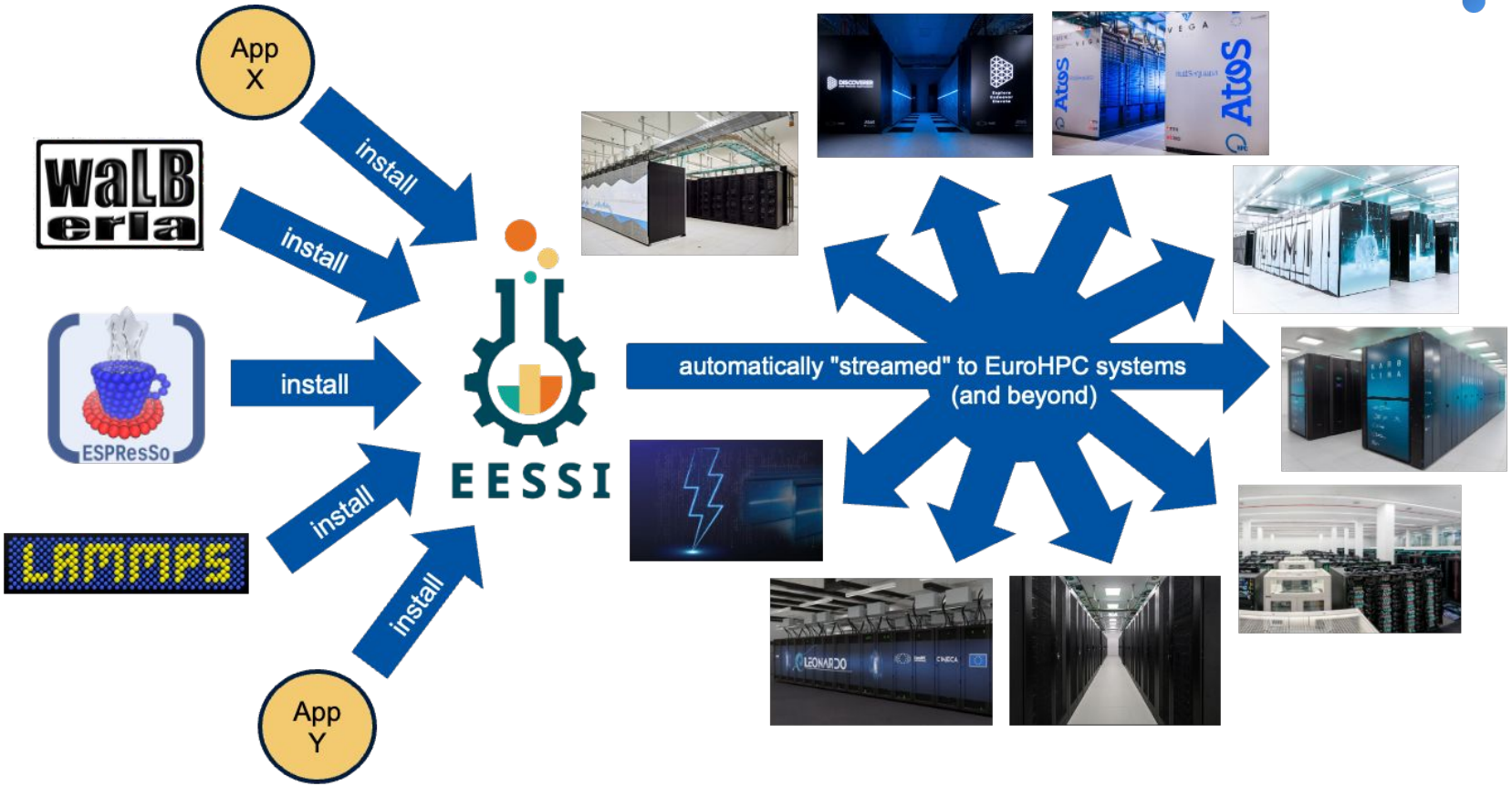


EESSI
EUROPEAN ENVIRONMENT FOR SCIENTIFIC SOFTWARE INSTALLATIONS

High-level overview of EESSI



EESSI as a shared software stack



Hands-on live demo

Getting access to EESSI

Using EESSI

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/

Demo: Using EESSI

eessi.io/docs/using_eessi/eessi_demos

s

```
/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
```

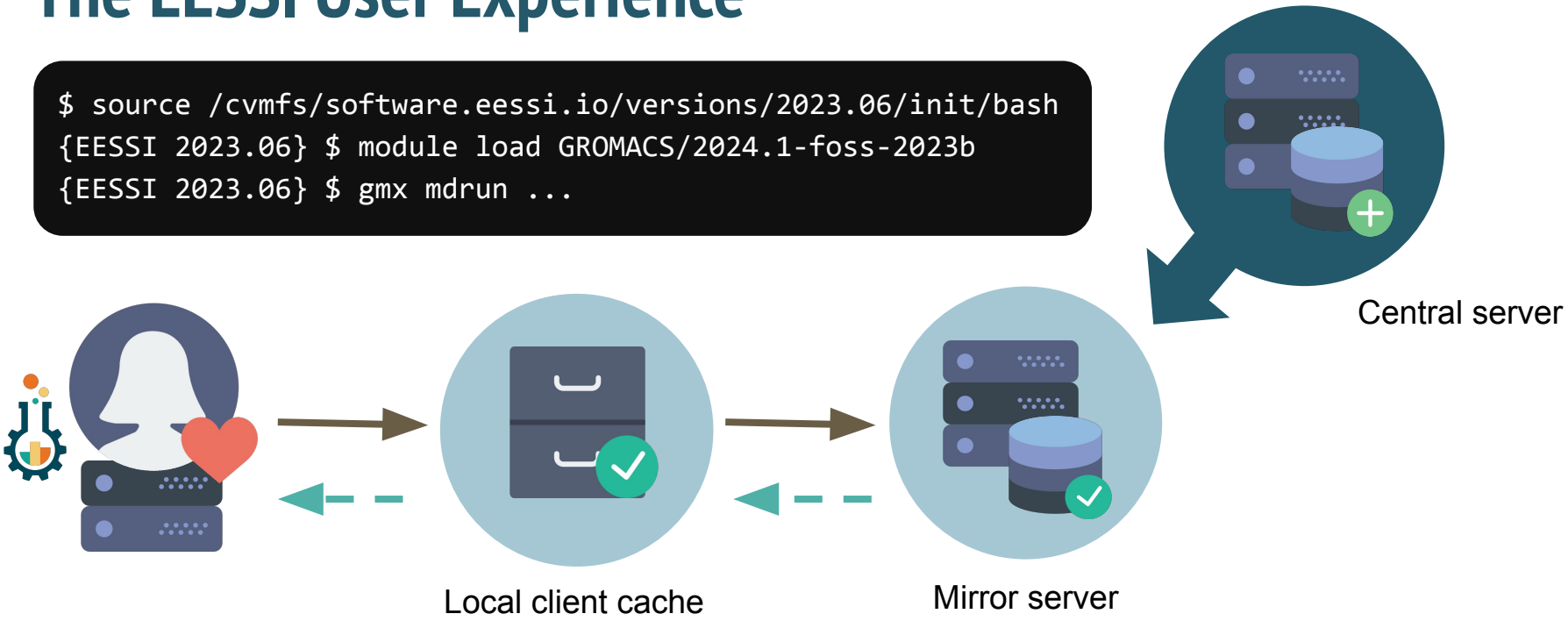
```
$ source /cvmfs/software.eessi.io/versions/2023.06/init/bash
Found EESSI pilot repo @
/cvmfs/software.eessi.io/versions/2023.06!
archdetect says x86_64/amd/zen3
Using x86_64/amd/zen3 as software subdirectory
...
Environment set up to use EESSI pilot software stack, have fun!

{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
```

Demo: Running LAMMPS in a Slurm job script

```
