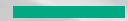




Hewlett Packard
Enterprise

User Module Environment



Kaylie Anderson, Software Engineer

May 7, 2023

Topics

CPE Module Environment High-level Overview

New TCL Module and Lmod Features

What's Next



CPE Module Environment High-level Overview

CPE Module Environments

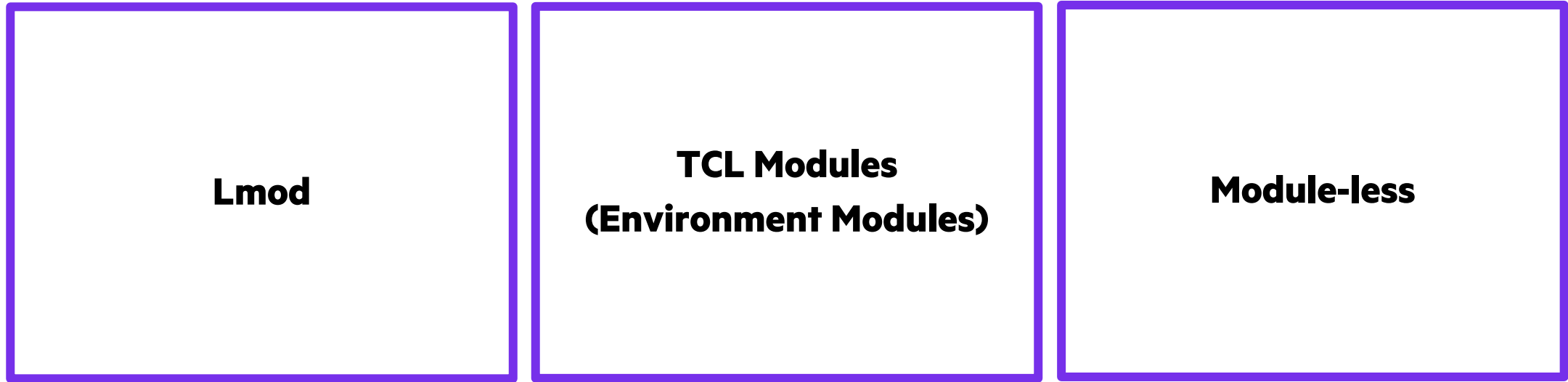


CPE Module Environment High-level Overview

Purpose of modules on a System:

- A module environment provides a dynamic mechanism for modifying a user's environment.
- Modules should provide an intuitive user interface for configuring a viable user environment based on the users desired software tools.

CPE Module environments:



CPE Module Environment High-level Overview

Factors for choosing the best module environment for your system:

- What are users used to?
- Level of user control over the environment is desired?
- Do users want to be guided?

