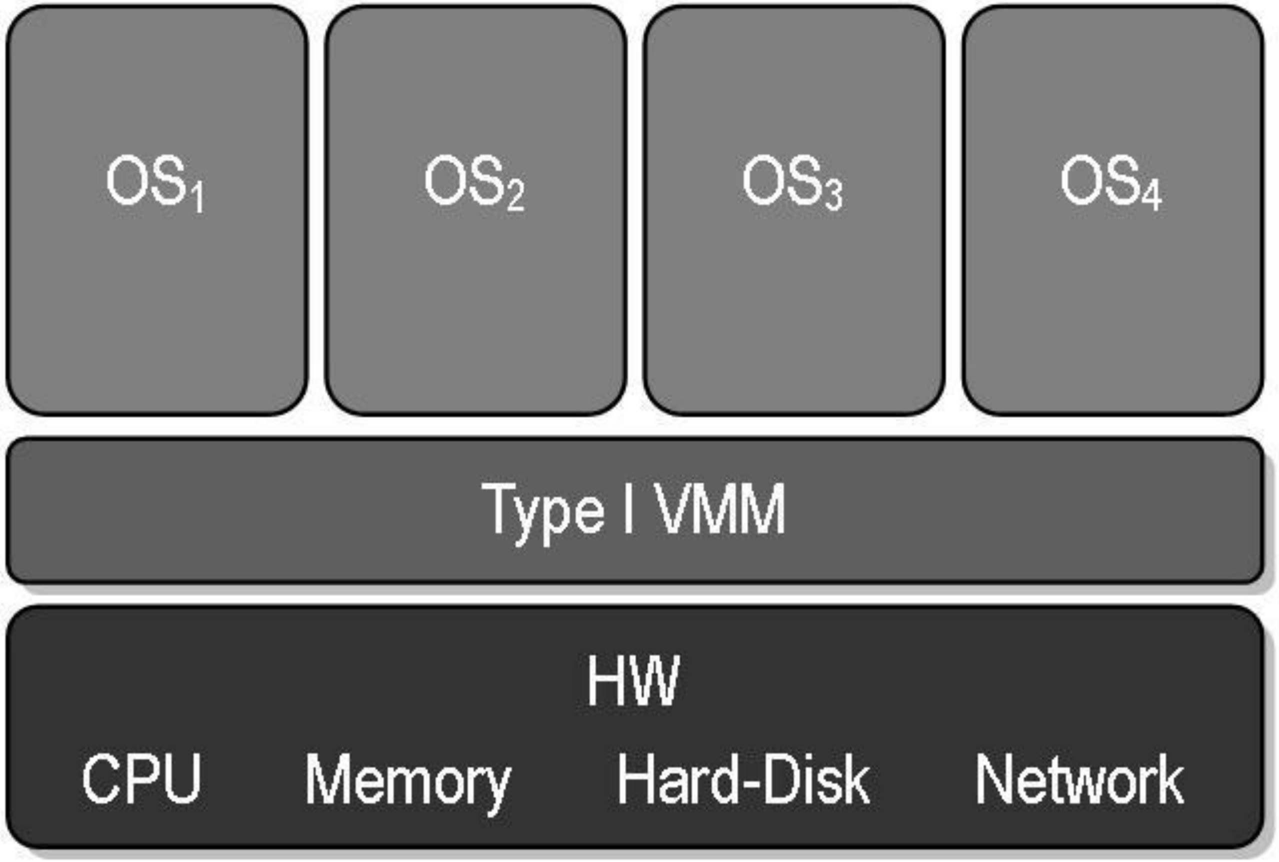
Using Singularity Containers in Virtual Machines and HPC

