GPU ACCELERATED MULTI-NODE HPC WORKLOADS WITH SINGULARITY

December 2018

AGENDA

What are containers?

Pulling containers

Running multi-node workloads

Building multi-node containers

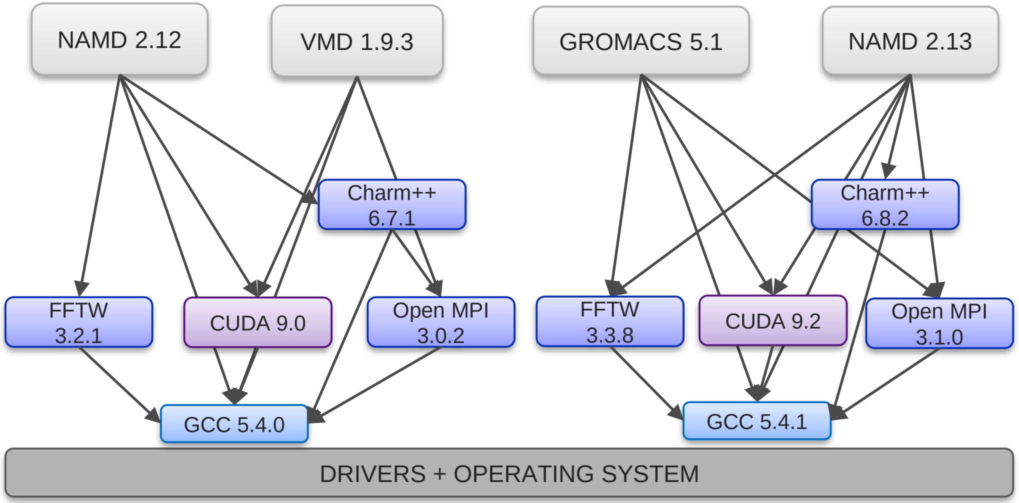
WHAT ARE CONTAINERS?

- Isolation technology based on Linux kernel namespaces
- Package everything needed to run an application
- Differ from virtualization
    - Containers run on common kernel as host
    - OS virtualization vs hardware abstraction
    - Containers are generally more lightweight and offer better performance than VMs
- Container runtimes Charlie Cloud, Docker, Shifter, Singularity, and more
    - NGC HPC containers are QAed with Docker and Singularity

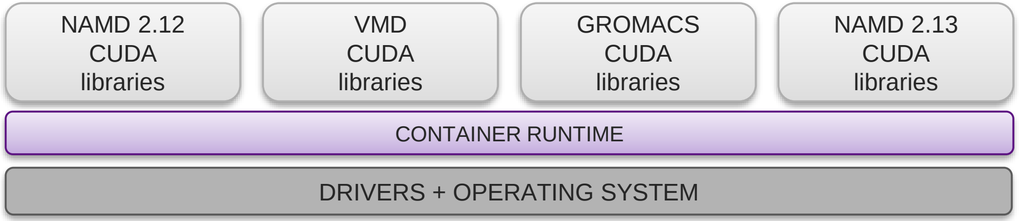
CONTAINER BENEFITS

- Enabling straddling of distros on a common Linux kernel
- Isolate environment and resources
- Encapsulate dependencies
- Straightforward deployment
- Drop in replacement for many workflows
- Promote reproducibility
- Equivalent performance to baremetal

BARE METAL VS CONTAINERS



BARE METAL



CONTAINERS

CONTAINER REGISTRIES

- Docker Hub - https://hub.docker.com
  - Official repositories for CentOS, Ubuntu, and more
  - NVIDIA: https://hub.docker.com/r/nvidia/cuda
- Singularity Hub - https://singularity-hub.org/
  - Registry of scientific Linux containers
- NVIDIA GPU Cloud (NGC) - https://ngc.nvidia.com
  - Optimized HPC, HPC Visualization, Deep Learning, and base containers
  - User Guide: http://docs.nvidia.com/ngc/ngc-user-guide/index.html

