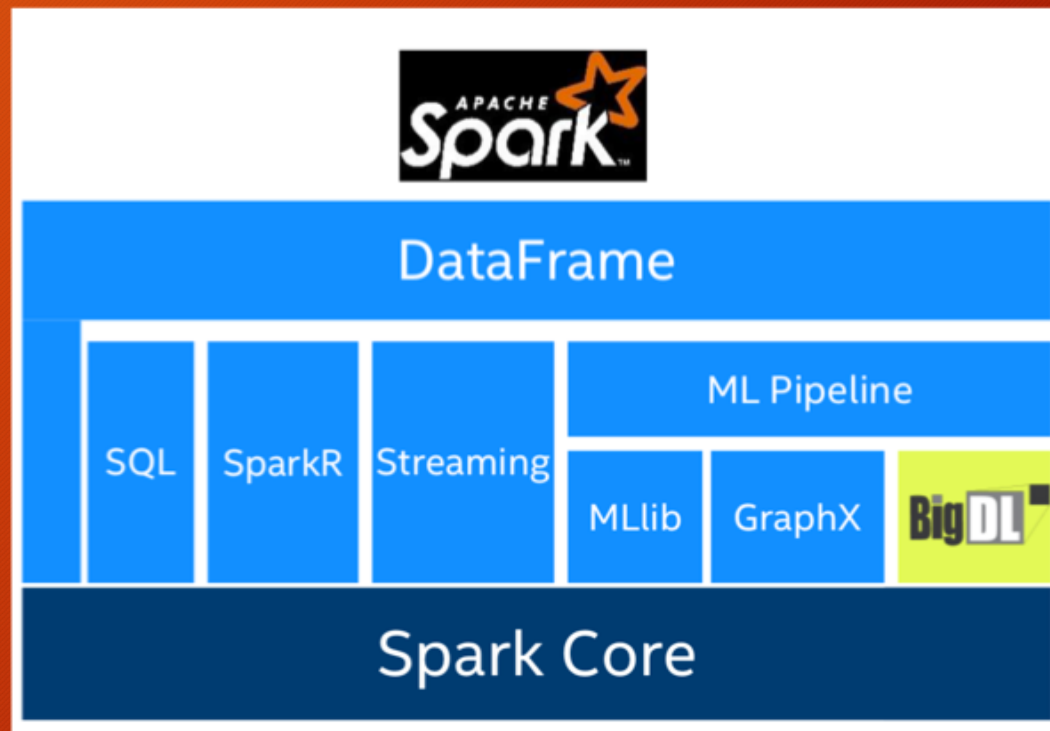
Spark BigDL

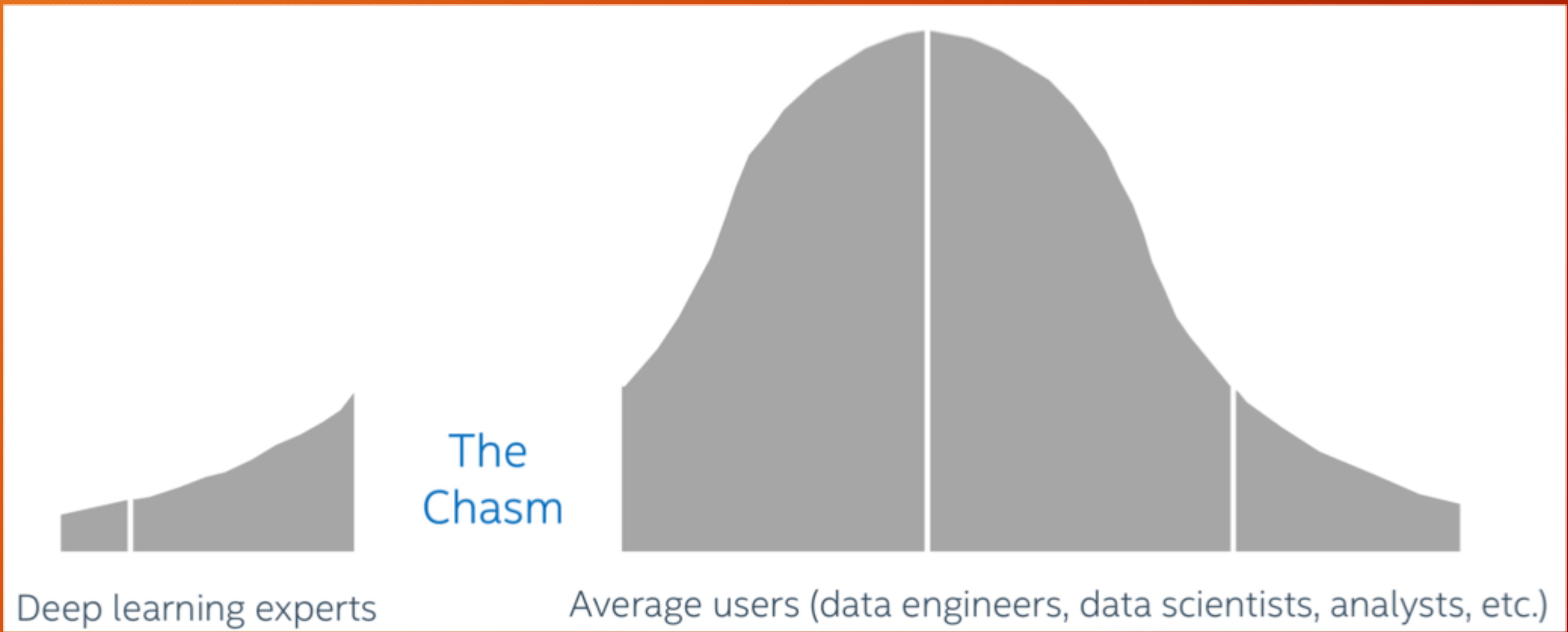
Slides based heavily on those by Jason Dai and Ding Ding, taken from AI Conference 2017 San Francisco

