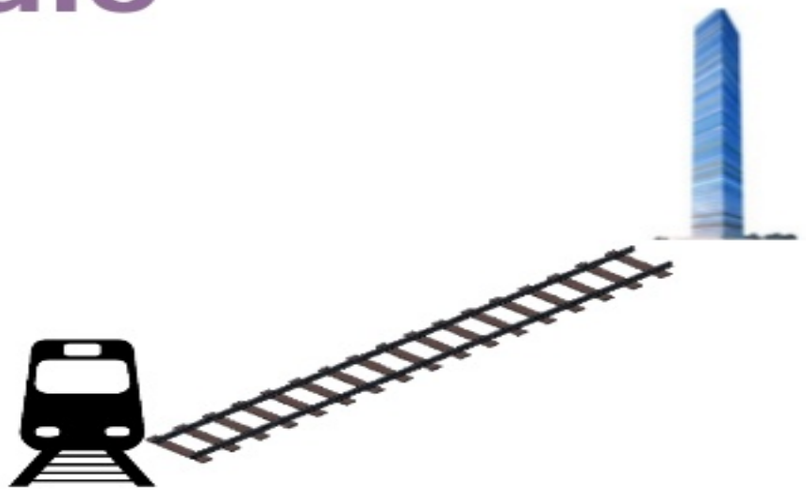
Distributed Deep Learning with Apache Spark and TensorFlow

Jim Dowling, Logical Clocks AB

jim_dowling

#SAISDL2

The Cargobike Riddle



?

Optimizing GPU Resource Utilization

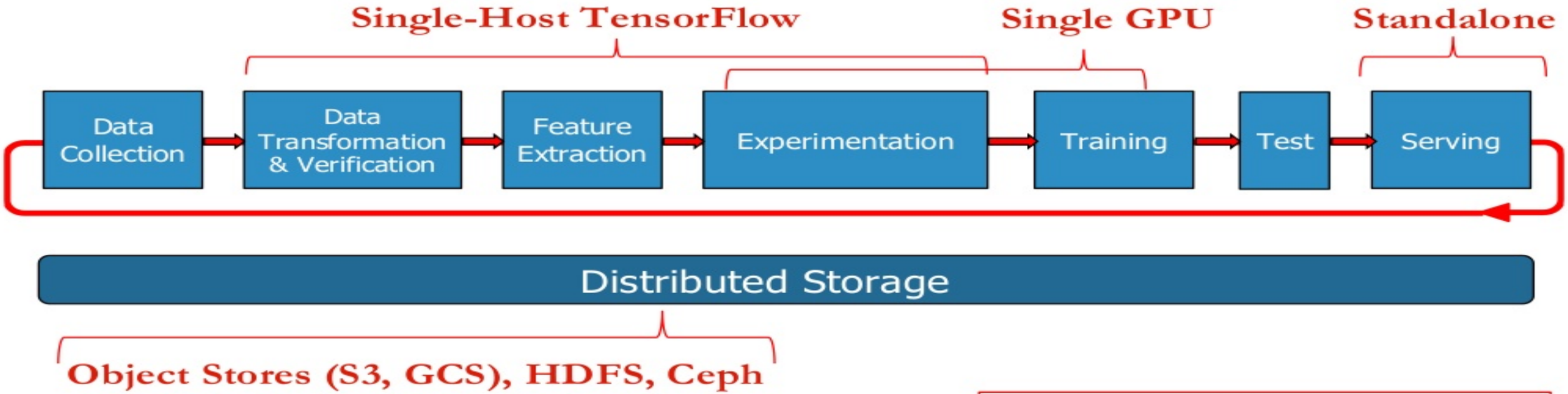
Dynamic Executors
(release GPUs when training finishes)

Spark & TensorFlow

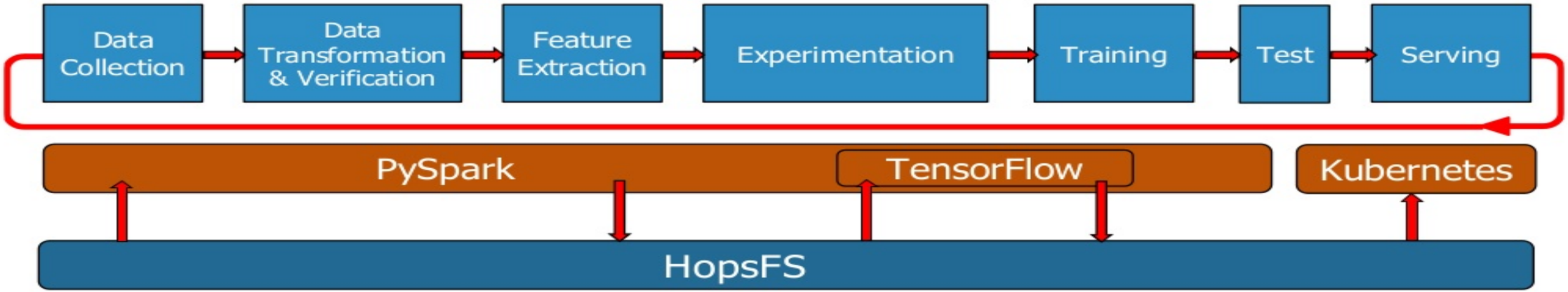
Blacklisting Executors
(for Fault Tolerant Hyperparameter Optimization)

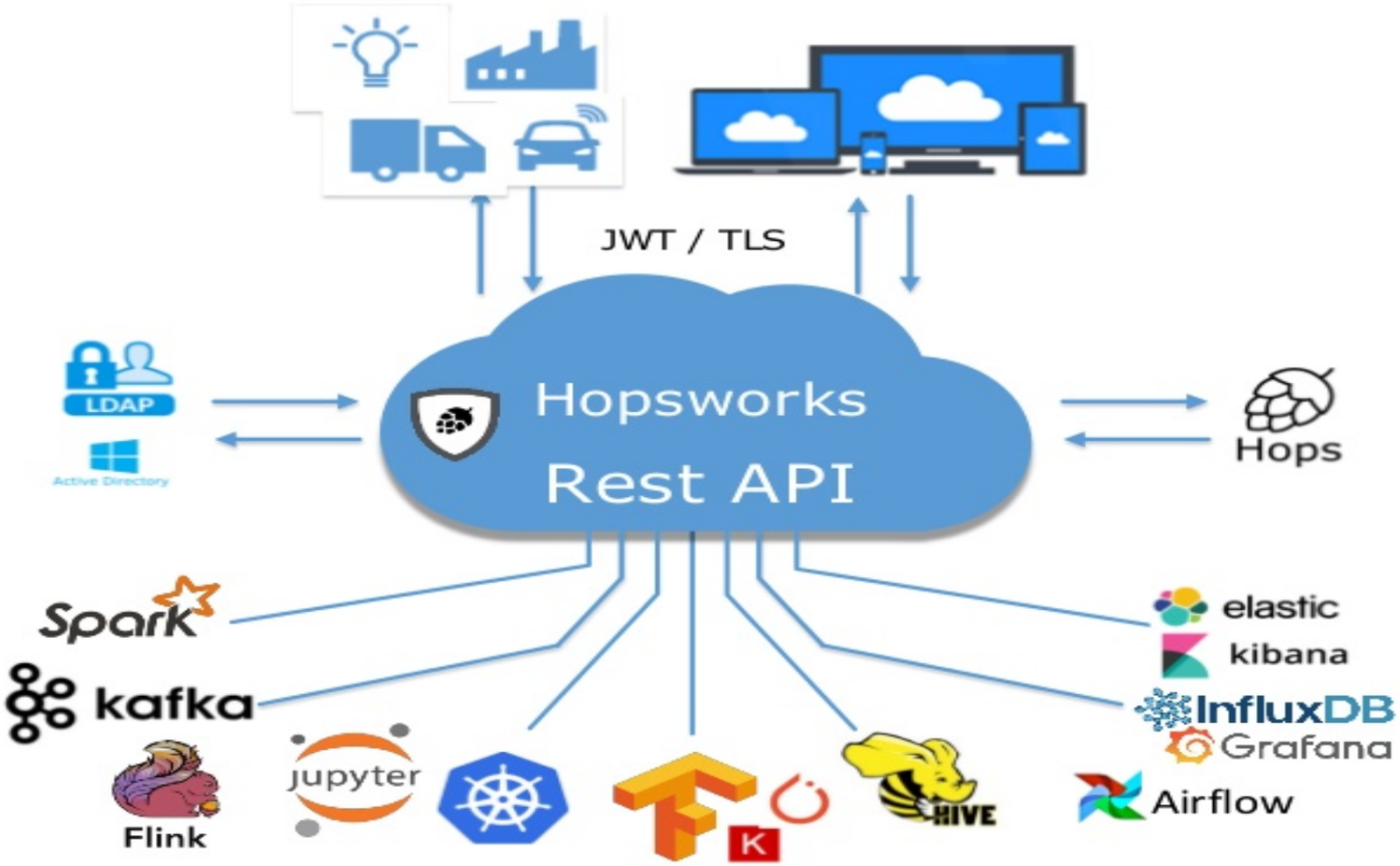
Container (GPUs)

