ENVIRONMENT MODULES

DE LA RECHERCHE À L'INDUSTRIE

Modules 5

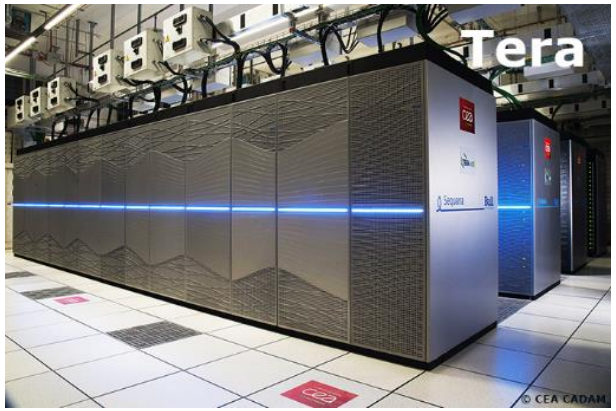
January 27th 2022, 7th EasyBuild User Meeting

Xavier DELARUELLE

WHOIAM

- **Xavier Delaruelle**
  - Work at CEA, the French Alternative Energies and Atomic Energy Commission
  - Operations manager of TGCC, CEA's computing center for the European research
  - Environment Modules project leader since July 2017

- **High Performance Computing @CEA**
  - HPC = a major topic for CEA (for both research and operational activities)
  - 2 large production-level facilities : one for Defense, the other for academics & industry
  - Investment in Open Source Solutions since almost 2 decades with contributions to existing solutions (Lustre, Slurm, etc) or developing our own (ClusterShell, Phobos, RobinHood, NFS Ganesha, etc)
    See https://github.com/cea-hpc/



ENVIRONMENT MODULES STATUS

- **Project is not dead, far from it!**
  - Development has restarted in 2017 and is active since then
- **2-3 feature releases each year**
  - With ≥ 5 new features in each
  - Among that 1 « big » feature each year
  - Bugfix releases made if bugs spotted
- **Available in 14 distribution families**
  - EL, Debian, Homebrew, etc
  - See https://repology.org/project/environment-modules/versions

NEW STUFF SINCE MODULES 3.2

New module sub-commands	New command-line switches	New configuration options	New modulefile commands
reload, source, search, save, restore, saverm, saveshow, savelist, path, paths, autoinit, aliases, test, append-path, prepend-path, remove-path, is-loaded, is-saved is-used, is-avail, info-loaded, config, sh-to-mod, edit, try-load, state, load-any	--debug, --default, --latest, --paginate, --no-pager, --auto, --no-auto, --indepth, --no-indepth, --color, --starts-with, --contains, --json, --trace, --all, -DD, -vv, --output, --width, --redirect, --no-redirect	auto_handling, avail_indepth, collection_pin_version, collection_target, color, colors, contact, extra_siteconfig, implicit_default, locked_configs, pager, rcfile, run_quarantine, silent_shell_debug, search_match, set_shell_startup, term_background, unload_match_order, verbosity, wa_277, advanced_version_spec, extended_default, home, icase, ml, nearly_forbidden_days, avail_output, avail_terse_output, implicit_requirement, list_output, list_terse_output, mcookie_version_check, shells_with_ksh_fpath, tag_abbrev, tag_color_name, term_width, editor, variant_shortcut, quarantine_support, redirect_output, mcookie_check	module-info command, getenv, reportError, reportWarning, module-info loaded, is-saved, is-used, is-avail, module-virtual, set-function, unset-function, source-sh, module-hide, module-forbid, module-info usergroups, module-info username, versioncmp, module-tag, module-info tags, variant, getvariant, prereq-any, require-fullname, depends-on, prereq-all, always-load, module load-any, family