#!/usr/bin/bash
#SBATCH --account=nn14000k
#SBATCH --job-name="EESSI_Demo_LAMMPS_lj"
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=4
#SBATCH --mem=16G
#SBATCH --time=00:05:00
#SBATCH --partition=normal

export CURL_CA_BUNDLE=/etc/ssl/ca-bundle.pem  # work around EESSI issue on Olivia
curl -o in.lj https://raw.githubusercontent.com/lammps/lammps/refs/heads/develop/bench/in.lj

module load EESSI/2023.06
module load LAMMPS/29Aug2024-foss-2023b-kokkos

export OMP_NUM_THREADS=1
mpirun -np 4 lmp -in in.lj
```

The EESSI User Experience

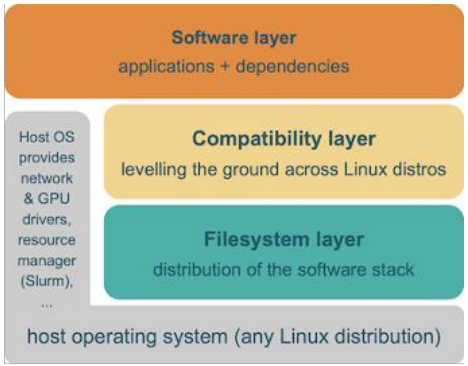
```
$ source /cvmfs/software.eessi.io/versions/2023.06/init/bash
{EESSI 2023.06} $ module load GROMACS/2024.1-foss-2023b
{EESSI 2023.06} $ gmx mdrun ...
```



EESSI provides **on-demand streaming**

of (scientific) software (like music, TV-series, …)

How does EESSI work?