NGC CONTAINER REGISTRY

Over 40 containers available today

Deep Learning

caffe

caffe2

cntk

cuda

digits

inferenceserver

mxnet

pytorch

tensorflow

tensorrt

tensorrtserver

theano

torch

HPC

bigdft

candle

chroma

gamess

gromacs

lammps

lattice-microbes

milc

namd

pgi

picongpu

qmcpack

relion

HPC Visualization

index

paraview-holodeck

paraview-index

paraview-optix

vmd

RAPIDS/ML

rapidsai

NVIDIA/K8s

Kubernetes
on NVIDIA GPUs

Partners

chainer

deep-learning-studio

h20ai-driverless

kinetica

mapd

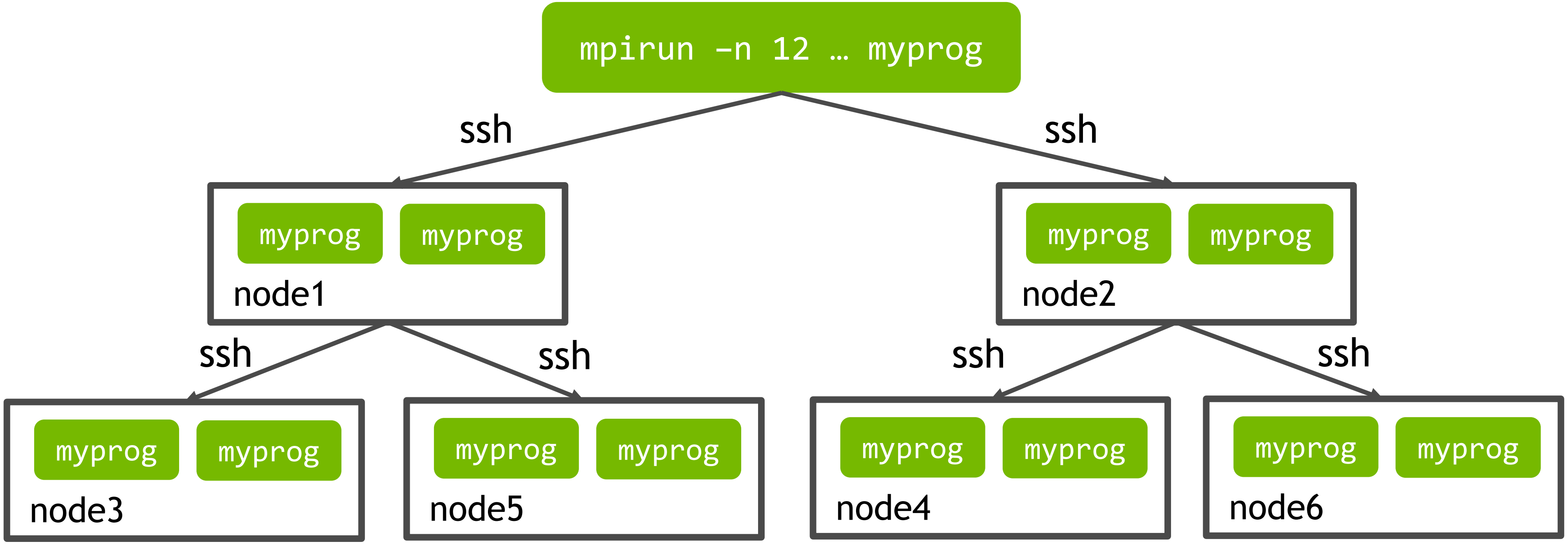
matlab

paddlepaddle

MULTI-NODE

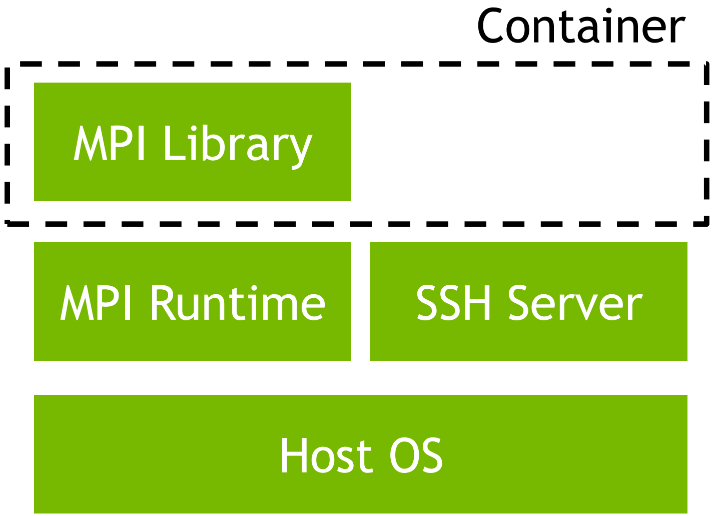
MPI BACKGROUND

MPI implementations provide a job launcher, mpirun or mpiexec, that initializes and wires up distributed MPI ranks (i.e., processes) on a multi-node cluster



