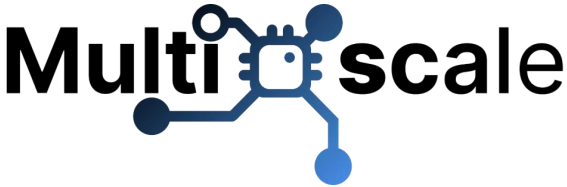
NVIDIA & AMD GPU support in EESSI

Caspar van Leeuwen

HPML Advisor @ SURF

23-04-2026



Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European High Performance Computing Joint Undertaking (JU) and countries participating in the project. Neither the European Union nor the granting authority can be held responsible for them.

What we'll talk about

- History of NVIDIA support in EESSI
- NVIDIA: Runtime vs Compile time support
  - Site admins need configure a CVMFS variant symlink & symlink drivers in the target path
- AMD GPU support status
- AMD GPU support – future work

A bit of history on NVIDIA support

- Apr 2023: Presentation at EUM23 (by me), current activities “NVIDIA GPU support”
- Q4 2023: First traces of CUDA support in EESSI

add placeholder page on GPU support in EESSI #128

Merged ocaisa merged 1 commit into EESSI:main from boegel:gpu on Nov 22, 2023

Build CUDA under `host_injections` and make EESSI aware of host CUDA drivers #368

Merged casparvl merged 19 commits into EESSI:2023.06-pilot.eessi-hpc.org from ocaisa:host_injections_cuda on Dec 1, 2023

- Oct 2024: Archdetect + EESSI init support

enhance archdetect to support detection of NVIDIA GPUs + using that in EESSI init script #767

Merged ocaisa merged 8 commits into EESSI:2023.06-software.eessi.io from boegel:2023.06-software.eessi.io_archdetect_nvidia on Oct 9, 2024

- Q1 2025: Move CUDA software to GPU-specific prefix

Add Lmod startup hook that prints an error when loading removed/relocated modules #962

Merged ocaisa merged 3 commits into EESSI:2023.06-software.eessi.io from bedroge:lmod_hook_removed_modules on Mar 4, 2025

- May 2025: CUDA sanity check in EasyBuild

add a CUDA device code sanity check #4692

Merged boegel merged 120 commits into easybuilders:develop from jfgrimm:cuda-device-code-sanity-check on May 20, 2025