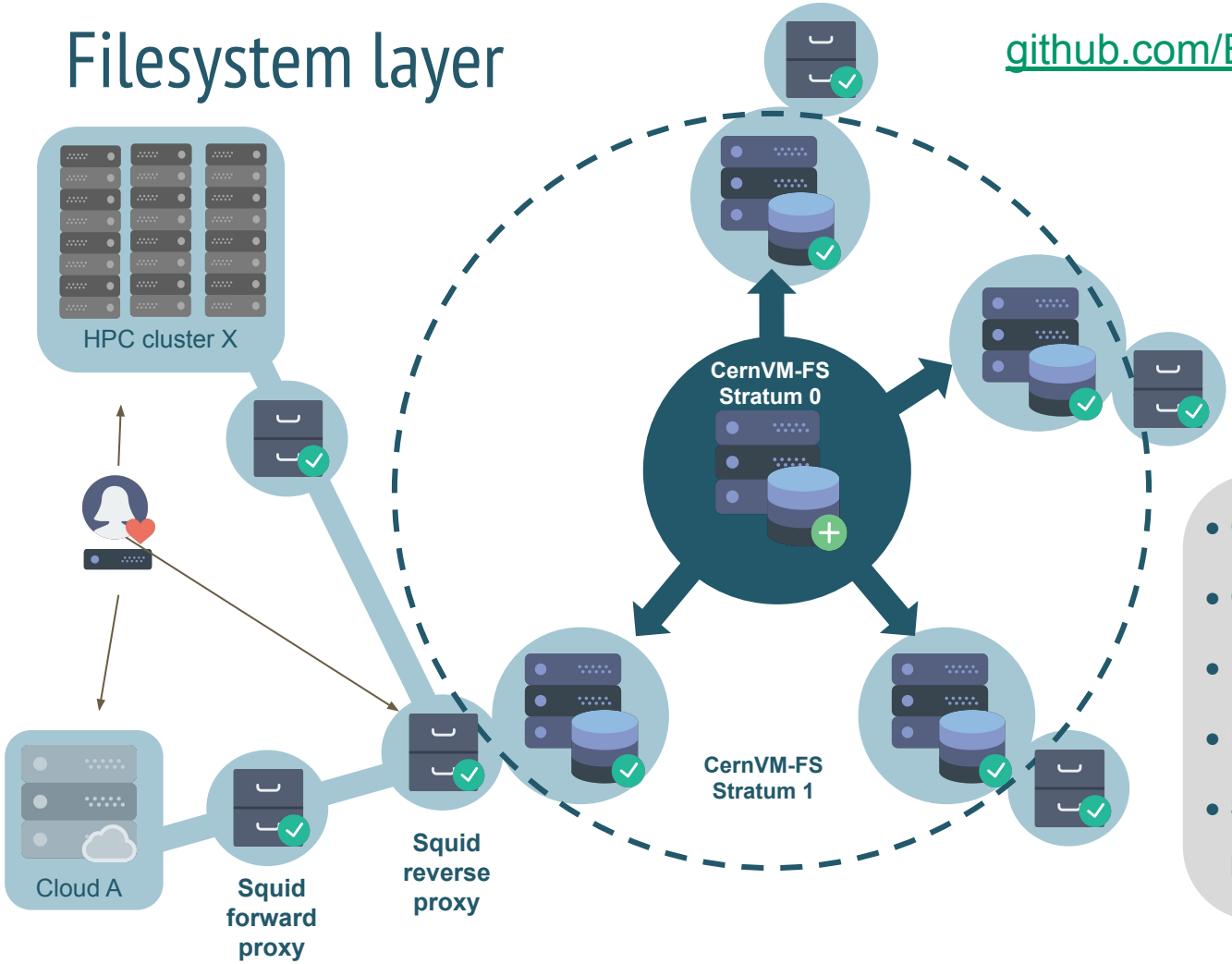


- Software installations included in EESSI are:
  - Automatically **“streamed in” on demand** (via CernVM-FS)
  - Built to be **independent of the host operating system**

    “Containers without the containing”
  - **Optimized** for specific CPU generations + specific GPU types
- Initialization script **auto-detects** CPU + GPU of the system

