

# Introduction to Singularity/Apptainer Containers

(Tsai) Wei Wu





- <https://www.arch.jhu.edu/singularity-containers/>
- <https://singularity-tutorial.github.io>
- [https://epcced.github.io/2021-07-29\\_Singularity\\_Online/](https://epcced.github.io/2021-07-29_Singularity_Online/)
- Ticket  
**[help@rockfish.jhu.edu](mailto:help@rockfish.jhu.edu)**
  - User ID at Rockfish.
  - Detailed explanation of the problem/issue.
  - Add snapshots if possible.
  - Path to the working directory, scripts and (slurm) output files.



- Why do we need containers  
What are the use cases
- Containers Hub
- Singularity/Apptainer v.s. Docker
- Access Singularity on Rockfish
- Modify an existing container
- Build a container from scratch
- Modulefiles
- How to access a container in batch mode, for example QE.

# Why use containers



## Common cases:

- My application requires glibc 2.32, but the cluster only has 2.28, could you update glibc?
- My code worked last year, but it does not work now. I need to repeat the analysis for my paper.
- Could you give me sudo privilege? I need it to install my application.
- I can install the package easily on my laptop, why the installation on the cluster failed?

# Why use containers



```
ImportError: /lib64/libc.so.6: version `GLIBC 2.32' not found (required by /home/bsu11/thrust_2/OpenPCDet/tools/../pcdet/ops/iou3d_nms/iou3d_nms_cuda.cpython-38-x86_64-linux-gnu.so)

ERROR:torch.distributed.elastic.multiprocessing.api:failed (exitcode: 1) local_rank: 0 (pid: 399160) of binary: /home/bsu11/miniconda3/envs/openpcd/bin/python
Traceback (most recent call last):
  File "/home/bsu11/miniconda3/envs/openpcd/bin/torchrun", line 8, in <module>
    sys.exit(main())
  File "/home/bsu11/miniconda3/envs/openpcd/lib/python3.8/site-packages/torch/distributed/elastic/multiprocessing/errors/_init__.py", line 345, in wrapper
    return f(*args, **kwargs)
  File "/home/bsu11/miniconda3/envs/openpcd/lib/python3.8/site-packages/torch/distributed/run.py", line 724, in main
    run(args)
  File "/home/bsu11/miniconda3/envs/openpcd/lib/python3.8/site-packages/torch/distributed/run.py", line 715, in run
    elastic_launch()
  File "/home/bsu11/miniconda3/envs/openpcd/lib/python3.8/site-packages/torch/distributed/launcher/api.py", line 131, in __call__
    return launch_agent(self._config, self._entrypoint, list(args))
  File "/home/bsu11/miniconda3/envs/openpcd/lib/python3.8/site-packages/torch/distributed/launcher/api.py", line 245, in launch_agent
    raise ChildFailedError(
torch.distributed.elastic.multiprocessing.errors.ChildFailedError:
=====
test.py FAILED

Failures:
[1]:
time      : 2024-10-16_00:11:35
host      : icgpu05.cm.cluster
rank      : 1 (local_rank: 1)
exitcode  : 1 (pid: 399161)
error_file: <N/A>
traceback : To enable traceback see: https://pytorch.org/docs/stable/elastic/errors.html
=====
Root Cause (first observed failure):
[0]:
time      : 2024-10-16_00:11:35
host      : icgpu05.cm.cluster
rank      : 0 (local_rank: 0)
exitcode  : 1 (pid: 399160)
error_file: <N/A>
traceback : To enable traceback see: https://pytorch.org/docs/stable/elastic/errors.html
=====
(openpcd) [bsu11@icgpu05 tools]$ ldd --version
ldd (GNU libc) 2.28
Copyright (C) 2018 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
Written by Roland McGrath and Ulrich Drepper.
```

# Solution: Containers



- Self-contained
- Portable
- Reliability
- Integration with HPC systems
  - A container **image** is simply a file (or collection of files ) saved on disk that stores everything you need to run a target application or applications.
  - **Installing applications into containers is much easier than installing them directly into our clusters.**

¿ LIMITATIONS ?

# Container Hubs (Registry)



# Singularity/Apptainer v.s. Docker

- Singularity is an open source container platform
- Singularity is designed for ease-of-use on shared multiuser systems and in HPC environments.
- Singularity is better for command line applications and accessing devices like GPUs or MPI hardware without jumping through hoops.
  
