Singularity on Hoffman2: Using containers on HPC resources

Charles Peterson
Office of Advance Research Computing
UCLA
April 29, 2021

Singularity

- I have uploaded the slides and examples from this talk on our gitlab page
  - https://gitlab.idre.ucla.edu/cpeterson/singularity_ws

- Feel free to clone this repo to follow along

```
git clone https://gitlab.idre.ucla.edu/cpeterson/singularity_ws.git
```

What is Singularity?

- Singularity is a Free and Open-Source software that can run **Operating System (OS) Virtualization** also know as **containerization**.

- First created in 2015 by researchers at Lawrence Berkeley National Lab

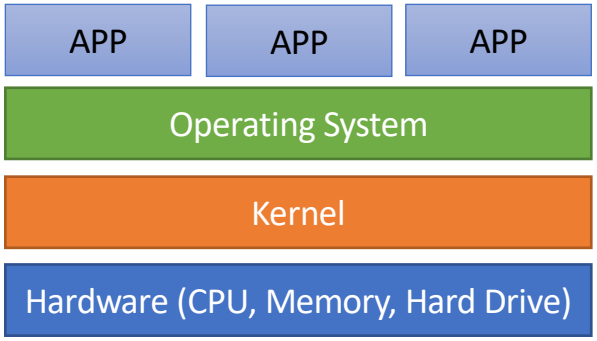
- Singularity was developed to run "**containers**" with scientific computing software on High-Performance Computing resources.

What is Virtualization?

Bare-metal setup

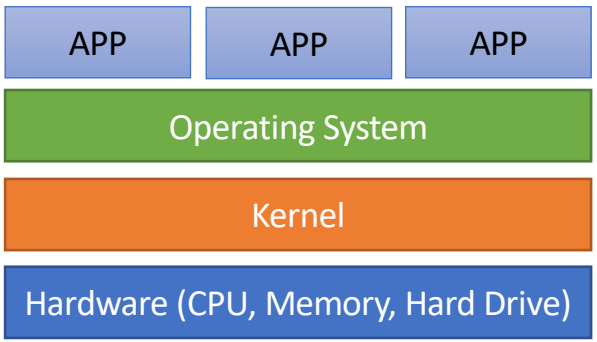
- Typical setup in which your software applications run directly on the OS from the physical hardware

- Many users in HPC center run compute nodes in this fashion

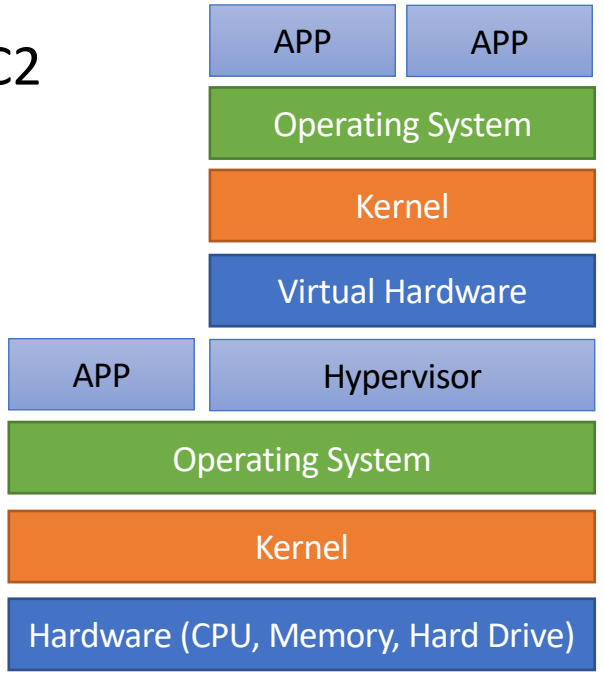


What is Virtualization? Virtual Machine setup

- Apps running inside VM are running on completely different (virtual) resources
- Examples: VirtualBox, VMWare, AWS EC2
- A ”Machine” within a ”Machine”