Filesystem layer

github.com/EESSI/filesystem-layer



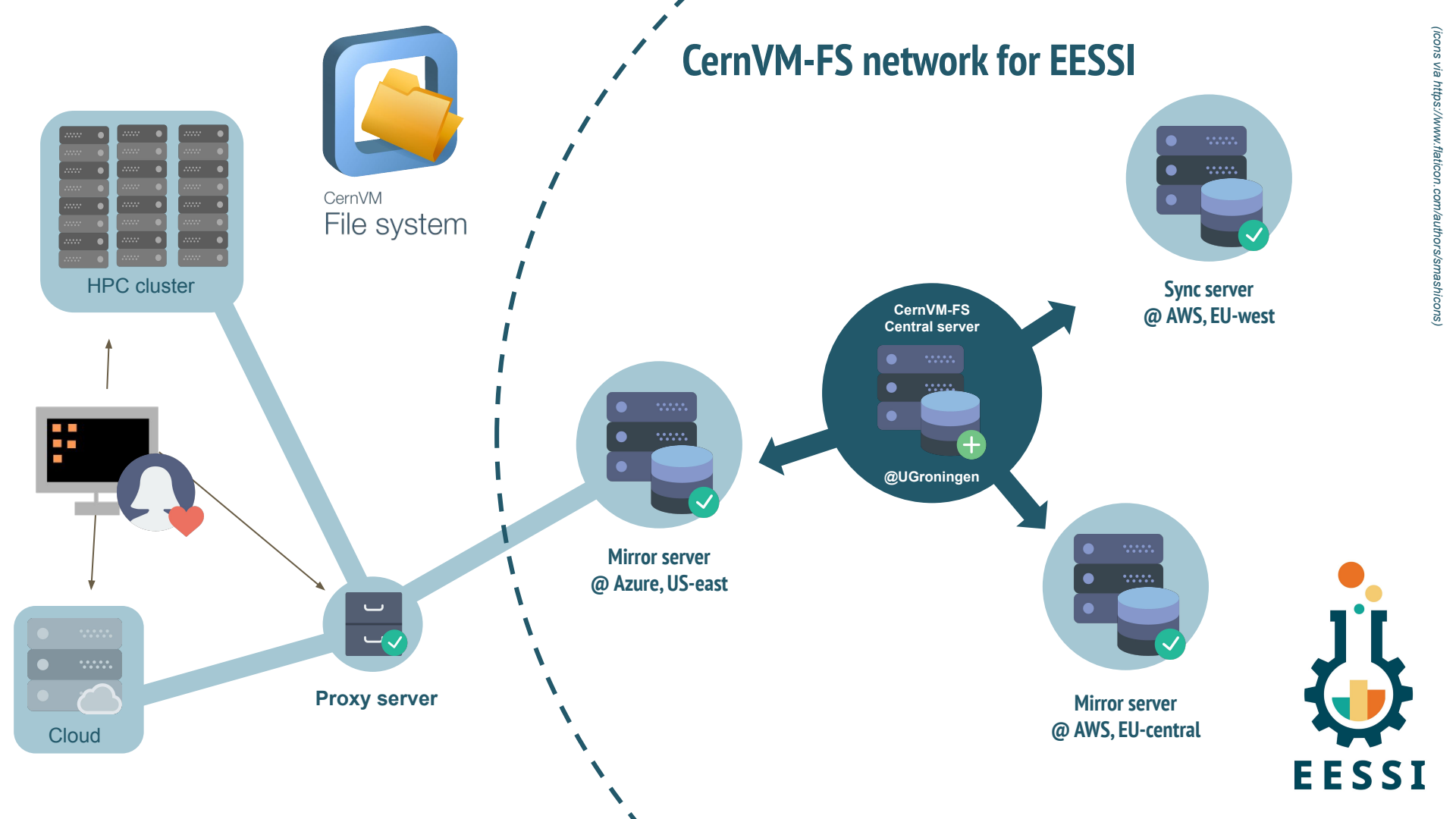
CernVM-FS

cvmfs.readthedocs.io

- 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, …

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

CernVM-FS network for EESSI



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

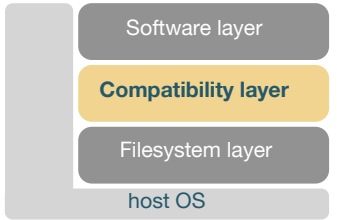
Compatibility layer

github.com/EESSI/compatibility-layer

EESSI

- 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

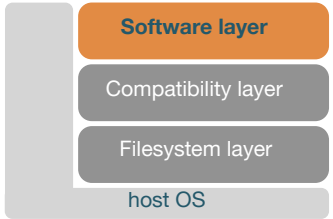


Software layer

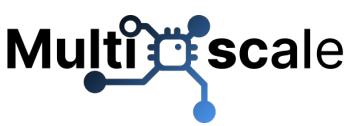
github.com/EESSI/software-layer

- 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

powered by



Current status of EESSI

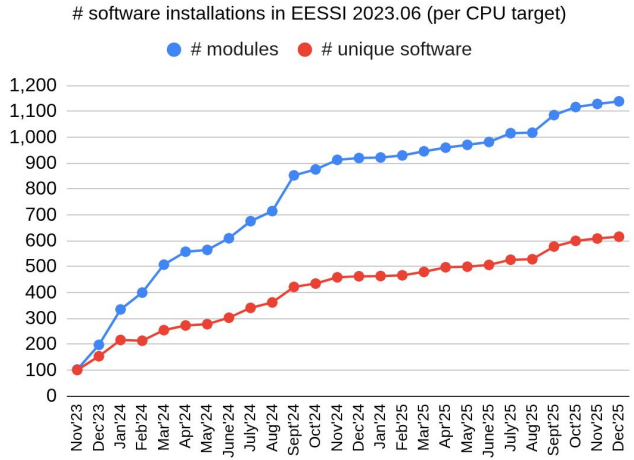


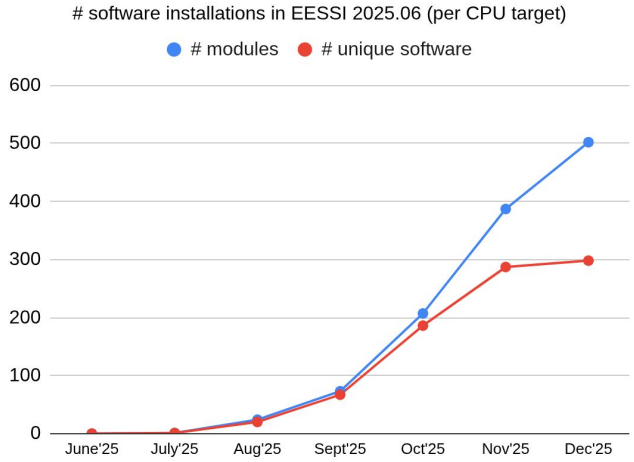
- Production CernVM-FS repository **`software.eessi.io`** available since Nov'23
- Ansible playbooks, scripts, docs available at https://github.com/eessi
- CernVM-FS: Stratum 0 @ Univ. of Groningen + two Stratum 1 servers in AWS + Azure
- **Supported by Azure and AWS**: sponsored credits to develop necessary infrastructure
- Additional build infrastructure at several HPC sites

Overview of available software

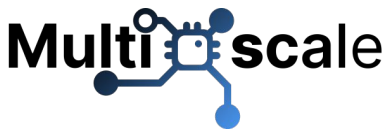
Currently ~1140/500 software installations available per CPU target via software.eessi.io CernVM-FS repository; increasing every day

- Over 600/300 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
- avg 1GB/day increase on central server



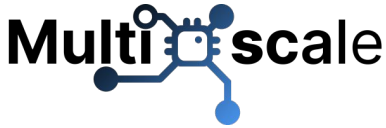


Supported system architectures



- Different generations of `x86_64` (Intel, AMD) and Arm 64-bit CPUs; RISC-V is WIP
  - Including A64FX (Deucalion) & NVIDIA Grace (JUPITER)
  - 14 CPU architectures: http://www.eessi.io/docs/software_layer/cpu_targets
  - Also works on laptops, in virtual machines in the cloud, on Raspberry Pi boards, etc.
- Different accelerators: **NVIDIA GPUs** (today) + **AMD GPUs** (soon)
  - For all* CPU architectures three CUDA compute capabilities supported: 7.0, 8.0 and 9.0
    - (*) except for A64FX
- **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**

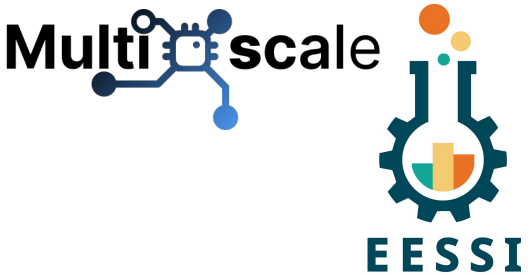
On which systems is EESSI already available?



