

# LUMI

A white wolf is the central focus, standing in a futuristic, blue-toned digital environment. The background is filled with vertical data streams, glowing particles, and a grid-like structure, creating a high-tech, cybernetic atmosphere. The wolf is looking slightly to the right of the viewer.

## Modules on LUMI

**Kurt Lust**  
LUMI User Support Team (LUST)  
University of Antwerp

May 2024

# Module environments

- Modules are used on HPC systems to enable users to create custom environments and select between multiple versions of applications
  - Applications on HPC systems are installed in non-standard places
  - Can set environment variables for the application also
- 3 systems in use
  - Original module tool written in C with modules in Tcl, development halted
  - New implementation in Tcl with many new features, developed at INRIA
    - Not supported by HPE Cray
  - Lmod, an implementation in Lua with native module files in Lua but support for most Tcl module files
- We chose Lmod for LUMI

# Exploring modules with Lmod

- Contrary to some other module systems, not all modules are immediately available for loading
  - **Installed modules**: All modules on the system that can be loaded one way or another
  - **Available modules**: Can be loaded without first loading another module
- Examples in the HPE Cray PE:
  - `cray-mpich` requires a compiler module and network target module first
  - Many of the performance monitoring tools require `perftools-base` first
  - `cray-fftw` only becomes available when a processor target module is loaded
- Tools
  - `module avail` searches in the available modules
  - `module spider` and `module keyword` search in the installed modules
    - But with some restrictions on LUMI

# Benefits of a hierarchy

- When well designed, you get some protection from loading modules that do not work together well
  - Only partially implemented on LUMI
- When “swapping” a module that makes other modules available with a different one, Lmod will try to look for equivalent modules in the new hierarchy
  - Example: Try `module load PrgEnv-gnu` in the default login environment and see what happens

# module load PrgEnv-gnu

```
kulust@uan03.lumi.csc - ~
kulust@uan03.lumi.csc - ~ (ssh)

* | - spack: If you are an experienced Spack user and know how it |
* | works. Spack is offered as-is. We do not do development or |
** | bug fixing in Spack but do offer a configuration compatible |
** | with the Cray PE. |
** `-----***-----***-----***-----***-----`

Did you know?
*****
LUMI has no mpirun or mpiexec command. MPI programs are started using the
Slurm srun command.

[lumi][kulust@uan03-1000 ~]$ module load PrgEnv-gnu

Lmod is automatically replacing "cce/17.0.1" with "gcc-native/13.2".

Lmod is automatically replacing "PrgEnv-cray/8.5.0" with "PrgEnv-gnu/8.5.0".

Due to MODULEPATH changes, the following have been reloaded:
  1) cray-libsci/24.03.0      2) cray-mpich/8.1.29
https://cray-mpich/8.1.29 [kulust@uan03-1001 ~]$
```

# Module names and families

- In Lmod you cannot have two modules with the same name loaded together
  - On LUMI, when loading a new module the other one with the same name will be automatically unloaded
  - Automatic protection from conflicts
- Extension: **family concept**: No two modules of the same family can be loaded together
  - E.g., make compilers member of the family “compiler”
  - On LUMI, the conflicting module of the same family will be unloaded automatically



# Extensions

- It would not make sense to have a separate module for each of the hundreds of R packages or tens of Python packages a software stack may contain.
  - Would actually also create a performance problem due to excess metadata access and long PATH variables
  - Bundle related packages in a single module
- Lmod solution: A module can define a list of [extensions](#), basically other packages provided by the module.
  - And the regular commands can be used to search for these
  - Unfortunately not used in the HPE Cray PE `cray-python` and `cray-R` modules

# module spider

- `module spider` : Long list of all installed software with short description
  - Will also look into modules for “extensions” and show those also, marked with an “E”
  - By default only in the main software stacks on LUMI
- `module spider FFTW` : Look for the FFTW libraries on the system
- `module spider cray-fftw/3.3.10.7` : Look for this specific version
  - But this immediately shows the problems with the HPE Cray PE
    - Some of the lines don't make much sense (see later)
    - Some options are missing also



## module spider (command) (1)

```
kulust@uan02.lumi.csc - ~
kulust@uan02.lumi.csc - ~ (ssh)

 *|  bug fixing in Spack but do offer a configuration compatible |
** |  with the Cray PE.                                         |
** `-----***-----**-----*-----*-----*-----*-----`

Did you know?
*****
The behaviour of many modules of the Cray Programming Environment depends
on the target modules that are loaded. All compilers on LUMI except for the
gcc implementation provided by the operating system support proper
optimizations for the zen2 and zen3 processors. The zen3 architectures is
the core architecture of the AMD Epyc Milan CPUs in the LUMI-C compute nodes
and the AMD EPYC Trento CPUs in the LUMI-G compute nodes. (Except for
gcc 10.3.0 that still lacks full zen3 support.) You may get get a little
extra performance by optimising for zen3 specifically. The CrayEnv environment
will automatically load the best target CPU, network and accelerator target
modules for each node. You can always restore them by reloading the
CrayEnv module (and there is no need to unload first).

In a job script, the environment is copied from the login nodes, so you'll
still have the Rome target modules. Here also reloading the CrayEnv module
will set the ones for the compute node you're running on.

[lumi][kulust@uan02-1000 ~]$ module spider
```