Scalable ML Pipeline

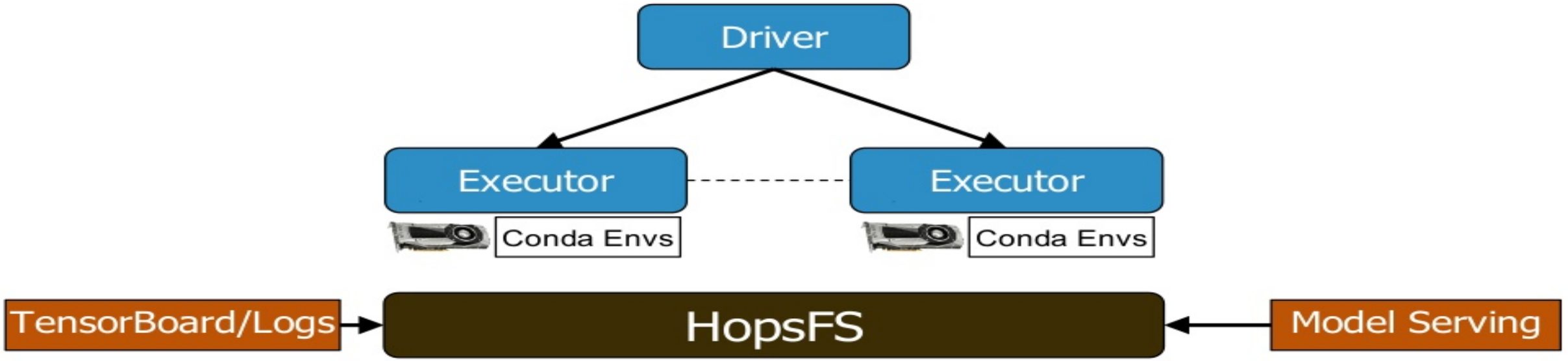


Scalable ML Pipeline (Hopsworks)



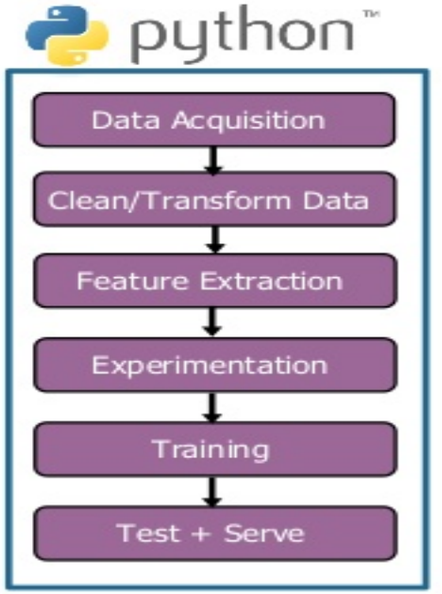

JWT / TLS
LDAP
Active Directory
Hopsworks
Rest API
Hops
Spark
kafka
Flink
jupyter
elastic
kibana
InfluxDB
Grafana
Airflow
HIVE

Spark/TensorFlow in Hopsworks



HopsML

- Experiments
  - Dist. Hyperparameter Optimization
  - Versioning of Models/Code/Resources
  - Visualization with Tensorboard
  - Distributed Training with checkpointing
- [Feature Store]
- Model Serving and Monitoring



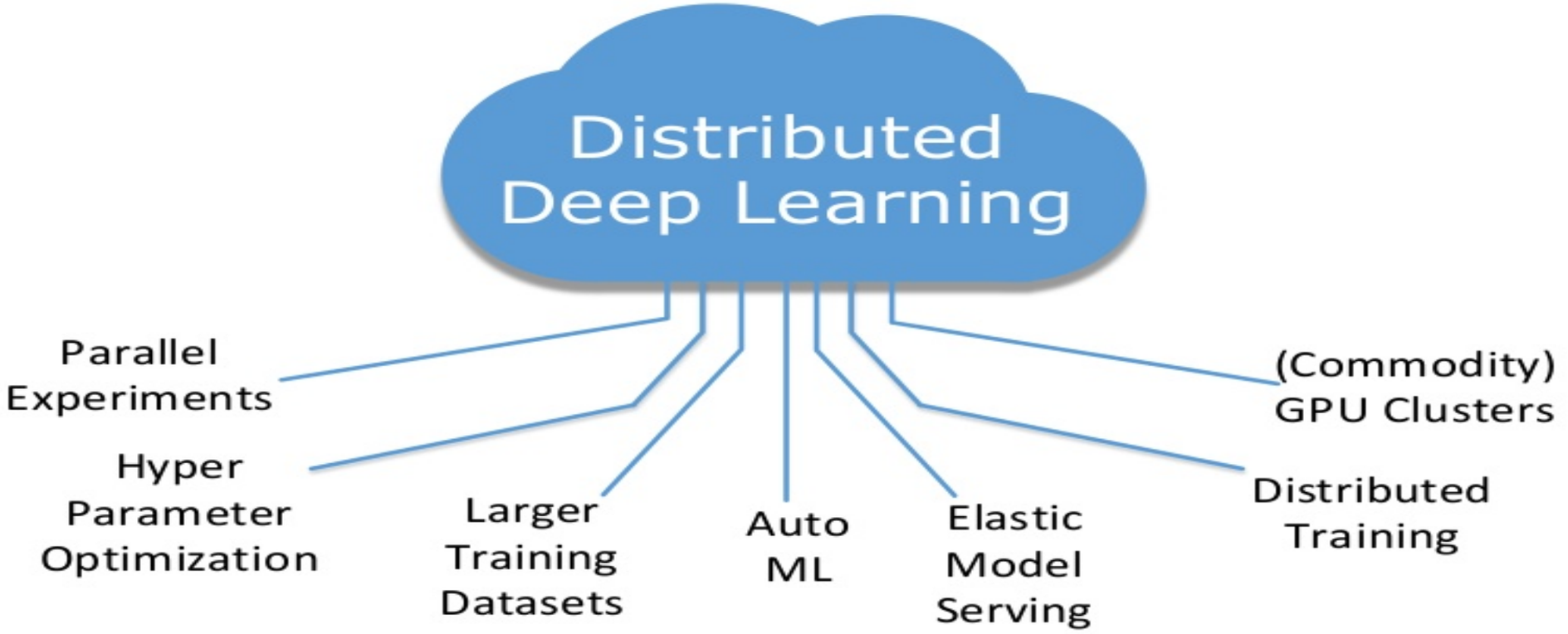
Why Distributed Deep Learning?

Prof Nando de Freitas @NandoDF

"ICLR 2019 lessons thus far: The deep neural nets have to be BIGGER and they're hungry for data, memory and compute."

https://twitter.com/NandoDF/status/1046371764211716096

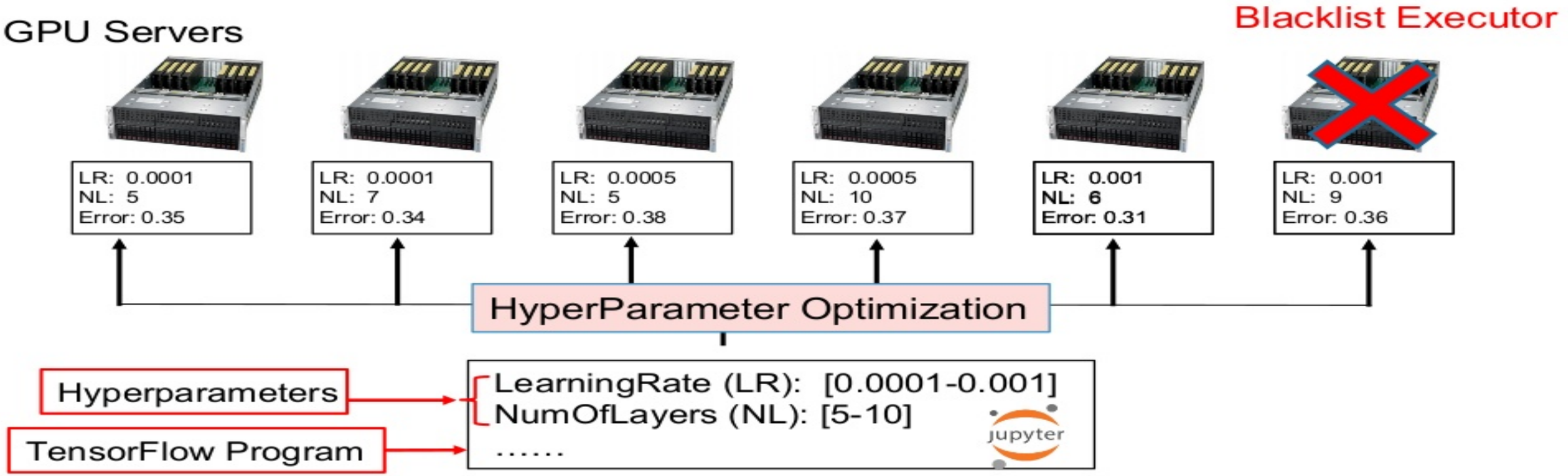
All Roads Lead to Distribution



(Because DL Theory Sucks!)