HOW THINGS ARE DEVELOPED

- **Stay as close as possible to standard tools and concepts**
  - `ls, grep, git, dnf/apt`
- **Stay backward compatible as much as possible**
  - Yet new features are introduced
  - But disabled by default if they change behavior with previous version
- **Extensive non-regression testsuite**
  - > 17k non-regression tests (using DejaGnu test framework)
  - Run through CI on various kind of hosts (EL-like, Debian-like, Suse, OS X, Windows, FreeBSD)
- **Documentation of new features**
  - Design notes for non-straightforward features: https://modules.readthedocs.io/en/latest/design.html
  - Highlight with small example: https://modules.readthedocs.io/en/latest/MIGRATING.html
  - Cookbook recipes: https://modules.readthedocs.io/en/latest/cookbook.html
  - Changes between versions: https://modules.readthedocs.io/en/latest/changes.html

MODULES 4

- **First release (4.0) published on October 2017**
- **Last release (4.8) published on July 2021**

- **Major new features introduced**
  - Automated module handling (v4.2)
  - Advanced module version specifiers (v4.4)
  - Hiding modules/Forbidding use of modules (v4.6)
  - Module tags (v4.7)
  - Module variants (v4.8)

AUTOMATED MODULE HANDLING (V4.2)

- **Keep track of loaded module dependencies in environment**
  - With environment variables (__MODULES_LMPREREQ & __MODULES_LMCONFLICT)
  - This information is checked when loading or unloading modules to ensure requirements and conflicts are still satisfied

- **Trigger automatic mechanisms to perform user's order yet keep things consistent**
  - Requirement Load
  - Dependent Unload
  - Useless Requirement Unload
  - Dependent Reload

- **Handle dependency cycles and healing of inconsistent environment**

```
$ module load appY
Loading appY/1.8
  Loading requirement: libb/1.10
$ module list
Currently Loaded Modulefiles:
 1) libb/1.10   2) appY/1.8
```

```
$ module list
Currently Loaded Modulefiles:
 1) libb/1.10   2) appY/1.8
$ module unload libb
Unloading libb/1.10
  Unloading dependent: appY/1.8
```

```
$ module load --no-auto --force appY
Loading appY/1.8
  WARNING: appY/1.8 requires libb/1.10
$ module load libb/1.10
Loading libb/1.10
  Reloading dependent: appY/1.8
$ module list
Currently Loaded Modulefiles:
 1) libb/1.10   2) appY/1.8
```

ADVANCED MODULE VERSION SPECIFIERS (V4.4)

- **Ability to specify finer constraints on module version**
  - Using Spack's terminology and syntax
  - Specify single version: `foo@1.2.3`
  - List of versions: `foo@1.2.3,1.10`
  - Range of versions:
    - Less or equal to (≥): `foo@1.2:`
    - Greater or equal to (≤): `foo@:1.3`
    - In between or equal to: `foo@1.2:1.3`
  - List of ranges: `foo@:1.1,1.3:1.7,1.9:`
- **Available everywhere a module specification is expected**
- **Same syntax whether it is used from the command-line or from a modulefile**

```
$ module av foo
-------------- /path/to/modulefiles --------------
foo/1.1.1(default)  foo/1.2.1  foo/1.10
foo/1.1.10          foo/1.2.3
$ module av foo@1.2:
-------------- /path/to/modulefiles --------------
foo/1.2.1  foo/1.2.3  foo/1.10
```

```
$ module load -v foo@1.2:1.3
Loading foo/1.2.3
```

```
$ module show bar@:2
-------------------------------------------------------
/path/to/modulefiles/bar/2.3:

prereq          foo@1.1.10,1.2.1
-------------------------------------------------------
$ module load --auto bar@:2
Loading bar/2.3
  Loading requirement: foo/1.2.1
```

HIDING MODULES/FORBIDDING USE OF MODULES (V4.6)

- **`module-hide`: dynamically hide modulefiles, module aliases or symbolic versions**
  - modulerc command
  - Can leverage Advanced version specifiers
  - Several hiding level: soft, regular or hard
  - Hide only before or after a given datetime
  - Do not hide for some users or groups
  - Also hide module once loaded

```
$ cat /path/to/modulefiles/qux/.modulerc
#%Module4.6
# softly hide all qux modules
module-hide --soft qux
```

```
$ ml av
-------------- /path/to/modulefiles --------------
bar/1.0  bar/2.0
$ ml av qux
-------------- /path/to/modulefiles --------------
qux/1.0  qux/2.0
$ module load -v qux
Loading qux/2.0
```

