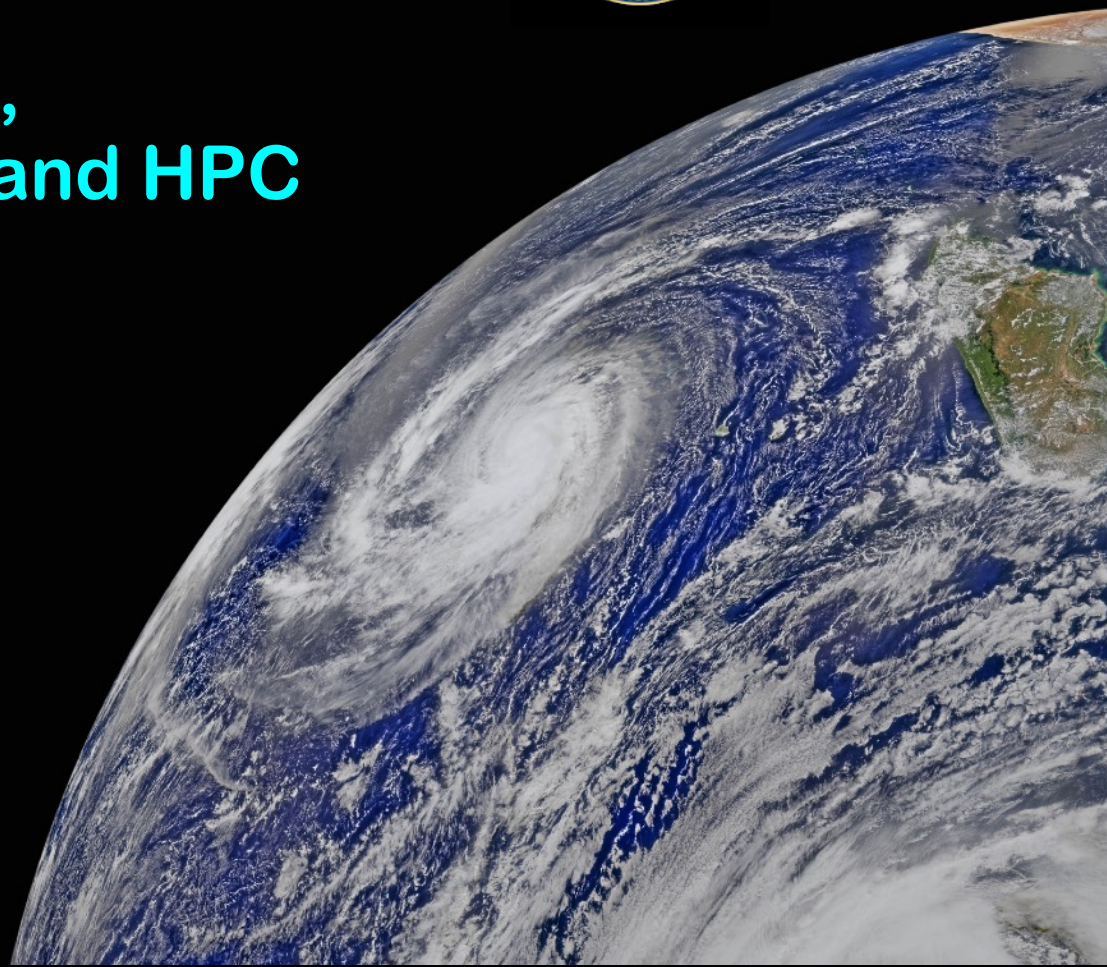
JEDI Portability Across Platforms

Containers, Cloud Computing, and HPC



Outline

I) JEDI Portability Overview
- ✦ Unified vision for software development and distribution

II) Container Fundamentals
- ✦ What are they? How do they work?
- ✦ Docker, Charliecloud, and Singularity

III) Using the JEDI Containers
- ✦ JEDI on your laptop/workstation
- ✦ JEDI in the cloud

IV) HPC and Cloud Computing
- ✦ Environment modules
- ✦ Containers in HPC?

JEDI Software Dependencies

- **Essential**
  - ✦ Compilers, MPI
  - ✦ CMake
  - ✦ SZIP, ZLIB
  - ✦ LAPACK / MKL, Eigen 3
  - ✦ NetCDF4, HDF5
  - ✦ udunits
  - ✦ Boost (headers only)
  - ✦ ecbuild, eckit, fckit
- **Useful**
  - ✦ ODB-API, eccodes
  - ✦ PNETCDF
  - ✦ Parallel IO
  - ✦ nccmp, NCO
  - ✦ Python tools (py-ncepbufr, netcdf4, matplotlib…)
  - ✦ NCEP libs
  - ✦ Debuggers & Profilers (ddt/TotalView, kdbg, valgrind, TAU…)

