

ACCRES SOFTWARE STACK

FROM SETPKGS TO LMOD

- Scientific researchers should focus on the **science** behind the software they use.
- Scientific software can be “challenging” due to the complexity of the dependencies needed to make it work correctly.
- Users often lack the knowledge to handle this themselves efficiently.



“If we knew what we were doing, it wouldn’t be called research.”

- Due to the shared nature an HPC cluster, lots of software is typically provided:
 - Different scientific domains
 - Multiple versions, builds, variants...
 - Built with different compilers and/or libraries
 - Non-standard locations
- Shell environment must be customized to use the desired software:
 - \$PATH
 - \$LD_LIBRARY_PATH
 - Software specific environment variables

Users need to easily search what software is available and get access to it.

Created by Alan Tackett and proudly serving the ACCRE community since 2001.

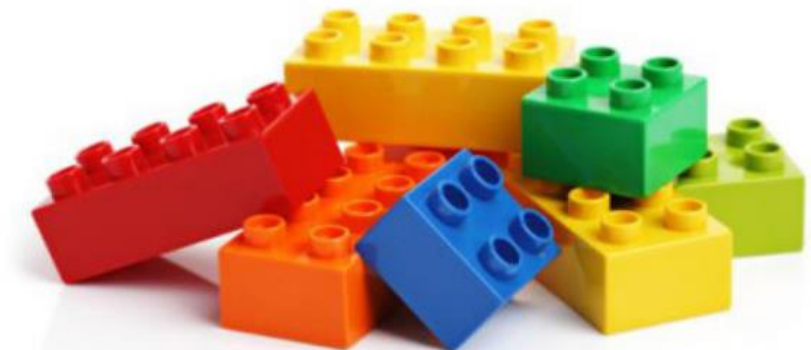
Limitations:

- Flat naming scheme
- Limited packages querying capability
- No software conflict control

```

    acml      AMD core math library
    amber16   Amber16 serial with CUDA 7.5 (GCC 4.9.3)
  amber16_mvapich2 Amber16 with MVAPICH2 2.1 and CUDA 7.5 (GCC 4.9.3)
  amber16_openmpi Amber16 with MVAPICH2 2.1 and CUDA 7.5 (GCC 4.9.3)
    anaconda2 Anaconda2 Python (2.7.8)
    anaconda3 Anaconda3 Python (3.5.1)
    ansys     ANSYS
    atlas     ATLAS Self-tuning BLAS (GCC) [blas]
  atlas_intel14 ATLAS Self-tuning BLAS (INTEL 14) [blas]
  atlas_nehalem_gcc ATLAS Self-tuning BLAS for Nehalem (GCC) [blas]
  atlas_nehalem_intel ATLAS Self-tuning BLAS (GCC) [blas]
  atlas_x86-64_gcc ATLAS Self-tuning BLAS (GCC) [blas]
    aztec     aztec parallel iterative solver library for solving sparse linear systems.
  bazel_0.3.2 Bazel, Google Build Tool (GCC 4.9.3)
    blast     FSA-BLAST (Basic Local Alignment Search Tool)
  blast_intel FSA-BLAST (Basic Local Alignment Search Tool)
    boost     BOOST 1.53.0 (GCC 4.6, Python 2.6) [boost]
  boost_1.58.0 BOOST 1.58.0 (GCC 4.6, Python 2.7.8, OpenMPI 1.8.4) [boost]
  boost_1.58.0_gcc490 BOOST 1.58.0 (GCC 4.9, Python 2.7.8, OpenMPI 1.8.4) [boost]
  boost_1.58.0_python_3.4.2 BOOST 1.58.0 (GCC 4.6, Python 3.4.2, OpenMPI 1.8.4) [boost]
  boost_147_sysgcc BOOST (GCC) [boost]
  boost_151_sysgcc BOOST (GCC) [boost]
  boost_python27_1490 BOOST (GCC) [boost]