	User Control	User Assist
Lmod	*	* * *
TCL Module	* *	* *
Module-less	* * *	*

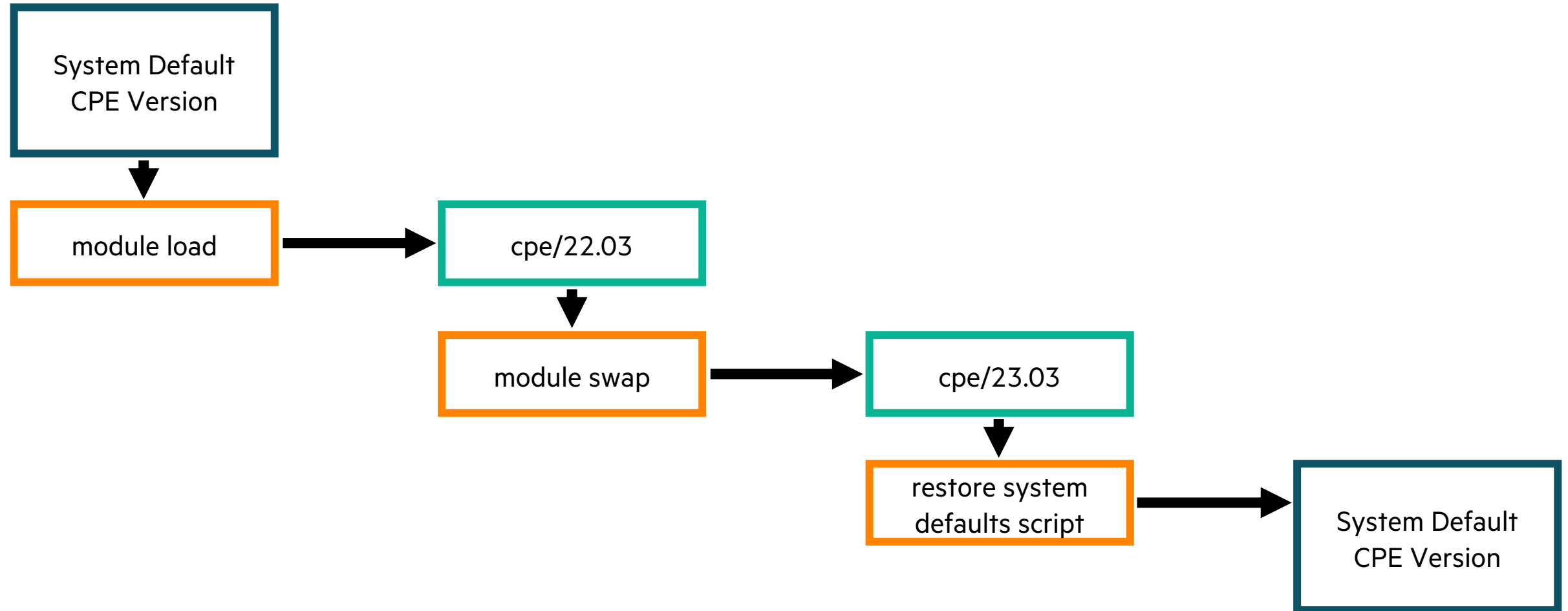
New TCL Module and Lmod Features

CPE's Extended Module Features



CPE Module

CPE module file provides users a time machine mechanism to switch between releases.



Lmod Custom Dynamic Hierarchy

Lmod Custom Dynamic Hierarchy provides users the ability to combined module hierarchies.

User Network Module Path

/home/suse/modulefiles/net/...

User created network module

ofi/1.0

ucx/1.0

usermod

usermod

CPE Network Module Path

/opt/cray/pe/lmod/modulefiles/net/...

CPE network modules

ofi/1.0

ucx/1.0

cray-openshmemx

cray-ucx



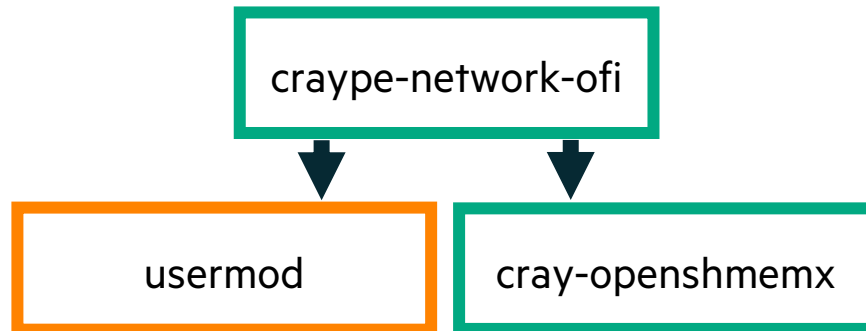
Lmod Custom Dynamic Hierarchy

Create and export environment variables for desired networks

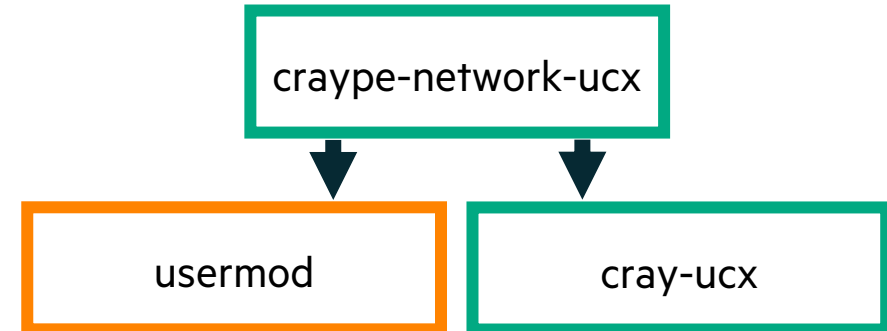
```
LMOD_CUSTOM_NETWORK_OFI_1_0_PREFIX = /home/suse/modulefiles/net/of/1.0
```

```
LMOD_CUSTOM_NETWORK_UCX_1_0_PREFIX = /home/suse/modulefiles/net/ucx/1.0
```

“module avail” with network ofi loaded



“module avail” with network ucx loaded





What's Next ?



