EESSI: A streamed, production-quality, multi-platform, optimised scientific software stack

SIGHPC Education Webinar

Kenneth Hoste (UGent), Alan O’Cais (CECAM)
About us

Kenneth Hoste

● HPC sysadmin + user support at Ghent University (Belgium) since 2010
● Lead dev. of EasyBuild - tool to install scientific software on HPC infra
● Active contributor to EESSI, partner in MultiXscale EuroHPC CoE

Alan O’Cais

● HPC RSE with CECAM, hosted at the University of Barcelona
● >10 years experience leading EU funded HPC projects
● Currently technical manager with MultiXscale, a EuroHPC CoE
The issue: getting scientific software installed

- **Scientific software is a different breed of software**

- Scope of (large-scale) computational science is expanding
  - More users (bioinformatics, AI/ML, …)
  - Less experience in software deployment, optimization, etc.

- Explosion of open source scientific software in recent years

- Increasing variety in hardware
  - CPUs: Intel, AMD, Arm, POWER, soon also RISC-V, …
  - Accelerators: NVIDIA, AMD, Intel Xe, …

- Rise of the cloud: Microsoft Azure, Amazon EC2, Google, Oracle, …

- In stark contrast: available manpower in HPC support teams!
EESSI in a nutshell

- European Environment for Scientific Software Installations (EESSI)
- Shared repository of (optimized!) scientific software *installations*
- Avoid duplicate work across (HPC) sites 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 (+ WSL, and possibly macOS) 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

https://www.eessi-hpc.org
https://eessi.github.io/docs (try out the pilot setup!)
Optimized scientific software installations

- Software should be optimized for the system it will run on
- 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

![Performance Chart]

70% speedup!
Major goals of EESSI

● **Avoid duplicate work** (for researchers, HPC support teams, …)
  ○ Tools that automate software installation process
    (EasyBuild, Spack) are not sufficient
  ○ 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 with containers/conda)

● Facilitate HPC training, development of (scientific) software, …
Sounds great, but… how? We need…

- A way to get the software distributed globally
- Abstraction from the OS (like a container)
- Optimized builds for a large range of hardware architectures
- Selection of the right optimization at runtime (ideally automatic)
Sounds great, but... how? We need...

- A way to get the software distributed globally
- Abstraction from the OS (like a container)
- Optimized builds for a large range of hardware architectures
- Selection of the right optimization at runtime (ideally automatic)
High-level overview of EESSI project

Software layer
Optimized applications + dependencies

Compatibility layer
Levelling the ground across client OSs

Filesystem layer
Distribution of the software stack

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

Host operating system (Linux, macOS, WSL)
- 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, …
Compatibility layer [https://github.com/EESSI/compatibility-layer](https://github.com/EESSI/compatibility-layer)

- **Gentoo Prefix** installation (in `/cvmfs/.../compat/<os>/<arch>/`)
- Set of 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, ppc64le, x86_64, riscv64)
- **Levels the ground for different client operating systems** (Linux distros, later also macOS?)
- Currently in pilot repository:
  - `/cvmfs/pilot.eessi-hpc.org/versions/2021.12/compat/linux/aarch64`
  - `/cvmfs/pilot.eessi-hpc.org/versions/2021.12/compat/linux/ppc64le`
  - `/cvmfs/pilot.eessi-hpc.org/versions/2021.12/compat/linux/x86_64`
Software layer  

https://github.com/EESSI/software-layer

- Provides scientific software applications, libraries, and dependencies
- Optimized for specific CPU microarchitectures (Intel Haswell, ...)
  - Separate subdirectory/tree for each (in /cvmfs/.../software/...)
- 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 archspec
Testing

Software testing is always important, but if everyone uses the same stack, even more so!

