LUMI Update Webinar February 2026

Harvey Richardson (HPE), Samuel Antao (AMD), Kurt Lust (LUMI support team)

February 11, 2026

Agenda

01	Introduction
02	What has been upgraded (HPE)
03	What stacks are available (HPE)
04	Changes to the Cray Programming Environment – CPE (HPE)
05	Changes to the ROCm software environment (AMD)
06	Updates in the LUMI environment supported by LUST (LUST)

History of LUMI updates

Software updates are infrequent:

September 2024

- OS upgrade: SLES 15 SP4 → SLES 15 SP5
- CPE 23.09 → CPE 24.03 (CCE 16.0.1 → 17.0.1 and many other changes)
- ROCm ROCM 5.2.3 (+ newer LUST versions) → ROCM 6.0.3
- No network software stack update

January 2026

- Subject of this webinar

HPE Cray Supercomputer software updates in general

- The Cray Supercomputing environment includes many components:
  - The Cray System Management software components that manage the service
  - The OS (COS) that runs on the login and compute nodes
  - Cray Programming Environment developer software stack (login and compute nodes)
- All were upgraded in January 2026
- COS updates bring in new Linux kernel (SUSE update), libraries, drivers, and updates to other low-level software (network, Lustre) components
- LUMI was upgraded from a base OS of SLES 15 SP5 to SLES 15 SP6
- The upgrades also introduce new hardware support, new features and optimizations but most are not visible to users
- OS and Programming Environment changes were needed to support the newer ROCm versions
- The most visible changes to users are to the Programming Environment and are identifiable by module changes

Cray Programming Environment modules

- The default programming environment has changed
  CPE 24.03 → 25.03
- Many components of the CPE have been updated
  - CCE 17.0.1 → 19.0.0
  - gcc-native 13.2 → 14.2
  - Cray-MPICH 8.1.29 → 8.1.32 (and 9.0.0, see later)
  - Libfabric 1.15 → 1.22
  - Cray-libsci 24.3.0 → 25.03.0
  - perftools-base 24.03.0 → 25.03.0
  - cray-R 4.2.1 → 4.4.0
  - AOCC cpu compilers updated (see later)

- The AMD ROCm stack is updated
  ROCM 6.0.3 (+ newer LUST versions) → ROCM 6.3.4

Status of older software

- Modules do exist for older software versions: cpe/23.09 and cpe/24.03
- Includes older ROCM modules (use module spider rocm) etc.

- You can use the cpe module to change versions of the corresponding modules:

```
> module av cpe
---------------------------- HPE-Cray PE modules -----------------------------
   cpe-cuda/25.09    cpe/23.09    cpe/24.03    cpe/25.03 (D)  cpe/25.09

  Where:
   D:  Default Module

> module load cpe/24.03

The following have been reloaded with a version change:
  1) PrgEnv-cray/8.6.0 => PrgEnv-cray/8.5.0         5) cray-mpich/8.1.32 => cray-mpich/8.1.29
  2) cce/19.0.0 => cce/17.0.1                       6) craype/2.7.34 => craype/2.7.31.11
  3) cray-dsmml/0.3.1 => cray-dsmml/0.3.0           7) perftools-base/25.03.0 => perftools-base/24.03.0
  4) cray-libsci/25.03.0 => cray-libsci/24.03.0

> export LD_LIBRARY_PATH=${CRAY_LD_LIBRARY_PATH}:$LD_LIBRARY_PATH
```

Needed because you are using non-default modules

- The old CPE versions are not officially supported, tread carefully!

MPI updates

- The pre-upgrade cray-mpich module and the new module (8.1.32) are based on ANL MPICH 3.4a2
  - This implements the MPI 3.1 standard
- The new cray-mpich (9.0.0) is based on Cray ANL MPICH 4.0.2
  - This implements the MPI 4 standard
  - Brings goodness such as large Count and displacement support and partitioned communication
  - Version 9.x will be the only supported release for the next CPE versions

- If you want to try this, better to try cray-mpich/9.0.1 (based on ANL MPICH 4.1.2)