What is BigDL?

- BigDL is a deep learning library **built for Apache Spark**
- Make deep learning more accessible
  - Write deep learning applications as **standard Spark programs**
  - Run on existing Spark/Hadoop clusters **(no changes needed)**
- Feature parity with Caffe, Torch, TensorFlow
- High-performance Intel MKL library in tasks, efficient all-reduce and SGD at scale
- Works with pre-trained Keras, Caffe, or Torch models



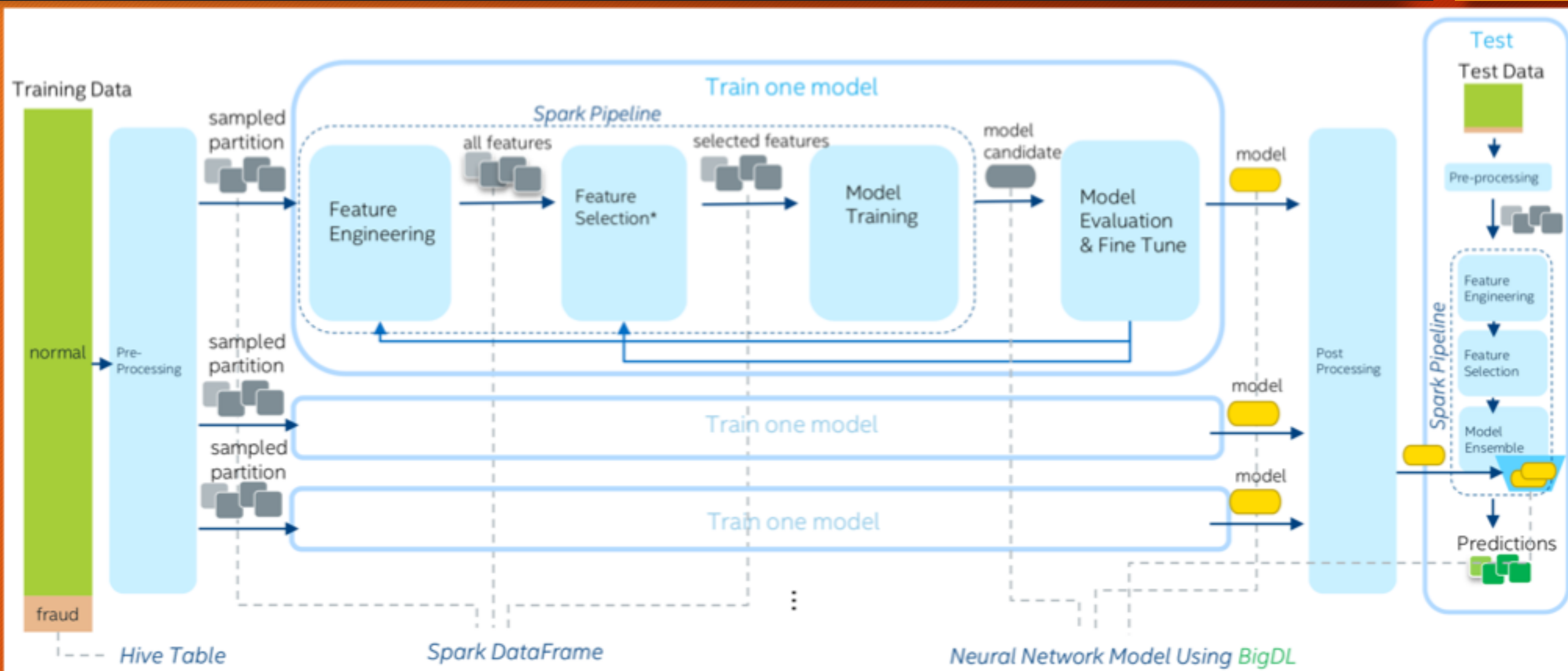
Making deep learning accessible



BigDL's Goals

- **Make deep learning more accessible to big data and data science communities**
- Continue the use of familiar software tools (Spark) and hardware infrastructure (Hadoop clusters) to build deep learning applications
- Analyze "big data" using deep learning on the same Hadoop/Spark cluster where the data are stored