- Jul 2025: Document supported targets

Add docs on GPU support #504

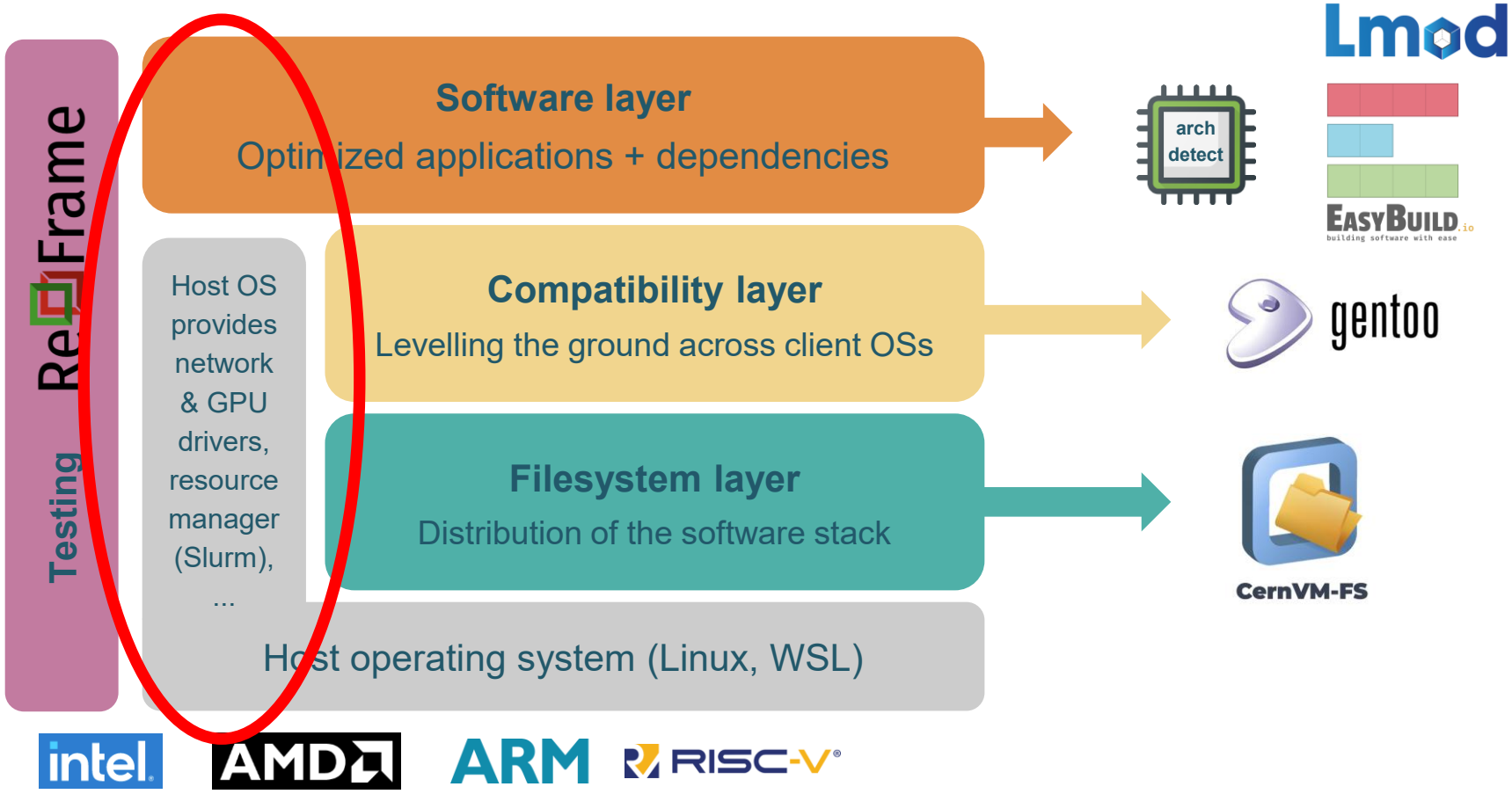
Merged bedroge merged 5 commits into EESSI:main from casparvl:add_supported_gpu_archs on Jul 7, 2025

- Feb 2026: CUDA support in 2025.06

{2025.06}[SYSTEM] CUDA 12.6.0,12.8.0, cuDNN 9.5.0.50,9.10.1.4 #1351

Merged bedroge merged 7 commits into EESSI:main from casparvl:CUDA_12.6.0_2025.06 on Feb 19

High-level overview of EESSI project



NVIDIA runtime support

Runtime support = support to run a binary from EESSI that uses CUDA

Challenge 1: How to make our runtime linker find the driver on host XYZ?

- Add trusted search path to runtime linker
- Site admins need configure a CVMFS variant symlink & symlink drivers in the target path

See https://www.eessi.io/docs/site_specific_config/gpu/

Demo NVIDIA GPU runtime support

Install CVMFS

```
# Installation commands for Debian-based distros like Ubuntu, ...

# install CernVM-FS
sudo apt-get install lsb-release
wget https://cvmrepo.s3.cern.ch/cvmrepo/apt/cvmfs-release-latest_all.deb
sudo dpkg -i cvmfs-release-latest_all.deb
rm -f cvmfs-release-latest_all.deb
sudo apt-get update
sudo apt-get install -y cvmfs

# install EESSI configuration for CernVM-FS
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

# create client configuration file for CernVM-FS (no squid 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
```

