TensorFlow: A System for Large Scale Machine Learning

Link to paper

Ajay Shridhar Joshi
cs15b047@smail.iitm.ac.in

SysDL Recitation
CS 6886

OVERVIEW

- Machine learning system designed to operate at large scale and in a variety of environments
- Highly flexible system which can work with a variety of hardware and gives the flexibility to be used in real-world deployments as well as in research
- Focused on training and inference of Deep Neural Networks

DESIGN IDEAS

- Dataflow graphs are used, which consist of operations and tensors
- Dividing the application into 2 phases:
  - Dataflow graph creation
  - Optimized execution
- Common abstraction for devices enables the use of heterogeneous devices like CPU, GPU, TPU, etc.

EXECUTION MODEL

- Dataflow Graphs
- Partial and concurrent execution
- Distributed execution
- Dynamic Control Flow

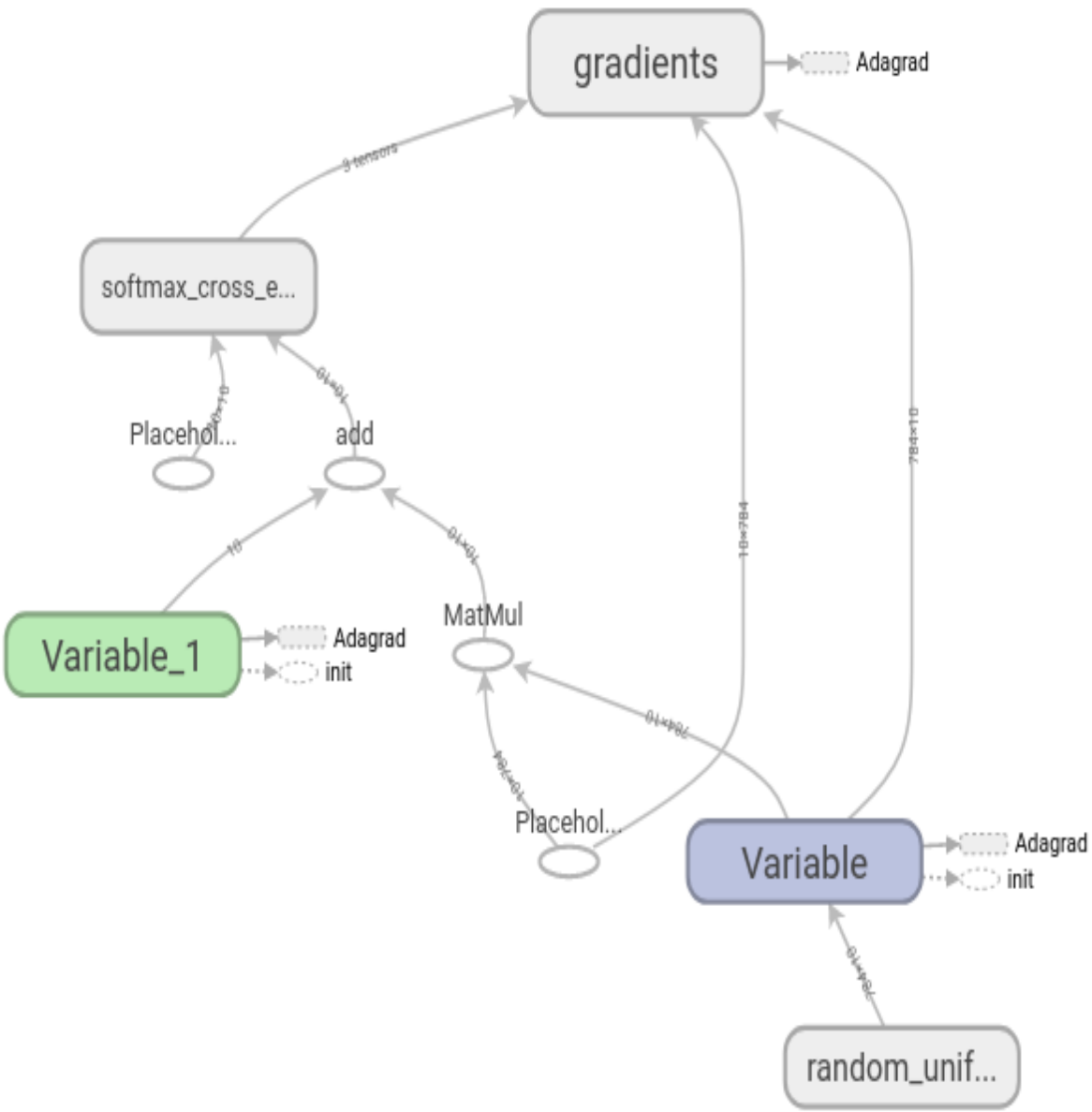
DATAFLOW GRAPHS

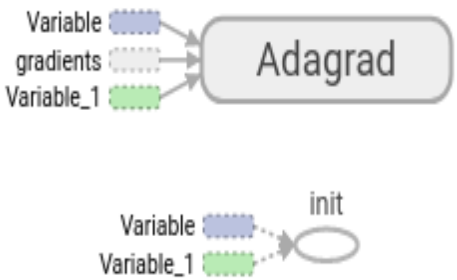
- Operations are represented as nodes and tensors flow along edges
- Operations eg: Add, MatMul, Variable, etc
- Operations can have "mutable state", eg: Variable

SAMPLE CODE

```
import tensorflow as tf
import numpy as np
BATCH_SIZE = 10
NUM_STEPS = 10000
########### Placeholder for inputs (images and labels)#############
x = tf.placeholder(tf.float32, [BATCH_SIZE, 784])
y = tf.placeholder(tf.float32, [BATCH_SIZE, 10])
#################################################################
#########Variables for parameters with initialization#########
W_1 = tf.Variable(tf.random_uniform([784, 10]))
b_1 = tf.Variable(tf.zeros([10]))
#################################################################
###############NN Layer####################
layer_1 = tf.matmul(x, W_1) + b_1
#########################################
########Loss and optimizer operations#####################
loss = tf.nn.softmax_cross_entropy_with_logits(logits = layer_1, labels = y)
train_op = tf.train.AdagradOptimizer(0.01).minimize(loss)
#########################################################
######Execute the graph###########
with tf.Session() as sess:      #Connect to the TF runtime.
  sess.run(tf.global_variables_initializer())   #Randomly initialize weights.
  for step in range(NUM_STEPS):
    x_data,y_data= np.random.normal(size=(BATCH_SIZE, 784)),np.random.normal(size=(BATCH_SIZE, 10))
    # Perform one training step
    sess.run(train_op, {x: x_data, y: y_data})
```

