BERLIN BUZZWORDS 2018 JUNE 10-12

LOGICAL CLOCKS

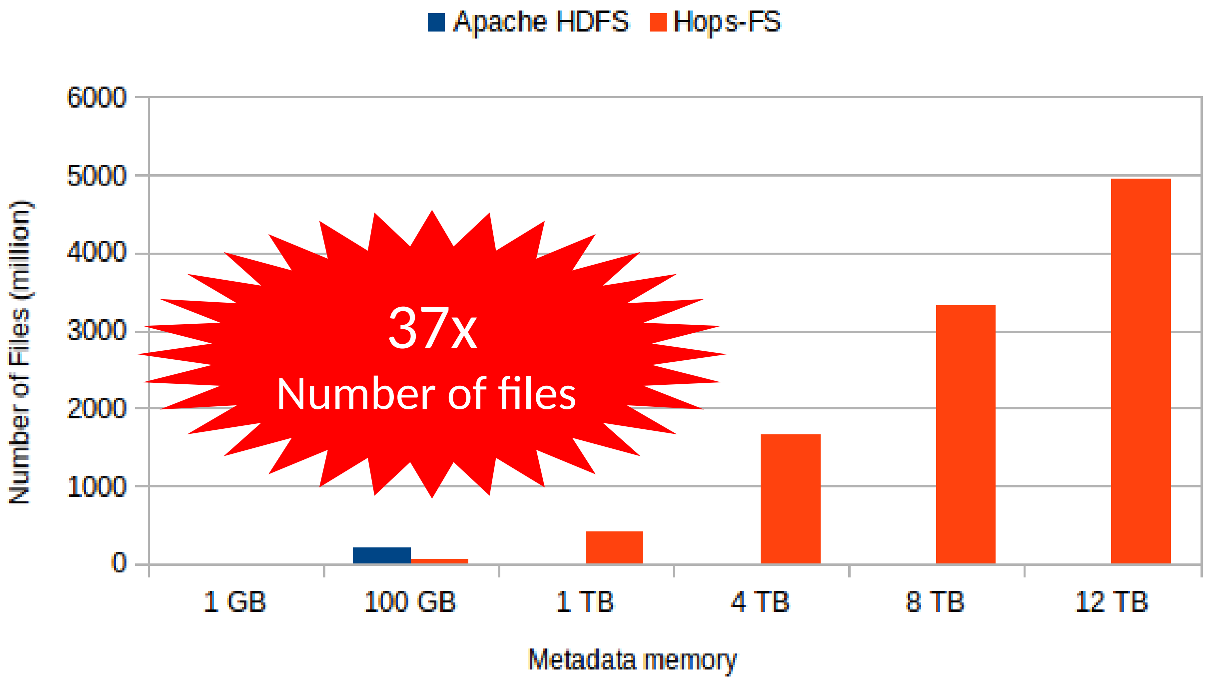
TensorFlow-on-Hops: Hotdog or Not with Toppings

jim_dowling

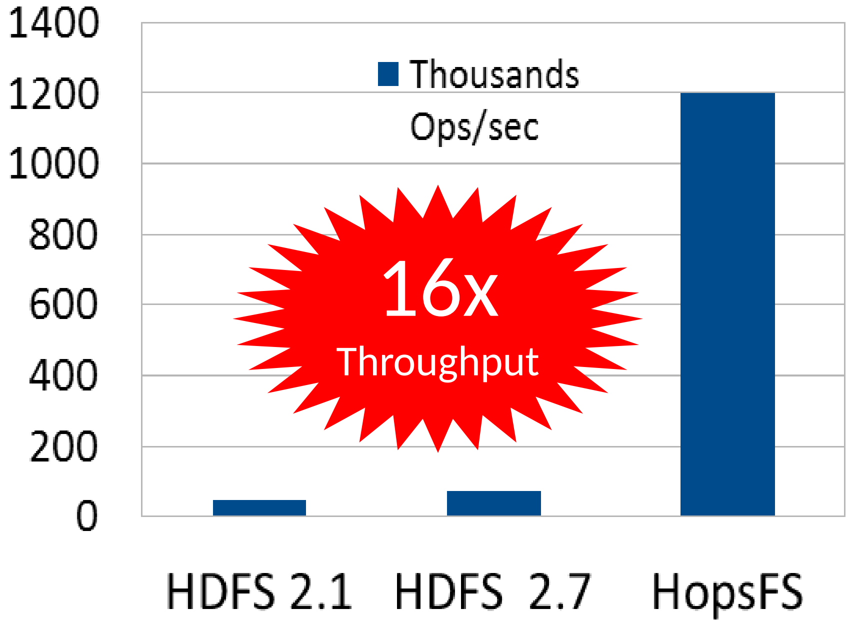
Register an account at:

http://hops.io/tf

World’s Fastest/Biggest Hadoop*



Bigger



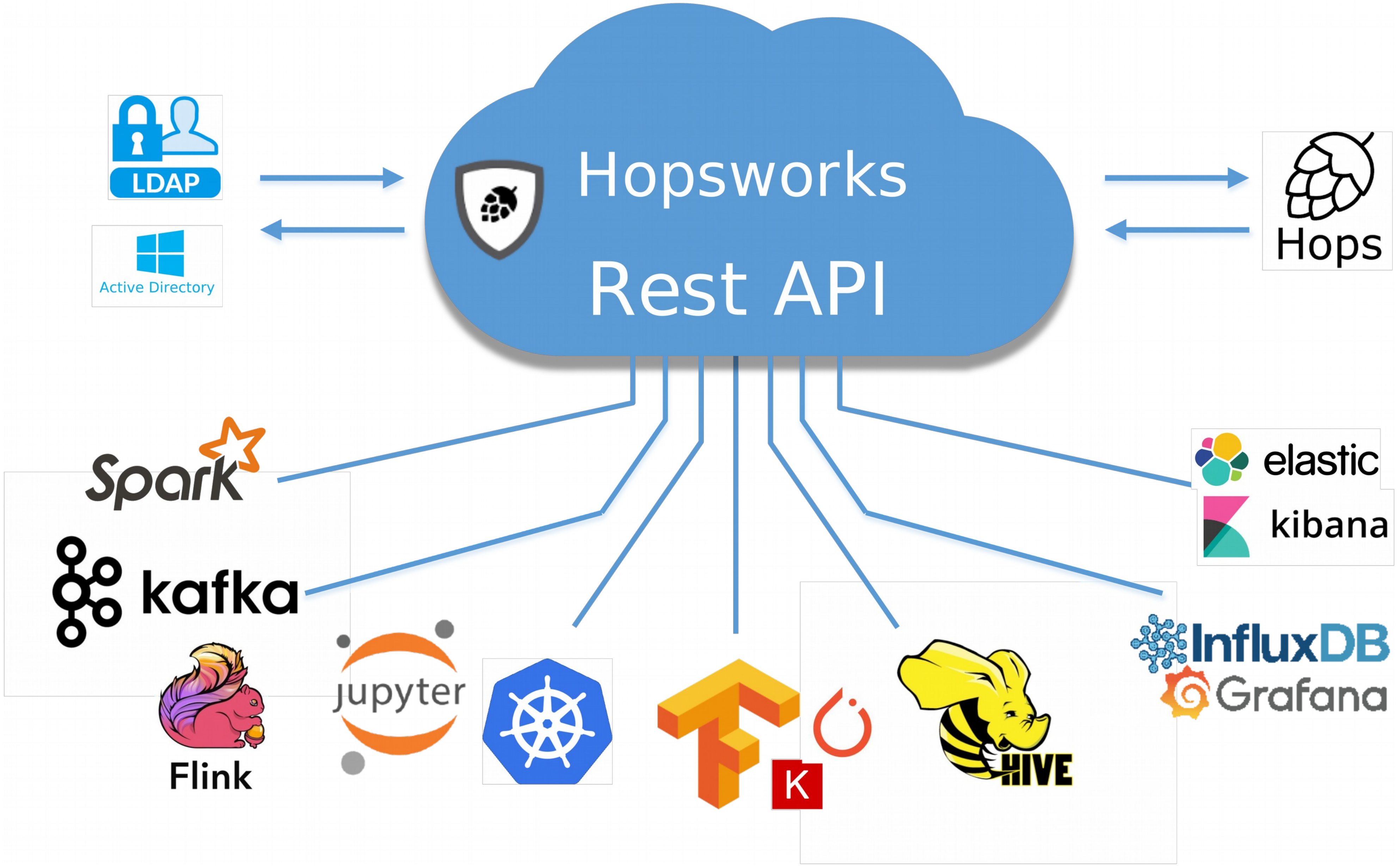
Faster – 1.2M ops/s

IEEE Scale Challenge Winner (2017)