Demo NVIDIA GPU runtime support

Initialize EESSI and try to use GPU software

```
# Initialize EESSI/2025.06
$ source /cvmfs/software.eessi.io/versions/2025.06/init/lmod/bash

# Load any CUDA-based module

$ module load OSU-Micro-Benchmarks/7.5-gompi-2024a-CUDA-12.6.0

Lmod has detected the following error:

You requested to load UCX-CUDA  which relies on the CUDA runtime environment and driver libraries. In
order to be able to use the module, you will need to make sure EESSI can find the GPU driver
libraries on your host system. The file being checked for on your system is
/cvmfs/software.eessi.io/versions/2025.06/compat/linux/x86_64/lib/nvidia/libcuda.so
You can override this check by setting the environment variable EESSI_OVERRIDE_GPU_CHECK but the loaded
application will not be able to execute on your system.
For more information on how to do this, see https://www.eessi.io/docs/site_specific_config/gpu/.

While processing the following module(s):
...
```

Configuring runtime support ¶

For CUDA-enabled software to run, it needs to be able to find the **NVIDIA GPU drivers** of the host system. The challenge here is that the NVIDIA GPU drivers are not *always* in a standard system location, and that we can not install the GPU drivers in EESSI (since they are too closely tied to the client OS and GPU hardware).

Enabling runtime support for a native EESSI installation (using the helper script)

To get runtime support, we need to ensure that the EESSI runtime linker can find the drivers. To do this, we symlink the drivers in a predictable location that is searched by the EESSI runtime linker.

Step 1: initialize a version of EESSI.

Demo NVIDIA GPU runtime support

Configure CUDA runtime support

```
# Configure CVMFS Variant Symlink
sudo mkdir -p /opt/eessi/nvidia-driver
sudo bash -c "echo 'EESSI_NVIDIA_OVERRIDE_DEFAULT=/opt/eessi/nvidia-driver' >>
/etc/cvmfs/domain.d/eessi.io.local"
sudo cvmfs_config reload
sudo chown cvanleeuwe:rsc_co_201934 /opt/eessi/nvidia-driver
ls -al /cvmfs/software.eessi.io/defaults/nvidia

lrwxrwxrwx 1 cvmfs cvmfs 10 Feb 17 13:09 /cvmfs/software.eessi.io/defaults/nvidia -> /opt/eessi/nvidia-
driver

# Run driver symlinking script
/cvmfs/software.eessi.io/versions/2025.06/scripts/gpu_support/nvidia/link_nvidia_host_libraries.sh

# Check that the drivers are now present in the dir from the original error message
ls -al /cvmfs/software.eessi.io/defaults/nvidia/

# Try to load module again:
module load OSU-Micro-Benchmarks/7.5-gompi-2024a-CUDA-12.6.0

# Try to actually run the code
mpirun -np 2 osu_bw -m 1048576:1048576 H D
```

NVIDIA runtime support

Why this works

```
$ module load EESSI/2025.06

# Show trusted search paths for the runtime linker

$ ld.so -help
…
Shared library search path:
…
  /cvmfs/software.eessi.io/versions/2025.06/compat/linux/x86_64/lib/nvidia (system search path)
  /cvmfs/software.eessi.io/versions/2025.06/compat/linux/x86_64/lib/amd (system search path)

# This is a CVMFS variant symlink (EESSI_202506_NVIDIA_OVERRIDE), default points to:

$ ls -al /cvmfs/software.eessi.io/versions/2025.06/compat/linux/x86_64/lib/nvidia
lrwxrwxrwx 1 cvmfs cvmfs 40 Feb 17 13:09
/cvmfs/software.eessi.io/versions/2025.06/compat/linux/x86_64/lib/nvidia ->
/cvmfs/software.eessi.io/defaults/nvidia

# This is another CVMFS variant symlink (EESSI_NVIDIA_OVERRIDE_DEFAULT)

$ ls -al /cvmfs/software.eessi.io/defaults/nvidia
lrwxrwxrwx 1 cvmfs cvmfs 24 Feb 17 13:09 /cvmfs/software.eessi.io/defaults/nvidia -> /opt/eessi/nvidia-
driver
```

