

Singularity Container

Yi Kuo @ HPC-I



Containers - Not only Docker!

- Docker is a commercial product by Docker, Inc., which uses the container technology
- There are other solutions
 - Podman - For rootless, daemonless docker
 - LXC - For system containers
 - Kubernetes - For container orchestration
 - **Singularity - For HPC**



Why Singularity in HPC (instead of Docker)

- Rootless
 - On most HPC systems, user does not have root access or UID mapping permissions
- Integrates with host MPI & Scheduler
 - So that applications can utilize the networking on the host for optimal performance & scale across multiple nodes in the cluster
- Portability
 - Singularity image is a single file, which can be transferred easily



Build a Singularity Image

```
$ sudo singularity build lolcow.sif docker://godlovedc/lolcow
```

- There are two types of images
 - Compressed read-only **Singularity Image File (.sif)**
 - Writable chroot **sandbox** directory for interactive development
 - With the `--sandbox` option
- Root permission is (mostly) required while building images
 - Build locally, and transfer to the HPC system
- https://docs.sylabs.io/guides/3.5/user-guide/build_a_container.html



Build a Singularity Image

- Images can be built from
 - Docker Hub (docker://)
 - Singularity container library (container://[repo])
 - Singularity Hub (shub://)
 - Singularity's official repo
 - Local Docker Image
 - Singularity Definition File
 - Singularity's version of Dockerfile
 - Local .sif file (to convert to sandbox)
 - Local sandbox directory (to convert to .sif)



Running Singularity

```
$ singularity run lolcow.sif
```

```
team1@comp1:~$ singularity run lolcow.sif
```

```
-----  
/ Never look up when dragons fly \  
\ overhead.                          /  
-----
```

```
      ^__^  
      (oo)\_____  
      (__)\\       )\\/\  
           ||----w |  
           ||     ||
```



Building an MPI Program in Singularity

```
# mpi_test.def
Bootstrap: docker
From: ubuntu:22.04

%setup
    wget http://140.114.91.164/sc24/mpi_test.c -O mpi_test.c

%files
    ./mpi_test.c /app/mpi_test.c

%post
    apt-get -y update
    apt-get -y install openmpi-bin openmpi-common libopenmpi-dev gcc
    mpicc /app/mpi_test.c -o /app/mpi_test

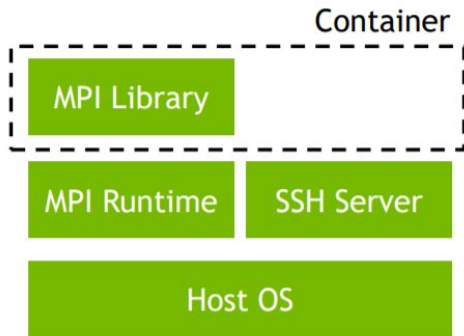
%runscript
    /app/mpi_test
```

```
$ sudo singularity build mpi_test.sif mpi_test.def
```



Running MPI with Singularity

“Outside-in”

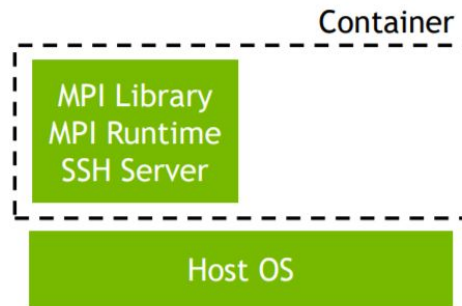


mpirun is invoked outside the container

```
$ mpirun singularity run ...
```

Application could integrate
with host's MPI & scheduler

“Inside-out”



mpirun is invoked inside the container

```
$ singularity run mpirun ...
```



Running MPI with Singularity

```
$ mpirun -host comp1:5,comp2:5 singularity run mpi_test.sif
```

```
team1@comp1:~$ mpirun -host comp1:5,comp2:5 singularity run mpi_test.sif
Hello world: rank 0 of 10 running on comp1
Hello world: rank 1 of 10 running on comp1
Hello world: rank 2 of 10 running on comp1
Hello world: rank 3 of 10 running on comp1
Hello world: rank 4 of 10 running on comp1
Hello world: rank 5 of 10 running on comp2
Hello world: rank 6 of 10 running on comp2
Hello world: rank 7 of 10 running on comp2
Hello world: rank 8 of 10 running on comp2
Hello world: rank 9 of 10 running on comp2
```



Exploring the container

```
$ singularity shell mpi_test.sif
```

- The files in the container overlays on the host's file system

```
team1@comp1:~$ ls /
bin    dev    home  lib32  libx32  media  opt    root  sbin  srv  tmp  var
boot  etc    lib   lib64  lost+found  mnt    proc   run   snap  sys  usr
team1@comp1:~$ singularity shell mpi_test.sif
Singularity> ls / --color
app    boot  environment  home  lib32  libx32  mnt  proc  run  singularity  sys  usr
bin    dev    etc          lib   lib64  media  opt  root  sbin  srv          tmp  var
```



HPC Container Registries

- There are some containers with optimized binaries provided by the vendors
- [NVIDIA NGC Containers](#)
 - [NGC Benchmarks \(HPL, HPCG, etc.\)](#)
 - [GROMACS](#)
 - [LAMMPS](#)
- [AMD Infinity Hub](#)



Demo - Running GROMACS from NGC on TWCC

```
#!/bin/bash

#SBATCH --job-name=GromacsRun
#SBATCH --nodes=1
#SBATCH --cpus-per-task=4
#SBATCH --time=00:10:00
#SBATCH --account=ACD110018
#SBATCH --partition=gpld
#SBATCH --gres=gpu:8
#SBATCH --ntasks-per-node=8

export GMX_ENABLE_DIRECT_GPU_COMM=1

SINGULARITY="singularity run --nv -B ${PWD}:/host_pwd --pwd /host_pwd
docker://nvcr.io/hpc/gromacs:2023.2"

$SINGULARITY \
    gmx mdrun -v -s stmv.tpr -ntmpi 8 -ntomp 4 \
        -nb gpu -pme gpu -bonded gpu -update gpu -npme 1 -pin on \
        -nsteps 30000 -resetstep 28000 -noconfout -nstlist 300 \
        -dlb no -gpu_id 01234567
```