## module spider (command) (2)

```
kurtlust@uan06.lumi.csc - ~
kurtlust@uan06.lumi.csc - ~ (ssh)

-----
The following is a list of the modules and extensions currently available:
-----

ARMForge: ARMForge/22.0.1
  Arm Forge debugging and profiling tools

Autoconf: Autoconf/2.71 (E), Autoconf/2.72 (E)

Autoconf-archive: Autoconf-archive/2022.02.11 (E), ...

Automake: Automake/1.16.5 (E)

Bison: Bison/3.8.2 (E)

Blosc: Blosc/1.21.1-cpeAMD-22.08, Blosc/1.21.1-cpeAOCC-22.08, Blosc/1.21.1-cpeCray-22.08, ...
  Blosc is an extremely fast, multi-threaded, meta-compressor library

Boost: Boost/1.79.0-cpeAMD-22.08, Boost/1.79.0-cpeAOCC-22.08, Boost/1.79.0-cpeCray-22.08, ...
  Boost provides free peer-reviewed portable C++ source libraries.

Brotli: Brotli/1.0.9-cpeAMD-22.08, Brotli/1.0.9-cpeAMD-22.12, Brotli/1.0.9-cpeAMD-23.09, ...

lines 1-22
```

## module spider (command) (3)

```
kurtlust@uan06.lumi.csc - ~
kurtlust@uan06.lumi.csc - ~ (ssh)

zlib: zlib/1.2.12-cpeAMD-22.08, zlib/1.2.12-cpeAMD-22.12, zlib/1.2.12-cpeAOCC-22.08, ...
Free lossless data-compression library, not covered by any patents.

zstd: zstd/1.5.2-cpeAMD-22.08, zstd/1.5.2-cpeAMD-22.12, zstd/1.5.2-cpeAOCC-22.08, ...

Names marked by a trailing (E) are extensions provided by another module.

-----

To learn more about a package execute:

$ module spider Foo

where "Foo" is the name of a module.

To find detailed information about a particular package you
must specify the version if there is more than one version:

$ module spider Foo/11.1

-----

[lumi][kurtlust@uan06-1002 ~]$
```

```
kurtlust@uan06.lumi.csc - ~
kurtlust@uan06.lumi.csc - ~ (ssh)
[lumi][kurtlust@uan06-1004 ~]$ module spider FFTW

-----
cray-fftw:
-----

Versions:
  cray-fftw/3.3.8.13
  cray-fftw/3.3.10.1
  cray-fftw/3.3.10.3
  cray-fftw/3.3.10.5
  cray-fftw/3.3.10.6
  cray-fftw/3.3.10.7

-----

For detailed information about a specific "cray-fftw" package (including how to load the modules) use the
module's full name.
Note that names that have a trailing (E) are extensions provided by other modules.
For example:

  $ module spider cray-fftw/3.3.10.7

-----
[lumi][kurtlust@uan06-1005 ~]$
```

```
kulust@uan02.lumi.csc - ~
kulust@uan02.lumi.csc - ~ (ssh)

-----
cray-fftw: cray-fftw/3.3.10.7
-----

You will need to load all module(s) on any one of the lines below before the "cray-fftw/3.3.10.7" module is available to load.

LUMI/22.12 partition/C craype-x86-genoa
LUMI/22.12 partition/C craype-x86-milan
LUMI/22.12 partition/C craype-x86-milan-x
LUMI/22.12 partition/C craype-x86-rome
LUMI/22.12 partition/C craype-x86-spr
LUMI/22.12 partition/C craype-x86-trento
LUMI/22.12 partition/G craype-x86-genoa
LUMI/22.12 partition/G craype-x86-milan
LUMI/22.12 partition/G craype-x86-milan-x
LUMI/22.12 partition/G craype-x86-rome
LUMI/22.12 partition/G craype-x86-spr
LUMI/22.12 partition/G craype-x86-trento
LUMI/22.12 partition/L craype-x86-genoa
LUMI/22.12 partition/L craype-x86-milan
```