- Development of portable test suite with ReFrame
- ReFrame: designed to run tests on HPC clusters
  - Can interact with batch systems
  - Allows testing of applications at scale
  - Also allows simple tests on a single node (laptop, cloud node)

- We can test functionality (correctness), but also performance + scaling

https://github.com/EESSI/test-suite
EESI paper (open access)  
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)
Current status

- Working **proof of concept** (see [https://eessi.github.io/docs/pilot](https://eessi.github.io/docs/pilot))
- Ansible playbooks, scripts, docs at [https://github.com/eessi](https://github.com/eessi)
- CernVM-FS: Stratum 0 @ Univ. of Groningen + four Stratum 1 servers
- Software (CPU-only): Bioconductor, GROMACS, OpenFOAM, R, TensorFlow, …
- Hardware targets:
  - {aarch64,ppc64le,x86_64}/generic
  - intel/{haswell,skylake_avx512}, amd/{zen2,zen3},
    aarch64/{graviton2,graviton3}, ppc64le/power9le
- Supported by Azure and AWS: sponsored credits to develop necessary infrastructure
- Going forward: **Funded via MultiXscale, a EuroHPC Centre of Excellence**
Current ongoing effort

- Developing a **bot to automate workflow of adding software** to EESSI repository
  - Enable community contributions to EESSI, via pull requests to GitHub repo
  - Automated building of software for different support CPU targets
  - Automated deployment of software into EESSI (under human supervision)
  - Eventually also testing of software installations before deploying them

- Add **support for leveraging (NVIDIA) GPUs** using software provided in EESSI
  - Complicating factors: (version of) GPU drivers, CUDA license, …

- Working towards **making EESSI stable and reliable**…

- More details available in various presentations from EESSI Community Meeting (Sept’22)
  [https://eessi.github.io/docs/meetings/2022-09-amsterdam](https://eessi.github.io/docs/meetings/2022-09-amsterdam)
MultiXscale: Key Points

- **EuroHPC Centre of Excellence**
  - 4 year project, started Q1 2023

- **Budget of ~6M EUR (50% EU funding, 50% national funding)**
  - Roughly 50% of funding for EESSI-related activities

- **Collaboration between EESSI and CECAM (total of 16 partners)**
  - EESSI primarily addresses technical aspects
  - CECAM network provides scientific expertise

- **Scientific target are multiscale simulations with 3 key use cases**
  - Helicopter design and certification for civil transport
  - Battery applications to support the sustainable energy transition
  - Ultrasound for non-invasive diagnostics and biomedical applications
Demo scenarios

https://github.com/EESSI/eessi-demo

- Demo 1: Using an “empty” **Ubuntu 22.04 VM** in AWS (**Arm Graviton2**)
  - No CernVM-FS installed, EESSI not available yet, but only takes 2 min.
  - Requires admin rights (**sudo** to install extra packages)
  - Set up EESSI environment by sourcing init script
  - Running EESSI demo scripts

- Demo 2: On HPC-UGent infrastructure (**RHEL 8.6, AMD Rome**)
  - EESSI CernVM-FS repository readily available (by the friendly HPC-UGent sysadmins)
  - Leverage software installations provided by EESSI in job scripts
  - Anyone who has an account on the HPC-UGent infrastructure can do this!
Demo 1: Ubuntu 22.04 Arm VM in AWS (1/3)

- We need to:
  - Install CernVM-FS packages
  - Install EESSI CernVM-FS configuration (`cvmfs-eessi-config*` package)
  - Set up minimal client configuration in `/etc/cvmfs/default.local`

- For production usage (especially large-scale), you should also:
  - Use a squid proxy, next to a local client cache (better start-up performance)
  - Set up your own Stratum-1 mirror server (protection against network disconnects)
  - Also recommended to “be a good citizen” in the EESSI CernVM-FS network

https://github.com/EESSI/eessi-demo
Commands to install CernVM-FS + EESSI configuration for CernVM-FS

Assumption: using Ubuntu as OS (only matters for `apt-get/dpkg` commands)

```bash
$ cat eessi-demo/scripts/install_cvmfs_eessi_Ubuntu.sh
sudo apt-get install lsb-release
wget https://ecsft.cern.ch/dist/cvmfs/cvmfs-release/cvmfs-release-latest_all.deb
sudo dpkg -i cvmfs-release-latest_all.deb
sudo apt-get update
sudo apt-get install -y cvmfs
wget https://github.com/EESSI/filesystem-layer/releases/download/latest/cvmfs-config-eessi_latest_all.deb
sudo dpkg -i cvmfs-config-eessi_latest_all.deb
sudo bash -c "echo 'CVMFS_CLIENT_PROFILE="single"' > /etc/cvmfs/default.local"
sudo bash -c "echo 'CVMFS_QUOTA_LIMIT=10000' >> /etc/cvmfs/default.local"
sudo cvmfs_config setup
```

https://github.com/EESSI/eessi-demo
Once CernVM-FS + EESSI configuration is installed, you’re good to go!

