Keeping the P in HPC

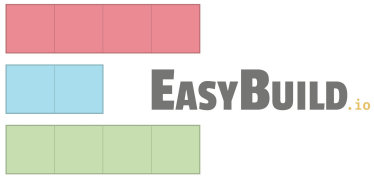
Kenneth Hoste (Ghent University) - `kenneth.hoste@ugent.be`

1 February 2026

FOSDEM'26 | Software Performance devroom

whoami

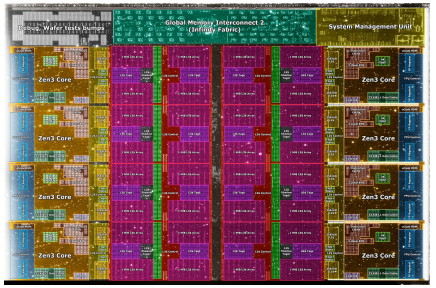
- HPC sysadmin @ Ghent University (Belgium) since 2010
- Open source software enthusiast for ~20 years
- I also like my wife & kids, (loud) gigs, beer (but I'm picky), stickers, dad jokes, ...
- FOSDEM attendee since 2012, (co-)organising HPC devroom since 2014
- Lead developer of EasyBuild, core contributor to EESSI, …
- Socials:
  - GitHub: `@boegel`
  - BlueSky: `@boegel.bsky.social`
  - Fediverse/Mastodon: `@boegel@mast.hpc.social`
  - LinkedIn



HPC?

- **H**igh **P**erformance **C**omputing, a.k.a. supercomputing
- Running demanding (scientific) workloads on **supercomputers**
- **Parallel** computing, distributed memory parallelism (typically using MPI)
- Supercomputers are really just a bunch of (beefy, expensive) **servers** which each have large number of **CPU cores** (easily > 100) + maybe some **GPUs**, and are connected to each other with a (fast, expensive) **interconnect** (network), and all have simultaneous access to a lot of (fast, expensive, large) shared **storage**
- These systems are **complex**, using them and/or programming for them is not easy…

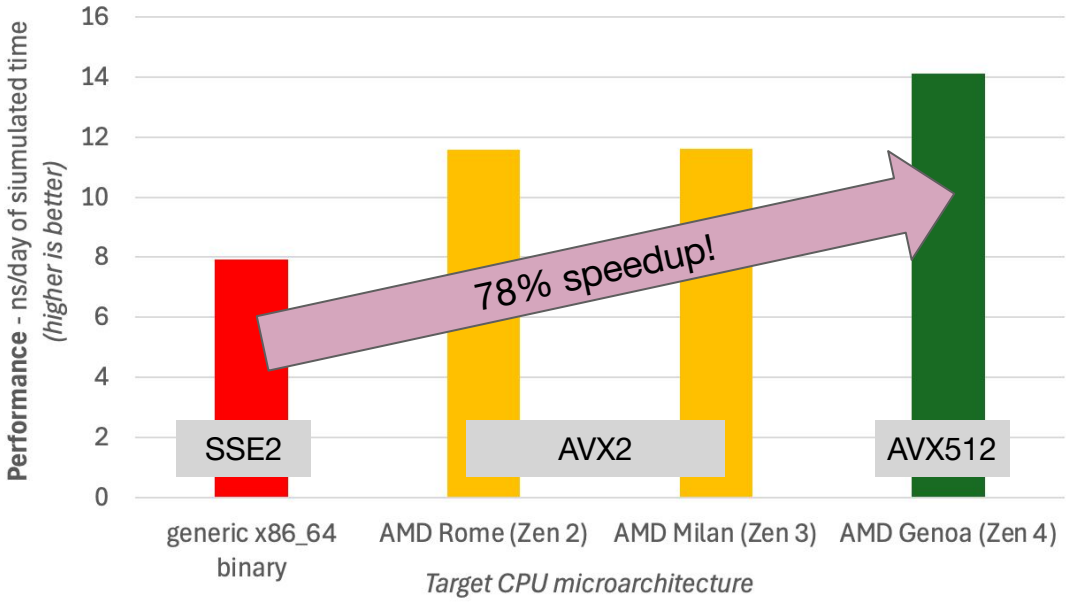
Modern CPUs



- Modern microprocessors support **vector instructions**
- Like AVX2, AVX-512, AVX10 (AMD, Intel), or SVE (Arm), …
- Parts of the microprocessor are *dedicated* to running these vector instructions
- If you run binaries that are not using these instructions,
  you're not using a significant part of what the CPU supports
- … and **performance suffers** (potentially a lot)
- Downside: binaries that do use vector instructions are less "portable"…

Keeping the P in HPC

- Software should be **optimized** for (the CPU/GPU of) the system it will be run on
- This implies **building software from source code**, using specific compiler options, …
- Impact on **software performance** is often *significant* for scientific software!



- Barplot shows performance of different GROMACS binaries,
  same source code & workload,
  same hardware & OS
- GROMACS version 2025.2

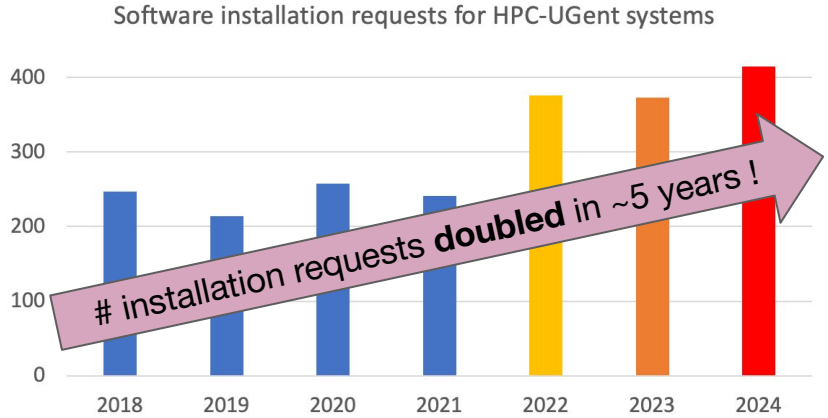
  Test Case B from PRACE UEABS v2.2
- 2x 96-core AMD EPYC 7532 (Zen 4)
- Hybrid run on 192 cores in total

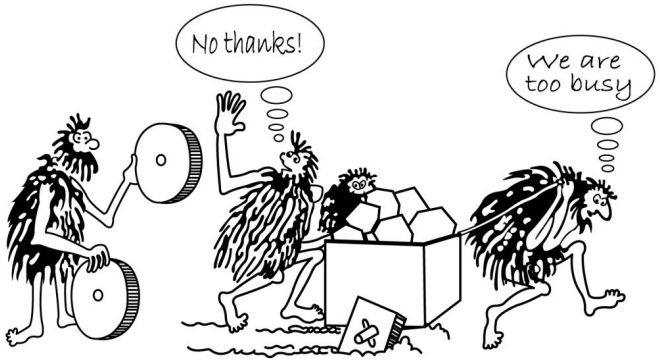
  48 MPI ranks with 4 OpenMP threads each

The landscape of scientific computing is changing

- **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~~ here (see Fugaku, JUPITER, …), RISC-V is coming (soon?)
- In strong contrast: available (wo)manpower **in HPC support teams is (still) limited**…





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?