NVIDIA runtime support

```
# To have runtime support, this should contain all the drivers:

$ ls -al /opt/eessi/nvidia-driver/
total 16
…
lrwxrwxrwx 1 cvanleeuwe rsc_co_201934   55 Apr 16 15:02 libOpenCL.so.1 ->
/usr/local/cuda/targets/x86_64-linux/lib/libOpenCL.so.1
lrwxrwxrwx 1 cvanleeuwe rsc_co_201934   36 Apr 16 15:02 libOpenGL.so.0 -> /lib/x86_64-linux-
gnu/libOpenGL.so.0
lrwxrwxrwx 1 cvanleeuwe rsc_co_201934   32 Apr 16 15:02 libcuda.so -> /lib/x86_64-linux-gnu/libcuda.so
lrwxrwxrwx 1 cvanleeuwe rsc_co_201934   34 Apr 16 15:02 libcuda.so.1 -> /lib/x86_64-linux-
gnu/libcuda.so.1
…
lrwxrwxrwx 1 cvanleeuwe rsc_co_201934   43 Apr 16 15:02 libnvidia-opencl.so.1 -> /lib/x86_64-linux-
gnu/libnvidia-opencl.so.1
…
```

NVIDIA runtime support

Some remarks

- Variant symlink setup was slightly different for 2023.06 (see https://www.eessi.io/docs/site_specific_config/gpu/)
- New setup (2025.06 and onwards) have the advantage that you only need to link drivers once, not once per EESSI version.
- Variant symlink `EESSI_202506_NVIDIA_OVERRIDE` is normally not needed, but would allow you to still do symlinking per EESSI version

NVIDIA compile time support

Compile time support = support to compile a new binary that uses CUDA, on top of EESSI

Challenge 2: not allowed to ship full CUDA Toolkit due to UELA

- Replace non-redistributable part by symlinks
- Site admins need configure a CVMFS variant symlink & install CUDA in the target path

See https://www.eessi.io/docs/site_specific_config/gpu/

NVIDIA GPU compile time support

Initialize EESSI and try to compile GPU software

```
# Initialize EESSI/2025.06
$ source /cvmfs/software.eessi.io/versions/2025.06/init/lmod/bash

# Load any CUDA based module

$ module load CUDA/12.9.1

Lmod has detected the following error:
You requested to load CUDA  but while the module file exists, the actual software is not entirely
shipped with EESSI due to licencing. You will need to install a full copy of the CUDA package where
EESSI
can find it.
For more information on how to do this, see https://www.eessi.io/docs/site_specific_config/gpu/.

While processing the following module(s):
…
```

Configuring compile time support ¶

To compile new CUDA software using dependencies from EESSI, additional configuration is needed.

The CUDA license and cuDNN license only allow redistribution of their *runtime* libraries. Thus, the installations of CUDA and cuDNN that come with EESSI have been stripped down to contain only the runtime libraries. A local installation of CUDA and cuDNN is required to compile new software.

A full CUDA SDK or cuDNN SDK is only needed to *compile* CUDA or cuDNN software

Without a full CUDA SDK or cuDNN SDK on the host system, you will still be able to *run* CUDA-enabled or cuDNN-enabled software from the EESSI stack (provided the required configuration for runtime support was done - see above), you just won't be able to *compile* additional CUDA or cuDNN software.

NVIDIA GPU compile time support

Configure CUDA compile time support

```
# Configure CVMFS Variant Symlink
sudo bash -c "echo 'EESSI_HOST_INJECTIONS=/opt/eessi' >> /etc/cvmfs/domain.d/eessi.io.local"
sudo cvmfs_config reload
sudo chown cvanleeuwe:rsc_co_201934 /opt/eessi

ls -al /cvmfs/software.eessi.io/host_injections

lrwxrwxrwx 1 cvmfs cvmfs 10 Oct  3  2023 /cvmfs/software.eessi.io/host_injections -> /opt/eessi

# Run CUDA installation script
/cvmfs/software.eessi.io/versions/2025.06/scripts/gpu_support/nvidia/install_cuda_and_libraries.sh --
accept-cuda-eula --accept-cudnn-eula

# Try to load module again:
module load CUDA/12.9.1

# Try to actually compile & run the code
nvcc hello.cu -o hello
./hello
Hello World from GPU!
```