Hyperparameter Optimization

Faster Experimentation



Declarative or API Approach?

- Declarative Hyperparameters in external files
  - Vizier/CloudML (yaml)
  - Sagemaker (json)*
- API-Driven Notebook-Friendly
  - Databrick's MLFlow
  - HopsML

*https://docs.aws.amazon.com/sagemaker/latest/dg/automatic-model-tuning-define-ranges.html

GridSearch for Hyperparameters on HopsML

```
def train(learning_rate, dropout):

  [TensorFlow Code here]

args_dict = {'learning_rate': [0.001, 0.005, 0.01],
             'dropout': [0.5, 0.6]}
experiment.launch(train, args_dict)
```

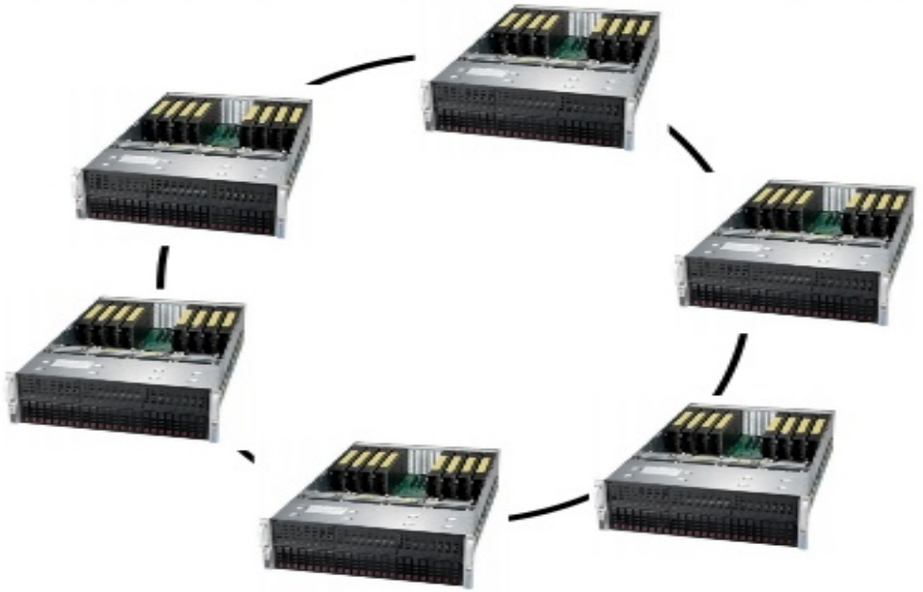
Dynamic Executors, Blacklisting

Launch 6 Spark Executors

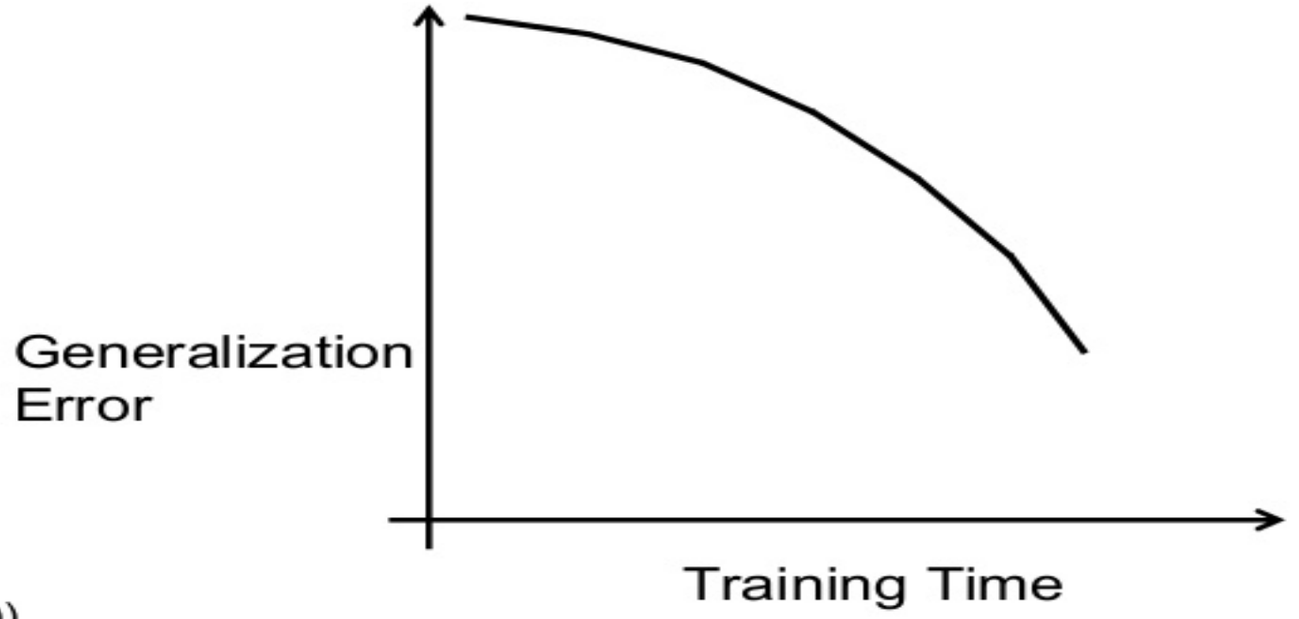
Image from @hardmaru on Twitter.

Distributed Training

Data Parallel Distributed Training

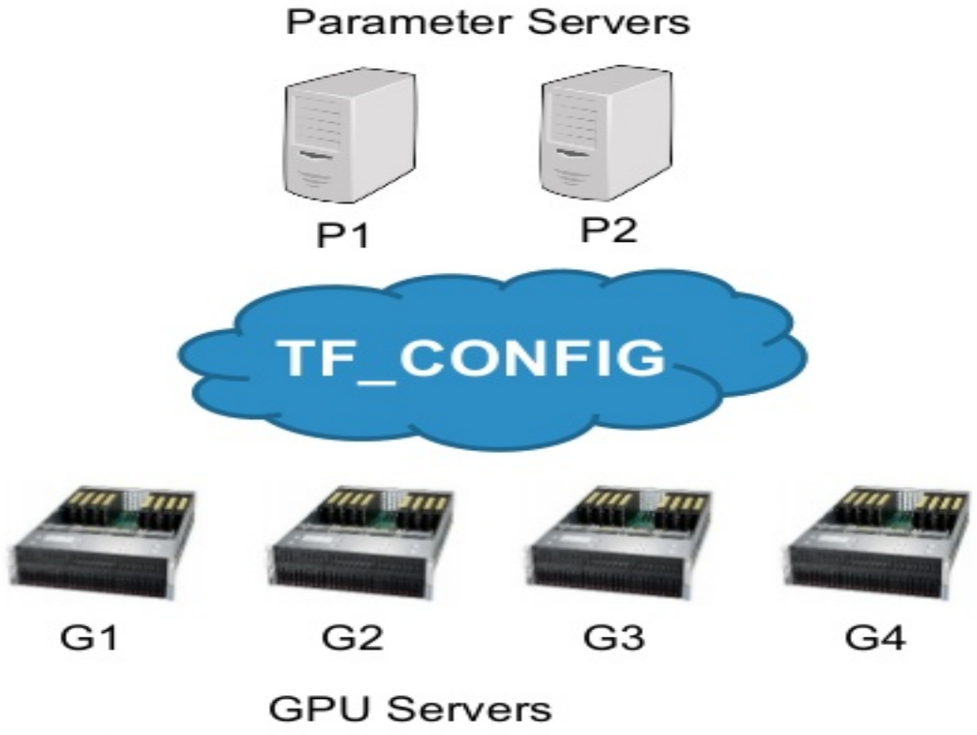


(Synchronous Stochastic Gradient Descent (SGD))



Frameworks for Distributed Training

Distributed TensorFlow / TfOnSpark

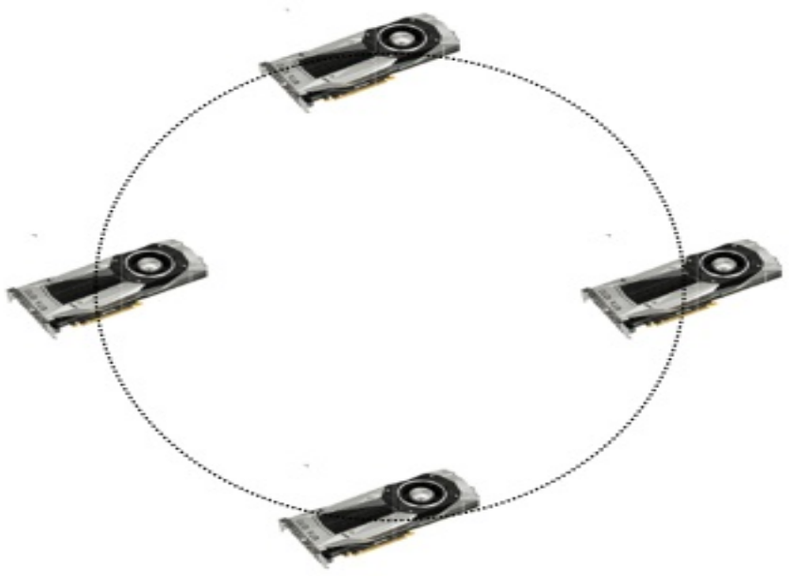


TF_CONFIG

Bring your own Distribution!