- Add deep learning functionalities to the Big Data (Spark) programs and/or workflow
- Leverage existing Hadoop/Spark clusters to run deep learning applications
  - Shared with other workloads (e.g., ETL, data warehouse, feature engineering, statistic machine learning, graph analytics, etc.) in a dynamic and elastic fashion

Case Study: Fraud Detection for UnionPay



https://mp.weixin.qq.com/s?__biz=MzI3NDAwNDUwNg==&mid=2648307335&idx=1&sn=8eb9f63eaf2e40e24a90601b9cc03d1f

Distributed Training in BigDL

Iterative

Mini-batch

Data parallel

Training

```
for (i <- 1 to N) {
  batch = next_batch()
  output = model.forward(batch.input)
  loss = criterion.forward(output, batch.target)
  error = criterion.backward(output, batch.target)
  model.backward(input, error)
  optimMethod.optimize(model.weight, model.gradient)
}
```

Synchronous SGD

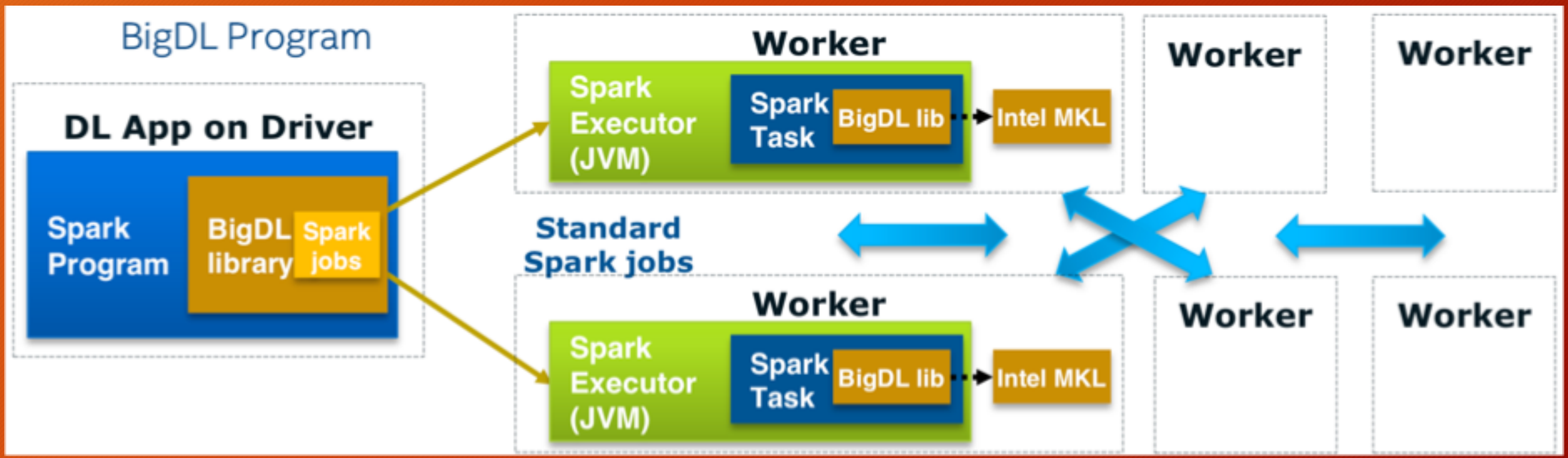
“Data parallel” vs “Model parallel”

