Singularity Workshop 2019

A field guide to contained academic computing

Mario Belledonne

November 13, 2019

Yale Psychology

This is a work in progress!

This presentation and examples can be found here:

`https://github.com/CNCLgithub/singularity_workshop_2019`

Table of contents

1. Containers
2. Singularity: Getting Started
3. Singularity: Using HPCs

Lecture Goals

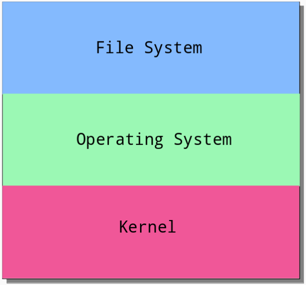
- What are containers?
- How can they help?
- How can we use them?

Containers

Computer

What is a computer (abstract)

- File System (FS)
- Operating System (OS)
- Kernel



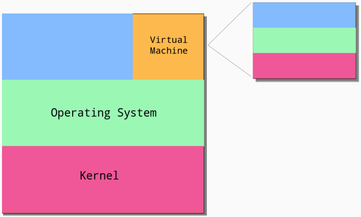
File System

Operating System

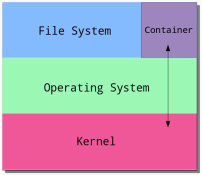
Kernel

Extensions

There are several ways to augment a computer

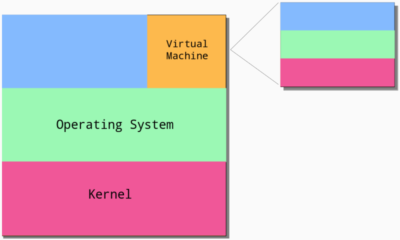


(a) Virtual Machine



(b) Container

Virtual Machines



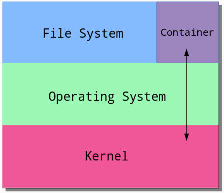
Pros:

- Relatively universal (assuming OS and hardware support)
- Extensive configuration

Cons:

- Resource intensive
- Security concerns

Containers



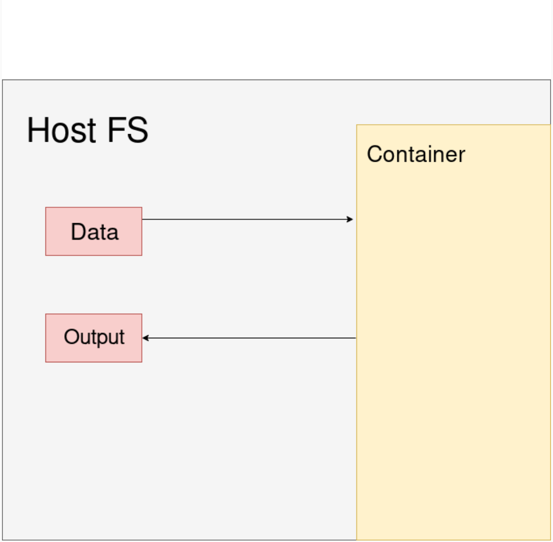
Pros:

- Low overhead
- Flexible deployment

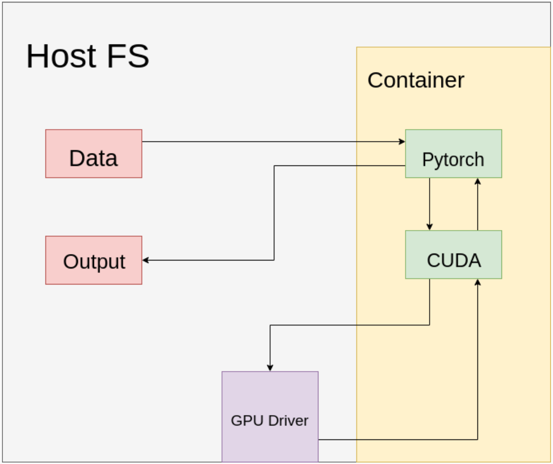
Cons:

- Kernel limitations

Container interactions with host



Container interactions with host



Singularity: Getting Started

Examples to follow along

This presentation can be found here:
`https://github.com/CNCLgithub/singularity_workshop_2019`

Pulling containers

Containers can be pulled from container repos

```
# pulls from sylabs repos
$ singularity pull alpine.sif library://alpine:latest
# pulls from docker repos
$ singularity pull tensorflow.sif \
docker://tensorflow/tensorflow:latest
```

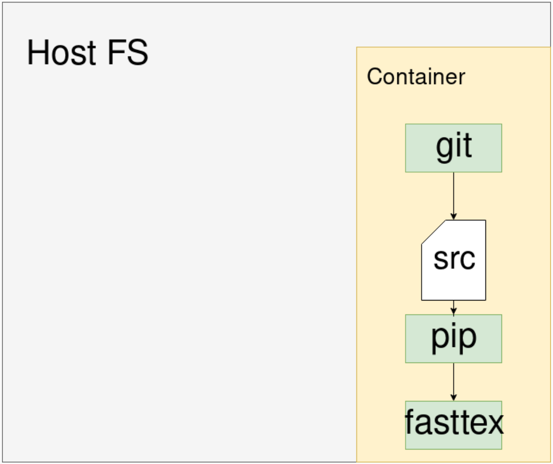
These are immutable!

