



# Meeting of the Technical Steering Committee (TSC) Board

Wednesday, February 24<sup>th</sup>, 2021  
11:00am ET

# Antitrust Policy Notice

- Linux Foundation meetings involve participation by industry competitors, and it is the intention of the Linux Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.
- Examples of types of actions that are prohibited at Linux Foundation meetings and in connection with Linux Foundation activities are described in the Linux Foundation Antitrust Policy available at <http://www.linuxfoundation.org/antitrust-policy>. If you have questions about these matters, please contact your company counsel, or if you are a member of the Linux Foundation, feel free to contact Andrew Updegrave of the firm of Gesmer Updegrave LLP, which provides legal counsel to the Linux Foundation.

# Agenda/Updates

- Announcements, Upcoming talks and deadlines
    - ISC 2021
      - Accepted BoF Sessions of ISC 2020 will be held at ISC 2021
    - PEARC'21 BoF submission – due May 9
    - SC'21 BoF submission – due August 6
    - Infrastructure outage (postponed due to Texas power outages):
      - OpenHPC rack at TACC will be relocating in their datacenter
      - CI and OBS build system will be affected
- 
- Linaro CI system updates?
  - Cloud working group
  - v2.1 stuff
    - singularity
    - slurm
    - misc. items

# OHPC Cloud Updates

- PEARC Paper on Deploying OpenHPC at AWS
  - [OpenSUSE Leap 15](#)
  - [Sections on:](#)
    - NFS vs EFS vs FSx for Lustre
    - Elastic Fabric Adapter setup and performance
    - ARM and x86 and recommended instance types
  - [David B. will be submitting an SC21 tutorial](#)

# Singularity updates

- Our singularity build in 2.0 release was based on an older upstream version from 2019
- Folks may recall previous discussion on the reasoning:
  - new singularity required newer go version than what was available in suse
  - we did not want to have significantly different versions across 2 distros
- Question came up on slack about why we are behind
  - we can now get go v1.14 on Leap 15.2 which is sufficient to build the latest version of Singularity (released on Jan. 12)
  - significant layout changes over our previous build, but no issues detected in CI

The screenshot displays the Singularity package page. At the top, the package name 'singularity' is shown in green, with the note 'No description set'. Below this are several action buttons: 'Branch package', 'Submit package', 'Edit description', 'Delete package', and 'Trigger services'. The 'Source Files' section contains a table with columns for 'Filename', 'Size', 'Changed', and 'Actions'. The table lists various files such as '\_service', '\_service:download\_files:singularity-3.7.1.tar.gz', and '\_service:extract\_file:OHPC\_macros'. To the right, the 'Build Results' section shows successful builds for 'CentOS\_8' and 'Leap\_15.2' on both 'x86\_64' and 'aarch64' architectures. A 'Refresh' button is also present.

Filename	Size	Changed	Actions
._service	1.8 KB	2 minutes	
._service:download_files:singularity-3.7.1.tar.gz	5.84 MB	about 1 month	
._service:extract_file:OHPC_macros	7.22 KB	about 24 hours	
._service:extract_file:rpmlintrc	136 Bytes	12 months	
._service:extract_file:singularity-suse-timezone.patch	398 Bytes	12 months	
._service:extract_file:singularity.spec	4.61 KB	12 minutes	
._service:tar_scm:_servicedata	233 Bytes	12 minutes	
._service:tar_scm:git-infra-1614129364.74a65f35a.ta r	50 KB	12 minutes	

# Singularity updates

- Potential module update for consideration:
  - singularity ships with a Bash completion script
  - installed as part of our RPM but a user would have to source by hand to access, e.g.

```
$ source /opt/ohpc/pub/libs/singularity/3.4.2/etc/bash_completion.d/singularity
```

- While working on updated .spec to build latest version, thought it might be nice to do this directly as part of the module load
  - there is different support functionality in Lmod depending on whether using .tcl or .lua module files
    - [https://lmod.readthedocs.io/en/latest/050\\_lua\\_modulefiles.html](https://lmod.readthedocs.io/en/latest/050_lua_modulefiles.html)
    - [https://lmod.readthedocs.io/en/latest/051\\_tcl\\_modulefiles.html](https://lmod.readthedocs.io/en/latest/051_tcl_modulefiles.html)
  - Lua parsing supports an “execute” command directly to be run at end of “module load”

```
execute {cmd="<any command>",modeA={"load"}}
```

Run any command with a certain mode. For example **execute** {cmd="ulimit -s unlimited",modeA={"load"}} will run the command **ulimit -s unlimited** as the last thing that the loading the module will do.

# Singularity updates

## tcl modulefile variant

```
##Module1.0#####  
  
proc ModulesHelp { } {  
  
    puts stderr " "  
    puts stderr "This module loads the singularity utility"  
    puts stderr "\nVersion 3.4.2\n"  
  
}  
module-whatis "Name: singularity"  
module-whatis "Version: 3.4.2"  
module-whatis "Category: runtime"  
module-whatis "Description: Application and environment  
virtualization"  
module-whatis "URL https://www.sylabs.io/singularity/"  
  
set      version          3.4.2  
  
prepend-path PATH          /opt/ohpc/pub/libs/singularity/3.4.2/bin  
prepend-path MANPATH       /opt/ohpc/pub/libs/singularity/3.4.2/share/man  
  
setenv   SINGULARITY_DIR  /opt/ohpc/pub/libs/singularity/3.4.2  
setenv   SINGULARITY_BIN  /opt/ohpc/pub/libs/singularity/3.4.2/bin
```