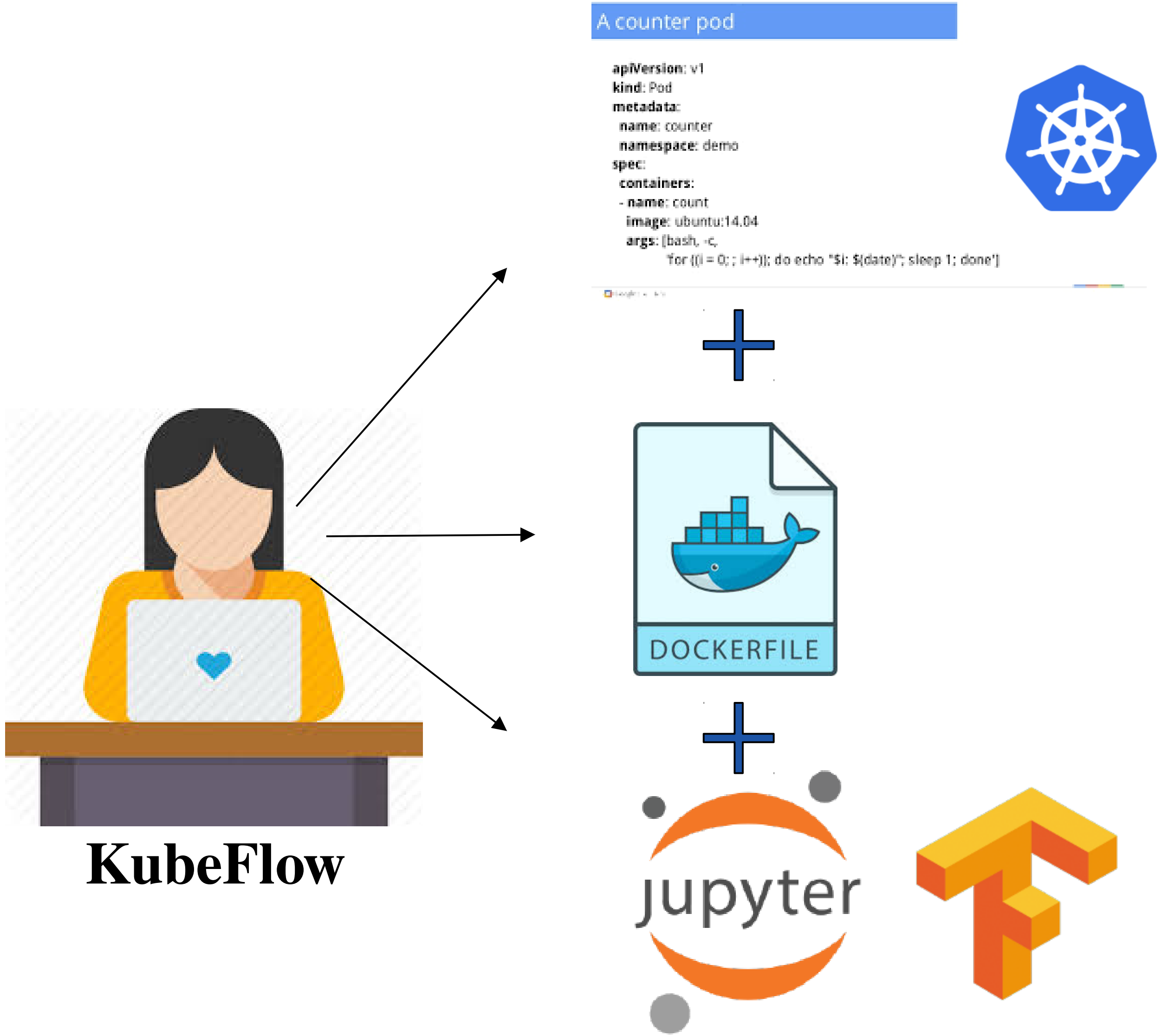
GPUs in YARN

*https://www.usenix.org/conference/fast17/technical-sessions/presentation/niazi

Hops Data Platform

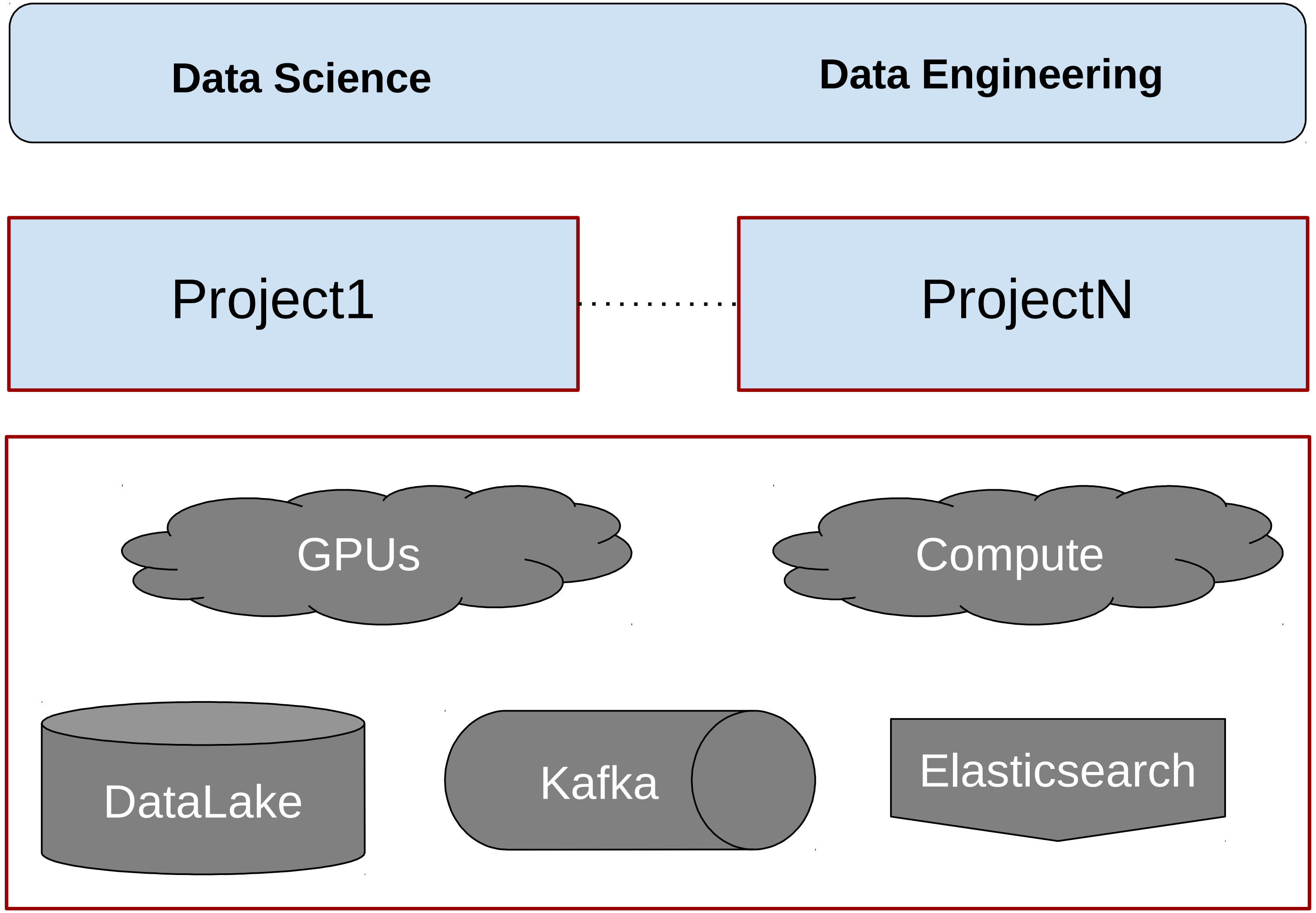


Why not KubeFlow?

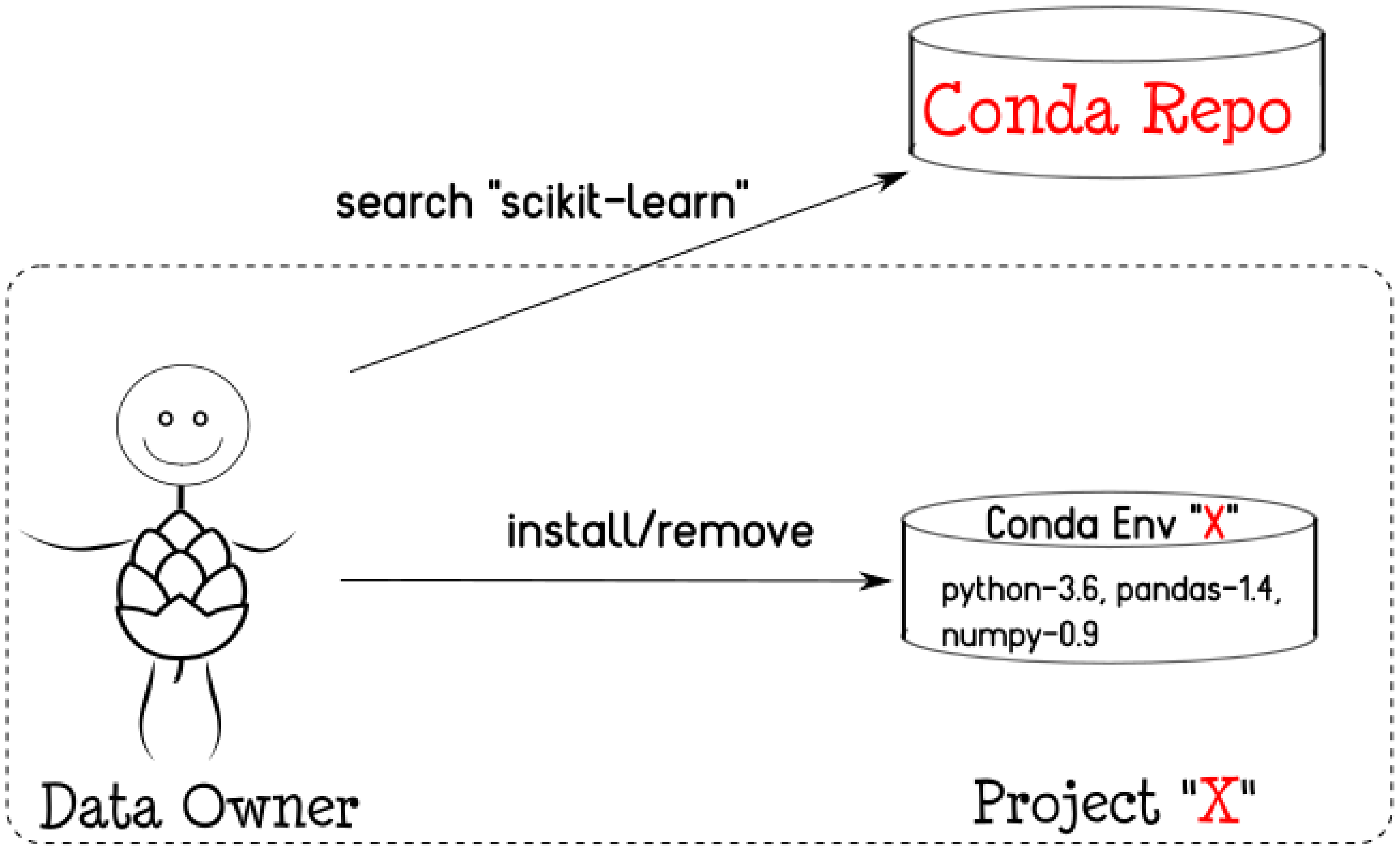


Hops – Projects and GPUs

1. **Security by Design**: Projects as a Sandbox for self-service and teams
2. **Ease of use:** Data Scientists need only code Python
3. **Scale-out Deep Learning:** Parallel experiments, Distributed Training



Python in the Cluster: Per-Project Conda Envs



Python libraries are usable by Spark/Tensorflow

Hops: TLS Certs for Security (not Kerberos)

- User Certificates:
  - Per-project users
- Service Certificates:
  - NameNode, ResourceManager, Kafka, HiveServer2, Livy, etc
- Application Certificates
- Supports Certification Revocation, Renewal, Reloading.