Venkat Mahadevan
UBC Advanced Research Computing (ARC)
5/26/2021

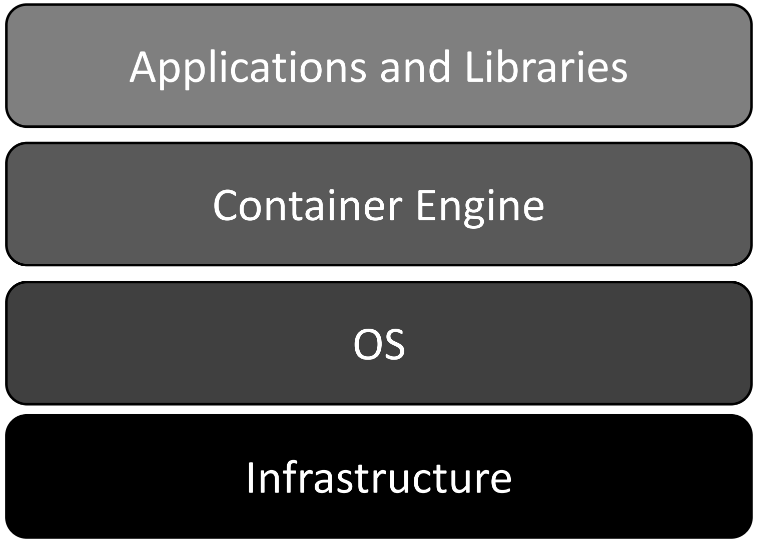
Virtualization



Containers

Containers

- Containers provide an additional layer of abstraction over virtualization.



Containers

- However, containers do not require virtual machines and can happily run on bare-metal.
- Lightweight compared to VMs in terms of memory and storage requirements.
- Increases portability of applications between datacenters, private clouds, and public clouds.

Containers

- Can provide a very customized environment specific to research use cases.

- Commercial software that requires old, obsolete libraries or operating system versions.

- Bottom line: use containers where it makes sense to your workflow.

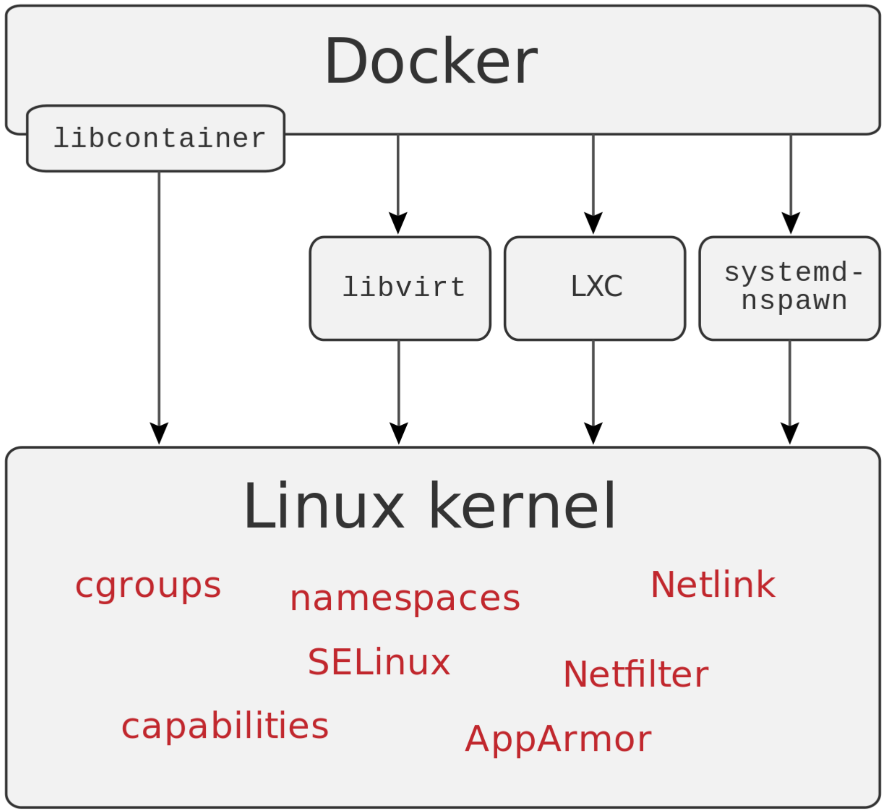
Why Use Containers

- Reproducibility.
- Portability.
- Isolation.
- Avoiding complexity aka dependency hell.

Docker

- The advent of Docker resulted in widespread adoption of container technology.
- Uses cgroups, kernel namespaces, OverlayFS (union filesystem) to allow containers to run in a singular Linux instance.
- Very mature technology with ~7 million DockerHub users.

Docker


Docker
libcontainer
libvirt
LXC
systemd-nspawn
Linux kernel
cgroups
namespaces
Netlink
SELinux
Netfilter
capabilities
AppArmor

Singularity

- However, Docker is not a perfect fit in the advanced research computing space.