NVIDIA compile time support

Why it works

```
# Symlinks from CUDA installations in EESSI now point to the local CUDA installation

$ ls -al
/cvmfs/software.eessi.io/versions/2025.06/software/linux/x86_64/intel/icelake/accel/nvidia/cc80/software/CUDA/12.9.1/bin/
total 25
…
lrwxrwxrwx  1 cvmfs cvmfs   77 Mar 10 17:22 nvcc ->
/cvmfs/software.eessi.io/host_injections/x86_64/software/CUDA/12.9.1/bin/nvcc

# Check what this actually points to
$ realpath /cvmfs/software.eessi.io/host_injections/x86_64/software/CUDA/12.9.1/bin/nvcc
/opt/eessi/x86_64/software/CUDA/12.9.1/bin/nvcc
```

NVIDIA compile time support

What `install_cuda_and_libraries.sh` does:

- Install all CUDA and cuDNN packages from all easystack files in `/cvmfs/software.eessi.io/versions/2025.06/scripts/gpu_support/nvidia/easystacks`
- Site admins should run regularly to make sure new versions are installed locally after they have been added to EESSI

See https://www.eessi.io/docs/site_specific_config/gpu/

NVIDIA GPU support – situation today

EESSI 2023.06:

3 CUDA Compute Capabilities (7.0, 8.0, 9.0), each for 13 CPU architectures

EESSI 2025.06:

5 CUDA Compute Capabilities (7.0, 8.0, 9.0, 10.0, 12.0), each for 15 CPU architectures

	CUDA compute capability				
CPU microarchitecture	7.0	8.0	9.0	10.0	12.0
aarch64/generic	2023.06: x, 2025.06: x	2023.06: x, 2025.06: x	2023.06: x, 2025.06: x	2025.06: x	2025.06: x
aarch64/a64fx	-	-	-	-	-
aarch64/neoverse_n1	2023.06: x, 2025.06: x	2023.06: x, 2025.06: x	2023.06: x, 2025.06: x	2025.06: x	2025.06: x
aarch64/neoverse_v1	2023.06: x, 2025.06: x	2023.06: x, 2025.06: x	2023.06: x, 2025.06: x	2025.06: x	2025.06: x
aarch64/nvidia/grace	2023.06: x, 2025.06: x	2023.06: x, 2025.06: x	2023.06: N, 2025.06: N	2025.06: x	2025.06: x
x86_64/generic	2023.06: x, 2025.06: x	2023.06: x, 2025.06: x	2023.06: x, 2025.06: x	2025.06: x	2025.06: x
x86_64/intel/haswell	2023.06: x, 2025.06: x	2023.06: x, 2025.06: x	2023.06: x, 2025.06: x	2025.06: x	2025.06: x
x86_64/intel/skylake_avx512	2023.06: N, 2025.06: N	2023.06: x, 2025.06: x	2023.06: x, 2025.06: x	2025.06: x	2025.06: x
x86_64/intel/cascadelake	2023.06: N, 2025.06: N	2023.06: x, 2025.06: x	2023.06: x, 2025.06: x	2025.06: x	2025.06: x
x86_64/intel/icelake	2023.06: x, 2025.06: x	2023.06: N, 2025.06: N	2023.06: x, 2025.06: x	2025.06: x	2025.06: x
x86_64/intel/sapphirerapids	2023.06: x, 2025.06: x	2023.06: x, 2025.06: x	2023.06: x, 2025.06: x	2025.06: x	2025.06: x
x86_64/amd/zen2	2023.06: x, 2025.06: x	2023.06: x, 2025.06: x	2023.06: x, 2025.06: x	2025.06: x	2025.06: x
x86_64/amd/zen3	2023.06: x, 2025.06: x	2023.06: N, 2025.06: N	2023.06: x, 2025.06: x	2025.06: x	2025.06: x
x86_64/amd/zen4	2023.06: x, 2025.06: x	2023.06: x, 2025.06: x	2023.06: N, 2025.06: N	2025.06: x	2025.06: x
x86_64/amd/zen5	2025.06: x	2025.06: x	2025.06: x	2025.06: x	2025.06: N

