OLCF Tutorial

Deep Learning Workflow with Jupyter

Junqi Yin

Analytics and AI Methods at Scale

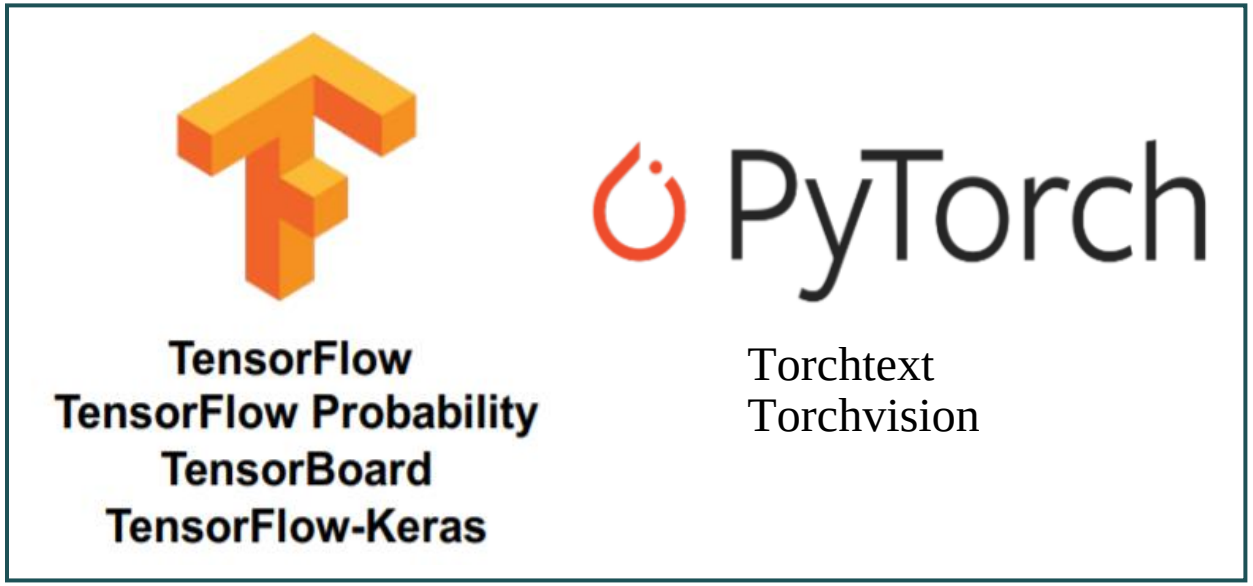
July 14, 2022

Outline

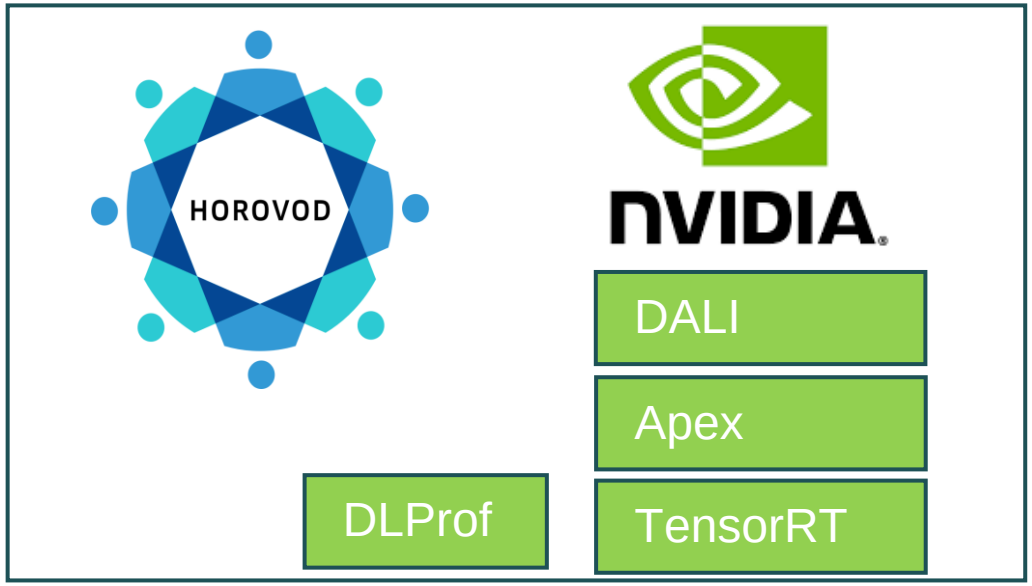
- Summit DL Environment
- Summit DL Workflow + Jupyter
  - Data Exploration
  - Training Visualization
  - Profiling Visualization
  - Model Exploration
- Demo