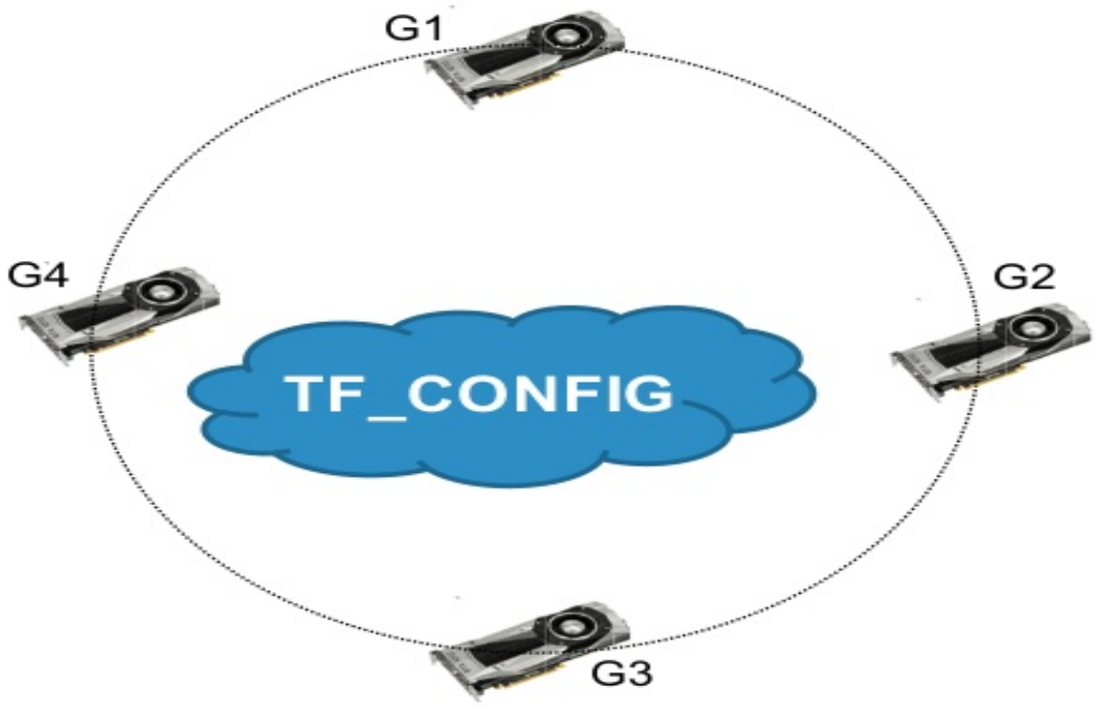
1. Start all processes for P1,P2, G1-G4 yourself
2. Enter all IP addresses in TF_CONFIG along with GPU device IDs.

Horovod



- Bandwidth optimal
- Builds the Ring, runs AllReduce using MPI and NCCL2
- Available in
  - Hopsworks
  - Databricks (Spark 2.4)

Tf CollectiveAllReduceStrategy



TF_CONFIG, again.

Bring your own Distribution!

1. Start all processes for G1-G4 yourself
2. Enter all IP addresses in TF_CONFIG along with GPU device IDs.

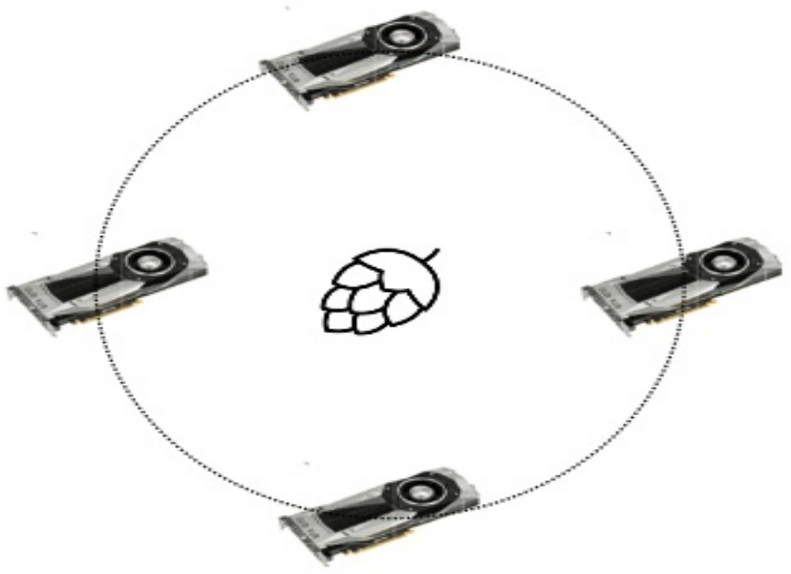
Available from TensorFlow 1.11

Tf CollectiveAllReduceStrategy Gotchas

- Specify GPU order in the ring statically
  - *gpu_indices*
- Configure the batch size for merging tensors
  - *allreduce_merge_scope*
    - Set to ‘1’ for no merging
    - Set to ’32’ for higher throughput.*

* https://groups.google.com/a/tensorflow.org/forum/#!topic/discuss/7T05tNV08Us

HopsML CollectiveAllReduceStrategy



- Uses Spark/YARN to add distribution to TensorFlow's CollectiveAllReduceStrategy
  - Automatically builds the ring (Spark/YARN)
  - Allocates GPUs to Spark Executors

https://github.com/logicalclocks/hops-examples/tree/master/tensorflow/notebooks/Distributed_Training

CollectiveAllReduce vs Horovod Benchmark

TensorFlow: 1.11
Model: **Inception v1**
Dataset: imagenet (synthetic)
Batch size: 256 global, 32.0 per device
Num batches: 100
Optimizer Momemtum
Num GPUs: 8
AllReduce: **collective**

Step	Img/sec	total_loss
1	images/sec: 2972.4 +/- 0.0	
10	images/sec: 3008.9 +/- 8.9	
100	images/sec: 2998.6 +/- 4.3	

total images/sec: **2993.52**

TensorFlow: 1.7
Model: **Inception v1**
Dataset: imagenet (synthetic)
Batch size: 256 global, 32.0 per device
Num batches 100
Optimizer Momemtum
Num GPUs: 8
AllReduce: **horovod**

Step	Img/sec	total_loss
1	images/sec: 2816.6 +/- 0.0	
10	images/sec: 2808.0 +/- 10.8	
100	images/sec: 2806.9 +/- 3.9	

total images/sec: **2803.69**

Small Model

https://groups.google.com/a/tensorflow.org/forum/#!topic/discuss/7T05tNV08Us

CollectiveAllReduce vs Horovod Benchmark

TensorFlow: 1.11
Model: **VGG19**
Dataset: imagenet (synthetic)
Batch size: 256 global, 32.0 per device
Num batches: 100
Optimizer Momemtum
Num GPUs: 8
AllReduce: **collective**

Step	Img/sec	total_loss
1	images/sec: 634.4 +/- 0.0	
10	images/sec: 635.2 +/- 0.8	
100	images/sec: 635.0 +/- 0.5	

total images/sec: **634.80**

TensorFlow: 1.7
Model: **VGG19**
Dataset: imagenet (synthetic)
Batch size: 256 global, 32.0 per device
Num batches 100
Optimizer Momemtum
Num GPUs: 8
AllReduce: **horovod**