See https://www.eessi.io/docs/software_layer/gpu_targets/

NVIDIA GPU support – situation today

- 2025.06 contains *some* modules that don't (actually: can't) work on the newest CUDA architectures
  - E.g. OSU-Micro-Benchmarks/7.5-gompi-2024a-CUDA-12.6.0 won't work on CC12.0: CUDA 12.6 simply does not support it!
  - Lmod hooks print an informative error message when these are loaded: `Lmod has detected the following error:  EasyConfigs using CUDA 12.6.0 or older are not supported for (all) requested Compute Capabilities: ['12.0f'].`
- QOL improvement: pushing 60 tarballs through our bucket & ingestion in CVMFS Stratum 0 take long, could be optimized.
- Overall: stable! 🎉

AMD GPU support

Very different challenges

- No mature support in EasyBuild (yet)
  - Not even mature LLVM toolchain support
- ROCm ecosystem changing quickly
- ROCm build system changing quickly (“The Rock” https://github.com/ROCm/TheRock)
- Kernel driver 🎉
- ROCm is redistributable! 🎉

AMD GPU support

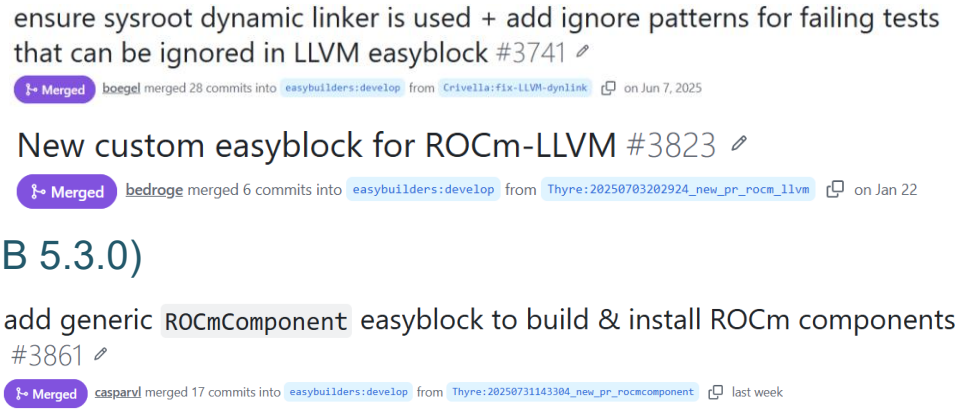
Framework support

- Support amdgcn-capabilities (EB 5.1.2)
- LLVM toolchain support (EB 5.2.0)
- ROCm-based toolchains (EB 5.3.0)

AMD GPU support

EasyBlock support

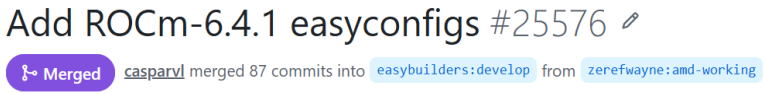
- Improve LLVM sysroot support (EB 5.1.1)
- EasyBlock for ROCm-LLVM (EB 5.2.1)
- Generic EasyBlock for ROCmComponent (EB 5.3.0)
  - Used to build e.g. RCCL, roCFFT, rocBLAS

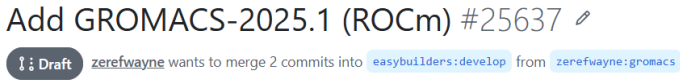


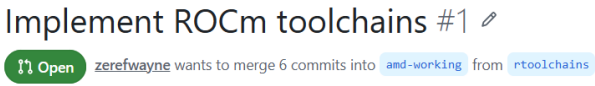
AMD GPU support

EasyConfig support

- Add ROCm-6.4.1 easyconfigs
  - RCCL, rocBLAS, rocFFT, etc… (35 in total)
- GROMACS-2025.1 w/ ROCm 6.4.1
  - TODO: move ROCm support to GROMACS easyblock
  - TODO: move to rfoss level
- Implement ROCm toolchains (rompi, rfbf, rfoss)
  - Very much work in progress