What's Next

- Rethink TCL Modules
 - What do our customers want from TCL Modules?
- Deeper Spack integration and support
- Work iteratively with customers to create intuitive and flexible module flows
 - What works?
 - Symmetric module flows such as:
environment -> compiler -> toolkit
 - What doesn't?
 - Lmod mixed modules

AMD MODULE FLOW (PROPOSED STATE)

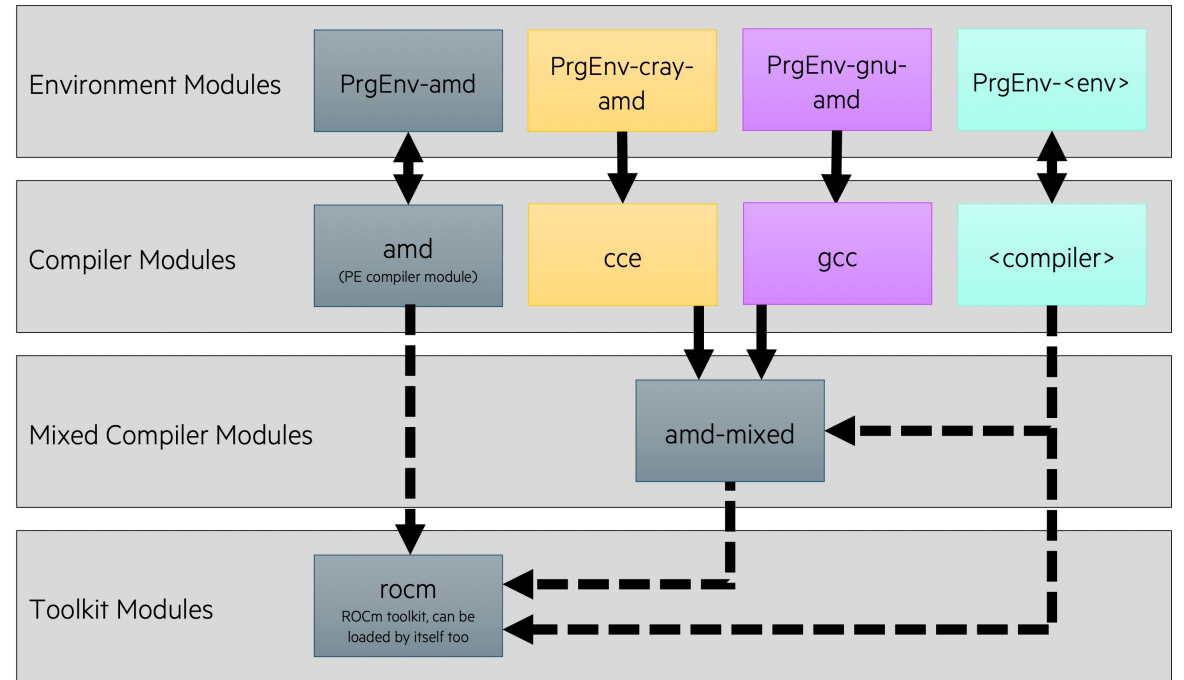


Image: A Lmod module flow designed iteratively with customers this year

Thank you

Kaylie Anderson
kaylie.anderson@hpe.com



New TCL Module and Lmod Features: CPE Module

Lmod Example

```
suse@sles-sp3-86x-build-cug:~> module load cpe/22.03
```

The following have been reloaded with a version change:

- 1) PrgEnv-cray/8.3.3 => PrgEnv-cray/8.3.2.
- 2) cce/15.0.1 => cce/13.0.2
- 3) cray-libsci/23.02.1.1 => cray-libsci/21.08.1.2.
- 4) cray-mpich/8.1.25 => cray-mpich/8.1.14
- 5) craype/2.7.20 => craype/2.7.14
- 6) perftools-base/23.03.0 => perftools-base/21.12.0

```
suse@sles-sp3-86x-build-cug:~> module swap cpe/22.03 cpe/23.03
```

Unloading the cpe module is insufficient to restore the system defaults.