European Environment for Scientific Software Installations

- **Shared repository of (optimized!) scientific software *installations***
- Avoid duplicate work across by collaborating on a shared software stack
- Uniform way of providing software to users, regardless of the system they use!
- Should work on any Linux OS 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**
- Development effort funded through **MultiXscale** EuroHPC Centre-of-Excellence

EESSI
EUROPEAN ENVIRONMENT FOR SCIENTIFIC SOFTWARE INSTALLATIONS

https://eessi.io

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, …

EESSI as shared software stack for HPC, cloud, and beyond

https://eessi.io

EasyBuild
Lmod

Optimized installations for
over 650 software projects
(plus >1,000 Python, R, ... extensions)

EESSI as shared software stack for HPC, cloud, and beyond

https://eessi.io

EasyBuild
Lmod

Optimized installations for **over 650 software projects**
(plus >1,000 Python, R, ... extensions)

Works on **any Linux system**
(laptop, cloud, HPC, ...)

EESSI as shared software stack for HPC, cloud, and beyond

https://eessi.io

EasyBuild
Lmod

Optimized installations for **over 650 software projects**
(plus >1,000 Python, R, ... extensions)

Works on **any Linux system**
(laptop, cloud, HPC, ...)

intel AMD ARM RISC-V NVIDIA

14 supported CPU targets
AMD, Intel, Arm
+ 3 generations of NVIDIA GPUs
(RISC-V + AMD GPUs are WIP)

EESSI as shared software stack for HPC, cloud, and beyond

https://eessi.io

EasyBuild
Lmod

Optimized installations for **over 650 software projects**
(plus >1,000 Python, R, ... extensions)

Works on **any Linux system**
(laptop, cloud, HPC, ...)

intel AMD ARM RISC-V NVIDIA

14 supported CPU targets
AMD, Intel, Arm
+ 3 generations of NVIDIA GPUs
(RISC-V + AMD GPUs are WIP)

Software is **streamed in on-demand**

CernVM-FS

EESSI as shared software stack for HPC, cloud, and beyond

EESSI

https://eessi.io

EasyBuild
Lmod

Optimized installations for **over 650 software projects**
(plus >1,000 Python, R, ... extensions)

Works on **any Linux system**
(laptop, cloud, HPC, ...)

intel AMD ARM RISC-V NVIDIA

14 supported CPU targets
AMD, Intel, Arm
+ 3 generations of NVIDIA GPUs
(RISC-V + AMD GPUs are WIP)

Software is **streamed in on-demand**
CernVM-FS

EuroHPC Joint Undertaking

Base of Federated Software Stack in **EuroHPC Federation Platform**

EESSI is powered by FOSS

EESSI

EUROPEAN ENVIRONMENT FOR
SCIENTIFIC SOFTWARE INSTALLATIONS

Software Layer

Optimized software installations for specific CPU microarchitectures

Intuitive user interface:
module avail,
module load, …

Lmod

Automatic selection of best suited part of software stack for CPU microarchitectures

EESSI is powered by FOSS

gentoo linux™

Compatibility layer

Abstraction from the host operating system

EESSI

EUROPEAN ENVIRONMENT FOR SCIENTIFIC SOFTWARE INSTALLATIONS

Software Layer

EasyBuild.io
building software with ease

Optimized software installations for specific CPU microarchitectures

Intuitive user interface:
module avail,
module load, …

Lmod

Automatic selection of best suited part of software stack for CPU microarchitectures

EESSI is powered by FOSS

EESSI

EUROPEAN ENVIRONMENT FOR SCIENTIFIC SOFTWARE INSTALLATIONS

Compatibility layer

gentoo linux™

Abstraction from the host operating system

Filesystem Layer

CernVM-FS

Global distribution of software installations (on-demand streaming)

Software Layer

EasyBuild.io
building software with ease

Optimized software installations for specific CPU microarchitectures

Intuitive user interface:
module avail,
module load, …

Lmod

arch detect

Automatic selection of best suited part of software stack for CPU microarchitectures

EESSI is powered by FOSS

gentoo linux™

Compatibility layer

Abstraction from the host operating system

CernVM-FS

Filesystem Layer

Global distribution of software installations (on-demand streaming)

ReFrame

Regression testing of software

EESSI

EUROPEAN ENVIRONMENT FOR SCIENTIFIC SOFTWARE INSTALLATIONS

Software Layer

EasyBuild.io
building software with ease

Optimized software installations for specific CPU microarchitectures

Intuitive user interface: module avail, module load, …

Lmod

arch detect

Automatic selection of best suited part of software stack for CPU microarchitectures

EESSI is powered by FOSS

EESSI

EUROPEAN ENVIRONMENT FOR SCIENTIFIC SOFTWARE INSTALLATIONS

Compatibility layer

gentoo linux™

Abstraction from the host operating system

Filesystem Layer

CernVM-FS

Global distribution of software installations (on-demand streaming)

Software Layer

EasyBuild.io
building software with ease

Optimized software installations for specific CPU microarchitectures

Intuitive user interface:
module avail,
module load, …

Lmod

arch detect

Automatic selection of best suited part of software stack for CPU microarchitectures

ReFrame

Regression testing of software

MAGIC CASTLE

Magic Castle to create (ephemeral) clusters in the cloud

EESSI as shared software stack for HPC, cloud, and beyond

https://eessi.io

EESSI as shared software stack for HPC, cloud, and beyond

GROMACS
TensorFlow
python
OpenFOAM®
Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS
QUANTUMESPRESSO
PyTorch
WRF
OpenCV
PETSc

EESSI

Laptops, PCs

VLAAMS SUPERCOMPUTER CENTRUM

Regional/national systems

https://eessi.io

EESSI as shared software stack for HPC, cloud and beyond

GROMACS
TensorFlow
python
OpenFOAM®
Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS
QUANTUMESPRESSO
PyTorch
WRF
OpenCV
PETSc