- The Docker daemon needs to run in the background with elevated privileges on every node (in the datacenter, private/public Cloud VM, etc.) that hosts containers.

- This is a security concern which can be mitigated but limits some features.

Singularity

- Docker also does not provide close integration with the standard HPC software stack such as MPI as well as schedulers and resource managers such as Slurm & PBS Pro.

- Container orchestration technologies such as Kubernetes can replicate some of the aforementioned functionality but the existing HPC stack is well entrenched and robust.

Singularity features

- Was developed at Lawrence Berkeley National Lab specifically for HPC workloads.

- No daemon process required; an executable is provided.

- The user permissions are maintained both in and outside the container; mitigates privilege escalation concerns.

Singularity features

- Support for both NVIDIA and AMD GPUs: pass the --nv or --rocm flag to the singularity command line.

- The host and container must have compatible versions of the GPU drivers and libraries installed.

- Support for MPI via a couple of different models: the hybrid model and the bind model.

Singularity features

- MPI via the Hybrid model:
  - Both the host and the container must have compatible versions of MPI libraries and runtime installed.
  - Simplifies the running of MPI applications (just run the singularity command after mpirun or srun if using Slurm).

- MPI via the Bind model:
  - Mount the MPI implementation of the host in the container.

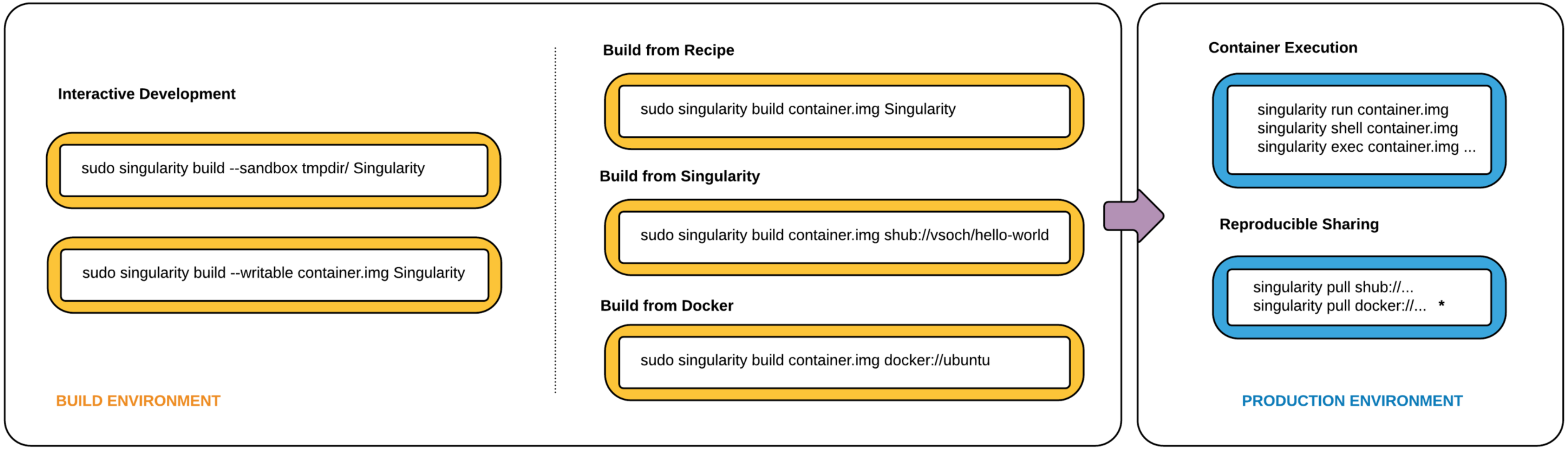
- https://sylabs.io/guides/3.7/user-guide/mpi.html

What about performance?

- On bare-metal, there is almost no performance gap between running in a Singularity container vs. directly on the host system.