```



gcc_compiler_4.9.3

gcc_compiler

intel_cluster_studio_compiler

R_3.3.2-gcc_4.6.1

openmpi_1.10.2

mpich2_intel_ether

openmpi_1.10.6_roce

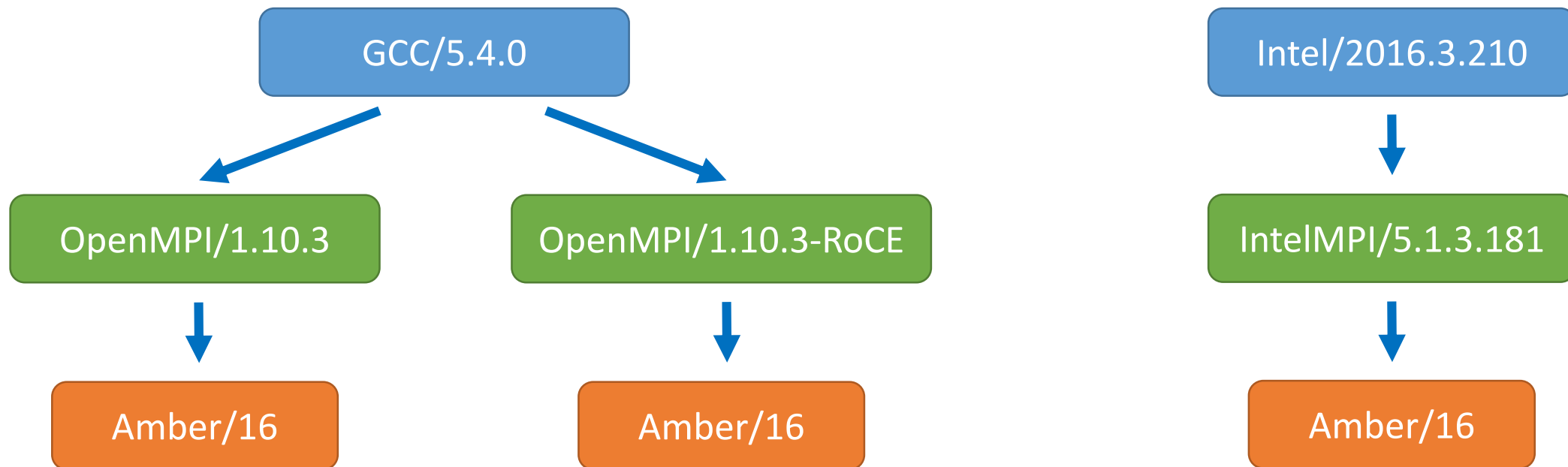
mpich2_intel14_ether

amber16_openmpi_cuda-7.5_roce

scalapack_mpich2_intel



Software is organized in a tree structure and displayed accordingly to the loaded dependencies.



Environment Modules is a well established HPC tool that simplify environment manipulation through modulefiles.

Modulefiles provide all the information required to allow users to run a particular application or provide access to a specific library.

Modules can be loaded and unloaded within the same shell session with an intuitive interface.

Limitations:

- Minimal support of hierarchical modules
- No software conflict control
- Only maintained, not actively developed.



Lmod is a Lua based interpretation of Environment Module created and developed by Robert McLay at TACC.



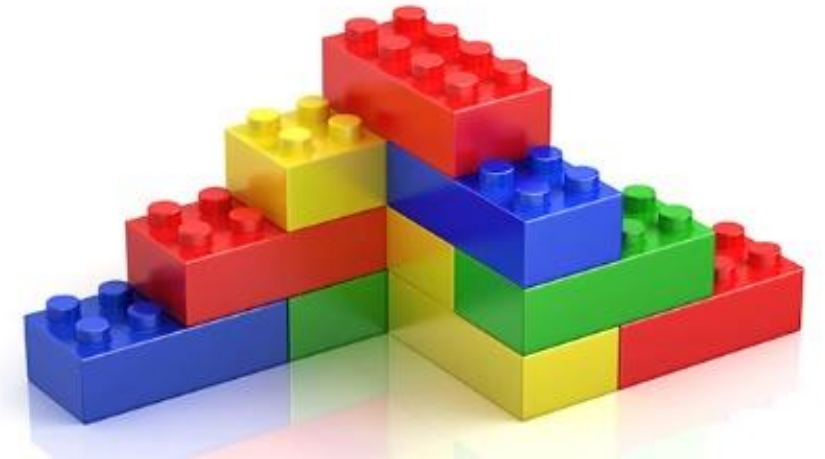
www.tacc.utexas.edu/research-development/tacc-projects/lmod



lmod.readthedocs.org

Advantages:

- Full support of hierarchical modules
- Extended software conflict control
- Actively maintained
- High responsiveness
- More functionalities than EM



module avail *<mod>*

- If no module is passed, print a list of all modules that are available to be loaded.
- If a module is specified, show all available modules with that name.

module load *mod1 mod2 ...*

- Load the specified modules.

module unload *mod1 mod2 ...*

- Unload the specified modules.