- Docker requires superuser privileges (if not creating a user in the container)
- Docker is better at running applications on VM or cloud infrastructure.
- Docker's build tooling is still aces though!



podman

# Singularity command on RF compute nodes

The command is **NOT AVAILABLE** on login nodes  
(Interactive) jobs are required.

```
RF [twu73@login03 ~]$ interact -N1 -n4 -p express -t 30
Tasks:    4
Cores/task: 1
Total cores: 4
Walltime: 30
Reservation:
Queue:    express
Command submitted: salloc -J interact -N 1-1 -n 4 --time=30 -p express srun --pty bash
salloc: Pending job allocation 21125279
salloc: job 21125279 queued and waiting for resources
salloc: job 21125279 has been allocated resources
salloc: Granted job allocation 21125279
salloc: Nodes sr02 are ready for job
RF [twu73@sr02 ~]$ w
 15:44:06 up 201 days, 14:12,  0 users, load average: 115.70, 122.70, 105.19
USER   TTY      FROM          LOGIN@    IDLE    JCPU    PCPU WHAT
RF [twu73@sr02 ~]$
RF [twu73@sr02 ~]$ which singularity
/usr/bin/singularity
RF [twu73@sr02 ~]$
```

# singularity pull (download)

- <https://hub.docker.com/>
- search for “r 4.4.2” and choose r-base and click on “Tags”
- go to your data or scratch space and type



```
RF [twu73@c148 ~/scr4_rfadmin/tmp]$ singularity pull r-base-442.sif docker://r-base:4.4.2
INFO: Converting OCI blobs to SIF format
WARNING: 'nodev' mount option set on /tmp, it could be a source of failure during build process
INFO: Starting build...
Getting image source signatures
Copying blob eb787fd8ee11 done
Copying blob 7e05d5a5442b done
Copying blob 2ab487ada242 done
Copying blob ed39eedd8d73 done
Copying blob 8849a3fc8a52 done
Copying blob 3b47bf5fe50f done
Copying config f4bd7c03eb done
Writing manifest to image destination
Storing signatures
2025/02/07 08:51:11 info unpack layer: sha256:eb787fd8ee11f9db51163f32c29c012215dab8f556e509fd8d9baeffae26cf2
2025/02/07 08:51:13 info unpack layer: sha256:7e05d5a5442bbb7d8a889e0be733c0642c21fcfd1e47d946939ab4dcfd273ec9fd
2025/02/07 08:51:13 info unpack layer: sha256:8849a3fc8a52cd74914addf49a9c467b761a7c9a9097d89e6a1db86750ea59d0
2025/02/07 08:51:13 info unpack layer: sha256:2ab487ada242ef09da4f9d6b13215e848d85b52a8cacc478b42d1232c6135375
2025/02/07 08:51:13 info unpack layer: sha256:3b47bf5fe50f28ea63431e38133306e528f19c4cf4c47235fadf00aa3d4b5c80
2025/02/07 08:51:13 info unpack layer: sha256:ed39eedd8d73b7a8fd42602e390369ecbb23a54138f45905221f696ea750e82b
INFO: Creating SIF file...
RF [twu73@c148 ~/scr4_rfadmin/tmp]$ ls r-base-442.sif
r-base-442.sif
RF [twu73@c148 ~/scr4_rfadmin/tmp]$
```

Caches are stored in \$HOME/.singularity/cache  
SINGULARITY\_CACHEDIR

# singularity shell

- go into the container and run commands interactively  
\$ singularity shell r-base-442.sif
- the prompt changes to indicate it's inside the container

```
RF [twu73@sr01 ~/scr4_rfadmin]$ singularity shell r-base-442.sif
Singularity>
```

- we can navigate the environment by going into the root folder.

```
Singularity> cd /
Singularity> ls
bin  dev      home  media  proc  sbin  singularity  tmp  vast
boot environment lib    mnt   root  scratch16  srv   usr
data  etc      lib64  opt    run   scratch4   sys   var
Singularity>
```

