CentOS 7 to Rocky 8: How the new operating system will affect FASRC clusters

Content

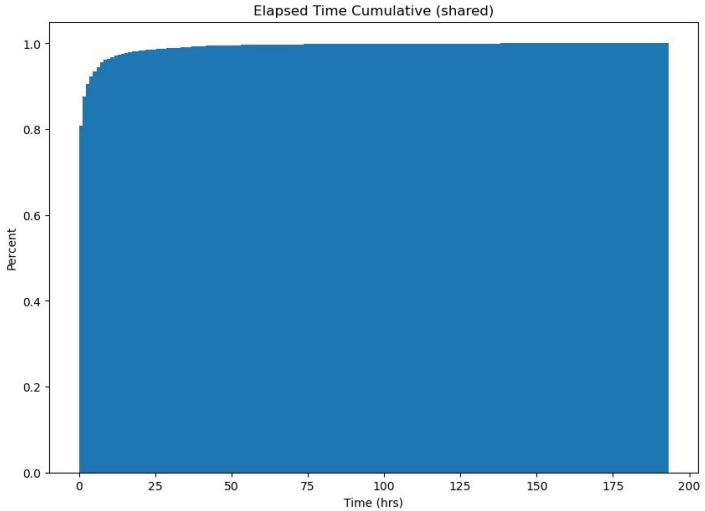
- Why will FASRC clusters will move to Rocky 8?
- Partition changes
  - Public partitions will have a 3 day time limit
  - New intermediate partition to handle 3-14 day jobs
  - Unrestricted partition will be expanded
- Software changes
  - Software will need to be rebuilt
  - New strategy for Software
- Login nodes will have hard memory and cpu usage limits but no more pcull
- Test cluster information

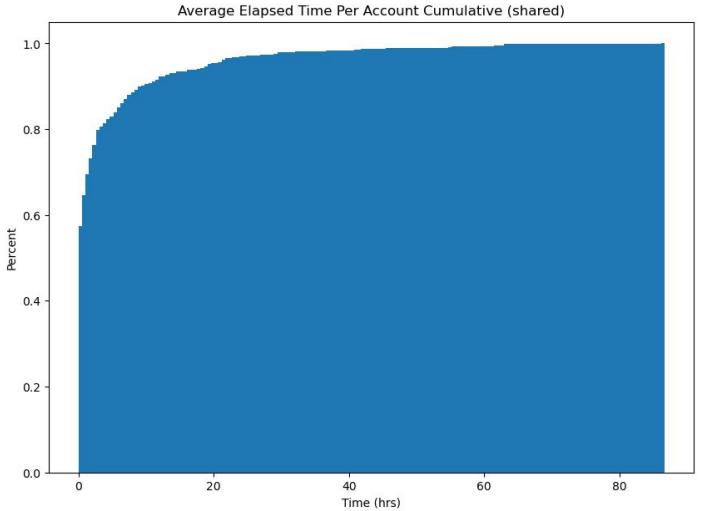
Why move to Rocky 8?

- CentOS 7: non-commercial version of RedHat's Enterprise Linux (RHEL)
- CentOS is being discontinued by RedHat
- New development ceased at the end of 2021
  - potential security vulnerabilities
  - incompatibility with new software
- Rocky 8 Linux
  - created by original CentOS developers
  - good portion of HPC community is also adopting it -> more community support

Partition time limit changes

- Public partitions will have a 3 day time limit
- New intermediate partition to handle 3-14 day jobs
- Unrestricted partition will be expanded





Software changes

- Most software built on CentOS 7 will not work
- Rocky 8 software:
  - HeLmod (modules): only compilers, basic libraries, and a few software
  - Spack for most software installs
  - Singularity
  - Julia
  - Python
  - R with Spack

(cont.) Software changes

- Legacy CentOS 7 support
  - Singularity container with full compute node environment and CentOS 7 modules
  - CentOS 7 will not be maintained for the compute environment
  - Slurm support for CentOS 7 will be dropped with the next major Slurm upgrade

HeLmod (modules)

- Compilers: gcc, Intel, openmpi, cuda
- Basic libraries (e.g., HDF5, gsl, NetCDF)
- Very common libraries (e.g., R, python, Mamba, LAMMPS, GAMESS, GROMACS, CP2K, etc. but customization will be left to the user through spack)
- Commercial software

```
[jharvard@rockylogin ~]$ module load matlab
[jharvard@rockylogin ~]$ module list

Currently Loaded Modules:
  1) matlab/R2022b-fasrc01
```

(cont.) HeLmod (modules)

```
[jharvard@rockylogin ~]$ module spider gcc
----------------------------------------------------------------------------
  gcc:
----------------------------------------------------------------------------
    Description:
      the GNU Compiler Collection

     Versions:
        gcc/10.2.0-fasrc01
        gcc/12.2.0-fasrc01

[jharvard@rockylogin ~]$ module spider open
----------------------------------------------------------------------------
  openmpi:
----------------------------------------------------------------------------
    Description:
      Full MPI-3.1 standards conformance

     Versions:
        openmpi/4.1.0-fasrc01
        openmpi/4.1.4-fasrc01
```