EESSI

Laptops, PCs

Regional/national systems

VLAAMS SUPERCOMPUTER CENTRUM

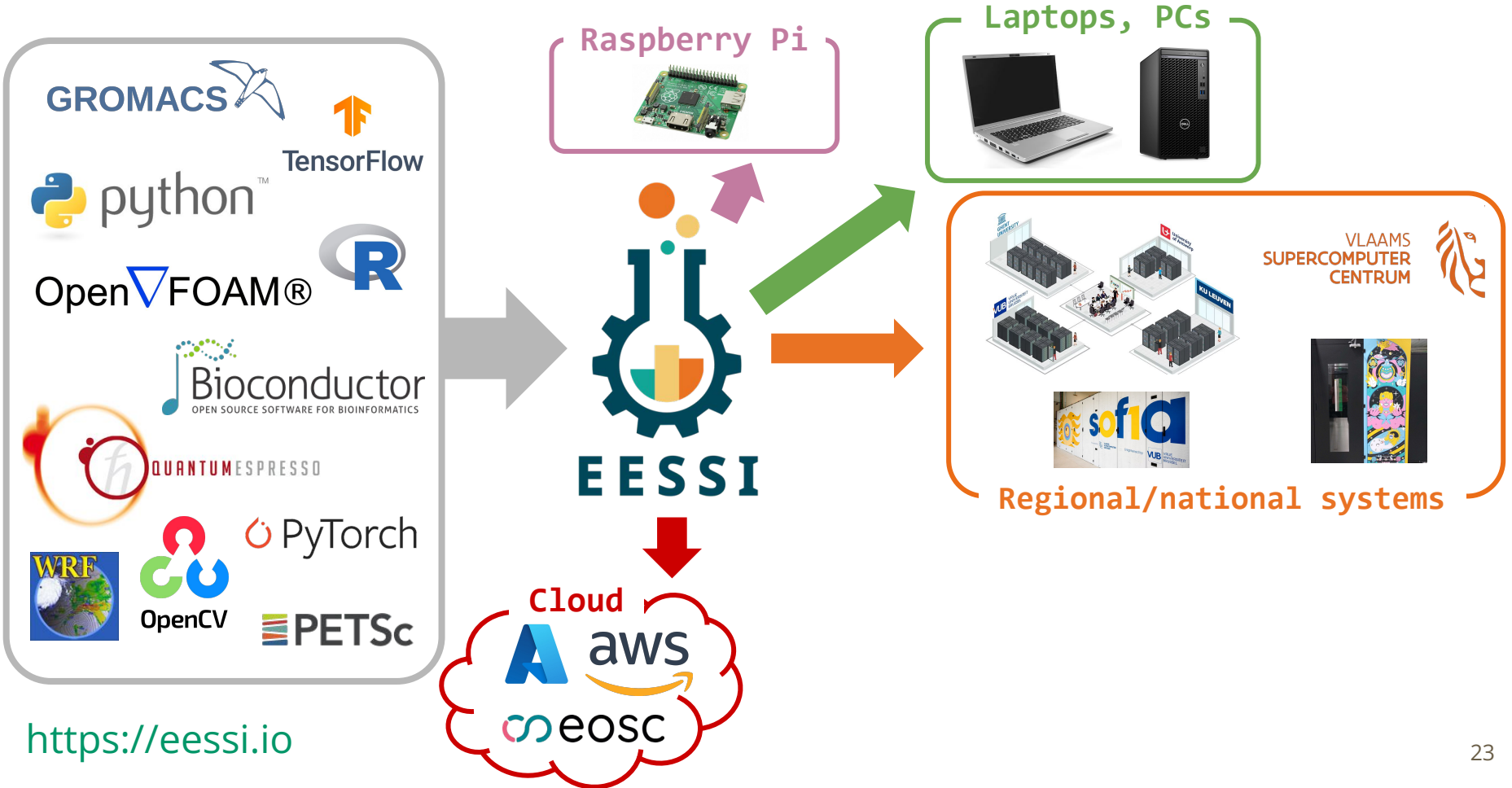
Cloud

aws

eosc

https://eessi.io

EESSI as shared software stack for HPC, cloud and beyond



https://eessi.io

EESSI as shared software stack for HPC, cloud and beyond

GROMACS
TensorFlow
python
OpenFOAM®
Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS
QUANTUMESPRESSO
PyTorch
WRF
OpenCV
PETSc

https://eessi.io

EESSI

Raspberry Pi

Laptops, PCs

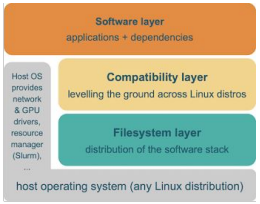
VLAAMS SUPERCOMPUTER CENTRUM

Regional/national systems

Cloud
aws
eosc

EuroHPC
Joint Undertaking

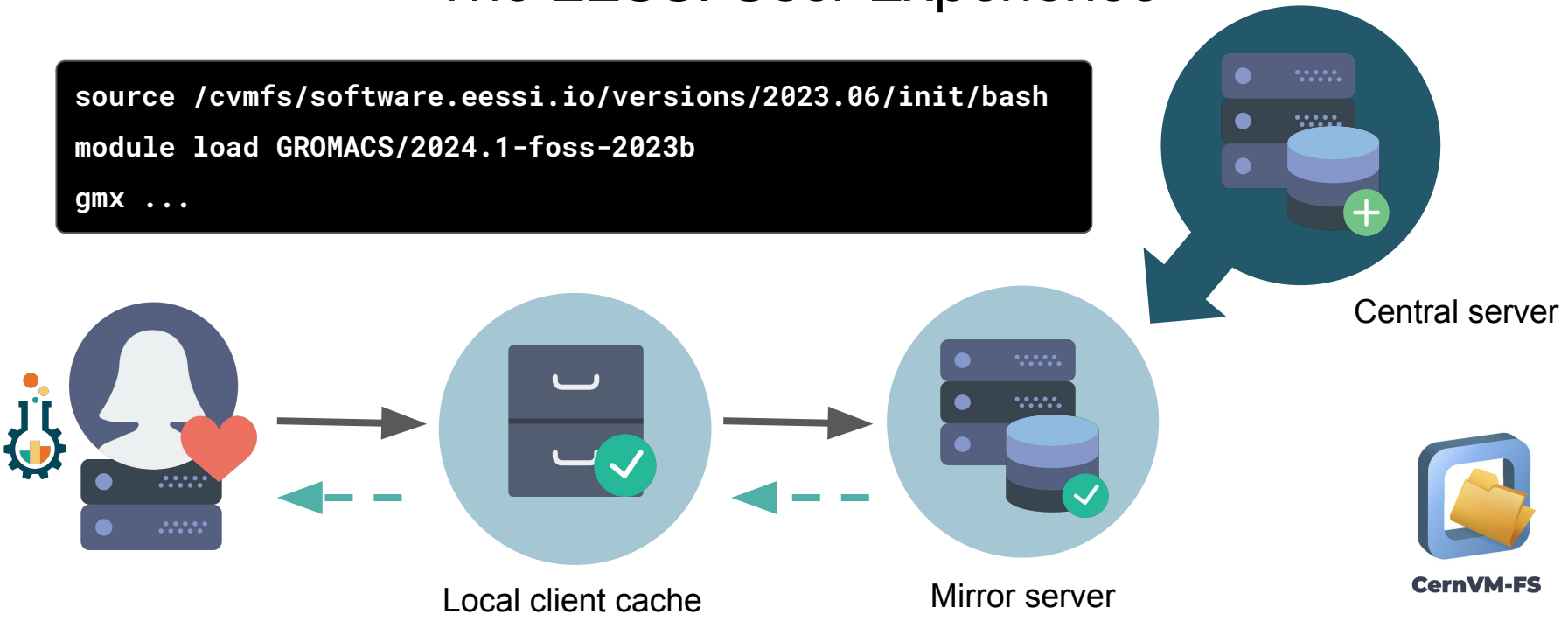
How does EESSI work?