- EuroHPC JU systems:
  - Native installation (via CernVM-FS) on **Vega** + **Karolina + Deucalion + Discoverer**
  - Semi native installation (via rsync) on **MareNostrum5**
  - Native installation on **MeluXina**, **LUMI, Leonardo, JUPITER** is a work-in-progress
- 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**

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…

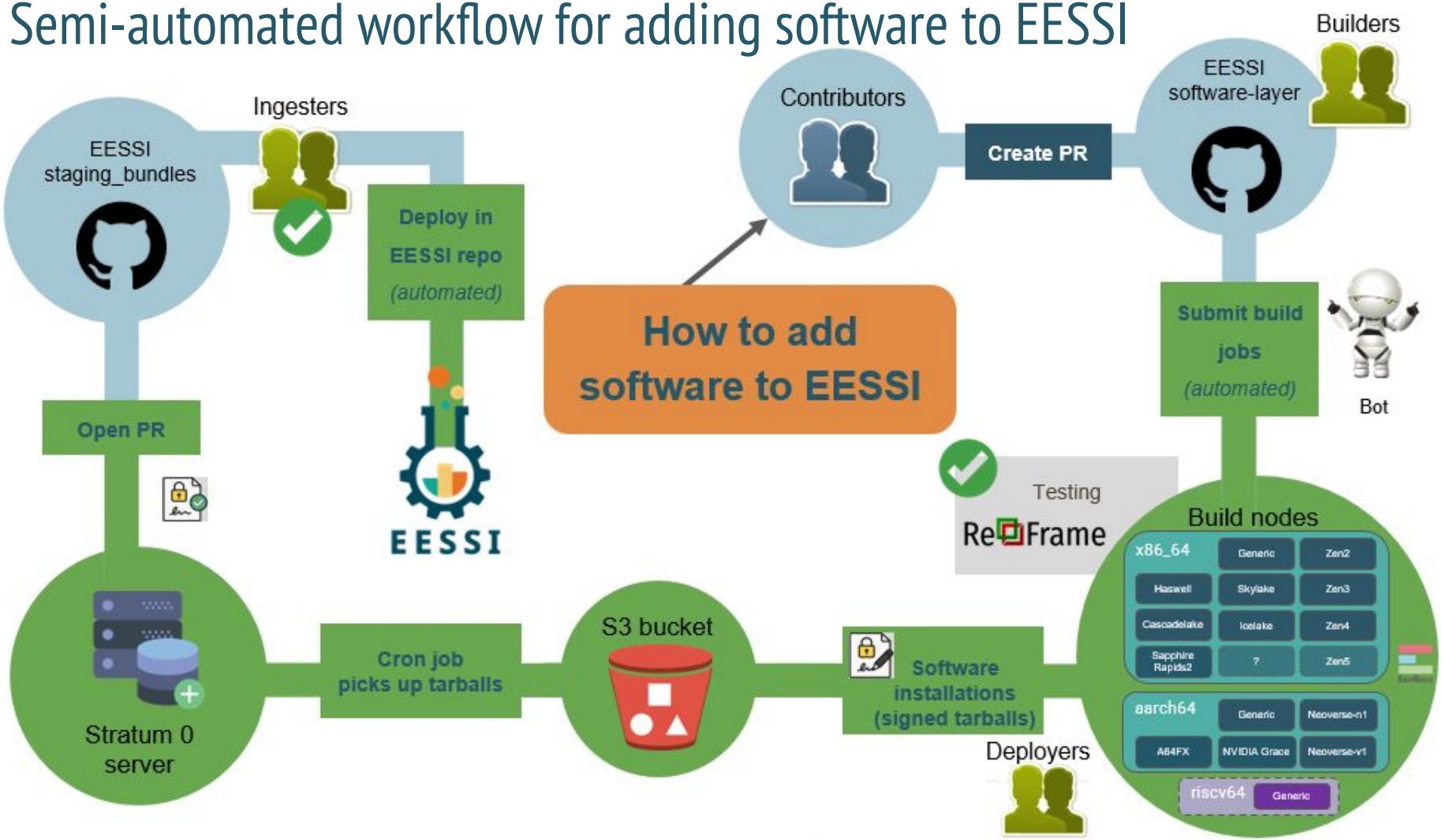
- In EESSI, we provide scripts to deal with both these problems:

    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_and_libraries.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, ESPResSo, LAMMPS, NCCL, OSU Micro-Benchmarks, GROMACS
- More CPU/GPU combos and software (PyTorch, TensorFlow, AlphaFold) coming soon…

