

# Distributed Deep Learning

**(And How to Get Involved)**

Sina Sheikholeslami

[sinash@kth.se](mailto:sinash@kth.se)

# Outline

- What Had Happened Was
- Distributed / Parallel Deep Learning
- Horovod
- TensorFlowOnSpark
- How to Get Involved
- Closing Notes
- Summary

**ONE DOES NOT SIMPLY**

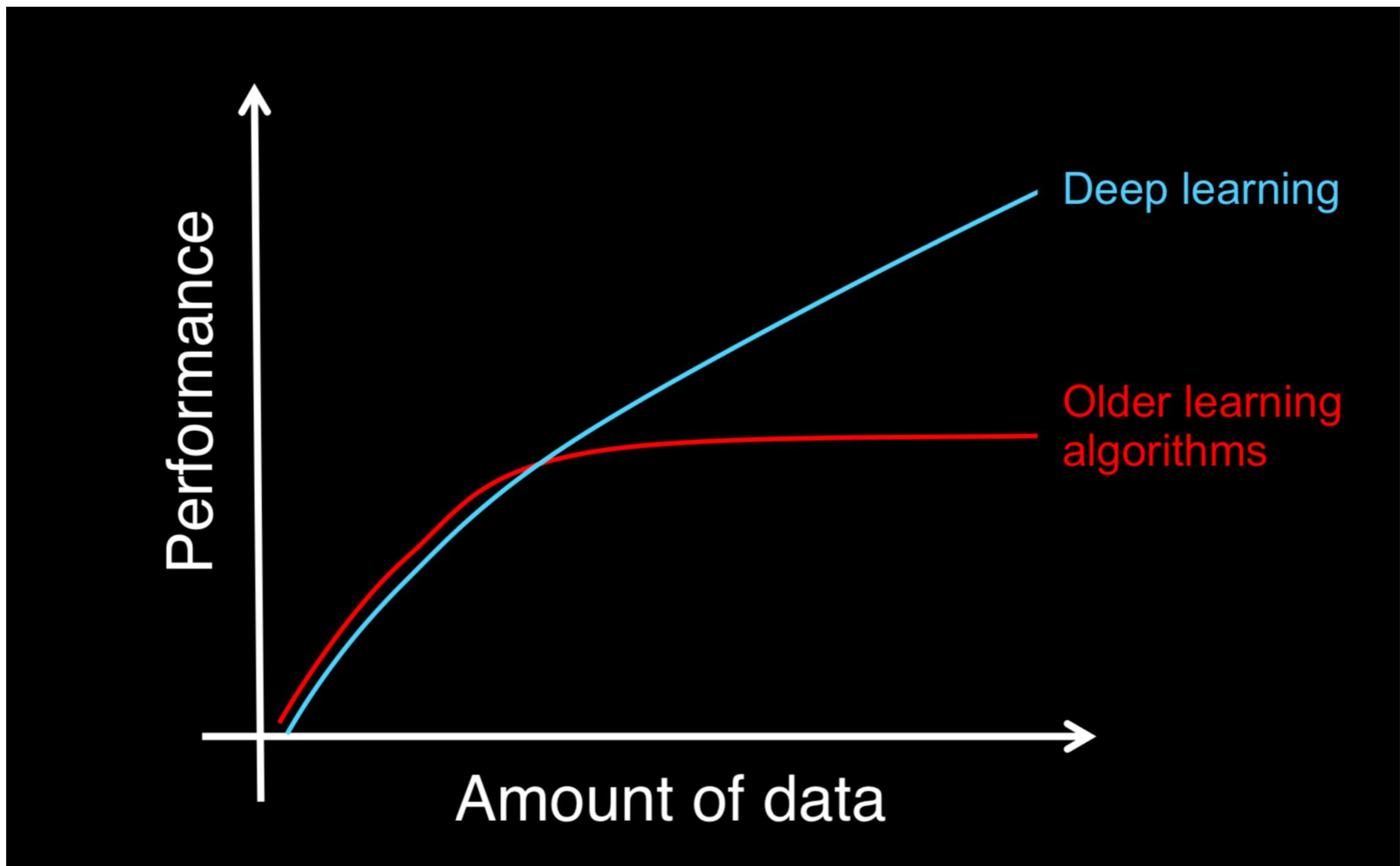
**LEARN DEEPLY**

imgflip.com

**What Had Happened Was...**

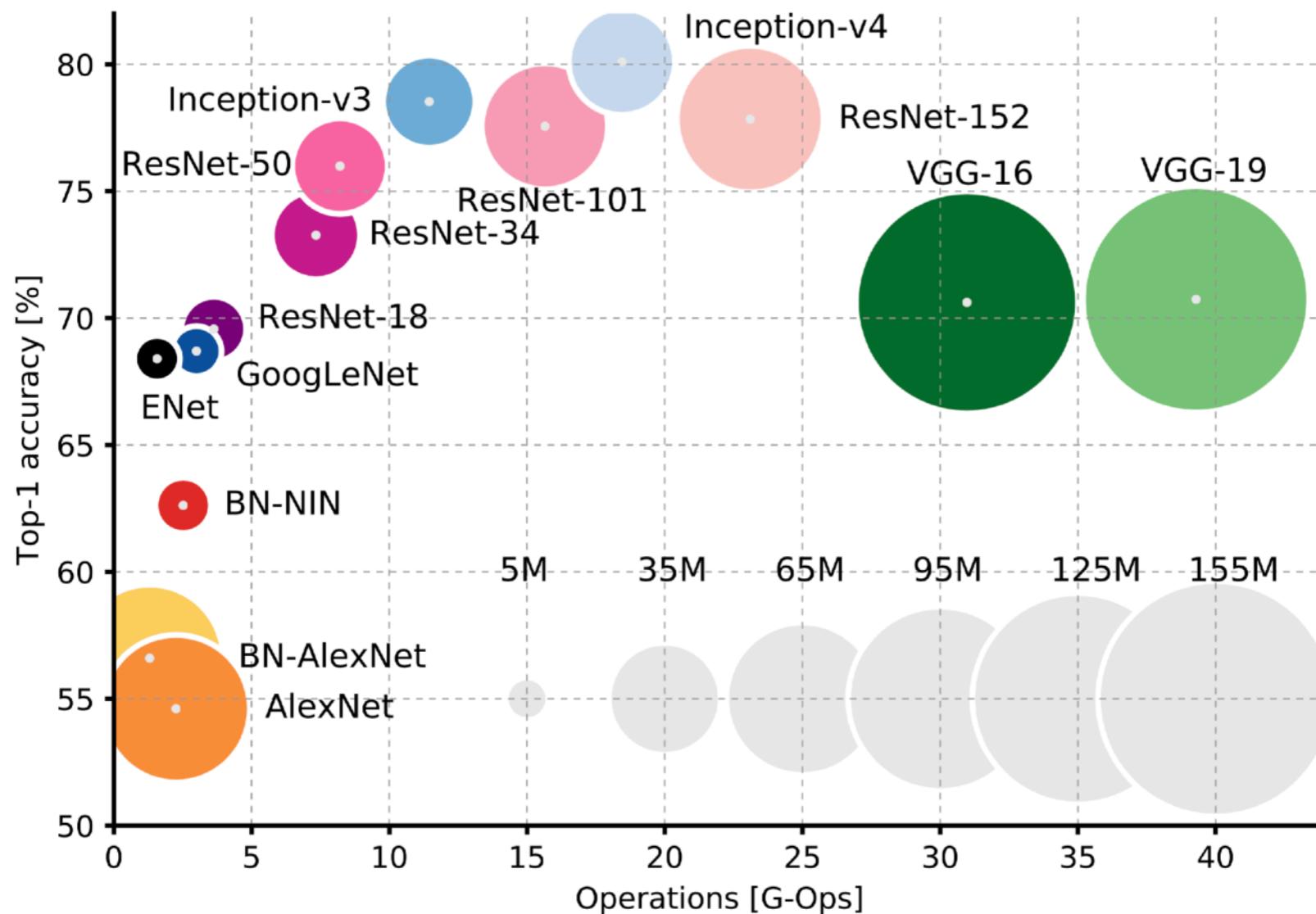
Motivations and Challenges for Distributed Deep Learning

