



TEXAS ADVANCED COMPUTING CENTER

WWW.TACC.UTEXAS.EDU



TEXAS

The University of Texas at Austin

# SC 22 Booth Talk: Lmod

Robert McLay

Nov. 15, 2022

# Introduction



- ▶ Features and History
- ▶ Advanced Topics
- ▶ Future work?

# Features

- ▶ Current version is Lmod 8.7.13
- ▶ Reads for TCL and Lua modulefiles
- ▶ One name rule.
- ▶ Support Software Hierarchy (but not required!)
- ▶ Spider Cache: fast `$ module avail`
- ▶ Properties (gpu, mic)
- ▶ family(“compiler”) family(“mpi”) support
- ▶ Optional Tracking: What modules are loaded?
- ▶ Many other features: ml, collections, hooks, extended default, nag ...
- ▶ Google analytics report about 2000 unique users around the world read docs every month.

# Recent Feature of Lmod 8+

- ▶ The TCL interpreter is now (optionally) embedded with Lmod.
- ▶ `depends_on()`
- ▶ New Function: `extensions("numpy/1.16.4  
scipy/1.4")`
- ▶ Checking your module tree 8.4.3+

# depends\_on()

- ▶ Modules X and Y depends on Module A
- ▶ `ml purge; ml X; ml unload X;  $\Rightarrow$  unload A`
- ▶ `ml purge; ml X Y; ml unload X;  $\Rightarrow$  keep A`
- ▶ `ml purge; ml X Y; ml unload X Y ;  $\Rightarrow$  unload A`
- ▶ `ml purge; ml A X Y; ml unload X Y ;  $\Rightarrow$  keep A`

# extensions() function

- ▶ `extensions()`: Tells users that a module has extensions
- ▶ E.G: python has numpy and scipy
- ▶ `extensions("numpy/1.16.4, scipy/1.4")`

# extensions() function (II)

- ▶ Users can use spider to find extensions.
- ▶ Users can use avail to list extensions base name

# Checking your module tree 8.4.3+

- ▶ New command added:  
`$LMOD_DIR/check_module_tree_syntax`
- ▶ Reports syntax errors across the entire `$MODULEPATH`
- ▶ Report which modules have multiple marked defaults sets
- ▶ Precedent order: default symlink, `.modulerc.lua`, `.modulerc`, `.version`
- ▶ Does not check `SYSTEM MODULERCFILE` for defaults.



# Lmod 8.6+ Features

- ▶ New Features now reported:  
[https://lmod.readthedocs.io/en/latest/025\\_new.html](https://lmod.readthedocs.io/en/latest/025_new.html)
- ▶ `module overview`
- ▶ `module -d avail`
- ▶ New config file: `/etc/lmod/lmod_config.lua`
- ▶ `LMOD_QUARANTINE_VARS`
- ▶ updates to `sh_to_modulefile`
- ▶ `source_sh()`: source a shell script inside a modulefile

# module overview

```
% module overview
----- /opt/apps/modulefiles/Core -----
StdEnv      (1)  hashrf      (2)  papi          (2)  xalt          (1)
ddt          (1)  intel       (2)  singularity  (2)
git          (1)  noweb        (1)  valgrind     (1)

----- /opt/apps/lmod/lmod/modulefiles/Core -----
lmod (1)    settarg (1)
```

# \$LMOD\_QUARANTINE\_VARS

- ▶ A module at TACC turn-off \$LMOD\_PAGER
- ▶ This#%& module made me mad.
- ▶ Tmod has a new feature kinda like this.
- ▶ \$LMOD\_QUARANTINE\_VARS was invented.

# \$LMOD\_QUARANTINE\_VARS (II)

- ▶ `export LMOD_QUARANTINE_VARS=LMOD_PAGER:LMOD_REDIRECT`
- ▶ This means that a module can't change those variables.
- ▶ This only works with regular env. vars.
- ▶ You can't quarantine PATH like variables.
- ▶ A user sets this variable in their `~/.bashrc` or similar file.
- ▶ This obviously won't work for modules loaded during the processing of `/etc/profile.d/*.sh` files
- ▶ Use <https://github.com/TACC/ShellStartupDebug> support users.