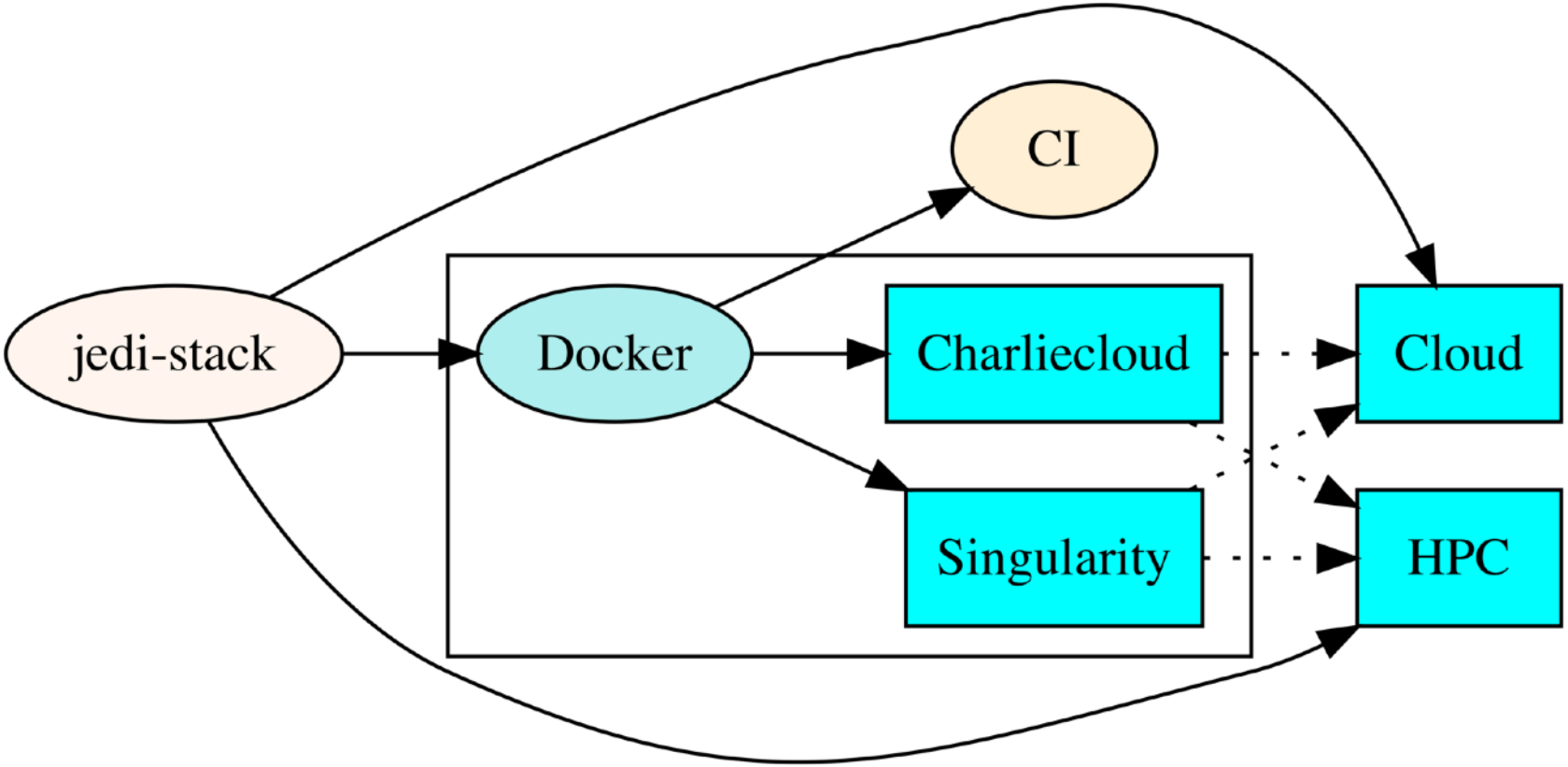
Common versions among users and developers minimize stack-related debugging

The JEDI Portability Vision

I want to run JEDI on…

- **My Laptop/Workstation/PC** — Development
  - ✦ We provide software containers and Vagrantfiles
- **In the Cloud** — Development, Applications
  - ✦ We provide containers, machine images (AMIs)
  - ✦ We (will) provide a Web-based Front End (in development)!
- **On an HPC System** — Applications
  - ✦ We provide environment modules on selected systems (S4, Discover, Cheyenne, Hera, Orion…)
  - ✦ We provide high-performance containers (in development)
  - ✦ We (will) provide access to selected HPC resources and JEDI applications via a web front end (in development)

Unified Build System



Tagged jedi-stack releases can be used to build tagged containers, AMIs, and HPC environment modules, ensuring common software environments across platforms

Part II: Container Fundamentals

Software container (working definition)

A packaged user environment that can be "unpacked" and used across different systems, from laptops to cloud to HPC