- Software installations provided by 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

The EESSI User Experience

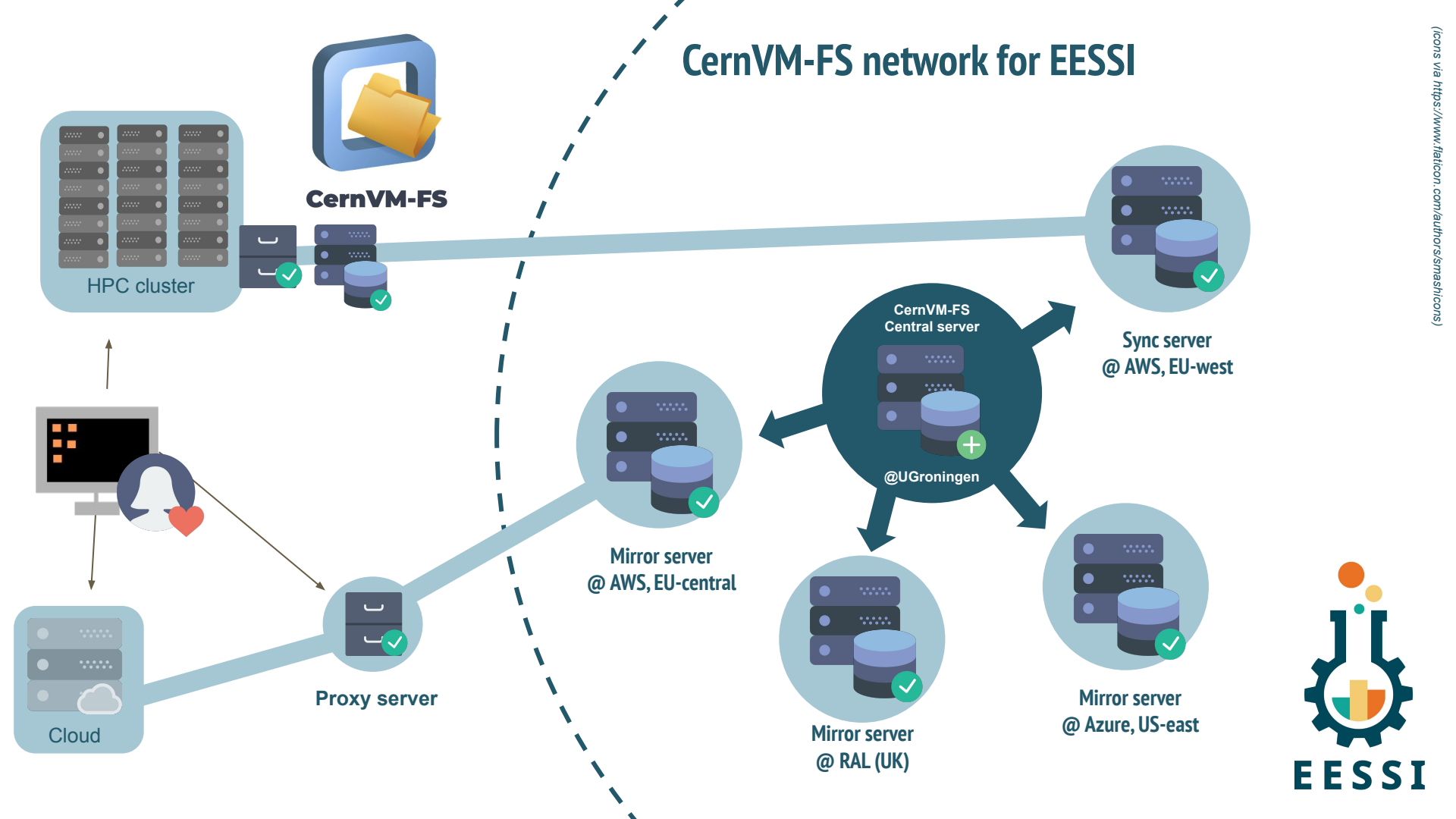
```
source /cvmfs/software.eessi.io/versions/2023.06/init/bash
module load GROMACS/2024.1-foss-2023b
gmx ...
```



EESSI provides **on-demand streaming**

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

CernVM-FS network for EESSI



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

Demo: The EESSI User Experience

```
source /cvmfs/software.eessi.io/versions/2023.06/init/bash
   Found EESSI repo @ /cvmfs/software.eessi.io/versions/2023.06!
   archdetect says x86_64/amd/zen3
   archdetect found supported accelerator for CPU target x86_64/amd/zen3: accel/nvidia/cc80
   Using x86_64/amd/zen3 as software subdirectory.
   Using /cvmfs/software.eessi.io/versions/2023.06/software/linux/x86_64/amd/zen3/modules/all
   Using ...
   ...
   Environment set up EESSI (2023.06), have fun!

git clone https://github.com/EESSI/eessi-demo.git
cd eessi-demo/GROMACS
./run.sh
```

Alternative ways of accessing EESSI are available, via a container image, via cvmfsexec, …
eessi.io/docs/getting_access/native_installation - eessi.io/docs/getting_access/eessi_container

Live demo: running GROMACS with EESSI

https://eessi.io

eessi-demo.sh

```
#!/bin/bash

source /cvmfs/software.eessi.io/versions/2025.06/init/bash
module load GROMACS/2025.2-foss-2025a

# download input file + unpack (if it's missing)
if [ ! -f ion_channel.tpr ]; then
  curl -OL https://repository.prace-ri.eu/ueabs/GROMACS/1.2/GROMACS_TestCaseA.tar.gz
  tar xfz GROMACS_TestCaseA.tar.gz
fi

# cleanup, to force new run
rm -f ener.edr logfile.log

# run GROMACS (downscaled to 1k steps instead of full run with 10k steps)
gmx mdrun -s ion_channel.tpr -maxh 0.50 -nsteps 1000 -g logfile
```

Live demo: running GROMACS with EESSI

https://eessi.io

```
macbook-pro $ lima shell eessi

lima $ ls /cvmfs/software.eessi.io
host_injections  init
README.eessi  versions

lima $ ./eessi-demo.sh
...
                (ns/day)    (hour/ns)
Performance:       2.215       10.834
```

EESSI

Live demo: running GROMACS with EESSI

https://eessi.io

```
macbook-pro $ lima shell eessi

lima $ ls /cvmfs/software.eessi.io
host_injections  init
README.eessi  versions

lima $ ./eessi-demo.sh
...
                 (ns/day)     (hour/ns)
Performance:        2.215        10.834
```

EESSI

```
$ ssh tier1.hpc.ugent.be

vsc $ qsub -l nodes=1:ppn=16 eessi-demo.sh
13001403
vsc $ cat eessi-demo.sh.*13001403
...
                 (ns/day)     (hour/ns)
Performance:       20.802        1.154
```

VLAAMS SUPERCOMPUTER CENTRUM

Live demo: running GROMACS with EESSI

https://eessi.io

```
macbook-pro $ lima shell eessi

lima $ ls /cvmfs/software.eessi.io
host_injections  init
README.eessi  versions

lima $ ./eessi-demo.sh
...
                  (ns/day)     (hour/ns)
Performance:         2.215        10.834
```