Step	Img/sec	total_loss
1	images/sec: 583.01 +/- 0.0	
10	images/sec: 582.22 +/- 0.1	
100	images/sec: 583.61 +/- 0.2	

total images/sec: **583.61**

Big Model

https://groups.google.com/a/tensorflow.org/forum/#!topic/discuss/7T05tNV08Us

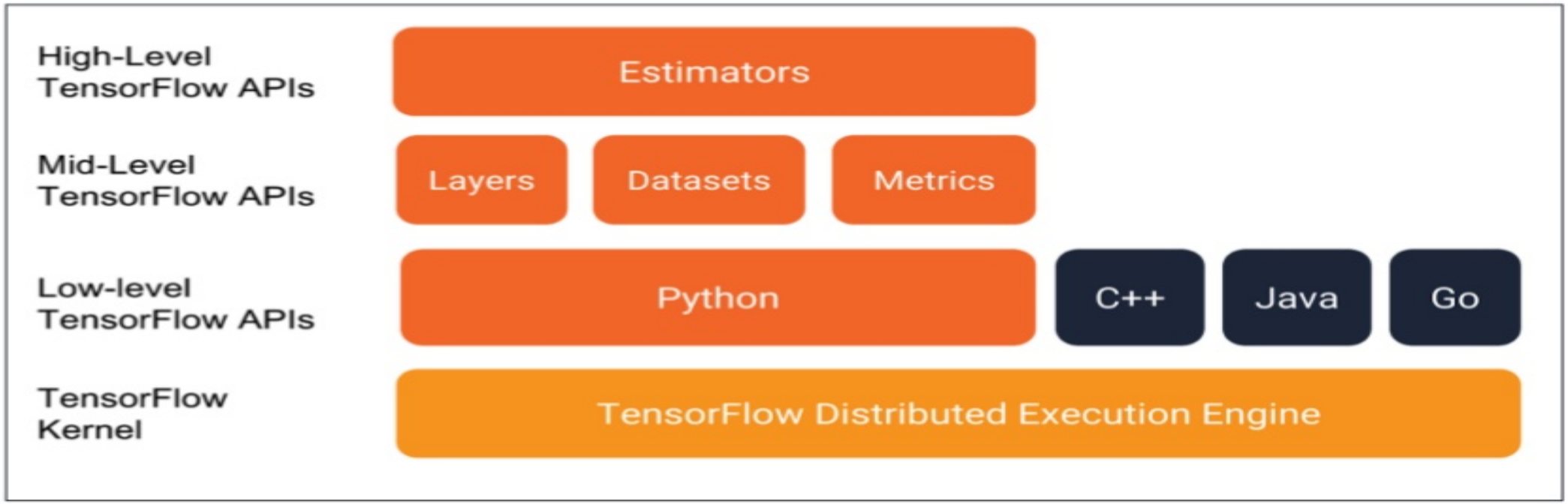
Reduction in LoC for Dist Training

Released	Framework	Lines of Code in Hops
March 2016	DistributedTensorFlow	~1000
Feb 2017	TensorFlowOnSpark*	~900
Jan 2018	Horovod (Keras)*	~130
June 2018	Databricks' HorovodEstimator	~100
Sep 2018	HopsML (Keras/CollectiveAllReduce)*	~100

*https://github.com/logicalclocks/hops-examples

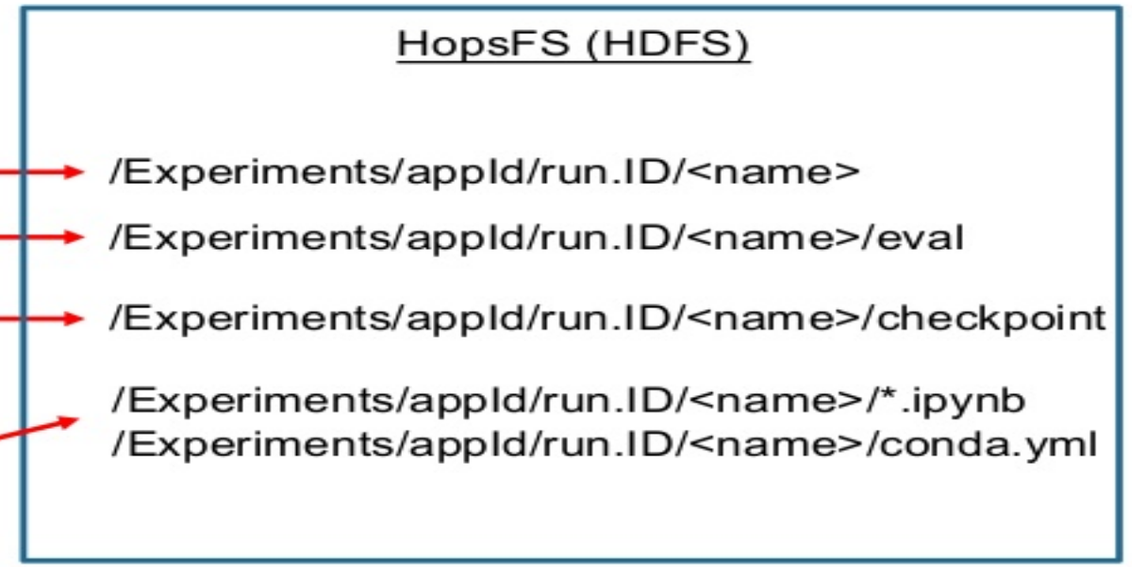
**https://docs.azuredatabricks.net/_static/notebooks/horovod-estimator.html

Estimator APIs in TensorFlow



Estimators log to the Distributed Filesystem

```
tf.estimator.RunConfig(
'CollectiveAllReduceStrategy'
model_dir
tensorboard_logs
checkpoints
)


experiment.launch(…)
```



HopsML CollectiveAllReduceStrategy with Keras

```
def distributed_training():
  def input_fn(): # return dataset
  model = …
  optimizer = …
  model.compile(…)
  rc = tf.estimator.RunConfig('CollectiveAllReduceStrategy')
  keras_estimator = tf.keras.estimator.model_to_estimator(….)
  tf.estimator.train_and_evaluate(keras_estimator, input_fn)

experiment.allreduce(distributed_training)
```

Add Tensorboard Support

```
def distributed_training():
  from hops import tensorboard
  model_dir = tensorboard.logdir()
  def input_fn(): # return dataset
  model = …
  optimizer = …
  model.compile(…)
  rc = tf.estimator.RunConfig('CollectiveAllReduceStrategy')
  keras_estimator = keras.model_to_estimator(model_dir)
  tf.estimator.train_and_evaluate(keras_estimator, input_fn)

experiment.allreduce(distributed_training)
```