Building containers: Definition File

https://github.com/belledon/docker-singularity/blob/master/Singularity

```
bootstrap: docker
from: python:3.7.4-alpine3.10

%post

  apk add --update \
      build-base \
      git

  cd /
  git clone https://github.com/facebookresearch/fastText.git
  cd fastText
  pip install .
  cd / && rm -r fastText

%runscript

  python $@
```

Whats going on?



Definition File components

- `%post` - Run installation commands **after** setting up the parent
- `%run` - Define the default executable behavior
- `%environment` - Any env variables you would like defined

Definition File components - extra

- `%setup` - Any steps to run before running anything else
- `%files` - Any files to copy to the host
- `%help` - Describe the purpose of the container

For more details please see the docs: `https://sylabs.io/guides/3.4/user-guide/definition_files.html`

Don’t reinvent the wheel!

You can often start from somewhere close!

`examples/Singularity.julia`

```
bootstrap:docker
from:julia:1.3.0-stretch

%environment
  export JULIA_DEPOT_PATH="${PWD}/.julia"
  export JULIA_PROJECT="$PWD"
```

But don’t be afraid

There will be times where you need to ”take a step back”

`examples/Singularity.pytorch-docker`

```
bootstrap: docker
from: pytorch/pytorch:1.3-cuda10.1-cudnn7-devel

%post
  . /opt/conda/etc/profile.d/conda.sh
  conda activate base
  pip install pandas
```

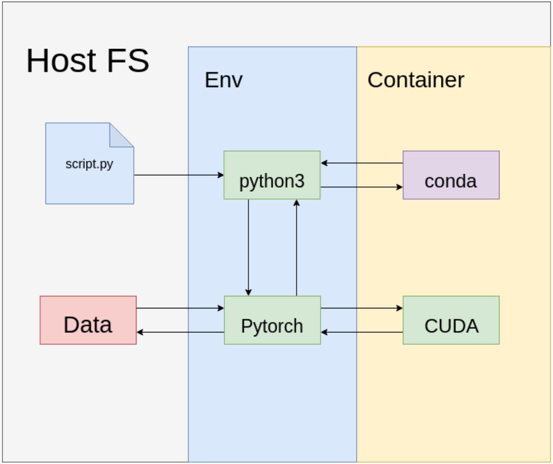
Leaving the nest

For development, immutable containers aren't ideal

`examples/Singularity.conda`

```
bootstrap: docker
from: nvidia/cuda:10.1-cudnn7-devel-ubuntu16.04

%environment
  # export path to conda executable
  export PATH=$PATH:/miniconda/bin

%post
   apt-get update
   apt-get install -y  build-essential \
           wget

   # download conda installer
   wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh -O conda.sh
   # run conda installer
   bash conda.sh -b -p /miniconda
   # cleanup installer script
   rm conda.sh
```

Leaving the nest: Package managers within containers



Singularity: Using HPCs

One module to rule them all

- A built container does not require sudo
- The container is one image file
- Ensures reproducibility

[3]

Getting your container on the HPC

The simplest way to get started is to build a container on your local machine.

```
$ scp $CONT \
> $USERNAME@<cluster>.hpc.yale.edu:/some/path
```

Alternatively, you could host your container online.

```
$ wget "http://www.my_cool_container/..."
```

Building on HPC

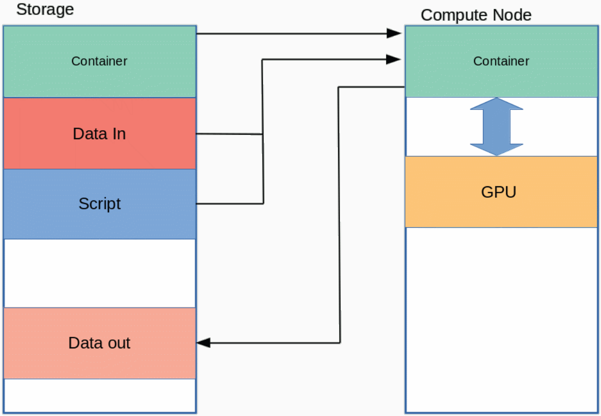
Not entirely reliable due to privileges.