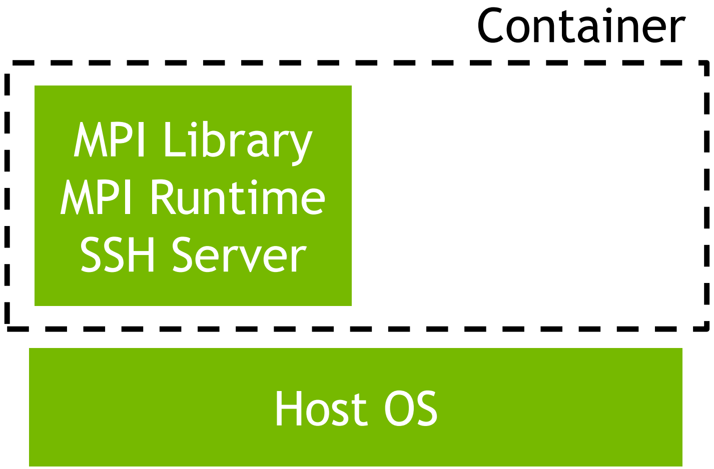
MPIRUN + CONTAINERS

“Outside-in”



mpirun is invoked outside the container

“Inside-out”



mpirun is invoked inside the container

MPIRUN + CONTAINERS

- “Outside-in”
  - Fits in more “naturally” into the traditional HPC workflow (SSH keys, etc.)
  - mpirun -hostfile hostfile –n 64 app
    becomes
    mpirun –hostfile hostfile –n 64 singularity run app.simg app
  - Requires a compatible MPI runtime on the host
- “Inside-out”
  - Must insert SSH keys into the container image by some other mechanism
  - Must orchestrate the launch of containers on other hosts
  - Completely self-contained, no host MPI dependencies

MULTI-NODE OUTSIDE-IN MILC RUN

On the cluster

Get the sample dataset

```
$ mkdir $HOME/milc-dataset && cd $HOME/milc-dataset
$ wget http://denali.physics.indiana.edu/~sg/SC15_student_cluster_competition/benchmarks.tar
$ tar –xf benchmarks.tar
```

Pull MILC container from NGC

```
$ module load singularity
$ singularity build milc.simg docker://nvcr.io/hpc/milc:quda0.8-patch4Oct2017
```

Get a 2 node allocation

Run the container using 2 nodes with 4 GPUs per node

```
$ module load openmpi
$ mpirun -n 8 -npernode 4 –wdir $HOME/milc-dataset/small singularity run --nv ~/milc.simg
  /milc/milc_qcd-7.8.1/ks_imp_rhmc/su3_rhmd_hisq -geom 1 1 2 4 small.bench.in
…
```

MULTI-NODE SLURM MILC RUN

On the cluster

Get the sample dataset

```
$ mkdir $HOME/milc-dataset && cd $HOME/milc-dataset
$ wget http://denali.physics.indiana.edu/~sg/SC15_student_cluster_competition/benchmarks.tar
$ tar –xf benchmarks.tar
```

Pull MILC container from NGC

```
$ module load singularity
$ singularity build milc.simg docker://nvcr.io/hpc/milc:quda0.8-patch4Oct2017
```

Run the container using 2 nodes with 8 GPUs per node

```
$ srun --nodes=2 --ntasks-per-node=8 --mpi=pmi2 singularity run --pwd $HOME/milc-dataset/small --nv
milc.simg su3_rhmd_hisq -geom 1 2 2 4 small.bench.in
```

GENERIC MULTI-NODE SLURM RUN

On the cluster

Pull container from NGC

```
$ module load singularity
$ singularity build myapp.simg docker://nvcr.io/hpc/myapp:tag
```

Run the container using 2 nodes with 8 GPUs per node

```
$ srun --nodes=2 --ntasks-per-node=8 --mpi=pmi2 singularity run --nv myapp.simg myapp
```

DEMO

BUILDING MULTI-NODE CONTAINERS

- Know your target hardware and software configurations
    - If possible, build on your target hardware
- Use multi stage builds to minimize the size of your final container image
    - Don’t include unneeded libraries
    - To get this advantage with Singularity, build a Docker image and convert it to Singularity
- Host integration vs. portability trade off