Runtime complexity

Run as a standard Spark program

- Standard Spark jobs
  - No changes to the Spark or Hadoop clusters needed
- Iterative
  - Each iteration of the training runs as a Spark job

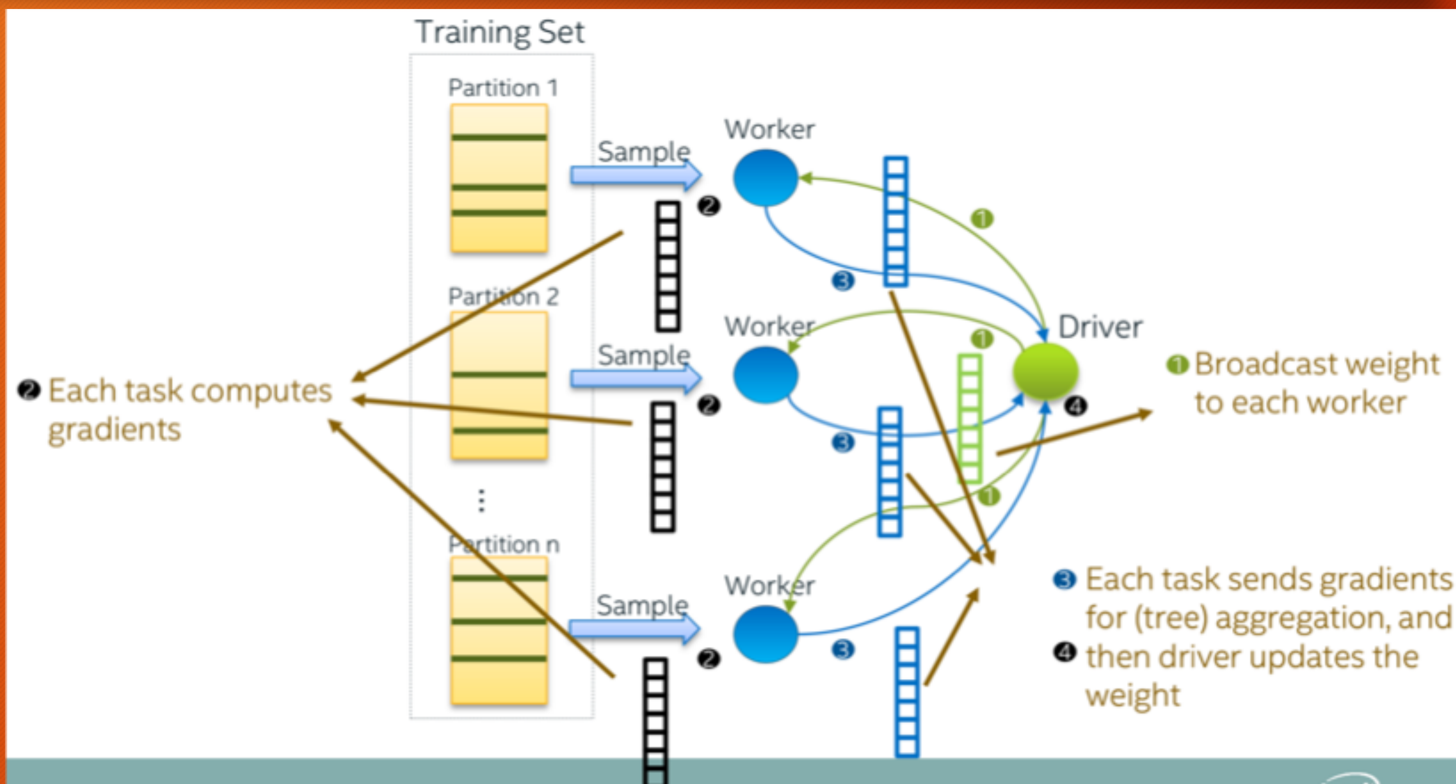
- Data parallel
  - Each Spark task runs the same model on a subset of the data (batch)



