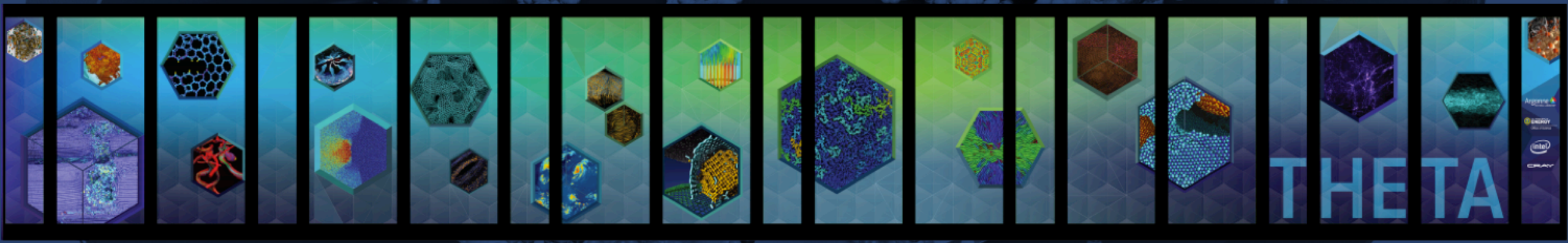
Argonne NATIONAL LABORATORY

Using Containers on Theta

Murat Keçeli

SIMULATION.DATA.LEARNING WORKSHOP



Thanks to J. Taylor Childers for the slides

Container Survey

Please vote based on your experience with:

- Virtual machines, hypervisors (VMware (1998), Virtualbox (2007))
- Docker (2013)
- Shifter (2015)
- Singularity (2016)

https://doodle.com/poll/2a723x2u9esbxyhh

or

https://tinyurl.com/thetasurvey2

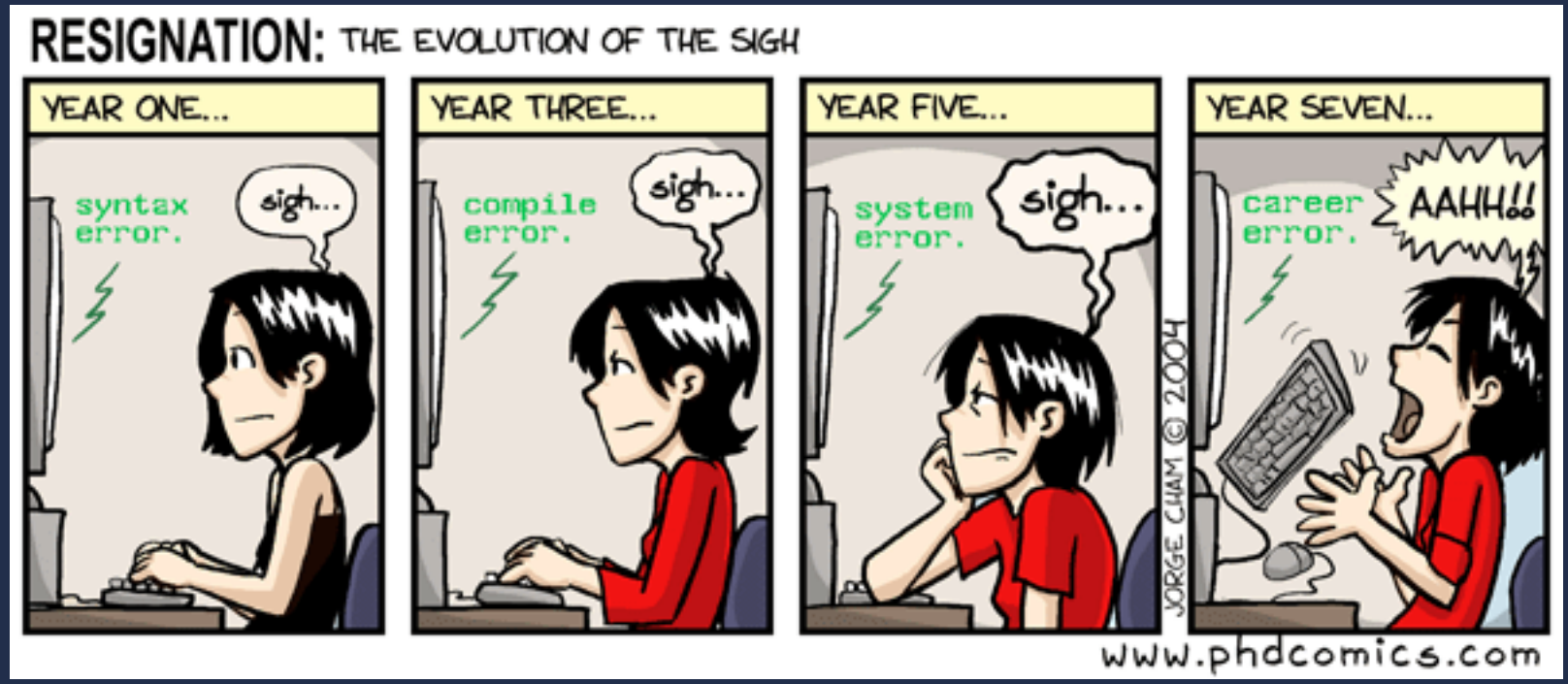
Virtualization, Conteinerization

The aim is to run an app in an isolated environment

- Virtual machines provide full virtualization
  - Requires an image file
  - Guest OS run on the host OS on a virtual hardware layer
- Containerization is OS level virtualization
  - Requires an image or a recipe file
  - Each isolated user-space instance is called a container

Argonne NATIONAL LABORATORY

Do we need containers?



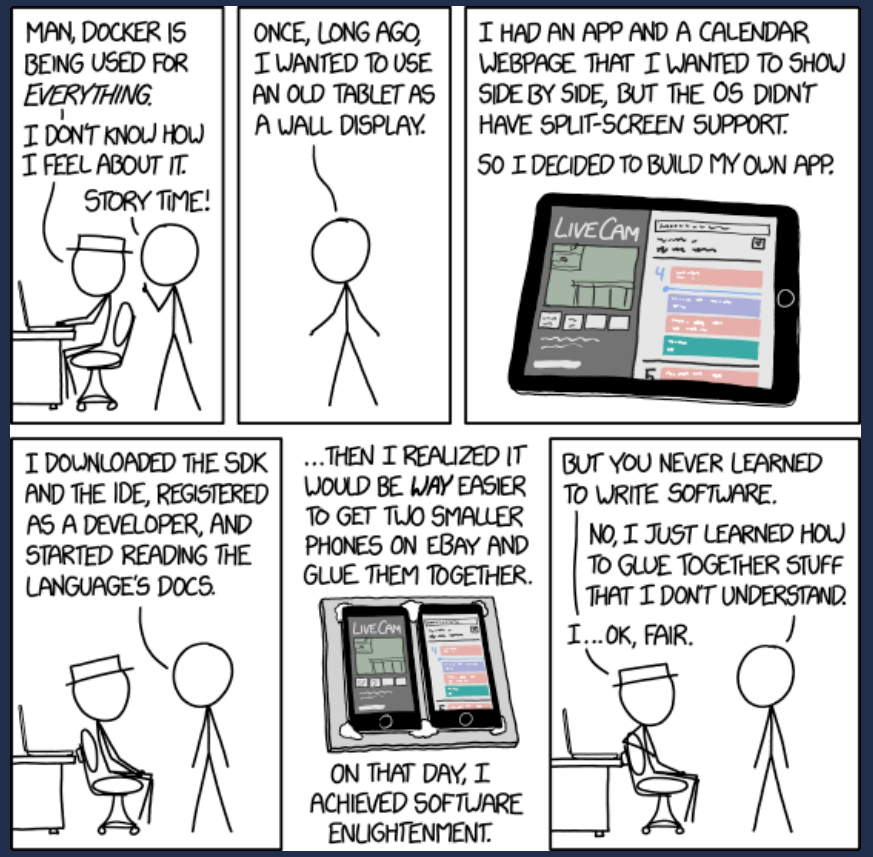
It can make your life easier.

http://phdcomics.com/comics/archive.php?comicid=531

Advantages

- Portability
  - You can use the same image on your laptop, local cluster or a *supercomputer*.
  - No need to ask system admins to install a library for you.
- Reproducibility
  - $28 billion per year is spent on preclinical research that is not reproducible.
  - Include all data required to run your application in the image.
- Faster development and production
  - You can build the image anywhere, no need to compile on login node.
  - You can create an image based on existing images.

A Silver Bullet?



May not be the best solution.