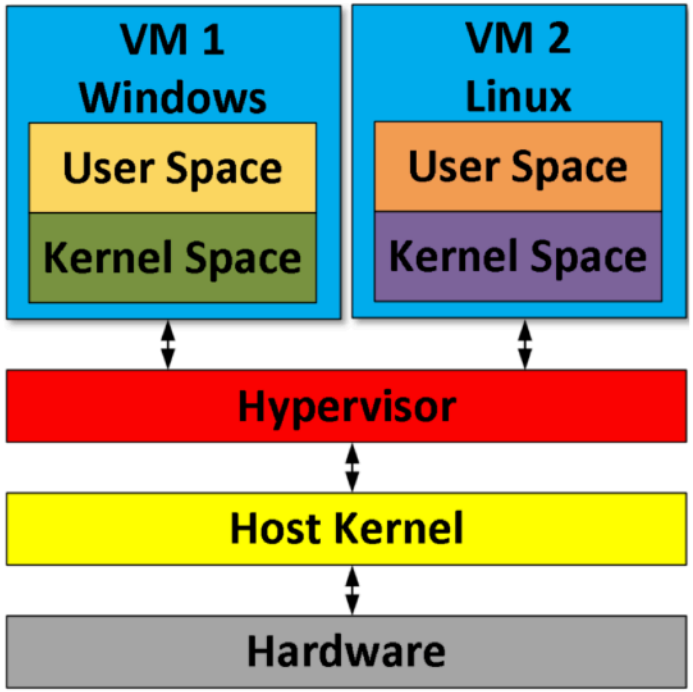
Container Benefits

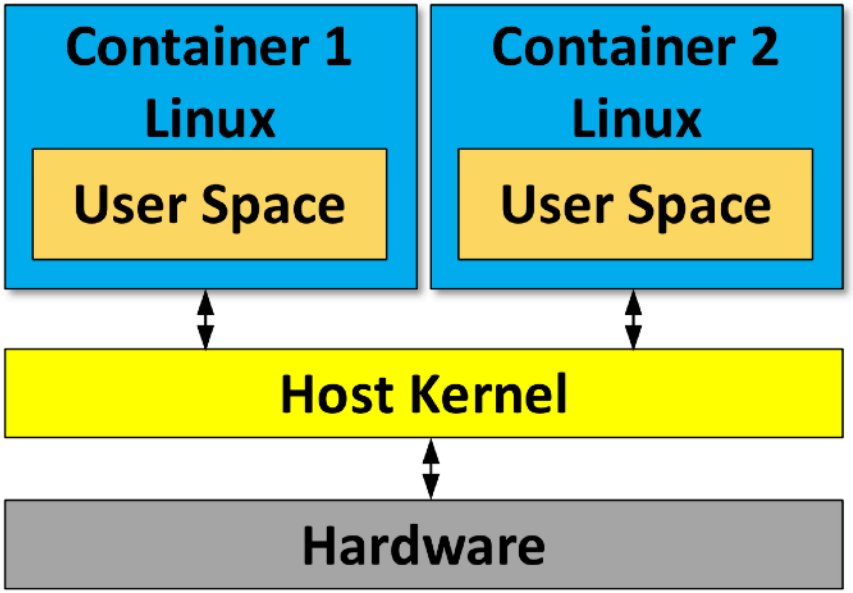
- BYOE: Bring your own Environment
- Portability
- Reproducibility
  - Version control (git)
- Workflow/Composability
  - Develop on laptops, run on cloud/HPC
  - Get new users up and running quickly

Container Providers

- Docker
- Charliecloud
- Singularity

Containers vs Virtual Machines





Containers work with the host system
Including access to your home directory

More lightweight and computationally efficient that a virtual machine

Julio Suarez
arm NEOVERSE

Example: Charliecloud

Containers exploit (linux 3.8)

User Namespaces

(..along with other linux features such as cgroups)
to define isolated user environments

```
ubuntu@ip-172-31-22-87:~/ch-jedi$ tree -L 2
.
└── ch-jedi-latest
    ├── bin
    ├── boot
    ├── dev
    ├── etc
    ├── home
    ├── lib
    ├── lib64
    ├── media
    ├── mnt
    ├── nwprod
    ├── opt
    ├── proc
    ├── root
    ├── run
    ├── sbin
    ├── srv
    ├── sys
    ├── tmp
    ├── usr
    ├── var
    ├── WEIRD_AL_YANKOVIC
    └── worktmp
```

Example:
Charliecloud

```
ch-jedi-latest/usr/local
├── bin
├── doc
├── etc
├── games
├── include
├── lib
├── man
├── sbin
├── share
└── src
```

This is where all the JEDI dependencies are installed

Example: CharlieCloud

A user "enters the container" with a simple command

```
ubuntu@ip-172-31-22-87:~/ch-jedi$ ch-run ch-jedi-latest -- bash
ubuntu@ip-172-31-22-87:/$ which ecbuild
/usr/local/bin/ecbuild
ubuntu@ip-172-31-22-87:/$ ls /usr/local/include/netcdf.h
/usr/local/include/netcdf.h
ubuntu@ip-172-31-22-87:/$
```

A user obtains the container by unpacking an image file

```
ubuntu@ip-172-31-22-87:~/ch-jedi$ wget http://data.jcsda.org/containers/ch-jedi-latest.tar.gz
--2019-05-20 18:16:43--  http://data.jcsda.org/containers/ch-jedi-latest.tar.gz
Resolving data.jcsda.org (data.jcsda.org)... 52.218.216.115
Connecting to data.jcsda.org (data.jcsda.org)|52.218.216.115|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 934877124 (892M) [application/x-tar]
Saving to: 'ch-jedi-latest.tar.gz'

ch-jedi-latest.tar.gz           100%[=====================================================>] 891.57M

2019-05-20 18:17:26 (20.9 MB/s) - 'ch-jedi-latest.tar.gz' saved [934877124/934877124]

ubuntu@ip-172-31-22-87:~/ch-jedi$ ch-tar2dir ch-jedi-latest.tar.gz .
creating new image ./ch-jedi-latest
./ch-jedi-latest unpacked ok
```

Container Technologies

- **Docker**
  - ✦ Main Advantages: industry standard, widely supported, runs on native Mac/Windows OS
  - ✦ Main Disadvantange: Security (root privileges)

- **Charliecloud**
  - ✦ Main Advantages: Simplicity, no need for root privileges
  - ✦ Main Disadvantages: Fewer features than Singularity, Relies on Docker (to build, not to run)

- **Singularity**
  - ✦ Main Advantages: Reproducibility, HPC support
  - ✦ Main Disadvantage: Not available on all HPC systems

Container Technologies

Kurtzer, Sochat & Bauer (2017)

Table 1. 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

This is why we will continue to support all three (Docker, Singularity, Charliecloud)

Container Types

- **Development Containers**
  - ✦Include dependencies as compiled binaries
  - ✦Include compilers
  - ✦JEDI code pulled from GitHub repos and built in container
- **Application Containers**
  - ✦Include dependencies as compiled binaries
  - ✦Runtime libraries only (no compilers)
  - ✦Include compiled (binary) releases of JEDI code
  - ✦Optimized for high performance

Each Distributed as Singularity and Charliecloud image files

Each tagged with release numbers to ensure consistent user environments

Part III: Using the JEDI Containers

JEDI on your Laptop/Workstation

I) Singularity container

- ✦ Easiest, quickest
- ✦ Need to install vagrant vm first for Mac, windows OS
- ✦ Described on ReadtheDocs (Vagrant, Singularity pages)

II) Docker container

- ✦ Vagrant not needed, but Docker learning curve
- ✦ Only recommended if you're already a Docker user

III) jedi-stack

- ✦ For more experienced users
- ✦ https://github.com/jcsda/jedi-stack

Using the JEDI Containers

JEDI on your Cluster/HPC system

I) Singularity container

- ✦ Easiest, quickest
- ✦ Described on ReadtheDocs (Vagrant, Singularity pages)

