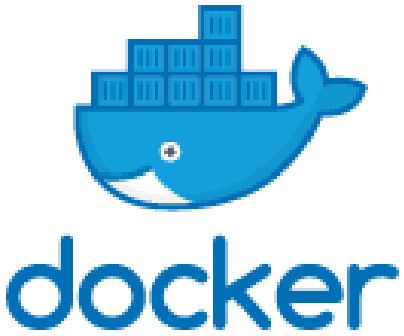
A **Smalltalk** on **Singularity** and **Containers**



CRUNCH GROUP
Brown University

Why this (Pep)Talk?

- Python 3.8 does not support TensorFlow 1.*
- New GPUs on Oscar **3090-gcondo** and **a6000-gcondo** are of Ampere Architecture and does not support < **cuda11** and < **cudann 8.0**
- Old GPUs **gcondo** does not support CUDA11.

Demo

```
sinfo -s  # Try
```

```
PARTITION     AVAIL  TIMELIMIT   NODES(A/I/O/T) NODELIST
batch*           up   infinite     233/0/35/268 node[1105-1128,1139,1161-1164,1305-1315,1317-1328,133
gpu              up   infinite       23/11/0/34 gpu[1201-1204,1401-1403,1414,2001-2010,2101-2116]
gpu-he           up   infinite         2/5/4/11 gpu[1210-1212,1404-1405,1501-1506]
bigmem           up   infinite          3/1/0/4 node[1609-1612]
vnc              up   infinite        14/9/0/23 gpu[717-718],node[1140-1160]
debug            up   infinite          0/4/0/4 node[1301-1304]
scavenge         up   infinite          0/1/0/1 node1301
viz              up   infinite          0/3/0/3 gpu[1206-1207,1209]
gpu-debug        up   infinite          0/3/0/3 gpu[1206-1207,1209]
gcondo           up   infinite          2/6/0/8 gpu[2101-2103,2109-2113]
3090-gcondo      up   infinite        6/10/0/16 gpu[2101-2116]
a6000-gcondo     up   infinite          1/1/4/6 gpu[1501-1506]
[kshukla1@login005 Singularity_Practice]$
```

What is Containers and Singularity?

- A container facilitate to put an application and all of its dependencies into a single package: **Plug and Play** * (EK)
- Container makes your code portable, shareable, and reproducible.
- Containers foster portability and reproducibility because they package **ALL** of an applications dependencies including its own tiny operating system!
- This means your application won't break when you port it to a new environment. Your app brings its environment with it.
- For reproducing result of your research paper, include a link to a container with all of the data and software that you used so that others can easily reproduce your results

Containers versus Virtual Machine

- **Install every last bit of an operating system (OS) right down to the core software that allows the OS to control the hardware (called the *kernel*).**

- **VMs are slow and resource hungry. You start a VM it has to bring up an entirely new OS.**

- **Not Flexible as VM. A Linux container must be run on a Linux host OS. (Although you can mix and match distributions.) In practice, containers are only extensively developed on Linux.**

