BLUE WATERS

SUSTAINED PETASCALE COMPUTING

Machine Learning With Python: Distributed Training and Data Resources on Blue Waters

Presented by: Aaron D. Saxton
saxton@illinois.edu
Data Scientist, Blue Waters Project Office

NCSA NSF GREAT LAKES CONSORTIUM FOR PETASCALE COMPUTATION CRAY

Todays Topics

- Distributed Training
  - Dive deeper at, https://bluewaters.ncsa.illinois.edu/webinars/data-analytics/ml-tensorflow
- Tensorflow 1.3 “out of the box” usage
- Tensorflow 1.3 with Cray ML plugin
- PyTorch 0.3.0
- Example training data: ImageNet

Distributed Training

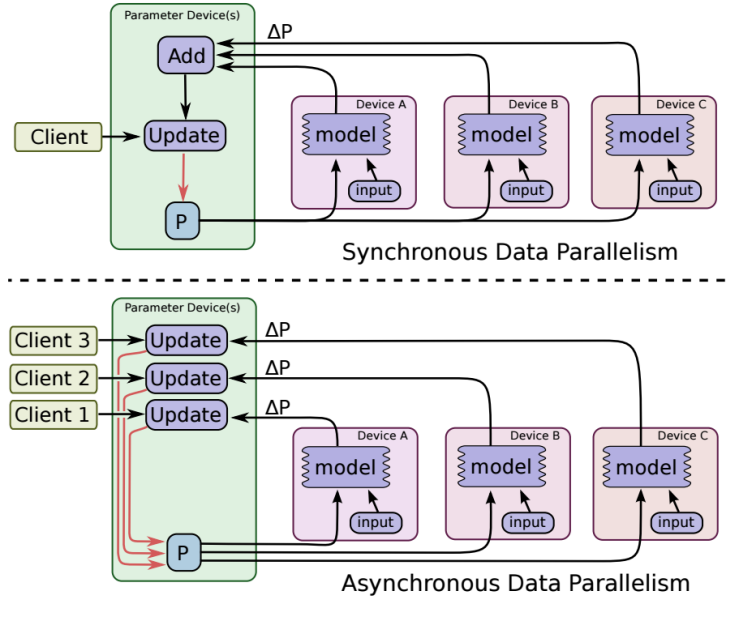
- Training life cycle: Typical SGD
  - Batch of input data (Training Data)
  - Feed forward through model
  - Compute Loss
  - Back propagation: gradients of trainable parameters
  - Advance trainable parameters with gradient
  - Repeat

Distributed Training: Data Parallel

- Share and Aggregate Gradients
- Bucket of Distributed Training Features
  - Parameter Server/Worker
  - Synchronous
  - Asynchronous
  - Per-op
  - All Recued