# Why Deep Learning?



(Andrew Ng, ExtractConf 2015)

# Lots of Computation



(Canziani et al., 2016)

# Two Main Challenges

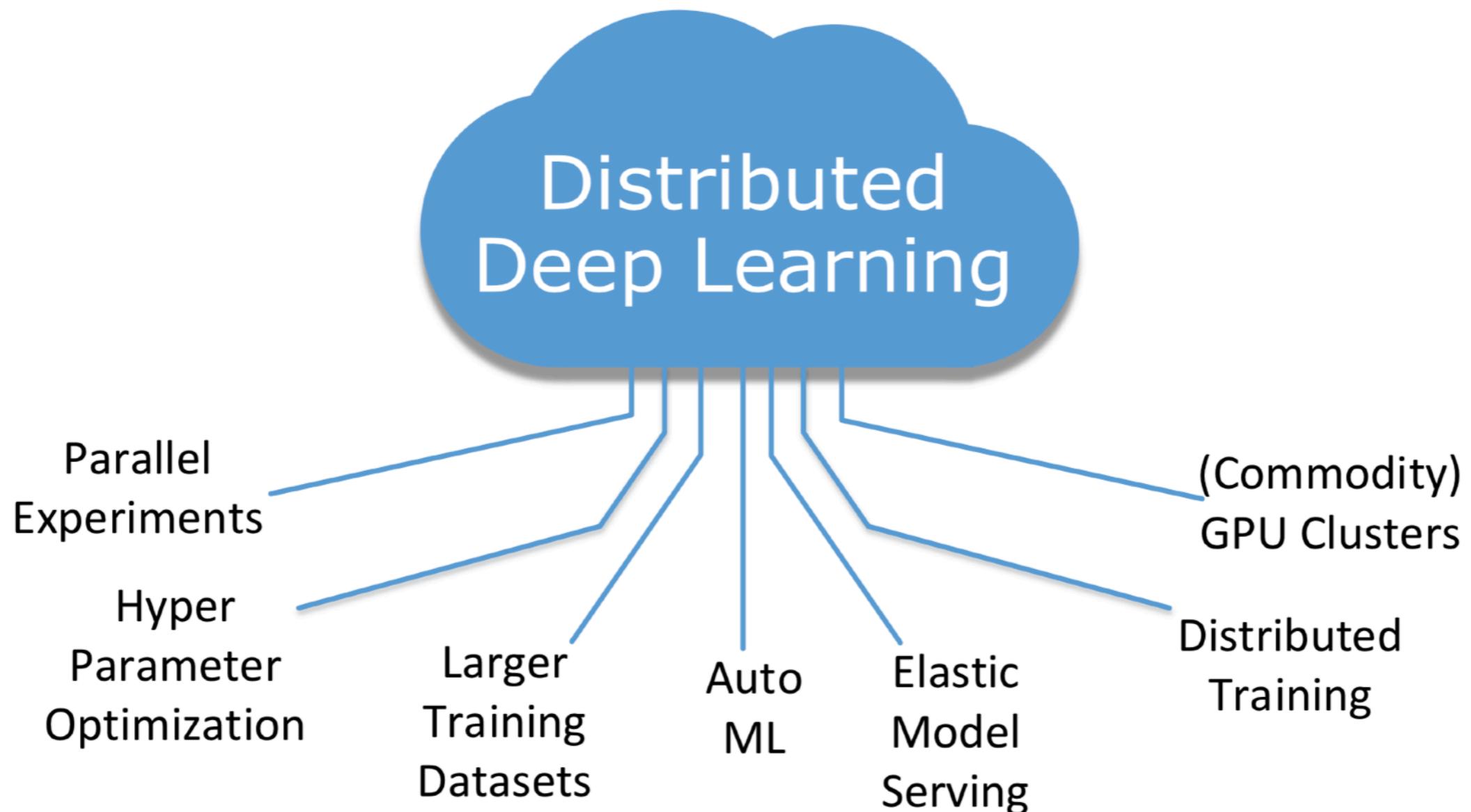
- Models are very large
- Training time is a lot



(“Parallel Training” - “Rail station with many trains on the tracks in New York” by [Maxim Melnikov](#))