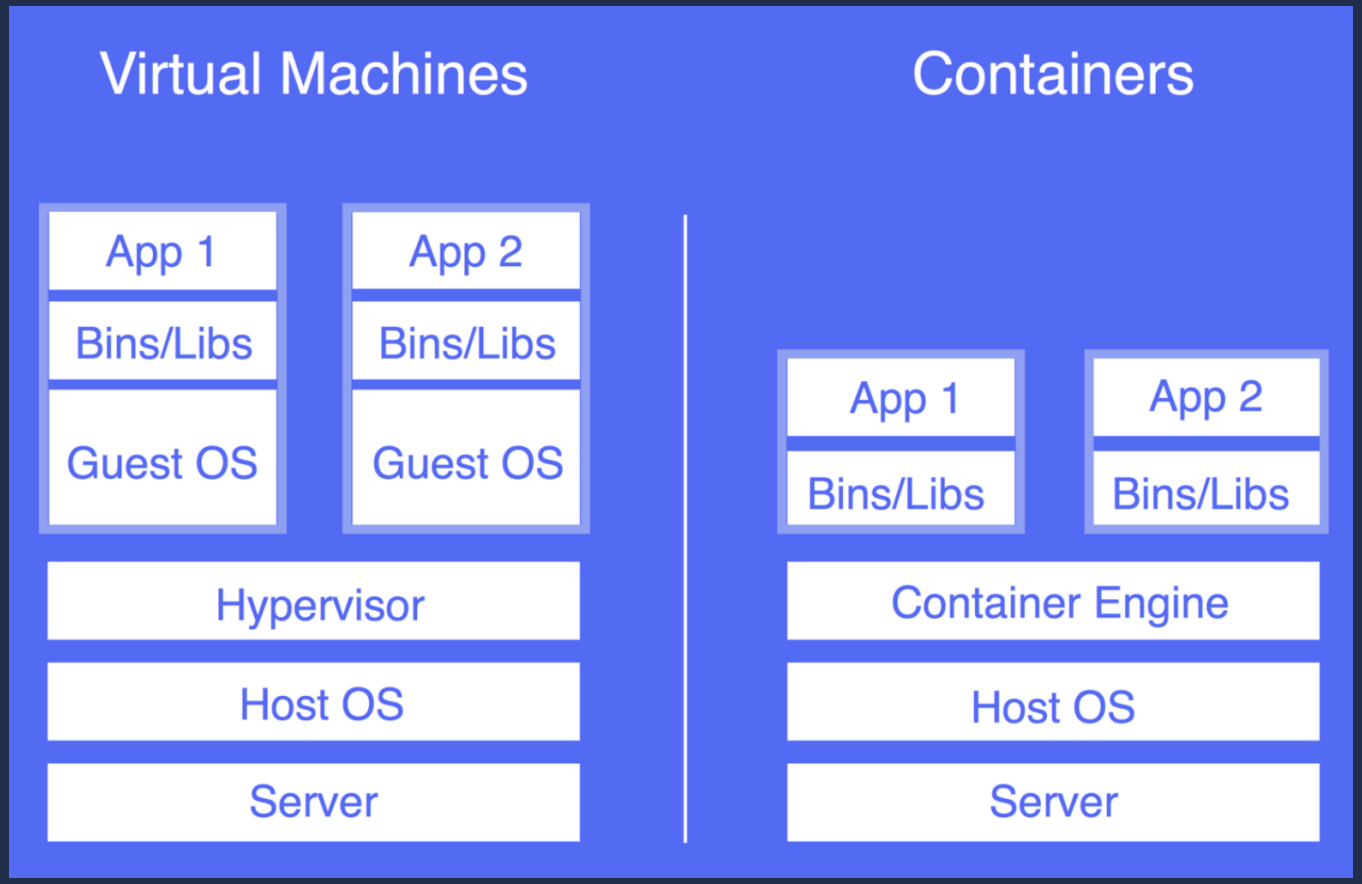
https://xkcd.com/1988/

A Silver Bullet?

“The average time required to implement a moderate sized application is equivalent to half-life of the parallel computing platform”, John Reynders, 1996.

Argonne NATIONAL LABORATORY

Virtual Machines vs Containers



Argonne NATIONAL LABORATORY

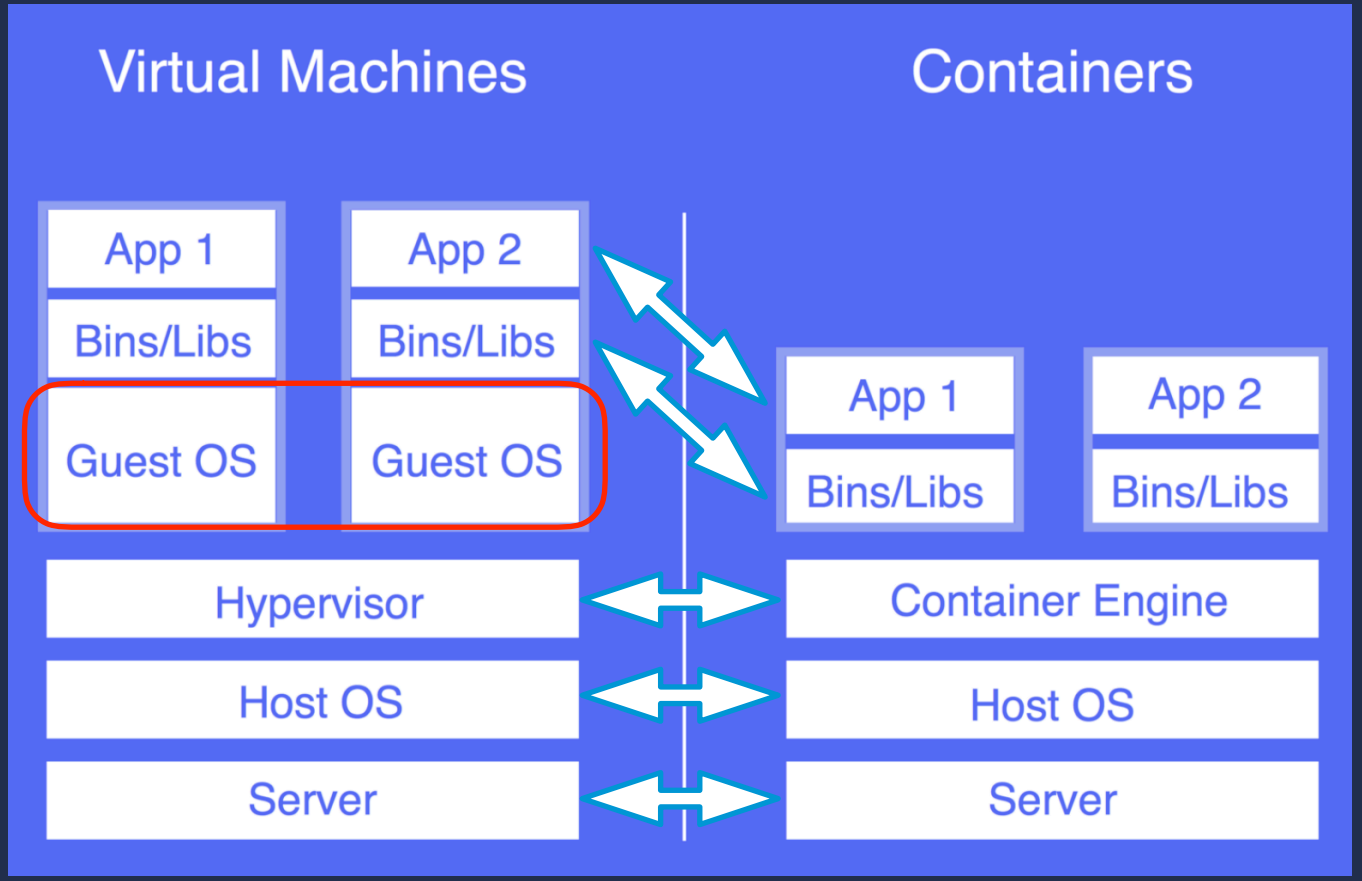
Virtual Machines vs Containers

Both Require:

- Host Operating System
- Hypervisor or Engine
- Image

Main Difference:

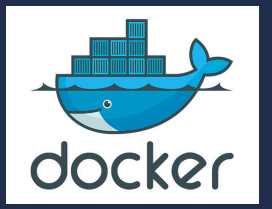
- VMs require entire internal operating system
- VMs virtualize system hardware



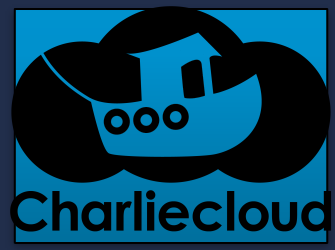
Argonne NATIONAL LABORATORY

Container Solutions

- Linux Containers (LXC)
  - Uses kernel namespaces and cgroups for resource management.
- Docker
  - Extension of LXC, current enterprise solution for micro-services.



- HPC containers:
  - Shifter (NERSC)
  - Charlie Cloud (LANL)
  - Singularity (LBNL, Sylabs Inc.)



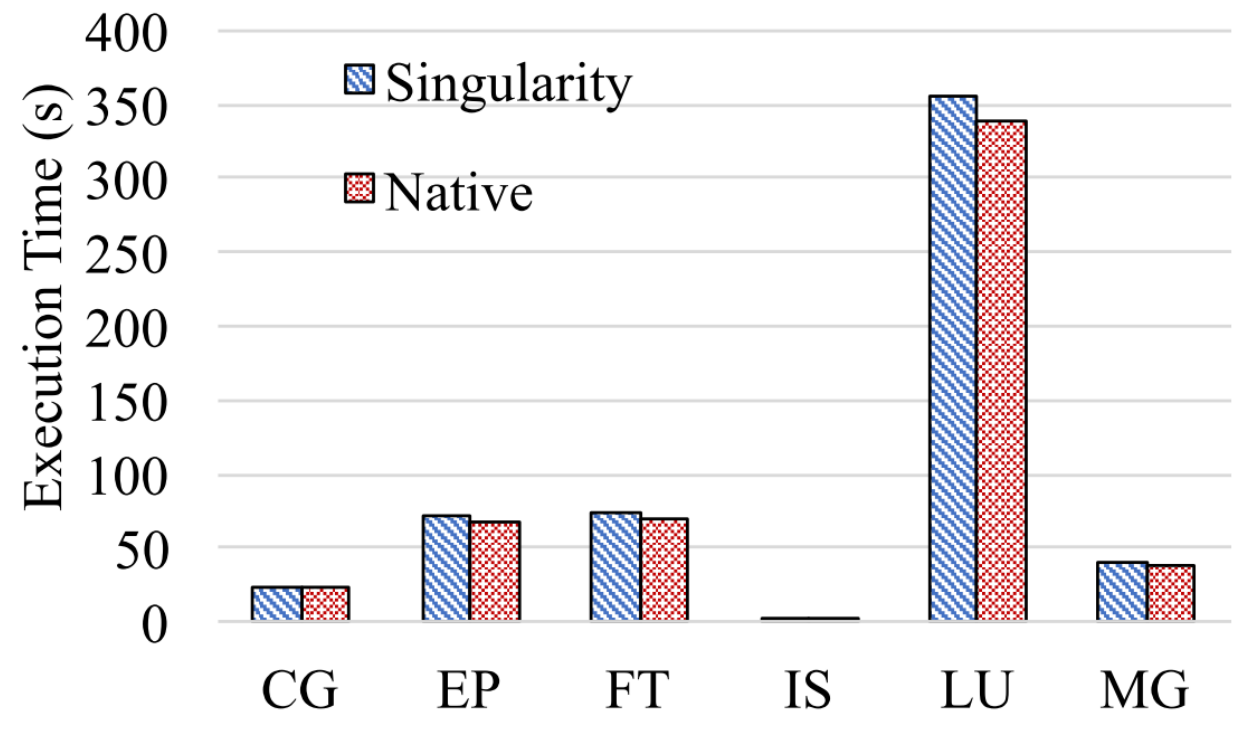
Container Comparison

	Singularity	Shifter	Charlie Cloud	Docker
Privilege model	SUID/UserNS	SUID	UserNS	Root Daemon
Supports current production Linux distros	Yes	Yes	No	No
Internal image build/bootstrap	Yes	No*	No*	No***
No privileged or trusted daemons	Yes	Yes	Yes	No
No additional network configurations	Yes	Yes	Yes	No
No additional hardware	Yes	Maybe	Yes	Maybe
Access to host filesystem	Yes	Yes	Yes	Yes**
Native support for GPU	Yes	No	No	No
Native support for InfiniBand	Yes	Yes	Yes	Yes
Native support for MPI	Yes	Yes	Yes	Yes
Works with all schedulers	Yes	No	Yes	No
Designed for general scientific use cases	Yes	Yes	No	No
Contained environment has correct perms	Yes	Yes	No	Yes
Containers are portable, unmodified by use	Yes	No	No	No
Trivial HPC install (one package, zero conf)	Yes	No	Yes	Yes
Admins can control and limit capabilities	Yes	Yes	No	No

G. M. Kurtzer, V. Sochat, and M. W. Bauer, “Singularity: Scientific containers for mobility of compute,” PLoS One, vol. 12, no. 5, pp. 1–20, 2017.