Typically done via vagrant and virtualbox

Running a container in a compute node

```
$ cat my_sbatch.sh
#!/bin/bash
#SBATCH array=0-4
#
# note on Grace/milgram, singularity is available by
# default on compute nodes
singularity exec ./train_nature_paper.sh
```

Running on a compute node



Problem: Modifying a container

Not possible on OM due to privileges.

In most cases this is not necessary (keeping source and data outside of the container).

However, there is an exception during development

Extra slides on HPCs

Summary

Get the source of this theme and the demo presentation from

`github.com/matze/mtheme`

The theme *itself* is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

🅭🅯🄎

Collaborative Documentation

Useful linux commands

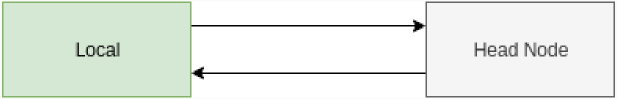
- man
- ssh
- scp
- module
- awk

High Performance Clusters

Interacting with an HPC

connecting to the head node

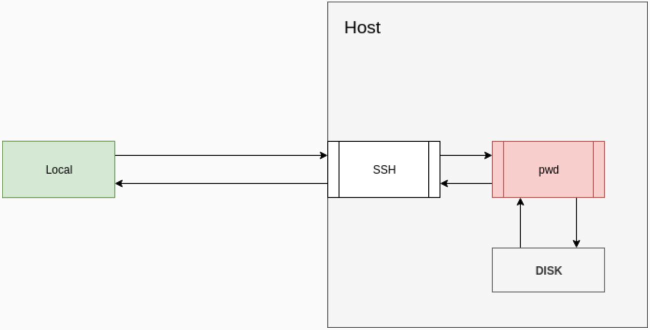
```
ssh "${USERNAME}@openmind7.mit.edu"
```



HelloWorld!

Running our first command on the headnode.

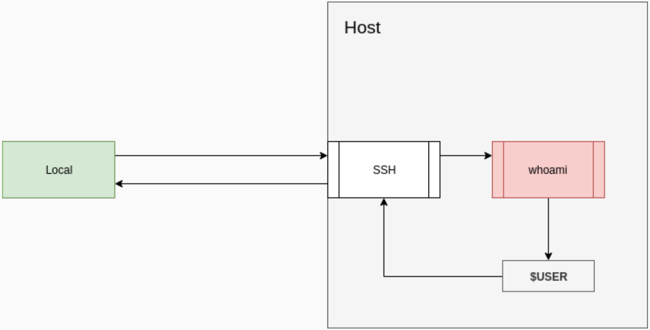
```
$ whoami
belledon
```



What is where?

Now that we know who we are, where are we?

```
$ pwd
/home/$USER
```



Home is where the config is

$HOME should only be used for private configs, ssh-keys...

```
$ ls -alh ~/
total 216K
drwxr-xr-x   6 belledon tenenbaum   81 Jun 20 11:31 .cache
drwxr-xr-x  20 belledon tenenbaum 4.0K Oct 23  2018 .config
drwx------   2 belledon tenenbaum 4.0K Oct 12  2018 .ssh
-rw-r--r--   1 belledon tenenbaum  125 May 25  2017 rsync.sh
```

HPC Storage: Lustre

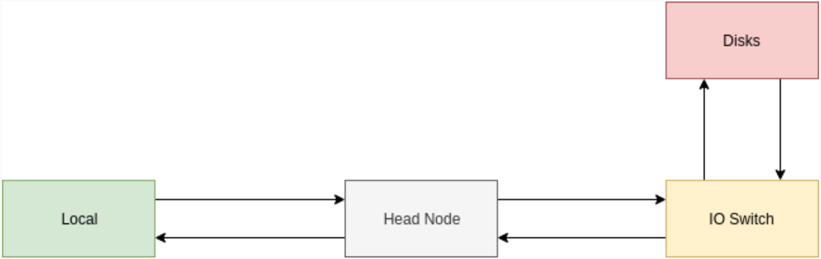
- Lustre file system supports fast, scalable IO [2]
- Sensitive to large number of files
- Resilient to the size of files
- Accessed via ”/om”

HPC Storage: NFS

- Supports basic network drives [1]
- Sensitive to IO
- Resilient to the number of files
- Accessed via "/om2"

