Which Is Deeper Comparison of Deep Learning Frameworks Atop Spark

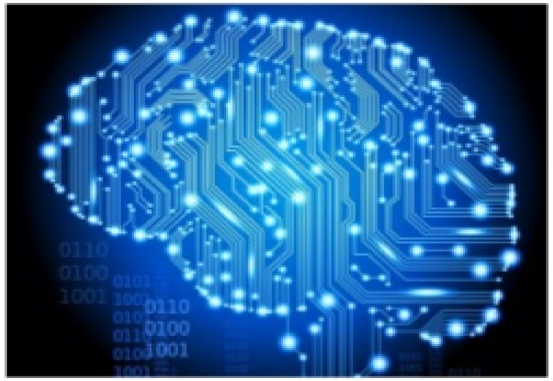
Zhe Dong, Dr. Yu Cao
EMC Corporation

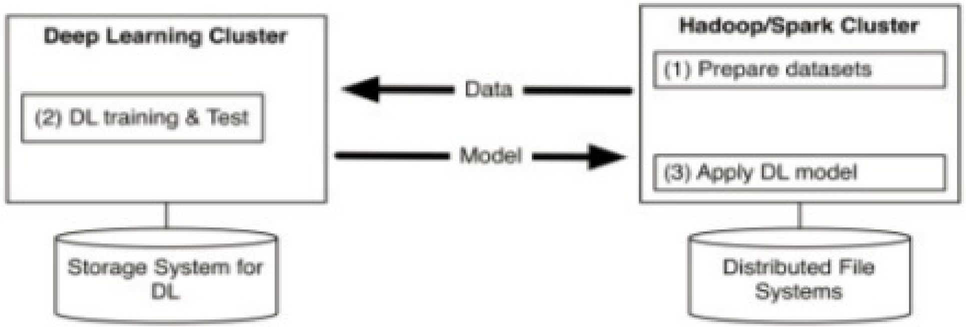
SPARK SUMMIT 2016
DATA SCIENCE AND ENGINEERING AT SCALE
JUNE 6-8, 2016 SAN FRANCISCO

Outline

- Motivation
- Theoretical Principle
- State-of-the-Art
- Evaluation Criteria
- Evaluation Results
- Summary
- Conclusion

Deep Learning on Spark Motivation

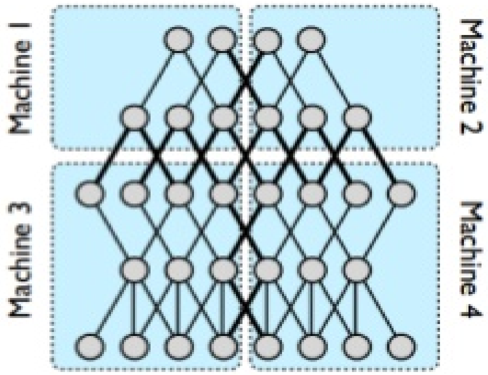


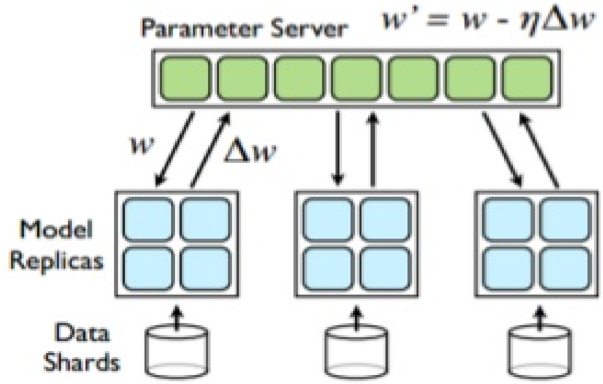


- Dedicated deep learning cluster
  - Massive data movement
  - High maintenance cost
- Spark+Deep Learning = Truly All-in-One

Theoretical Principle

- Large Scale Distributed Deep Networks,Jeffrey Dean,2012
  - Model parallelism
  - Data parallelism