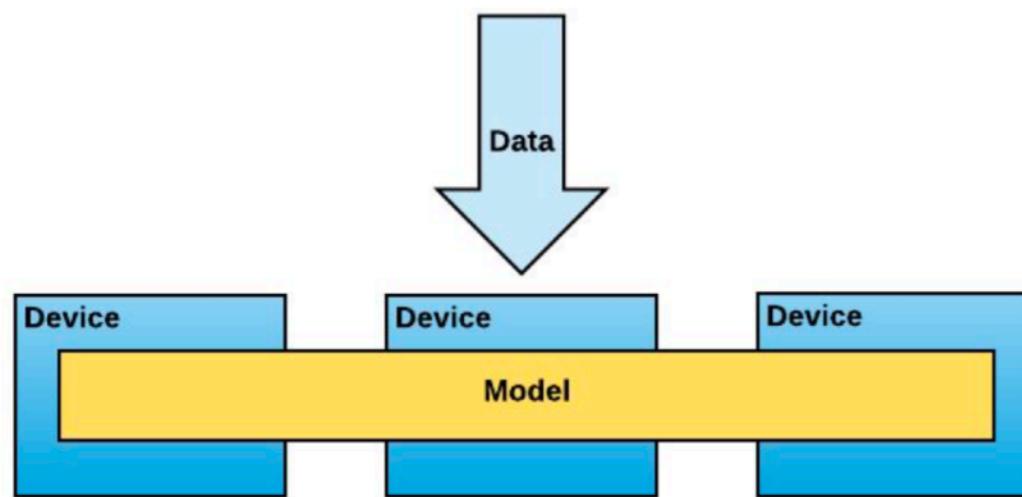
# Distributed / Parallel Deep Learning

# Distributed Deep Learning: (Some) Applications

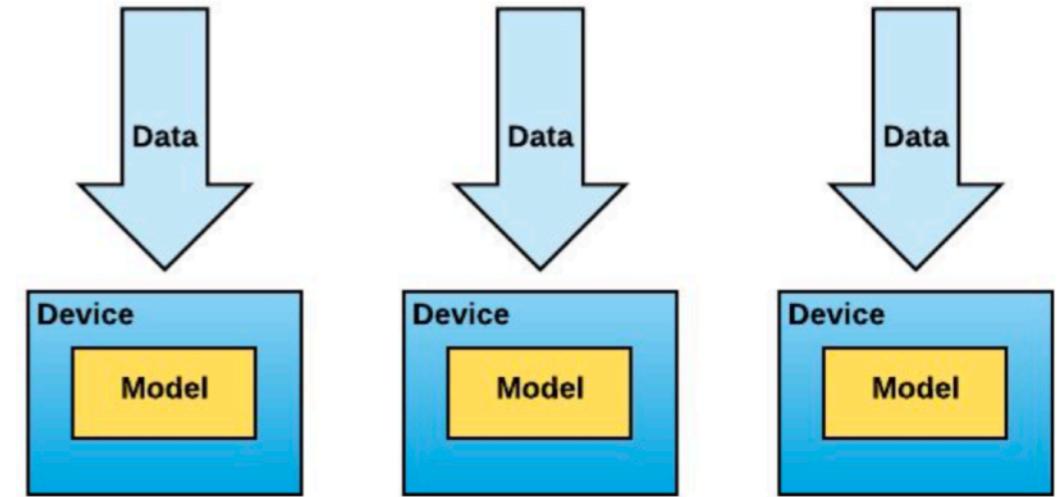


*(Courtesy of Logical Clocks AB)*

# Distributed Deep Learning Models



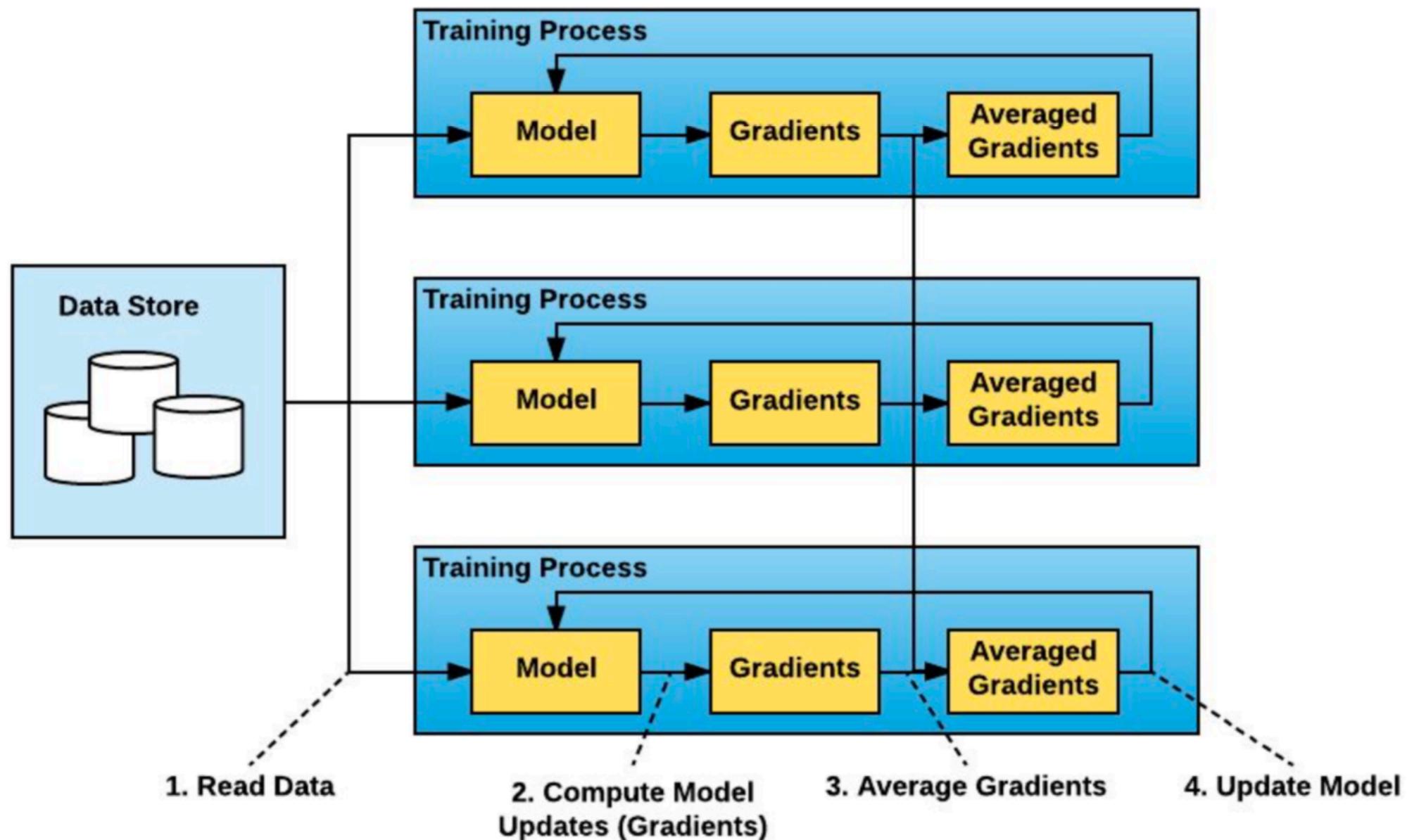
Model Parallelism