Please run 'source /opt/cray/pe/cpe/22.03/restore_lmod_system_defaults.[csh|sh]'.

The following have been reloaded with a version change:

- 1) PrgEnv-cray/8.3.2 => PrgEnv-cray/8.3.3
- 2) cce/13.0.2 => cce/15.0.1
- 3) cpe/22.03 => cpe/23.03
- 4) cray-libsci/21.08.1.2 => cray-libsci/23.02.1.1
- 5) cray-mpich/8.1.14 => cray-mpich/8.1.25
- 6) craype/2.7.14 => craype/2.7.20
- 7) perftools-base/21.12.0 => perftools-base/23.03.0

```
suse@sles-sp3-86x-build-cug:~> module unload cpe/23.03
```

Unloading the cpe module is insufficient to restore the system defaults.

Please run 'source /opt/cray/pe/cpe/23.03/restore_lmod_system_defaults.[csh|sh]'.

```
suse@sles-sp3-86x-build-cug:~> source /opt/cray/pe/cpe/23.03/restore_lmod_system_defaults.sh
```

Switching PrgEnv-cray. Switching cce. Switching cray-libsci.
Switching cray-mpich. Switching craype. Switching perftools-base.

TCL Module Example

```
suse@sles-sp3-86x-build-cug:~> module load cpe/22.03
```

Switching to PrgEnv-cray/8.3.2. Switching to cce/13.0.2.
Switching to cray-libsci/21.08.1.2. Switching to cray-mpich/8.1.14.
Switching to craype/2.7.14. Switching to perftools-base/21.12.0.

```
suse@sles-sp3-86x-build-cug:~> module swap cpe/22.03 cpe/23.03
```

Switching to PrgEnv-cray/8.3.3. Switching to cce/15.0.1.
Switching to cray-libsci/23.02.1.1. Switching to cray-mpich/8.1.25.
Switching to craype/2.7.20. Switching to perftools-base/23.03.0.

```
suse@sles-sp3-86x-build-cug:~> module unload cpe/23.03.
```

Unloading the cpe module is insufficient to restore the system defaults.

Please run 'source /opt/cray/pe/cpe/23.03/restore_system_defaults.sh'.

```
suse@sles-sp3-86x-build-cug:~> source /opt/cray/pe/cpe/23.03/restore_system_defaults.sh
```

Switching PrgEnv-cray. Switching cce.
Switching cray-libsci. Switching cray-mpich.
Switching craype. Switching perftools-base.

Disclaimer: Featured code was produced using a limited package set. Code text is unaltered, however; spacing and fronts were modified to improve readability.

New TCL Module and Lmod Features: Lmod Custom Dynamic Hierarchy

Command Line Example

```
suse@sles-sp3-86x-build-cug:~> ls /home/suse/modulefiles/net
ofi ucx
suse@sles-sp3-86x-build-cug:~> ls /home/suse/modulefiles/net/of1/1.0/
usermod
suse@sles-sp3-86x-build-cug:~> ls /home/suse/modulefiles/net/ucx/1.0/
usermod

suse@sles-sp3-86x-build-cug:~> ml
Currently Loaded Modules:
 1) craype-network-ofi  2) cray-dsmml/0.2.2  3) cray-openshmemx/11.5.8

suse@sles-sp3-86x-build-cug:~> export
LMOD_CUSTOM_NETWORK_OFI_1_0_PREFIX=/home/suse/modulefiles/net/of1/1.0

suse@sles-sp3-86x-build-cug:~> ml avail usermod
No module(s) or extension(s) found! < abbreviated Lmod message text >

suse@sles-sp3-86x-build-cug:~> ml craype-network-ofi

suse@sles-sp3-86x-build-cug:~> ml avail usermod
----- /home/suse/modulefiles/net/of1/1.0 -----
 usermod/1.1 < abbreviated Lmod message text >

suse@sles-sp3-86x-build-cug:~> ml usermod
suse@sles-sp3-86x-build-cug:~> ml

Currently Loaded Modules:
 1) cray-dsmml/0.2.2  2) craype-network-ofi  3) cray-openshmemx/11.5.8  4) usermod/1.1

suse@sles-sp3-86x-build-cug:~> ml whatis usermod
usermod/1.1 : My user mod for network OFI
```