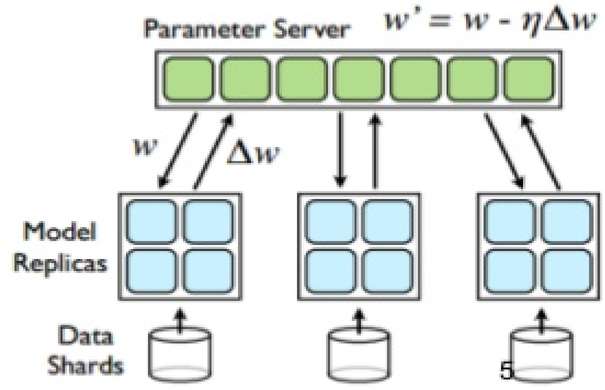




https://papers.nips.cc/paper/4687-large-scale-distributed-deep-networks.pdf

Data Parallelism for distributed SGD

- Model is replicated on worker nodes
- Two repeating steps
  - Train each model replica with mini-batches
  - Synchronize model parameters across cluster
- Specific implementations can be different
  - How parameters are combined
  - Synchronization (strong or weak)
  - Parameter server (centralized or not)



DownpourSGD Client Pseudo code

Algorithm 7.1: DOWNPOURSGDCLIENT($\alpha, n_{fetch}, n_{push}$)

```
procedure StartAsynchronouslyFetchingParameters(parameters)
  parameters ← GetParametersFromParamServer()

procedure StartAsynchronouslyPushingGradients(accruedgradients)
  SendGradientsToParamServer(accruedgradients)
  accruedgradients ← 0

main
  global parameters, accruedgradients
  step ← 0
  accruedgradients ← 0
  while true
       ⎧ if (step mod n_fetch) == 0
       ⎪   then StartAsynchronouslyFetchingParameters(parameters)
       ⎪ data ← GetNextMinibatch()
       ⎪ gradient ← ComputeGradient(parameters, data)
    do ⎨ accruedgradients ← accruedgradients + gradient
       ⎪ parameters ← parameters − α * gradient
       ⎪ if (step mod n_push) == 0
       ⎪   then StartAsynchronouslyPushingGradients(accruedgradients)
       ⎩ step ← step + 1
```

http://www.cs.toronto.edu/~ranzato/publications/DistBeliefNIPS2012_withAppendix.pdf

DL on Spark – State-of-the-Art

- AMPLab SparkNet
- Yahoo! CaffeOnSpark
- Arimo Tensorflow On Spark
- Skymind DeepLearning4J
- DeepDist
- H2O Spark

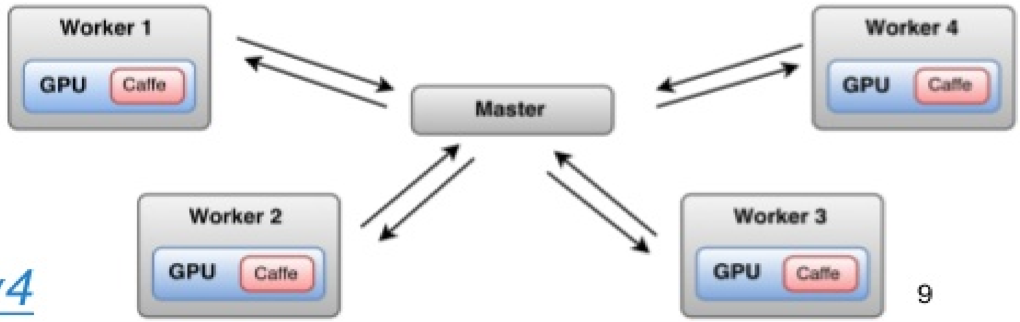
Which is deeper?