SAMPLE DATAFLOW GRAPH

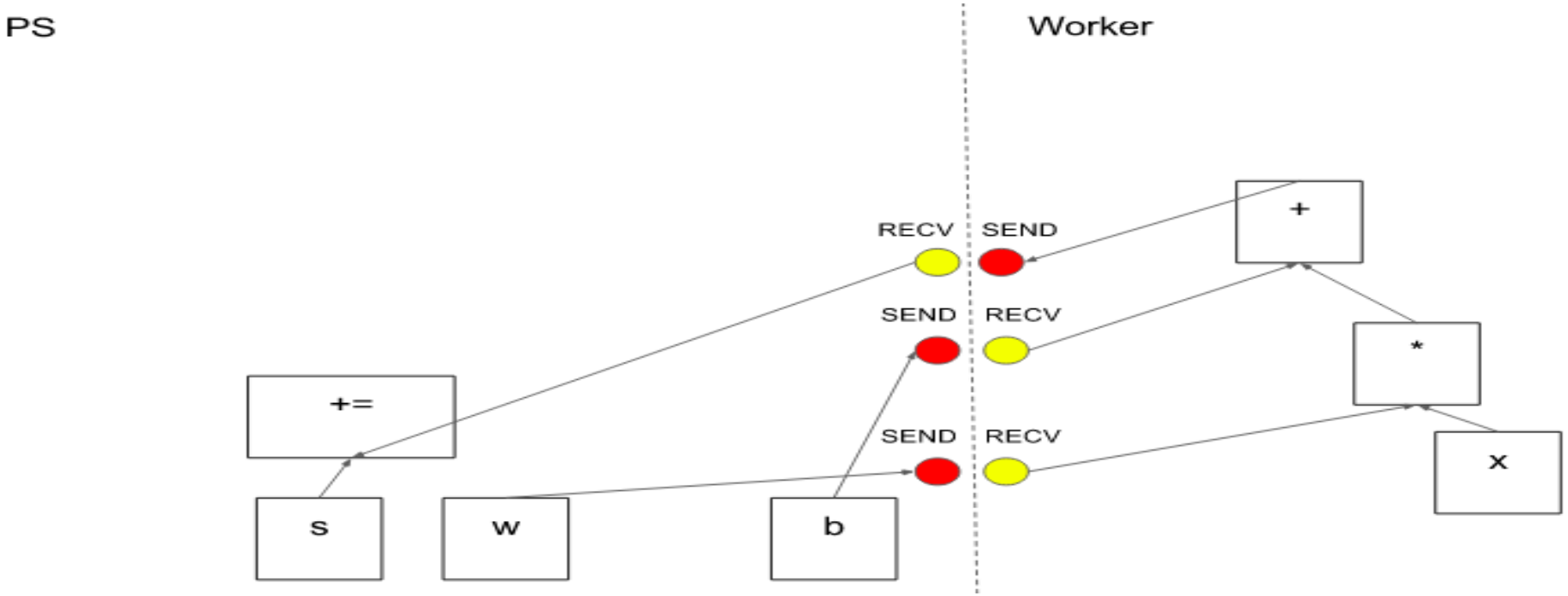




PARTIAL AND CONCURRENT EXECUTION

- Dataflow graph ordered and pruned according to executed operations
- Concurrent execution of subgraphs
- Data-parallel training is enabled by concurrent preprocessing of multiple batches and shared mutable states(eg: parameters)