GPU Resource Requests in Hops YARN

10 GPUs on 1 host

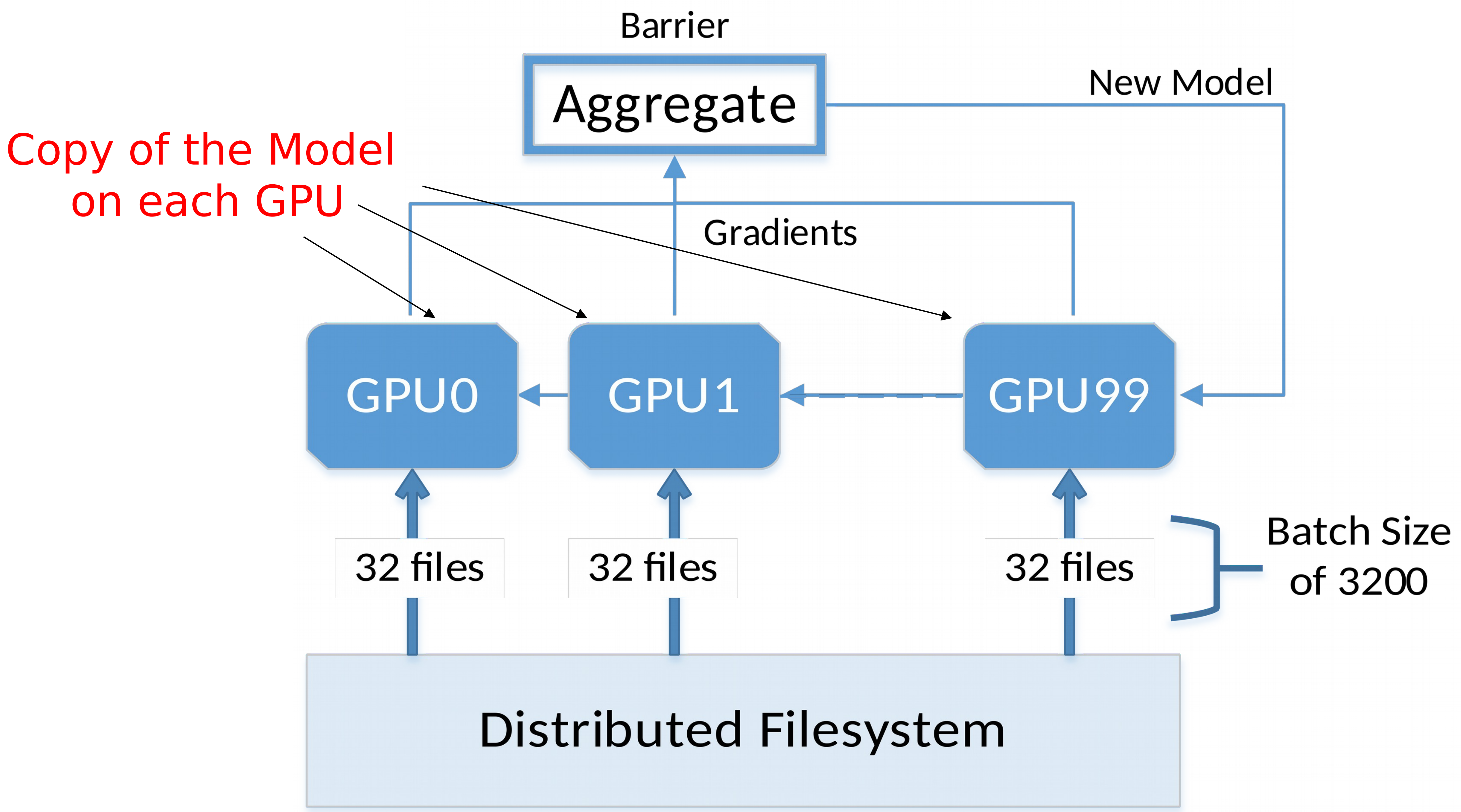
4 GPUs on any host

100 GPUs on 10 hosts with ‘Infiniband’

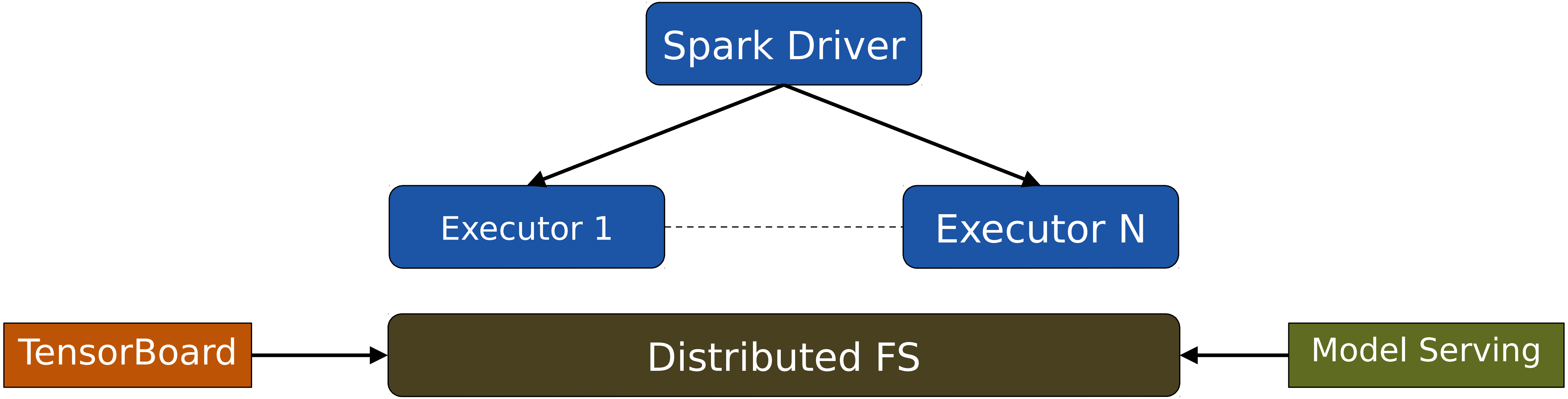
HopsYARN

Hops supports a Hetrogenous Mix of GPUs

Data Parallelism on Hops/TensorFlow



Parallel Experiments on Hops

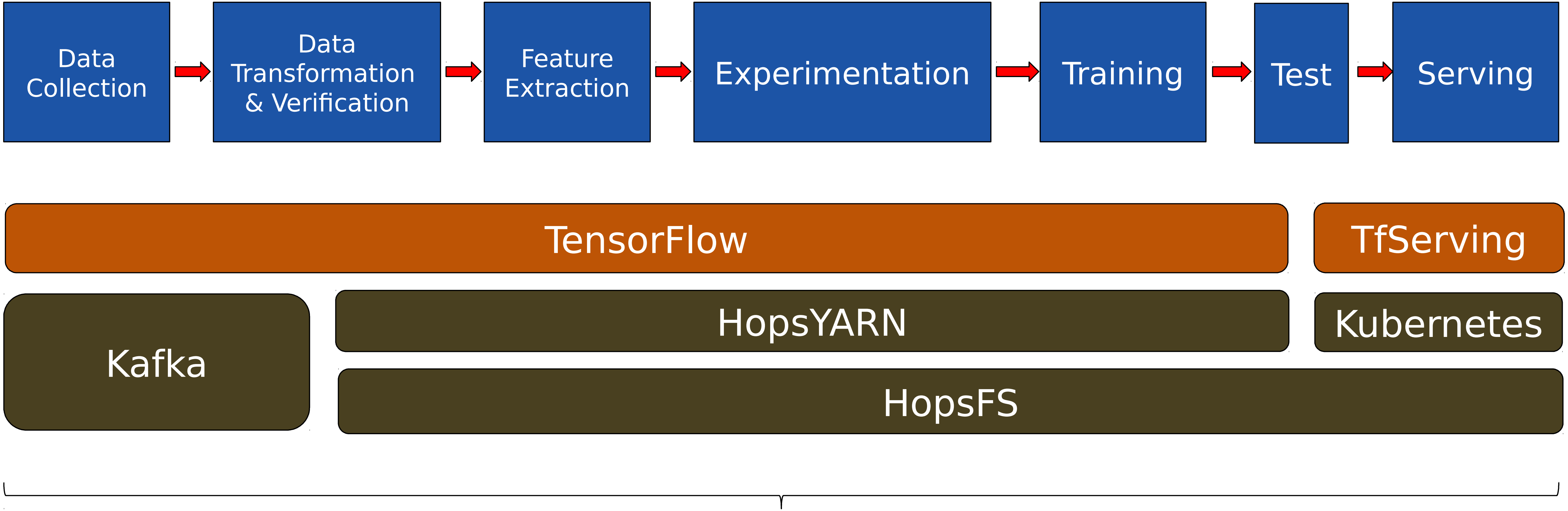


- Big Training/test datasets
- Model checkpointing
- Evaluation
- Distribute Training, Parallel Experiments
- Model Serving