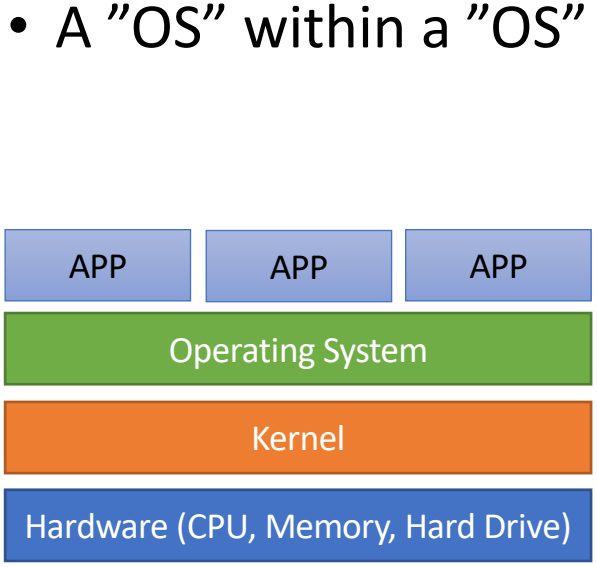
Bare-Metal



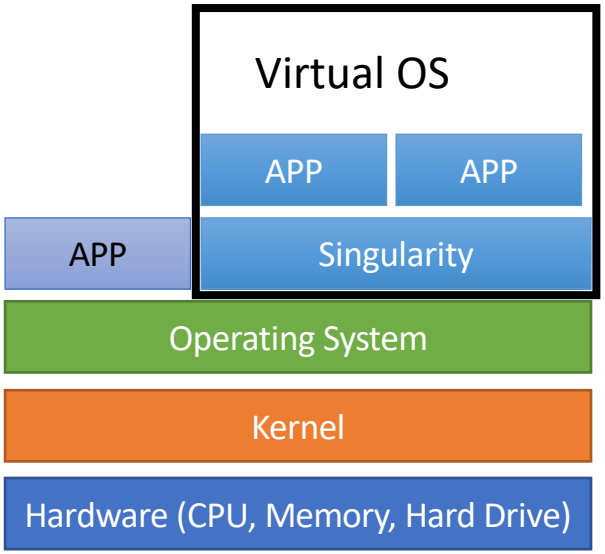
Virtual Machine (type 2)

What is Virtualization? Container setup

- Apps running inside a container are running using same kernel and physical resources as the host OS
- A ”OS” within a ”OS”



Bare-Metal



Containers

Containers

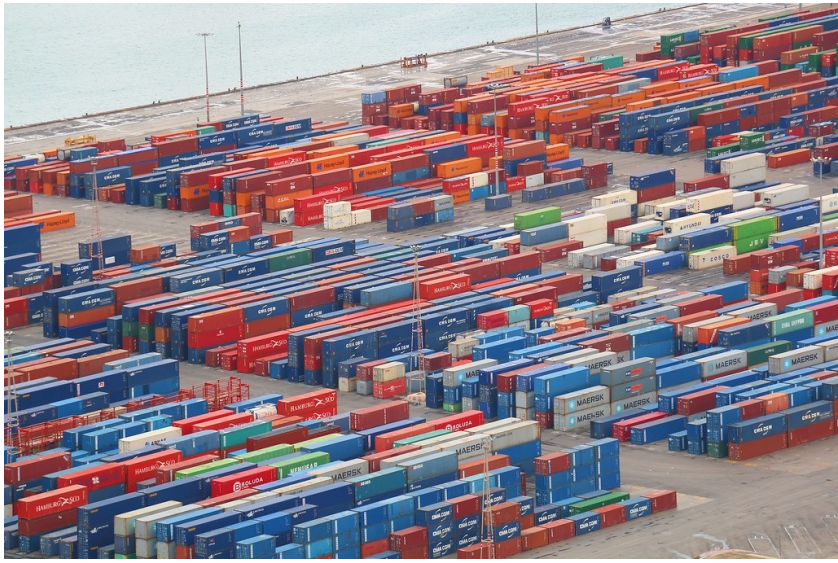
- Bring your own OS
- Portable
- Reproducibility
- Design your own environment

Containers on HPC: Problems installing apps

- HPC resources (like Hoffman2) are **SHARED** resources
  - Every researcher has their own apps that can cause conflicting dependences and versions
- Typical users cannot change system directories (i.e. /usr/local) or install apps in these default locations
  - No ‘sudo’ or admin access for users
  - Limited yum/apt-get
- Sys Admin can spend a lot of time installing apps/libraries in custom locations and create many different environments so everyone can use the HPC resources
  - Setting up environments with many dependences in custom location can be difficult to manage