II) Charliecloud container

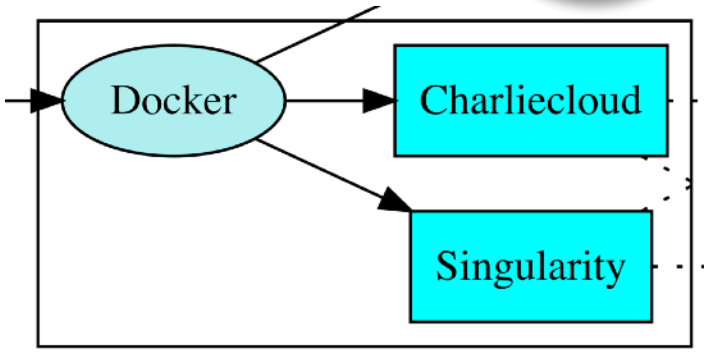
- ✦ If Singularity isn't available

III) jedi-stack

- ✦ For more experienced users
- ✦ When you're beyond the initial development stage and ready for more optimization, flexibility

Building the JEDI Containers

The JEDI Docker image is built in two steps



- **docker_base**
  - ✦ Bootstrap from ubuntu 18.04
  - ✦ Installs compilers, MPI libraries
  - ✦ Leverages NVIDIA's HPC container maker to optimize MPI configuration (e.g. Mellanox drivers for infiniband)
    https://github.com/NVIDIA/hpc-container-maker

- **docker**
  - ✦ Bootstraps from docker_base
  - ✦ Build and installs jedi-stack

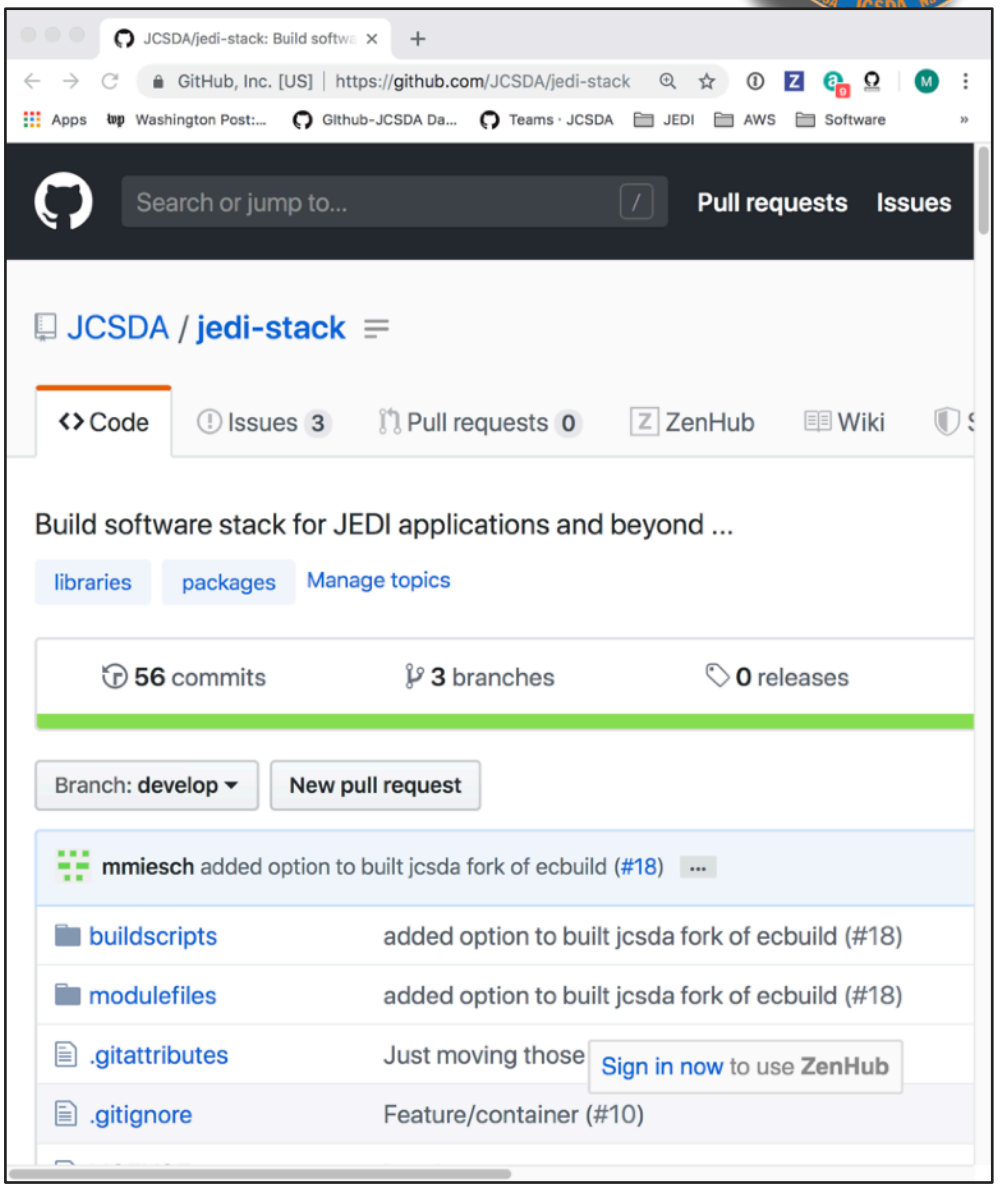
JEDI Stack

Jedi-stack is a public repo

Installs customizable hierarchy of environment modules for different compiler/mpi combinations