DISTRIBUTED EXECUTION

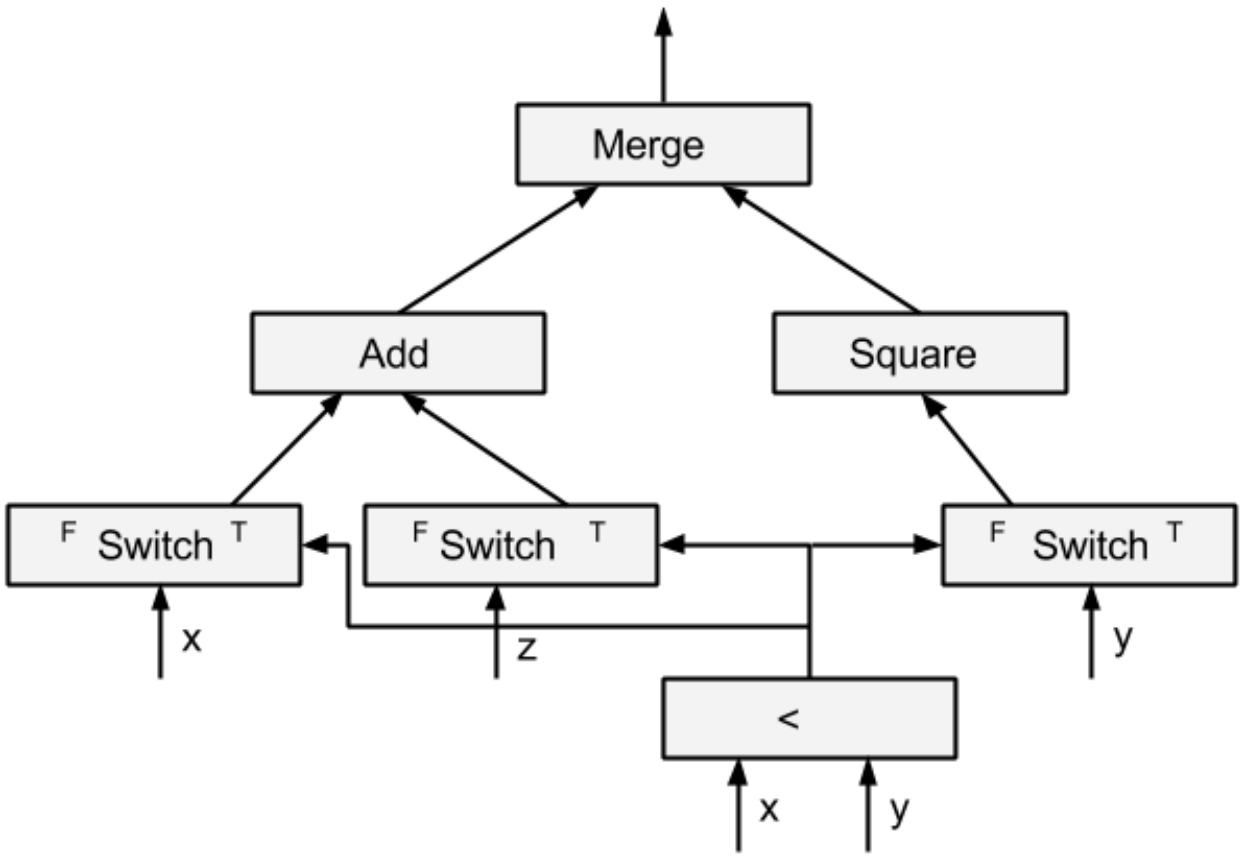


Distributing s += wx + b across 2 tasks

(PS: Task which acts as the parameter server)

- Every operation is part of a task and resides on a particular device