Argonne NATIONAL LABORATORY

Performance / Overhead

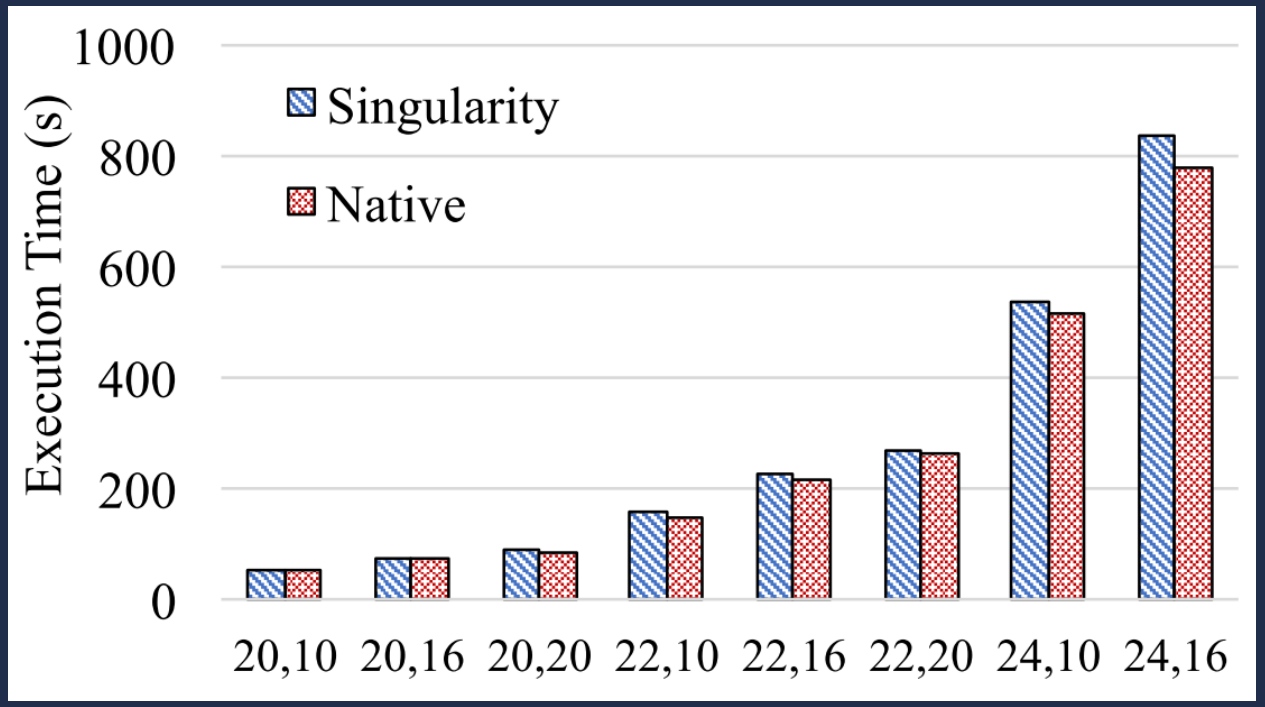


CG	Conjugate Gradient
EP	Embarrassingly Parallel
FT	3D FFT, all-to-all communication
IS	Integer Sort, random memory access
LU	Lower-Upper Gauss-Seidel solver
MG	Multi-Grid, memory intensive

X. Lu and D. K. Panda, “Is Singularity-based Container Technology Ready for Running MPI Applications on HPC Clouds?,” Proc. 10th Int. Conf. Util. Cloud Comput. (UCC ’17), pp. 151–160, 2017.

Argonne NATIONAL LABORATORY

Performance / Overhead



Graph-data analytics workload

Point-to-point communications

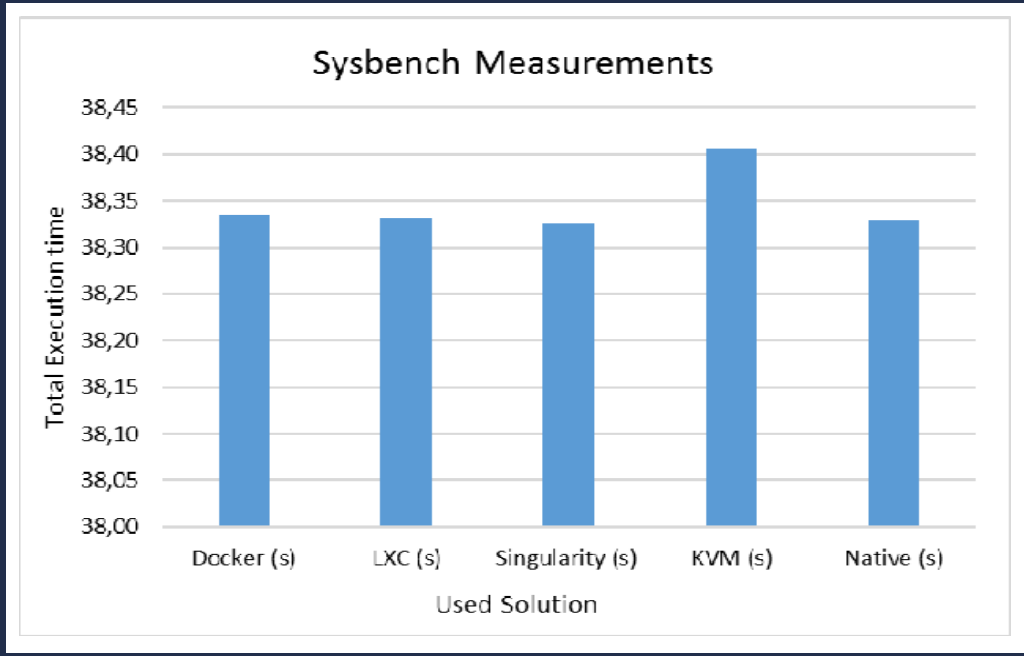
20,10 means 2^{20} vertices and 2^{10} edges

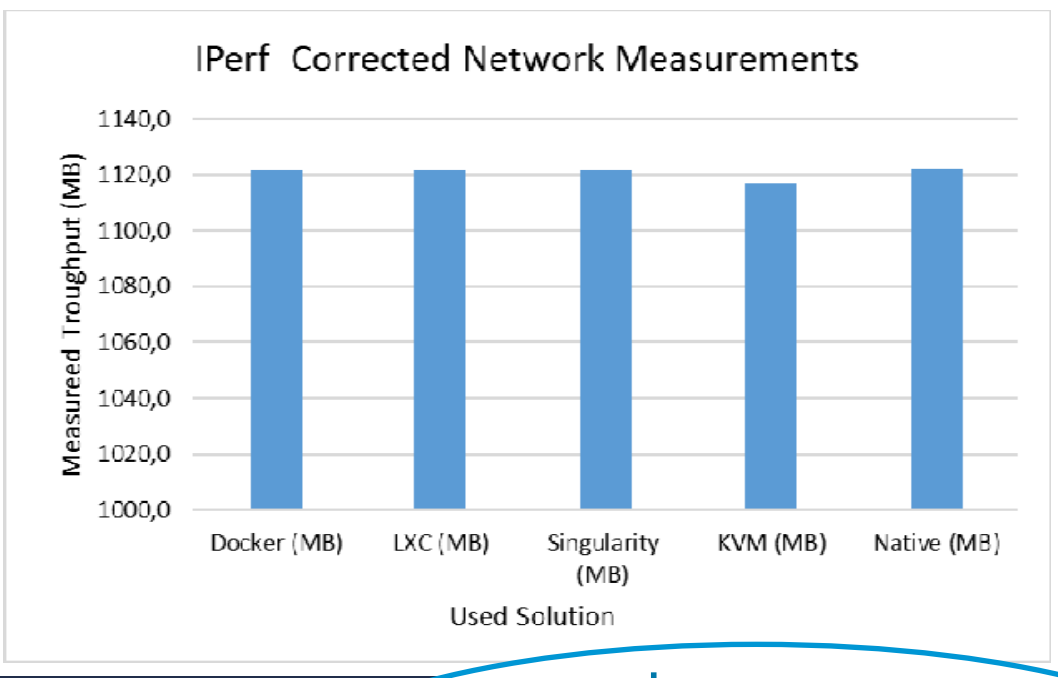
2 KNL nodes, 128 processes

Less than 8% overhead for CPU, memory, network, and IO

X. Lu and D. K. Panda, "Is Singularity-based Container Technology Ready for Running MPI Applications on HPC Clouds?," Proc. 10th Int. Conf. Util. Cloud Comput. (UCC '17), pp. 151–160, 2017.