AMD GPU support

First results: comparison of EESSI-extend vs local module

- EESSI-extend:
  - GROMACS-2025.1 w/ ROCm 6.4.1 from PR 25637
  - HIP-based AMD GPU support => PP tasks: offload short-ranged interactions
  - 1x AMD MI210 (SP: 19.49 TFLOPS)
- LUMI local module
  - GROMACS 2025.1 w/ ROCm 6.3.4
  - SYCL-based AMD GPU support => PP tasks: offload short-ranged interactions, update and constrain coordinates. PME tasks: all on GPU.
  - 1 die of AMD MI250X (SP: 47.9/2 = 23.95 TFLOPS/die)

AMD GPU support

First results: comparison of EESSI-extend vs local module

- EESSI-extend:
  - `mpirun -np 1 --map-by slot:PE=14 --report-bindings gmx mdrun -s benchmark.tpr -nb gpu -bonded cpu -pme cpu -update cpu -maxh 0.2 -npme -1 -ntomp 14 -ntmpi 1`
  - Performance: 336 (+/- 1) ns/day
- LUMI local module
  - `gmx_mpi mdrun -s benchmark.tpr -nb gpu -bonded cpu -pme cpu -update cpu -maxh 0.2 -npme -1 -ntomp 14`
  - Performance: 266 (+/- 15) ns/day

AMD GPU support

EESSI @ MI210	336 ns/day
Local module @ MI250X	266 ns/day
EB local module @ A100	500 ns/day

Why the difference in speed?

- Implementation of HIP vs SYCL (vs CUDA)?
  - Even when offloading pme, updates and bonded to GPU with SYCL, it was still slower
- Differences in host? (it's a very light test case – maybe too light for the GPUs)

Preliminary conclusion: performance with ROCm on EESSI looks fine

AMD GPU support – DEMO TIME ☺

Let’s see this in practice

```
# Get GROMACS example
mkdir -p gmx_eum && cd gmx_eum
curl -LJO
https://github.com/victorusu/GROMACS_Benchmark_Suite/raw/1.0.0/HECBioSim/Crambin/benchmark.tpr

# Load EESSI and EESSI-extend (to get access to locally installed AMD GPU stack)
module load EESSI/2025.06
module load EESSI-extend/2025.06-easybuild
module load GROMACS/2025.1-rocm-compilers-19.0.0-ROCm-6.4.1

# Run GROMACS
mpirun -np 1 --map-by slot:PE=14 --report-bindings gmx mdrun -s benchmark.tpr -nb gpu -bonded cpu -pme
cpu -update cpu -maxh 0.2 -npme -1 -ntomp 14 -ntmpi 1
```

AMD GPU support

You're cheating, EESSI-extend just installs locally – how can you claim this tests EESSI?

- EESSI-extend is *also* used by our build bot
  - Guarantees same EasyBuild config (just different install prefix)
  - In partciular: uses EESSI compat layer (through `EASYBUILD_SYSROOT`) & filters same deps
- From our experience 95% of local builds done with EESSI-extend *will* work when trying to deploy in EESSI
  - Only exceptions are typically (micro-)architecture specific issues

AMD GPU support – ~~war stories~~ fun facts

- EasyConfigs PR #25576 contains 35 easyconfigs (and there's more ROCm components)
- A fat build for all easyconfigs in #25576 took just under 40 hours (single gfx target: 12 hours)
- `composable_kernel` takes 6-8h to build for 1 gfx target
  - It *also* has high memory requirement
  - A fat build on jsc-zen3 took 78 hours on 4 cores
- `hipBLASLt` goes OOM with 4 cores / 28 GB
  - Passed on 4 cores for 1 gfx target

- The upside: once it's in, it will all be yours within 10 seconds if you use EESSI 

AMD GPU support: current status

Hot off the press (last weeks efforts):