Deep Learning stack on Summit

- Open-CE (latest: open-ce-olcf/1.5.2-py39-0)
- Ray, DLProf, SmartSim



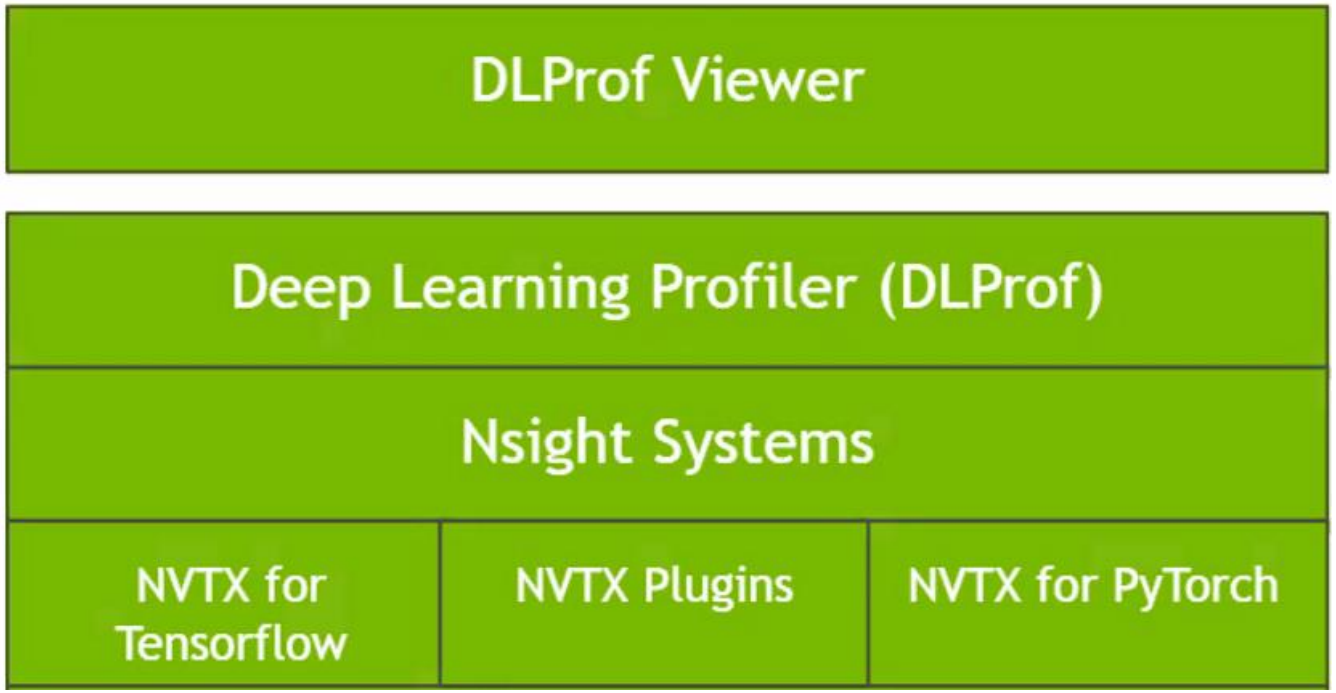
Frameworks

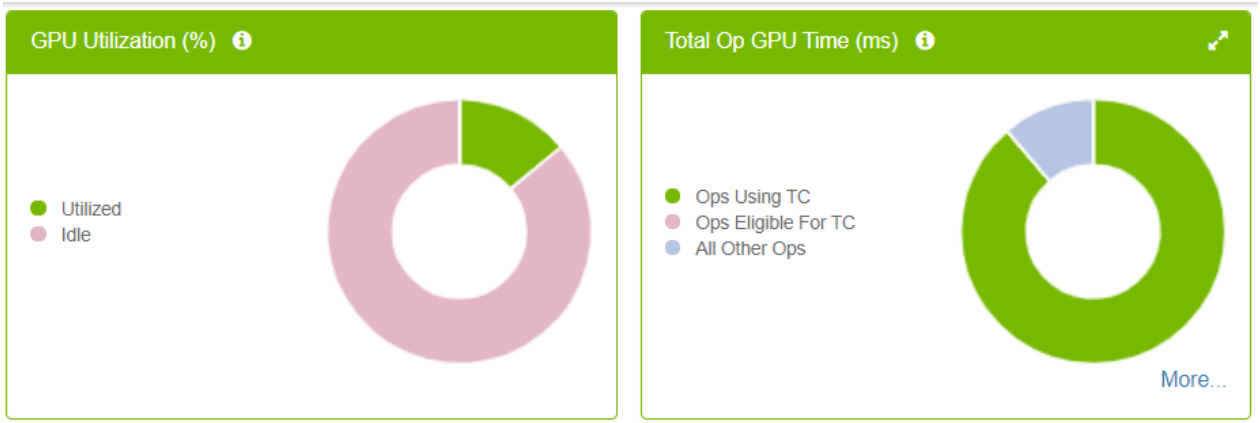


Plugins and Tools

DLProf on Summit

- On top of Nsight
- Ease of use
- Dashboard visualization





DLProf on Summit

- Module (for PyTorch)

```
module module use /sw/aaims/summit/modulefiles