Performance / Overhead

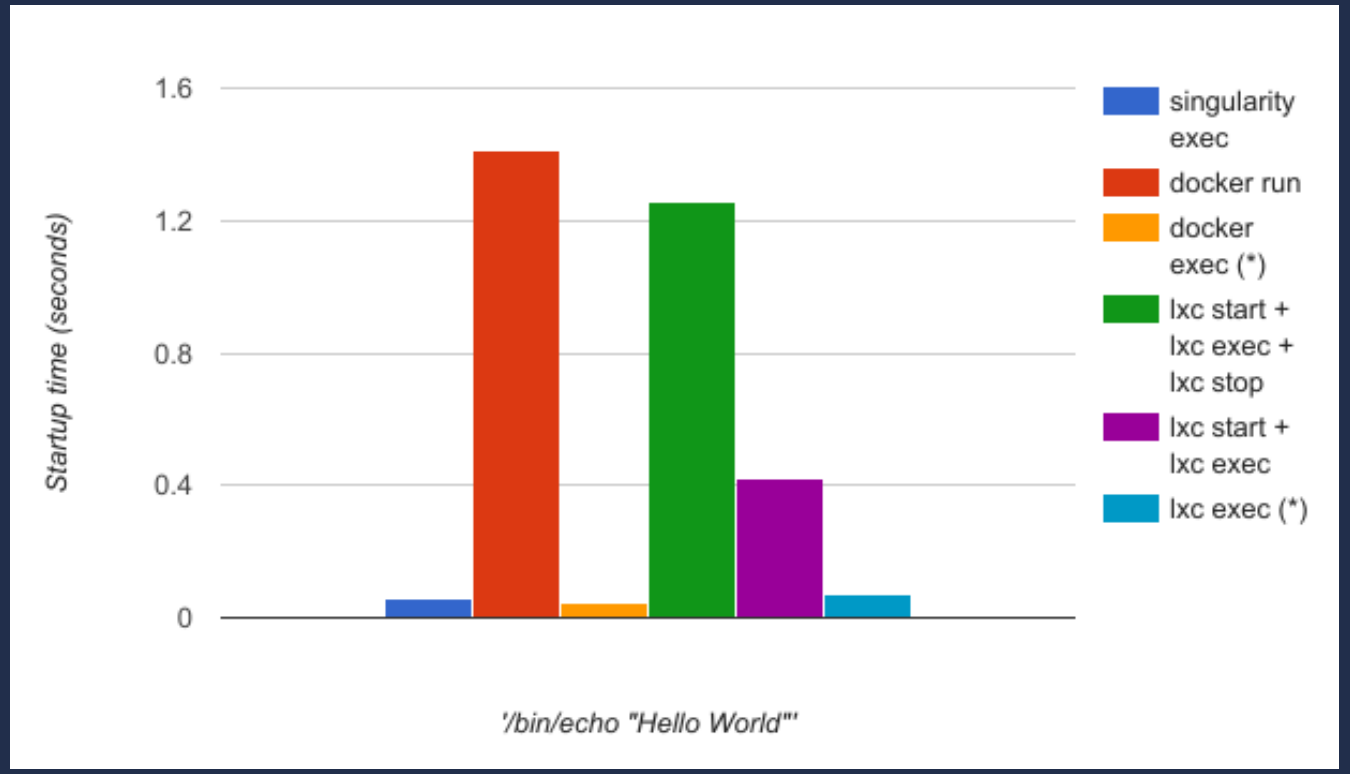




Less than 1% overhead for CPU, and network

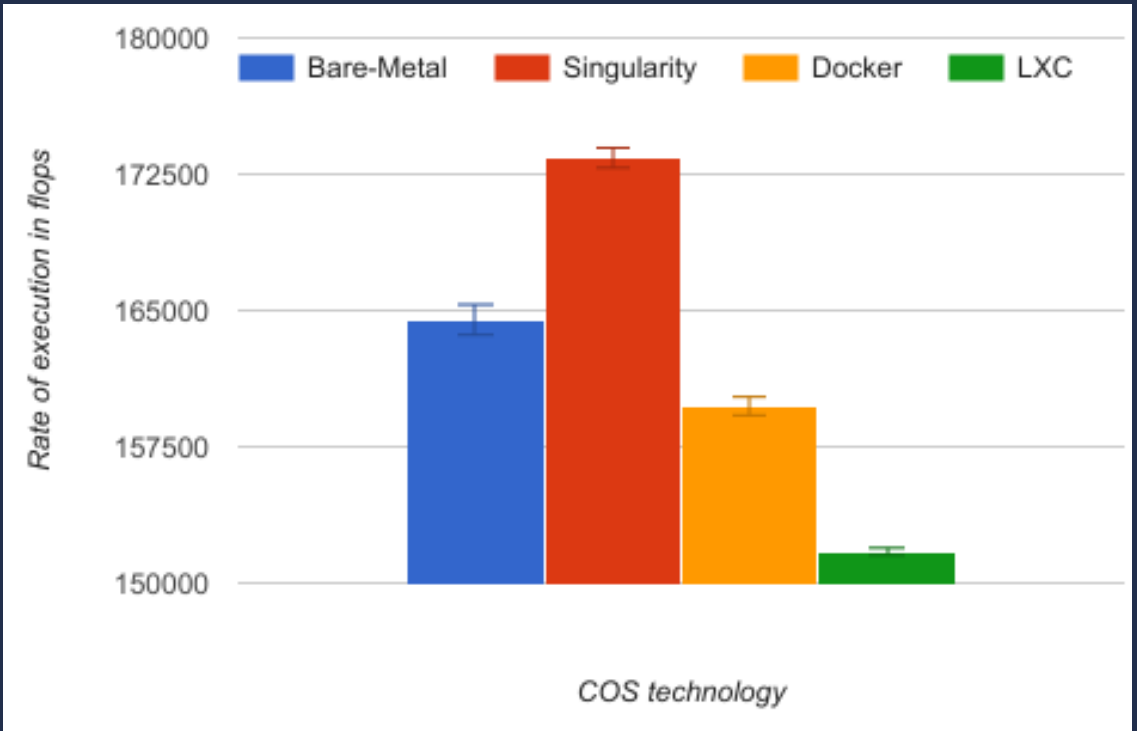
Á. Kovács, "Comparison of different linux containers," 20 Telecommun. Signal Process. TSP 2017, vol. 2017–January, pp. 47–51, 2017.

Performance / Overhead



C. Arango, R. Dernat, and J. Sanabria, “Performance Evaluation of Container-based Virtualization for High Performance Computing Environments,” 2017.

Performance / Overhead



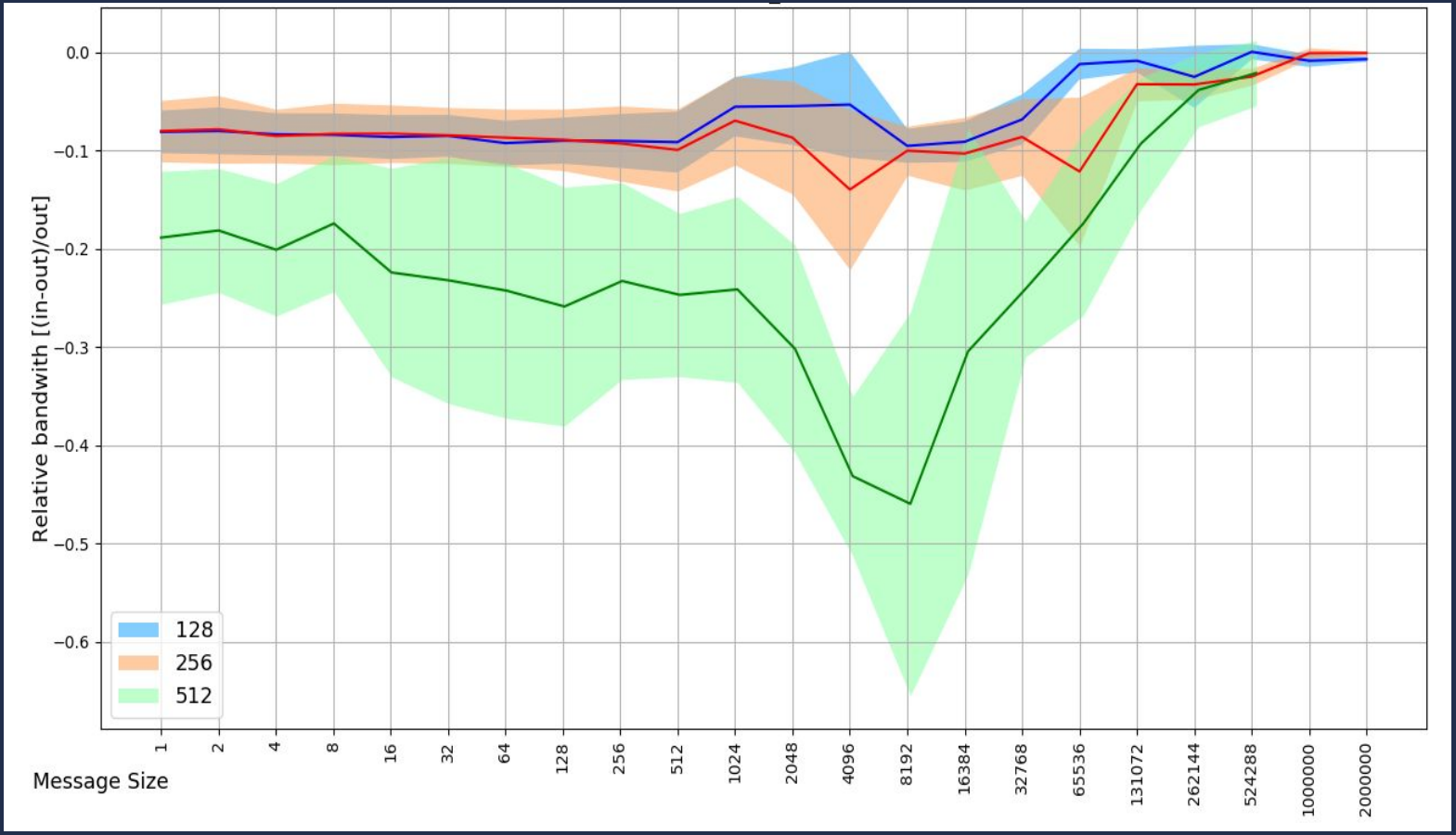
Singularity was able to achieve a **better** performance than **native** with 5.42% because is not emulating a full hardware level virtualization (only the mount namespace) paradigm and as the image itself is only a single metadata lookup this can yield in very high performance benefits.

HPL benchmark, higher is better

C. Arango, R. Dernat, and J. Sanabria, "Performance Evaluation of Container-based Virtualization for High Performance Computing Environments," 2017.