Semi-automated workflow for adding software to EESSI

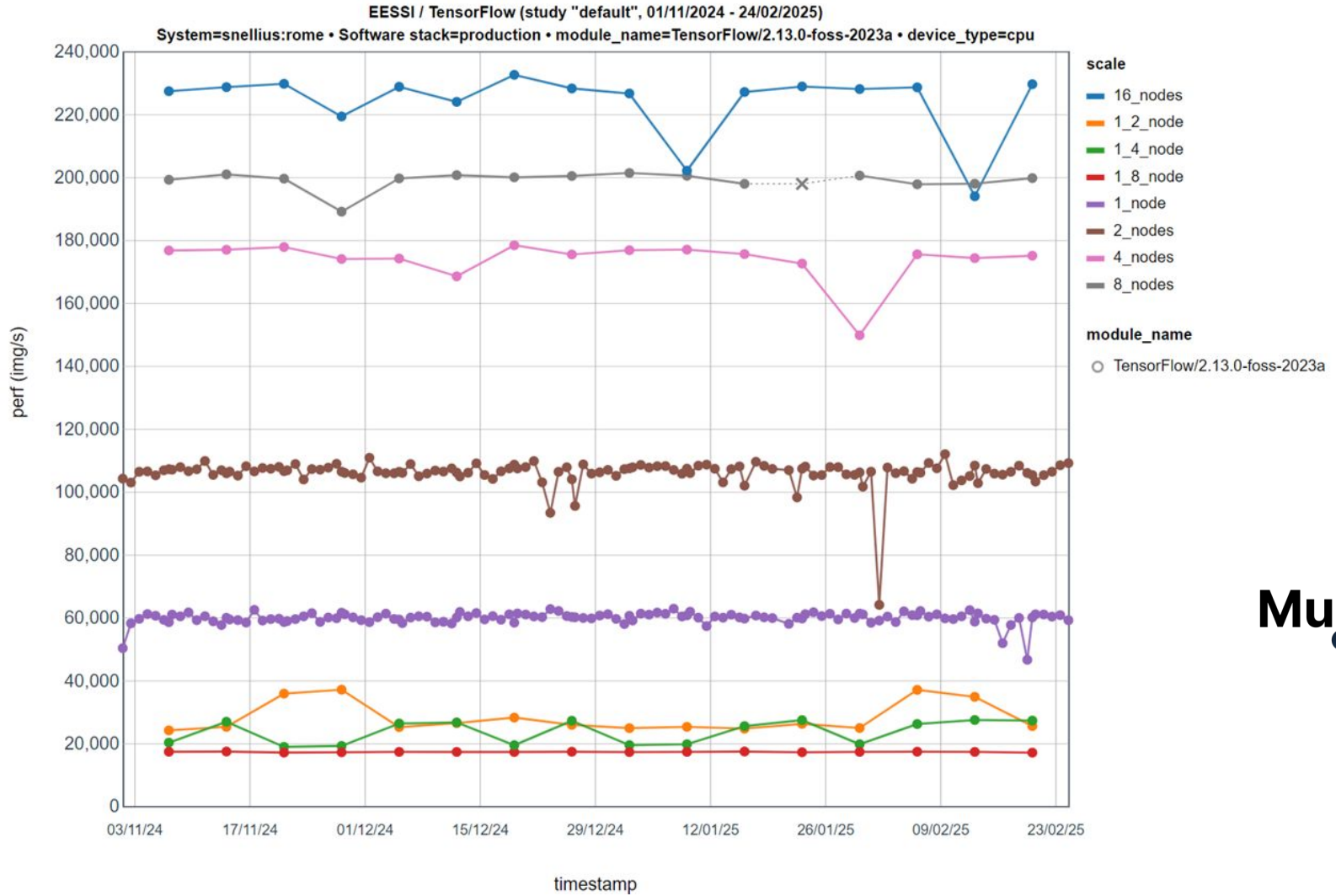


Software testing is an important part of EESSI

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

Software testing is an important part of EESSI

- EESSI test suite: eessi.io/docs/test-suite
  - Collection of *portable* tests for software available in EESSI
  - Periodically (daily/weekly) on about 6 different systems
  - Running of selected (single node) tests when building new Software for EESSI (before deployment)
  - Can also be used for other software stacks than the EESSI software stack
  - based on ReFrame



EESSI

Multi scale

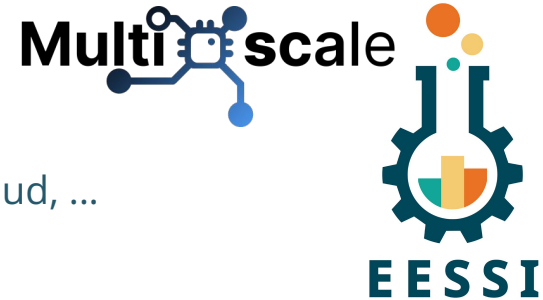
EESSI use cases

Use cases enabled by EESSI

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

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

EESSI as a uniform software stack

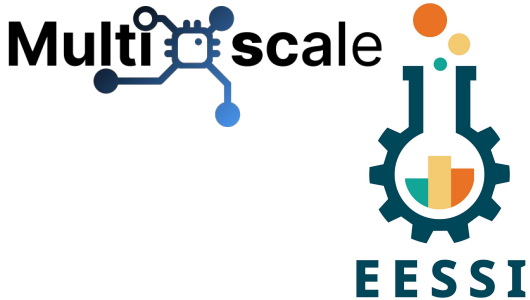


- Main goal: **same software everywhere**: laptop, server, HPC, cloud, …
- Wide variety of systems supported
  - CPUs: `x86_64` (Intel, AMD), `aarch64` (Arm), `riscv64` (work-in-progress…)
  - OS: any Linux distribution, on Windows via WSL, on macOS via Lima
  - High-speed interconnects (Infiniband), GPUs, etc.
- **Without compromising on software performance**
  - Optimized software installations for specific CPU microarchitectures + auto-detection
  - Large contrast with generic binaries often used in containers
- **Facilitates migrating** from laptop to HPC, cloud bursting, …

EESSI enables portable workflows

- Portable workflows are significantly easier when relying on EESSI
- They often involve running a broad set of tools, which all need to be available
- Workflows definitions (Snakemake, Nextflow,...) can leverage (or be included in) EESSI
- You can ship your execution environment inside your git repository using `direnv`
- If your users have EESSI and `direnv`, then can start running your workflow after cloning!