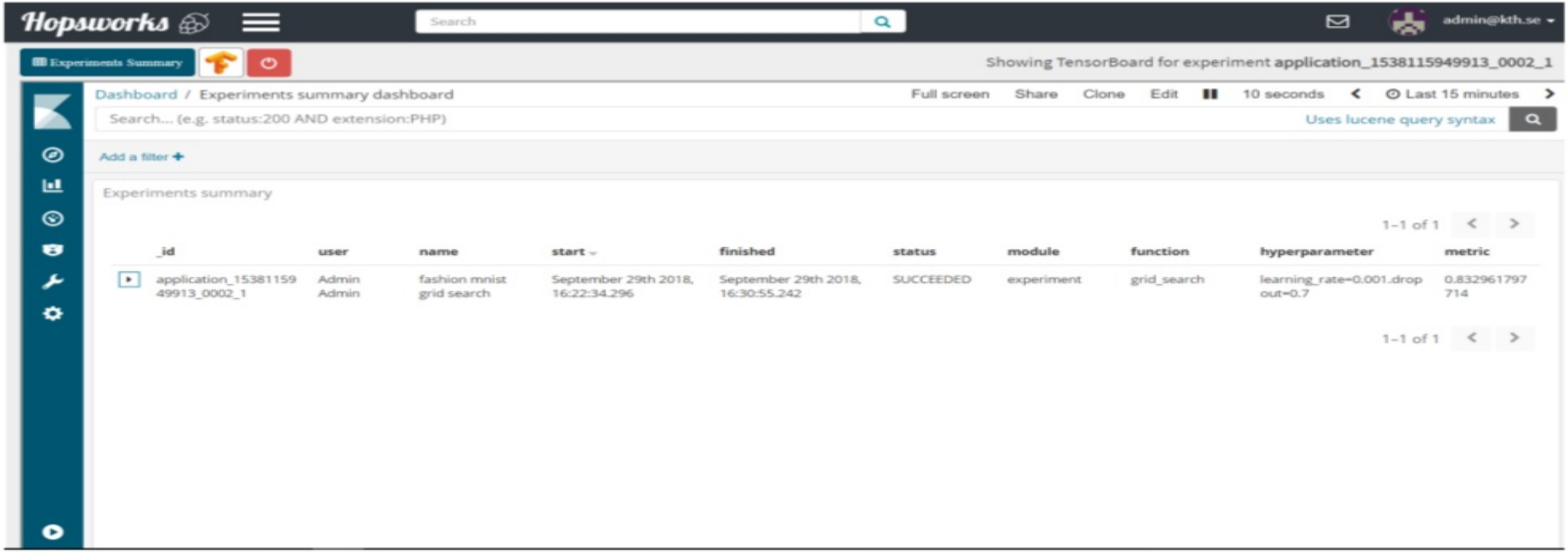
GPU Device Awareness

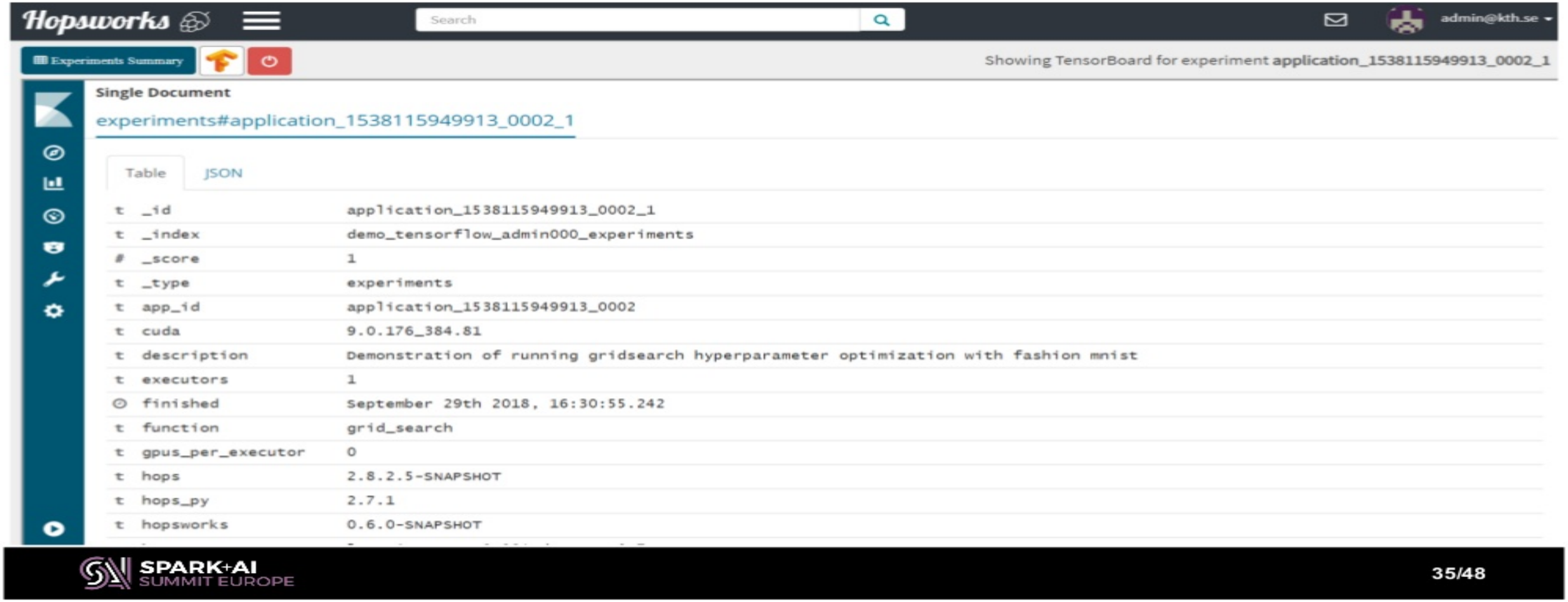
```
def distributed_training():
  from hops import devices
  def input_fn(): # return dataset
  model = …
  optimizer = …
  model.compile(…)
  est.RunConfig(num_gpus_per_worker=devices.get_num_gpus())
  keras_estimator = keras.model_to_estimator(…)
  tf.estimator.train_and_evaluate(keras_estimator, input_fn)

experiment.allreduce(distributed_training)
```

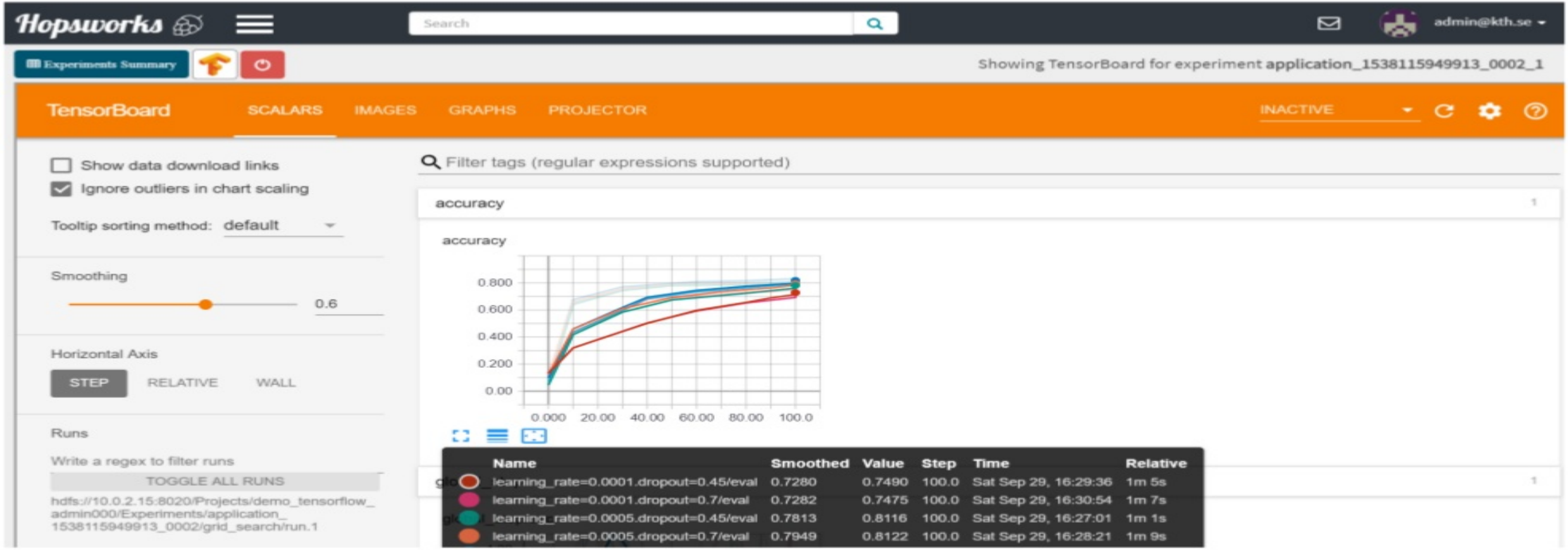
Experiment Versioning (.ipynb, conda, results)

```
def distributed_training():
  def input_fn(): # return dataset
  model = …
  optimizer = …
  model.compile(…)
  rc = tf.estimator.RunConfig('CollectiveAllReduceStrategy')
  keras_estimator = keras.model_to_estimator(…)
  tf.estimator.train_and_evaluate(keras_estimator, input_fn)

notebook = hdfs.project_path()+'/Jupyter/Experiment/inc.ipynb'
experiment.allreduce(distributed_training, name='inception',
   description='A inception example with hidden layers',
   versioned_resources=[notebook])
```

Experiments/Versioning in Hopsworks

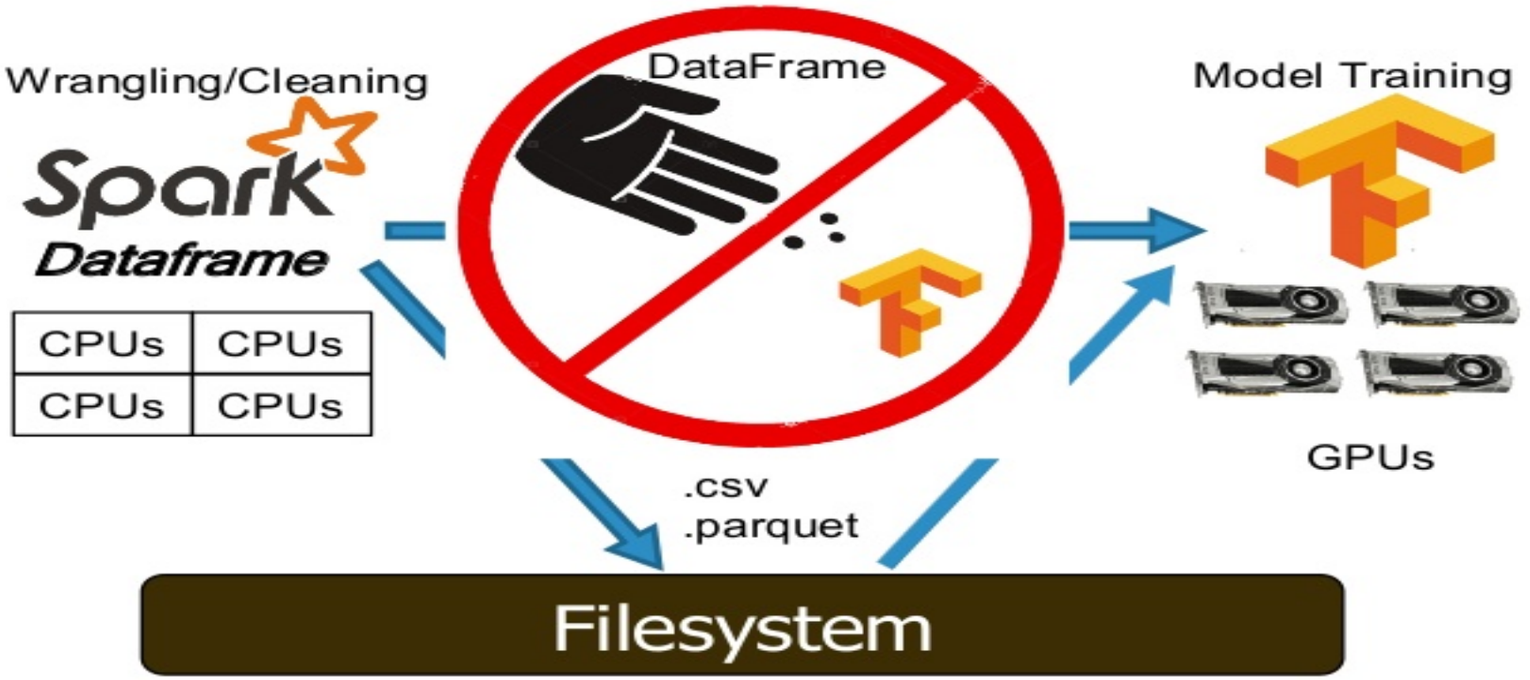

Hopsworks
Search
admin@kth.se
Experiments Summary
Showing TensorBoard for experiment application_1538115949913_0002_1
Dashboard / Experiments summary dashboard
Full screen
Share
Clone
Edit
10 seconds
Last 15 minutes
Search... (e.g. status:200 AND extension:PHP)
Uses lucene query syntax
Add a filter +
Experiments summary
1–1 of 1
_id
user
name
start
finished
status
module
function
hyperparameter
metric
application_1538115949913_0002_1
Admin Admin
fashion mnist grid search
September 29th 2018, 16:22:34.296
September 29th 2018, 16:30:55.242
SUCCEEDED
experiment
grid_search
learning_rate=0.001,dropout=0.7
0.8329617977714
1–1 of 1


