Introduction to Open-CE on Summit

Junqi Yin
Analytics/AI Methods at Scale (AAIMS)
Oak Ridge Leadership Computing Facility
Outline

• What’s Open-CE?
• How’s Open-CE deployed on Summit?
• Customize your own environment
• Performance baselines
• Documentations
IBM Watson Machine Learning Community Edition (WML-CE)

Multi-node Multi-GPU ML/DL

module load ibm-wml-ce

Machine learning

RAPIDS

- cuDF
- cuML

Dask, CuPy, XGBoost

Deep learning

PyTorch

- TensorFlow
- TensorFlow Probability
- TensorFlow Keras

Torchtext
Torchvision

Frameworks

Plugins and Tools

- IBM LMS
- NVIDIA
- DALI
- Apex
- TensorRT
Open Cognitive Environment (Open-CE)

- IBM WML-CE is deprecated after v1.7.0.
- IBM released most conda receipts for WML-CE as an open-source project – Open-CE
- ibm-wml-ce modules are still available on Summit
- open-ce module provides more up-to-date DL frameworks
Open-CE

Multi-node Multi-GPU ML/DL

Deep learning

Frameworks

Plugins and Tools

Machine learning

py-XGBoost

( Rapids will be deployed as a separated module )
# IBM-WML-CE vs Open-CE

<table>
<thead>
<tr>
<th>Environment</th>
<th>ibm-wml-ce/1.6.1</th>
<th>ibm-wml-ce/1.7.0</th>
<th>open-ce/0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Package</td>
<td>IBM DDL 1.5.0</td>
<td>IBM DDL 1.5.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tensorflow 1.15</td>
<td>Tensorflow 2.1</td>
<td>Tensorflow 2.3</td>
</tr>
<tr>
<td></td>
<td>Pytorch 1.2.0</td>
<td>Pytorch 1.3.1</td>
<td>Pytorch 1.6.0</td>
</tr>
<tr>
<td></td>
<td>Caffe (IBM-enhanced) 1.0.0</td>
<td>Caffe (IBM-enhanced) 1.0.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Horovod v0.18.2 (IBM-DDL Backend)</td>
<td>Horovod v0.19 (NCCL Backend)</td>
<td>Horovod v0.19.5 (NCCL Backend)</td>
</tr>
<tr>
<td>Complete List</td>
<td>1.6.2 Software Packages</td>
<td>1.7.0 Software Packages</td>
<td>Open-CE Software Packages</td>
</tr>
</tbody>
</table>
Open-CE deployment on Summit

• Build opence env

```bash
./open-ce build env envs/opence-env.yaml
   --build_types cuda
   --output_folder condabuild
```

• Build single package

```bash
./open-ce build feedstock
   --working_directory horovod-feedstock
```

• Summit channel

OPENCE_CHANNEL=/sw/sources/open-ce/conda-channel

```
---
imported_envs:
- pytorch-env.yaml
- tensorflow-env.yaml
- xgboost-env.yaml
- dali-env.yaml 
  # [build_type == 'cuda']
- spacy-env.yaml
- transformers-env.yaml
- horovod-env.yaml 
  # [build_type == 'cuda']
```

```
envs/opence-env.yaml:
```

```
horovod-feedstock:
diff --git a/recipe/meta.yaml b/recipe/meta.yaml
   index ffc5140..9b675cf 100644
   --- a/recipe/meta.yaml
   +++ b/recipe/meta.yaml
   @@ -28,7 +28,6 @@ requirements:
   - openmpi {{ openmpi }}
   - nccl {{ nccl }}
   - tensorflow {{ tensorflow }}
   - pytorch {{ pytorch }}
```

```
OPENCE_CHANNEL=/sw/sources/open-ce/conda-channel
```

```
OPENCE_CHANNEL=/sw/sources/open-ce/conda-channel
```

```
OPENCE_CHANNEL=/sw/sources/open-ce/conda-channel
```

```
OPENCE_CHANNEL=/sw/sources/open-ce/conda-channel
```

```
OPENCE_CHANNEL=/sw/sources/open-ce/conda-channel
```

```
OPENCE_CHANNEL=/sw/sources/open-ce/conda-channel
```

```
OPENCE_CHANNEL=/sw/sources/open-ce/conda-channel
```

```
OPENCE_CHANNEL=/sw/sources/open-ce/conda-channel
```

```
OPENCE_CHANNEL=/sw/sources/open-ce/conda-channel
```

```
OPENCE_CHANNEL=/sw/sources/open-ce/conda-channel
```

```
OPENCE_CHANNEL=/sw/sources/open-ce/conda-channel
```
Customize your environment: examples – install Apex