Continued ...

```
suse@sles-sp3-86x-build-cug:~> export
LMOD_CUSTOM_NETWORK_UCX_1_0_PREFIX=/home/suse/modulefiles/net/ucx/1.0

suse@sles-sp3-86x-build-cug:~> ml craype-network-ucx

Lmod is automatically replacing "craype-network-ofi" with "craype-network-ucx".

Inactive Modules:  1) cray-openshmemx/11.5.8

Due to MODULEPATH changes, the following have been reloaded:  1) usermod/1.1

suse@sles-sp3-86x-build-cug:~> ml
Currently Loaded Modules:
 1) cray-dsmml/0.2.2  2) cray-ucx/2.7.0-1  3) craype-network-ucx  4) usermod/1.1

Inactive Modules:  1) cray-openshmemx/11.5.8

suse@sles-sp3-86x-build-cug:~> ml whatis usermod
usermod/1.1 : My user mod for network UCX

suse@sles-sp3-86x-build-cug:~> module avail
----- /home/suse/modulefiles/net/ucx/1.0 -----
 usermod/1.1 (L)
----- /opt/cray/pe/lmod/modulefiles/net/ucx/1.0 -----
 cray-ucx/1.14.0  cray-ucx/2.7.0-1 (L,D)
----- /opt/cray/pe/lmod/modulefiles/core -----
 < abbreviated module list >
----- /opt/cray/pe/lmod/modulefiles/craype-targets/default -----
 craype-network-ofi  craype-network-ucx (L)
< abbreviated module list >
< abbreviated Lmod message text >
```

Disclaimer: Featured code was produced using a limited package set. Code text is unaltered, however; spacing, fronts, and messages were modified to improve readability.

Frontier's Programming Environment

John K. Holmen, Oak Ridge National Laboratory

Slides Courtesy of Wael Elwasif

**EXPERIENCE
ORNL**
MEET. EXPLORE. LEARN.

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

Overview



Contributors to Frontier Programming Environment

Vendor-Provided

- Cray Programming Environment (CPE)
 - Includes Cray compiler for C, C++, and Fortran plus GCC compiler. All the Cray profiling, tuning, and debugging tools. OpenMP and Cray MPI optimized for AMD GPU direct.
- AMD ROCm programming environment
 - Includes LLVM compiler to generate optimized code for both the AMD Trento CPU and MI250X GPU.
 - Support: C, C++, and Fortran and have GPU offload support. HIP, a CUDA-like direct GPU programming model (with CUDA to HIP conversion utilities).

Other Sources

- ECP
 - LLVM enhancements: Flang (Fortran front-end), OpenMP, OpenACC
 - Kokkos and RAJA
 - HIP LZ (HIP support for Aurora)
 - MPI, HPCToolkit, PAPI enhancements
 - ...
- ALCF + OLCF
 - Pilot implementation of DPC++/SYCL for Frontier
- OLCF
 - GCC enhancements to better support OpenACC, OpenMP, Fortran on Summit and Frontier

Programming Environment

- Compilers Offered

- Cray PE (C/C++ LLVM-based; Cray Fortran)
- AMD ROCm (LLVM-based)
- GCC

Items in green are also available on Summit

- Programming Languages & Models Supported (in which compilers)

- C, C++, Fortran (all)
- OpenACC (Cray Fortran OpenACC 2.0+ & GCC 2.6 substantially complete, 2.7 planned)
- OpenMP (all) 5.0-5.2 in progress – most priority features complete, details vary
- HIP (Cray, AMD)
- Kokkos/RAJA (all)
- UPC (Cray, GCC)