Argonne NATIONAL LABORATORY

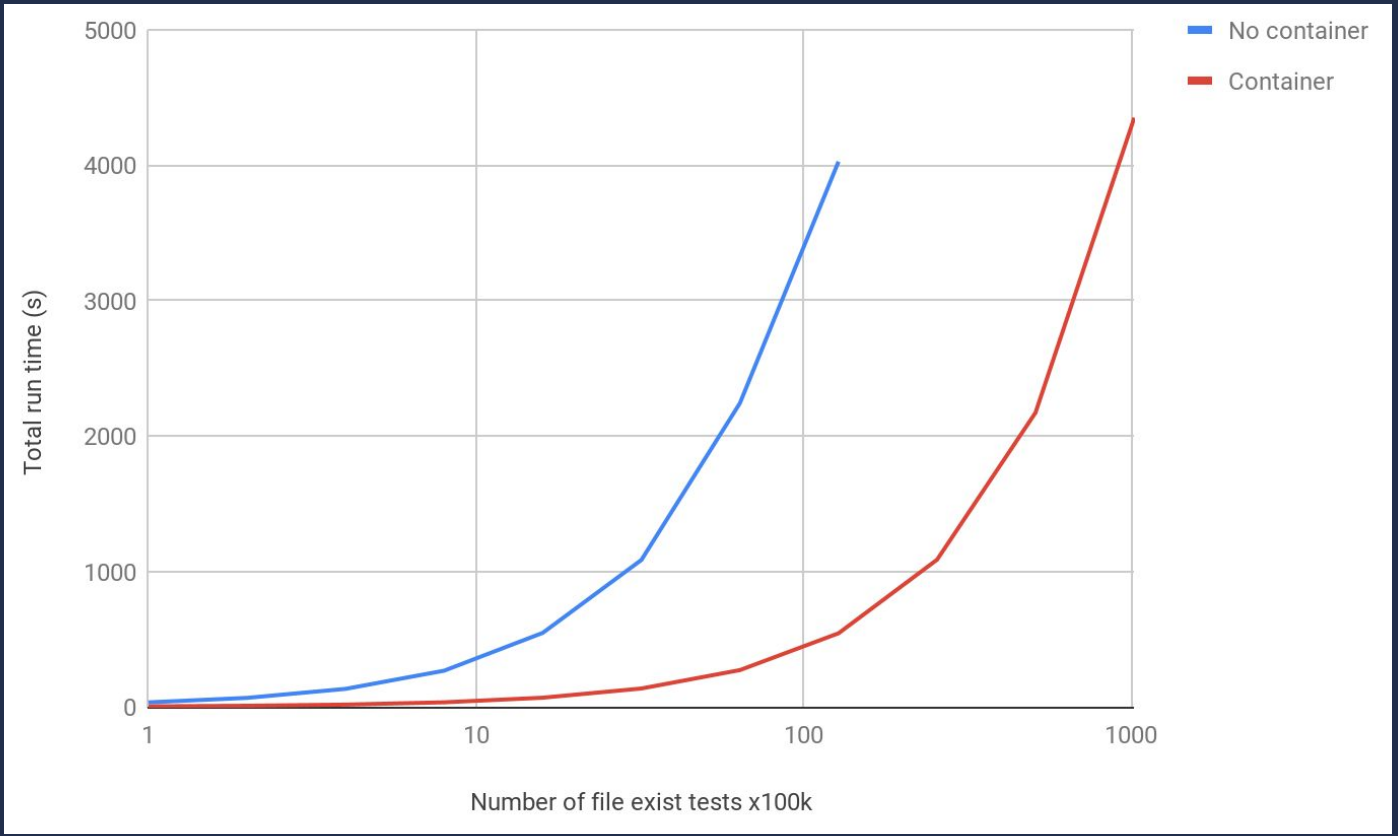
Theta Benchmarks - Bandwidth



- OSU micro benchmark
- 1000 runs for each message size
- In container vs Out

Spencer Williams, J. Taylor Childers
https://github.com/spencer-williams/sgww_argonne

Theta Benchmarks - File checks



- One script creates *n* files for *n* MPI ranks.
- Another one checks the files if they exist
- Singularity caches its files making it ~6x faster

Spencer Williams, J. Taylor Childers
https://github.com/spencer-williams/sgww_argonne

Argonne NATIONAL LABORATORY

How to start?

- Official Singularity documentation
  - https://www.sylabs.io/docs/
  - singularity-container.slack.com
- How to use Singularity on Theta
  - https://www.alcf.anl.gov/user-guides/singularity
- Similar tutorials from other HPC centers:
  - http://www.sdsc.edu/support/user_guides/tutorials/singularity.html
  - https://github.com/NIH-HPC/Singularity-Tutorial
  - https://ulhpc-tutorials.readthedocs.io/en/latest/containers/singularity/
- Github repo to check Singularity source and issues
  - https://github.com/sylabs/singularity
- Singularity registry
  - https://www.singularity-hub.org/
- Docker registry
  - https://hub.docker.com/

Argonne NATIONAL LABORATORY

How to start?

- If you have sudo access to a laptop, or even a Raspberry Pi:
  - Optional: Install Docker
    - https://docs.docker.com/get-started/
  - Install Singularity (not packaged)
    - https://www.sylabs.io/guides/2.6/user-guide/installation.html
- Else if you have a GitHub account:
  - Login Singularity Hub with your GitHub account. You can create Singularity recipe files in a GitHub repo and they will be built automatically if you link this repo to shub.
- Else:
  - Search Singularity Hub (tip: jtchilders, keceli, theta)
  - You can pull/build images from Singularity or Docker hub on Theta.
  - Open a GitHub account.

Argonne NATIONAL LABORATORY

Containers and Images

Docker manual: "Build an image and run it as one container"

- **An image is an executable package that includes everything needed to run an application--the code, a runtime, libraries, environment variables, and configuration files.**
- **A container is a runtime instance of an image--what the image becomes in memory when executed (that is, an image with state, or a user process), i.e. a container is launched by running an image, but you first build the image.**

Singularity manual: "Build a Container"

- **Uses image to refer to the *.img, *.simg files. Refers the build process as building container.**

Argonne NATIONAL LABORATORY

Using Singularity

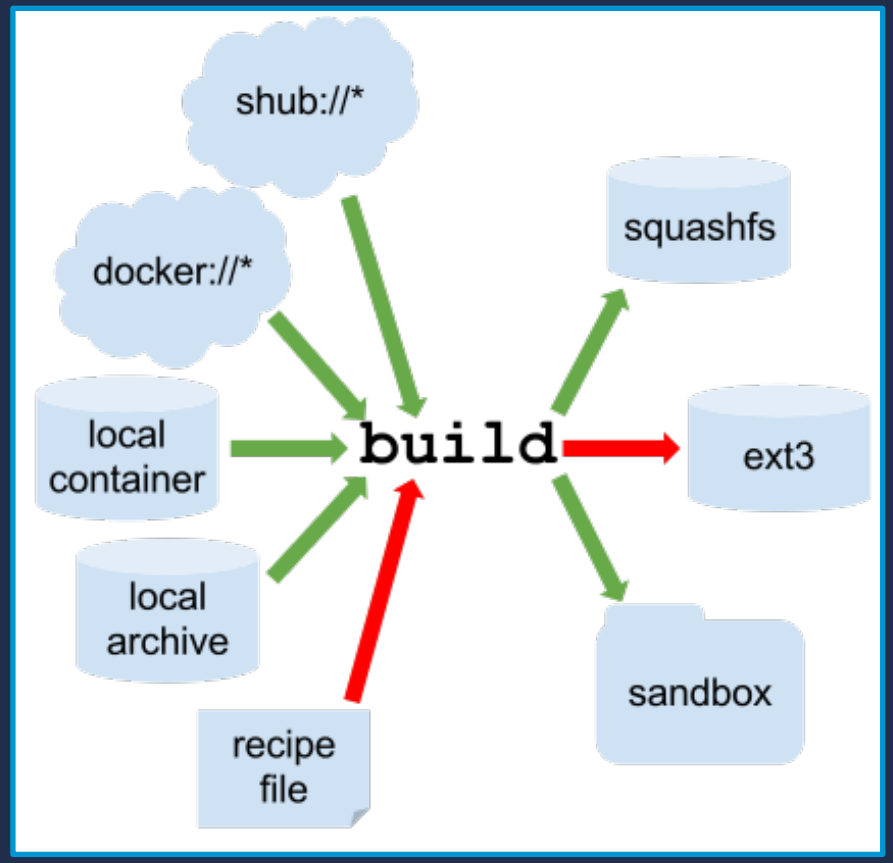
```
$> singularity
USAGE: singularity [global options...] <command> [command options...] ...

GLOBAL OPTIONS:
    -d|--debug    Print debugging information
    -h|--help     Display usage summary
    -s|--silent   Only print errors
    -q|--quiet    Suppress all normal output
       --version  Show application version
    -v|--verbose  Increase verbosity +1
    -x|--sh-debug Print shell wrapper debugging information
```

- **Before you report an error, run Singularity with -d flag**
- **-h is useful to remember about the syntax for each command**
- **Useful commands:**

```
build       Build a new Singularity container
shell       Run a Bourne shell within container
exec        Execute a command within container
run         Launch a runscript within container
apps        List available apps within a container
pull        Pull a Singularity/Docker container to $PWD
```


Argonne NATIONAL LABORATORY

Building container images



Build based on an image on a hub:
(Does not require sudo)

```
$>singularity build <OPT> <IMG> shub://xxx/yy:z
$>singularity build <OPT> <IMG> docker://xxx/yy:z
```

Build based on a recipe file (Requires sudo)

```
$>sudo singularity build <OPT> <IMG> SingularityFile
```

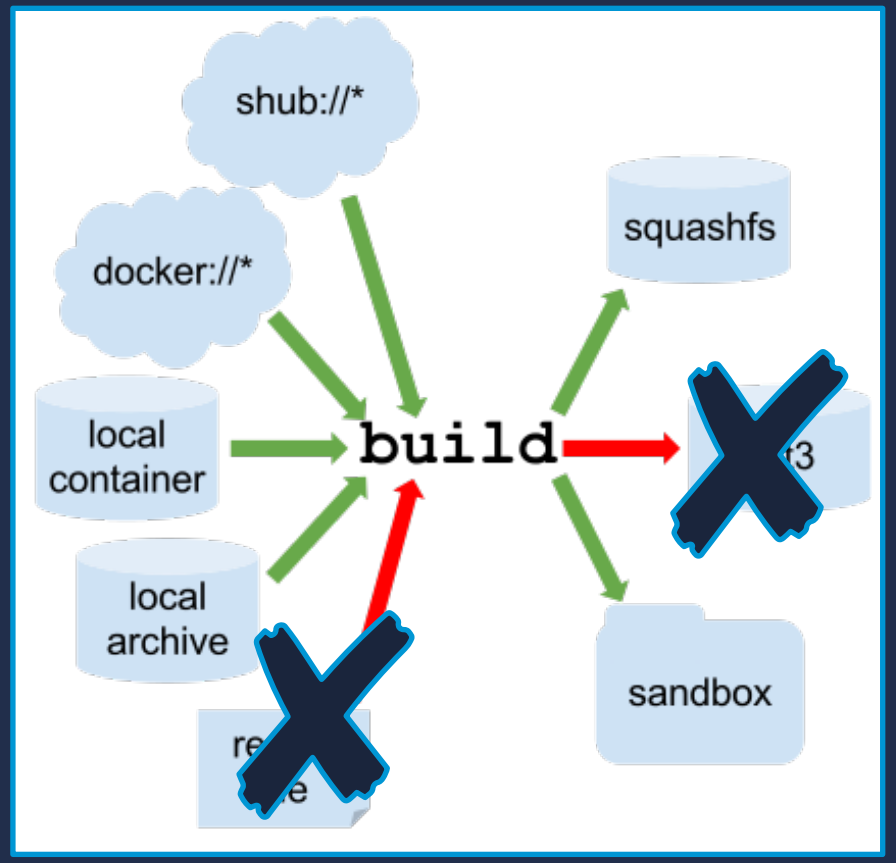
By default created image <IMG> is read only in squashfs format.
Writable image can be created: (Requires sudo)

```
--sandbox:  Writable container within a directory
--writable: Legacy writable image format (ext3)
```

https://www.sylabs.io/guides/2.6/user-guide/build_a_container.html

Argonne NATIONAL LABORATORY

Singularity on Theta



If you are not going to run the container in parallel, you can use *any* images from Singularity or Docker hub.

```
$> singularity build quip.simg shub://libAtoms/QUIP
```