FOR BEST INTEGRATION

- Exactly match InfiniBand userspace component versions
    - (M)OFED version should match host
    - If available, nv_peer_mem, gdr_copy, and xpmem/knem should match host
- Exactly match host MPI flavor and version
    - Should match configure options as well

FOR BEST PORTABILITY

- (M)OFED drivers
  - MOFED 4.4+ will maintain forwards/backwards compatibility
  - Otherwise, OFED drivers generally have fewer compatibility issues than MOFED drivers but you will lose out on some features
- Use OpenMPI
  - "Plugin" design can support many systems with choices delayed until runtime
  - Can build support for lots of transport backends, resource managers, filesystem support, etc in a single build
  - If possible, use 3.x or 4.x for best compatibility

FOR BEST PORTABILITY CONT’D

- Use UCX
  - Replaces deprecated openIB OpenMPI component
  - UCX is default starting with OpenMPI 4.0
  - Supports intra/inter node optimized transports
  - When built with nv_peer_mem, gdr_copy, knem, xpmem, cma it will automatically pick the best backend based on host support

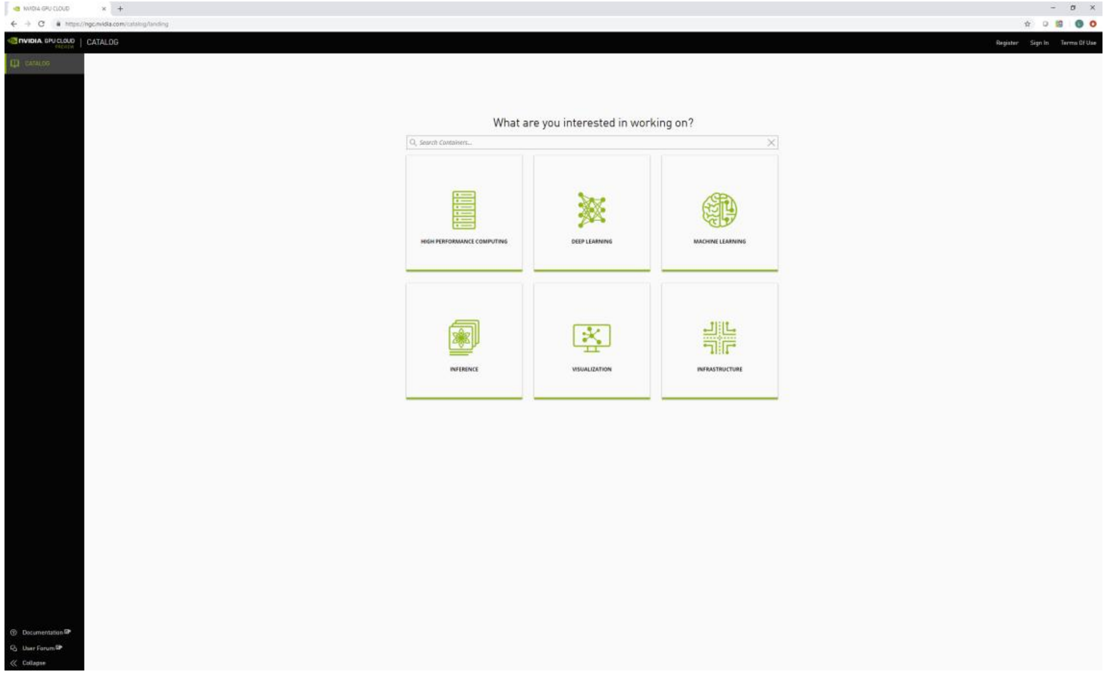
HPC CONTAINER MAKER (HPCCM)

- Simplifies the creation of container specification files
- Building block abstraction of components from implementation
  - Best practices for free
  - Updates to building blocks can be leveraged with a re-build
- Full power of Python in container recipes
- User arguments allow a single recipe to produce multiple containers

For more information on HPCCM, reference the “Containers Made Easy with HPC Container Maker” webinar or view the project’s README and source at https://github.com/NVIDIA/hpc-container-maker

GET STARTED TODAY WITH NGC

Sign Up and Access Containers for Free



What are you interested in working on?

To learn more about all of the GPU-accelerated software from NGC, visit:

nvidia.com/cloud

To sign up or explore NGC, visit:

ngc.nvidia.com

NVIDIA®