## lua modulefile variant (w/ bash completion)

```
help (
[[

This module loads the singularity utility

Version 3.7.1

]])

whatis ("Name: singularity")
whatis ("Version: 3.7.1")
whatis ("Category: runtime")
whatis ("Description: Application and environment virtualization")
whatis ("URL: https://www.sylabs.io/singularity/")

prepend_path("PATH", "/opt/ohpc/pub/libs/singularity/3.7.1/bin")
prepend_path("MANPATH", "/opt/ohpc/pub/libs/singularity/3.7.1/share/man")

setenv ("SINGULARITY_DIR", "/opt/ohpc/pub/libs/singularity/3.7.1")
setenv ("SINGULARITY_BIN", "/opt/ohpc/pub/libs/singularity/3.7.1/bin")

-- Load bash completion
if (myShellName() == "bash") then
    execute{cmd="source /opt/ohpc/pub/libs/singularity/3.7.1/etc/bash_completion.d/singularity", modeA={"load"}}
end
```

# Singularity updates

- resulting behavior with the .lua variant

```
[ohpc-test@sms005 ~]$ ml singularity
```

```
[ohpc-test@sms005 ~]$ singularity [TAB] [TAB]
```

```
build      config    inspect   oci       push      run-help  sif       verify
cache      delete    instance  plugin    remote    search    sign      version
capability exec      key       pull      run       shell     test
```

- how do folks feel about having .lua based modulefile for singularity?
- aversion to slowly migrating to .lua files for all our packaging?



# SLURM update

- recall from last time, intent to update to latest slurm (v20.11.3) which restores job launch capability for openmpi
- build was updated, but this broke everything initially in CI
  - slurm not starting due to our example config file now using deprecated feature for accounting

```
* Changes in Slurm 20.11.0rc1
=====
...
-- accounting_storage/filetxt has been removed as an option.
```

- we used this previously to demo a simple way to get at job completion accounting for slurm (ie. allow use of **sacct**)

# SLURM update

- have removed this accounting option so that slurm daemons fire up
- restores CI functionality in all but 1 test
  - we had 1 which verifies a working **sacct** binary which is no longer true
  - temporarily disabled this check in our test harness
- potential options going forward (see <https://slurm.schedmd.com/accounting.html>) :
  1. leave as is currently; no accounting mechanism in effect
  2. update config to demonstrate creation of a simple job completion file
    - sites can parse themselves for accounting purposes, e.g.  
**JobCompType=jobcomp/filetxt**  
**JobCompLoc=/var/log/slurm/job\_completions**
    - above gives you storage of basic stats for all jobs in ascii file, but **sacct** won't read it
  3. suffer thru setting up slurmdbd accounting
    - this is now the only supported path to enable interrogation of job accounting with **sacct**

## 2.1 – other updates

- From last time:
  - fixed warewulf-vnfs build on CentOS
  - updated Lustre client to accommodate CentOS8.3 kernel
    - Lustre v2.12.6 - this works fine for CentOS 8.3
    - unfortunately, newer Lustre version breaks build on Leap 15.2
      - proposing to stick with previous Lustre v2.12.5 version for Leap that works ok
  - squashed a few CI failures for mvapich2
    - mpi4py ok (env variable tweaks)
  - still have some lingering mv2 failures in a few packages...

# 2.1 – other updates

- Example CI failure for MV2
  - mostly consistent between SLURM and OpenPBS
  - do have 2 MFEM test failures that are only seen on OpenPBS

(2.1) - (centos8.3,x86\_64) (warewulf+openpbs) (fabric=ib) - UEFI #4 Test Results

## Test Result

7 failures (-4)

1,461 tests (±0)  
Took 13 min.  
[add description](#)

### All Failed Tests

Test Name	Duration	Age
<a href="#">UserLevelTests.Extrae.[Extrae] Parallel trace merge runs under resource manager (openpbs/gnu9/mvapich2)</a>	0 ms	4
<a href="#">UserLevelTests.FFTW.[libs/FFTW] MPI C binary runs under resource manager (openpbs/gnu9/mvapich2)</a>	0 ms	4
<a href="#">UserLevelTests.Mfem.[libs/MFEM] p_laplace_perf MPI binary runs under resource manager (openpbs/gnu9/mvapich2)</a>	0 ms	4
<a href="#">UserLevelTests.Mfem.[libs/MFEM] p_diffusion MPI binary runs under resource manager (openpbs/gnu9/mvapich2)</a>	0 ms	4
<a href="#">UserLevelTests.Minife.[Apps/miniFE] run miniFE on multi nodes under resource manager (openpbs/gnu9/mvapich2)</a>	0 ms	4
<a href="#">UserLevelTests.PTScotch.[libs/PTScotch] dgraph_redist binary runs under resource manager (openpbs/gnu9/mvapich2)</a>	0.98 sec	4
<a href="#">UserLevelTests.PTScotch.[libs/PTScotch] dgord binary runs under resource manager (openpbs/gnu9/mvapich2)</a>	1 sec	4

### All Tests

Package	Duration	Fail	(diff)	Skip	(diff)	Pass	(diff)	Total	(diff)
InstallTests	0.54 sec	0		0		7		7	
RootLevelTests	19 sec	0		0		36		36	
UserLevelTests	13 min	7	-4	0		1411	+4	1418	