  If you have the PrgEnv-cray environment loaded…

```
module load cce/20.0.0
module load cray-mpich/9.0.1
export LD_LIBRARY_PATH=${CRAY_LD_LIBRARY_PATH}:$LD_LIBRARY_PATH
cc …
export MPICH_VERSION_DISPLAY=1 # Display MPI version at runtime, just to be sure
srun …
```

Needed because you are using non-default modules

CPE 25.09

- Loading the latest CPE version can mean changing another module first…

- So, for the PrgEnv-cray environment…

```
% module load cce/20.0.0
The following have been reloaded with a version change:
1) cce/19.0.0 => cce/20.0.0
2) cray-libsci/25.03.0 => cray-libsci/25.09.0
3) cray-mpich/8.1.32 => cray-mpich/9.0.1

% module load cpe/25.09
The following have been reloaded with a version change:
1) craype/2.7.34 => craype/2.7.35
2) perftools-base/25.03.0 => perftools-base/25.09.0
```

```
% export LD_LIBRARY_PATH=${CRAY_LD_LIBRARY_PATH}:$LD_LIBRARY_PATH
```

CPE 25.09…

- After loading cpe/25.09

```
> module list
Currently Loaded Modules:
1) craype-x86-rome                          9) PrgEnv-cray/8.6.0
2) libfabric/1.22.0                        10) cray-dsmml/0.3.1
3) craype-network-ofi                      11) cray-libsci/25.09.0
4) xpmem/2.11.5-1.3_g73ade43320bc          12) cray-mpich/9.0.1
5) ModuleLabel/label                (S)    13) craype/2.7.35
6) lumi-tools/24.05                 (S)    14) perftools-base/25.09.0
7) init-lumi/0.2                    (S)    15) cpe/25.09
8) cce/20.0.0
```

- For the other programming environments just load them, e.g…

```
module swap PrgEnv-cray PrgEnv-amd
module load cpe/25.09
```

CCE developments

- For C and C++, CCE versions track Clang and LLVM versions
  - C fully supports c89 with partial support for c99, c11, and c17.
  - C++ fully supports C++17 with partial support for C++20 and C++23.
- We had CCE 17.0.1 and now have 19.0.0 and 20.0.0 on LUMI

For Fortran…

- CCE 17.0.1 supports Fortran 2018 with one exception relating to TEAMS, it had some Fortran 2023 features
- CCE 19.0.0 supports Fortran 2018 with more Fortran 2023 features
  - Supports Full OpenMP 5.0 with some OpenMP 5.1 and OpenMP 5.2
  - CCE supports full OpenACC 2.0 and partial OpenACC 2.6 for Fortran (OpenACC is not supported for C or C++).

- CCE 20.0.0 supports Fortran 2023 with one exception related to runtime aspects of conditional expressions and arguments (.NIL.)
  - Supports Full OpenMP 5.0 with some OpenMP 5.1, 5.2 and 6.0
  - No change in OpenACC support

CCE developments…

Some things to be aware of…

- CCE 19+ link to internal version of ROCm, to override this use `-rocm-device-lib-path`
- CCE 20 uses a Fortran module format that is incompatible with previous versions

See documentation for detailed description of new features

- CCE 19 (CPE 25.03)
  - https://h41374.www4.hpe.com/docs/25.03/getting_started/CPE-CCE-Overview.html
  - https://h41374.www4.hpe.com/docs/25.03/getting_started/CPE-CCE-Fortran.html
- CCE 20 (CPE 25.09)
  - https://cpe.ext.hpe.com/docs/latest/getting_started/CPE-CCE-Overview.html
  - https://cpe.ext.hpe.com/docs/latest/getting_started/CPE-CCE-Fortran.html

Recent Changes: perftools

- More mature python profiling support:
  - for Sampling:
    $ pat_run python3 app.py
  - for Tracing
    $ pat_run -w python3 app.py
  - an API is available (for example to define regions) (import pat_api)
  - see man pat_python