You can run the image like any other application.

```
$> singularity run quip.simg
```

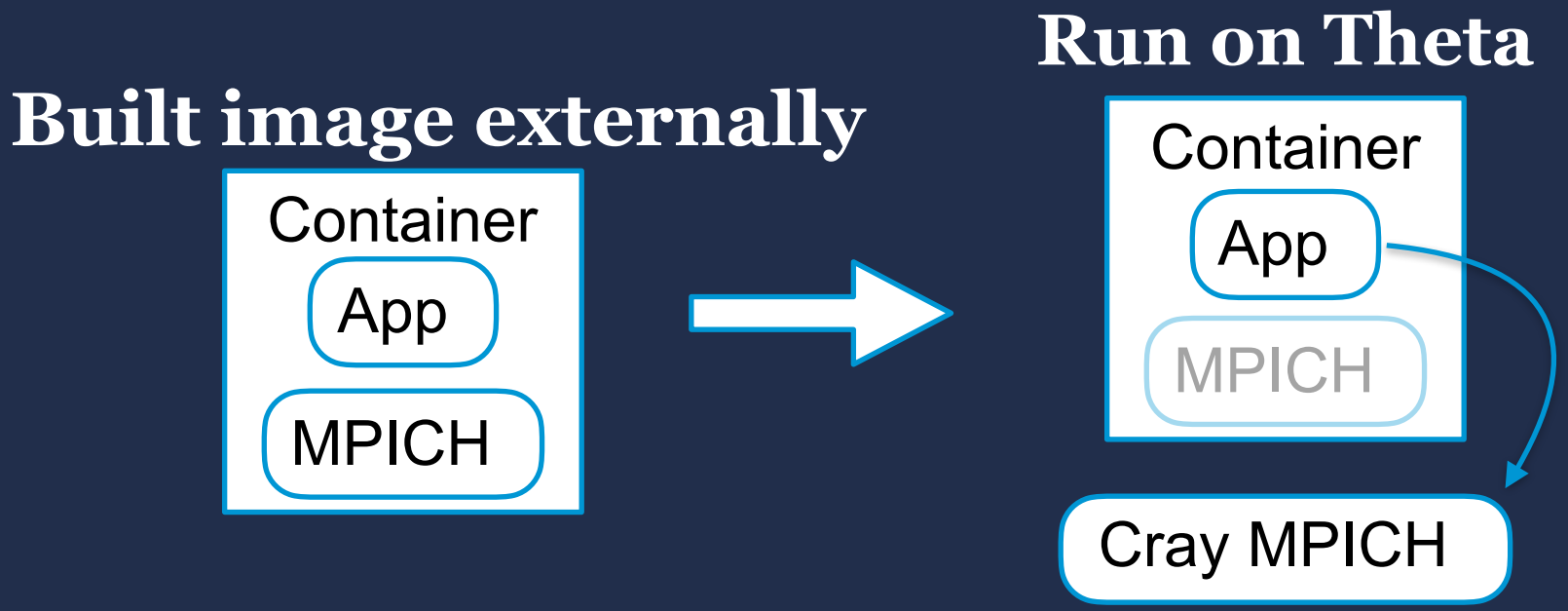
You can bind a directory on Theta to your container with -b <host_path>:<container_path>:<opt>
<opt> = ro for read-only,
<opt> = rw for read/write

```
$> singularity run -B ./mydata:/data:rw quip.simg
```

Singularity on Theta (MPI applications)

We need to use Cray MPI on Theta, so we cannot use any image.

We can build our special image with a Singularity recipe file



Source of base image

Make working directory.
Copy files from into image.

During the ‘setup’ phase, the image does not yet exist and is still on the host filesystem at the path `SINGULARITY_ROOTFS`
This creates app directory at ‘/myapp’ in the image

```
1  Bootstrap: docker
2  From: centos
3
4  %setup
5      echo ${SINGULARITY_ROOTFS}
6      mkdir ${SINGULARITY_ROOTFS}/myapp
7      cp pi.c ${SINGULARITY_ROOTFS}/myapp/
8
9  %post
10     yum update -y
11     yum groupinstall -y "Development Tools"
12     yum install -y gcc
13     yum install -y gcc-c++
14     yum install -y wget
15     cd /myapp
16     # install MPICH
17     wget http://www.mpich.org/static/downloads/3.2.1/mpich-3.2.1.tar.gz
18     tar xf mpich-3.2.1.tar.gz
19     cd mpich-3.2.1
20     # disable the addition of the RPATH to compiled executables
21     # this allows us to override the MPI libraries to use those
22     # found via LD_LIBRARY_PATH
23     ./configure --prefix=$PWD/install --disable-wrapper-rpath
24     make -j 4 install
25     # add to local environment to build pi.c
26     export PATH=$PATH:$PWD/install/bin
27     export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$PWD/install/lib
28     cd ..
29     mpicc -o pi -fPIC pi.c
30
31 %runscript
32     /myapp/pi
```

Source of base image

Make working directory.
Copy files from into image.

Commands required for installing your application.

Specify the executable to run with container is called

```
1  Bootstrap: docker
2  From: centos
3
4  %setup
5     echo ${SINGULARITY_ROOTFS}
6     mkdir ${SINGULARITY_ROOTFS}/myapp
7     cp pi.c ${SINGULARITY_ROOTFS}/myapp/
8
9  %post
10    yum update -y
11    yum groupinstall -y "Development Tools"
12    yum install -y gcc
13    yum install -y gcc-c++
14    yum install -y wget
15    cd /myapp
16    # install MPICH
17    wget http://www.mpich.org/static/downloads/3.2.1/mpich-3.2.1.tar.gz
18    tar xf mpich-3.2.1.tar.gz
19    cd mpich-3.2.1
20    # disable the addition of the RPATH to compiled executables
21    # this allows us to override the MPI libraries to use those
22    # found via LD_LIBRARY_PATH
23    ./configure --prefix=$PWD/install --disable-wrapper-rpath
24    make -j 4 install
25    # add to local environment to build pi.c
26    export PATH=$PATH:$PWD/install/bin
27    export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$PWD/install/lib
28    cd ..
29    mpicc -o pi -fPIC pi.c
30
31 %runscript
32    /myapp/pi
```

Source of base image

Make working directory.
Copy files from into image.

Commands to install my image with the application.

Typically containers are built to run one executable.

```
singularity run myapp.img
```

Specify the executable to run with container is called

```
1  Bootstrap: docker
2  From: centos
3
4  %setup
5     echo ${SINGULARITY_ROOTFS}
6     mkdir ${SINGULARITY_ROOTFS}/myapp
7     cp pi.c ${SINGULARITY_ROOTFS}/myapp/
8
9  %post
10    yum update -y
11    yum groupinstall -y "Development Tools"
12    yum install -y gcc
13    yum install -y gcc-c++
14    yum install -y wget
15    cd /myapp
16    # install MPICH
17    wget http://www.mpich.org/static/downloads/3.2.1/mpich-3.2.1.tar.gz
18    tar xf mpich-3.2.1.tar.gz
19    cd mpich-3.2.1
20    # disable the addition of the RPATH to compiled executables
21    # this allows us to override the MPI libraries to use those
22    # found via LD_LIBRARY_PATH
23    ./configure --prefix=$PWD/install --disable-wrapper-rpath
24    make -j 4 install
25    # add to local environment to build pi.c
26    export PATH=$PATH:$PWD/install/bin
27    export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$PWD/install/lib
28    cd ..
29    mpicc -o pi -fPIC pi.c
30
31 %runscript
32    /myapp/pi
```

Using a mpich installed Image as the base

Commands to install PETSc

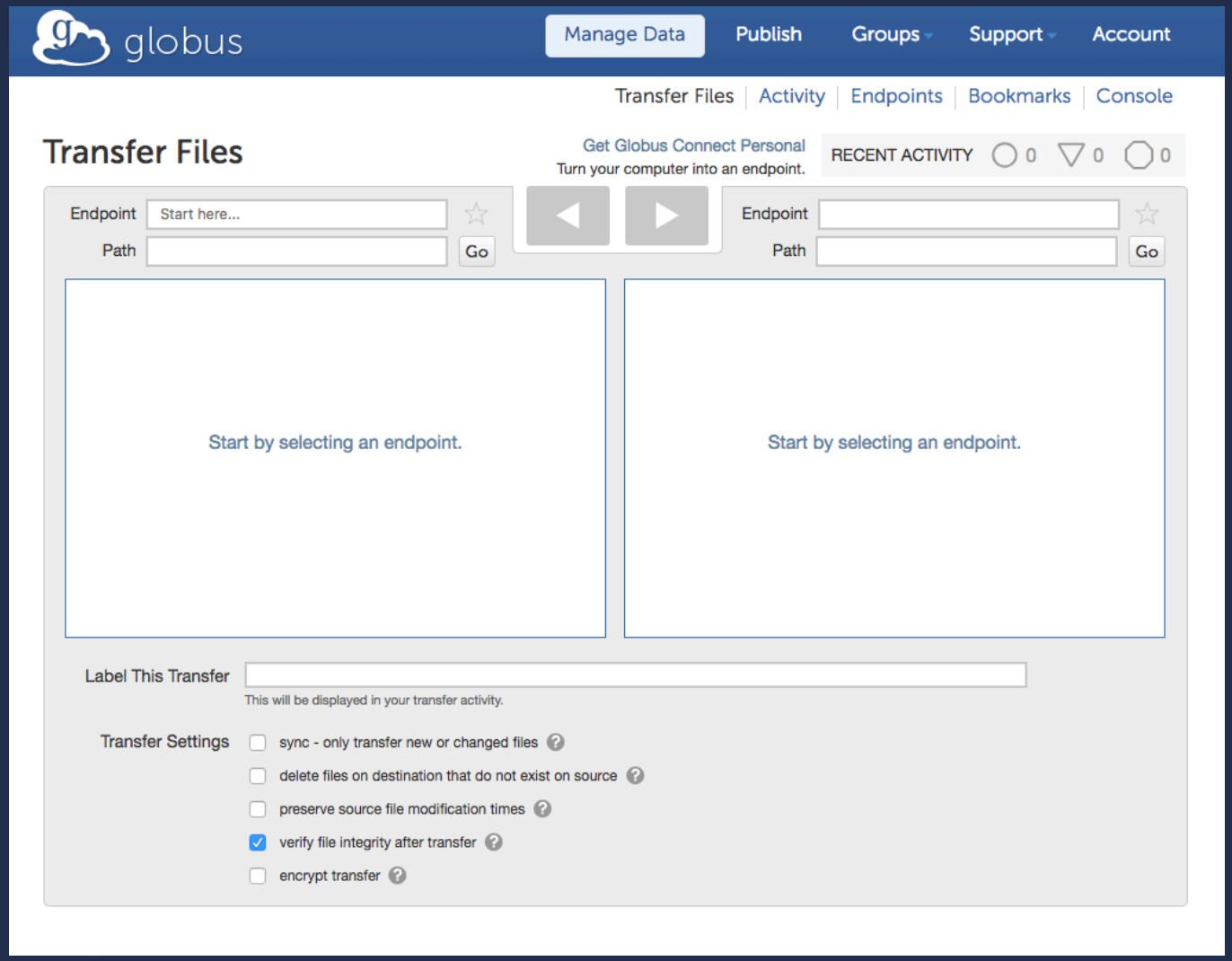
Note: This recipe builds on my laptop but gives an error on Singularity hub, since they do not allow configure script to run executables.

```
Bootstrap: shub
From: keceli/mpi_benchmark:theta

%setup
   echo ${SINGULARITY_ROOTFS}
   cd ${SINGULARITY_ROOTFS}/container

%post
   yum update -y
   git clone -b maint https://bitbucket.org/petsc/petsc petsc
   cd petsc
   PATH=$PATH:/mpich-3.2.1/install/bin/
   LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/mpich-3.2.1/install/lib
   export PETSC_DIR=/petsc
   export PETSC_ARCH=arch-container
   ./configure --with-shared-libraries=1 --with-debugging=1 --
download-fblaslapack --with-cc=mpicc --with-cxx=mpicxx --with-
fc=mpif90
   make -j 4 PETSC_DIR=/petsc PETSC_ARCH=arch-container all
   cd /petsc/src/ksp/ksp/examples/tutorials
   make ex5

%environment
   export PETSC_DIR=/petsc
   export PETSC_ARCH=arch-container

%runscript
   /petsc/src/ksp/ksp/examples/tutorials/ex5
```