Distributed Training

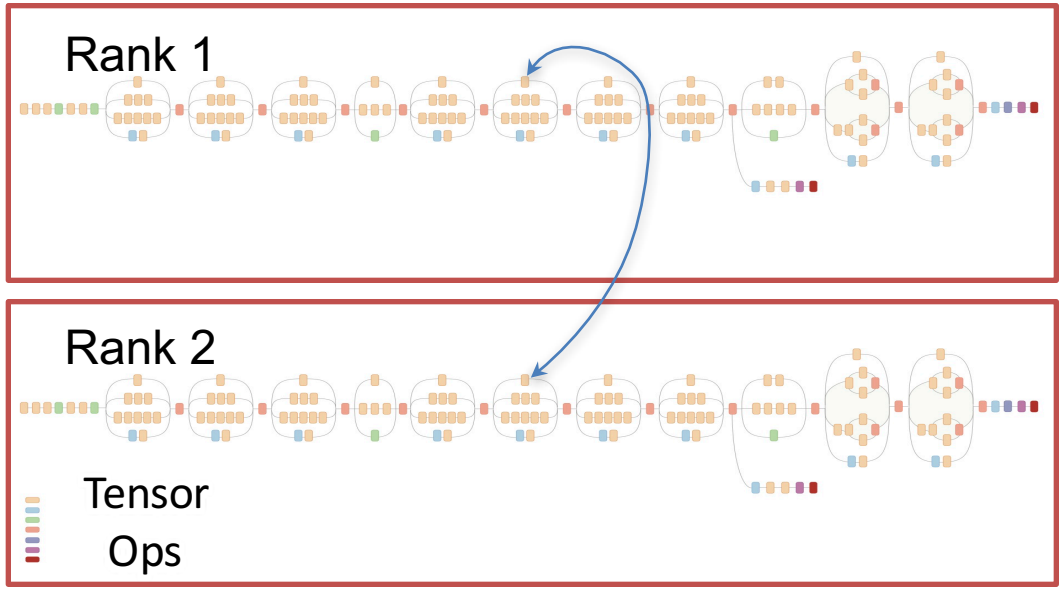
- TensorFlow Parameter Server/Worker



http://ischlag.github.io/2016/06/12/async-distributed-tensorflow/

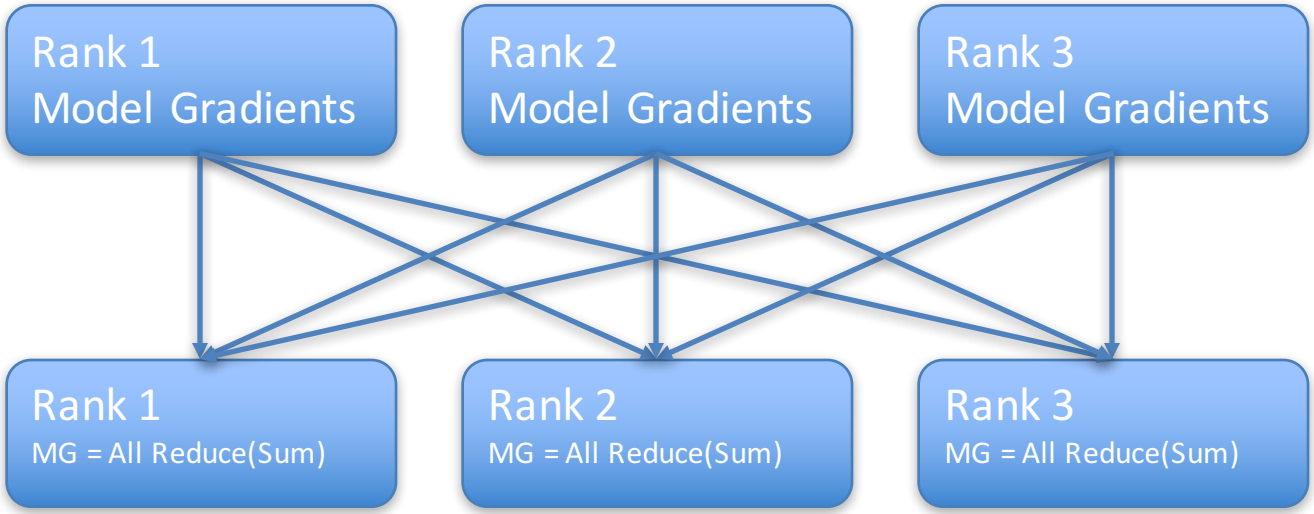
Distributed Training

- Per-OP



Distributed Training

- All Reduce



TensorFlow 1.3 "out of the box"

- https://github.com/asaxton/ncsa-bluewaters-tensorflow

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

asaxton removing GBP work — Latest commit 1591204 2 days ago

BWDistributedTrain	lessons learned from last few months	2 days ago
benchmarks	clening, sterilizing paths	5 months ago
datasets	lessons learned from last few months	2 days ago
models	adding GBP run scripts and supporting material	2 months ago
run_scripts	removing GBP work	2 days ago
.gitignore	refactor inception example	3 months ago
README.md	updating README doc	6 months ago

TensorFlow 1.3 “out of the box”

- Prepare Data

Branch: master ▾ ncsa-bluewaters-tensorflow / datasets / imagenet / Create new file | Upload files | Find file | History

asaxton lessons learned from last few months — Latest commit 2673b53 3 days ago

	..		
	build_imagenet_data.pbs	changing it so that training data is the first default data to processes	4 months ago
2.	build_imagenet_data.py	adjusting bwpy version back to 0.3.0 + tf	4 months ago
1.	extract_data_from_archive.pbs	lessons learned from last few months	3 days ago
	map.sh	adding Done message to log files	5 months ago
	testing_files.txt	adding data extraction and processing tools	5 months ago
	train_files.txt	changing it so that training data is the first default data to processes	4 months ago
	validation_files.txt	adding data extraction and processing tools	5 months ago

TensorFlow 1.3 “out of the box”