- **`module-forbid`: dynamically forbid the evaluation of modulefiles**
  - modulerc command
  - Can leverage Advanced version specifiers
  - Forbid only before or after a given datetime
  - Do not forbid for some users or groups
  - Output specific message when module is nearly forbidden or forbidden

```
$ cat /path/to/modulefiles/qux/.modulerc
#%Module4.6
module-forbid --nearly-message {Version 1.0 will soon expire, please now use\
    version 2.0} --after 2020-09-15 qux/1.0
$ date
Tue 08 Sep 2020 06:49:43 AM CEST
$ ml qux/1.0
Loading qux/1.0
  WARNING: Access to module will be denied starting '2020-09-15'
    Version 1.0 will soon expire, please now use version 2.0
```

MODULE TAGS (V4.7)

- **Associate piece of information to modulefiles**
  - Tags may be inherited from
    - the module state set by a modulefile command
    - consequence of a module action
  - Tags may also be associated by using `module-tag` modulerc command
  - Tags are reported on `avail` and `list` sub-commands output
    - By applying color to associated module
    - Or adding tag name or abbreviation next to module name
- **Special tags:**
  - Inherited: loaded, auto-loaded, keep-loaded, forbidden, nearly-forbidden, hidden, hidden-loaded
  - Set with `module-tag`: sticky, super-sticky

```
$ cat /path/to/modulefiles/foo/.modulerc
#%Module
module-tag mytag foo
module-tag othertag foo/1.0
$ ml av
-------------- /path/to/modulefiles --------------
foo/1.0 <mytag:othertag>  foo/2.0 <mytag>
$ ml foo/1.0
$ ml
Currently Loaded Modulefiles:
 1) foo/1.0 <mytag:othertag>
```

```
$ ml av
-------------- /path/to/other/modulefiles --------------
baz/1.0  baz/2.0

----------------- /path/to/modulefiles -----------------
bar/1.0  bar/2.0  foo/1.0  foo/2.0  foo/2.2

Key:
modulepath       module-alias  sticky
default-version  forbidden
```

MODULE VARIANTS (V4.8)

- **Pass arguments to evaluated modulefiles**
  - Achieve different environment setup or module requirement with a single modulefile
  - Using Spack's terminology and syntax
  - Valued-variant or Boolean-variant
  - Shortcuts could be set to ease specification

```
#%Module4.8
variant toolchain a b c
variant --boolean --default off debug

# select software build depending on variant values
set suffix -[getvariant toolchain]
if {$ModuleVariant(debug)} {
    append suffix -dbg
}

prepend-path PATH /path/to/bar-1.2$suffix/bin
prepend-path LD_LIBRARY_PATH /path/to/bar-1.2$suffix/lib
```

```
$ module purge
$ module config variant_shortcut toolchain=%
$ module load foo/2.1 %a
Loading foo/2.1{%a}
  Loading requirement: bar/1.2{-debug:%a}
```

```
$ module list
Currently Loaded Modulefiles:
 1) bar/1.2{-debug:%a}  2) foo/2.1{%a}

Key:
auto-loaded  {-variant}={variant=off}  {%value}={toolchain=value}  {variant=va
```

HOW TO CONFIGURE MODULES

- **Configuration option values can be set at build time**
  - With `--enable-x` or `--with-y` options passed to `./configure`
- **Once installed configuration can be changed in Modules initialization file**
  - `/etc/environment-modules/initrc`
  - Written as a Tcl script supporting modulefile commands
    - Change default configuration with `module config`
    - Setup default modulepaths with `module use/module restore`
    - Load default modules for user with `module load/module restore`
- **Extend `modulecmd.tcl` code with site-specific script**
  - `/etc/environment-modules/siteconfig.tcl`
  - Hook existing internal procedures by superseding them (with `rename` Tcl command) or execute defined procedure at start or end (with `trace` Tcl command)
  - Example: trace all modulefile evaluations to log them (see https://modules.readthedocs.io/en/latest/cookbook/log-module-commands.html)

MODULES 5

- **First release (5.0) published on September 2021**

- **Why changing the major version number and not staying in the 4.x cycle?**
  - Enable by default new features introduced in version 4 to benefit from an enhanced experience by default
  - Catch the train toward RHEL 9 with this new major version