- Papers benchmarking the performance of Singularity with various applications have found near native performance (within 1-2% performance) on bare-metal HPC systems:
  - https://ieeexplore.ieee.org/abstract/document/8855563
  - https://people.csail.mit.edu/dpaz/img/PDF/a065-LePaz.pdf

Typical workflow



* Docker construction from layers not guaranteed to replicate between pulls

DockerHub

tensorflow/tensorflow ☆

Pulls **50M+**

By tensorflow • Updated 44 minutes ago

Official Docker images for the machine learning framework TensorFlow (http://www.tensorflow.org)

Container

Overview | Tags

Filter Tags

Sort by Newest

TAG

nightly-gpu-jupyter

Last pushed **44 minutes ago** by tensorflowpackages

docker pull tensorflow/tensorflow:night...

DIGEST	OS/ARCH	COMPRESSED SIZE
33f893623392	linux/amd64	2.65 GB
33f893623392	linux/amd64	2.65 GB

Virtual Machines and HPC

- There is always going to be a performance impact of virtualization vs bare-metal.

- However, running containers on virtual machines and whole HPC clusters on VMs and Clouds is gaining popularity.

- Performance impacts can be mitigated by tuning VMs and underlying hardware (e.g. AWS Elastic Network/Fabric Adapter).

Deep Learning with Horovod

- Horovod is a distributed deep learning training framework for TensorFlow, Keras, PyTorch, and Apache MXNet.
- https://github.com/horovod/horovod/
- The stated goal is to take single-GPU training and scale it across many GPUs in parallel.

Distributed training example

- Build singularity container using DockerHub:

```
singularity pull horovod.sif docker://horovod/horovod:sha-40cbc9b
```

- Run on a single GPU:

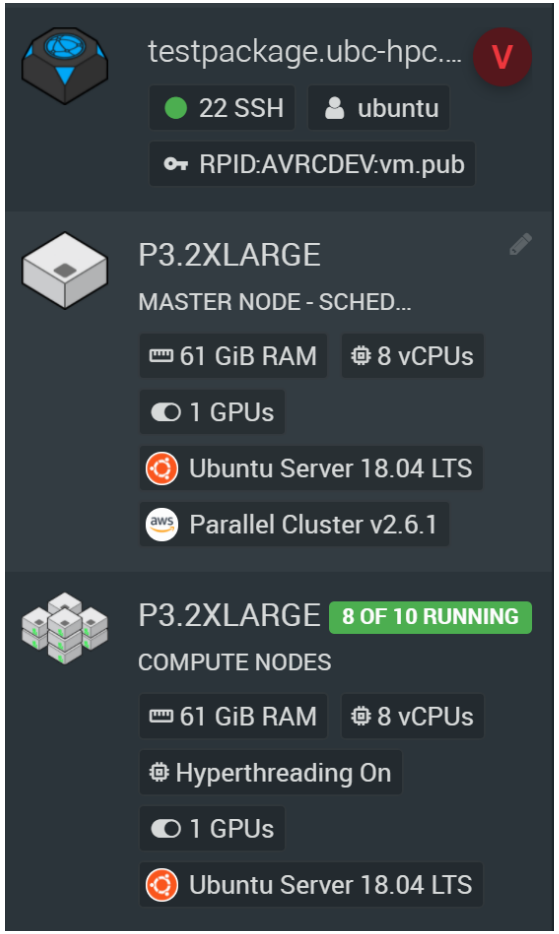
```
singularity exec --nv horovod.sif horovodrun -np 1 -H localhost:1
python ~/pytorch_imagenet_resnet50.py --train-dir ~/tiny-imagenet-
200/train --val-dir ~/tiny-imagenet-200/val
```

Distributed training example

- Will run the example on a virtual HPC on AWS Parallel Cluster.

- 8 GPU nodes with V100 running on AWS EC2 P3 instances.

- Slurm scheduler with Open MPI.