- Run Model

Branch: master ▾ ncsa-bluewaters-tensorflow / run_scripts / Create new file | Upload files | Find file | History

asaxton removing GBP work — Latest commit 1591204 2 days ago

..		
README.md	lessons learned from last few months	3 days ago
inception_imagenet_distributed_cray_train.pbs	bug fixes	2 days ago
inception_imagenet_distributed_train.pbs	lessons learned from last few months	3 days ago
inception_imagenet_validate.pbs	bug fixes	2 days ago
restore_graph.pbs	bug fixes	2 days ago

TensorFlow 1.3 “out of the box”

```
#!/bin/bash
#PBS -l nodes=4:ppn=32:xe+16:ppn=16:xk
##PBS -l nodes=10:ppn=16:xk
#PBS -l walltime=10:00:00
#PBS -N inception_imagenet_distributed_train
#PBS -e logs/log.${PBS_JOBNAME}_NN_${PBS_NUM_NODES}_${PBS_JOBID}.err
#PBS -o logs/log.${PBS_JOBNAME}_NN_${PBS_NUM_NODES}_${PBS_JOBID}.out

echo "Starting"
cd $PBS_O_WORKDIR
mkdir -p logs

NUM_GPU=$(aprun -n ${PBS_NUM_NODES} -N 1 -- /sbin/lspci | grep NVIDIA | wc -l)
let NUM_PS=${PBS_NUM_NODES}-${NUM_GPU}
NUM_WORKER=${NUM_GPU}
if [ "${NUM_PS}" -eq '0' ];
then
echo "all nodes have a GPU, giving some of them to the PS"
NUM_PS=$((${NUM_GPU}/4))
NUM_WORKER=$((${PBS_NUM_NODES}-${NUM_PS}))
fi

echo "NUM_PS ${NUM_PS}, NUM_WORKER ${NUM_WORKER}"
```

TensorFlow 1.3 “out of the box”

```
module load bwpy
module load bwpy-mpi


MBS=32
NUM_STEPS=$(echo "${NUM_TRAINING_EXAMPLES}*${NUM_EPOCHS} / ( ${MBS} )" | bc )
# NUM_STEPS=38545500 # 3200 epochs at batchsize 1
# NUM_STEPS=1204500 # 100 epochs at batch size 32
NUM_STEPS=300 # 0 epochs for testing
DATA_DIR="${HOME}/scratch/ImageNet/tf_records"

#UNIQUE_CHECKPOINT_NAME="_$(cat /dev/urandom | tr -dc 'A-Z0-9' | fold -w 3 | head -n 1)"

LEARNING_RATE=$(echo "0.4 * sqrt(${NUM_WORKER})" | bc)
echo "Learning Rate: ${LEARNING_RATE}"

APOUT_LOGS="${PBS_O_WORKDIR}/logs/apout.${PBS_JOBNAME}_NW_${NUM_WORKER}_MBS_${MBS}_${PBS_JOBID}"
CHECKPT_DIR="checkpoint_dir_${PBS_JOBNAME}_${PBS_JOBID}"


echo "output at ${APOUT_LOGS}.*"
echo "checkpoint_dir at ${CHECKPT_DIR}"
```

TensorFlow 1.3 “out of the box”

```
RUN_CMD="python ${PBS_O_WORKDIR}/../BWDistributedTrain/inception_imagenet_distributed_train.py \
--data_dir $DATA_DIR/train \
--num_steps $NUM_STEPS \
--num_train_examples 385455 \
--batch_size ${MBS} \
--initial_learning_rate ${LEARNING_RATE} \
--checkpoint_dir ${CHECKPT_DIR}"
echo "Running Comand"
echo ${RUN_CMD}
aprun -b -cc none -n ${NUM_PS} -N 1 $RUN_CMD --ps_worker ps : -n ${NUM_WORKER} -N 1 $RUN_CMD --ps_worker worker \
1> ${APOUT_LOGS}.out \
2> ${APOUT_LOGS}.err
```

TensorFlow 1.3 “out of the box”

Branch: master ▾ ncsa-bluewaters-tensorflow / BWDistributedTrain / Create new file | Upload files | Find file | History

asaxton lessons learned from last few months — Latest commit 2673b53 3 days ago

..		
alexnet_imagenet_distributed_train.py	lessons learned from last few months	3 days ago
inception_imagenet_distributed_train.py	lessons learned from last few months	3 days ago
inception_imagenet_distributed_train_cray.py	lessons learned from last few months	3 days ago
inception_imagenet_horovod_distributed_tr...	adding GBP run scripts and supporting material	2 months ago
inception_imagenet_validate.py	lessons learned from last few months	3 days ago
main.py	refactor inception example	3 months ago
resnet50_imagenet_distributed_train.py	lessons learned from last few months	3 days ago
restore_graph.py	lessons learned from last few months	3 days ago
simple_regression.py	lessons learned from last few months	3 days ago

TensorFlow 1.3 “out of the box”

- Set up TensorFlow PS/WORKER
- Begin Data Load Queue
- Define Optimizer
- Define Model Inference
- Define Loss
- Add compute gradients to graph
- Begin Training Loop

- Code Tour!