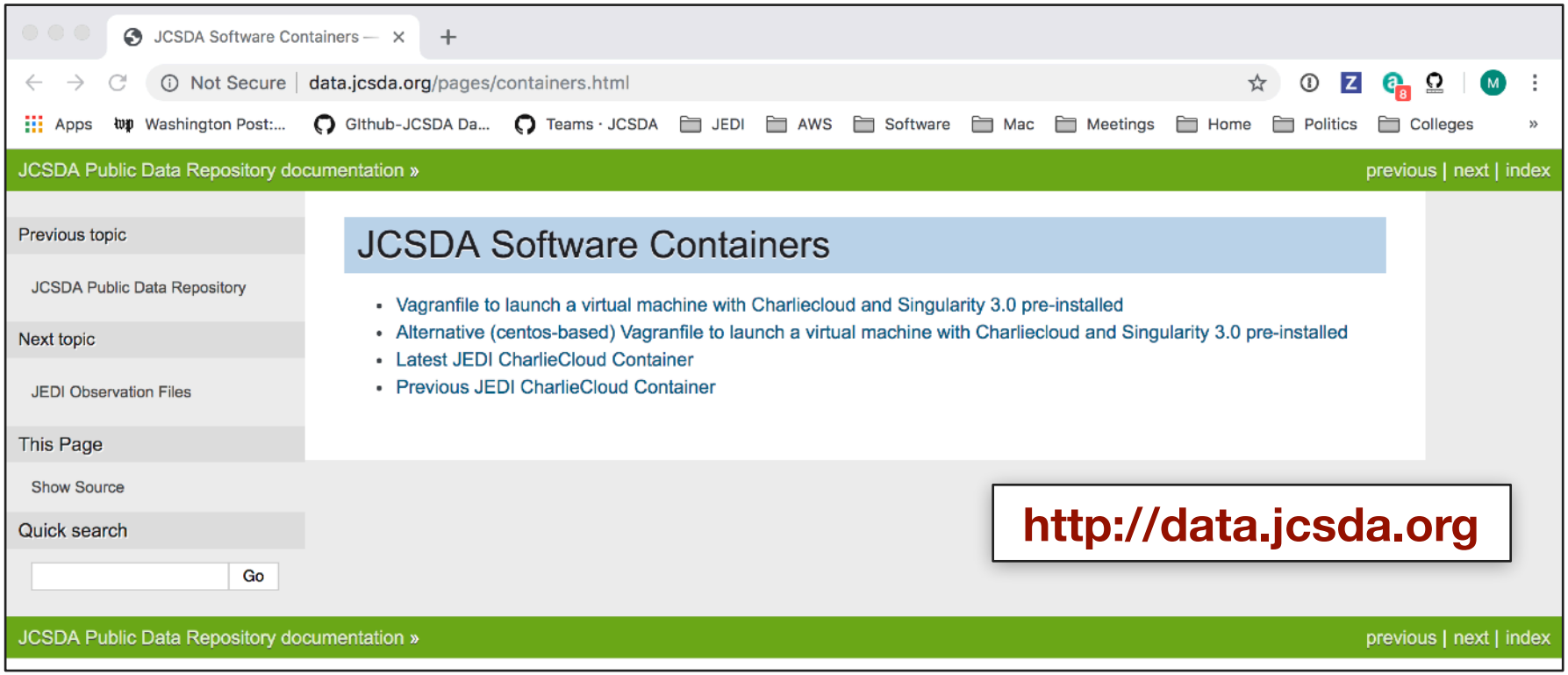
Used for AWS, Cheyenne, Discover, S4, Theia, Hera, Orion, Mac OSX

No modules in containers
Libs installed in /usr/local
Separate container for each compiler/MPI combo



How to get the JEDI Charliecloud container

JCSDA Public Data Repository




```
wget http://data.jcsda.org/containers/ch-jedi-gnu-openmpi-dev.tar.gz
ch-tar2dir ch-jedi-gnu-openmpi-dev.tar.gz
ch-run ch-jedi-latest — bash
```

How to install Charliecloud

```
mkdir ~/build
cd ~/build
git clone --recursive https://github.com/hpc/charliecloud.git
cd charliecloud
make
make install PREFIX=$HOME/charliecloud
```

You can install this yourself in your home directory
Even if you do not have root privileges
No need to rely on system administrators

How to get the JEDI Singularity Container

Sylabs ZCloud

Root privileges required to install but not to run Singularity

```
singularity pull library://jcsda/public/jedi-gnu-openmpi-dev
singularity shell -e jedi-gnu-openmpi_latest.sif
```

Using the Containers on a Mac

Mac OS does not currently support the linux user namespaces and other features that many container technologies rely on

So, to run Singularity or Charliecloud on a Mac you have to first create a linux environment by means of a virtual machine (VM)

Vagrant (HashiCorp) provides a convenient interface to Oracle's Virtualbox VM platform

```
brew cask install virtualbox
brew cask install vagrant
brew cask install vagrant-manager
```

Similar actions needed on a Windows Machine

JEDI Vagrantfile

We provide a Vagrant configuration file that is provisioned with both Singularity and Charliecloud

```
wget http://data.jcsda.org/containers/Vagrantfile
vagrant up
vagrant ssh
```

For much more information on how to use Vagrant, Singularity, and Charliecloud, see the JEDI Documentation

https://jointcenterforsatellitedataassimilation-jedi-docs.readthedocs-hosted.com

Current JEDI Containers

Currently available JEDI public development containers
(Singularity, Charliecloud, Docker)

- gnu/7.3.0-openmpi/3.1.2
- clang/8.0.0-mpich/3.3.1 (with gfortran 7.3)

Currently available JEDI private development containers
(Charliecloud, Docker)

- intel/impi 17.0.1
- intel/impi 19.0.5

amazon web services

JCSDA provides a public ubuntu 18.04 AMI that comes with Singularity, Charliecloud, and Docker pre-installed

Part IV: HPC and Cloud Computing

Containers in HPC?

- An attractive option, particularly for new JEDI users
- Need to access native compilers, MPI for peak performance

Containers in the Cloud?