Set up EESSI environment by sourcing the init script, load modules, run.

```bash
$ ls /cvmfs/pilot.eessi-hpc.org
host_injections latest versions

$ source /cvmfs/pilot.eessi-hpc.org/latest/init/bash
...
Environment set up to use EESSI pilot software stack, have fun!

$ module avail GROMACS TensorFlow OpenFOAM Bioconductor

-------- /cvmfs/pilot.eessi-hpc.org/versions/2021.12/software/linux/aarch64/graviton2/modules/all --------

GROMACS/2020.1-foss-2020a-Python-3.8.2 OpenFOAM/9-foss-2021a
GROMACS/2020.4-foss-2020a-Python-3.8.2 (D) R-bundle-Bioconductor/3.11-foss-2020a-R-4.0.0
OpenFOAM/v2006-foss-2020a TensorFlow/2.3.1-foss-2020a-Python-3.8.2
OpenFOAM/8-foss-2020a
```

https://github.com/EESSI/eessi-demo

Demo 1: Ubuntu 22.04 Arm VM in AWS (3/3)
Demo 2: On HPC-UGent infrastructure

- OS: RHEL 8.6 - Slurm
- CPUs: mix of different generations of Intel and AMD CPUs
- Assumption: EESSI is already available to use
- HPC team has installed and configured CernVM-FS to provide access to EESSI
- Incl. properly setting up squid proxy (cache) + local Stratum-1 (caching + reliability)
- Researchers who have an HPC account can leverage software provided by EESSI
- **Just source EESSI init script, load modules, and you’re ready to go!**
  
  ```bash
  source /cvmfs/pilot.eessi-hpc.org/latest/init/bash
  ```
Try out EESSI yourself using Apptainer!

- Only Apptainer (or Singularity) is required to run the EESSI client container
- **Should work on any Linux distribution**, on Intel/AMD/Arm/POWER CPUs
- Detailed instructions available at [https://eessi.github.io/docs/pilot](https://eessi.github.io/docs/pilot)

```
$ apptainer shell --fusemount "$EESSI_PILOT" docker://ghcr.io/eessi/client-pilot:centos7
...
Apptainer> ls /cvmfs/pilot.eessi-hpc.org/
 2021.06  host_injections  latest  versions
Apptainer> source /cvmfs/pilot.eessi-hpc.org/latest/init/bash

Found EESSI pilot repo @ /cvmfs/pilot.eessi-hpc.org/versions/2021.12!
archspec says x86_64/amd/zen2
Using x86_64/amd/zen2 as software subdirectory.
Using /cvmfs/pilot.eessi-hpc.org/versions/2021.12/software/linux/x86_64/amd/zen2/modules/all as the
directory to be added to MODULEPATH.
Found Lmod configuration file at
Initializing Lmod...
Prepending /cvmfs/pilot.eessi-hpc.org/versions/2021.12/software/linux/x86_64/amd/zen2/modules/all to
$MODULEPATH...
```
EESSI and Education
What can the impact be?
The European context - EuroHPC Joint Undertaking

- Developing a pan-European supercomputing infrastructure (including 2 in top 5 of TOP500)

- Supporting research and innovation activities:
  - Developing a European supercomputing ecosystem
  - Stimulating a technology supply industry
  - Making supercomputing resources available to a large number of public and private users, including small and medium-sized enterprises

- €1.1 billion for the period 2018-20

- For 2021-2027, total budget of about €7 billion

- Expectation:
  - Resources will be open to all but access will be competitive
  - Of the 32 EuroHPC countries, those with “culture” of HPC will have an implicit advantage
The European context - HPC Education and Training

- 2018 technical report by European Commission’s science and knowledge service
  - “Academic offer and demand for advanced profiles in the EU - Artificial Intelligence, High Performance Computing and Cybersecurity”

<table>
<thead>
<tr>
<th>Domain</th>
<th>Number of tracked programmes</th>
<th>% of each domain over total programmes in any domain</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>All levels</td>
<td>Bachelor</td>
</tr>
<tr>
<td>AI</td>
<td>2,054</td>
<td>765</td>
</tr>
<tr>
<td>HPC</td>
<td>1,102</td>
<td>369</td>
</tr>
<tr>
<td>CS</td>
<td>1,179</td>
<td>532</td>
</tr>
<tr>
<td>Total nr of programmes in AI, HPC and CS</td>
<td>3,472</td>
<td>1,339</td>
</tr>
</tbody>
</table>
HPC Education and Training Crisis?