Argonne NATIONAL LABORATORY

Globus for Data Transfer

- Web Interface to transfer files between Globus Endpoints (NERSC,ALCF,OLCF,BNL,etc.)
- Login using ANL Credentials or other institutes
- Must authenticate with the myproxy server of source and destination.



https://www.globus.org/app/transfer

Globus for Data Transfer

https://docs.globus.org/api/transfer/

- There is also a Python/Java API for doing this

https://github.com/globusonline/transfer-api-client-python

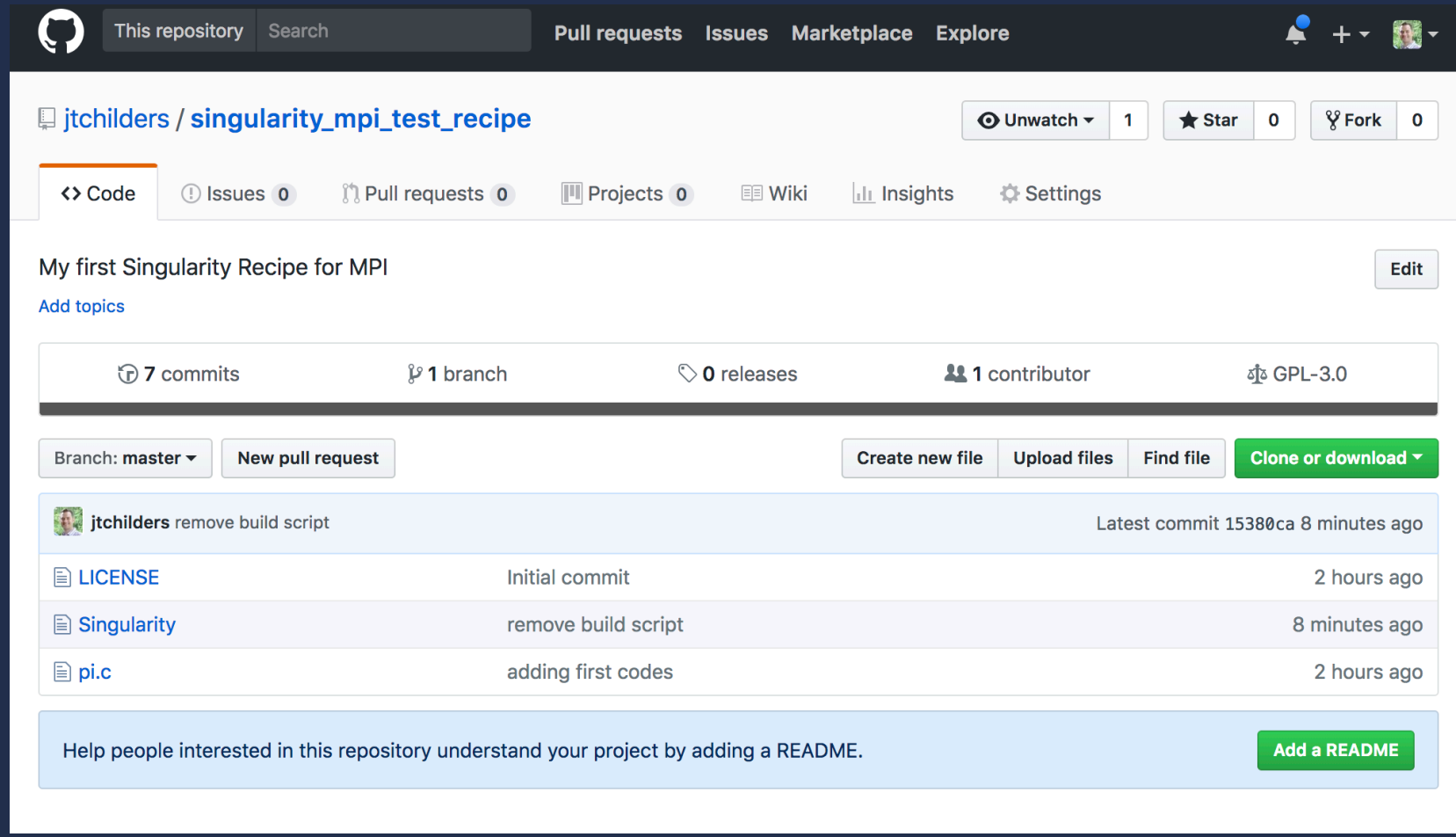
- Example Python implementation

```
from globusonline.transfer import api_client

api = api_client.TransferAPIClient(username="myusername",
                                   cert_file="/path/to/client/credential",
                                   key_file="/path/to/client/credential")
status_code, status_message, data = api.task_list()
```

- Provides effective transfer rates at the scale of 300MB/s between large facilities

Create new Github Repository

- https://github.com/jtchilders/singularity_mpi_test_recipe
- Need to add recipe file inside with filename 'Singularity'
- Add file pi.c from previous link

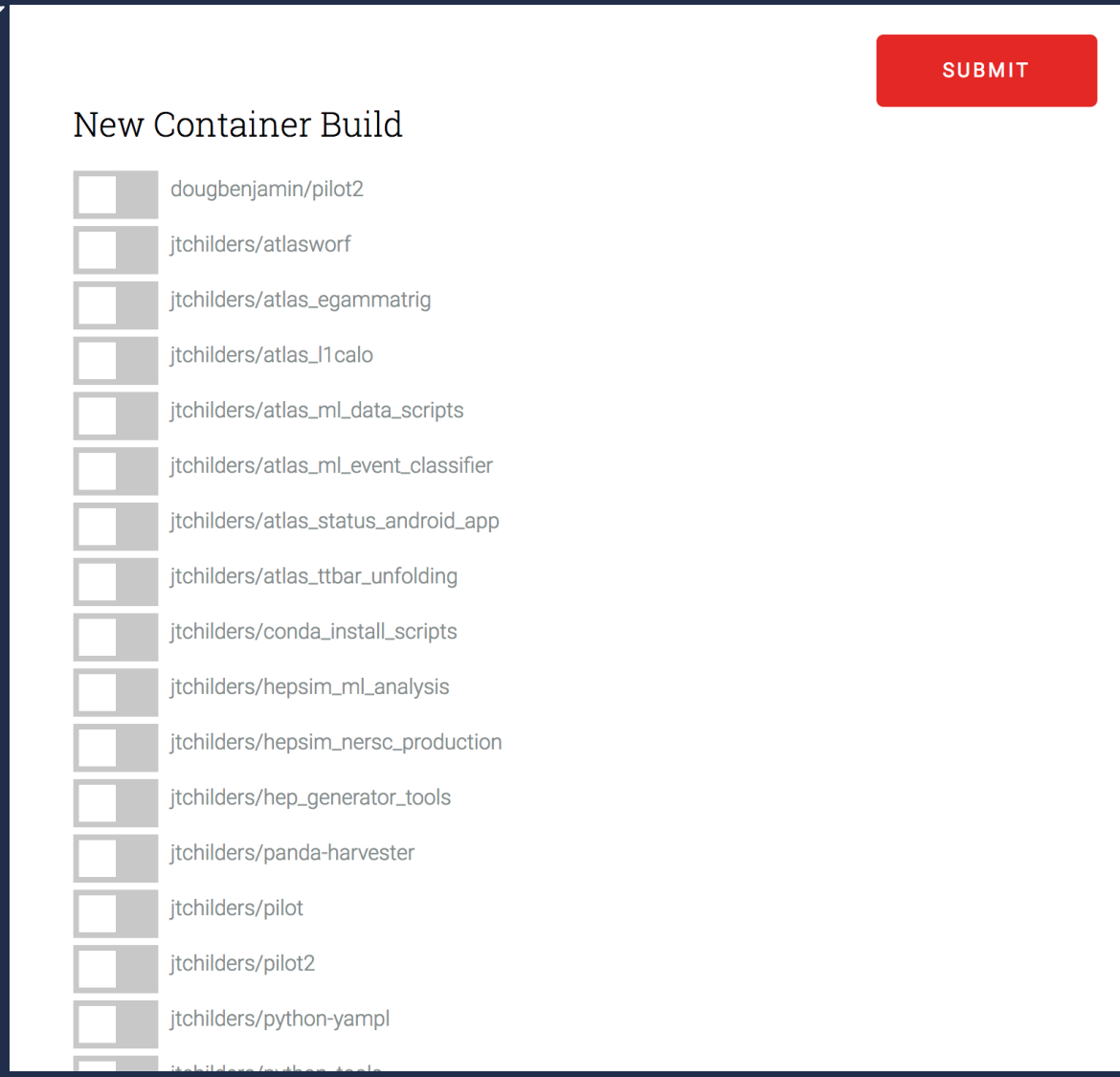


Create Singularity Hub Account

- Goto: https://www.singularity-hub.org/login/
- Authenticate using your Github account
- You can then add github repositories to your container collection.
- Click the big red button