TensorFlow 1.3 Cray ML Plugin

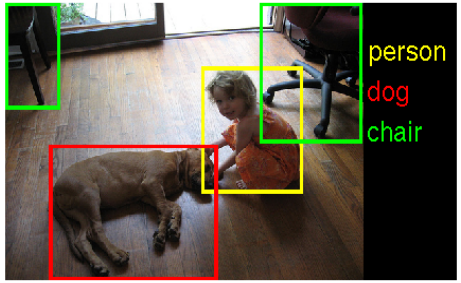
- Set up TensorFlow PS/WORKER
- Begin Data Load Queue
- Define Optimizer
- Define Model Inference
- Define Loss
- Add compute gradients to graph
- Add message passing hooks
- Begin Training Loop

- Code Tour!

PyTorch 0.3.0 usage

- Begin Data Load Queue
- Define Optimizer
- Define Model Inference
- Define Loss
- Loop
  - Compute Loss
  - Compute gradients
  - Share and aggregate gradients
  - Update model
- https://github.com/asaxton/ncsa-bluewaters-pytorch
  - Not ready for release
- Code Tour!

ImageNet



- www.image-net.org
- Large High Quality Dataset
  - 14,197,122 Images
  - 21841 synsets
- Runs the Large Scale Visual Recognition Challenge (ILSVRC)
- Annotated
  - Bounding Boxes
  - synset
    - WordNet (http://wordnet.princeton.edu)

ImageNet

- Blue Waters hosts copy of ImageNet
- Legal Term of Access
  - Create account on www.image-net.org
  - Navigate to Term of Access
  - Accept Term of Access
  - Take screen shot or print to PDF Term of Access with your name on it.
  - Email to saxton@illinois.edu
- After I receive your Term of Access I will give your Blue Waters user read permission to data

Term Of Access

You have been granted access for **non-commercial research/educational** use. By accessing the data, you have agreed to the following terms. Note: Our terms of access have changed. By continuing to download and/or access ImageNet data you agree to the new terms of access.

Aaron Saxton (the "Researcher") has requested permission to use **the ImageNet database** (the "Database") at **Princeton University and Stanford University.** In exchange for such permission, Researcher hereby agrees to the following terms and conditions:

1. Researcher shall use the Database only for non-commercial research and educational purposes.
2. Princeton University and Stanford University make no representations or warranties regarding the Database, including but not limited to warranties of non-infringement or fitness for a particular purpose.
3. Researcher accepts full responsibility for his or her use of the Database and shall defend and indemnify the ImageNet team, Princeton University, and Stanford University, including their employees, Trustees, officers and agents, against any and all claims arising from Researcher's use of the Database, including but not limited to Researcher's use of any copies of copyrighted images that he or she may create from the Database.
4. Researcher may provide research associates and colleagues with access to the Database provided that they first agree to be bound by these terms and conditions.
5. Princeton University and Stanford University reserve the right to terminate Researcher's access to the Database at any time.
6. If Researcher is employed by a for-profit, commercial entity, Researcher's employer shall also be bound by these terms and conditions, and Researcher hereby represents that he or she is fully authorized to enter into this agreement on behalf of such employer.
7. The law of the State of New Jersey shall apply to all disputes under this agreement.

TensorFlow 1.3 “out of the box”

- Prepare Data

Branch: master ▾ ncsa-bluewaters-tensorflow / datasets / imagenet / Create new file | Upload files | Find file | History

asaxton lessons learned from last few months — Latest commit 2673b53 3 days ago

	..		
	build_imagenet_data.pbs	changing it so that training data is the first default data to processes	4 months ago
2.	build_imagenet_data.py	adjusting bwpy version back to 0.3.0 + tf	4 months ago
1.	extract_data_from_archive.pbs	lessons learned from last few months	3 days ago
	map.sh	adding Done message to log files	5 months ago
	testing_files.txt	adding data extraction and processing tools	5 months ago
	train_files.txt	changing it so that training data is the first default data to processes	4 months ago
	validation_files.txt	adding data extraction and processing tools	5 months ago

Thank You!

Questions