- Transition Paths

- CUDA: semi-automatic translation to HIP
- CUDA Fortran: HIP kernels called from Fortran (a more portable approach)
 - CUDA Fortran kernels need to be translated to C++/HIP (manual process)
 - Fortran bindings to HIP and ROCm libraries and HIP runtime available through AMD's hipfort project

Programming Tools

Debuggers and Correctness Tools

Tool
<i>System-Level Tools</i>
Arm DDT
Cray CCDB
Cray ATP
STAT
<i>Node-Level Tools</i>
ROCgdb
Cray GDB4HPC

Items in green are also available on Summit

Performance Tools

Tool
<i>System-Level Tools</i>
Arm MAP/Performance Reports
CrayPat/Apprentice2 (Cray)
Reveal (Cray)
TAU
HPCToolkit
Score-P / VAMPIR
<i>Node-Level Tools</i>
gprof
PAPI
ROCprof
ROC-profiler & ROC-tracer libraries

Scientific Libraries and Tools

Functionality	CPU	GPU	Notes
BLAS	Cray LibSci, AMD BLIS, PLASMA	Cray LibSci_ACC, AMD roc/hipBLAS, AMD rocAMD ROCm Tensile, MAGMA	MAGMA and PLASMA are open source software led by the UTK Innovative Computing Laboratory
LAPACK	Cray LibSci, AMD libFlame, PLASMA	Cray LibSci_ACC, AMD roc/hipSolver, MAGMA	
ScaLAPACK	Cray LibSci	ECP SLATE, Cray LibSci_ACC	
Sparse		AMD roc/hipSparse, AMD rocALUTION	
Mixed-precision iterative refinement	Cray IRT, MAGMA	MAGMA	
FFTW or similar	Cray, AMD, ECP FFTX, FFT-ECP	AMD rocFFT, ECP FFTX, FFT-ECP	FFT-ECP focuses on 3D FFTs
PETSc, Trilinos, HYPRE, SUNDIALS, SuperLU			Spack recipes from ECP xSDK

Functionality in **green** is also available on Summit

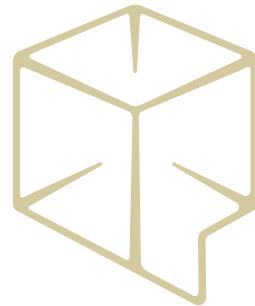


Modules @ Pawsey

Dr. Pascal Jahan Elahi

Pawsey Supercomputing Research Centre

CUG 23



Pawsey Supercomputing Research Centre

- Headquarters located in Perth, Western Australia
- Offers critical support to radioastronomy research around the Square Kilometre Array (SKA).
- Support uses in a large number of different of science domains with a wide variety of workflows



Pawsey

Module Hierarchies

- Pawsey's software stack must deal with multiple compilers & CPU architectures.
- This motivates use of software hierarchies, where paths reflect currently loaded modules and underlying CPU architecture.
- We use Lmod to handle software hierarchies.

CPU architecture

Compiler name

Compiler version

Category

Module name

Module version

Cray Programming Environment & Modules

Cray's customized Lmod installation implements an undocumented custom way of handling hierarchies in substitution for the standard Lmod way.

- Compiler, CPU architecture and MPI library modules scan the shell environment for a specific variable that sets additional module paths.
- Paths are (un)set when the relevant module is (un)loaded. For example, modules for a GCC compiler will look for `LMOD_CUSTOM_COMPILER_GNU_8_0_PREFIX`.
- `/opt/cray/pe/admin-pe/lmod_scripts/lmodHierarchy.lua`.
- We implement this process for Pawsey-built software packages through an auto-loaded module allowing dynamic hierarchy

CPU architecture

Compiler name

Compiler version

Category

Module name

Module version

Architecture & Modules

Setonix hosts a variety of AMD CPU architectures (Zen-2, Zen-3) that serve specific purpose (data mover, compute)

- Module hierarchies are based on the CPU architecture using `lscpu` at user login time to the node.
- CPU architecture is saved in an environment variable for later use in forming search paths for module files.
- We implement this process for Pawsey-built software packages through an auto-loaded module allowing static hierarchy