Distributed training example

```
2021-05-26 14:16:22.116100: I tensorflow/stream_executor/platform/default/dso_loader.cc:53] Successfully opened dynamic library libcudart.so.11.0
Train Epoch     #7:   1%|          | 3/391 [00:02<04:36,  1.40it/s, loss=3.4, accuracy=22.9]
Train Epoch     #7:   2%|▏         | 6/391 [00:03<02:58,  2.16it/s, loss=3.5, accuracy=21.4]
Train Epoch     #7:   2%|▏         | 9/391 [00:05<02:30,  2.53it/s, loss=3.46, accuracy=22]
Train Epoch     #7:   3%|▎         | 12/391 [00:06<02:20,  2.69it/s, loss=3.46, accuracy=23.1]
Train Epoch     #7:   4%|▍         | 16/391 [00:07<02:15,  2.76it/s, loss=3.45, accuracy=23.1]
Train Epoch     #7:   5%|▍         | 19/391 [00:08<02:17,  2.71it/s, loss=3.45, accuracy=23.4]
Train Epoch     #7:   6%|▌         | 22/391 [00:09<02:18,  2.66it/s, loss=3.43, accuracy=23.7]
Train Epoch     #7:   6%|▋         | 25/391 [00:11<02:16,  2.68it/s, loss=3.43, accuracy=23.6]
Train Epoch     #7:   7%|▋         | 28/391 [00:12<02:17,  2.64it/s, loss=3.44, accuracy=23.4]
Train Epoch     #7:   8%|▊         | 31/391 [00:13<02:14,  2.67it/s, loss=3.44, accuracy=23.6]
Train Epoch     #7:   9%|▉         | 35/391 [00:14<02:10,  2.74it/s, loss=3.43, accuracy=23.7]
Train Epoch     #7:  10%|▉         | 38/391 [00:15<02:10,  2.71it/s, loss=3.41, accuracy=24.1]
Train Epoch     #7:  10%|█         | 41/391 [00:16<02:14,  2.60it/s, loss=3.4, accuracy=24.2]
Train Epoch     #7:  11%|█▏        | 44/391 [00:18<02:21,  2.45it/s, loss=3.4, accuracy=24.3]
Train Epoch     #7:  12%|█▏        | 47/391 [00:19<02:12,  2.59it/s, loss=3.41, accuracy=24.2]
Train Epoch     #7:  13%|█▎        | 50/391 [00:20<02:17,  2.48it/s, loss=3.41, accuracy=24.2]
Train Epoch     #7:  14%|█▍        | 54/391 [00:21<02:08,  2.62it/s, loss=3.4, accuracy=24.3]
Train Epoch     #7:  15%|█▍        | 57/391 [00:23<02:10,  2.56it/s, loss=3.41, accuracy=24.2]
Train Epoch     #7:  15%|█▌        | 60/391 [00:24<02:09,  2.56it/s, loss=3.41, accuracy=24.3]
Train Epoch     #7:  16%|█▌        | 63/391 [00:25<02:08,  2.55it/s, loss=3.41, accuracy=24.2]
Train Epoch     #7:  17%|█▋        | 66/391 [00:26<02:04,  2.60it/s, loss=3.41, accuracy=24.2]
Train Epoch     #7:  18%|█▊        | 69/391 [00:28<02:03,  2.60it/s, loss=3.41, accuracy=24.3]
Train Epoch     #7:  18%|█▊        | 72/391 [00:29<01:57,  2.72it/s, loss=3.4, accuracy=24.3]
Train Epoch     #7:  19%|█▉        | 75/391 [00:30<01:53,  2.78it/s, loss=3.4, accuracy=24.4]
Train Epoch     #7:  20%|█▉        | 78/391 [00:31<01:46,  2.95it/s, loss=3.39, accuracy=24.3]
Train Epoch     #7:  21%|██        | 82/391 [00:32<01:50,  2.81it/s, loss=3.4, accuracy=24.3]
:51,  2.78it/s, loss=3.4, accuracy=24.3]
ubuntu@ip-10-255-2-228:~$
```

TTK Demo on HPC

References

- Singularity on Compute Canada HPC:
  https://docs.computecanada.ca/wiki/Singularity

- Sylabs.io Singularity Official User Guide:
  https://sylabs.io/guides/3.7/user-guide/

- Westgrid YouTube channel:
  https://www.youtube.com/channel/UCfgds4Qf7VFOv4ORRvFFmhw

Attributions

- https://en.wikipedia.org/wiki/Intermodal_container#/media/File:Line3174_-_Shipping_Containers_at_the_terminal_at_Port_Elizabeth,_New_Jersey_-_NOAA.jpg

- https://commons.wikimedia.org/wiki/File:VMM-Type1.JPG

- https://en.wikipedia.org/wiki/Docker_(software)#/media/File:Docker-linux-interfaces.svg

- http://singularity.lbl.gov/assets/img/diagram/singularity-2.4-flow.png

UBC

WESTGRID