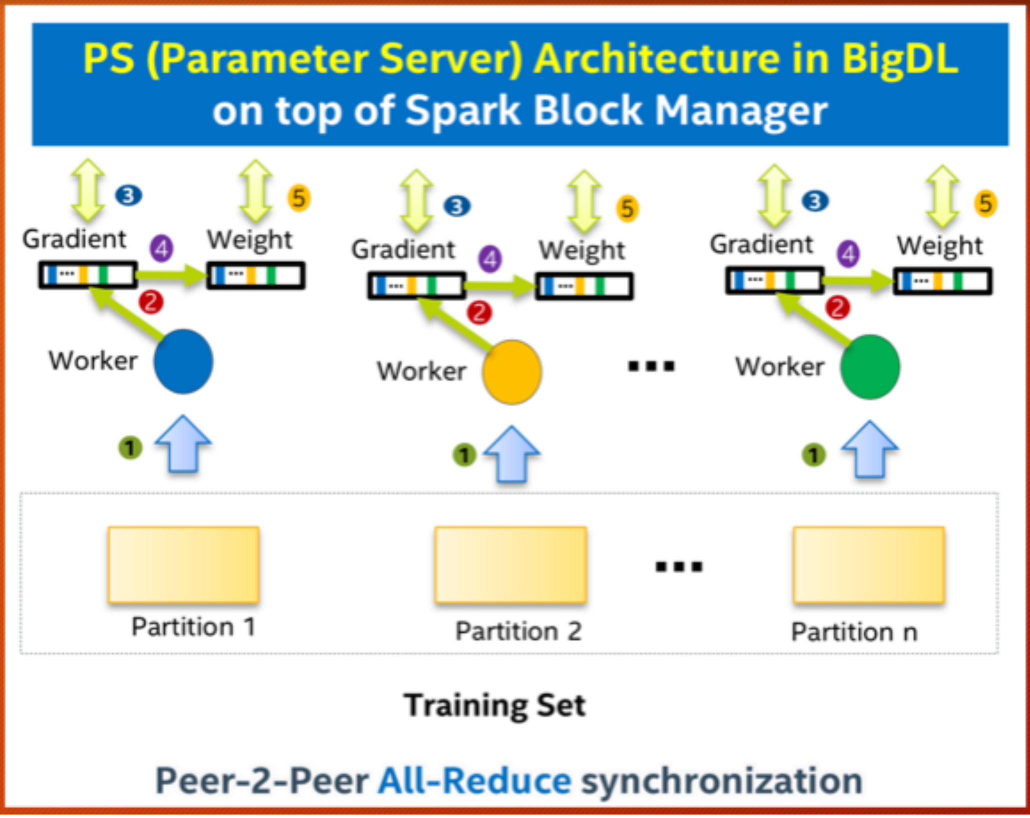
Considerations in large-scale distributed training

- Optimizing parameter synchronization and aggregation
- Optimizing task scheduling
- Scaling batch size

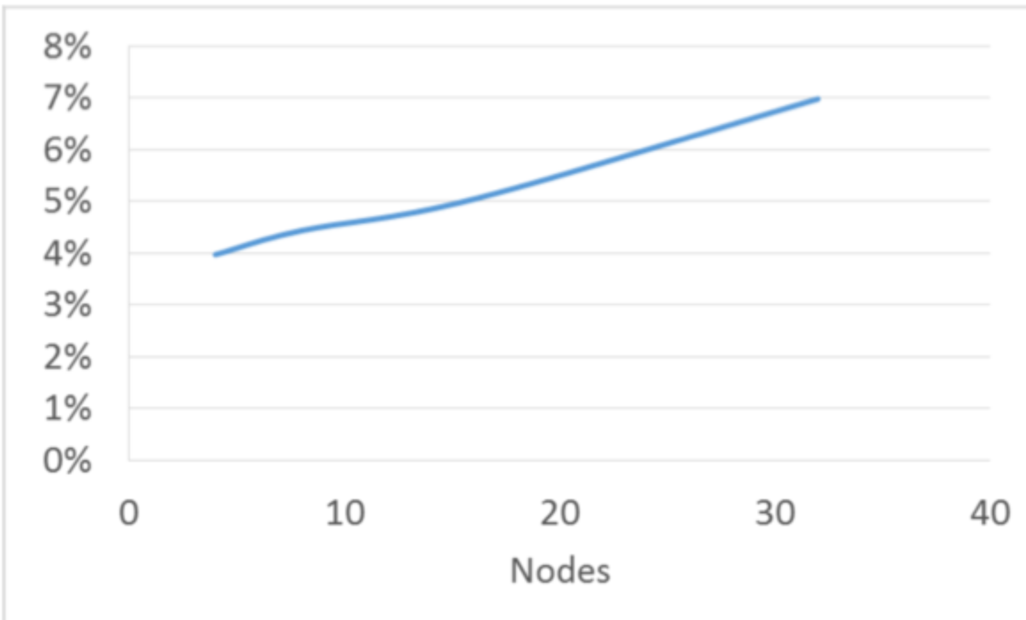
Parameter Synchronization in Spark MLlib