Data Parallelism

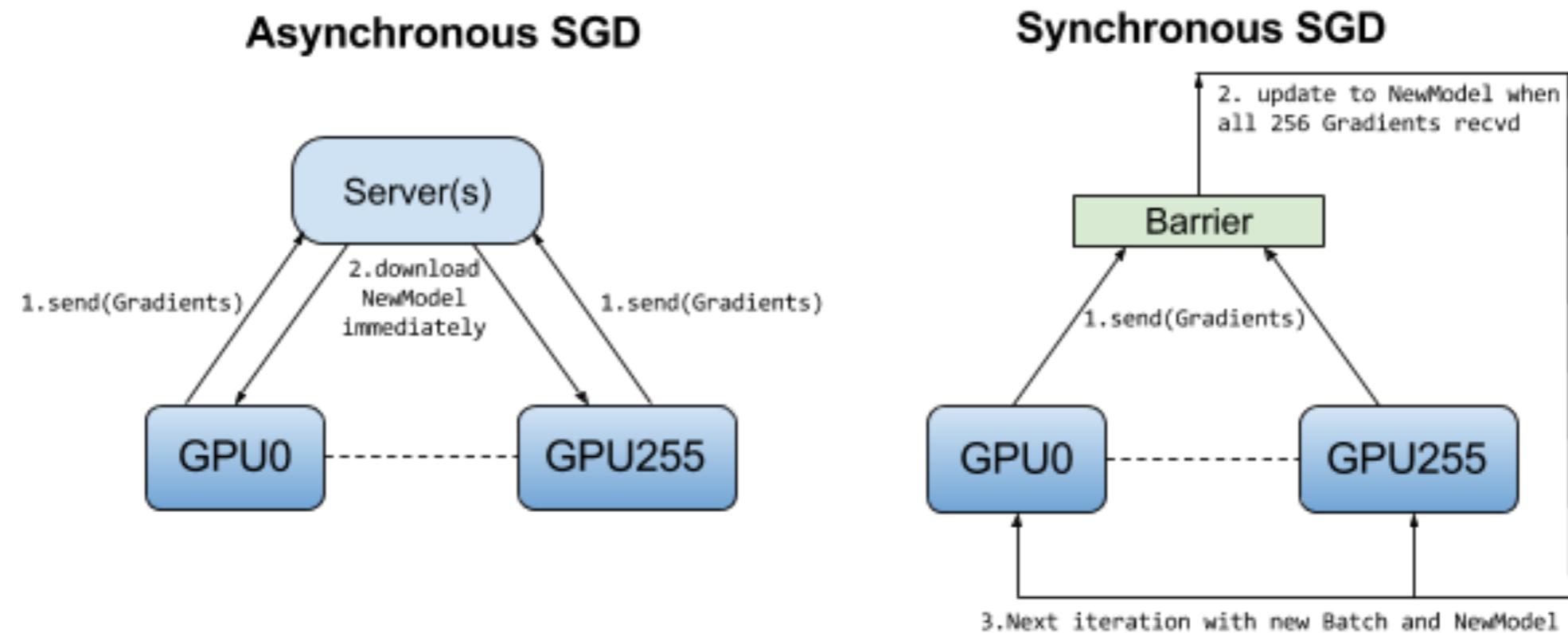
*(Courtesy of Horovod Team - Uber)*

# Example: Computing Gradients



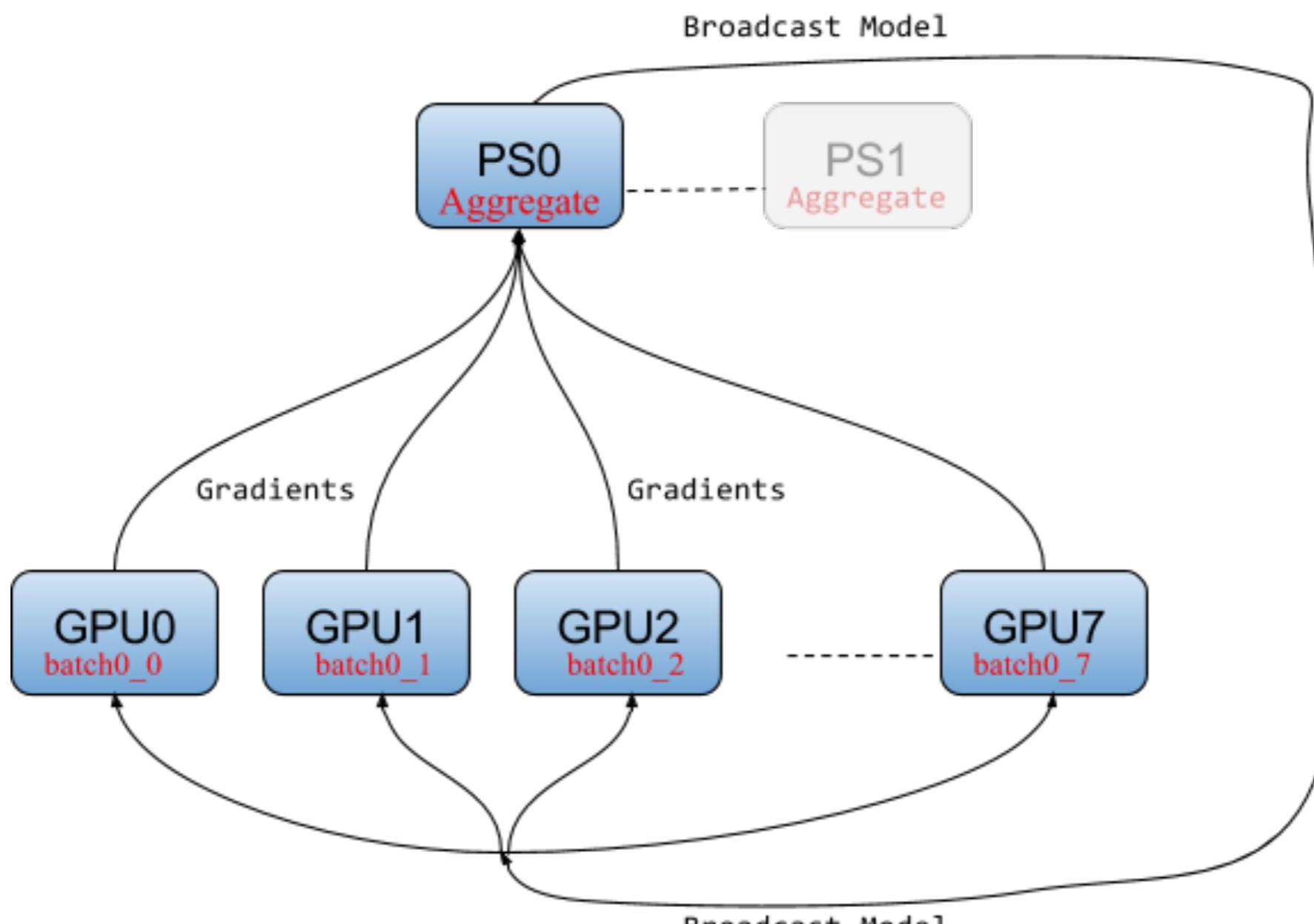
(Courtesy of Horovod Team - Uber)

# Example: Computing Gradients



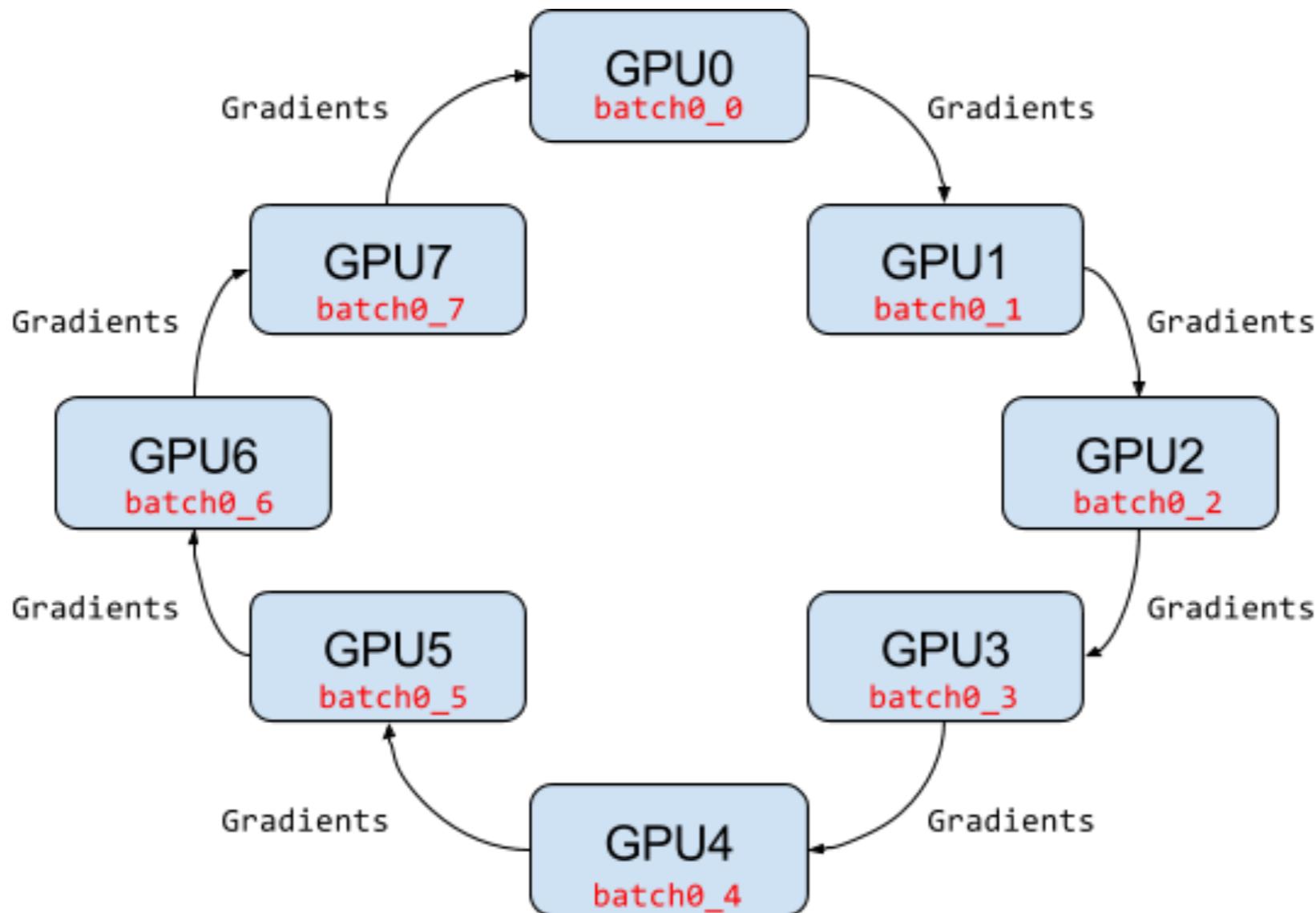
(Courtesy of Jim Dowling)