Evaluation Criteria

Evaluation Criteria	Dimensions	For Example
Ease of Getting Started	Documentation	Are there detailed, well-organized, up-to-date documents?
	Installation	How automatic it is?
	Built-in Examples	Examples available for quick warming up?
Ease of Use	Interface	Programming language support
	Model Encapsulation	Model/Layer/Node
Functionality	Built-in Models	Which NN models have been implemented?
	Parallelism	Model parallelism or data parallelism
Performance	Performance	MNIST benchmark results
Status Quo	Community Vitality	Github project statistics
	Enterprise Support	Contributions from organizations?

SparkNet

- Started by AMPLab from 2015
- Wrapper of Caffe and Tensorflow
- Centralized parameter server
- Strong SGD synchronization
- Differentiating feature: A fixed number (τ) of iterations (mini-batch) on its subset of data



http://arxiv.org/pdf/1511.06051v4

Evaluation Criteria	Dimensions	SparkNet	Score
Ease of Getting St	Documentation	Paper: No Blog: README.md in Github	

```
val netParams = NetParams(
  RDDLayer("data", shape=List(batchsize, 1, 28, 28)),
```

```
// initialize nets on workers
workers.foreach(_ => {
```

	1	2	3	4
Iterations	1000	2000	5000	10000
Time (seconds)	2130	4218	10471	21003
Accuracy	94.13%	94.26%	94.01%	94.22%

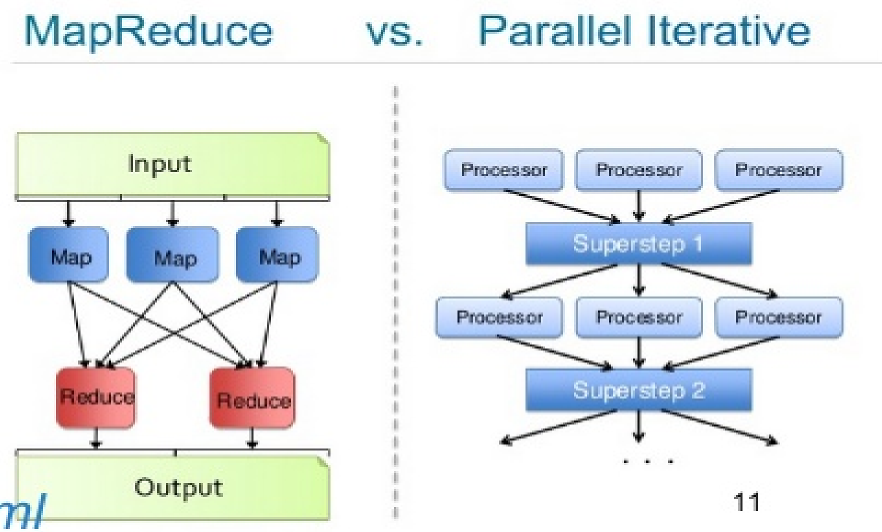
```
  SoftmaxWithLoss("loss", List("ip2", "label"))
)
```

Listing 2: Example network specification in SparkNet

Support

Deeplearning4J

- Started by Skymind from 2014
- An open-source, distributed deep-learning project in Java and Scala
- Parameter server: IterativeReduce
- Strong SGD synchronization



http://deeplearning4j.org/iterativereduce.html

Evaluation Criteria	Dimensions	DL4J	Score
Ease of Getting Started	Docu		
	Instal		
	Built-i		
Ease of Use	Interf		
	Mode Enca		
Functior			
Perform			
Status C	Vitali		
	Enter Support		