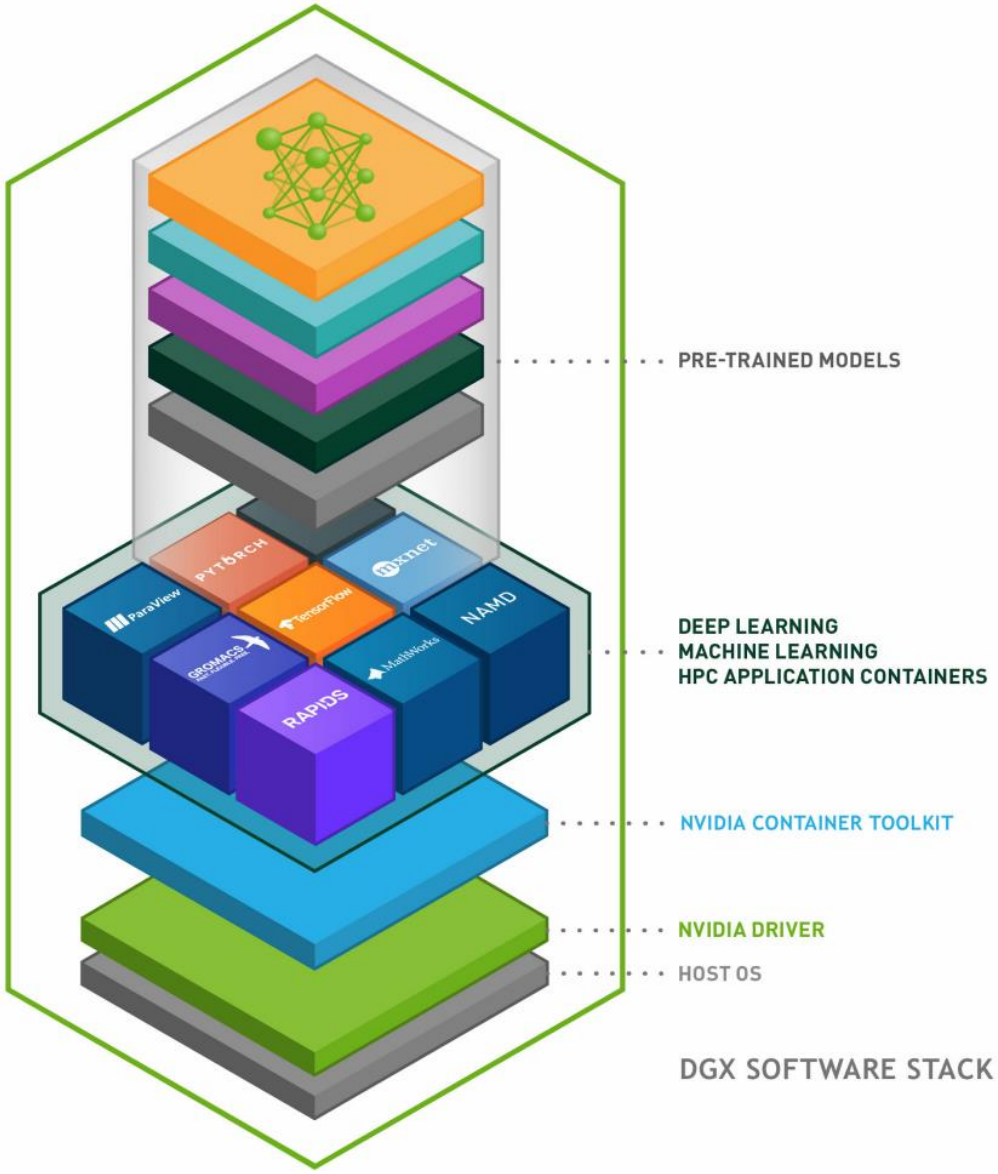
- Much faster and lighter weight than VMs. Just a few MB.

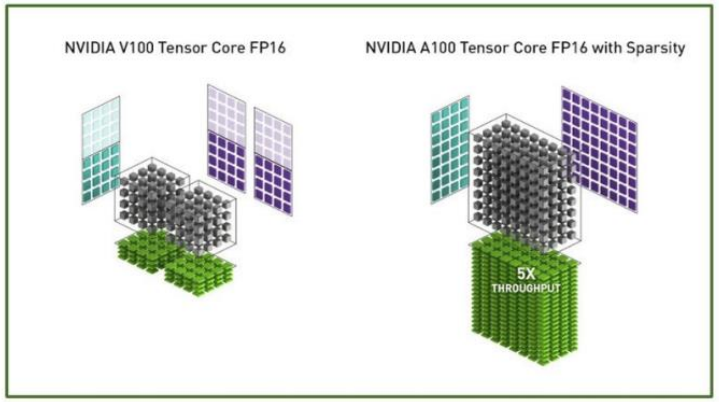
- Start and stop quickly and are suitable for running single apps.