# singularity shell (cont'd)

- check what R packages are installed in the container in /lib/R/library
- launch R

```
Singularity> R  
R version 4.4.2 (2024-10-31) -- "Pile of Leaves"  
Copyright (C) 2024 The R Foundation for Statistical Computing  
Platform: x86_64-pc-linux-gnu  
  
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.  
  
Natural language support but running in an English locale  
  
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.  
> library(lattice)  
>
```

- press CTRL-D to exit the container

# Modifying the container

- convert the image file into a sandbox with the flag —sandbox

```
RF [twu73@sr01 ~/scr4_rfadmin/tmp]$ singularity build --sandbox r-base-442-sandbox r-base-442.sif
INFO: Starting build...
INFO: Verifying bootstrap image r-base-442.sif
WARNING: integrity: signature not found for object group 1
WARNING: Bootstrap image could not be verified, but build will continue.
INFO: Creating sandbox directory...
INFO: Build complete: r-base-442-sandbox
RF [twu73@sr01 ~/scr4_rfadmin/tmp]$ ls
r-base-442-sandbox  r-base-442.sif
RF [twu73@sr01 ~/scr4_rfadmin/tmp]$
```

- go into the folder with —writable flag

```
RF [twu73@sr01 ~/scr4_rfadmin/tmp]$ singularity shell --writable r-base-442-sandbox/
WARNING: Skipping mount /data [binds]: /data doesn't exist in container
WARNING: Skipping mount /scratch4 [binds]: /scratch4 doesn't exist in container
WARNING: Skipping mount /scratch16 [binds]: /scratch16 doesn't exist in container
Singularity>
```

- we can now install R packages, e.g. Bioconductor packages.

# Modifying the container (cont'd)

- install BiocManager in R

```
Singularity> R

R version 4.4.2 (2024-10-31) -- "Pile of Leaves"
Copyright (C) 2024 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> if (!require("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
Installing package into '/usr/local/lib/R/site-library'
(as 'lib' is unspecified)
```

- install S4Vectors

```
> BiocManager::install("S4Vectors")
'getOption("repos")' replaces Bioconductor standard repositories
'help("repositories", package = "BiocManager")'
Replacement repositories:
  CRAN: https://cloud.r-project.org
Bioconductor version 3.20 (BiocManager 1.30.25)
Installing package(s) 'BiocVersion', 'S4Vectors'
also installing the dependency 'BiocGenerics'

trying URL 'https://bioconductor.org/packages/3.20/bioc/src/contrib/Archive/S4Vectors_1.30.25.tar.gz'
Content type 'application/x-gzip' length 57066
=====
downloaded 55 KB

trying URL 'https://bioconductor.org/packages/3.20/bioc/src/contrib/Archive/BiocVersion_1.30.25.tar.gz'
Content type 'application/x-gzip' length 983 bytes
=====
downloaded 983 bytes
```

- quit out of the container and go back in, the package should be still there.

# Workflow



1. singularity pull a base image  
\$ singularity pull docker://r-base:4.4.2
2. convert it into sandbox  
\$ singularity build --sandbox r-base-sandbox r-base\_4.4.2.sif
3. shell into the sandbox with --writable flag  
\$ singularity shell --writable r-base-sandbox
4. modify the sandbox
5. convert the sandbox back into image  
\$ singularity build r-new.sif r-base-sandbox

# Build a container from scratch

- Need a definition file
- Also need sudo privilege

```
Bootstrap: library
From: ubuntu:22.04

%post
    apt-get -y update
    apt-get -y install cowsay lolcat

%environment
    export LC_ALL=C
    export PATH=/usr/games:$PATH

%runscript
    date | cowsay | lolcat

%labels
    Author Sylabs
```

```
RF [twu73@sr01 ~/scr4_rfadmin/tmp]$ 
RF [twu73@sr01 ~/scr4_rfadmin/tmp]$ singularity build ubuntu.sif ubuntu-22.04.def
FATAL: --remote, --fakeroot, or the proot command are required to build thisRRR
RF [twu73@sr01 ~/scr4_rfadmin/tmp]$
```

- Options: install docker/singularity in your laptop
  - Windows -> Virtual Machine or Windows Sub-Linux System
  - MAC -> should be easier than windows

<https://pmitev.github.io/UPPMAX-Singularity-workshop/installation/>  
<https://tinyurl.com/install-singularity>