• Install from clone env:

```bash
module load open-ce/0.1-0
module load gcc/7.4.0

# create cloned conda environment
conda create --name my-opence --clone open-ce-0.1-0
conda activate my-opence

# install Apex
git clone https://github.com/NVIDIA/apex
cd apex
pip install -v --no-cache-dir --global-option="--cpp_ext" --global-option="--cuda_ext" ./
```
Customize your environment: examples – install DALI

• Install from scratch:

```bash
module load gcc/7.4.0
export OPENCE_CHANNEL=/sw/sources/open-ce/conda-channel

# install miniconda
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-ppc64le.sh
bash Miniconda3-latest-Linux-ppc64le.sh -b -p ~/.miniconda
export PATH=~/.miniconda/bin:$PATH

# create env for dali
conda create --name dali-env python=3.6
conda activate dali-env

# install dali
conda install -c $OPENCE_CHANNEL dali
```
Performance baselines

• open-ce module includes horovod examples

• To run distributed training on synthetic data, simply submit the job script

```bash
#BSUB -P STF011
#BSUB -W 0:10
#BSUB -nnodes 8
#BSUB -q batch
#BSUB -J mldl_test_job
#BSUB -o logs/pyt%J.out
#BSUB -e logs/pyt%J.err
module load open-ce

# TensorFlow
jsrun -bpacked:7 -g6 -a6 -c42 -r1 python
$CONDA_PREFIX/horovod/examples/tensorflow2_synthetic_benchmark.py

# PyTorch
jsrun -bpacked:7 -g6 -a6 -c42 -r1 python
$CONDA_PREFIX/horovod/examples/pytorch_synthetic_benchmark.py
```
Performance baselines

Open-CE on Summit
Horovod, batch-size=32 per GPU, FP32, synthetic imagenet

Images/s

# of nodes

1 2 4 8 16

0 5000 10000 15000 20000 25000 30000

1788.6 3637.3 7094.8 14063.3 27858 1788.4 3581.3 7035 13887.1 27238.8

TensorFlow v2.3  PyTorch v1.6
Performance baselines: ResNet50 on ImageNet

TF_CNN_Benchmark on Summit: ResNet50
batch-size = 256 per GPU

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Mini-batch size</th>
<th>Top-1 Val accuracy</th>
<th>Training time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>12288</td>
<td>0.750</td>
<td>27</td>
</tr>
<tr>
<td>32</td>
<td>12288</td>
<td>0.766</td>
<td>17</td>
</tr>
<tr>
<td>64</td>
<td>15360</td>
<td>0.763</td>
<td>12</td>
</tr>
</tbody>
</table>

TF.distribute & Horovod + LARS

code: TensorFlow distributed example
Performance baselines: BERT on Wikipedia

- Throughput: 63% (phase1) 83% (phase2)
- Time-to-Solution: ~64min on 1536 V100 vs ~76min on 1024 TPUv3 ([arXiv:1904.00962](https://arxiv.org/abs/1904.00962))

**code:** PyTorch BERT example

Apex DDP + LAMB
Documentations

- Open-CE repo: https://github.com/open-ce/open-ce
- Summit module docs: https://docs.olcf.ornl.gov/software/analytics/ibm-wml-ce.html
- Distributed DL tutorial: https://code.ornl.gov/olcf-analytics/summit/distributed-deep-learning-examples
- Previous trainings:
  - WML-CE on Summit (slides | recording)
  - Scaling up deep learning application on Summit (slides | recording)
  - ML/DL on Summit (slides | recording)