module load dlprof
```

- Instrument your script

```
#Import and initialize PyProf:
import pyprof
pyprof.init(enable_function_stack=True)

#Wrap training loop with PyTorch NVTX context manager:
with torch.autograd.profiler.emit_nvtx():
```

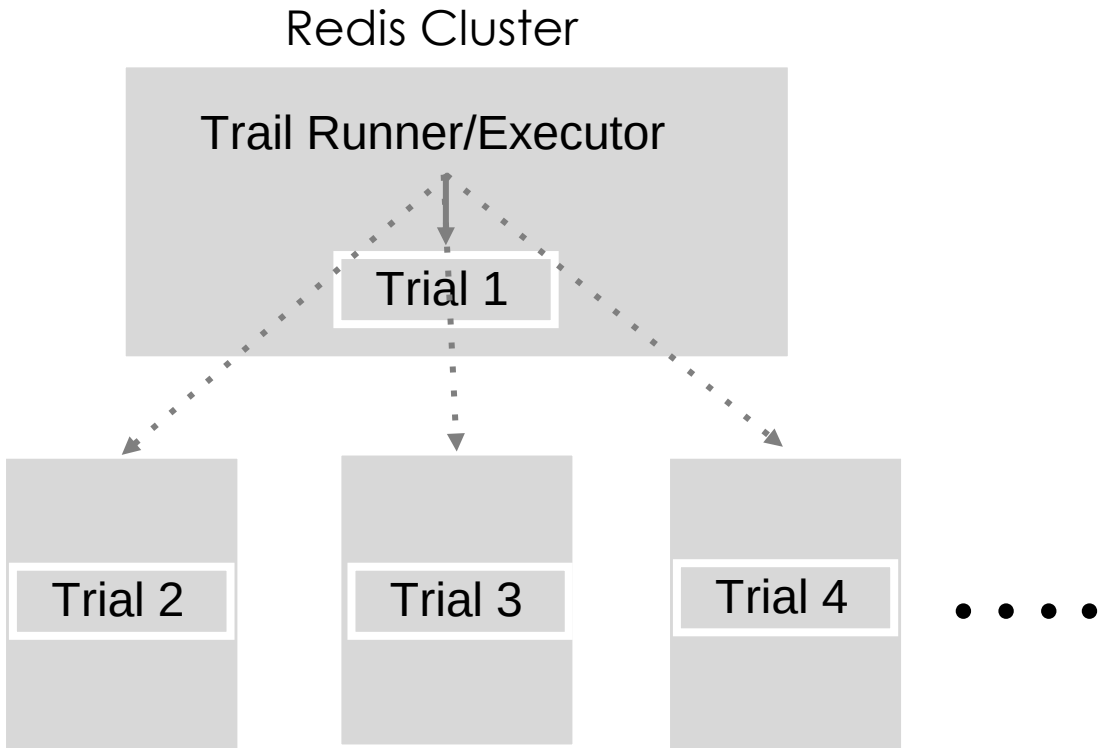
- Launch with dlprof

```
# job script
dlprof --mode="pytorch" python -u main.py ...
```