Hopsworks
Search
admin@kth.se
Experiments Summary
Showing TensorBoard for experiment application_1538115949913_0002_1
Single Document
experiments#application_1538115949913_0002_1
Table
JSON
t _id application_1538115949913_0002_1
t _index demo_tensorflow_admin000_experiments
_score 1
t _type experiments
t app_id application_1538115949913_0002
t cuda 9.0.176_384.81
t description Demonstration of running gridsearch hyperparameter optimization with fashion mnist
t executors 1
finished September 29th 2018, 16:30:55.242
t function grid_search
t gpus_per_executor 0
t hops 2.8.2.5-SNAPSHOT
t hops_py 2.7.1
t hopsworks 0.6.0-SNAPSHOT


Hopsworks
Search
admin@kth.se
Experiments Summary
Showing TensorBoard for experiment application_1538115949913_0002_1
TensorBoard
SCALARS
IMAGES
GRAPHS
PROJECTOR
INACTIVE
Show data download links
Ignore outliers in chart scaling
Tooltip sorting method: default
Smoothing
0.6
Horizontal Axis
STEP
RELATIVE
WALL
Runs
Write a regex to filter runs
TOGGLE ALL RUNS
hdfs://10.0.2.15:8020/Projects/demo_tensorflow_admin000/Experiments/application_1538115949913_0002/grid_search/run.1
Filter tags (regular expressions supported)
accuracy
accuracy
0.800
0.600
0.400
0.200
0.00
0.000
20.00
40.00
60.00
80.00
100.0
Name
Smoothed
Value
Step
Time
Relative
learning_rate=0.0001.dropout=0.45/eval
0.7280
0.7490
100.0
Sat Sep 29, 16:29:36
1m 5s
learning_rate=0.0001.dropout=0.7/eval
0.7282
0.7475
100.0
Sat Sep 29, 16:30:54
1m 7s
learning_rate=0.0005.dropout=0.45/eval
0.7813
0.8116
100.0
Sat Sep 29, 16:27:01
1m 1s
learning_rate=0.0005.dropout=0.7/eval
0.7949
0.8122
100.0
Sat Sep 29, 16:28:21
1m 9s

The Data Layer (Foundations)

Feeding Data to TensorFlow



Project Hydrogen: Barrier Execution mode in Spark: JIRA: SPARK-24374, SPARK-24723, SPARK-24579

Filesystems are not good enough

Uber on Petastorm:

"[Using files] is hard to implement at large scale, especially using modern distributed file systems such as HDFS and S3 (these systems are typically optimized for fast reads of large chunks of data)."

https://eng.uber.com/petastorm/

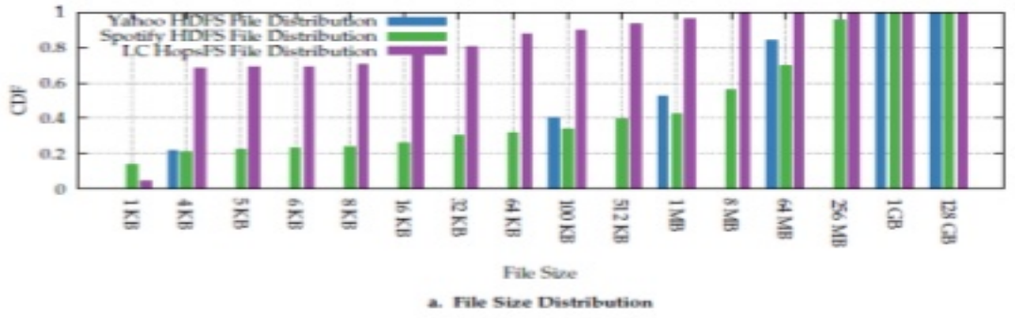
PetaStorm: Read Parquet directly into TensorFlow

```
with Reader('hdfs://myhadoop/dataset.parquet') as reader:
    dataset = make_petastorm_dataset(reader)
    iterator = dataset.make_one_shot_iterator()
    tensor = iterator.get_next()
    with tf.Session() as sess:
        sample = sess.run(tensor)
        print(sample.id)
```

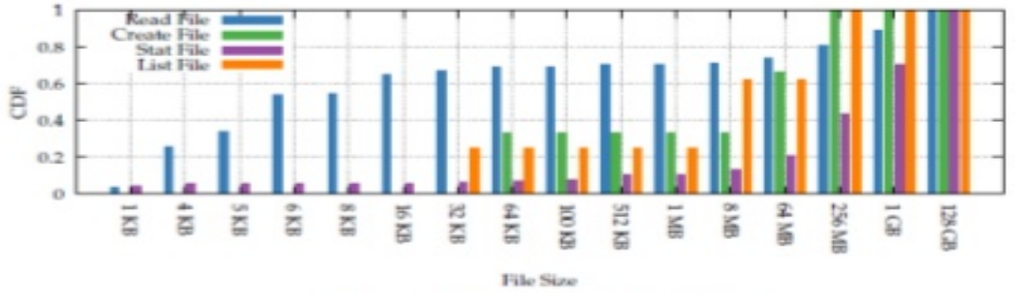
NVMe Disks – Game Changer

- HDFS (and S3) are designed around large blocks (optimized to overcome slow random I/O on disks), while new **NVMe hardware supports orders of magnitude faster random disk I/O**.
- Can we support faster random disk I/O with HDFS?
  - Yes with HopsFS.

Small files on NVMe



a. File Size Distribution



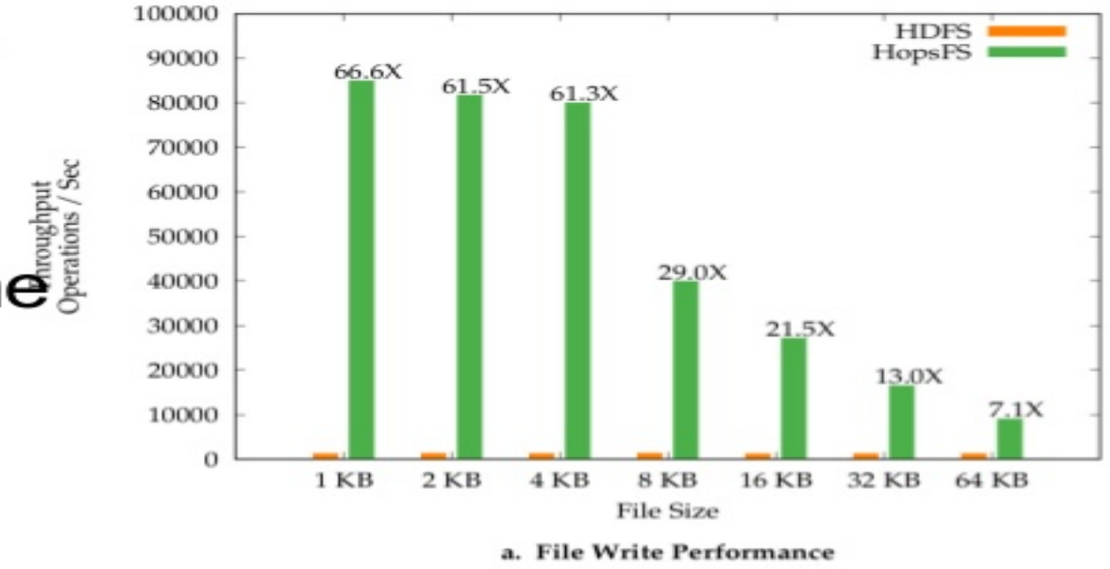
c. Breakdown of File System Operations at Spotify

- At Spotify's HDFS:
  - 33% of files < 64KB in size
  - 42% of operations are on files < 16KB in size

*Size Matters: Improving the Performance of Small Files in Hadoop, Middleware 2018. Niazi et al

HopsFS – NVMe Performance

- HDFS with Distributed Metadata
  - Winner IEEE Scale Prize 2017
- Small files stored replicated in the metadata layer on NVMe disks*
  - Read 10s of 1000s of images/second from HopsFS