EESSI

```
$ ssh slurm-cluster-aws

aws $ P=aarch64-neoverse-v1-node
aws $ sbatch -p $P -c 8 eessi-demo.sh
113508
aws $ cat slurm-113508.out
...
                  (ns/day)     (hour/ns)
Performance:         7.758        3.094
```

```
$ ssh tier1.hpc.ugent.be

vsc $ qsub -l nodes=1:ppn=16 eessi-demo.sh
13001403
vsc $ cat eessi-demo.sh.*13001403
...
                  (ns/day)     (hour/ns)
Performance:        20.802        1.154
```

Live demo: running GROMACS with EESSI

https://eessi.io

```
macbook-pro $ lima shell eessi

lima $ ls /cvmfs/software.eessi.io
host_injections  init
README.eessi  versions

lima $ ./eessi-demo.sh
...
                    (ns/day)     (hour/ns)
Performance:           2.215        10.834
```

EESSI

```
$ ssh slurm-cluster-aws

aws $ P=aarch64-neoverse-v1-node
aws $ sbatch -p $P -c 8 eessi-demo.sh
113508
aws $ cat slurm-113508.out
...
                    (ns/day)     (hour/ns)
Performance:           7.758        3.094
```

```
$ ssh tier1.hpc.ugent.be

vsc $ qsub -l nodes=1:ppn=16 eessi-demo.sh
13001403
vsc $ cat eessi-demo.sh.*13001403
...
                    (ns/day)     (hour/ns)
Performance:          20.802        1.154
```

```
$ ssh login.deucalion.macc.fccn.pt

deucalion $ sbatch -p normal-arm eessi-demo.sh
Submitted batch job 7654321

deucalion $ cat slurm-7654321.out
...
                    (ns/day)     (hour/ns)
Performance:           8.960        2.678
```

Getting access to EESSI via CernVM-FS (live demo?)


```
# Native installation of EESSI using CernVM-FS, requires admin privileges
# Installation commands are for RHEL-based distros
# Like CentOS, Rocky Linux, Almalinux, Fedora, ...

# Install CernVM-FS to get access to CernVM-FS repositories (incl. EESSI)
sudo yum install -y
  https://ecsft.cern.ch/dist/cvmfs/cvmfs-release/cvmfs-release-latend.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"

# Complete setup of CernVM-FS
sudo cvmfs_config setup
```

Alternative ways of accessing EESSI are available, via a container image, via cvmfsexec, …
eessi.io/docs/getting_access/native_installation - eessi.io/docs/getting_access/eessi_container

Performance aspects of EESSI

- EESSI provides **optimized** software installations
  - Built specific CPU microarchitectures, which one to use is auto-detected
  - Performance is not sacrificed for mobility-of-compute (cfr. containers)
- You can **get access** to EESSI on any Linux system in **minutes** by installing CernVM-FS
- Software is **streamed in on-demand** as it is being used
  - Only what's actually required gets downloaded (as opposed to containers)
  - CernVM-FS uses multi-level cache hierarchy to avoid re-downloading
- Improved **startup performance** vs software installations on parallel filesystems
  - Typical access pattern of software installations is a bad fit for GPFS, Lustre, etc.
  - Metadata access are kept local to the client (+ extensive caching)
- **Humans waste less time** in getting scientific software installed

Software startup performance

- **Starting software** can be *slow* on supercomputers (WTF?!)
- Parallel filesystems are optimized for large (data) files & parallel I/O
- Software installations typically consist of (lots of) small files + have typical access pattern
- Installing your software on a parallel filesystem like GPFS or Lustre is ***not* recommended**
  - Often is the only feasible option (no other filesystem available)…
  - Putting the software in a container image helps a lot
  - But if you still want to build from source (to optimize your binaries to host CPU)
  - Container images for a central software stack are not really an option…

https://eessi.io/docs/training-events/2025/tutorial-best-practices-cvmfs-hpc/performance

Software startup performance - TensorFlow

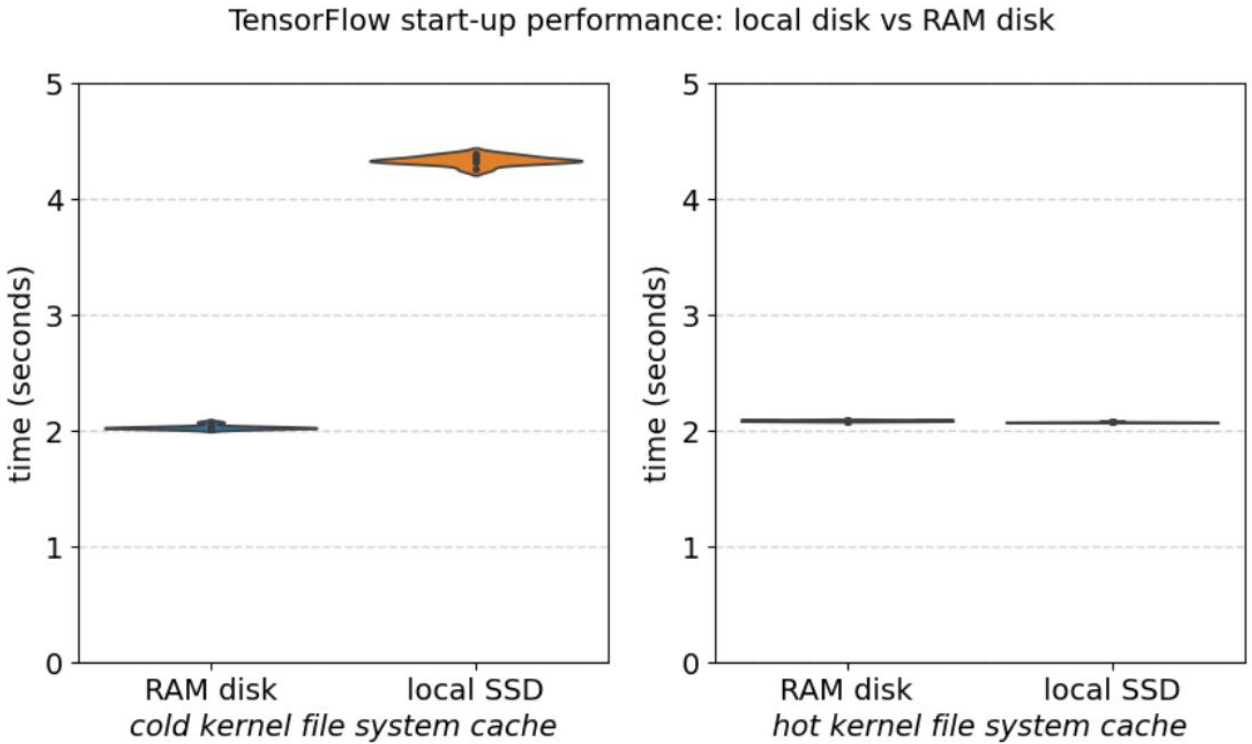
- Let's use TensorFlow v2.13.0 as a test case…

  How long does it take do to `python -c 'import tensorflow'` ?