- **UK had 62% of all EU HPC Programmes**
- Spending money on hardware is “easy”, but hardware investment lifetime in HPC is ~5 years
- Bootstrapping that investment so that you see the impact in your IP and SMEs is hard
  - Need a lot of people who can leverage, improve and contribute to these kinds of infrastructures
  - Importing expertise is not a sustainable/desirable option (including internal EU “brain drain”)
  - Need a talent identification/development/integration pipeline...

* The report used an AI methodology that was acknowledged as heavily biased towards English language courses
Scalable HPC Training

- HPC should not be for the elite, it’s in everyone's interest for it to be an accessible tool for all scientists (both academia and industry)

- If you want to serve 32 countries (in which there are 1000s of 3rd level institutions and >500 million people), *everything* you do should be inherently scalable

- To cast a wide net, you need to address a lot of challenges
  - Technical:
    - connectivity, hardware infrastructure, **software infrastructure**, configuration
  - Pedagogical:
    - (multilingual) education materials and the instructors to teach them
Cloud-based clusters as scalable resources

- “Real” systems come with strict security requirements and plenty of bureaucracy for the instructor/learner
  - These are all intimidating barriers to the learning experience (particularly for beginners)
- If I run a large HPC centre, would I seriously consider running training courses for arbitrary European institutions and companies on my production system?
- **Cloud-based clusters are unencumbered**
  - Bring them up, take them down, throw them away
  - Can agree in advance that nothing there is considered secure
- **Cloud-based clusters don’t have to be toys**
  - Successfully tested EESSI with Infiniband fabric on Azure and EFA fabric on AWS
  - Can be configured to run with scalable file systems (e.g., Lustre)
- **Cloud-based clusters can be reproducible**, and reproducible means **scalable** (e.g. event specific)
  - Can contribute centrally to deployment tools (such as Magic Castle) to include lots of additional features such as JupyterHub, RStudio, VScode, visualisation capabilities,...
EESSI Facilitates HPC training

- EESSI can significantly reduce effort required to set up software infrastructure for HPC training sessions (introductory, software-specific, …)
  - System independence of training content (to an extent)
  - Facilitate transition to new systems, and new hardware

- EESSI can be used as a base to provide additional (scientific) software that is required for the training

- Setting up a throwaway Slurm cluster in the cloud is made easy via tools such as CitC or Magic Castle (where EESSI is already a native option)
  - Also vendor-specific options such as CycleCloud and ParallelCluster

- Attendees can easily set up the same software environment later on their own system(s) by leveraging EESSI
We’ve seen EESSI can be used as an infrastructure for educational content delivery.

Can also be leveraged to support content development, for example:

- “HPC development best practices” are a recurring theme
- Continuous integration in an HPC context is very hard
  - Need bleeding-edge compilers, MPI runtimes, dependencies,...
  - Need access to a spectrum of hardware
- EESSI can help support the development of CI workflows
  - Quickly reproducible building/testing environments on many architectures
  - Access to hardware as a service (via cloud providers)
We have an EESSI GitHub Action as a shorthand for this:

```yaml
name: ubuntu_gromacs
on: [push, pull_request]
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
    - uses: actions/checkout@v2
    - uses: eessi/github-action-eessi@main
      with:
        eessi_stack_version: '2021.06'
    - name: Test EESSI
      run: |
        module load GROMACS
        gmx --version
        shell: bash
```

See it in action in the [github-essi-action repository](https://github.com/EESSI/github-action-eessi):

- [github.com/EESSI/github-action-eessi](https://github.com/EESSI/github-action-eessi)
Leveraging EESSI GitHub Action

https://github.com/EESSI/github-action-eessi/actions/runs/3638276955/jobs/6140289351
Paper (open access): https://doi.org/10.1002/spe.3075

Website: https://www.eessi-hpc.org

Join our mailing list & Slack channel
https://www.eessi-hpc.org/join

Documentation: https://eessi.github.io/docs

GitHub: https://github.com/eessi

Twitter: @eessi_hpc

EESSI YouTube channel

Monthly online meetings (first Thursday, 2pm CEST)