ML in Production: Machine Learning Pipelines

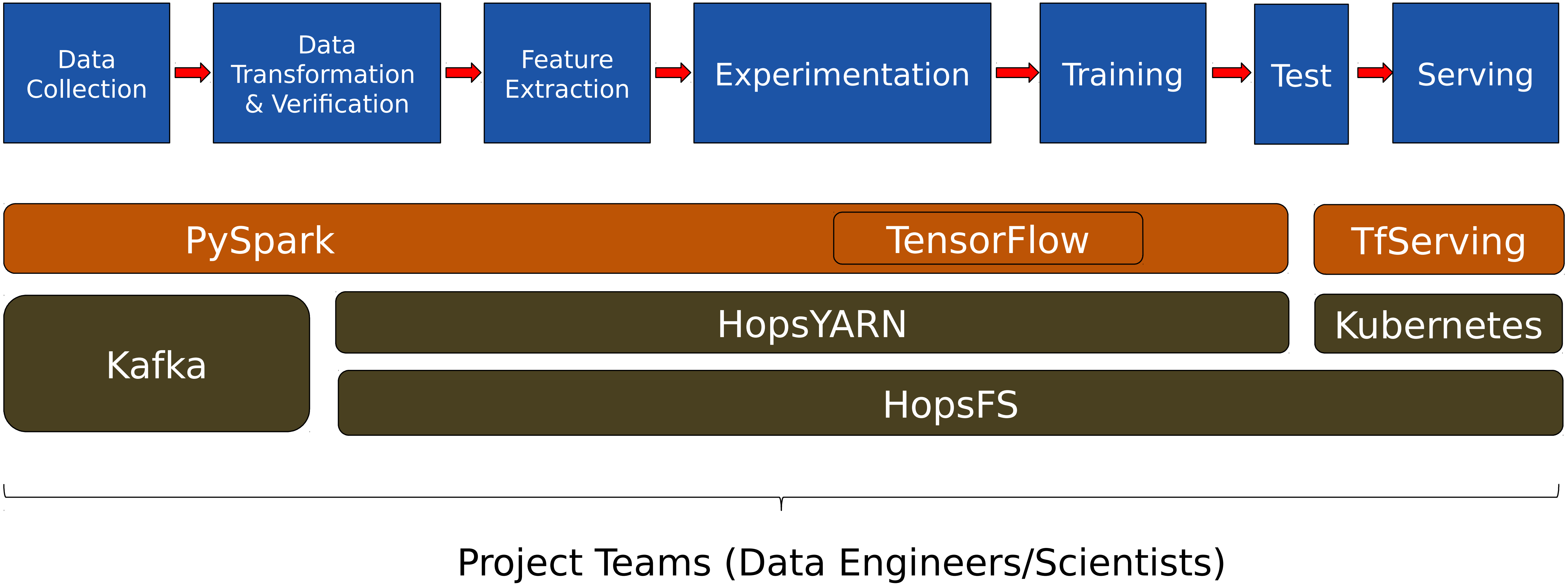
A Machine Learning Pipeline

Hops Small Data ML Pipeline



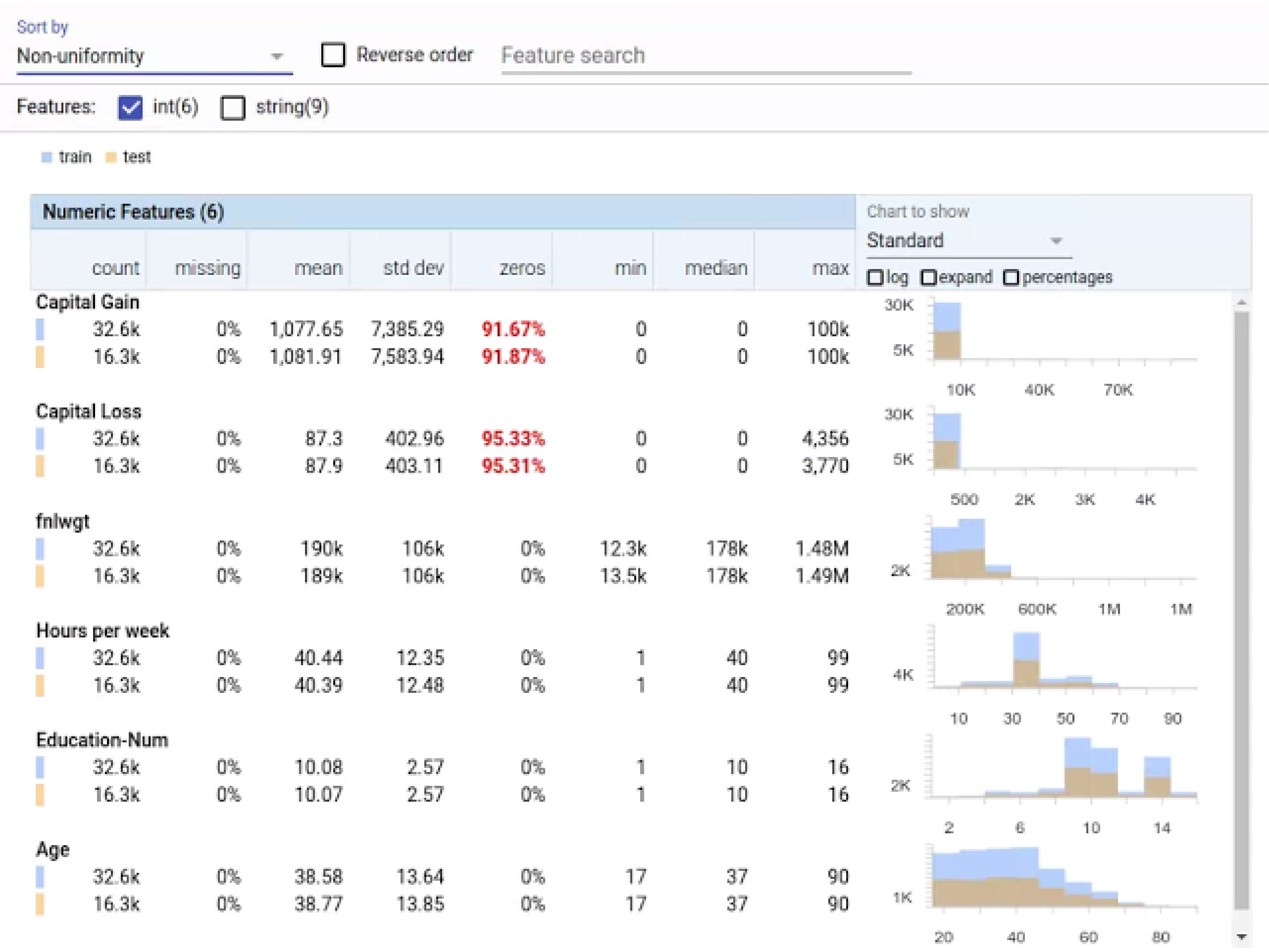
Project Teams (Data Engineers/Scientists)