# /etc/lmod/lmod\_config.lua configuration file

- ▶ This file is evaluated during Lmod startup.
- ▶ This location is the default during configuration.
- ▶ A site can change this location at configuration.

```
-- Example /etc/lmod/lmod_config.lua
require("strict")
local cosmic = require("Cosmic"):singleton()

cosmic:assign("LMOD_SITE_NAME", "XYZZY")
local function foo()
    ...
end
sandbox_registration { foo = foo }
```

# Sourcing shell scripts inside a modulefile w/ `source_sh()`

- ▶ This was first implemented in Tmod 4.7
- ▶ Xavier told me that he did this during Covid Lockdown in France.
- ▶ Lmod 8.6 re-implements this feature in a similar way.
- ▶ It knows about env. vars and shell functions and aliases.

# source\_sh() Implementation

- ▶ It captures the env. vars/functions/alias before and after the running the shell script.
- ▶ It computes the difference and saves it into the ModuleTable in env.
- ▶ It can be safely unloaded, shown.
- ▶ script path and arguments must not change between load and unload.
- ▶ `module refresh` works
- ▶ Obvious points:
  - ▶ It is better to use `sh_to_modulefile` and convert once.
  - ▶ But `sh_to_modulefile` is not dynamic (e.g. `$HOME`)
  - ▶ Can't have run the script in the user environment before loading the script.

# ml -mt

```
_ModuleTable_ = {  
  MTversion = 3,  
  mT = {  
    wrapperSh = {  
      fn = "/home/user/w/lmod/rt/sh_to_modulefile/mf/wrapperSh/1.0.lua",  
      fullName = "wrapperSh/1.0",  
      loadOrder = 1,  
      mcmdT =  
        ["/home/user/w/lmod/rt/sh_to_modulefile/second.sh arg1"] = {  
          "setenv(\"SECOND\", \"FOO_BAR\")",  
        },  
        ["/home/user/w/lmod/rt/sh_to_modulefile/tstScript.sh"] = {  
          "setenv(\"MY_NAME\", \"tstScript.sh\")",  
          "prepend_path(\"PATH\", \"/home/user/w/lmod/rt/sh_to_modulefile/bin\")",  
          "set_alias(\"fooAlias\", \"foobin -q -l\")",  
          [[set_shell_function("banner", " \\  
local str=\"$1\";\\  
local RED='\27[1;31m';\  
local NONE='\27[0m';\  
echo \"$RED$str$NONE\"\  
")]],  
        },  
      },  
    },  
  },  
}
```



# Lmod Monthly Zoom Mtg

## ► Previous Topics

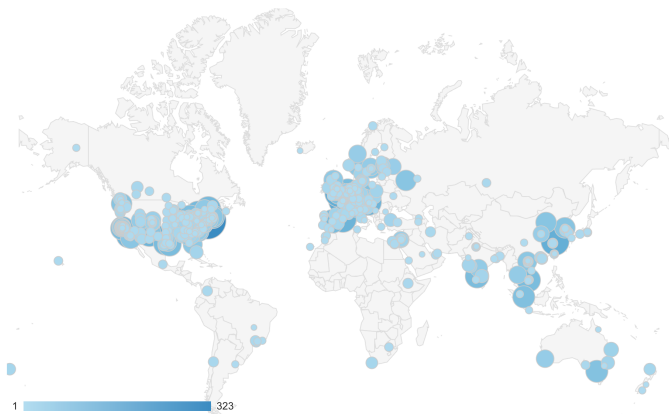
1. Lmod Hooks Discussion
2. Debugging Modulefiles
3. Lmod 8.6 new features
4. Settag and integrating Lmod with build system
5. How Module collections works
6. How to get current module info into hooks
7. How Lmod testing works

## ► Future Topics:

1. How TCL modules are interpreted

- Zoom Mtg Usually 1st Tuesday at 15:30 UTC (9:30 US Central)
- See Mailing list or <https://github.com/TACC/Lmod/wiki> for details
- Next Meeting Dec. 6th at 15:30 UTC (9:30 US Central)

# Lmod Doc usage by City



# Future Work

- ▶ Lmod can optionally track usage.
- ▶ Future: Make it easier to not remember loads after 1 year.
- ▶ A monthly discussion group? (YES!!)
- ▶ Remember text of modules to unload.

# Conclusions: Lmod 8+



- ▶ Latest version: <https://github.com:TACC/lmod.git>
- ▶ Stable version: <http://lmod.sf.net>
- ▶ Documentation: <http://lmod.readthedocs.org>