# Parameter Server Model for Synchronous SGD



(Courtesy of Jim Dowling)

# Ring-AllReduce Model for Synchronous SGD



(Courtesy of Jim Dowling)



(Russian Dance Group Berezka, taken from [laughandshare.com](http://laughandshare.com))

# Horovod

# Horovod in a Nutshell



- Open-source platform by Uber
- Wrapper for TensorFlow, Keras, PyTorch
- Separation of Infrastructure from ML

# Horovod: 5 Easy Steps

- Initialize the library

```
import tensorflow as tf
import horovod.tensorflow as hvd

# Initialize Horovod
hvd.init()
```

- Pin a GPU to each worker

```
# Pin GPU to be used to process local rank (one GPU per process)
config = tf.ConfigProto()
config.gpu_options.visible_device_list = str(hvd.local_rank())

# Build model...
loss = ...
opt = tf.train.AdagradOptimizer(0.01)
```

- Wrap the optimizer

```
# Add Horovod Distributed Optimizer
opt = hvd.DistributedOptimizer(opt)

# Add hook to broadcast variables from rank 0 to all other processes during
# initialization.
hooks = [hvd.BroadcastGlobalVariablesHook(0)]
```

- Synchronize state across workers

```
# Make training operation
train_op = opt.minimize(loss)
```

- Checkpoint on the first worker

```
# The MonitoredTrainingSession takes care of session initialization,
# restoring from a checkpoint, saving to a checkpoint, and closing when done
# or an error occurs.
with
tf.train.MonitoredTrainingSession(checkpoint_dir="/tmp/train_logs",
                                    config=config,
                                    hooks=hooks) as mon_sess:
while not mon_sess.should_stop():
    # Perform synchronous training.
    mon_sess.run(train_op)
```



TensorFlow



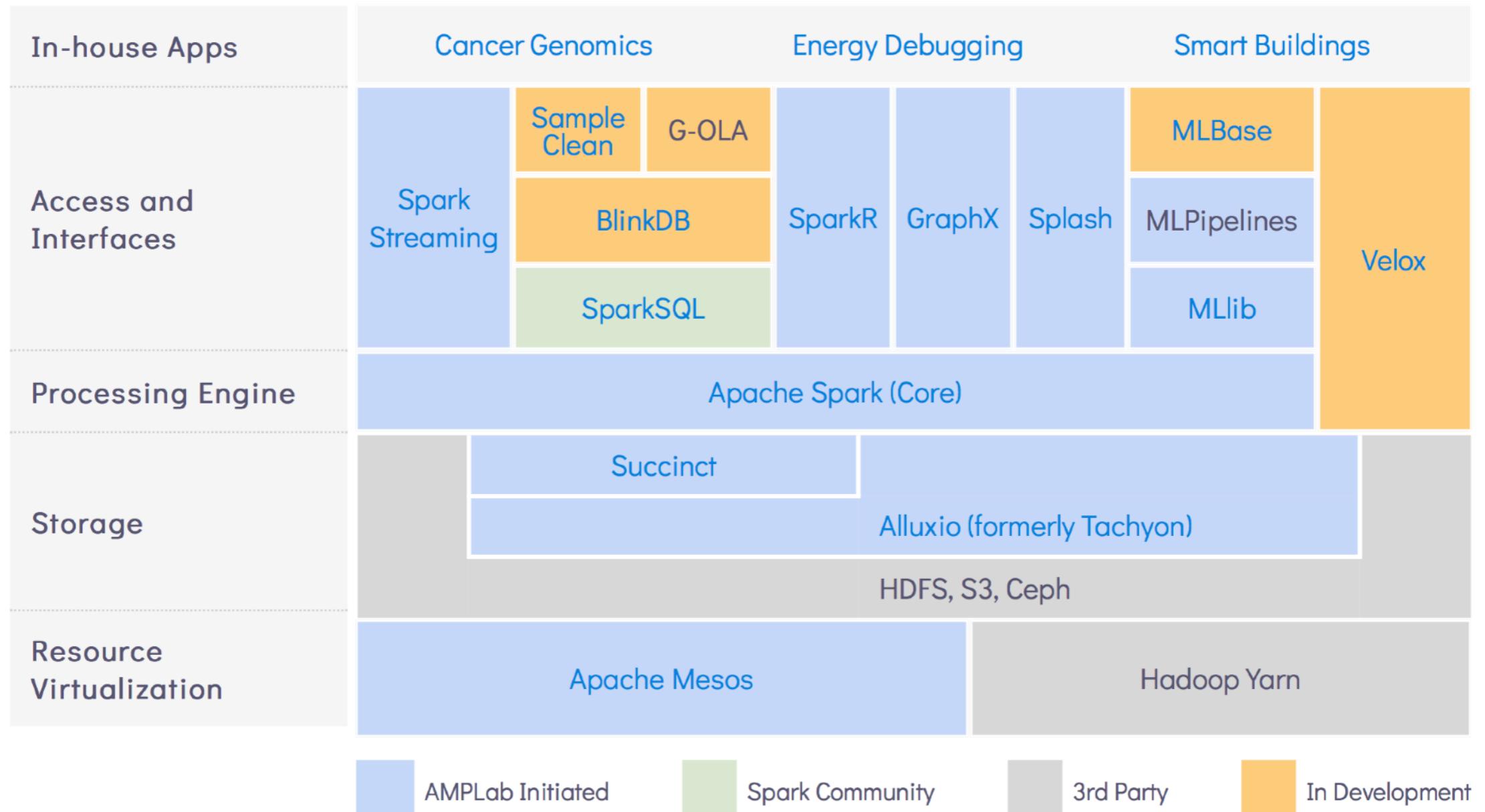
**TensorFlowOnSpark**

# But First... Apache Spark

- In-Memory Distributed Processing Platform
- Similar Semantics for Batch & Stream Processing
- Initially started by Matei Zaharia at UC Berkeley's AMPLab in 2009
- Became a top-level Apache Project in February 2014
- Written primarily in Scala, more than 1M lines of code