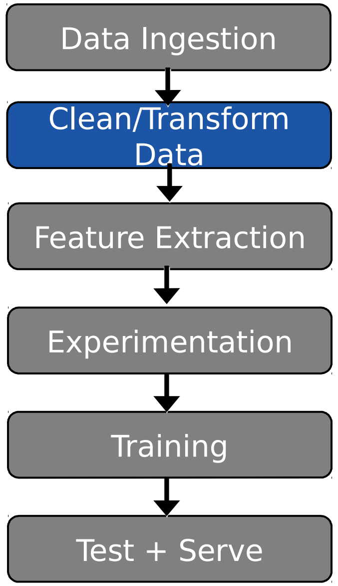
Hops Big Data ML Pipeline



Google Facets Overview

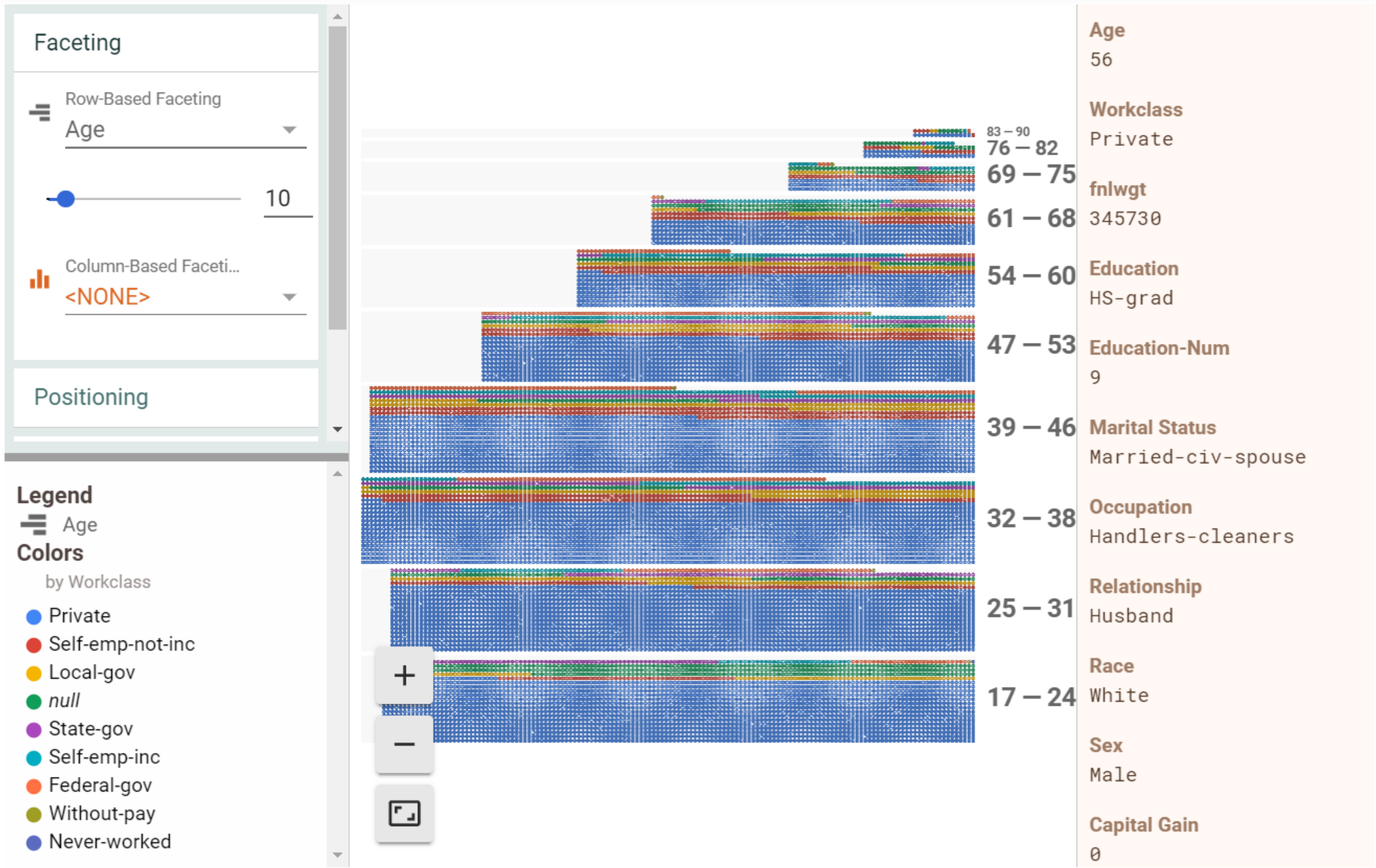
- Visualize data distributions
- Min/max/mean/median values for features
- Missing values in columns
- Facets Overview expects test/train datasets as input

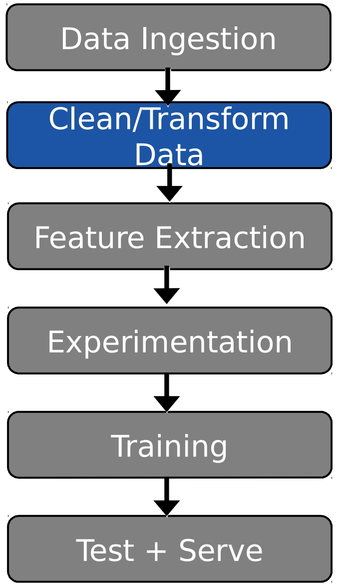




Google Facets Dive

- Visualize the relationship between the data points across the different features of a dataset.



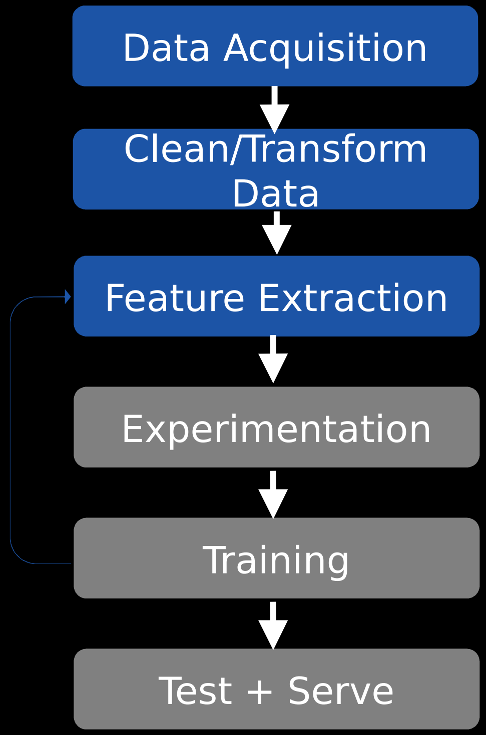


Data Ingestion and Google Facets

```
features = ["Age", "Occupation", "Sex", …, "Country"]

h = hdfs.get_fs()
with h.open_file(hdfs.project_path() +
 "/TestJob/data/census/adult.data", "r") as trainFile:
    train_data =pd.read_csv(trainFile, names=features,
   sep=r'\s*,\s*', engine='python', na_values="?")
    test_data = …


facets.overview(train_data, test_data)
facets.dive(test_data.to_json(orient='records'))
```

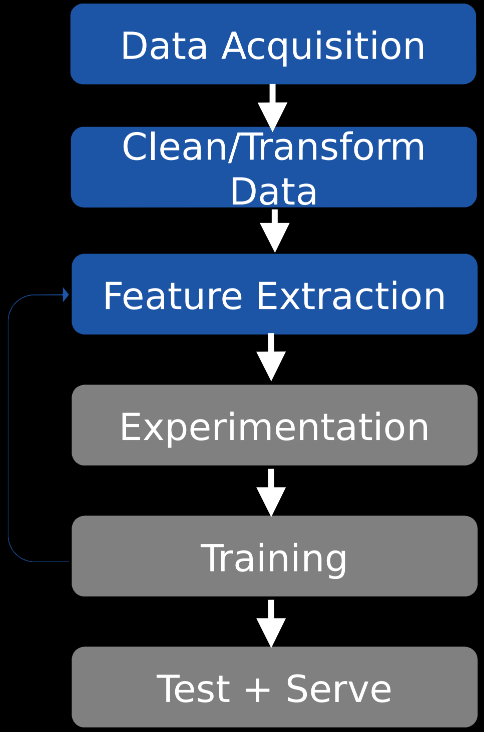
- Data Acquisition
- Clean/Transform Data
- Feature Extraction
- Experimentation
- Training
- Test + Serve