CPU architecture

Compiler name

Compiler version

Category

Module name

Module version

Users View

- Users must provide not just module name but version. Ensures reproducibility.
- Versions contain not just version number but build information

```
----- /opt/cray/pe/lmod/modulefiles/mpi/gnu/8.0/ofi/1.0/cray-mpich/8.0 -----
cray-hdf5-parallel/1.12.2.1    craype-dl-plugin-ftr/22.06.1.2    craype-dl-plugin-py3/21.04.1    craype-dl-plugin-py3/22.08.1
cray-parallel-netcdf/1.12.3.1    craype-dl-plugin-py3/21.02.1.3    craype-dl-plugin-py3/22.06.1.2    craype-dl-plugin-py3/22.09.1 (D)

----- /software/projects/pawsey0001/pelahi/setonix/modules/zen3/gcc/12.1.0 -----
adios2/2.8.3-r2abbiz (D)    amber/20-rwga7ch    amber/20-3ro75r7 (D)    casacore/3.5.0-mcjdrs3

----- /software/setonix/current/modules/zen3/gcc/12.1.0/astro-applications -----
apr-util/1.6.1    casacore/3.4.0-adios2-openmp    cppunit/1.14.0    pgplot/5.2.2    wcslib/7.3-nocfitsio
apr/1.7.0    casacore/3.4.0-adios2    cppzmq/4.7.1    py-astropy/4.2.1    wcslib/7.3 (D)
casacore/3.2.1    casacore/3.4.0-openmp    idg/0.8.1    py-astropy/5.1 (D)    wcstools/3.9.7
casacore/3.3.0-adios2-openmp    casacore/3.4.0    libzmq/4.3.3    py-emcee/3.1.1    xerces-c/3.2.3-transcoder-gnuiconv
casacore/3.3.0-adios2    casacore/3.5.0-openmp (D)    log4cxx/0.12.1-c++11    py-funcsigs/1.0.2
casacore/3.3.0-openmp    cfitsio/4.0.0    log4cxx/0.12.1-c++17 (D)    py-healpy/1.14.0
casacore/3.3.0    chgcentre/1.6    mcpp/2.7.2    subversion/1.10.6

----- /software/setonix/current/modules/zen3/gcc/12.1.0/bio-applications -----
beast1/1.10.4    beast2/2.6.7    exabayes/1.5.1    examl/3.0.22

----- /software/setonix/current/modules/zen3/gcc/12.1.0/applications -----
cp2k/8.2    gromacs/2020.4    gromacs/2021.4 (D)    lammps/20210929.3    namd/2.14    nektar/5.0.2    nwchem/7.0.2    quantum-espresso/6.8

----- /software/setonix/current/modules/zen3/gcc/12.1.0/libraries -----
adios2/2.7.1-hdf5    hdf5/1.10.7-api-v18    hdf5/1.12.1-parallel-api-v112 (D)    netcdf-cxx/4.2    parmetis/4.0.3
blaspp/2021.04.01    hdf5/1.10.7-api-v110    hpx/1.6.0    netcdf-cxx4/4.3.1    petsc/3.15.5
boost/1.80.0-c++14-python    hdf5/1.10.7-parallel-api-v18    kokkos-amd-gfx90a/3.4.01-hpx    netcdf-fortran/4.5.3    plasma/20.9.20
boost/1.80.0-c++98-python (D)    hdf5/1.10.7-parallel-api-v110    kokkos/3.4.01-hpx    netlib-lapack/3.9.1    plumed/2.6.6
charmmp/6.10.2    hdf5/1.12.1-api-v18    kokkos/3.4.01-openmp    netlib-scalapack/2.1.0    plumed/2.7.2
eigen/3.4.0    hdf5/1.12.1-api-v110    kokkos/3.4.01-rocm    openblas/0.3.15    plumed/2.8.2 (D)
fftw/2.1.5    hdf5/1.12.1-api-v112    kokkos/3.7.01-rocm (D)    opencl/3.4.12    scotch/6.1.1
fftw/3.3.9 (D)    hdf5/1.12.1-parallel-api-v18    metis/5.1.0    opencl/4.5.2 (D)    slate/2021.05.02
gsl/2.6    hdf5/1.12.1-parallel-api-v110    netcdf-c/4.8.1    parallel-netcdf/1.12.2    trilinos/13.0.1
```