lines 1-21

```
kulust@uan02.lumi.csc - ~
kulust@uan02.lumi.csc - ~ (ssh)

LUMI/22.12 partition/C craype-x86-genoa
LUMI/22.12 partition/C craype-x86-milan
LUMI/22.12 partition/C craype-x86-milan-x
LUMI/22.12 partition/C craype-x86-rome
LUMI/22.12 partition/C craype-x86-spr
LUMI/22.12 partition/C craype-x86-trento
LUMI/22.12 partition/G craype-x86-genoa
LUMI/22.12 partition/G craype-x86-milan
LUMI/22.12 partition/G craype-x86-milan-x
LUMI/22.12 partition/G craype-x86-rome
LUMI/22.12 partition/G craype-x86-spr
LUMI/22.12 partition/G craype-x86-trento
LUMI/22.12 partition/L craype-x86-genoa
LUMI/22.12 partition/L craype-x86-milan
LUMI/22.12 partition/L craype-x86-milan-x
LUMI/22.12 partition/L craype-x86-rome
LUMI/22.12 partition/L craype-x86-spr
LUMI/22.12 partition/L craype-x86-trento

Help:
  Documentation: `man intro_fftw3`

[lumi][kulust@uan02-1004 ~]$
```

# module spider for a regular package

L U M I

- `module spider gnuplot` : Shows all versions of gnuplot on the system
- `module spider gnuplot/5.4.10-cpeGNU-24.03` : Shows help information for the specific module, including what should be done to make the module available



```
kurtlust@uan06.lumi.csc - ~
kurtlust@uan06.lumi.csc - ~ (ssh)

-----
gnuplot:
-----

Description:
  Gnuplot is a portable command-line driven graphing utility

Versions:
  gnuplot/5.4.3-cpeAMD-22.08
  gnuplot/5.4.3-cpeAOCC-22.08
  gnuplot/5.4.3-cpeCray-22.08
  gnuplot/5.4.3-cpeGNU-22.08
  gnuplot/5.4.6-cpeAMD-22.12
  gnuplot/5.4.6-cpeAOCC-22.12
  gnuplot/5.4.6-cpeCray-22.12
  gnuplot/5.4.6-cpeCray-23.03
  gnuplot/5.4.6-cpeGNU-22.12
  gnuplot/5.4.8-cpeAMD-23.12
  gnuplot/5.4.8-cpeAOCC-23.09
  gnuplot/5.4.8-cpeAOCC-23.12
  gnuplot/5.4.8-cpeCray-23.12
  gnuplot/5.4.8-cpeGNU-23.09
```

lines 1-22

## module spider gnuplot (2)

```
kurtlust@uan06.lumi.csc - ~
kurtlust@uan06.lumi.csc - ~ (ssh)

gnuplot/5.4.6-cpeCray-22.12
gnuplot/5.4.6-cpeCray-23.03
gnuplot/5.4.6-cpeGNU-22.12
gnuplot/5.4.8-cpeAMD-23.12
gnuplot/5.4.8-cpeAOCC-23.09
gnuplot/5.4.8-cpeAOCC-23.12
gnuplot/5.4.8-cpeCray-23.12
gnuplot/5.4.8-cpeGNU-23.09
gnuplot/5.4.8-cpeGNU-23.12
gnuplot/5.4.10-cpeAMD-24.03
gnuplot/5.4.10-cpeAOCC-24.03
gnuplot/5.4.10-cpeCray-24.03
gnuplot/5.4.10-cpeGNU-24.03

-----
For detailed information about a specific "gnuplot" package (including how to load the modules) use the
module's full name.
Note that names that have a trailing (E) are extensions provided by other modules.
For example:

    $ module spider gnuplot/5.4.10-cpeGNU-24.03

-----
[lumi][kurtlust@uan06-1006 ~]$
```

```
kulust@uan02.lumi.csc - ~
kulust@uan02.lumi.csc - ~ (ssh)

-----
gnuplot: gnuplot/5.4.10-cpeGNU-24.03
-----

Description:
  Gnuplot is a portable command-line driven graphing utility

You will need to load all module(s) on any one of the lines below before the "gnuplot/5.4.10-cpeGNU-24.03" module is available to load.

LUMI/24.03  partition/C
LUMI/24.03  partition/G
LUMI/24.03  partition/L

Help:
  Description
  =====
  Gnuplot is a portable command-line driven graphing utility available for many platforms. The source code is copyrighted but freely distributed (i.e., you don't have to pay for it). It was originally created to allow scientists and students to visualize mathematical functions and data interactively, but has
```

lines 1-21

```
kulust@uan02.lumi.csc - ~
kulust@uan02.lumi.csc - ~ (ssh)

=====
Gnuplot is a portable command-line driven graphing utility available for many
platforms. The source code is copyrighted but freely distributed (i.e., you
don't have to pay for it). It was originally created to allow scientists and
students to visualize mathematical functions and data interactively, but has
grown to support many non-interactive uses such as web scripting. It is also
used as a plotting engine by third-party applications like Octave. Gnuplot has
been supported and under active development since 1986.

This version of GNUplot does not use Qt5 for its GUI, so the GUI is rather
primitive.

More information
=====
- Homepage: http://gnuplot.sourceforge.net/
- Documentation:
  - Web-based documentation: http://gnuplot.sourceforge.net/documentation.html
  - Manual page for gnuplot
- Site contact: LUMI User Support @ https://lumi-supercomputer.eu/user-support/need-help/

[lumi][kulust@uan02-1008 ~]$
```