- Triggers ~3,680 open() calls + ~510 opendir calls (which includes non-existing paths)
- Requires ~3,470 files, including:
  - ~2,200 files from the TensorFlow installation itself (~94% *.pyc files)
  - ~950 files from Python packages outside of the TensorFlow installation directory
  - 17 files from the EESSI compatibility layer
- Pulls in about ~1.1GB of data in total

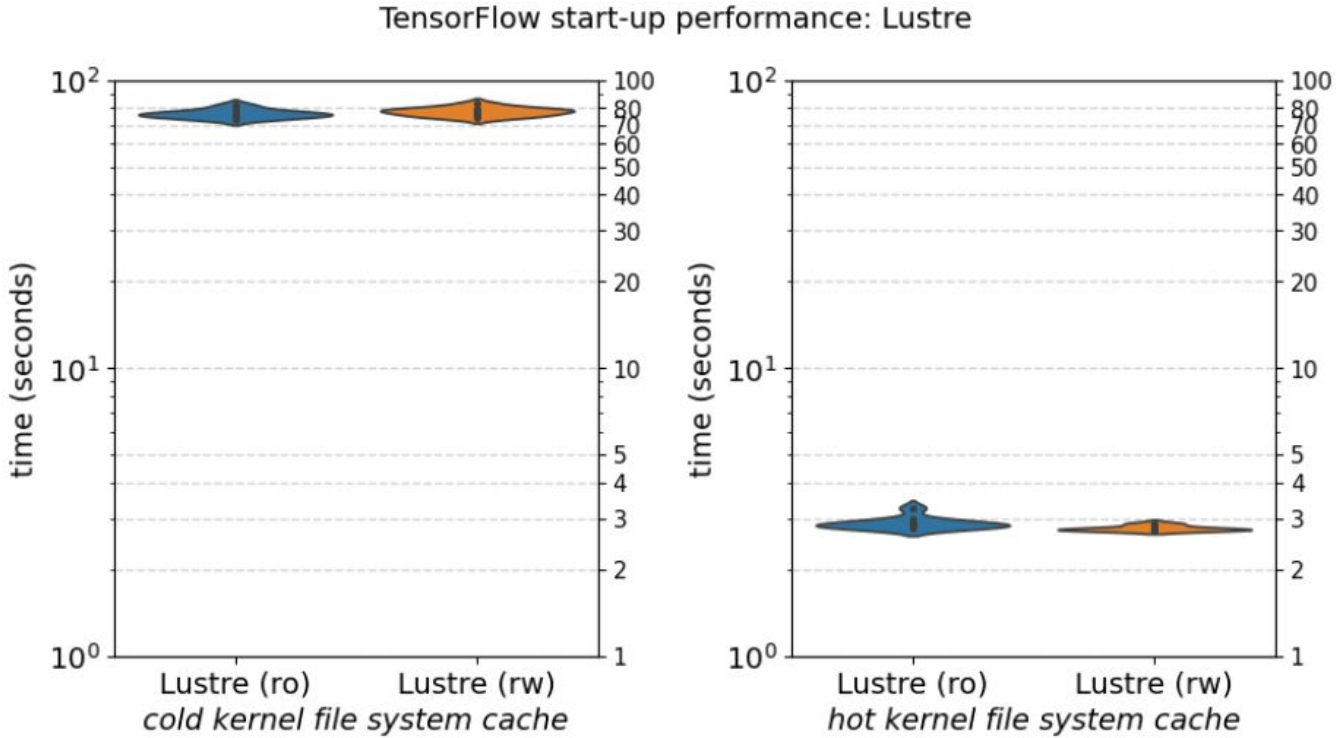
https://eessi.io/docs/training-events/2025/tutorial-best-practices-cvmfs-hpc/performance

Software startup performance - TensorFlow



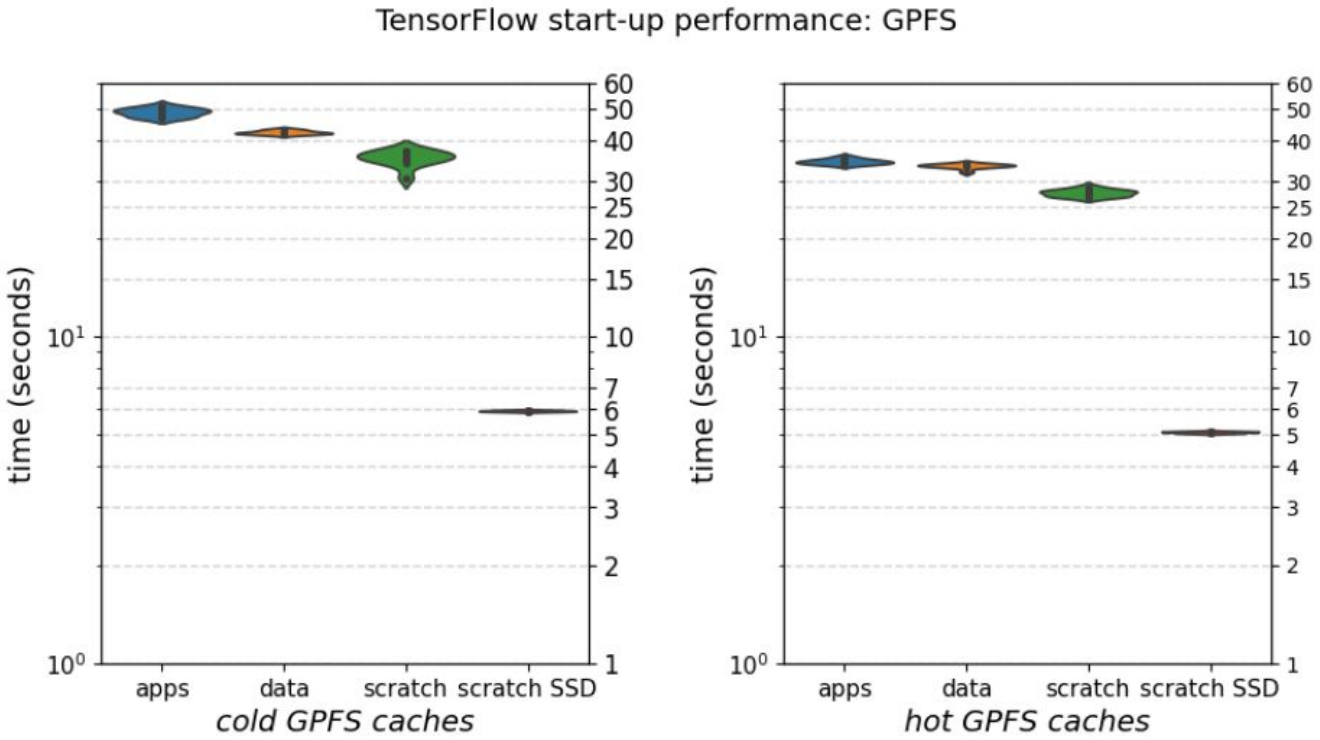
https://eessi.io/docs/training-events/2025/tutorial-best-practices-cvmfs-hpc/performance