- TF partitions operations into subgraphs and places them on devices subject to constraints in the graph
- Data sent across devices using Send and Recv operations

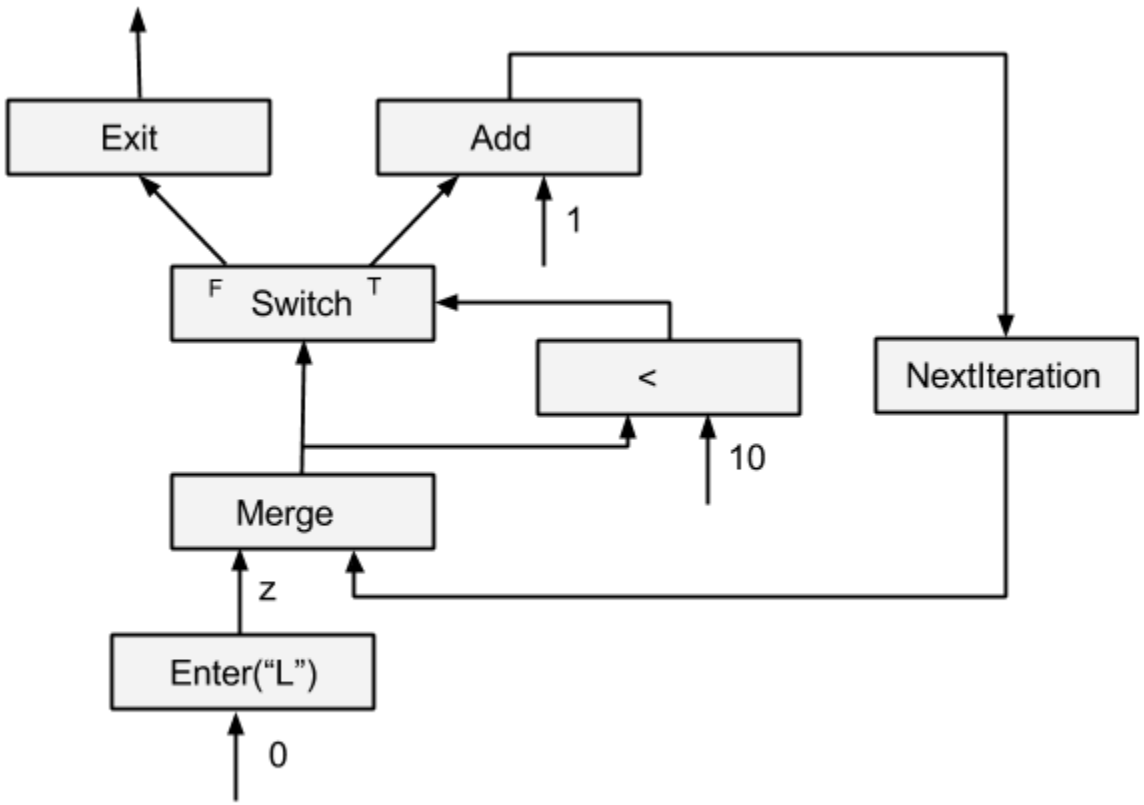
DYNAMIC CONTROL FLOW




```
tf.cond(x < y, lambda: tf.add(x, z), lambda: tf.square(y))
```