# module spider for extensions

- No example in the default Cray modules, so examples come from the LUMI software stacks
- `module spider CMake`
- `module spider CMake/3.27.7` : Will tell you which module contains this version of CMake and how to load it

```
kulust@uan02.lumi.csc - ~
kulust@uan02.lumi.csc - ~ (ssh)

-----
CMake:
-----

Versions:
  CMake/3.24.0 (E)
  CMake/3.25.2 (E)
  CMake/3.27.7 (E)
  CMake/3.29.3 (E)
Other possible modules matches:
  cmake  cmake-3.26.3-gcc-7.5.0-3og5f6c  cmake-3.26.3-gcc-7.5.0-cl73x27  rocm-cmake

Names marked by a trailing (E) are extensions provided by another module.

-----

To find other possible module matches execute:

  $ module -r spider '.*CMake.*'

-----

For detailed information about a specific "CMake" package (including how to load the modules) use the mo
lines 1-22
```

```
kulust@uan02.lumi.csc - ~  
kulust@uan02.lumi.csc - ~ (ssh)  
-----  
CMake: CMake/3.29.3 (E)  
-----  
This extension is provided by the following modules. To access the extension you must load one of the  
following modules. Note that any module names in parentheses show the module location in the software hier  
archy.  
  
  buildtools/24.03 (LUMI/24.03 partition/L)  
  buildtools/24.03 (LUMI/24.03 partition/G)  
  buildtools/24.03 (LUMI/24.03 partition/C)  
  buildtools/24.03 (CrayEnv)  
  buildtools/23.12 (LUMI/23.12 partition/L)  
  buildtools/23.12 (LUMI/23.12 partition/G)  
  buildtools/23.12 (LUMI/23.12 partition/C)  
  buildtools/23.12 (CrayEnv)  
  
Names marked by a trailing (E) are extensions provided by another module.  
  
lines 1-20
```



# module keyword

- Searches in the module short description and help message for the keyword.
  - E.g., try  
`module keyword https`
- We do try to put enough information in the modules to make this a suitable additional way to discover software that is already installed on the system

```
kurtlust@uan06.lumi.csc - ~
kurtlust@uan06.lumi.csc - ~ (ssh)

-----

The following modules match your search criteria: "https"

-----

cURL: cURL/7.83.1-cpeAMD-22.08, cURL/7.83.1-cpeAOCC-22.08, cURL/7.83.1-cpeCray-22.08, ...
      Command line tool and library for transferring data with URLs.

wget: wget/1.21.3-cpeAMD-22.08, wget/1.21.3-cpeAMD-22.12, wget/1.21.3-cpeAMD-23.09, ...
      wget - GNU wget, a free software package for retrieving files using HTTP, HTTPS and FTP

-----

To learn more about a package execute:

    $ module spider Foo

where "Foo" is the name of a module.

To find detailed information about a particular package you
must specify the version if there is more than one version:

lines 1-22
```

```
kurtlust@uan06.lumi.csc - ~
kurtlust@uan06.lumi.csc - ~ (ssh)

-----

cURL: cURL/7.83.1-cpeAMD-22.08, cURL/7.83.1-cpeAOCC-22.08, cURL/7.83.1-cpeCray-22.08, ...
  Command line tool and library for transferring data with URLs.

wget: wget/1.21.3-cpeAMD-22.08, wget/1.21.3-cpeAMD-22.12, wget/1.21.3-cpeAMD-23.09, ...
  wget - GNU wget, a free software package for retrieving files using HTTP, HTTPS and FTP

-----

To learn more about a package execute:

  $ module spider Foo

where "Foo" is the name of a module.

To find detailed information about a particular package you
must specify the version if there is more than one version:

  $ module spider Foo/11.1

-----

[lumi][kurtlust@uan06-1007 ~]$
```