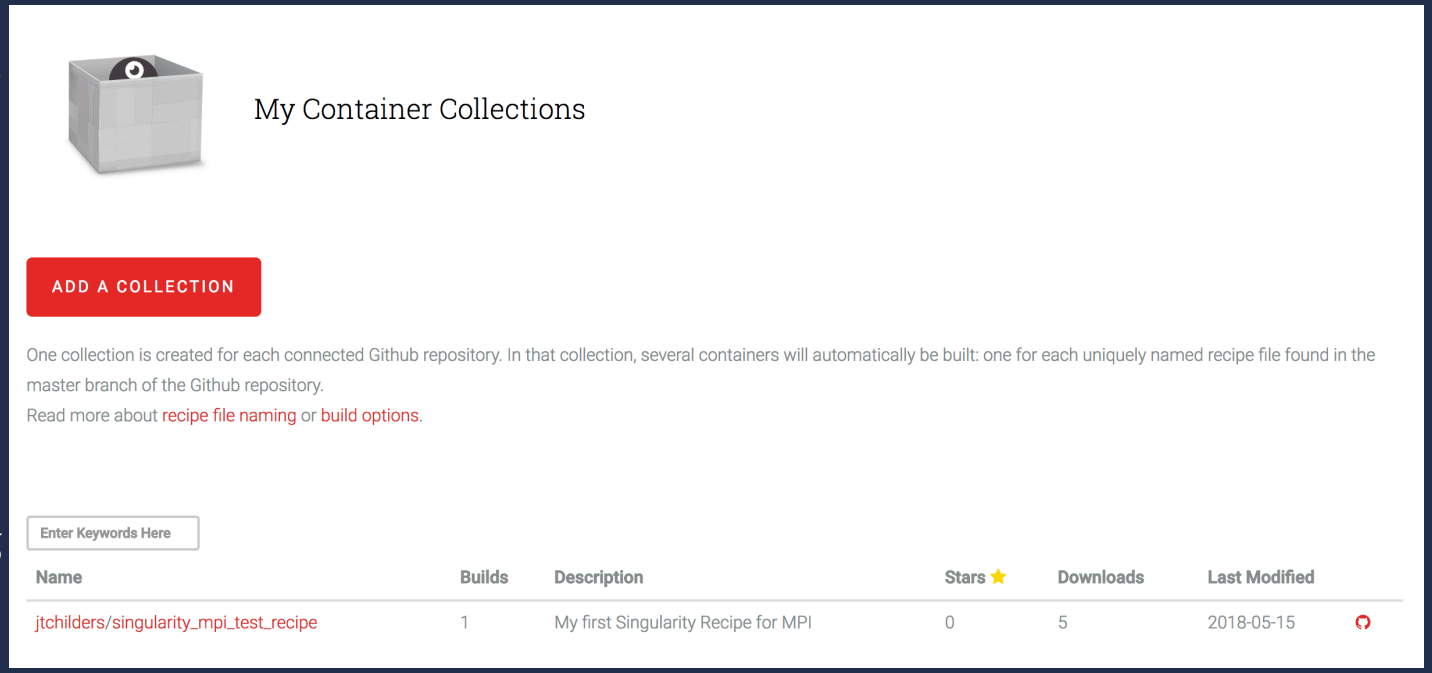
Create Singularity Hub Account

- Goto: https://www.singularity-hub.org/login/
- Authenticate using your Github account
- You can then add github repositories to your container collection.
- Click the big red button
- Select your new repository and click the big red button



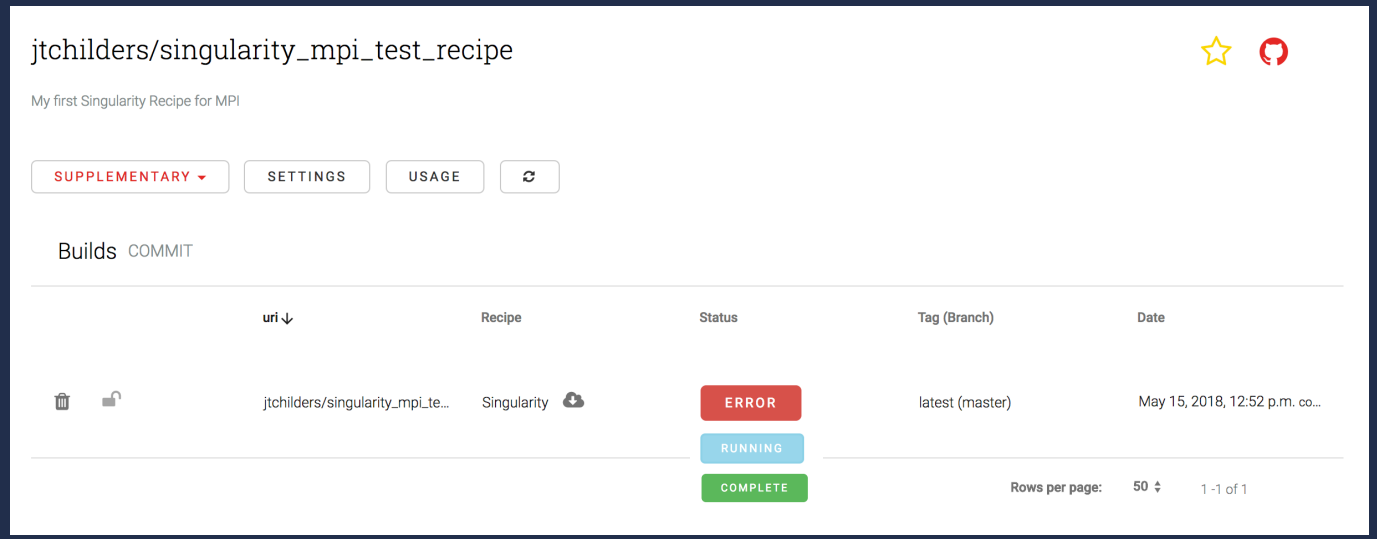
Create Singularity Hub Account

- Go to: https://www.singularity-hub.org/login/
- Authenticate using your Github account
- You can then add github repositories to your container collection.
- Click the big red button
- Select your new repository and click the big red button
- Now you have your recipe listed and Singularity Hub will begin recursively searching the repo for any files named 'Singularity' and building those recipes
- Our example only has 1 recipe
- Click on the recipe



Create Singularity Hub Account

- Goto: https://www.singularity-hub.org/login/
- Authenticate using your Github account
- You can then add github repositories to your container collection.
- Click the big red button
- Select your new repository and click the big red button
- Now you have your recipe listed and Singularity Hub will begin recursively searching the repo for any files named 'Singularity' and building those recipes
- Our example only has 1 recipe
- Click on the recipe to see it's build status
- Error messages during build can be seen by clicking the big red button
- Otherwise it will list the container as COMPLETE



Running Singularity on Theta

```
$> singularity build test.img shub://keceli/mpi_benchmark:theta
$> qsub submit.sh
```

```
#!/bin/bash
#COBALT -t 30
#COBALT -q training
#COBALT -n 2
#COBALT -A SDL_Workshop

module swap PrgEnv-intel PrgEnv-gnu
# Use Cray's Application Binary Independent MPI build
module swap cray-mpich cray-mpich-abi

export LD_LIBRARY_PATH=$CRAY_LD_LIBRARY_PATH:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH=/opt/cray/wlm_detect/1.2.1-6.0.4.0_22.1__gd26a3dc.ari/lib64/:$LD_LIBRARY_PATH
export SINGULARITYENV_LD_LIBRARY_PATH=$LD_LIBRARY_PATH

echo $SINGULARITYENV_LD_LIBRARY_PATH

aprun -n 8 -N 4 singularity run -B /opt/cray:/opt/cray:ro -B /var/opt:/var/opt:ro test.img
```

Summary

- Containers can be helpful for
  - Portability
    - HPC environment requires special care.
  - Reproducibility
  - Faster development cycles
- Minimal overhead
  - There might be additional performance penalties due to dynamic linking, fat images, moderate optimization
- Very useful for complicated software stacks with legacy dependencies.
- With more HPC interest in containers, technology will evolve faster.

Resources

- Official Singularity documentation
  - https://www.sylabs.io/docs/
  - singularity-container.slack.com
- How to use Singularity on Theta
  - https://www.alcf.anl.gov/user-guides/singularity
- Similar tutorials from other HPC centers:
  - http://www.sdsc.edu/support/user_guides/tutorials/singularity.html
  - https://github.com/NIH-HPC/Singularity-Tutorial
  - https://ulhpc-tutorials.readthedocs.io/en/latest/containers/singularity/
- Github repo to check Singularity source and issues
  - https://github.com/sylabs/singularity
- Singularity registry
  - https://www.singularity-hub.org/
- Docker registry
  - https://hub.docker.com/

Argonne NATIONAL LABORATORY

References

[1] G. M. Kurtzer, V. Sochat, and M. W. Bauer, “Singularity: Scientific containers for mobility of compute,” PLoS One, vol. 12, no. 5, pp. 1–20, 2017
[2] R. Priedhorsky and T. Randles, “Charliecloud,” Proc. Int. Conf. High Perform. Comput. Networking, Storage Anal. - SC ’17, pp. 1–10, 2017.
[3] Á. Kovács, “Comparison of different linux containers,” 2017 40th Int. Conf. Telecommun. Signal Process. TSP 2017, vol. 2017–Janua, pp. 47–51, 2017.
[4] A. J. Younge, K. Pedretti, R. E. Grant, and R. Brightwell, “A Tale of Two Systems: Using Containers to Deploy HPC Applications on Supercomputers and Clouds,” Proc. Int. Conf. Cloud Comput. Technol. Sci. CloudCom, vol. 2017–Decem, pp. 74–81, 2017.
[5] C. Arango, R. Dernat, and J. Sanabria, “Performance Evaluation of Container-based Virtualization for High Performance Computing Environments,” 2017.
[6] X. Lu and D. K. Panda, “Is Singularity-based Container Technology Ready for Running MPI Applications on HPC Clouds?,” Proc. 10th Int. Conf. Util. Cloud Comput. (UCC ’17), pp. 151–160, 2017.

Container Survey

Please vote based on your experience with:

- Virtual machines, hypervisors (VMware (1998), Virtualbox (2007))
- Docker (2013)
- Shifter (2015)
- Singularity (2016)

https://doodle.com/poll/2a723x2u9esbxyhh

or

https://tinyurl.com/thetasurvey2

Any Questions?

Argonne NATIONAL LABORATORY

References

- "A Tale of Two Systems: Using Containers to Deploy HPC Applications on Supercomputers and Clouds"
- "Charliecloud: unprivileged containers for user-defined software stacks in HPC"
- "Singularity: Scientific containers for mobility of compute"
- "Contain This, Unleashing Docker for HPC"
- "Performance Evaluation of Container-Based Virtualization for High Performance Computing Environments" **(There is a 2013 paper on IEEE with the same title)**
- "Comparison of Different Linux Containers"
-