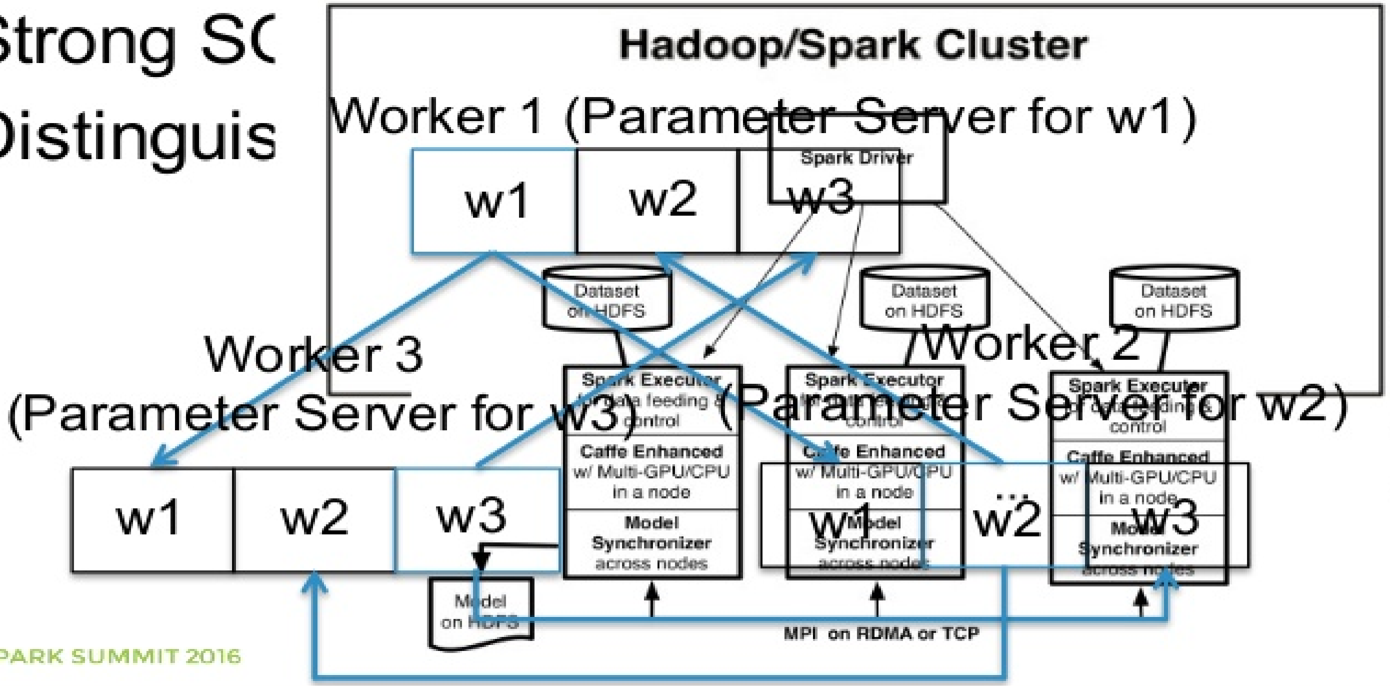
```
MultiLayerConfiguration conf = new NeuralNetConfiguration.Builder()
     .seed(12345)
     .iterations(1)
     .weightInit(WeightInit.XAVIER)
     .updater(Updater.ADAGRAD)
     .activation("relu")
     .optimizationAlgo(OptimizationAlgorithm.STOCHASTIC_GRADIENT_DESCENT)
     .learningRate(0.05)
     .regularization(true).l2(0.0001)
     .list()
```

```
     .pretrain(false).backprop(true)
     .build();
```

	1	2	3	4
Epochs	5	10	15	20
Time (seconds)	2098	4205	6303	8367
Accuracy	70%	79%	82.7%	84.6%

CaffeOnSpark

- Started by Yahoo! from 2015
- Peer-to-Peer parameter server
- Strong SC
- Distinguis nfiniband