# The Bigger Picture



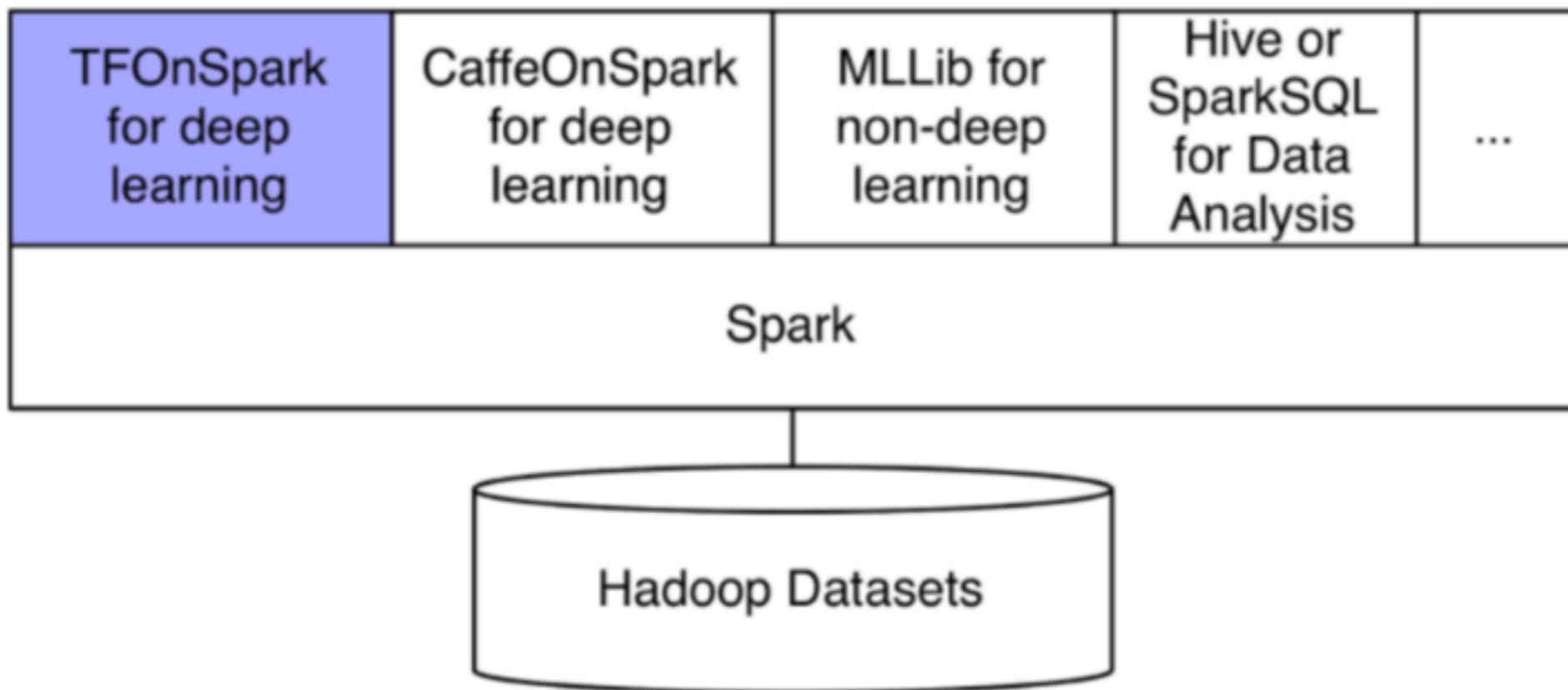
(BDAS, the Berkeley Data Analytics Stack)

# TensorFlowOnSpark in a Nutshell



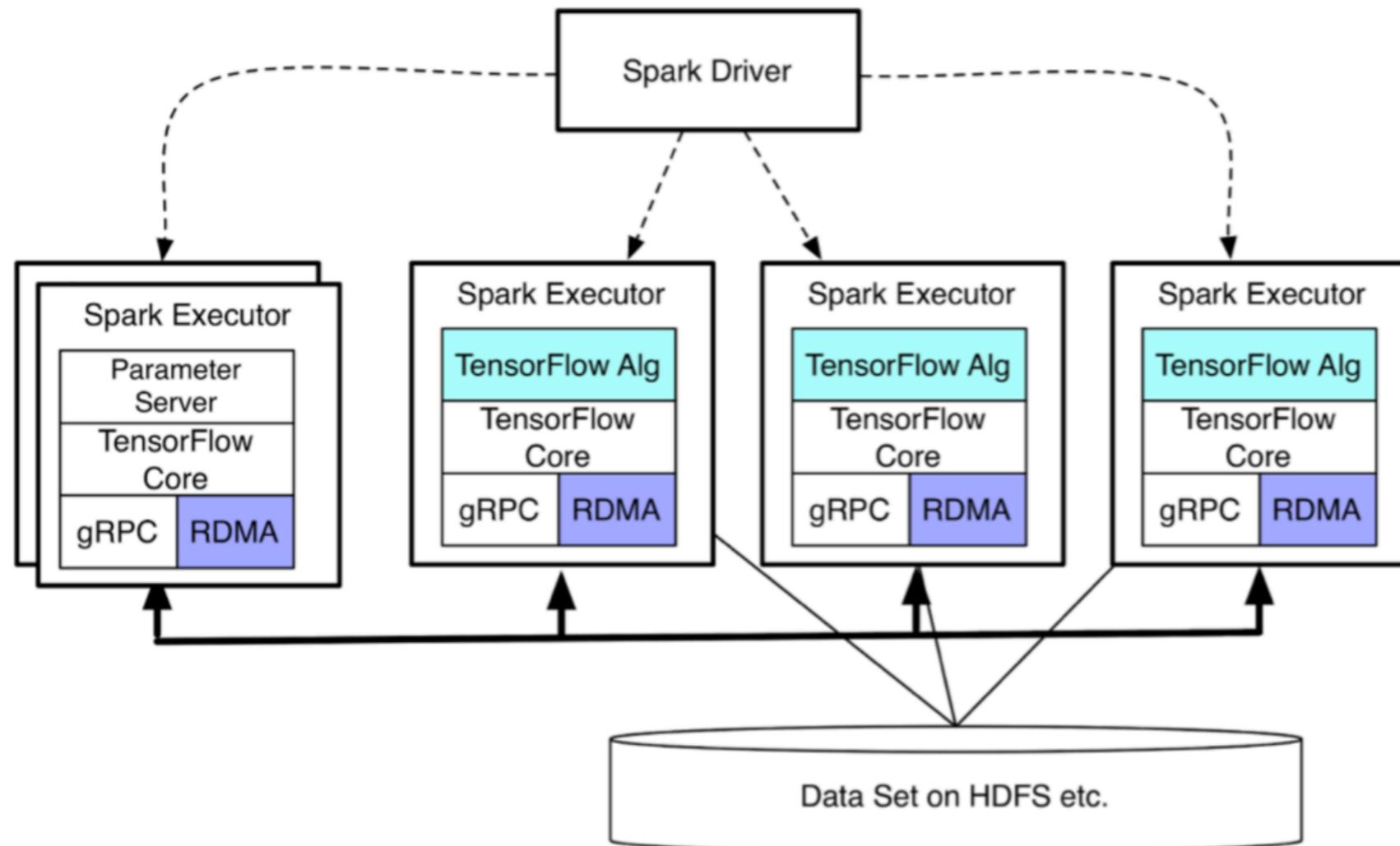
- Basically a PySpark wrapper of TensorFlow Code
- Launches distributed TensorFlow clusters
- Also compatible with TensorBoard

# Recall the Big Picture



*(Courtesy of Yahoo Big Data ML Platform Team)*

# TensorFlowOnSpark Architecture



(Courtesy of Yahoo Big Data ML Platform Team)

# TensorFlowOnSpark Workflow

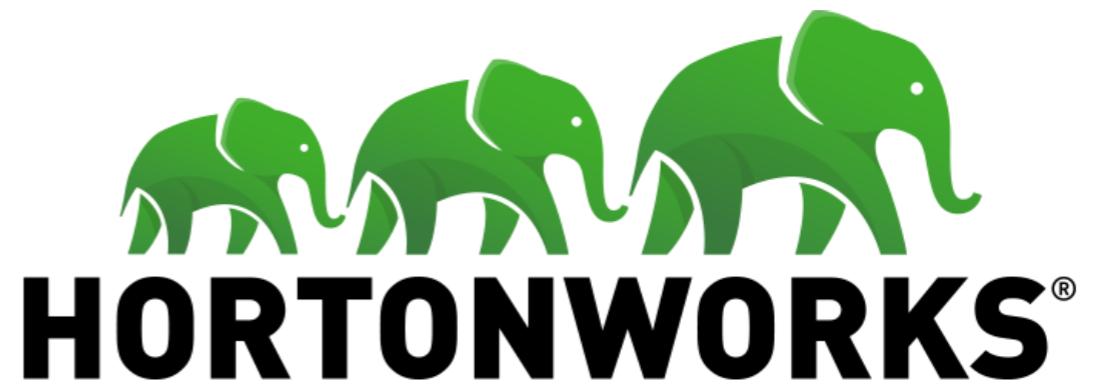
```
cluster = TFCluster.run(sc, map_fn, args, num_executors,  
num_ps, tensorboard, input_mode)  
cluster.train(dataRDD, num_epochs=0)  
cluster.inference(dataRDD)  
cluster.shutdown()
```