Ray on Summit

https://ray.readthedocs.io/en/latest/tune.html

- Setup on Summit



- Scripts to start/stop Ray cluster

```
nodes=($(cat ${LSB_DJOB_HOSTFILE} | sort |
uniq | grep -v login | grep -v batch))
head=${nodes[0]}

ssh $head ray start --head --no-ui --
port=6379 --temp-dir=$tmpdir --num-cpus=42 --
num-gpus=6

for worker in ${nodes[@]}; do
  ssh $worker ray start --
address="$head:6379" --temp-dir=$tmpdir --
num-cpus=42 --num-gpus=6 &
  if [ $? -eq 0 ]; then
    echo "Ray worker started on $worker"
  fi
done
wait
```

Ray on Summit

- Using ray.tune.Trainable class

```
class Cifar10Model(Trainable):
  def setup(self, config):
    model = self._build_model(depth=self.config["depth"])
    opt = tf.keras.optimizers.Adam(lr=self.config["lr"],
                                   decay=self.config["decay"])
  def step(self):
    self.model.fit_generator(generator=gen, steps_per_epoch=self.config["batch_size"],
                                            epochs=self.config["epochs"])
```

- Run experiments

```
ray.init(address=args.address)
pbt = PopulationBasedTraining(perturbation_interval=10,...)
run_experiments({"pbt_cifar10": train_spec}, scheduler=pbt)
```

https://code.ornl.gov/olcf-analytics/summit/distributed-deep-learning-examples/tree/master/examples/ray