- Can be an attractive option but sometimes unnecessary with the availability of machine images (e.g. AMIs)

Environment Modules

- Greater flexibility for testing and optimization
  - JEDI Test Node on AWS
- Maximum Performance (built from native compiler/mpi modules)
- Maintained on selected HPC systems (S4, Discover, Cheyenne, Hera, Orion…)

Environment modules

```
ubuntu@ip-172-31-20-178:/opt/modules$ tree -L 2
.
├── clang-6.0.1
│   ├── lapack
│   ├── openmpi
│   ├── openmpi-3.1.2
│   ├── szip
│   ├── udunits
│   └── zlib
├── core
│   ├── boost
│   ├── ecbuild
│   └── eigen
├── gnu-7.4.0
│   ├── lapack
│   ├── mpich
│   ├── mpich-3.2.1
│   ├── openmpi
│   ├── openmpi-3.1.2
│   ├── szip
│   ├── udunits
│   ├── xerces
│   └── zlib
├── intel-17.0.1
│   ├── impi-17.0.1
│   ├── szip
│   ├── udunits
│   └── zlib
└── modulefiles
    ├── compiler
    ├── core
    └── mpi
```

```
ubuntu@ip-172-31-20-178:/opt/modules$ tree -L 2 gnu-7.4.0
gnu-7.4.0
├── lapack
│   └── 3.7.0
├── mpich
│   └── 3.2.1
├── mpich-3.2.1
│   ├── eckit
│   ├── fckit
│   ├── hdf5
│   ├── nccmp
│   ├── nco
│   ├── netcdf
│   ├── odb-api
│   ├── pio
│   └── pnetcdf
├── openmpi
│   └── 3.1.2
├── openmpi-3.1.2
│   ├── eckit
│   ├── fckit
│   ├── hdf5
│   ├── nccmp
│   ├── nco
│   ├── netcdf
│   ├── odb-api
│   ├── pio
│   └── pnetcdf
├── szip
│   └── 2.1.1
├── udunits
│   └── 2.2.26
├── xerces
│   └── 3.1.4
└── zlib
    └── 1.2.11
```

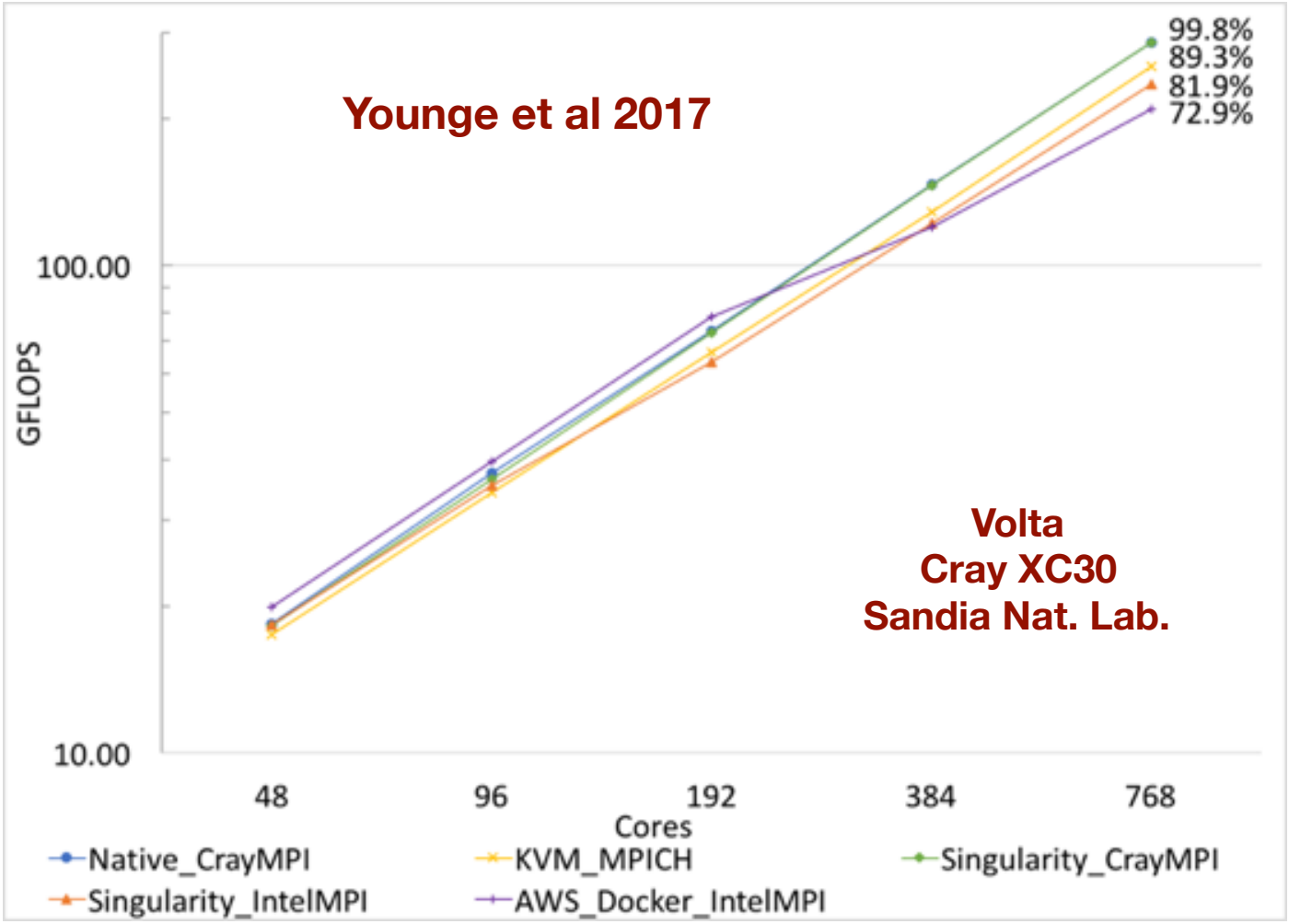
JEDI test node on AWS

Similar structure on HPC systems

Tagged “Meta-Modules” linked with container releases

```
module load jedi/gnu-openmpi
module load jedi/intel-impi
```

Containers can achieve near-native performance (negligible overhead) but only if you tap into the native MPI libraries



HPC containers promising, but currently not "plug and play"

Containers on HPC systems

When running on a single node (sufficient for most development work)

```
singularity run mpirun -np 216 fv3jedi_var.x conf/hyb_3dvar.yaml
```