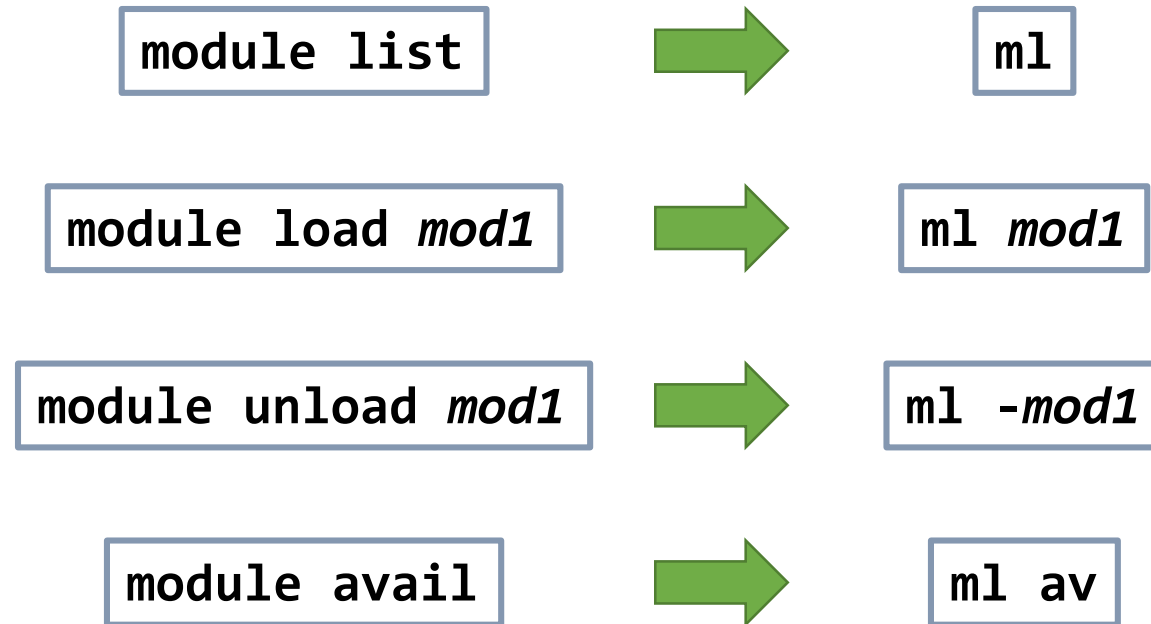
module purge

- Remove all loaded modules from the environment.

module list

- Show all modules loaded in the current environment.

A FEW SHORTCUTS



“Traditional science is all about finding shortcuts”

Rudy Rucker

- 1. Only one version of a given software can be loaded at any time.**

Lmod will automatically unload previously loaded modules on name clash.

- 2. Incompatible programs are prevented to be loaded at the same time.**

E.g. Anaconda2 / Anaconda3 / Python X.Y.Z

- 3. If the version is not specified, the default module is loaded for that software.**

The default module corresponds to the most recent version of that software.

- 4. When unloading a module, Lmod does not automatically unload its dependencies.**

In this way a module unload will not break the dependencies of another loaded module.



How can I search among the visible modules?

`module avail pattern`

- Show only the visible modules that contain the desired pattern.



How can I search through all the modules, even the non visible ones?

`module spider pattern`

- Search all the modules that contain the desired pattern.

- When users log in, no modules are loaded by default. Users can still add `module load` statements in their `~/.bashrc` files.
- There is an alternative (better) way with **named collections**:

```
module save collection_name
```

- Save the list of current loaded modules in `~/.lmod.d/collection_name`.

```
module restore collection_name
```

- Restore the desired named collection in the current environment.

In an HPC environment software installation is done by compiling from source and optimize the codes to the hardware architecture.

ACCRE is now using **EasyBuild** to easily and reproducibly deploy software on the cluster.

EasyBuild is a HPC community-driven framework that supports over 1,200 software tools and application from multiple scientific areas.



Can I get the program *XYZ* installed on the cluster?

If the software is already supported in EasyBuild, it can certainly be installed.

If it is not yet available via EasyBuild, submit a request via helpdesk.

With EasyBuild, all the dependencies are built from source with the available compilers.

The whole software stack will be (mostly) independent from OS libraries.

All non-essential dependencies are hidden for user clarity.

```
module --show-hidden avail
```

- Show all the visible modules, including hidden ones.

```
module --show-hidden spider pattern
```

- Search across all modules, including hidden ones.

To load a hidden module, the version must be specified.

- As of now, 90% of the software available via **setpkgs** is present under **Lmod**.
- With the transition to CentOS 7, **setpkgs** will not be available anymore!
- New compilers/MPI with relative software stacks updated every 12 months.
- Software stacks older than 3 years will not be available.
- Only **EasyBuild** supported software will be installed.



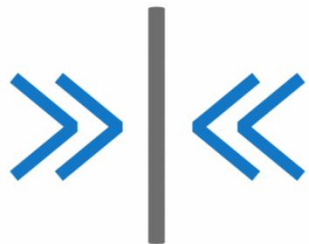
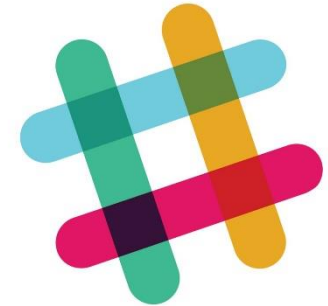
www.accre.vanderbilt.edu/lmod

lmod.readthedocs.org

Join our ACCRE-Forum Slack team with your VU email

<https://accre-forum.slack.com/signup>

And post in the **#Software** channel!



Submit a ticket at our helpdesk web page:

www.accre.vanderbilt.edu/help



May, 24 - 12pm @ Featheringhill Auditorium 134

May, 30 - 2pm @ Light Hall 202

June, 8 - 1pm @ Wilson Hall 112

June, 13 - TBD @ MRB III

Please help us spread the word!