(cont.) HeLmod (modules)

```
[jharvard@rockylogin ~]$ module load intel
[jharvard@rockylogin ~]$ module avail

------------------ /n/sw/helmod-rocky8/modulefiles/Comp/intel/23.0.0-fasrc01 -------------------
   intelmpi/2021.8.0-fasrc01    mpich/4.1-fasrc01    openmpi/4.1.4-fasrc01

--------------------------- /n/sw/helmod-rocky8/modulefiles/Core ---------------------------------
   Anaconda2/2019.10-fasrc01             gmp/6.2.1-fasrc01                python/2.7.16-fasrc01
   IDL/8.7.2-fasrc01                     gsl/2.7-fasrc01                  python/3.10.9-fasrc01
(D)
   Mambaforge/22.11.1-fasrc01            intel-mkl/23.0.0-fasrc01         rstudio/2023.03.0-fasrc01
   R/4.2.2-fasrc01                       intel/23.0.0-fasrc01       (L)   sas/9.4-fasrc01
   cmake/3.25.2-fasrc01                  mathematica/12.1.1-fasrc01       stata/17.0-fasrc01
   cuda/12.0.1-fasrc01                   matlab/R2022b-fasrc01      (L)   szip/2.1.1-fasrc01
   cudnn/8.8.0.121_cuda12-fasrc01        mpc/1.2.1-fasrc01                vscode/1.76-fasrc01
   flex/2.6.4-fasrc01                    mpc/1.3.1-fasrc01          (D)   zlib/1.2.11-fasrc01
   gcc/10.2.0-fasrc01                    mpfr/4.1.0-fasrc01               zlib/1.2.13-fasrc01
(D)
   gcc/12.2.0-fasrc01             (D)    mpfr/4.2.0-fasrc01         (D)
```

Spack

One-time setup

- Clone Spack repo in your lab storage (better performance than home dir)
- Source spack
- Install packages with Spack - some software can take a few hours to build

Job submission

- Source spack
- Load packages/software with Spack
- Run code

Spack: one-time setup

```
# request interactive job
[jharvard@rockylogin ~]$ salloc -p rocky --mem 12g -t 0-04:00 -c 8

# use lab storage
[jharvard@holy7c12104 ~]$ cd /n/holylabs/LABS/jharvard_lab/Lab/software/

# clone spack
[jharvard@holy7c12104 software]$ git clone -c feature.manyFiles=true
https://github.com/spack/spack.git
[jharvard@holy7c12104 software]$ cd spack/
[jharvard@holy7c12104 spack]$ git checkout releases/v0.19

# source spack
[jharvard@holy7c12104 spack]$ . share/spack/setup-env.sh

# install packages
[jharvard@holy7c12104 spack]$ spack install bzip2                  # install latest version
[jharvard@holy7c12104 spack]$ spack install bzip2@1.0.8            # specify version
[jharvard@holy7c12104 spack]$ spack install zlib@1.2.13%gcc@8.5.0  # specify version and compiler
```

Slurm job with Spack

```
#!/bin/bash
#SBATCH -J r_spack           # Job name
#SBATCH -c 1                 # Number of cores (--cpus-per-task)
#SBATCH -t 0-00:10           # Runtime in D-HH:MM, minimum of 10 minutes
#SBATCH -p test              # Partition to submit to
#SBATCH --mem=4g             # Memory for all cores in GB (see also --mem-per-cpu)
#SBATCH -o myoutput_%j.out   # File to which STDOUT will be written, %j inserts jobid
#SBATCH -e myerrors_%j.err   # File to which STDERR will be written, %j inserts jobid

# source spack
. /n/holylabs/LABS/jharvard_lab/Users/jharvard/spack/share/spack/setup-env.sh

# load spack packages
spack load r-codetools
spack load r-rgdal
spack load r-raster
spack load r-terra

# run R code
Rscript --vanilla r_spack_load_libs.R > r_spack_load_libs.Rout
```

Singularity

- You can build containers
  - from existing container in SingularityCE container library
  - from existing container in Docker Hub
  - from SingularityCE definition file on Sylabs cloud
  - **NEW**: from SingularityCE definition file and `proot` directly on Cannon

Julia

```
# use lab storage
[jharvard@holy7c12104 ~]$ cd /n/holylabs/LABS/jharvard_lab/Users/jharvard/software/

# download julia and extract
[jharvard@holy7c12104 software]$ wget
https://julialang-s3.julialang.org/bin/linux/x64/1.8/julia-1.8.5-linux-x86_64.tar.gz
[jharvard@holy7c12104 software]$ tar xvfz julia-1.8.5-linux-x86_64.tar.gz

# add julia to path
[jharvard@holy7c12104 julia-1.8.5]$ export
PATH=$PATH:/n/holylabs/LABS/jharvard_lab/Users/jharvard/software/julia-1.8.5/bin

[jharvard@holy7c12104 julia-1.8.5]$ julia

               _
   _       _ _(_)_     |  Documentation: https://docs.julialang.org
  (_)     | (_) (_)    |
   _ _   _| |_  __ _   |  Type "?" for help, "]?" for Pkg help.
  | | | | | | |/ _` |  |
  | | |_| | | | (_| |  |  Version 1.8.5 (2023-01-08)
 _/ |\__'_|_|_|\__'_|  |  Official https://julialang.org/ release
|__/                   |

julia>
```