Single container for all mpi tasks

When running on multiple nodes (needed for many applications)

```
export SINGULARITY_BINDPATH="/opt/mpich/mpich-3.1.4/apps"
export SINGULARITYENV_LD_LIBRARY_PATH="/opt/mpich/mpich-3.1.4/apps/lib"
mpirun -getenv -np 216 singularity run fv3jedi_var.x conf/hyb_3dvar.yaml
```

Multiple containers: each mpi task launches its own container

Need to make sure:

- all necessary system directories are accessible from the container
- all necessary drivers are installed in the container (e.g. Mellanox infiniband)
- MPI implementations inside & outside container are compatible

Cloud computing

- ***Agile, on-demand computing resources***
- ***Get what you need and pay as you go***
- ***State-of-the-art chip hardware, services***
- ***Bring computation to data***
- ***Flexible data access / distribution***
- ***Interconnects, cost can be a down side (but getting better!)***

Cloud Computing at JCSDA (currently)

JEDI Testing/Optimization/Applications/Training

- CI with multiple compiler/mpi combinations
- Scalable configurations for Parallel applications
- JEDI Academy
- Near real-time H(x)
- …more…

NWP with FV3-GFS

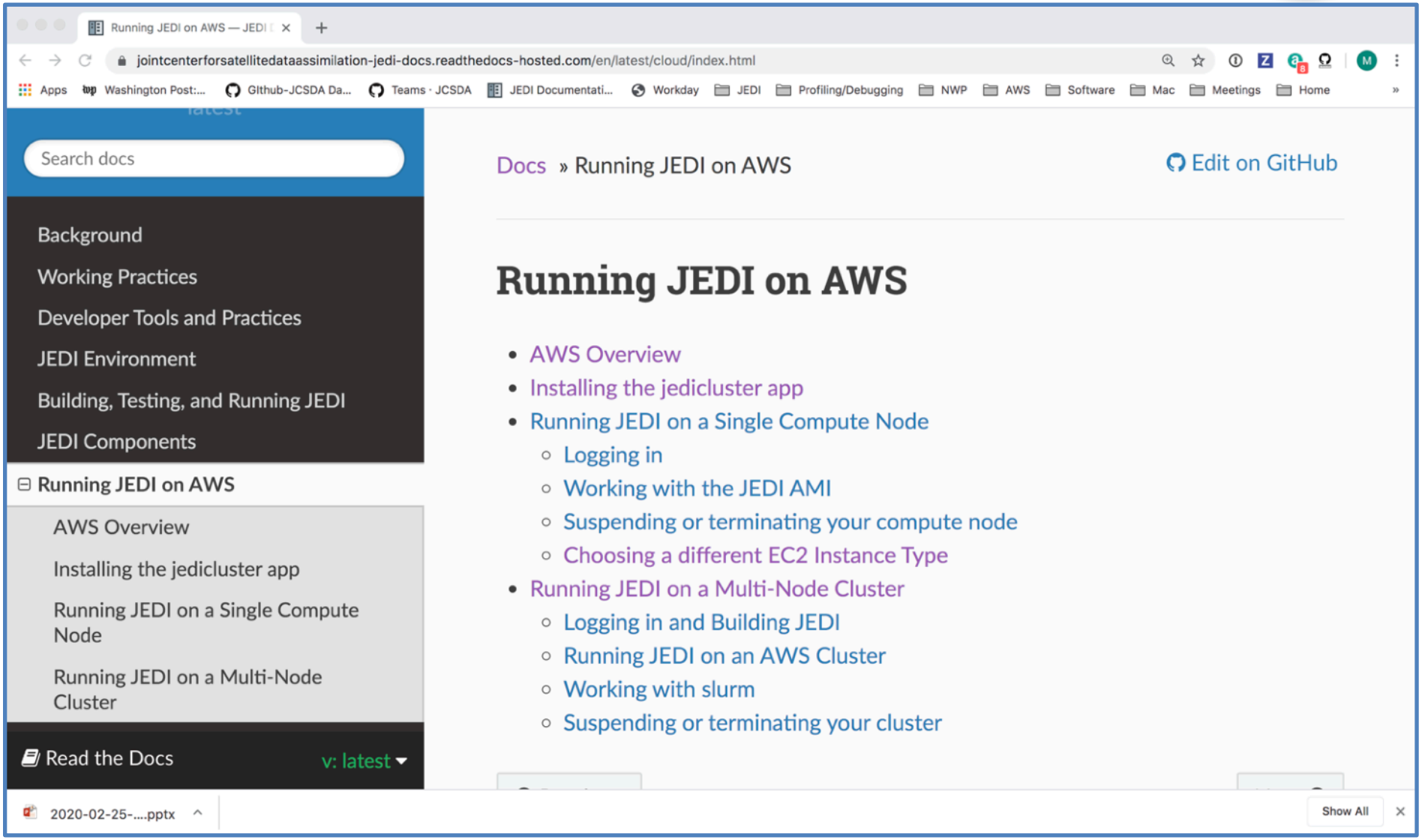
- 10-day forecast at operational resolution on AWS
  - Pre-oerational configuration
  - c5.18xlarge nodes (36 cores, 144 GiB, 25 Gbps)
  - 10-day forecast in 74 min (7.4 min/day) on 48 nodes (1536 cores)
  - 125 min (12.5 min/day) on 27 nodes (768 cores)