- Launch the TensorFlow cluster
- Feed data to the TensorFlow app
- Shutdown the TensorFlow cluster

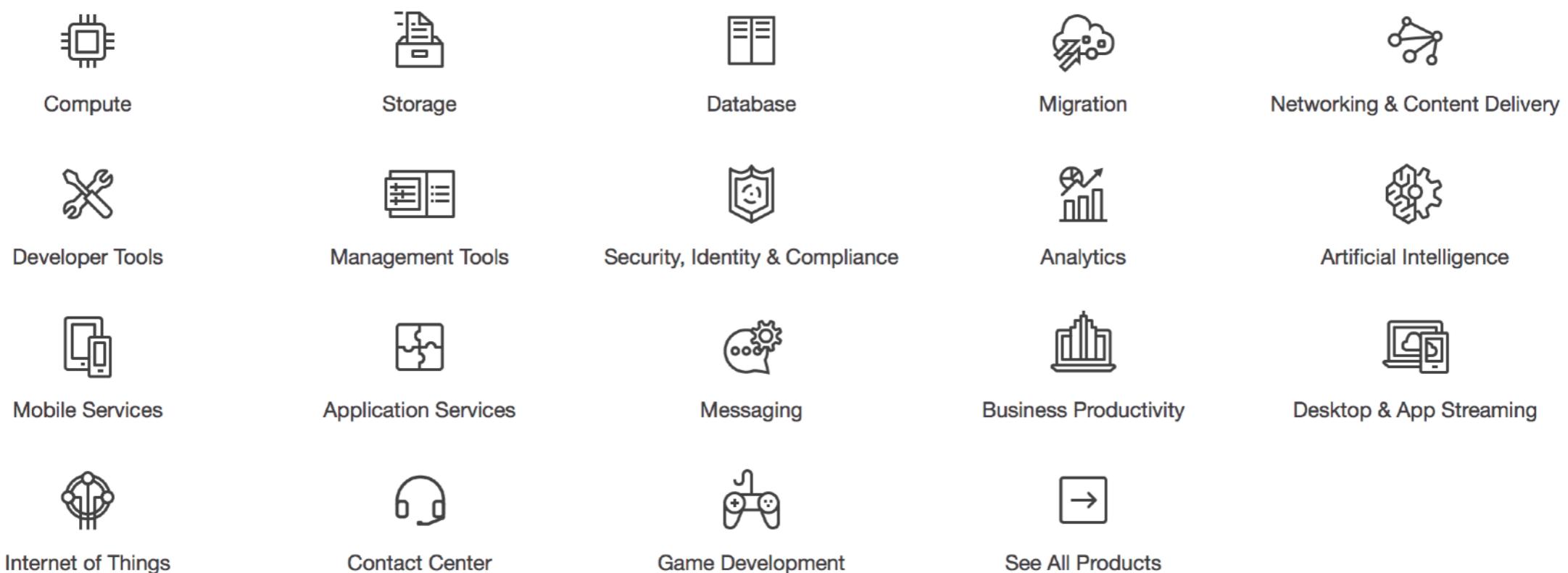
# How to Get Involved



FLOYDHUB

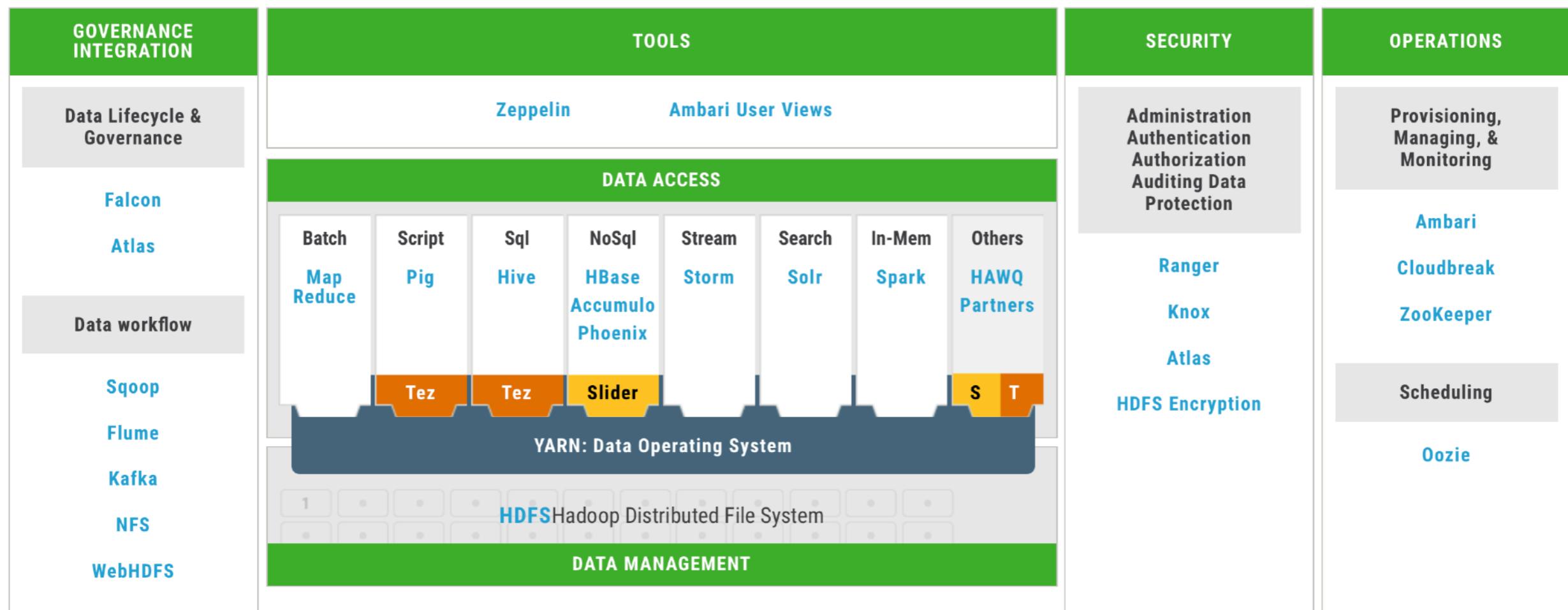


# Amazon Web Services



(<https://aws.amazon.com>)