Tracing

Sampling

Table 1: Calltree View with Callsite Line Numbers

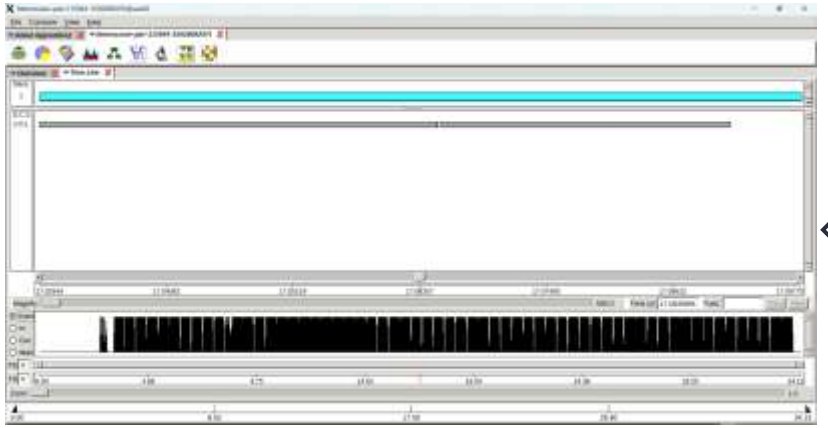
Samp%	Samp	Calltree PE=HIDE
100.0%	634.5	Total
38.5%	244.0	read
20.1%	127.2	dlopen
15.3%	97.0	python.main:heat-p2p.py:line.126
9.3%	59.0	python.iterate:heat-p2p.py:line.83
4.4%	28.0	python.evolve:heat-p2p.py:line.51
		ufunc_generic_fastcall
2.8%	18.0	python.evolve:heat-p2p.py:line.53
2.8%	17.5	ufunc_generic_fastcall
0.0%	0.2	array_true_divide
0.0%	0.2	python.evolve:heat-p2p.py:line.53(exclusive)
1.6%	10.2	python.evolve:heat-p2p.py:line.50
1.3%	8.2	ufunc_generic_fastcall
0.3%	2.0	array_assign_subscript
0.4%	2.0	python.evolve:heat-p2p.py:line.55
		array_assign_subscript
6.0%	38.0	python.iterate:heat-p2p.py:line.82
4.1%	26.0	python.exchange:heat-p2p.py:line.74
4.0%	25.5	MPI_Sendrecv
0.0%	0.2	MPI_Type_get_extent
0.0%	0.2	python.exchange:heat-p2p.py:line.74(exclusive)

Table 1: Calltree View with Callsite Line Numbers

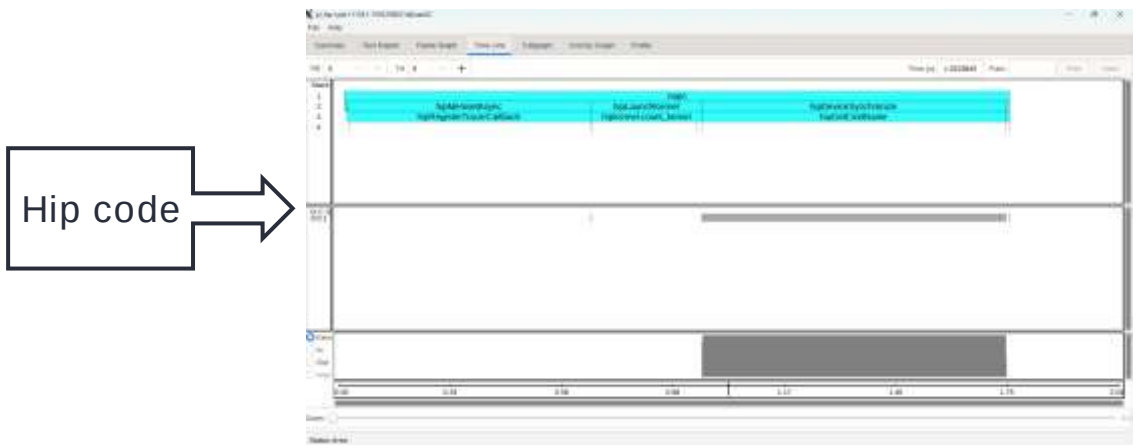
Time%	Time	Calls	Calltree PE=HIDE
100.0%	1.504430	--	Total
45.1%	0.678178	202.0	=unknownFunction=
42.7%	0.642537	1.0	main:python.c:line.14
12.2%	0.183708	--	[R]__main__.exchange:_pat_base.py:line.82
			[R]__main__.evolve:_pat_base.py:line.82
			[R]__main__.iterate:_pat_base.py:line.82
			[R]__main__.main:_pat_base.py:line.83
8.7%	0.130284	--	python.main:heat-p2p_tracing.py:line.130
			[R]__main__.exchange:_pat_base.py:line.82
			[R]__main__.evolve:_pat_base.py:line.82
			[R]__main__.iterate:_pat_base.py:line.82
			[R]__main__.main:_pat_base.py:line.83
			python.iterate:heat-p2p_tracing.py:line.86
8.2%	0.123446	--	[R]__main__.exchange:_pat_base.py:line.82
			[R]__main__.evolve:_pat_base.py:line.82
			[R]__main__.iterate:_pat_base.py:line.82
			[R]__main__.main:_pat_base.py:line.83
5.7%	0.085323	--	python.exchange:heat-p2p_tracing.py:line.77
5.7%	0.085300	200.0	MPI_Sendrecv
0.0%	0.000023	300.0	MPI_Type_get_extent