Dataflow graph for handling condition

DYNAMIC CONTROL FLOW

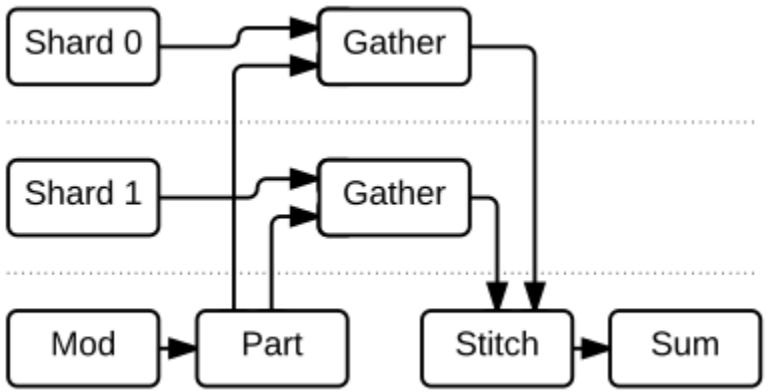



```
tf.while_loop(lambda i: i < 10, lambda i: tf.add(i, 1), [0])
```

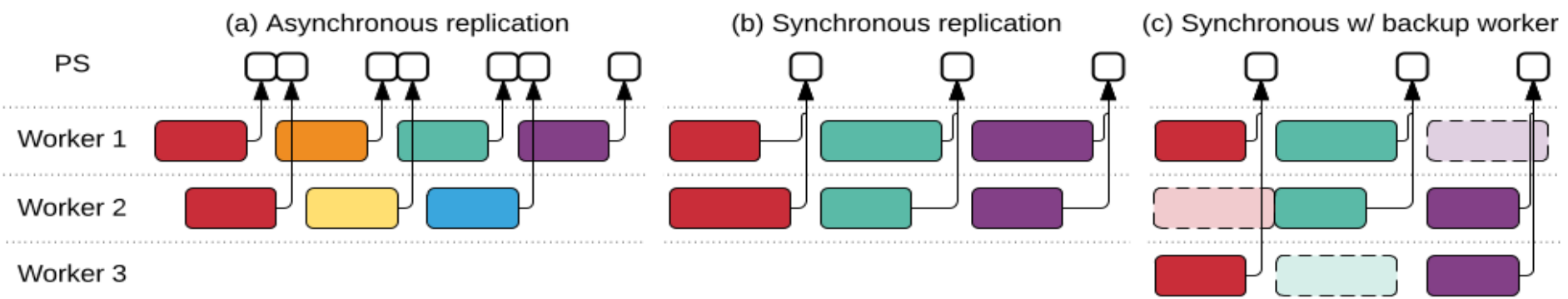
Dataflow graph for a while loop

EXTENSIBILITY

- Differentiation and Optimization
  - Custom operations
  - Custom optimizers
- Training large models
- Fault tolerance
- Synchronization