Software startup performance - TensorFlow



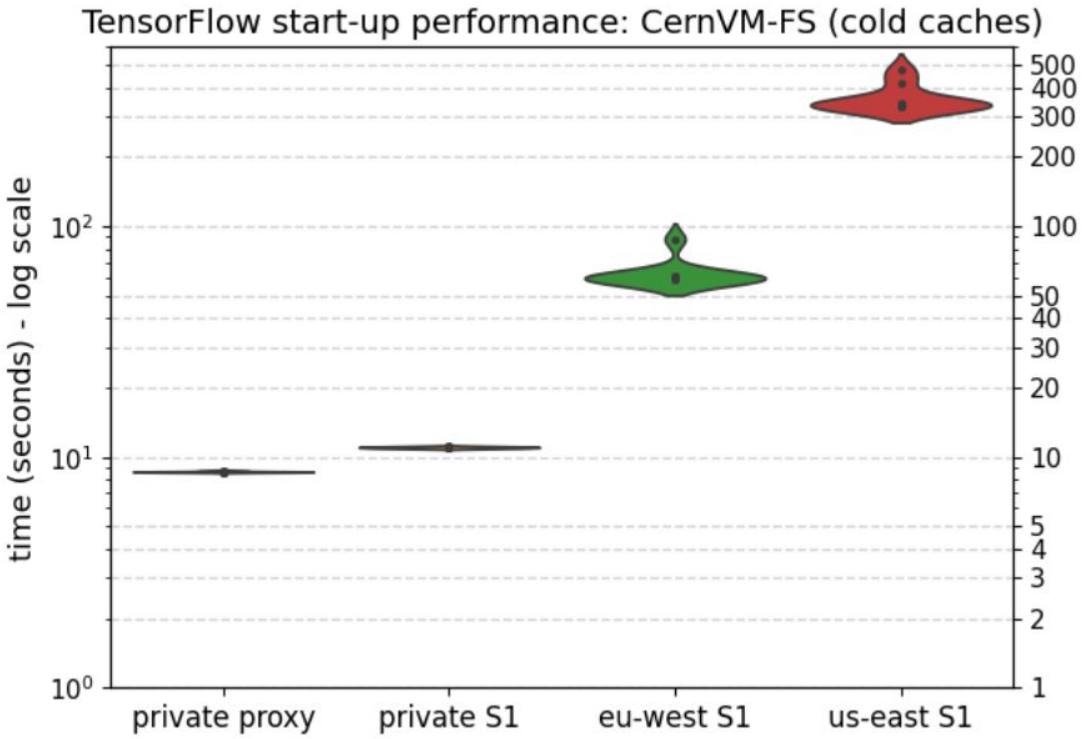
https://eessi.io/docs/training-events/2025/tutorial-best-practices-cvmfs-hpc/performance

Software startup performance - TensorFlow



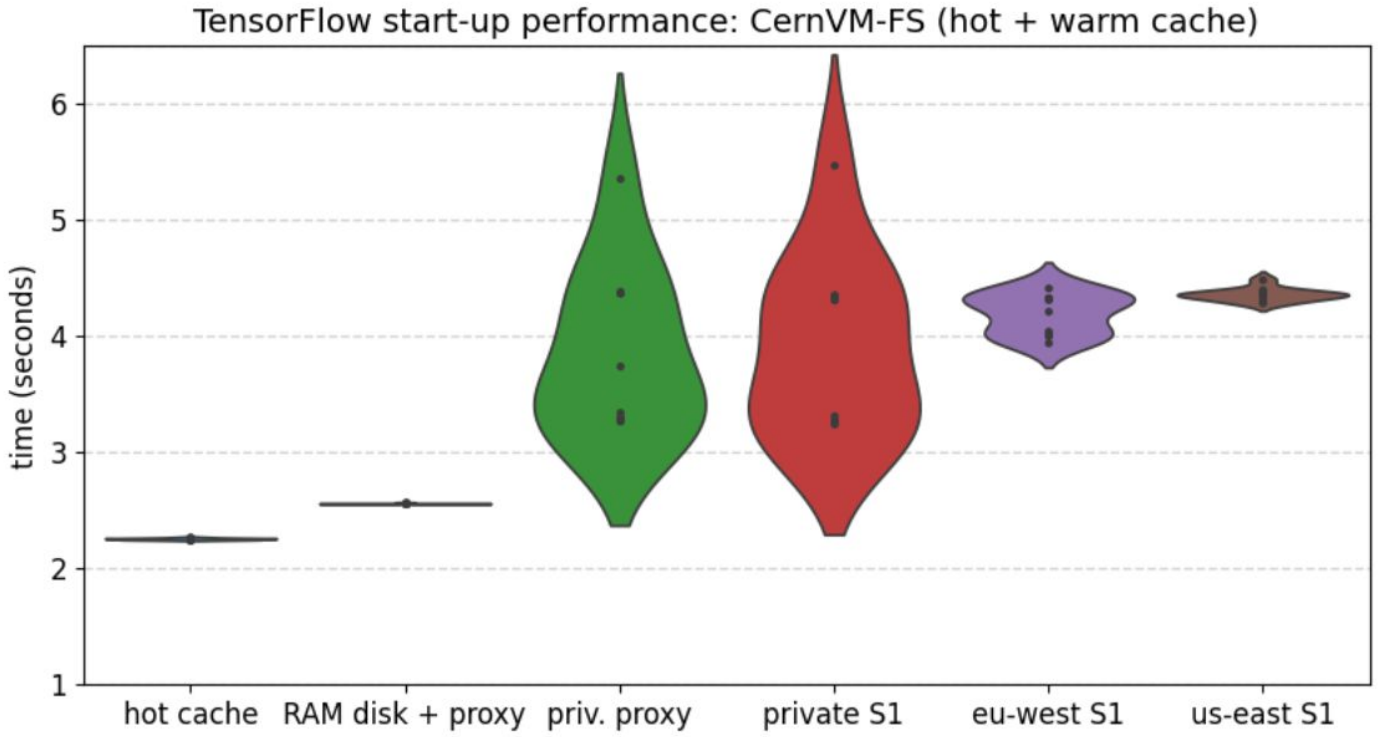
https://eessi.io/docs/training-events/2025/tutorial-best-practices-cvmfs-hpc/performance

Software startup performance - TensorFlow



https://eessi.io/docs/training-events/2025/tutorial-best-practices-cvmfs-hpc/performance

Software startup performance - TensorFlow

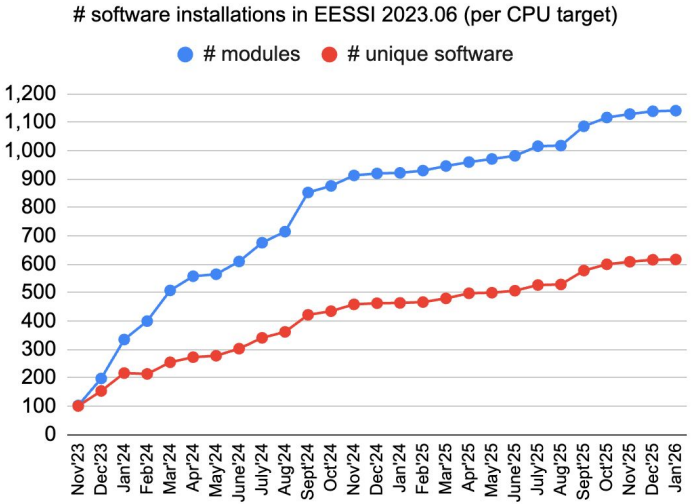


https://eessi.io/docs/training-events/2025/tutorial-best-practices-cvmfs-hpc/performance