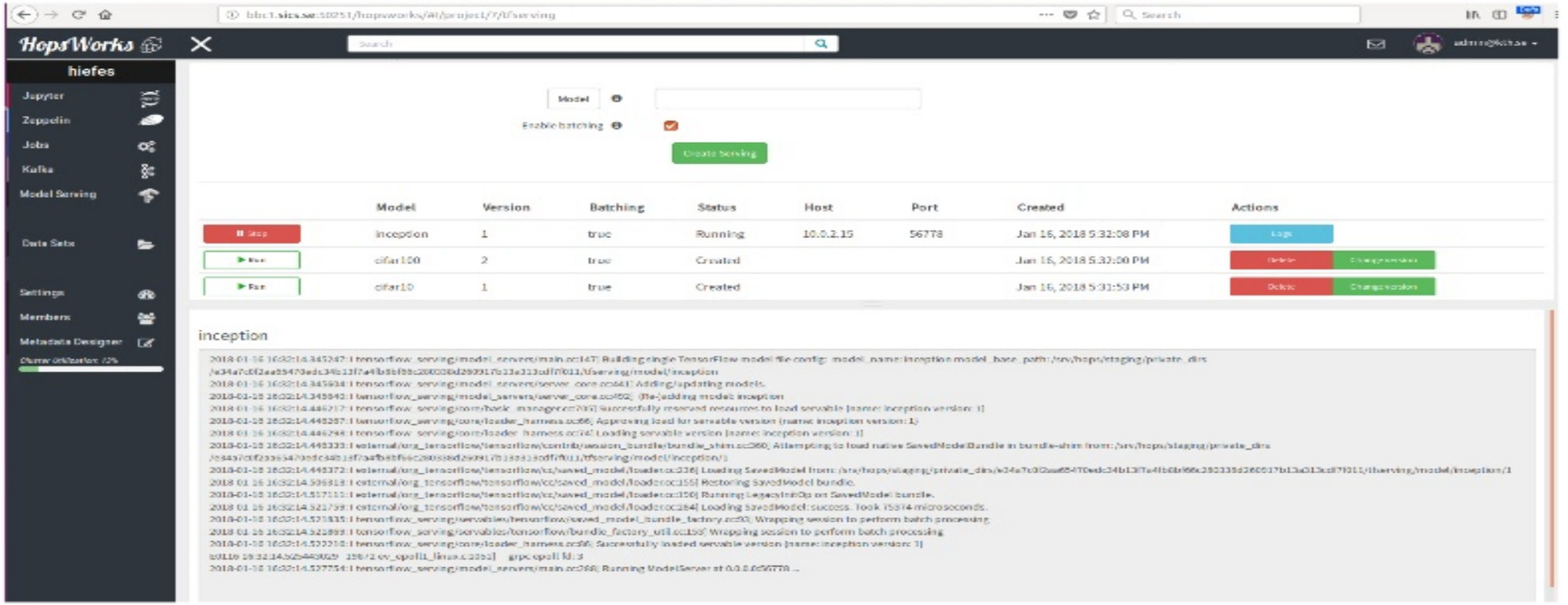


a. File Write Performance

*Size Matters: Improving the Performance of Small Files in Hadoop, Middleware 2018. Niazi et al

Model Serving

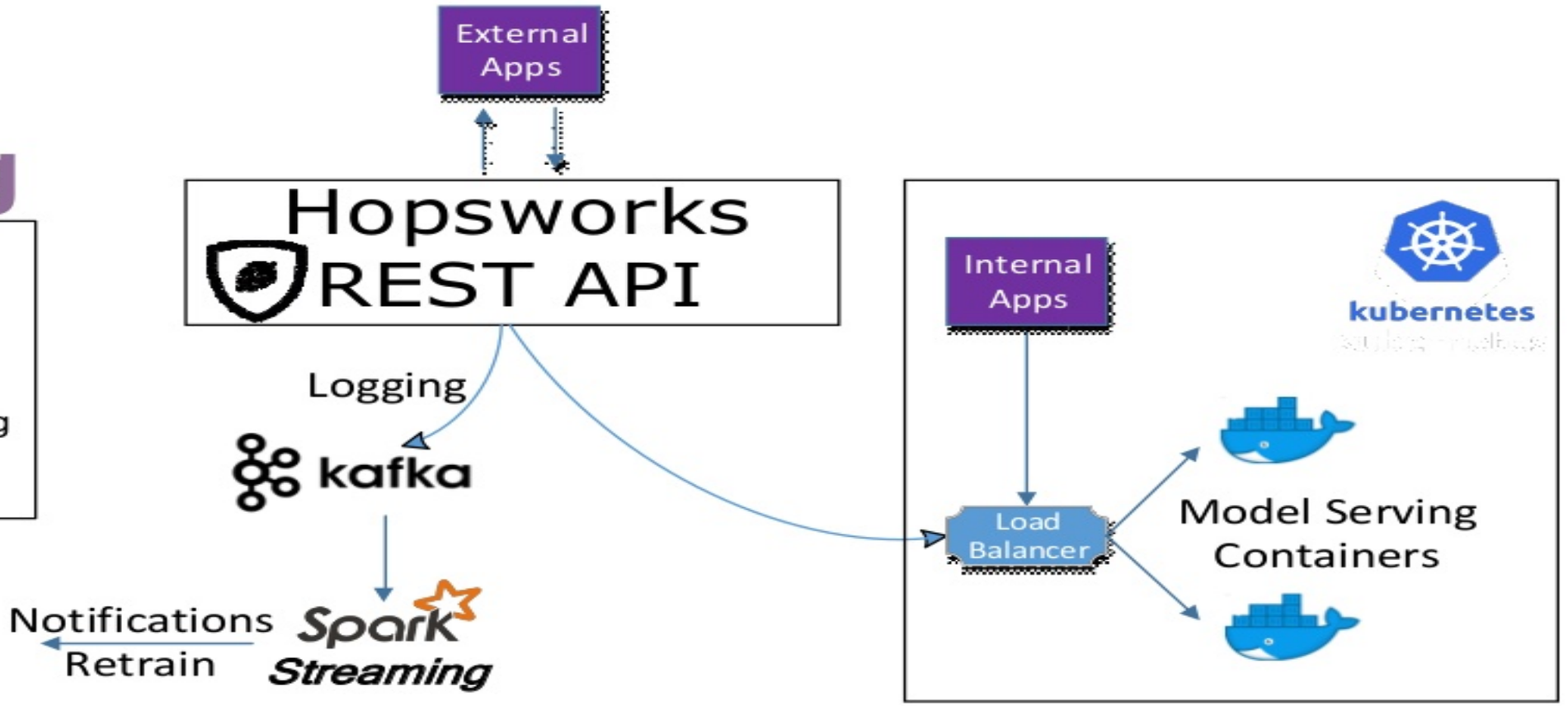
Model Serving on Kubernetes



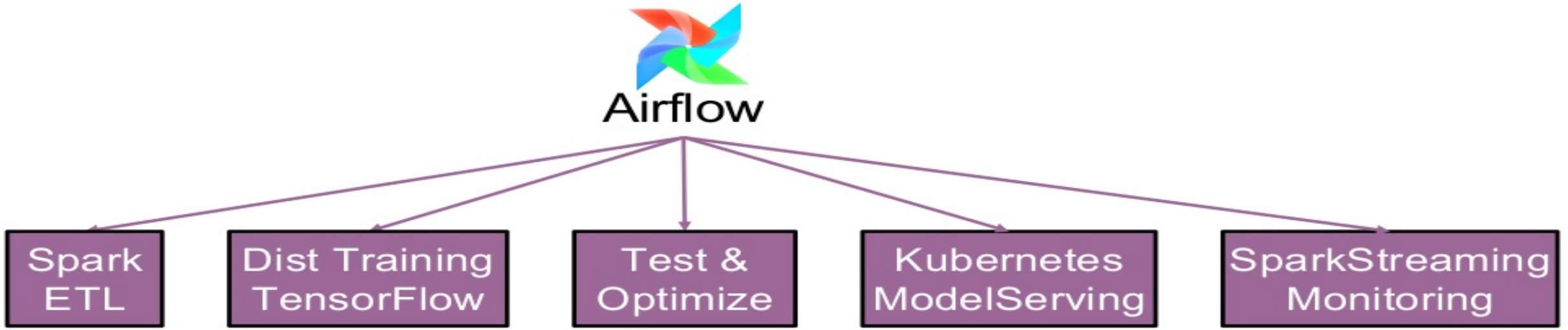
Model Serving

Features:
- Canary
- Multiple Models
- Scale-Out/In

Frameworks:
- ✓ TensorFlow Serving
- ✓ MLeap for Spark
- ✓ scikit-learn



Orchestrating ML Pipelines with Airflow



Summary

- The future of Deep Learning is Distributed
  https://www.oreilly.com/ideas/distributed-tensorflow

- Hops is a new Data Platform with first-class support for Python / Deep Learning / ML / Data Governance / GPUs

hopshadoop

www.hops.io

logicalclocks

www.logicalclocks.com

LOGICAL CLOCKS