# Sticky modules and module purge

- On some systems, you will be taught to avoid `module purge` (which unloads all modules)
- Sticky modules are modules that are not unloaded by `module purge`, but reloaded.
  - They can be force-unloaded with `module --force purge` and `module --force unload`
- Used on LUMI for the software stacks and modules that set the display style of the modules
  - But keep in mind that the modules are reloaded which can have side effects

```
kulust@uan04.lumi.csc - ~
kulust@uan04.lumi.csc - ~ (ssh)

----- EasyBuild managed systemwide software -----
ARMForge/22.0.1          lumi-tools/23.03 (S)   lumi-tools/24.05   (S,L,D)   lumio-ext-tools/1.0.0
Vampir/10.0.0           lumi-tools/23.04 (S)   lumi-vnc/20230110  lumio/1.0.0
Vampir/10.2.1 (D)      lumi-tools/23.11 (S)   lumi-workspaces/0.1

----- HPE-Cray PE modules -----
PrgEnv-amd/8.3.3
PrgEnv-amd/8.4.0
PrgEnv-amd/8.5.0 (D)
PrgEnv-aocc/8.3.3
PrgEnv-aocc/8.4.0
PrgEnv-aocc/8.5.0 (D)
PrgEnv-cray-amd/8.3.3
PrgEnv-cray-amd/8.4.0
PrgEnv-cray-amd/8.5.0 (D)
PrgEnv-cray/8.3.3
PrgEnv-cray/8.4.0
PrgEnv-cray/8.5.0 (L,D)
PrgEnv-gnu-amd/8.3.3
PrgEnv-gnu-amd/8.4.0
PrgEnv-gnu-amd/8.5.0 (D)
lines 1-22
```

```
kulust@uan04.lumi.csc - ~
kulust@uan04.lumi.csc - ~ (ssh)

PrgEnv-gnu/8.3.3
PrgEnv-gnu/8.4.0
PrgEnv-gnu/8.5.0          (D)
PrgEnv-nvhpc/8.5.0
PrgEnv-nvidia/8.5.0
amd-mixed/6.0.3
amd/6.0.3                (5.0.2:5.1.0:5.2.0:5.2.3:5.5.1:5.7.0:6.0.0)
aocc-mixed/3.2.0        (D)
aocc-mixed/4.1.0
aocc/3.2.0              (D)
aocc/4.1.0
atp/3.14.13
atp/3.14.16
atp/3.14.18
atp/3.15.1
atp/3.15.2
atp/3.15.3              (D)
cce-mixed/14.0.2
cce-mixed/15.0.0
cce-mixed/15.0.1
cce-mixed/16.0.1
cce-mixed/17.0.0

lines 23-44
```

```
kulust@uan04.lumi.csc - ~
kulust@uan04.lumi.csc - ~ (ssh)

cce-mixed/17.0.1      (D)
cce/14.0.2
cce/15.0.0
cce/15.0.1
cce/16.0.1
cce/17.0.0
cce/17.0.1          (L,D)
cpe/22.08
cpe/22.12
cpe/23.03
cpe/23.09
cpe/23.12
cpe/24.03          (D)
cray-R/4.1.3.1
cray-R/4.2.1.1
cray-R/4.2.1.2
cray-R/4.3.1
cray-R/4.3.2      (D)
cray-ccdb/4.12.13
cray-ccdb/5.0.1
cray-ccdb/5.0.2
cray-ccdb/5.0.3   (D)

lines 45-66
```