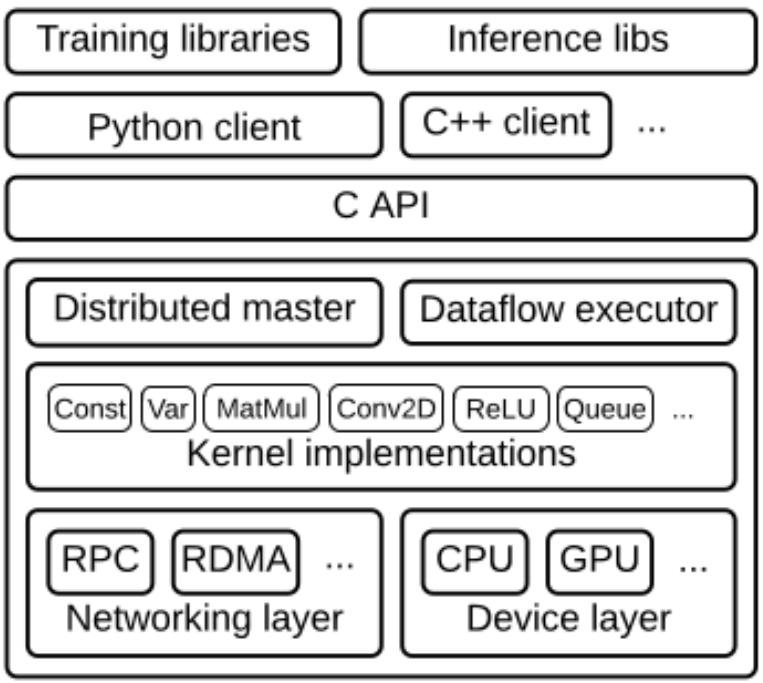
2-way sharded embedding matrix



3 sychronization schemes for parallel optimization

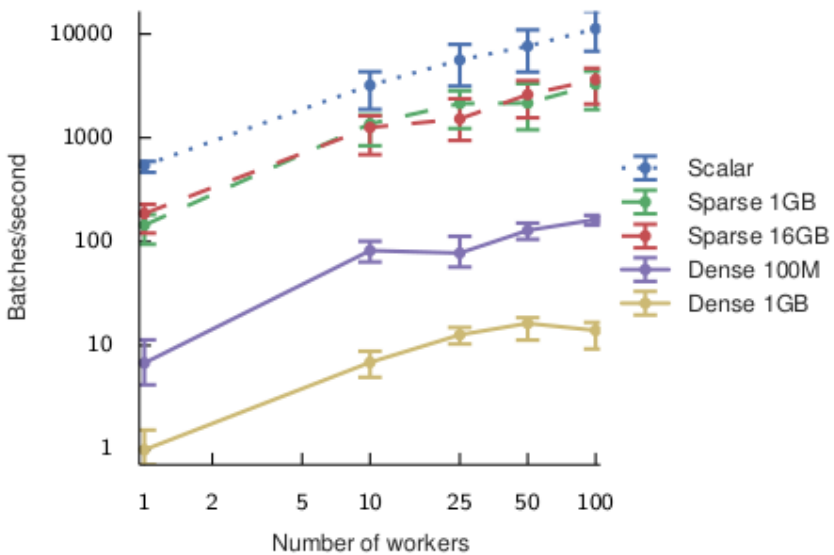
ARCHITECTURE

- The user interacts with TF using the Python or C++ client
- Core runtime implemented in C++ for portability and speed
- Distributed Master prunes and partitions the graph into tasks and assigns it to devices
- Dataflow executor executes operations in a subgraph



PERFORMANCE

- Single-machine benchmarks
- Synchronous-replica microbenchmark
- Image classification
- Language modeling