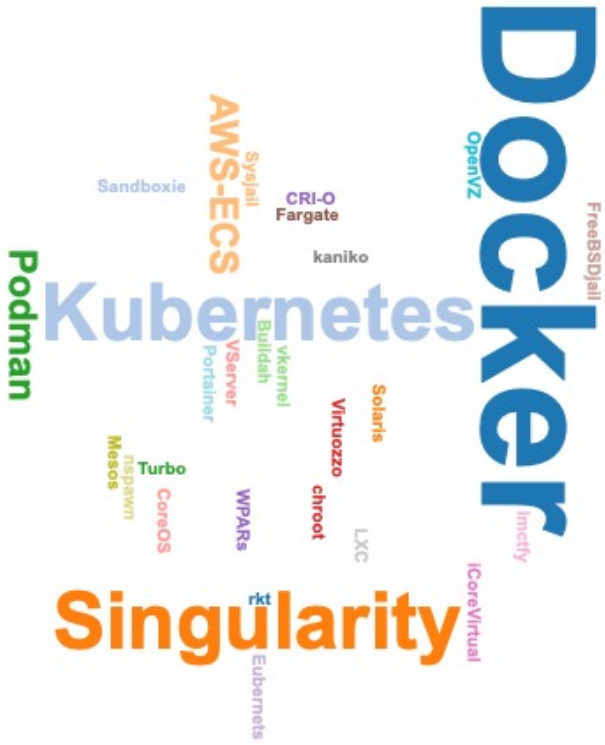
Containers on HPC

- With Containers, users can create a ‘virtual’ OS and install their app easily in system default PATH/LIB locations.
  - Along with any dependencies
- Then port them to Hoffman2 or any other HPC system to run without needing to ask a sys admin help
- The container is an isolated OS
  - Great you need different system libraries then the host OS on Hoffman2 compute nodes to run your apps
    - Too old or need specific version
    - OS specific requirements
  - Great if you need to run different packages that may conflict with each other if everything were to be installed in HOME
  - Easily share among people



Many Containerize software

- Docker
  - Maybe the most popular container engine
  - DockerHub: repository of containers
  - Will not be likely that Docker will be installed on HPC systems
  - Users can be root
  - MPI not well supported
- Singularity
  - Most common amongst HPC resources
  - Designed and developed by HPC people



Benefits of Singularity

- Increase availability at HPC centers
  - UCLA, SDCS, ORNL, MSU, OSU, TACC, PSC, etc.
- Singularity cannot have escalation outside container
- Singularity is a process owned by the user and has permissions similar to the user running the app
- Supports Infiniband, GPUs, different filesystems, MPI
- Can run Docker containers

https://sylabs.io/singularity/

Singularity Workflow

- Create
  - Build a container by installing Singularity on your local computer (where you have root/sudo access) and build your software app
  - Use a pre-built container (i.e. DockerHub)
- Transfer
  - Bring your container to Hoffman2
- Run
  - Perform Singularity commands to run your container either by Interactive (qrsh) or Batch (qsub) use

Singularity on Hoffman2

- Hoffman2 has an unprivileged, non-setuid build of Singularity (version 3.7)
  - Cannot perform privileged commands (i.e. sudo)
  - Only sandbox directories containers can run
  - SIF container cannot run (conversion possible)
  - --userns option will be needed (create new user namespace in your container)

- Accessible to the CentOS 7 compute nodes
  - Add –l rh7 to your qrsh/qsub job script

```
qrsh –l rh7,h_data=20G,h_rt=2:00:00
```

Singularity: Common Usage Running on Hoffman2

- On Hoffman2, first load the module

```
module load singularity/3.7
```

- Then run the command, Singularity, which has the following available subcommands
  - Build a Singularity image/container

```
singularity build [options]
```

  - Run a command within a container

```
singularity exec [options]
```

  - Run an interactive session within your container

```
singularity shell [options]
```

Running Singularity on Hoffman2 Example 1: TensorFlow

- Look at EX1 in the git repo
- Example of TensorFlow deep learning model with training MNIST dataset
- This example uses containers pre-built on Hoffman2

```
ls $H2_CONTAINER_LOC
cat $H2_CONTAINER_LOC/README
```

- Using Interactive and Batch jobs

Running Singularity on Hoffman2 Example 2: TensorFlow (v1)

- Look at EX2 in the git repo
- This example uses a Random Forest model with MINST dataset using TensorFlow v1
- This example downloads pre-built container from dockerhub

Creating your containers: Installing Singularity

- What if you want to create your own containers to run on Hoffman2
  - You cannot modify containers from Singularity on Hoffman2
- First, you will need a Linux computer that you have admin/sudo access
  - To transfer and run container across multiple systems, you would want to use ‘similar’ OS kernels
- Building containers with MacOS or WSL may not work on HPC systems using Linux
  - Best to use a Linux VM on local Mac/Windows systems.
  - Example: VirtualBox

Installing Singularity

- Installing dependencies

```
sudo apt-get update && sudo apt-get install -y \
        build-essential \
        uuid-dev \
        libgpgme-dev \
        squashfs-tools \
        libseccomp-dev \
        wget \
        pkg-config \
        git \
        cryptsetup-bin
```