Small Data Preparation with tf.data API



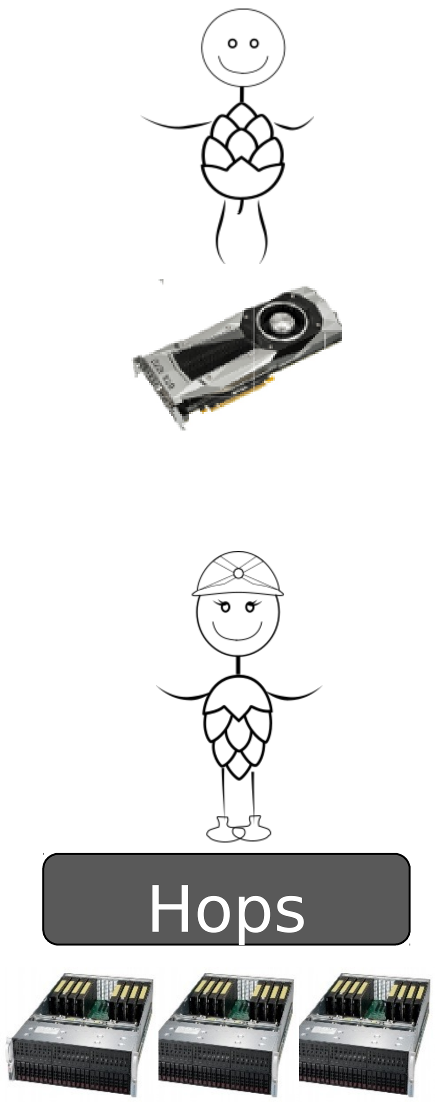

```
def input_fn(batch_size):
  files = tf.data.Dataset.list_files(IMAGES_DIR)

  def tfrecord_dataset(filename):
   return tf.data.TFRecordDataset(filename,
       num_parallel_reads=32, buffer_size=8*1024*1024)

  dataset = files.apply(tf.data.parallel_interleave
                     (tfrecord_dataset, cycle_length=32, sloppy=True)
  dataset = dataset.apply(tf.data.map_and_batch(parser_fn, batch_size,
                        num_parallel_batches=4))
  dataset = dataset.prefetch(4)
  return dataset
```

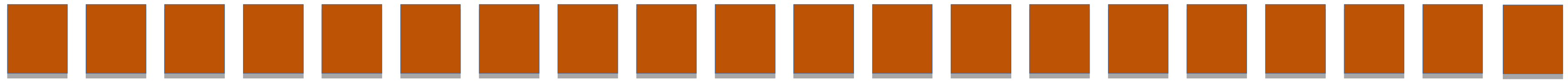
Big Data Preparation with PySpark

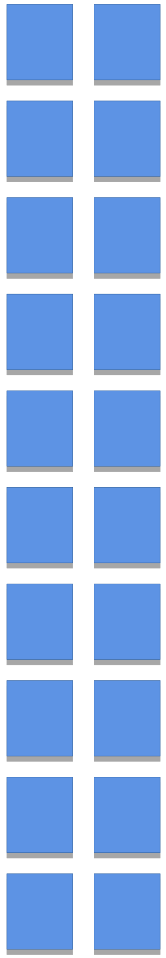



```
images = spark.readImages(IMAGE_PATH, recursive = True,
          numPartitions=10, sampleRatio = 0.1).cache()

tr = (ImageTransformer().setOutputCol(“transformed”)
      .resize(height = 200, width = 200)
      .crop(0, 0, height = 180, width = 180) )
smallImages = tr.transform(images).select(“transformed”)

# Output .tfrecords using TensorFlowOnSpark utility
dfutil.saveAsTFRecords(smallImages, OUTPUT_DIR)
```

Parallel Experiments

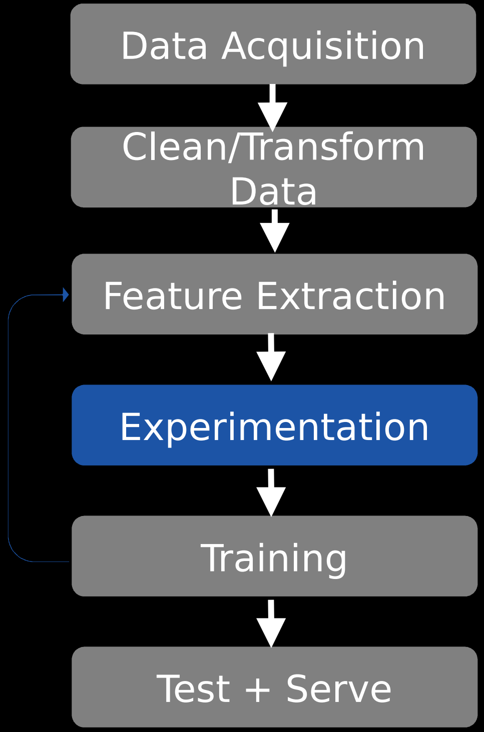






The Outer Loop (hyperparameters): *"I have to run a hundred experiments to find the best model," he complained, as he showed me his Jupyter notebooks. "That takes time. Every experiment takes a lot of programming, because there are so many different parameters.*
[Rants of a Data Scientist]

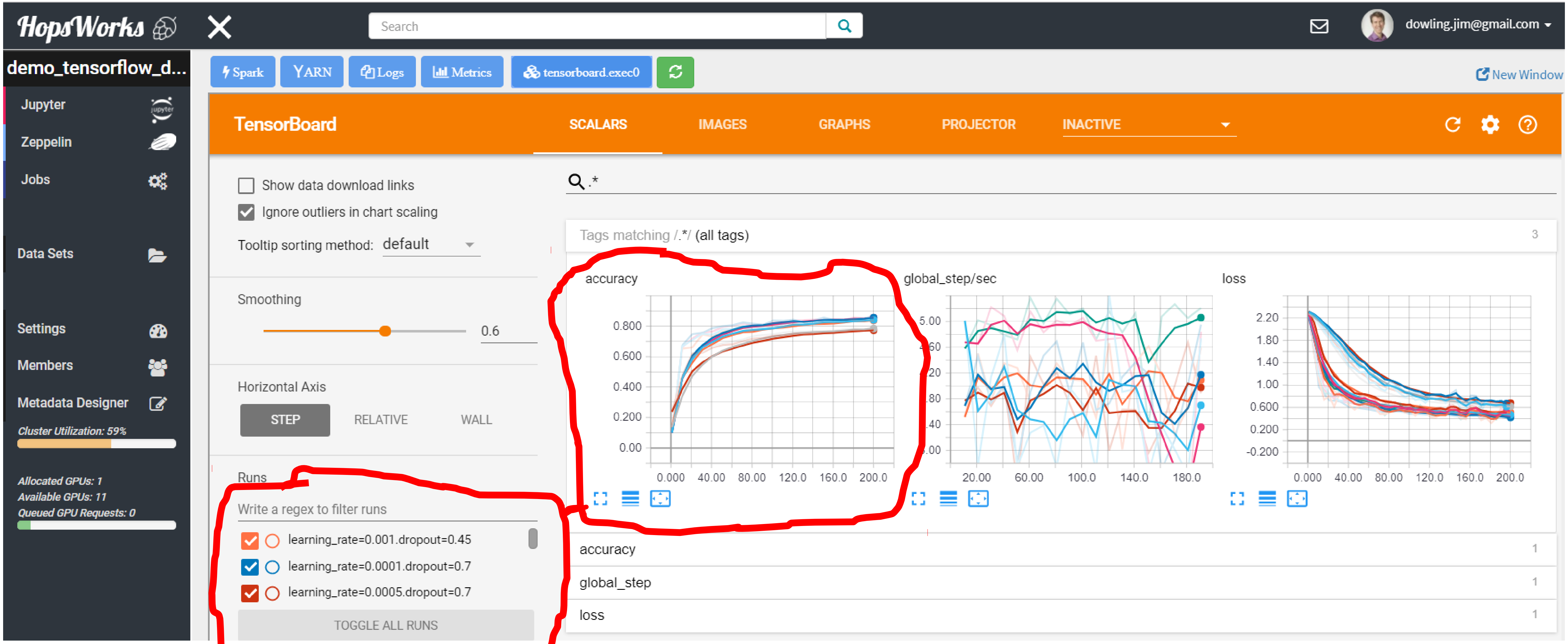
Hyperparam Opt. with Tf/Spark on Hops




```
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(spark, train, args_dict)
```