Parameter Synchronization in BigDL

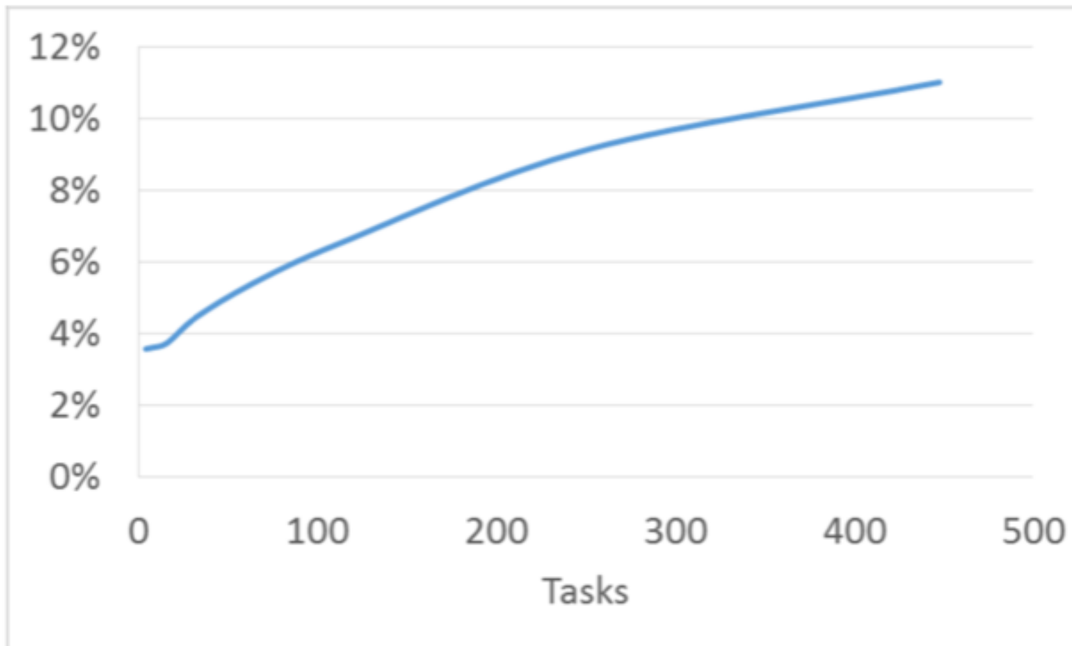


Performance of BigDL Parameter Synchronization



Parameter synchronization time as a fraction of average compute time for Inception v1 training