Python

- Mamba replacing Anaconda: fast, robust, and cross-platform package manager
- Python 3

```
[jharvard@rockylogin ~]$ module load python/3.10.9-fasrc01
[jharvard@rockylogin ~]$ module list

Currently Loaded Modules:
  1) Mambaforge/22.11.1-fasrc01   2) python/3.10.9-fasrc01
```

- Python 2

```
[jharvard@rockylogin ~]$ module load python/2.7.16-fasrc01
[jharvard@rockylogin ~]$ module list

Currently Loaded Modules:

  1) Anaconda2/2019.10-fasrc01   2) python/2.7.16-fasrc01
```

Python and Mamba

```
# load modules
[jharvard@holy7c12104 ~]$ module load python/3.10.9-fasrc01

# create and activate mamba environment
[jharvard@holy7c12104 ~]$ mamba create -n python_env1 python=3.10 pip wheel
[jharvard@holy7c12104 ~]$ source activate python_env1

# install packages
(python_env1)[jharvard@holy7c12104 ~]$ mamba install -y numpy

# uninstall packages
(python_env1)[jharvard@holy7c12104 ~]$ mamba uninstall PACKAGE

# deactivate environment
(python_env1)[jharvard@holy7c12104 ~]$ conda deactivate
[jharvard@holy7c12104 ~]$
```

R with Spack

Same as other software with Spack:

- One-time setup
  - Clone Spack repo in your lab storage (better performance than home dir)
  - Source spack
  - Install packages with Spack - some software can take a few hours to build
- Job submission
  - Source spack
  - Load packages/software with Spack
  - Run code

Installing R packages with Spack

```
# request interactive job
[jharvard@rockylogin ~]$ salloc -p rocky --mem 12g -t 0-04:00 -c 8

# use lab storage
[jharvard@holy7c12104 ~]$ cd /n/holylabs/LABS/jharvard_lab/Lab/software/spack

# source spack
[jharvard@holy7c12104 spack]$ . share/spack/setup-env.sh

# install R packages with spack
[jharvard@holy2c02302 spack]$ spack install r-rgdal

# load spack packages
[jharvard@holy2c02302 spack]$ spack load r-rgdal

# launch R and load libraries
[jharvard@holy2c02302 spack]$ R
> library(rgdal)
```

Legacy CentOS 7 support

- **Last resort option**
- Singularity image with CentOS 7
  - same environment of compute nodes
  - can load CentOS 7 modules
  - cannot submit jobs inside the container
  - can modify CentOS 7 image by building a new container based on CentOS 7 image

```
[jharvard@holy7c12102 ~]$ singularity exec
/n/singularity_images/FAS/centos7/compute-el7-noslurm-2023-03-29.sif /bin/bash
Singularity> module load gcc
Singularity> module load matlab
Singularity> module list

Currently Loaded Modules:
  1) gmp/6.2.1-fasrc01   2) mpfr/4.1.0-fasrc01   3) mpc/1.2.1-fasrc01   4)
gcc/12.1.0-fasrc01   5) matlab/R2022b-fasrc01
```

Partitions on Rocky 8 Test Cluster

Partitions	Nodes	Cores per Node	CPU Core Types	Mem per Node (GB)	Time Limit	Max jobs	Max cores	MPI suitable	GPU capable
rocky	36	48	Intel “Cascade Lake”	184	3 days	none	none	Yes	No
rocky_gpu	48	64	Intel “Ice Lake”	499	3 days	none	none	Yes	Yes (28 A100 5GB MIG)

For more information about partition:

```
$ sinfo -p rocky
$ scontrol show partition rocky
```

FASSE and Open OnDemand

FASRC is working on

- FASSE test cluster
- Open OnDemand/VDI both on Cannon and FASSE clusters
- Timeline: mid May

Request Help - Resources

- https://docs.rc.fas.harvard.edu/kb/support/
  - Rocky 8 Transition Guide
    - https://docs.rc.fas.harvard.edu/kb/rocky-8-transition-guide/
  - Portal
    - http://portal.rc.fas.harvard.edu/rcrt/submit_ticket
  - Email
    - rchelp@rc.fas.harvard.edu
  - Office Hours
    - Wednesday noon-3pm https://harvard.zoom.us/j/255102481
  - Consulting Calendar
    - https://www.rc.fas.harvard.edu/consulting-calendar/
  - Training Calendar
    - https://www.rc.fas.harvard.edu/upcoming-training/