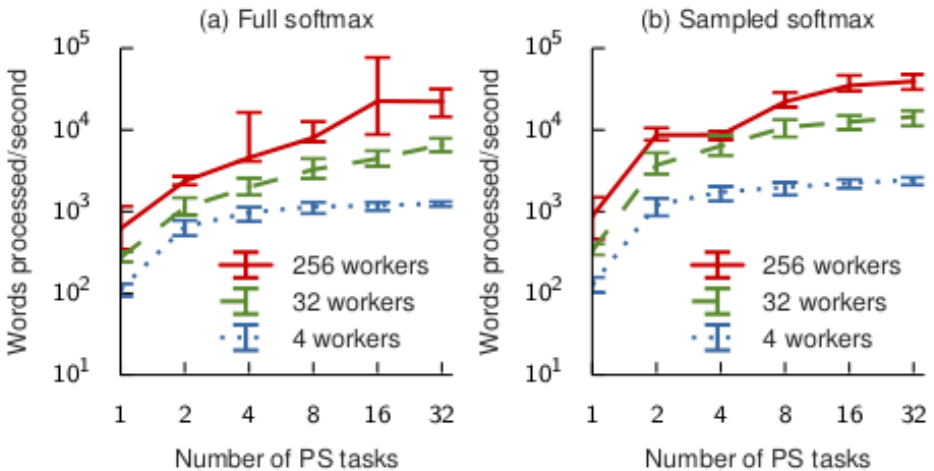
Synchronous replication results with null model

Library	Training step time (ms)			
	AlexNet	Overfeat	OxfordNet	GoogleNet
Caffe [38]	324	823	1068	1935
Neon [58]	87	**211**	**320**	**270**
Torch [17]	**81**	268	529	470
TensorFlow	**81**	279	540	445

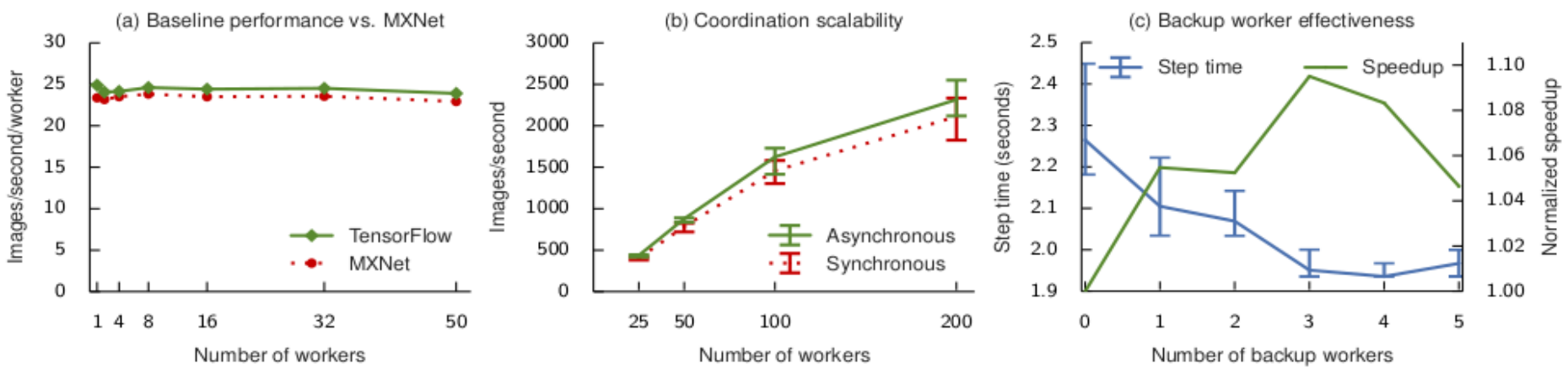
Single-machine benchmarks

PERFORMANCE

- Image classification
- Language modeling



Language model training



Evaluation for Inception-v3 training

LIMITATIONS

- Applications having different computation graph for different sized inputs are difficult to support in TF
- Eg: DRL algorithms, parse trees

Even this shall pass away.

-Theodore Tilton

Ajay Shridhar Joshi
cs15b047@smail.iitm.ac.in

SysDL Recitation
CS 6886