# Hortonworks Data Platform



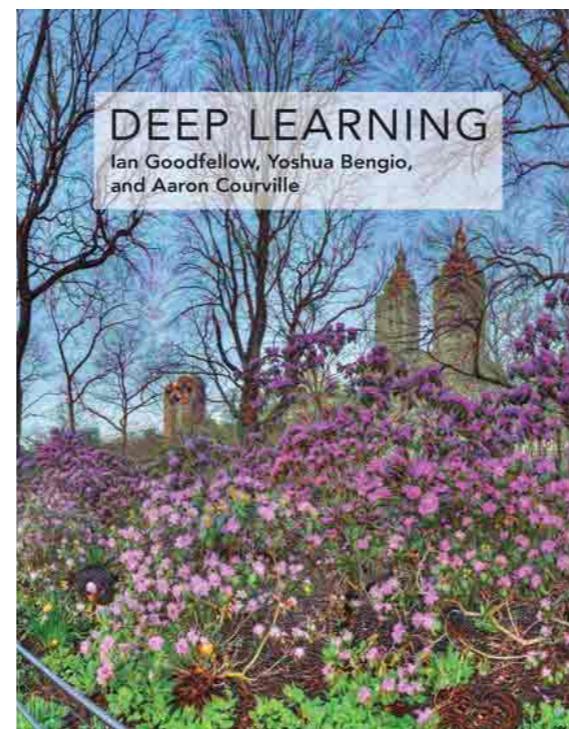
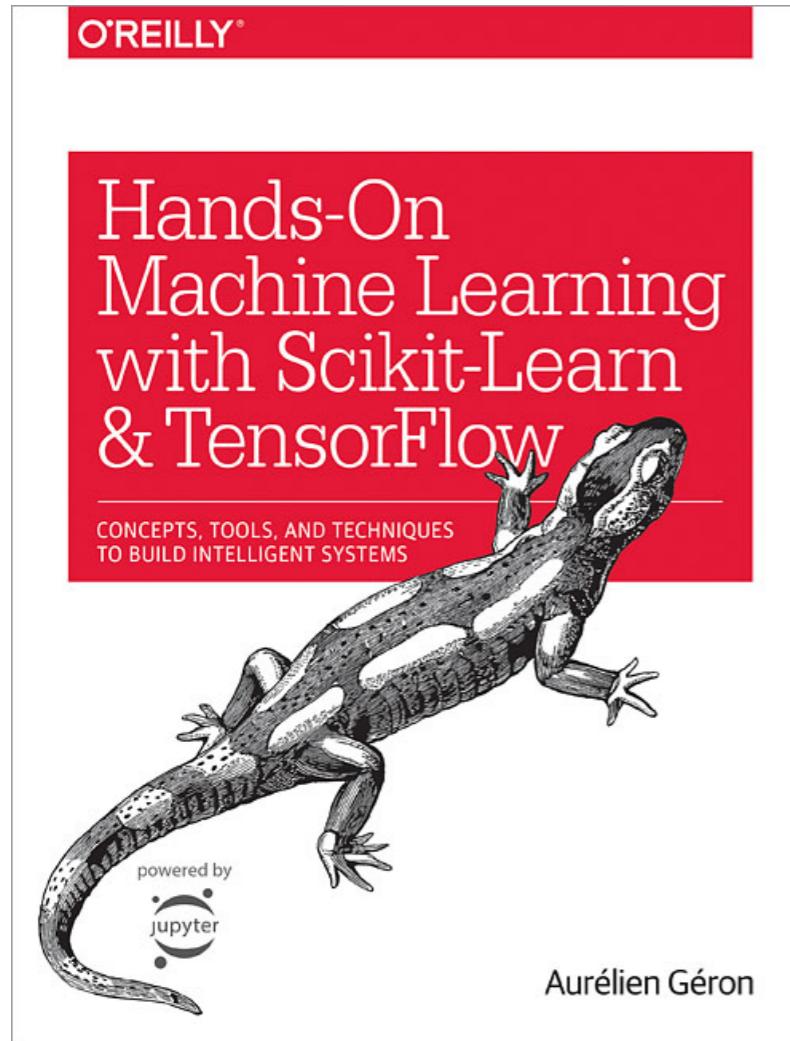
(<https://hortonworks.com/products/data-center/hdp/>)

# Open-source!

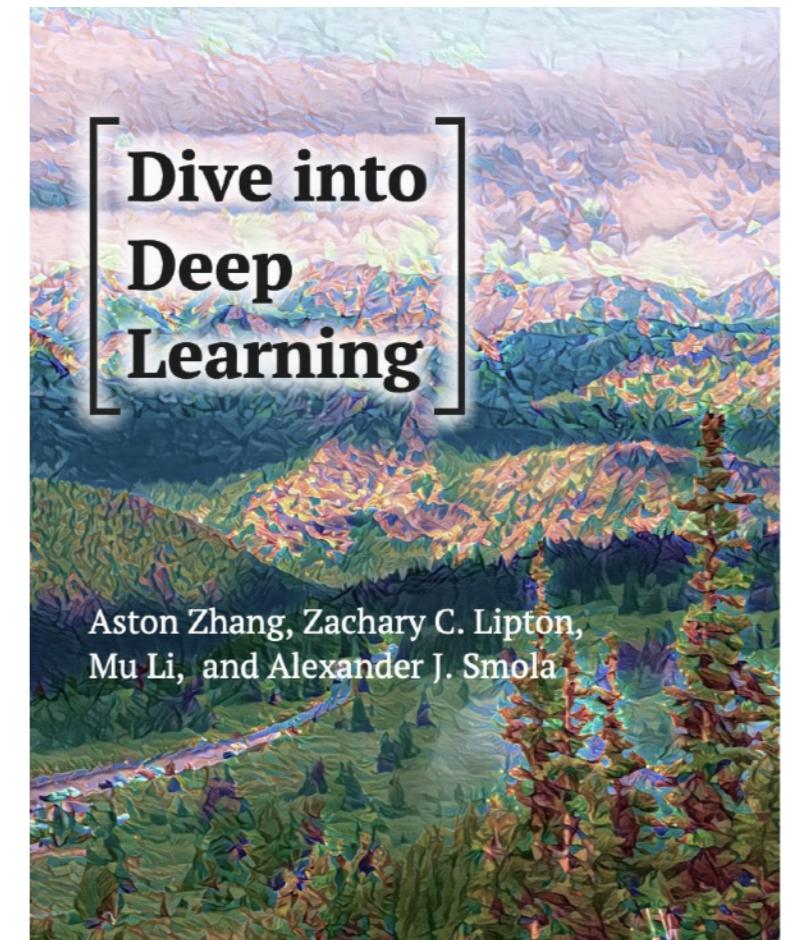


**Go fork 'em on GitHub!**

# Some GOOD Resources



**Mathematics for Machine Learning**  
Companion webpage to the book "Mathematics for Machine Learning". Copyright 2018 by Marc Peter Deisenroth, A Aldo Faisal, and Cheng Soon Ong. To be published by Cambridge University Press.



# Summary

- Challenges and Motivations
- Distributed / Parallel Deep Learning Models
  - Data Parallelism
  - Model Parallelism
  - Parameter Servers and Ring-AllReduce
- Horovod
- Spark and TensorFlowOnSpark
- How to Get Involved

Thank  
you!

# References

- Andrew Ng talk at ExtractConf 2015 [<https://www.slideshare.net/ExtractConf/andrew-ng-chief-scientist-at-baidu>]
- CNN Architectures: LeNet, ... (Medium) [<https://medium.com/@sidereal/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5>]
- Canziani, Alfredo, Adam Paszke, and Eugenio Culurciello. "An analysis of deep neural network models for practical applications." *arXiv preprint arXiv:1605.07678* (2016).
- Ben-Nun, Tal, and Torsten Hoefer. "Demystifying Parallel and Distributed Deep Learning: An In-Depth Concurrency Analysis." *arXiv preprint arXiv:1802.09941* (2018).
- Goodbye Horovod, Hello CollectiveAllReduce | Logical Clocks AB [<https://www.logicalclocks.com/goodbye-horovod-hello-tensorflow-collectiveallreduce/>]
- ImageNet Summary and Statistics (2010) [<http://image-net.org/about-stats>]
- Li, Mu, et al. "Scaling Distributed Machine Learning with the Parameter Server." *OSDI*. Vol. 14. 2014.
- Distributed TensorFlow | Jim Dowling on O'Reilly Ideas [<https://www.oreilly.com/ideas/distributed-tensorflow>]
- Horovod | Uber Engineering [<https://eng.uber.com/horovod/>]
- TensorFlowOnSpark | Yahoo [<https://github.com/yahoo/TensorFlowOnSpark>]