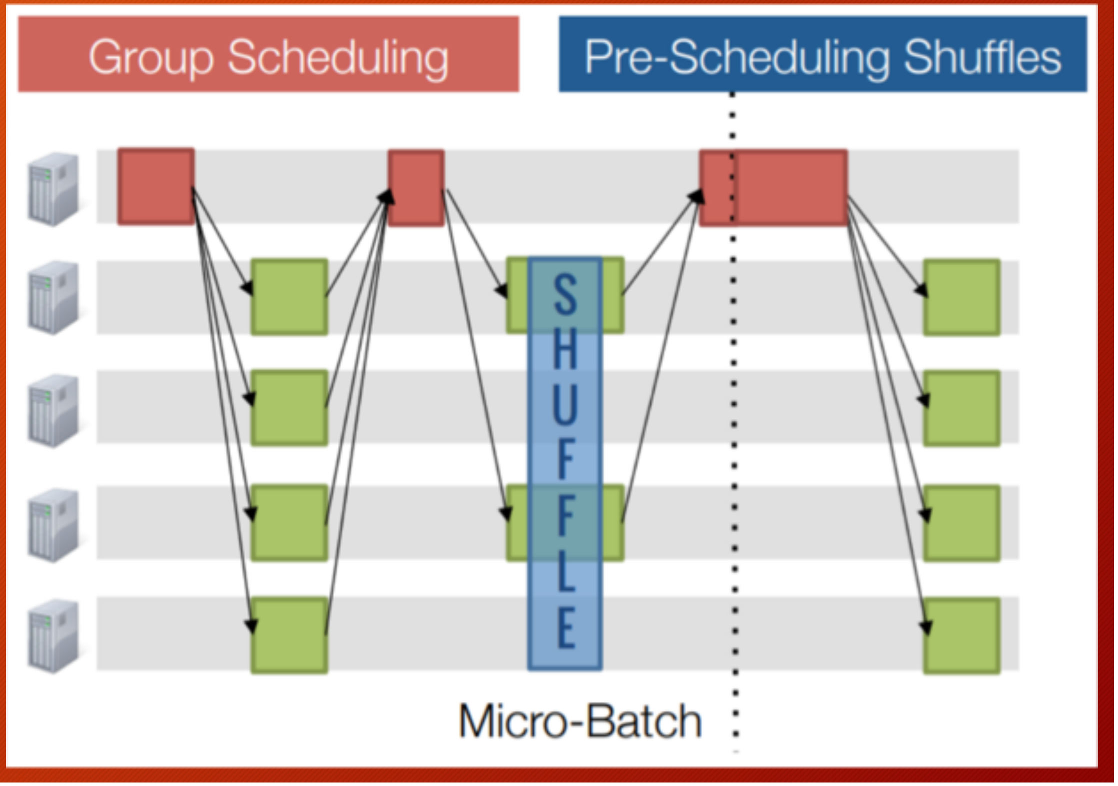
Spark Task Scheduling Overheads



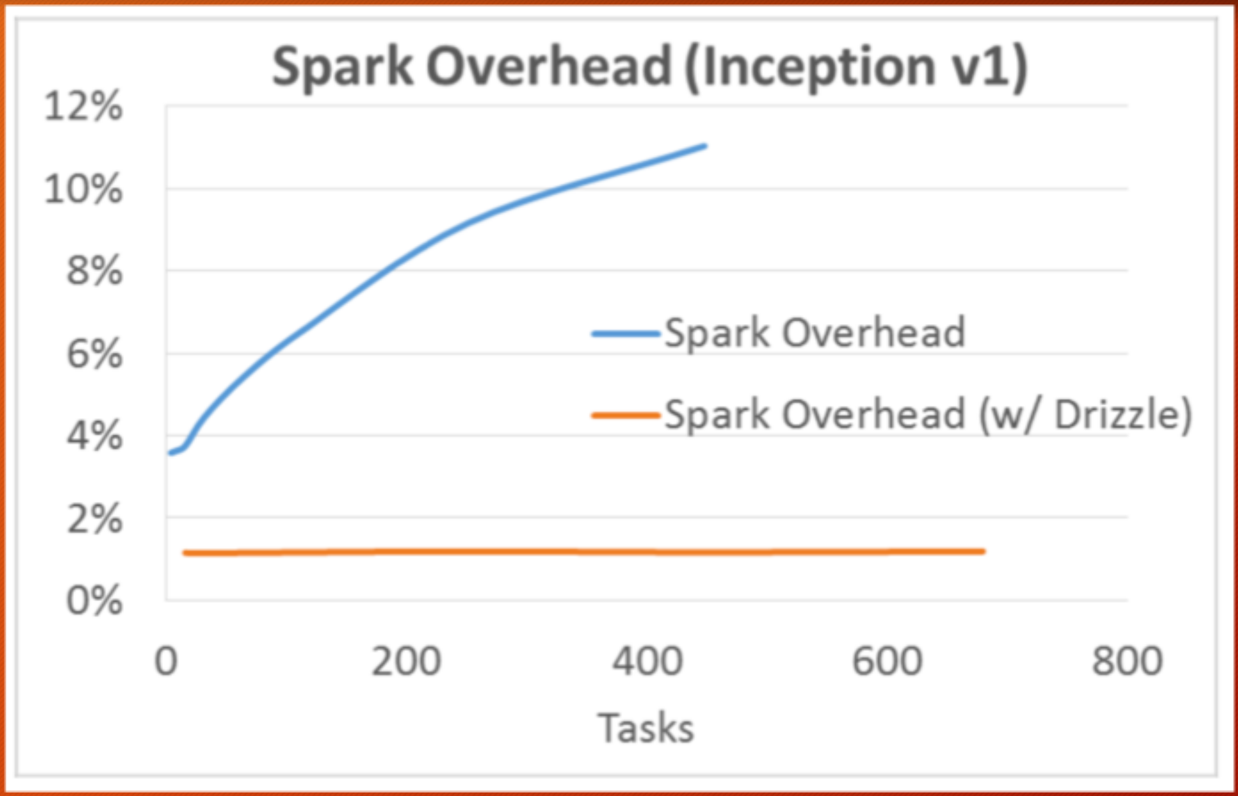
Spark overheads (task scheduling, task serde, task fetch) as a fraction of average compute time for Inception v1 training