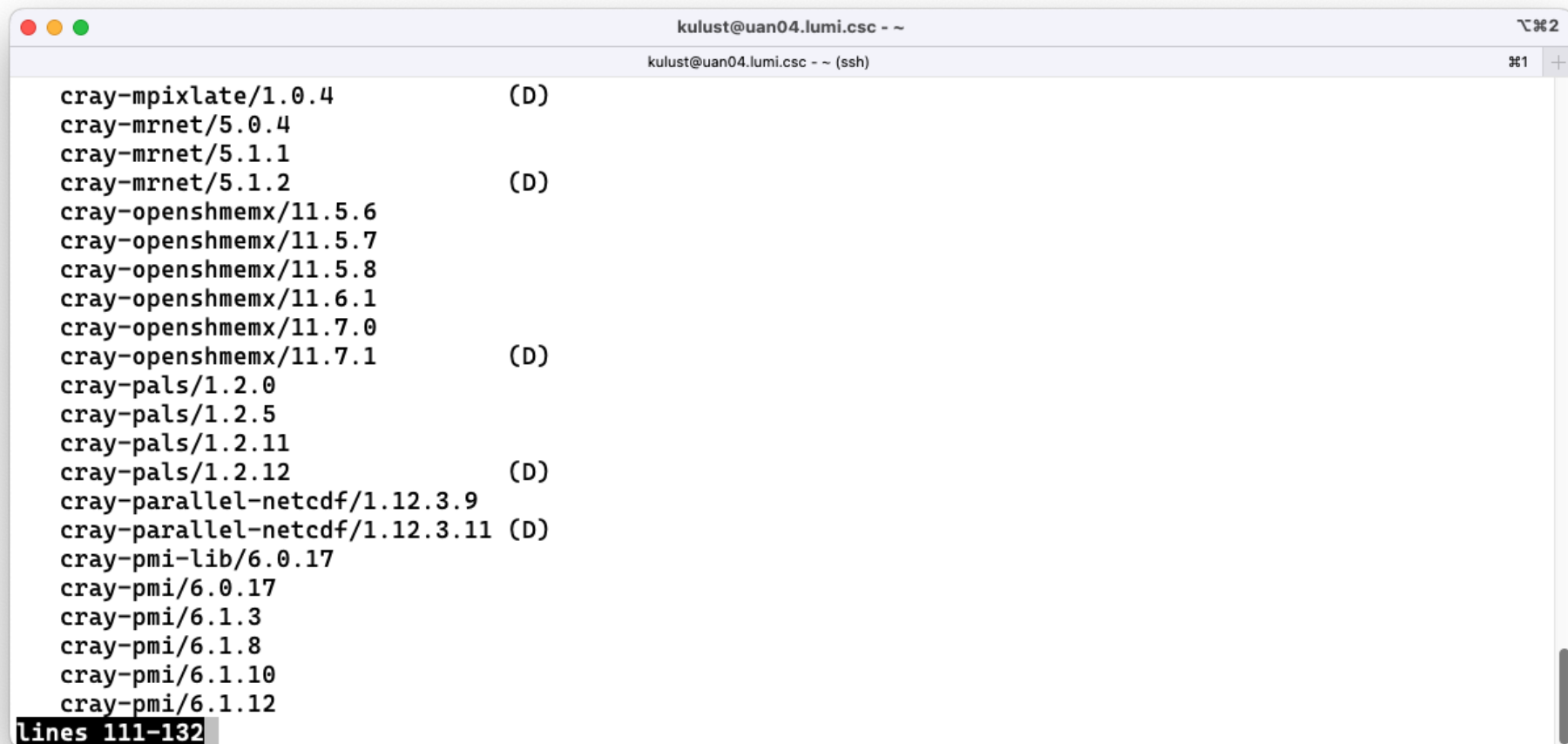


```
kulust@uan04.lumi.csc - ~  
kulust@uan04.lumi.csc - ~ (ssh)  
  
cray-cti/2.15.13  
cray-cti/2.15.14  
cray-cti/2.16.0  
cray-cti/2.17.1  
cray-cti/2.17.2  
cray-cti/2.18.1  
cray-cti/2.18.2  
cray-cti/2.18.3 (D)  
cray-dsmml/0.2.2  
cray-dsmml/0.3.0 (L,D)  
cray-dyninst/12.1.1  
cray-dyninst/12.3.0  
cray-dyninst/12.3.1 (D)  
cray-fftw/3.3.8.13  
cray-fftw/3.3.10.1  
cray-fftw/3.3.10.3  
cray-fftw/3.3.10.5  
cray-fftw/3.3.10.6  
cray-fftw/3.3.10.7 (D)  
cray-hdf5-parallel/1.12.2.9  
cray-hdf5-parallel/1.12.2.11 (D)  
cray-hdf5/1.12.2.9  
lines 67-88
```

```
kulust@uan04.lumi.csc - ~
kulust@uan04.lumi.csc - ~ (ssh)

cray-hdf5/1.12.2.11      (D)
cray-libpals/1.2.0
cray-libpals/1.2.5
cray-libpals/1.2.11
cray-libpals/1.2.12      (D)
cray-libsci/21.08.1.2
cray-libsci/22.08.1.1
cray-libsci/22.12.1.1
cray-libsci/23.02.1.1
cray-libsci/23.09.1.1
cray-libsci/23.12.5
cray-libsci/24.03.0      (L,D)
cray-libsci_acc/22.08.1.1
cray-libsci_acc/22.12.1.1
cray-libsci_acc/23.09.1.1
cray-libsci_acc/23.12.0
cray-libsci_acc/24.03.1  (D)
cray-mpich-abi/8.1.28
cray-mpich-abi/8.1.29    (D)
cray-mpich/8.1.28
cray-mpich/8.1.29       (L,D)
cray-mpixlate/1.0.3

lines 89-110
```