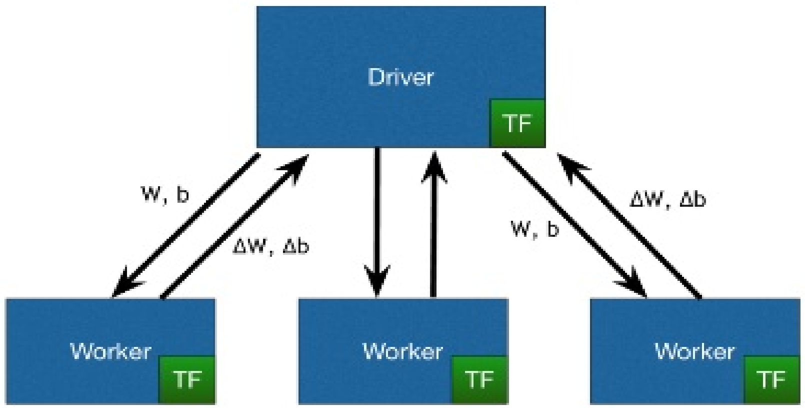
Weights propagation (Gradients are sent in reverse direction)

Evaluation Criteria	Dimensions	CaffeOnSpark	Score
Ease of Getting Started	Documentation	Blog; README.md in github	
	Installation	Have to install all Caffe needed in each node	
	Built-in Examples	Cifar10/MNIST	
Ease of Use	Interface	Java/Scala, DataFrames	
Functionali			
Performan			
Status Quo	Community Vitality	Watch 105 Star 626 Fork 157 66 commits 4 contributors	
	Enterprise Support	Yahoo!	

	1	2	3	4
Iterations	1000	2000	5000	10000
Time(seconds)	224	445	1113	2229
Accuracy	97%	99.4%	99.7%	99.6%

Tensorflow on Spark

- Started by Arimo from 2014
- A data-parallel Downpour SGD implementation on Spark
- Centralized parameter server
- Weak SGD synchronization