BigDL + "Drizzle"

- A low-latency execution engine for Apache Spark, packaged in BigDL
- Fine-grained execution with coarse-grained scheduling
- Group scheduling
  - Scheduling a group of iterations at once
  - Fault tolerance, scheduling at group boundaries
- Coordinating shuffles: **pre-scheduling**
  - Pre-schedule tasks on executors
  - Trigger tasks once dependencies are met



Spark Task Scheduling Overheads, Redux



Drizzle increases mini-batch size

- Distributed synchronous mini-batch SGD
  - Increased mini-batch size
  - **Can lead to loss in test accuracy**

total_batch_size = batch_size_per_worker * num_of_workers

- State-of-art method for scaling mini-batch size
  - Linear scaling rule
  - Warm-up
  - Layer-wise adaptive rate scaling
  - Adding batch normalization

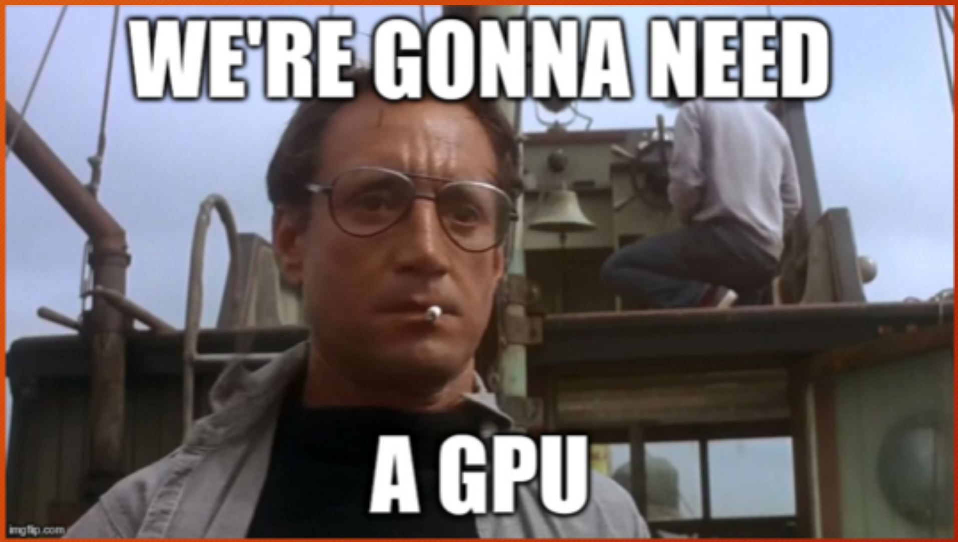
"Accurate, Large Minibatch SGD: Training ImageNet in 1Hour"

"Scaling SGD Batch Size to 32K for ImageNet Training"

Want to get started?

- As easy as *pip install*
  - https://bigdl-project.github.io/master/#PythonUserGuide/install-from-pip/
  - (add to custom Python startup script for Dataproc)
- A few configuration changes to make on Dataproc, but not too bad
  - https://github.com/intel-analytics/BigDL/blob/master/docs/docs/ProgrammingGuide/run-on-dataproc.md

Questions?



References

- https://bigdl-project.github.io/master/#
- https://github.com/intel-analytics/BigDL/

Project Notes

- P1 is due ***Thursday, February 1 at 11:59:59pm.***
  - AutoLab shuts down, and I stop considering changes on GitHub
- P2 will be released on Thursday!
  - AutoLab assignment will show up
  - Teams will be announced on Slack
  - **Due Thursday, February 15 (2 weeks) at 11:59:59pm.**
- P1 Lightning Talks *next Wednesday!*
  - Each team gives a *5-minute overview* of their work (slides please)
  - Highlight the approach you took (theory, engineering, teamwork) and how it paid off (or not)—what worked, what didn't, what you'd keep, what you'd change
  - Teams will be called up randomly, so be ready to go!