A terminal window titled 'kulust@uan04.lumi.csc - ~' with a subtitle 'kulust@uan04.lumi.csc - ~ (ssh)'. The window displays a list of modules and their status. The list includes: 'cray-mpixlate/1.0.4 (D)', 'cray-mrnet/5.0.4', 'cray-mrnet/5.1.1', 'cray-mrnet/5.1.2 (D)', 'cray-openshmemx/11.5.6', 'cray-openshmemx/11.5.7', 'cray-openshmemx/11.5.8', 'cray-openshmemx/11.6.1', 'cray-openshmemx/11.7.0', 'cray-openshmemx/11.7.1 (D)', 'cray-pals/1.2.0', 'cray-pals/1.2.5', 'cray-pals/1.2.11', 'cray-pals/1.2.12 (D)', 'cray-parallel-netcdf/1.12.3.9', 'cray-parallel-netcdf/1.12.3.11 (D)', 'cray-pmi-lib/6.0.17', 'cray-pmi/6.0.17', 'cray-pmi/6.1.3', 'cray-pmi/6.1.8', 'cray-pmi/6.1.10', and 'cray-pmi/6.1.12'. At the bottom left, a black box highlights the text 'lines 111-132'.

```
kulust@uan04.lumi.csc - ~  
kulust@uan04.lumi.csc - ~ (ssh)  
  
cray-mpixlate/1.0.4          (D)  
cray-mrnet/5.0.4  
cray-mrnet/5.1.1  
cray-mrnet/5.1.2          (D)  
cray-openshmemx/11.5.6  
cray-openshmemx/11.5.7  
cray-openshmemx/11.5.8  
cray-openshmemx/11.6.1  
cray-openshmemx/11.7.0  
cray-openshmemx/11.7.1    (D)  
cray-pals/1.2.0  
cray-pals/1.2.5  
cray-pals/1.2.11  
cray-pals/1.2.12         (D)  
cray-parallel-netcdf/1.12.3.9  
cray-parallel-netcdf/1.12.3.11 (D)  
cray-pmi-lib/6.0.17  
cray-pmi/6.0.17  
cray-pmi/6.1.3  
cray-pmi/6.1.8  
cray-pmi/6.1.10  
cray-pmi/6.1.12  
  
lines 111-132
```

```
kulust@uan04.lumi.csc - ~
kulust@uan04.lumi.csc - ~ (ssh)

cray-pmi/6.1.13
cray-pmi/6.1.14 (D)
cray-python/3.9.12.1
cray-python/3.9.13.1
cray-python/3.10.10
cray-python/3.11.5
cray-python/3.11.7 (D)
cray-stat/4.11.12
cray-stat/4.11.13
cray-stat/4.12.1
cray-stat/4.12.2 (D)
craype/2.7.17
craype/2.7.19
craype/2.7.20
craype/2.7.23
craype/2.7.30
craype/2.7.31.11 (L,D)
craypkg-gen/1.3.25
craypkg-gen/1.3.28
craypkg-gen/1.3.30
craypkg-gen/1.3.31
craypkg-gen/1.3.32 (D)

lines 133-154
```

```
kulust@uan04.lumi.csc - ~
kulust@uan04.lumi.csc - ~ (ssh)

gcc-mixed/11.2.0
gcc-mixed/12.2.0          (D)
gcc-native-mixed/12.3
gcc-native-mixed/13.2   (D)
gcc-native/12.3
gcc-native/13.2         (D)
gcc/10.3.0
gcc/11.2.0
gcc/12.2.0              (D)
gdb4hpc/4.14.2
gdb4hpc/4.14.6
gdb4hpc/4.14.7
gdb4hpc/4.15.1
gdb4hpc/4.16.0
gdb4hpc/4.16.1         (D)
iobuf/2.0.10
papi/6.0.0.15
papi/6.0.0.17
papi/7.0.0.1
papi/7.0.1.1
papi/7.0.1.2
papi/7.1.0.1           (D)
lines 155-176
```

```
kulust@uan04.lumi.csc - ~
kulust@uan04.lumi.csc - ~ (ssh)

perftools-base/22.06.0
perftools-base/22.12.0
perftools-base/23.03.0
perftools-base/23.09.0
perftools-base/23.12.0
perftools-base/24.03.0      (L,D)
perftools-lite-events
perftools-lite-gpu
perftools-lite-hbm
perftools-lite-loops
perftools-lite
perftools-preload
perftools
rocm/6.0.3                 (5.0.2:5.1.0:5.2.0:5.2.3:5.5.1:5.7.0:6.0.0)
sanitizers4hpc/1.0.1
sanitizers4hpc/1.0.4
sanitizers4hpc/1.1.1
sanitizers4hpc/1.1.2      (D)
valgrind4hpc/2.12.10
valgrind4hpc/2.12.11
valgrind4hpc/2.13.1
valgrind4hpc/2.13.2      (D)
```

lines 177-198

```

kulist@uan04.lumi.csc - ~
kulist@uan04.lumi.csc - ~ (ssh)

----- /opt/cray/pe/lmod/lmod/modulefiles/Core -----
lmod  settarg

----- HPE-Cray PE target modules -----

craype-accel-amd-gfx908  craype-arm-grace  craype-hugepages4M  craype-x86-milan-x
craype-accel-amd-gfx90a  craype-hugepages128M  craype-hugepages512M  craype-x86-milan
craype-accel-amd-gfx940  craype-hugepages16M  craype-hugepages64M  craype-x86-rome (L)
craype-accel-amd-gfx942  craype-hugepages1G  craype-hugepages8M  craype-x86-spr-hbm
craype-accel-host  craype-hugepages256M  craype-network-none  craype-x86-spr
craype-accel-nvidia70  craype-hugepages2G  craype-network-ofi (L)  craype-x86-trento
craype-accel-nvidia80  craype-hugepages2M  craype-network-ucx
craype-accel-nvidia90  craype-hugepages32M  craype-x86-genoa

----- Software stacks -----

CrayEnv (S)  LUMI/23.03 (S)  LUMI/24.03 (S,D)  spack/22.08  spack/23.03-2
LUMI/22.08 (S)  LUMI/23.09 (S)  Local-CSC/default (S)  spack/22.08-2  spack/23.09 (D)
LUMI/22.12 (S)  LUMI/23.12 (S)  Local-quantum/default (S)  spack/23.03

----- Modify the module display style -----

ModuleColour/off (S)  ModuleFullSpider/off (S)  ModuleLabel/system (S)
ModuleColour/on (S,D)  ModuleFullSpider/on (S,D)  ModulePowerUser/LUMI (S)

lines 199-220