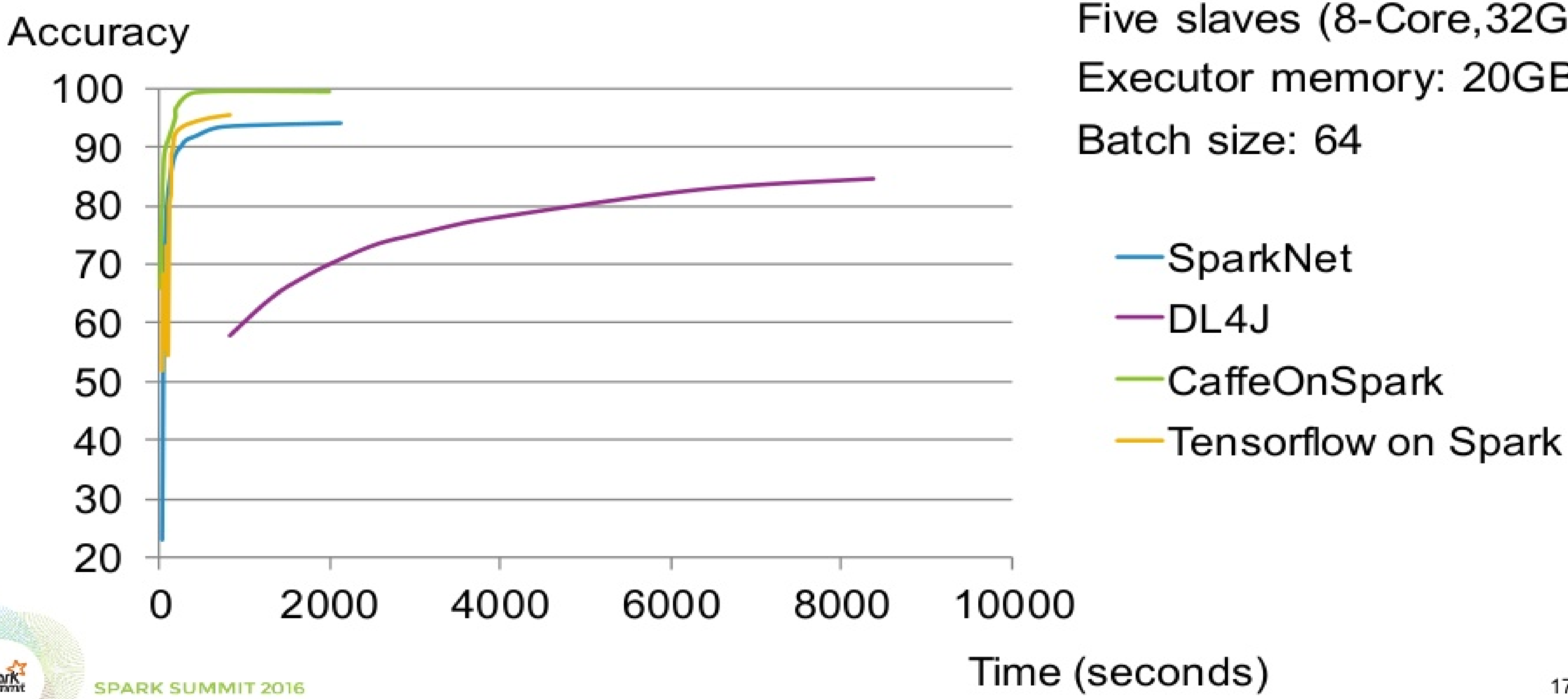


Evaluation Criteria	Dimensions	Tensorflow on Spark	Score
Ease of Getting Started	Documen...		
	Installatio...		
	Built-in Examples		
Ease of Use	Interface		
	Model		
Functi...			
Perfor...			
Status	Vitality		
	Enterprise Support		

```
def __init__(self):
    session = tf.InteractiveSession()
    x = tf.placeholder("float", shape=[None, 784], name='x')
    x_image = tf.reshape(x, [-1,28,28,1], name='reshape')
    y_ = tf.placeholder("float", shape=[None, 10], name='y_')
    W_conv1 = weight_variable([5, 5, 1, 32], 'W_conv1')
    b_conv1 = bias_variable([32], 'b_conv1')
    h_conv1 = tf.nn.relu(conv2d(x_image, W_conv1) + b_conv1)
    h_pool1 = max_pool_2x2(h_conv1)
    W_conv2 = weight_variable([5, 5, 32, 64], 'W_conv2')
    ...
    W_fc2 = weight_variable([1024, 10], 'W_fc2')
    b_fc2 = bias_variable([10], 'b_fc2')
```