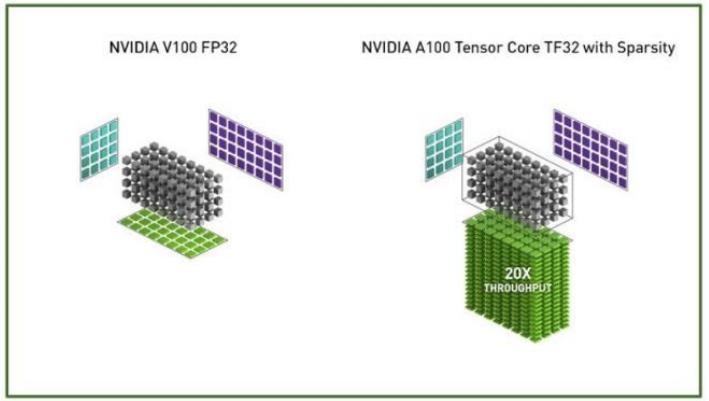
Few Words about Singularity

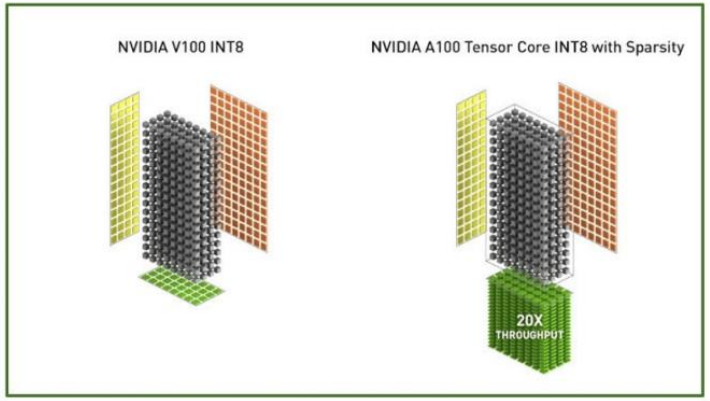
- Singularity is a new container software invented by Greg Kurtzer while at Lawrence Berkley National labs.
- Now developed by his company Sylabs.
- It was developed with security, scientific software, and HPC systems in mind.

NVIDIA A100 GPU Verticals and Tensorcores









Lets build the Image for TensorFlow

Demo

Go to NVIDIA NGC Container:

https://catalog.ngc.nvidia.com

On Login node

singularity build crunch_tf2.simg docker://nvcr.io/nvidia/tensorflow:21.12-tf2-py3

Get a GPU:

interact –q 3090-gcondo –n 8 –g 1 –m 32g –t 1:00:00

Get A Singularity Shell

singularity shell -B /gpfs/scratch,/gpfs/data --nv crunch_tf2.simg

Or Export File system as Using Singularity variable:

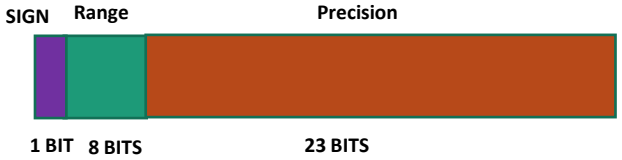
export SINGULARITY_BINDPATH="/gpfs/scratch,/gpfs/data”

singularity shell–nv crunch_tf2.simg

Lets Run a Code: interact mode

IEEE-754

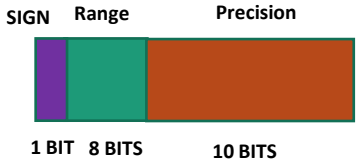
Floating Point: FP32:



1.57 = 0 011 1111 1100 1000 1111 0101 1100 0011

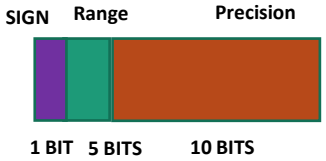
Stored Value in Comp: 1.57000005245208740234375

TensorFlow32: TF32



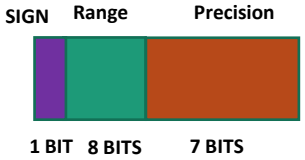
1.57 = 0 011 1111 1100 1000 11 ~~11 0101 1100 0011~~

Float16: FP16
Half Precision



1.57 = 0 011 11~~11~~ 1100 1000 11 ~~11 0101 1100 0011~~

BFLOAT16:



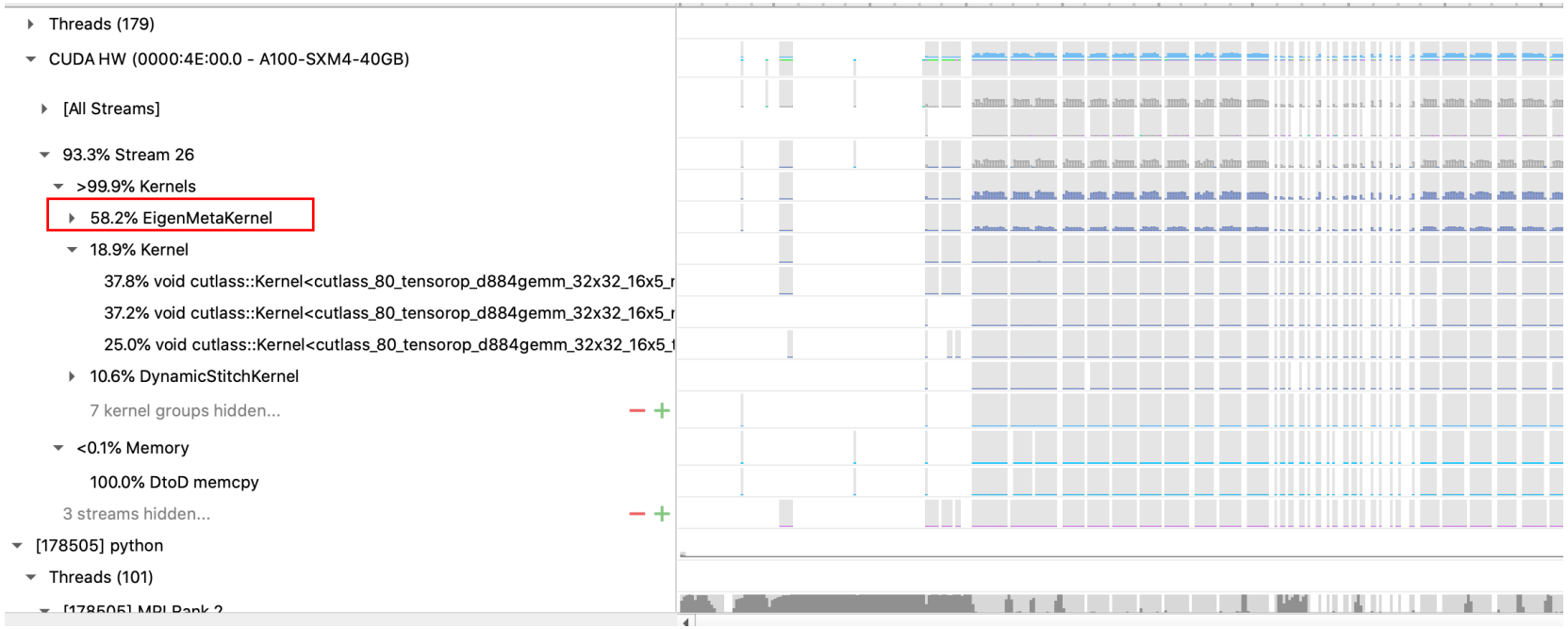
1.57 = 0 011 1111 1100 100~~0 1111 0101 1100 0011~~

or more information, see Performance models and Benchmarks.