Leveraging EESSI in CI environments



- EESSI can be used in CI environments like:
  - GitHub: github.com/marketplace/actions/eessi
  - GitLab: gitlab.com/explore/catalog/eessi/gitlab-eessi
- 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, …)

Leveraging EESSI in CI environments

We have an EESSI GitHub Action that provides EESSI+`direnv`:

See it in action in the `github-eessi-action` repository:

github.com/EESSI/github-action-eessi

github.com/EESSI/github-action-eessi/blob/main/.github/workflows/tensorflow-usage.yml

```yaml
name: ubuntu_tensorflow
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: |
       module load TensorFlow
       python -c 'import tensorflow; print(tensorflow.__version__)'
```

Leveraging EESSI in CI environments

We have an EESSI GitHub Action that provides EESSI:

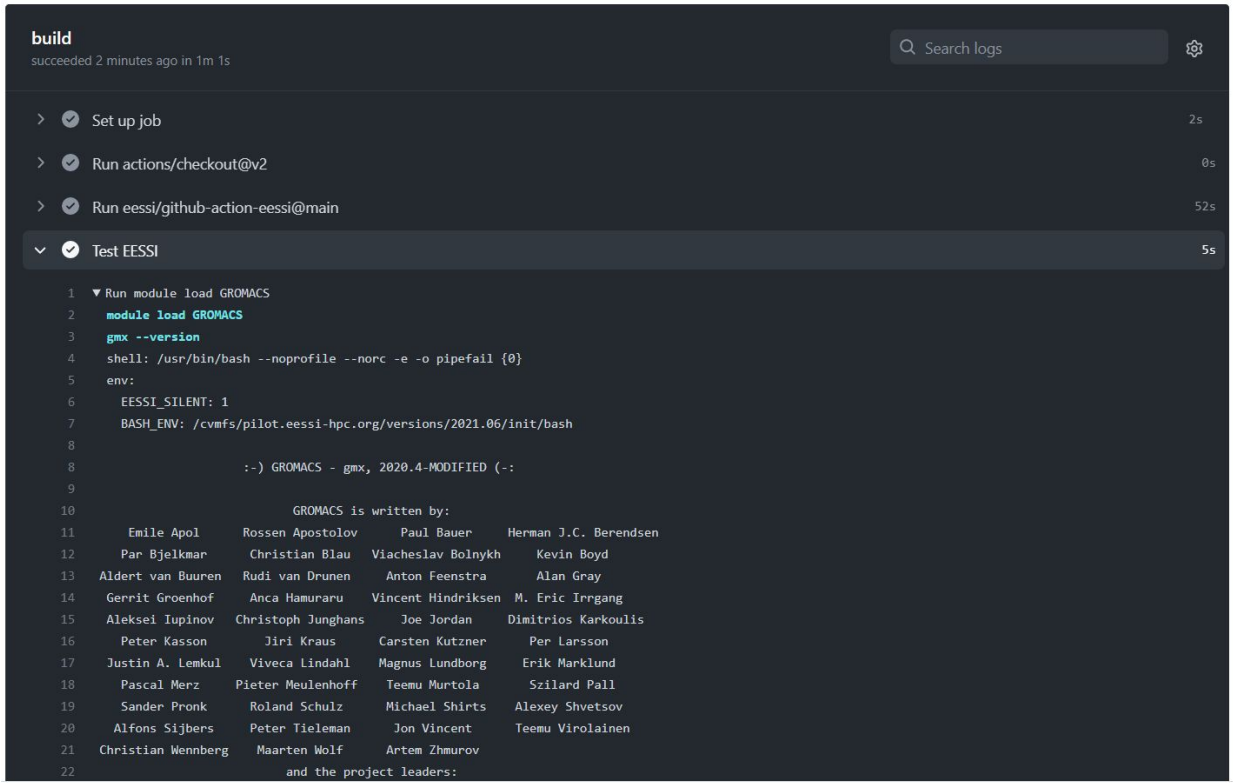
See it in action in the `github-eessi-action` repository:

github.com/EESSI/github-action-eessi

github.com/EESSI/github-action-eessi/blob/main/.github/workflows/tensorflow-usage.yml

```yaml
name: ubuntu_tensorflow
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: |
       module load TensorFlow
       python -c 'import tensorflow; print(tensorflow.__version__)'
```

Leveraging EESSI GitHub Action



https://github.com/EESSI/github-action-eessi/actions/runs/11183032689/job/31090668500

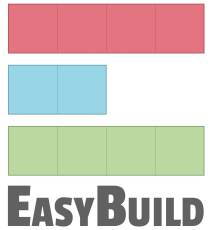
Facilitate HPC training

- EESSI can significantly reduce effort required to set up infrastructure for HPC training sessions (introductory, software-specific, …)
- Setting up a throwaway Slurm cluster in the cloud is easy via **Magic Castle**
  - Simple process once you have cloud credits:
    - Create cluster by editing a single file
    - Automatically configured within 20 minutes
    - Includes support for GPU and fast interconnects (Infiniband, EFA)
  - EESSI project uses Magic Castle for some of the build-and-deploy “bots”
  - There are also commercial alternatives that can/do support EESSI (Azure/AWS)
- **EESSI can provide (scientific) software that is required for the training**
- Attendees can easily set up the same software environment later on their own system(s) by leveraging EESSI

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, …

Webinar series: Different aspects of EESSI



5 Mondays in a row May-June 2025 (likely repeated in 2026)

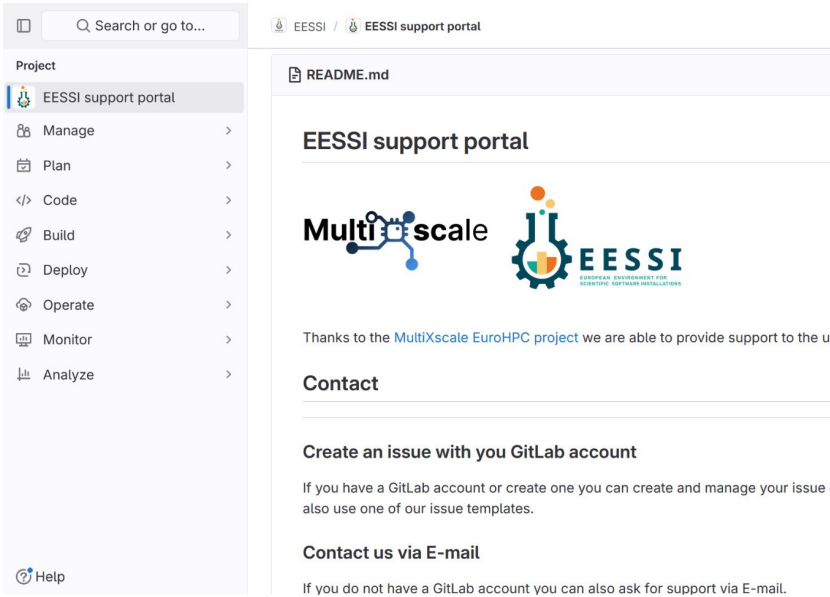
- **Introduction to EESSI**
- **Introduction to CernVM-FS**
- **Introduction to EasyBuild**
- **EESSI for CI/CD**
- **Using EESSI as the base for a system stack**
- https://www.eessi.io/docs/training-events/2025/webinar-series-2025Q2/