- **New features of version 4 enabled by default starting Modules 5.0**
  - Automated module handling
  - Extended default
  - Advanced module version specifiers
  - Colored output
  - Case insensitive module search

V5.1: MODULE SEARCH PERFORMANCE IMPROVEMENT

- **By default all files under enabled modulepaths are read**
  - to check if they start by #%Module to determine if they are modulefiles

- **Introduce new configuration option `mcookie_check`**
  - When set to `eval`, skip file verification when searching for modulefiles
  - All files under modulepaths are considered modulefiles
  - Save 3 I/O operations per existing modulefile (`open`, `read`, `close`)
  - May observe substantial I/O load reduction

```
$ module -o "" avail -t | wc -l
1098
$ module config mcookie_check always
$ strace -f -c -S name -e open,read,close -U calls,name \
    --silence=attach $MODULES_CMD bash avail 2>always.out
$ module config mcookie_check eval
$ strace -f -c -S name -e open,read,close -U calls,name \
    --silence=attach $MODULES_CMD bash avail 2>eval.out
$ ./icdiff --cols=66 always.out eval.out
always.out                    eval.out
...
    calls syscall                 calls syscall
--------- ---------------  --------- ---------------
     2056 close                    944 close
     1734 open                     612 open
     2146 read                    1034 read
--------- ---------------  --------- ---------------
     5936 total                   2590 total
```

V5.1: COMPATIBILITY WITH LMOD TCL MODULEFILES

- **Goal: make Modules able to evaluate a Tcl modulefile written for Lmod**
  - Ease the life of people writing modulefiles or writing tools to generate modulefiles,

- **Add support for the Tcl modulefile commands introduced by Lmod**
  - v4.2: `set-function`, `unset-function`
  - v4.8: `module try-load`
  - v5.1 (dummy support): `add-property`, `remove-property`, `extensions`
  - v5.1: `prereq-any`, `require-fullname`, `depends-on`, `always-load`, `module load-any`, `family`, `pushenv`

A LMOD/MODULES COMPARISON (JANUARY 2022)

Lmod unique features (v8.6)	**Modules** unique features (upcoming v5.1)
software hierarchy · modulefile cache · Lua modulefile support · one name rule · module auto-swap · inactive modules · module spider · hook functions · path entry priority · atleast/between/latest modifier functions · find best module version · custom label for modulepaths · i18n · nag message · module properties · module extensions · module overview · LMOD_QUARANTINE_VARS · smooth integration with XALT	modulefile constraint consistency · automated module handling · explicit conflict constraint · modulescript sourcing · virtual modules · environment direct handling command · full path modulefile · advanced module version specifiers · module-forbid · module tags · module variants · no-indepth avail · module config · icase load · Windows support · avail/list output configuration · super-sticky modules · hidden-loaded modules · module edit · modulecmd quarantine run

V5.1/V5.2: IMPROVED TAGGING CAPABILITIES

- **Tag module when loading it**
  - `module load --tag sticky foo/1.0`
  - Either when loaded from the command-line or from modulefile

- **Extend tags to modulepaths**
  - `module use --tag sticky /path/to/modulefiles`

V5.2: ADVANCED MODULE SEARCH QUERIES

- **Search for modulefiles based of their properties**
  - Tags
  - Dependencies
  - Variants

- **Some examples (not definitive syntax!)**
  - `module avail toolchain=foss21a`
  - `module avail conflict:foo`
  - `module avail require:bar`
  - `module avail tag:nearly-forbidden`

SOME OF THE COOL STUFF AHEAD

- **Modulefile cache**
  - 1 cache file per-modulepath
  - That contains everything to avoid evaluation of other files yet keeping things dynamic
- **New automated module handling mechanisms**
  - Conflict Unload
  - Load Compatible
- **`module stash`/`module pop`, relying on collections**
- **…**

Thanks for your attention

Website: http://modules.sourceforge.net/

Code: https://github.com/cea-hpc/modules

Documentation: https://modules.readthedocs.io

News: https://twitter.com/EnvModules