

Module environment on “Hawk”

Björn Dick (HLRS), Philipp Offenhäuser (HLRS)



Dependencies

- Libraries to be used need to be built with the same compiler / MPI
- On Hazel Hen: Everything included in PrgEnv / libsci can be used with every compiler / MPI
- → we aspire to provide the same on Hawk

- Realized via **Lmod** (new implementation of the module command with improved functionality), not “**Tmod**” (TCL-based old one) anymore
- Enable it via `touch ~/.lmod` (and login again)
- Will probably be default on Hawk

- Cf. https://kb.hlrz.de/platforms/index.php/Module_environment (Hawk)

Lmod

- Basic subcommands are the same as in Tmod
- Layout of outputs yet slightly differs
- Modules organized in hierarchical manner

- Libraries to be used need to be built with the same compiler / MPI
- → *only those compatible with the currently loaded compiler and MPI are available*
- If swapping compiler and/or MPI, dependent modules are also swapped automatically
- If not available for the new compiler/MPI, modules are set to 'inactive' state
- 'inactive' modules will be reloaded as soon as respective module becomes available again (by swapping compiler and/or MPI)
- GCC 9.1.0 and MPT 2.20 loaded by default

Use this feature to test different compilers / MPIs

→ might reduce runtime significantly and is almost *for free!*



- Four categories of modules:

1. *System modules*

- Just ignore them
- Always available

2. *Core modules*

- Independent of loaded compiler / MPI
- In particular: compilers themselves
- Always available

3. *Compiler-dependent modules*

- Content depends on loaded compiler
- In particular: MPI modules

4. *MPI-dependent modules*

- Content depends on loaded MPI and compiler
- → also (implicitly) compiler-dependent

The spider command

- List all available modules without considering dependencies:
`module spider`
- List all variants (w.r.t. compiler/MPI) of a module:
`module spider <modulename>`
- List dependencies of a given module:
`module spider <modulename>/<moduleversion>`

- Not allowed to have more than one version of a module loaded (in particular only one compiler!)
- no version specified → Lmod picks “highest” / “first” one (actual rule is more complicated, cf. Lmod documentation)
- Only Bash is supported
- scripts should start with `#!/bin/bash`, **not** `#!/bin/sh`
- switch off pager output by `export LMOD_PAGER=None`