Support for installing, using, contributing to EESSI

GitLab

eessi.io/docs/support

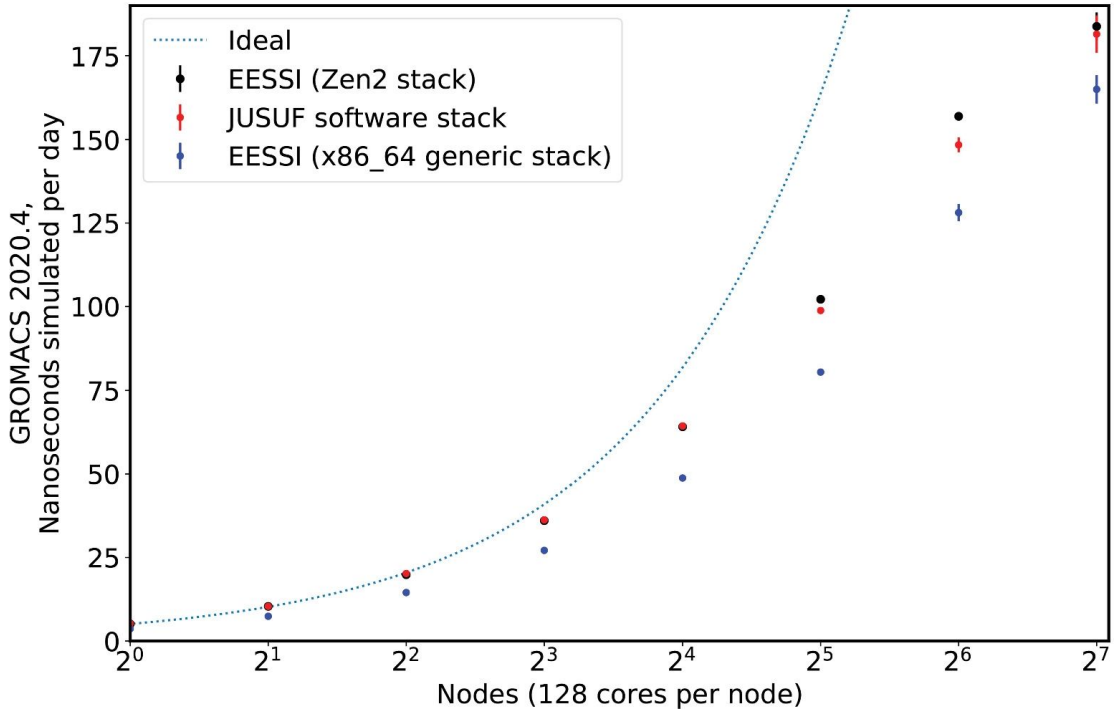
- Via GitLab, or via email: `support@eessi.io`
- Report problems
- Ask questions
- Request additional software
- Get help with contributing to EESSI
- Suggest enhancements, additional features, ...
- Confidential tickets possible (security issues, ...)



Dedicated support team, thanks to EuroHPC Centre-of-Excellence Multiscale

EESSI paper (open access, Feb'22)

doi.org/10.1002/spe.3075



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)

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)

EESSI Happy Hour (every Monday, 2pm CE(S)T, drop-in)

MultiXscale

Web page: multixscale.eu

LinkedIn: MultiXscale

Co-funded by the European Union

EuroHPC Joint Undertaking

KEMIJSKI INŠTITUT

UNIVERSITAT DE BARCELONA

Universität Stuttgart

UNIVERSITAS BERGENSIS

SORBONNE UNIVERSITÉ

Université de Toulouse

Consiglio Nazionale delle Ricerche

MAX-PLANCK-GESELLSCHAFT

HPC Now!

university of groningen

SURF

JÜLICH Forschungszentrum

GHENT UNIVERSITY

BSC Barcelona Supercomputing Center Centro Nacional de Supercomputación

LEONARDO

iit ISTITUTO ITALIANO DI TECNOLOGIA

Funded by the European Union. This work has received funding from the European High Performance Computing Joint Undertaking (JU) and countries participating in the project under grant agreement No 101093169.

Summary

- EESSI provides a shared stack of software installations
- Several access mechanisms: CernVM-FS recommended -> "streaming" on demand
- Enables various use cases: CI, portable workflows, ...
- Support available through EuroHPC CoE MultiXscale
- Contributions (add "my" software) are welcome!
- Community project
- htttps://eessi.io - https://eessi.io/docs

Exercises (30 minutes)

Pick one or two

1. Just try EESSI on Olivia (10 minutes)
2. Run an EESSI-demo on Olivia (10 minutes)
3. Use EESSI in GitHub Actions (30 minutes)
4. Install & configure CernVM-FS to get access to EESSI on a machine (10 minutes)

Detailed information available at (link available via #03-röblitz channel on Slack)

Bonus-demo: Run EESSI GROMACS demo on RISC-V.