- ROCm components EasyConfig PR with 35 components merged (to do: rfoss toolchain, move GROMACS to rfoss)
- Add ROCm support in EESSI-extend
- Archdetect support for *overriding* with ROCm architecture (to do: automatic *detection*)

AMD GPU support: future aspirations

- Get a build bot in place on a native AMD GPU system
- Run ROCmValidationSuite to test functionality of ROCm components in EESSI
- Modify EESSI's EasyBuild hooks to add 'gpu' property to AMD GPU builds
- AMD GPU sanity check? (difficult: various ways to do AMD offloading, only some detected by `roc-obj-ls`, which is deprecated)
- Add support in EESSI test suite for testing AMD GPU-enabled builds
- Add ROCm 7.X (via TheRock) in EasyBuild

AMD GPU support – questions to *you*

- What ROCm architectures do you care about?
  - CDNA (MI100) – gfx908
  - CDNA2 (MI210, MI250, MI250X) – gfx90a
  - CDNA3 (MI300A, MI300X, MI325X) – gfx942
  - CNDA4 (MI350X, MI355X) – gfx950
  - RDNA series? (Radeon PRO, Radeon 7XXX, Radeon 9XXX)

AMD GPU support – questions to *you*

For NVIDIA GPUs, we have at least 1 native build per CUDA Compute Capability. For AMD GPUs, we have access to much less hardware.

- Should we try to build natively ?
  - Yes, 1 native build per AMD GPU target (even if this means support will be substantially delayed)
  - Yes, 1 native build for *any* AMD GPU (even if this means support will be substantially delayed)
  - No, get the support in as soon as possible, and rebuild once you have native access
  - No, get the support in as soon as possible, and don’t care about rebuilding once we have native access
  - Other, …

AMD GPU support – bonus question

Why do we not need to symlink the AMD GPU driver?

AMD GPU support – bonus question

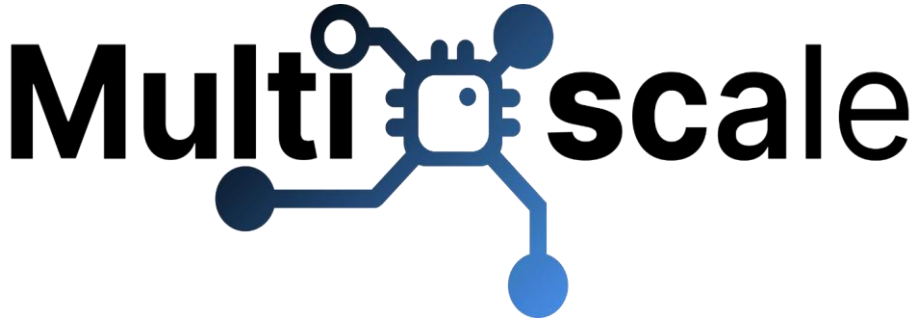
Because these are *kernel drivers*. Like a container the EESSI compatibility layer isolates from the host OS (and anything in it, such as a `libcuda.so`). This is why we need to symlink it, essentially poking a hole in this isoluation. But EESSI *does* use the host *kernel*!

Acknowledgement

NVIDIA and AMD GPU support in EasyBuild / EESSI have been a joint effort. Some notable contributions:

- LLVM toolchain in EasyBuild: Davide Grassano
- Exploratory ROCm work in EasyBuild & EESSI: Inuits (funded by Microsoft)
- Upstreaming ROCm support in EasyBuild: Jan Andre Reuter, Aayush Joglekar
- GROMACS tests: Aayush Joglekar
- CUDA sanity check in EasyBuild: Jasper Grimm, Alan O'Cais.
- CUDA support in EESSI: Alan O'Cais.
- Evaluating CUDA support in EESSI: Inuits (funded by Microsoft)
- And probably many more…

Also thanks to SURF Experimental Technologies Platform for providing AMD MI210 access



Co-funded by the European Union

EuroHPC Joint Undertaking

Web page: multixscale.eu

Twitter: @MultiXscale

LinkedIn: MultiXscale

Facebook: MultiXscale

Youtube channel: 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.