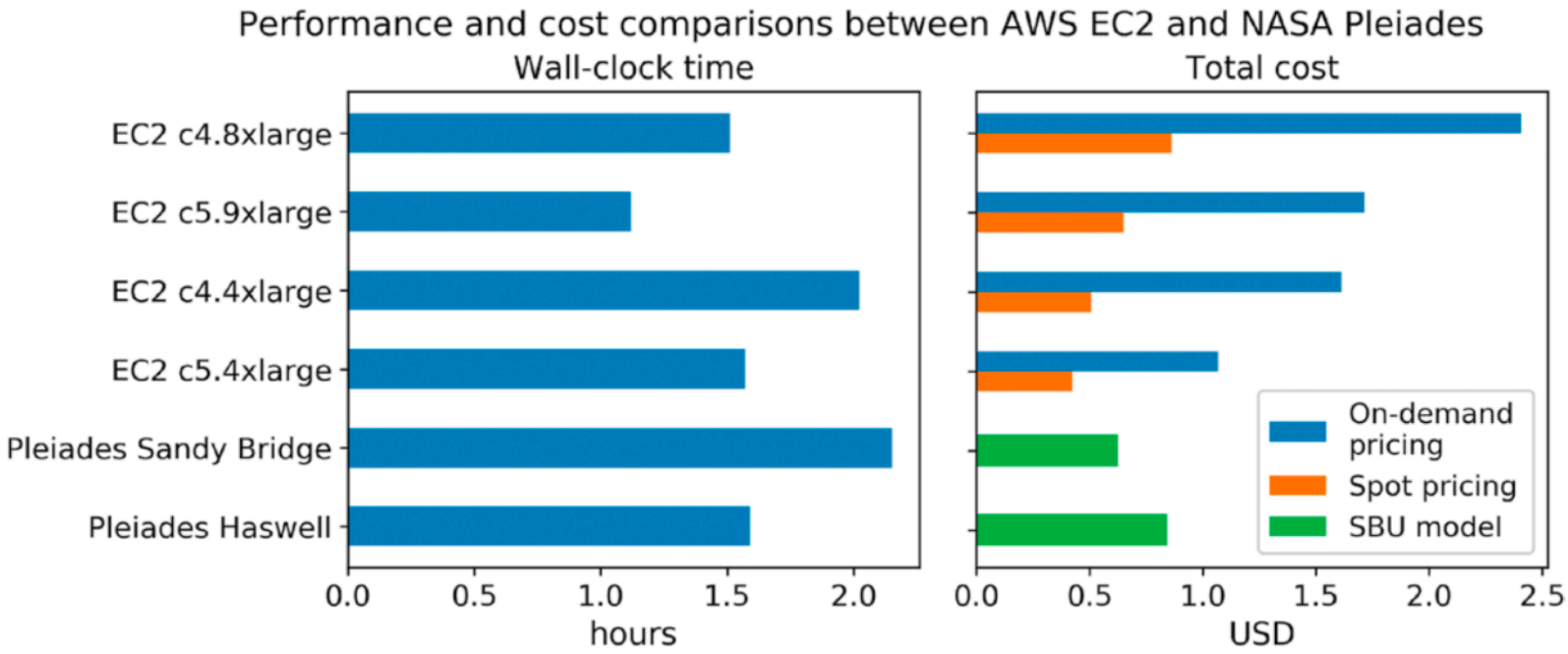
…And more

- Machine learning
- FSOI (https://ios.jcsda.org)
- Data Repository

New technology should improve performance further!
FSx, EFA

Running JEDI on AWS





GEOS-Chem atmospheric chemistry model

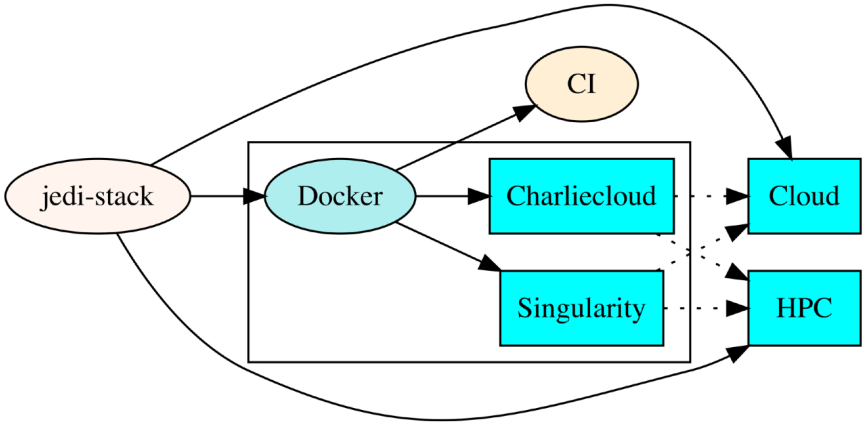
Zhuang et al 2019

Instance/node type[a]	Processor information[b]
	AWS
EC2 c4.8xlarge	Intel Xeon CPU E5–2666v3, 2.9 GHz, 32 vCPUs
EC2 c5.9xlarge	Intel Xeon Platinum 8124M, 3.0 GHz, 32 vCPUs
EC2 c4.4xlarge	Intel Xeon CPU E5–2666v3, 2.9 GHz, 16 vCPUs
EC2 c5.4xlarge	Intel Xeon Platinum 8124M, 3.0 GHz, 16 vCPUs
	NASA HECC
Pleiades Sandy Bridge	Intel Xeon E5–2680v2, 2.8 GHz, 16 CPU cores
Pleiades Haswell	Intel Xeon E5–2680v3, 2.5 GHz, 24 CPU cores

Summary

I want to run JEDI on…

- **My Laptop/Workstation/PC**
  - ✦ Singularity/Charliecloud/Vagrant
- **In the Cloud**
  - ✦ AMIs, Containers
- **On an HPC System**
  - ✦ Environment modules on selected systems (S4, Discover, Cheyenne, Hera, Orion…)
  - ✦ High-performance containers
  - ✦ jedi-stack



Unified, module-based build system with tagged releases

Performance Estimates

Preliminary comparison (in core hours) of a moderate fv3-jedi application run on 216 cores on AWS and Discover

	AWS (6 c5n.18xlarge nodes)	Discover
bumpparameters_loc_geos	1.7	26
bumpparameters_cor_geos	11	39
hyb-3dvar_geos	8.8	7.7

Cheyenne	Native	Charliecloud
FV3-bundle unit tests	808.19 s	808.52 s