Overview of available software in EESSI

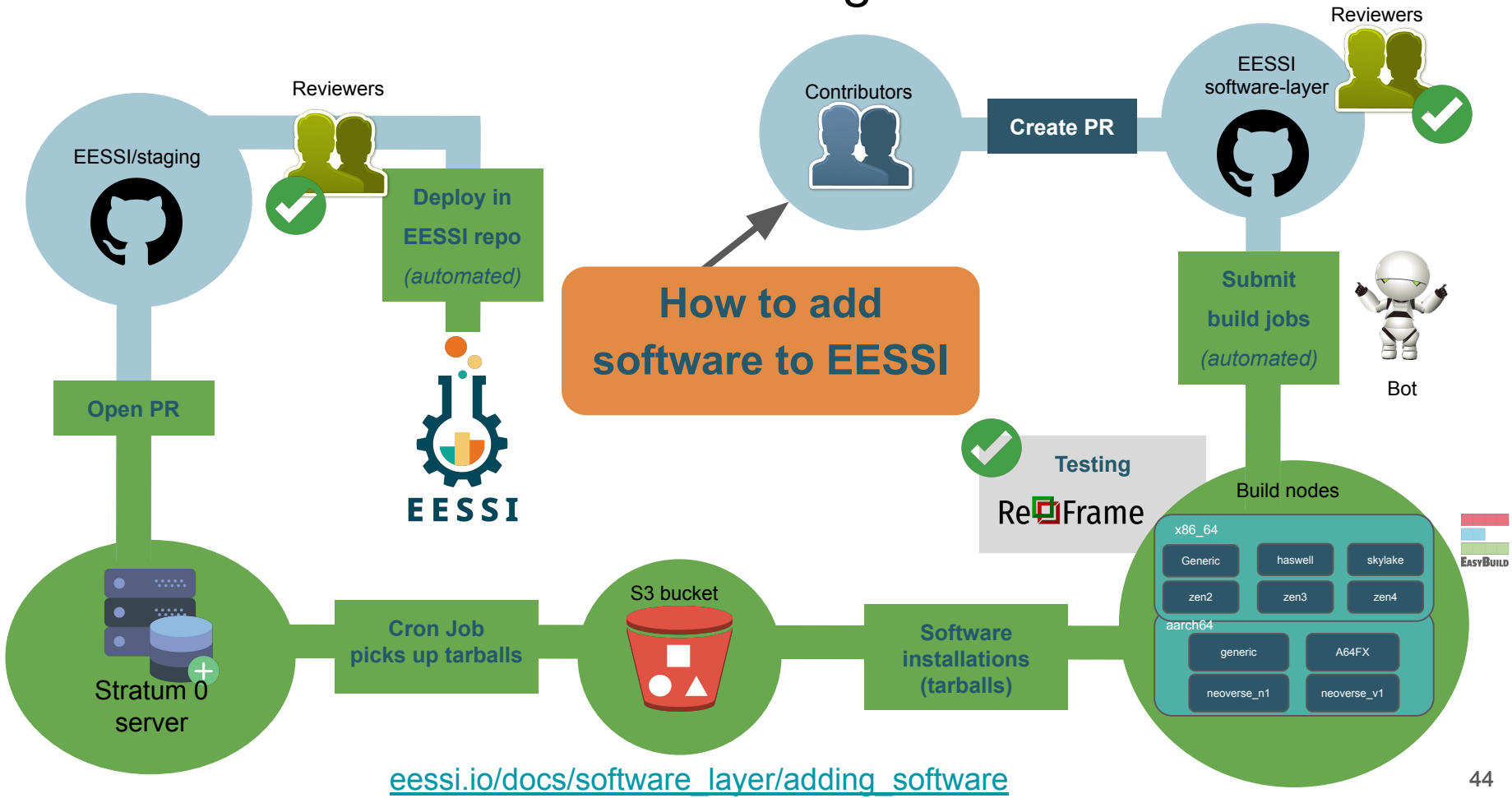
Currently more than 1,700 software installations available per supported CPU target via `software.eessi.io` CernVM-FS repository, increasing every week

- **14 supported CPU targets** (x86_64 + Arm), see https://eessi.io/docs/software_layer/cpu_targets
- **~650 different software packages**, excl. extensions: Python packages, R libraries
- **~25,000 software installations in total**
- Including ESPResSo, GROMACS, LAMMPS, OpenFOAM, PyTorch, R, QuantumESPRESSO, TensorFlow, waLBerla, WRF, …
- https://eessi.io/docs/available_software/overview



- Software built with `foss/2023a` and `foss/2023b` toolchains in EESSI 2023.06
- Software built with `foss/2024a`, `foss/2025a`, `foss/2025b` toolchains in EESSI 2025.06

Semi-automated workflow for adding software to EESSI



eessi.io/docs/software_layer/adding_software

On which systems is EESSI available today?

- On supercomputers in Flanders (Vlaams Supercomputer Centrum, VSC):
  - **Readily available on Tier-2 infrastructure at UGent & VUB + Tier-1 Hortense**
  - Tier-1 cloud: can deploy it yourself and have access to a full software stack in minutes
- EESSI is already available on various other European systems (and beyond)
  - EuroHPC JU systems incl. Vega, Karolina, MareNostrum 5, Deucalion, Discoverer, …
  - Snellius @ SURF, EMBL, Univ. of Stuttgart, Sigma2 in Norway, etc.
- EESSI can be used in virtual machine in European Open Science Cloud (EOSC),

  see also https://www.eessi.io/docs/blog/2025/10/22/eosc
- **Overview of (known) systems that have EESSI available at https://eessi.io/docs/systems**

EuroHPC Federation Platform

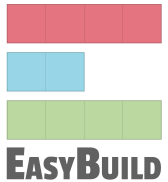
- Ghent University is part of the consortium that is developing the EuroHPC Federation Platform (EFP)
- EFP will be “one-stop shop” for people to access & use EuroHPC systems
- For entire EuroHPC ecosystem: supercomputers, AI factories, quantum
- **We are integrating EESSI into EFP as Federated Software Stack**
- First version of EFP expected to be operational in April 2026…
- More info via https://my-eurohpc.eu
- 1st webinar series coming (really) soon: https://my-eurohpc.eu/training

Webinar series: Different aspects of EESSI

Series of presentations (May-June 2025)

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

- Introduction to EESSI
- Introduction to CernVM-FS
- Introduction to EasyBuild
- EESSI for CI/CD
- Using EESSI as the base for a system stack



Slides + recordings available, we hope to repeat these soon!

MultiXscale

Co-funded by the European Union

EuroHPC Joint Undertaking

Web page: multixscale.eu

Facebook: MultiXscale

Twitter: @MultiXscale

LinkedIn: MultiXscale

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.

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: Mondays 2pm CE(S)T