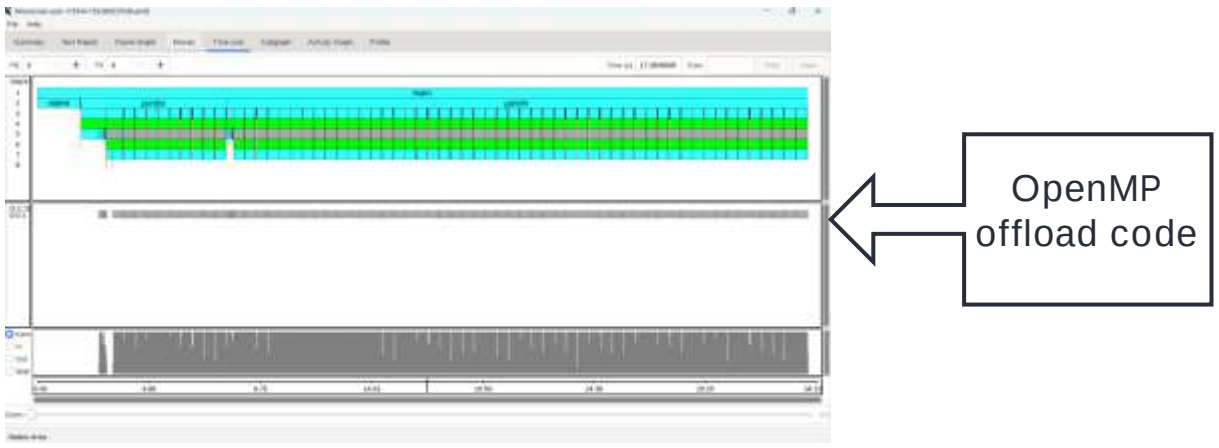
Perftools GPU timeline

- Apprendice 2



- Apprendice 3: new version with more functionality (will replace Apprendice 2)
  - https://cpe.ext.hpe.com/docs/25.03/performance-tools/apprentice3/apprentice3.html





GPU binding in SLURM

- Request the number of GPUs per node with sbatch flag `--gpus-per-node=`, for example in the submission script

```
#SBATCH --gpus-per-node=8
```

- Access the “closest” GPU and enable fast GPU to GPU intro-communications with the srun options:

```
--gpu-bind=map:4,5,2,3,6,7,0,1 --gres-flags=allow-task-sharing
```

  - The GPU bind will set the proper GPU per each rank via ROCR_VISIBLE_DEVICES environment variable

Check sbatch and srun man pages for more info

AMD modules

- AMD compilers are updated:
  - Module amd/6.3.4 (GPU): based on LLVM 18.0.0
    - flang-classic version 18.0.0
    - (AMD to provide more details on new flang compiler)
  - Module aocc/5.0.0 (CPU only): based on LLVM 17.0.6

Where to get information on new releases

- HPE Cray Programming Environment website:

  https://cpe.ext.hpe.com/docs/latest/index.html

- CPE 25.03 guides:

  https://cpe.ext.hpe.com/docs/25.03/index.html

ROCm Updates (AMD)

((separate presentation)

LUST Software Updates

(separate presentation)