Ray on Summit

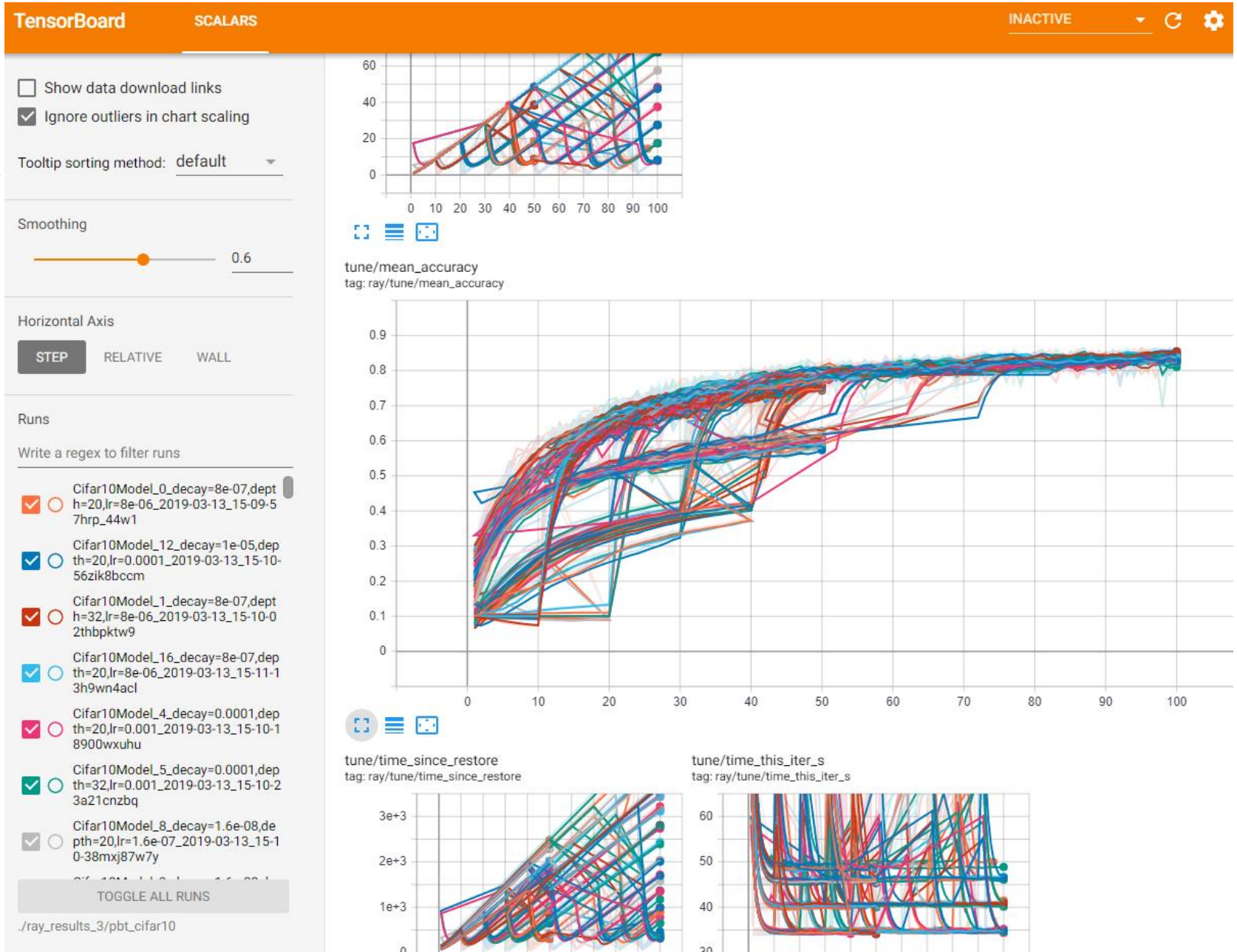
- Run experiments

```
train_spec = {
        "run": Cifar10Model,
        "resources_per_trial": {
            "cpu": 42,
            "gpu": 6
        },
        "stop": {
            "mean_accuracy": 0.90,
            "training_iteration": 50,
        },
        "config": {
            "epochs": 10,
            "batch_size": 64*6,
            "lr": grid_search([10**-3, 10**-4]),
            "decay": sample_from(lambda spec:
                                 spec.config.lr / 10.0),
            "depth": grid_search([20,32,44,50]),
        },
}
```

Main tuning parameters

Ray on Summit

- Vis with TensorBoard
  - TensorFlow
  - PyTorch
- Population based training example



Ray Tune example

- Steps to run
  - git clone https://code.ornl.gov/olcf-analytics/summit/distributed-deep-learning-examples
  - cd distributed-deep-learning-examples/examples/ray
  - Edit tune.lsf: change the project ID
  - bsub tune.lsf
- Visualize on JupyterHub
  - Login to https://jupyter.olcf.ornl.gov/
  - Open distributed-deep-learning-examples/examples/ray/ray-tune-tb.ipynb

DLProf example

- Steps to profile PyTorch ImageNet example
  - git clone --recursive https://github.com/at-aaims/dlprof-examples
  - cd dlprof-examples/DeepLearningExamples
  - git apply ../pytorch/ConvNets.patch
  - cd ../pytorch
  - bsub prof.lsf
- Visualize on Andes via port forwarding
  - ssh to andes
  - module load python
  - source activate /gpfs/alpine/world-shared/stf011/junqi/dlprof-env
  - tensorboard --logdir /gpfs/alpine/world-shared/stf011/junqi/dlprof-env/event_files --host localhost
  - Port forward to local browser: ssh -L 6006:localhost:**6006** andes-login**x**.olcf.ornl.gov
  - Open browser at http://localhost:6006

junqi@login5:/gpfs/alpine/scratch/junqi/stf011/distributed-deep-learning-examples/examples/ray

```
[junqi@login5.summit ray]$ bsub -P stf218 -q debug -W 1:00 -nnodes 16 -alloc_flags nvme -Is /bin/bash
Job <2250731> is submitted to queue <debug>.
<<Waiting for dispatch ...>>
<<Starting on batch4>>
bash-4.4$
```