<https://singularity-tutorial.github.io/06-runscript/>  
[https://docs.sylabs.io/guides/latest/user-guide/definition\\_files.html](https://docs.sylabs.io/guides/latest/user-guide/definition_files.html)

# Build a container from scratch (cont'd)

on my own laptop, I installed singularity in WSL and I can build from definition file.

```
root@DESKTOP-UCTGA49:~/tmp# which singularity
/usr/bin/singularity
root@DESKTOP-UCTGA49:~/tmp# singularity build ubuntu.sif ubuntu.def
INFO: Starting build...
INFO: Downloading library image
5.0MiB / 28.4MiB [=====>-----]
--] 18 % 753.6 Ki5.0MiB / 28.4MiB [=====>-----]
-----] 18 % 7065.0MiB / 28.4MiB [=====>-----]
-----] 1810.0MiB / 28.4MiB [=====>-----]
28.4MiB / 28.4MiB [=====>-----] 100 % 1.8 MiB/s 0s
INFO: Verifying bootstrap image /root/.singularity/cache/library/sha256.7a63c14842
a5c9b9c0567c1530af87afbb82187444ea45fd7473726ca31a598b
INFO: Running post scriptlet
+ apt-get -y update
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [45.
kB]
```



# Modulefiles

we can help generate the modulefiles if you have built your own container.



```
RF [twu73@sr01 ~]$ ml spider HOOMD-blue/3.9.0-cpu
-----
HOOMD-blue: HOOMD-blue/3.9.0-cpu
-----
Properties:
  Singularity container
  This module can be loaded directly: module load HOOMD-blue/3.9.0-cpu
  Help:
    This module loads HOOMD-blue
    Documentation can be found at: https://hoomd-blue.readthedocs.io/en/v3.9.0/
      python3

RF [twu73@sr01 ~]$ ml show HOOMD-blue/3.9.0-cpu
-----
  /data/apps/lmod/linux-centos8-x86_64/ext/HOOMD-blue/3.9.0-cpu.lua:
  -----
help([[This module loads HOOMD-blue
Documentation can be found at: https://hoomd-blue.readthedocs.io/en/v3.9.0/
python3
]])
whatis("HOOMD-blue is a Python package that runs simulations of particle systems on CPUs. It performs hard particle Monte Carlo simulations of a variety of shape classes and molecular dynamics simulations of particles with a range of pair, bond, angle, and other potentials. Many features are targeted at the soft matter research community, though the code is general and capable of many types of particle simulations.")
add_property("environ","container")

RF [twu73@sr01 ~]$
```

# Example of running a batch job with singularity container

```
#!/bin/bash

#SBATCH -N1
#SBATCH -n4
#SBATCH --cpus-per-task=1
#SBATCH --gres=gpu:2
#SBATCH -p a100
#SBATCH -q qos_gpu
#SBATCH -t 00:20:00

TOPDIR=~/scr16_rfadmin/tmp
SIF='quantum_espresso_qe-7.3.1.sif'

cd $TOPDIR
singularity pull docker://nvcr.io/hpc/quantum_espresso:qe-7.3.1
mkdir -p ausurf && cd ausurf

wget https://repository.prace-ri.eu/git/UEABS/ueabs/-/raw/master/quantum_espresso/test_cases/small/Au.pbe-nd-van.UPF
wget https://repository.prace-ri.eu/git/UEABS/ueabs/-/raw/master/quantum_espresso/test_cases/small/ausurf.in

singularity run --nv ./SIF \
    mpirun -np ${SLURM_NTASKS} \
    pw.x -input ausurf.in -npool 2
```

[https://catalog.ngc.nvidia.com/orgs/hpc/containers/quantum\\_espresso](https://catalog.ngc.nvidia.com/orgs/hpc/containers/quantum_espresso)

<https://researchcomputing.princeton.edu/support/knowledge-base/quantum-espresso>

# Singularity path binds

## Specifying bind paths

Here's an example of using the `--bind` option and binding `/data` on the host to `/mnt` in the container (`/mnt` does not need to already exist in the container):

```
$ ls /data  
bar  foo  
  
$ singularity exec --bind /data:/mnt my_container.sif ls /mnt  
bar  foo
```

# Questions?

THANK YOU.

[help@rockfish.jhu.edu](mailto:help@rockfish.jhu.edu)