```



```
kulust@uan04.lumi.csc - ~
kulust@uan04.lumi.csc - ~ (ssh)

ModuleExtensions/hide (S)      ModuleLabel/label      (S,L,D)      ModuleStyle/default
ModuleExtensions/show (S,D)   ModuleLabel/PEhierarchy (S)          ModuleStyle/reset   (D)

----- System initialisation -----
init-lumi/0.2 (S,L)

----- Non-PE HPE-Cray modules -----
dvs/2.15_4.7.82-1.0_23.1__gfba2684e      xpmem/2.8.2-1.0_5.1__g84a27a5.shasta (L)
libfabric/1.15.2.0                      (L)

----- This is a list of module extensions. Use "module --nx avail ..." to not show extensions. -----
rclone (E)      restic (E)      s3cmd (E)

These extensions cannot be loaded directly, use "module spider extension_name" for more information.

Where:
L:      Module is loaded
S:      Module is Sticky, requires --force to unload or purge
Aliases: Aliases exist: foo/1.2.3 (1.2) means that "module load foo/1.2" will load foo/1.2.3
D:      Default Module
E:      Extension that is provided by another module

lines 221-242
```

```
kulust@uan04.lumi.csc - ~
kulust@uan04.lumi.csc - ~ (ssh)

libfabric/1.15.2.0          (L)

----- This is a list of module extensions. Use "module --nx avail ..." to not show extensions. -----
  rclone (E)    restic (E)    s3cmd (E)

These extensions cannot be loaded directly, use "module spider extension_name" for more information.

Where:
L:      Module is loaded
S:      Module is Sticky, requires --force to unload or purge
Aliases: Aliases exist: foo/1.2.3 (1.2) means that "module load foo/1.2" will load foo/1.2.3
D:      Default Module
E:      Extension that is provided by another module

Additional ways to search for software:
* Use "module spider" to find all possible modules and extensions.
* Use "module keyword key1 key2 ..." to search for all possible modules matching any of the "keys".
See the LUMI documentation at https://docs.lumi-supercomputer.eu/runjobs/lumi\_env/Lmod\_modules/ for more information on searching modules.
If then you still miss software, contact LUMI User Support via https://lumi-supercomputer.eu/user-support/need-help/.

[lumi][kulust@uan04-1003 ~]$
```

# Changing how the module list is displayed

L U M I

- You may have noticed that you see descriptive texts in the module, not directories
- This can be changed by loading a module
  - `ModuleLabel/label` : The default view
  - `ModuleLabel/PEhierarchy` : Descriptive texts and unfolded PE hierarchy
  - `ModuleLabel/system` : Module directories
- Turn colour on or off using `ModuleColour/on` or `ModuleColour/off`
- Show or hide the module extensions with `ModuleExtensions/show` or `ModuleExtensions/hide`
- Index all modules with the spider command using `ModuleFullSpider/on`
- Show some hidden modules with `ModulePowerUser/LUMI`
  - This will also show undocumented/unsupported modules!
  - Can use `module --show_hidden avail` instead
- More customisation possible via LMOD environment variables

# Getting help

- `module help` is the command to get help information for available modules
  - Without further arguments: help about the module command
  - We do try to add a bit more help information about what a module provides to the modules than default EasyBuild or Spack installations tend to do.
- Examples (require loading `CrayEnv`):  
`module help cray-mpich`  
`module help cray-python/3.11.7`  
`module help buildtools/24.03`
- `module whatis` can produce a short description  
`module whatis Subversion`  
`module whatis Subversion/1.14.3`

# A note on caching

- Large module system = lots of small module files = Lustre not very happy
  - But Lmod does use caches by default
  - Currently no system cache, only a user cache in `$HOME/.cache/lmod`
- Cache refreshed automatically every 24 hours
  - You'll notice when the `spider` or `available` commands are slow
  - But you may need to clean the cache after creating a new module as on LUMI Lmod does not always detect the change
- Also clear the cache if you notice very strange answers from `module spider`.
  - Looks like the HPE Cray PE sometimes causes cache problems

# A note on other commands

- `module load`, `module unload`, `module list` are fairly standard commands and the basic operation is the same in all module systems
  - Note that `module list` may also show inactive modules: Modules that were loaded at some point but got unloaded when a module closer to the root of the hierarchy got unloaded
- `module swap`:
  - Equivalent to an unload followed by a load
  - For two modules of the same family `module swap` is more efficient as Lmod does not first have to discover the family conflict
  - But it is not essential as LUMI has autoswap enabled

**Questions?**