- Installing Go

```
export VERSION=1.14.12 OS=linux ARCH=amd64 && \
        wget https://dl.google.com/go/go$VERSION.$OS-$ARCH.tar.gz && \
        sudo tar -C /usr/local -xzvf go$VERSION.$OS-$ARCH.tar.gz && \
        rm go$VERSION.$OS-$ARCH.tar.gz
export PATH=/usr/local/go/bin:$PATH
```


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

Installing Singularity

- Downloading Singularity

```
export VERSION=3.7.0 && # adjust this as necessary \
    wget
https://github.com/hpcng/singularity/releases/download
/v${VERSION}/singularity-${VERSION}.tar.gz && \
    tar -xzf singularity-${VERSION}.tar.gz && \
    cd singularity
```

- Installing Singularity

```
./mconfig && \
    make -C ./builddir && \
    sudo make -C ./builddir install
```

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

Building a container: Singularity Definition file

- Recipe file to build a container

- naked-singularity
  - repository of many definition files
  - https://github.com/mkandes/naked-singularity

```
sudo singularity build ubuntu-18.04.sif Singularity.ubuntu-18.04
```

- A Singularity container is created (ubuntu-18.04.sif)
- Create an interactive shell within container

```
sudo singularity shell ubuntu-18.04.sif
```

Building a container
Modifying an existing container

- You can download and create existing container
  - Perfect if you want to add more packages/libraries
  - Can ONLY do this on computer you have sudo/root access

- DockerHub https://hub.docker.com
  - great repo to search for containers

- Download sandbox docker image

```
sudo singularity build --sandbox ubuntu docker://ubuntu:18.04
```

- Go inside container and install apps!

```
sudo singularity shell --writable ubuntu
```

Building a container Example 3: PySCF

- Look at EX3 in the git repo

- PySCF is a chemistry python package
  - https://pyscf.org/

- First, install Singularity on your local machine
- Then download ubuntu container from dockerhub

```
sudo singularity build --sandbox ubuntu/ docker://ubuntu:20.04
```

- Then get inside container

```
sudo singularity shell --writable ubuntu/
```

- Install pyscf python package

```
apt update
apt install python3-dev python3-pip
pip3 install pyscf
```

Building a container Example 3: PySCF

- convert sandbox to SIF

```
sudo singularity build pyscf.sif ubuntu/
```

- Transfer SIF to Hoffman2

```
scp pyscf.sif dtn.hoffman2.idre.ucla.edu:
```

- Create job script (pyscf.job) and submit job

```
qsub pyscf.job
```

Using Singularity on GPU Example 4: PyTorch

- Singularity can be ran on GPU nodes
- No need to install drivers or libraries, or load anything (expect Singularity) in your env
- Check out EX4

```
-l gpu,exclusive,gpu,RTX2080Ti
```

```
qsub pytorch-gpu.job
```

```
singularity exec --nv --userns $H2_CONTAINER_LOC/pytorch-21.04-py3.sif /
    python3 pytorch-gpu.py > pytorch-gpu.out
```

- Look at the singularity exec line
  - added --nv option to enable Nvidia support
  - allows to bind mount the GPU drive

Things to consider
Size of container

- Try to keep size of container small and minimal
  - Only install the things necessary for the app to run

- Large containers will need more RAM to run (increasing h_data)

- Since Singularity on Hoffman2 can only run sandbox containers, Singularity will convert any SIF file before running.
  - Increasing size of container will increase the time of conversion.

  - For large containers, build a sandbox container in order to save time starting Singularity.

```
singularity build ubuntu-SB/ ubuntu.sif
```

Other places for Singularity information

- Singularity User Guide
  - https://sylabs.io/guides/3.7/user-guide/
- Intro workshop from SDSC
  - https://www.sdsc.edu/event_items/2019.02_cometweb.html
- Container Repos
  - DockerHub
    - https://hub.docker.com/
  - SingularityHub
    - https://singularityhub.github.io/
  - Singularity Container Services
    - https://cloud.sylabs.io/home
  - Nvidia NGC
    - https://ngc.nvidia.com/catalog/containers

Thank you!

Questions?

Charles Peterson
cpeterson@oarc.ucla.edu

GitLab rep of this workshop
https://gitlab.idre.ucla.edu/cpeterson/singularity_ws

Hoffman2 Online Doc on Singularity
https://www.hoffman2.idre.ucla.edu/Using-H2/Software/Software.html#containers

Hoffman2 Online Support
https://support.idre.ucla.edu/helpdesk/Tickets/New