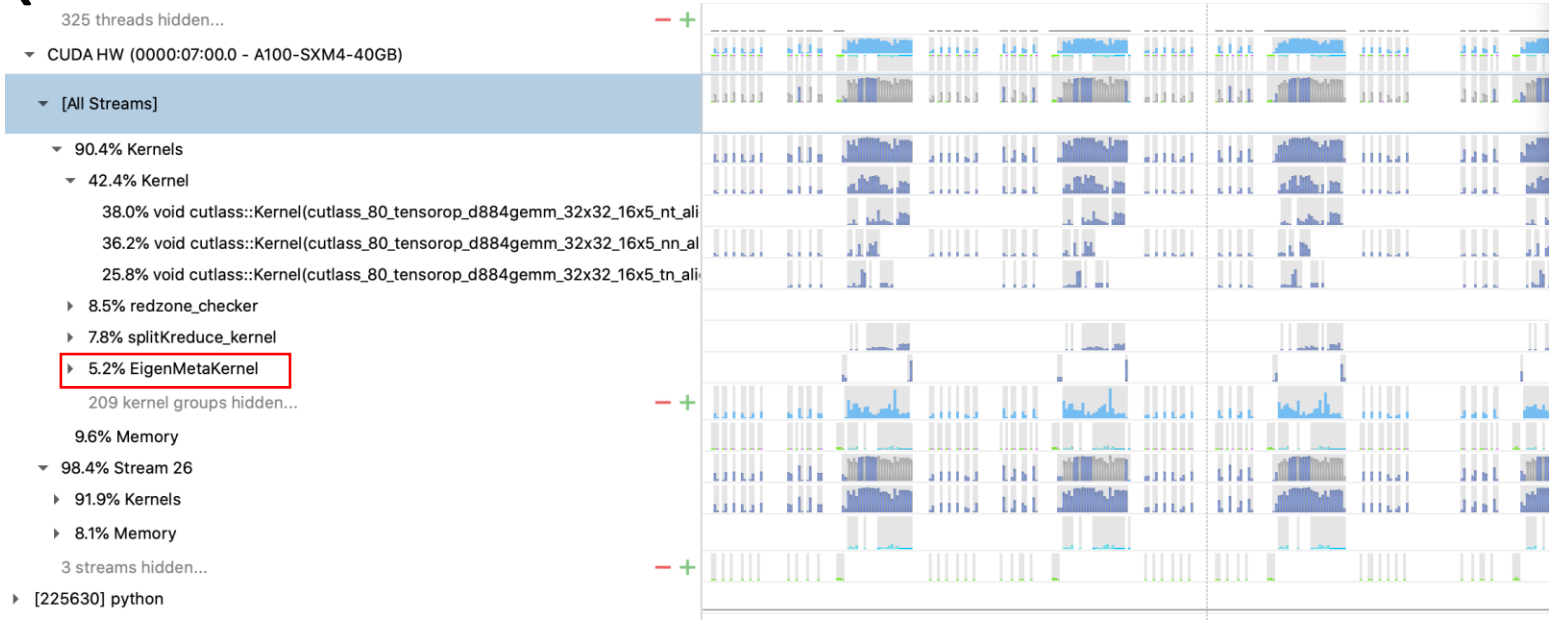
8.1. Tensor Core Math

The TensorFlow container includes support for Tensor Cores starting in Volta's architecture, available on Tesla V100 GPUs. Tensor Cores deliver up to 12x higher peak TFLOPs for training. The container enables Tensor Core math by default; therefore, any models containing convolutions or matrix multiplies using the `tf.float16` data type will automatically take advantage of Tensor Core hardware whenever possible.

In case you want to see detail



2x



Scheduling Job through Container

Demo

```
#!/bin/bash

#SBATCH -J My_Cool_Science
#SBATCH --ntasks=1
#SBATCH --ntasks-per-node=1
#SBATCH --time=1:00:00
#SBATCH --mem=32GB
#SBATCH --partition=3090-gcondo
#SBATCH --gres=gpu:1
#SBATCH -o tf_sing_job_%j.o
#SBATCH -e tf_sing_job_%j.e

# Print key runtime properties for records
echo Master process running on `hostname`
echo Directory is `pwd`
echo Starting execution at `date`
echo Current PATH is $PATH

export SINGULARITY_BINDPATH="/gpfs/scratch,/gpfs/data"

CONTAINER=/gpfs/data/gk/crunch_package/CRUNCH_TALK_SINGULARITY/crunch_tf2.simg
SCRIPT=/gpfs/data/gk/crunch_package/CRUNCH_TALK_SINGULARITY/pinn_laplace_TF2.py


# Run The Job Through Singularity
singularity exec --nv $CONTAINER python $SCRIPT
```

Lets build the Image for PyTorch

Go to NVIDIA NGC Container:

https://catalog.ngc.nvidia.com

On Login node

singularity build crunch_pytorch.simg docker://nvcr.io/nvidia/pytorch:21.12-py3

Get a GPU:

interact –q 3090-gcondo –n 8 –g 1 –m 32g –t 1:00:00

Get A Singularity Shell

singularity shell -B /gpfs/scratch,/gpfs/data --nv crunch_pytorch.simg

Or Export File system as Using Singularity variable:

export SINGULARITY_BINDPATH="/gpfs/scratch,/gpfs/data”

singularity shell–nv crunch_tf2.simg

Thank You!

Happy Computing!