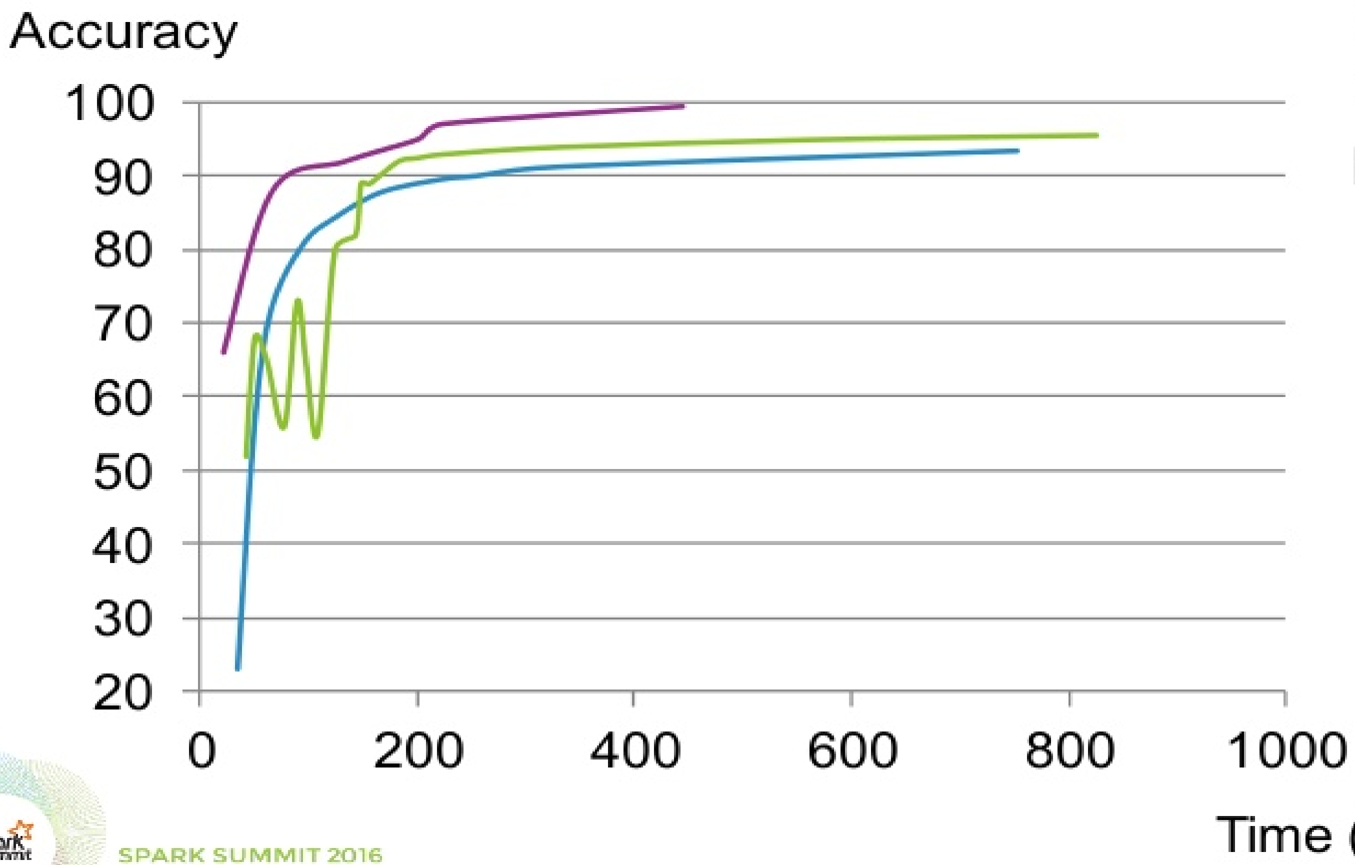
	1	2	3	4
Epochs	5	10	15	20
Time(seconds)	223	415	615	828
Accuracy	93%	94%	94.2%	95.4%

Benchmark – MNIST



One master (16-Core,64GB)
Five slaves (8-Core,32GB)
Executor memory: 20GB
Batch size: 64

Benchmark – MNIST



One master (16-Core,64GB)
Five slaves (8-Core,32GB)
Executor memory: 20GB
Batch size: 64

Evaluation Criteria	Dimensions	SparkNet	DL4J	CaffeOnSpark	Tensorflow on Spark
Ease of Getting Started	Documentation	☆☆☆☆	☆☆☆☆	☆☆☆☆	☆☆☆☆
	Installation	☆☆☆☆☆	☆☆☆☆☆	☆☆☆	☆☆☆☆
	Built-in Examples	☆☆☆☆	☆☆☆☆	☆☆☆	☆☆☆☆
Ease of Use	Interface	☆☆☆☆	☆☆☆☆	☆☆☆☆☆	☆☆☆☆
	Model Encapsulation	☆☆☆☆☆	☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆
Functionality	Built-in Models	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆	☆☆☆☆
	Parallelism	☆☆☆	☆☆☆	☆☆☆	☆☆☆
Performance	Performance	☆☆☆☆	☆☆☆	☆☆☆☆☆	☆☆☆☆
Status Quo	Community Vitality	☆☆☆	☆☆☆☆	☆☆☆☆	☆☆☆
	Enterprise Support	☆☆☆	☆☆☆☆	☆☆☆☆	☆☆☆☆

Conclusion

- Common issues
  - Lack of model parallelism
  - Potential network congestion
  - Early-stage development
- Future evaluation work
  - GPU integration
  - SGD synchronization
  - Scalability

THANK YOU.

Zhe.Dong@emc.com