Accessing remote disks

```
ls -l /om
```



Getting in Trouble

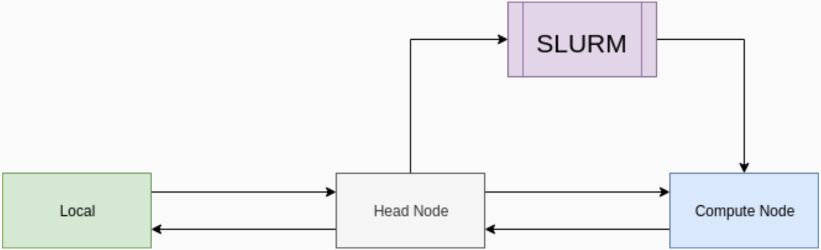
Finally... now lets get started

```
./train_nature_paper.sh
```

Accessing resources

Requesting an interactive job

```
srun -c 4 --mem=8G --qos="$GROUP" -t 1-0 --pty bash
```

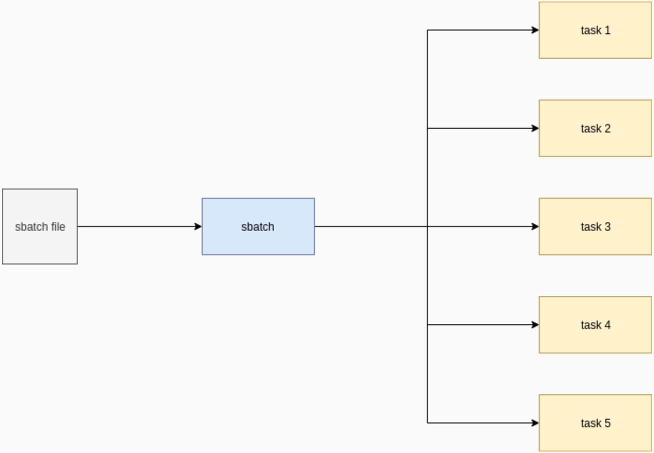


Accessing more resources

Requesting a batch job

```
$ cat example.sh
#!/bin/bash
#SBATCH --array=1-5
echo "The array id is: ${SLURM_ARRAY_TASK_ID}"

$ sbatch example.sh
Submitted batch job 13873280

$ find . -name "*.out" | \
  while read src; do echo "${src} -> $(cat ${src})"; done
./slurm-13873280_5.out -> The array id is: 5
./slurm-13873280_2.out -> The array id is: 2
./slurm-13873280_3.out -> The array id is: 3
./slurm-13873280_1.out -> The array id is: 1
./slurm-13873280_4.out -> The array id is: 4
```

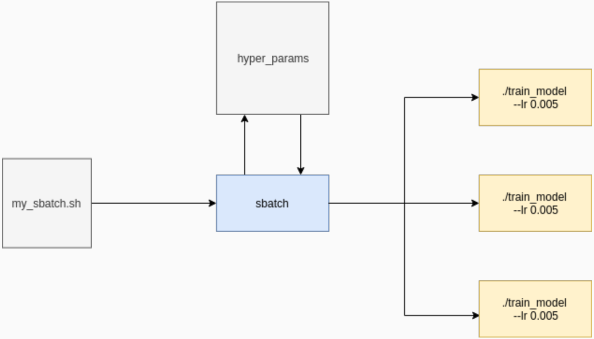
Exploring SBATCH Tasks



SBATCH Example: Hyperparameter search

```
$ cat network_params.txt
--lr 0.0005
--lr 0.005
--lr 0.00005

$ cat my_sbatch.sh
#!/bin/bash
#SBATCH --time=1-0
#SBATCH --array=1-5
#SBATCH --gres=gpu:1
ARGUMENTS=$(sed "${SLURM_ARRAY_TASK_ID}q;d" network_params.txt)
./train_model "$ARGUMENTS"
```

SBATCH Example contd



Adding project dependencies

On many HPCs, managing software dependencies is done via ”module”.

`module avail`

Shows modules you can add

`module add`

Adds a module to your environment

`module list`

Shows which modules you have

`module purge`

Removes modules from environment

Adding a package

This snippet adds "singularity" to my env

```
    -bash-4.2$ module add openmind/singularity/3.2.0
    -bash-4.2$ singularity --version
    singularity version 3.2.0-1
```

Problem: Computing environment

- Required module doesn’t exist?
- Can’t install dependency?
- Project environment behaves erratically?

HPCs, they're simple as 123

HPCs are to PCs as modern industrial economies are to aggrarian societies.

Division of labour:

- Compute
- IO
- Disk

Regulatory Trade Agencies

- Job Scheduling
- Service Monitoring

Chaos

- You (users)
- The man (admins)
- God (hardware failures)

References i

FreeBSD.
29.3. network file system (nfs).

EOFS OpenSFS.
About the lustre file system, 2019.

SyLabs.
Definition files.