Launch 6 Spark Executors

HyperParam Opt. Visualization on TensorBoard



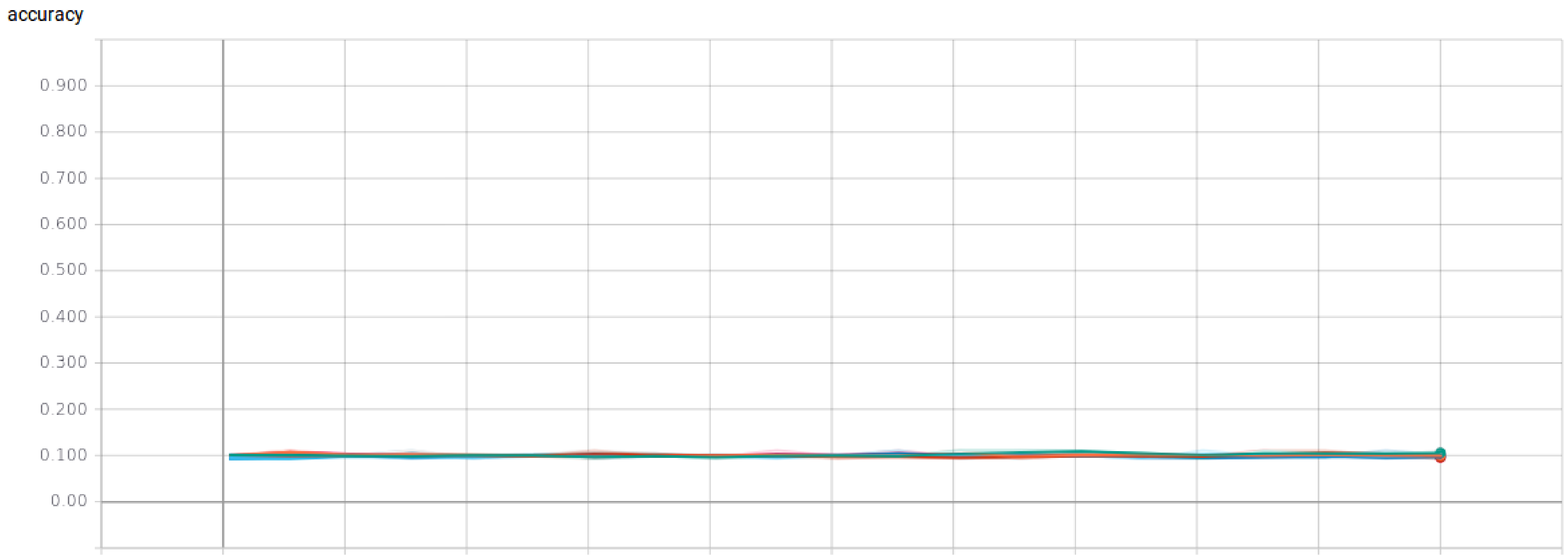
Hyperparam Opt Results Visualization

Model Architecture Search on TensorFlow/Hops

```
def train_cifar10(learning_rate, dropout):

 [TensorFlow Code here]

dict =
{'learning_rate': [0.005, 0.00005], 'dropout': [0.01, 0.99],
'num_layers': [1,3]}
experiment.evolutionary_search(spark, train_cifar10, dict,
direction='max',
    popsize=10, generations=3, crossover=0.7,  mutation=0.5)
```

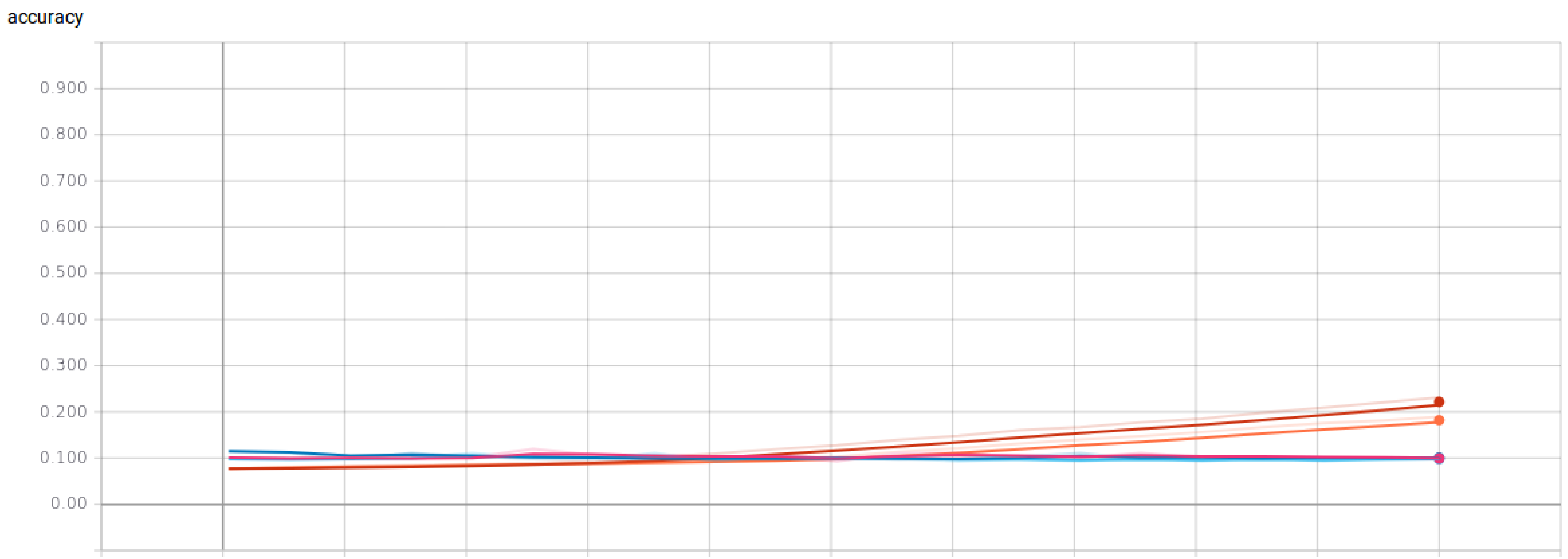

LOGICAL CLOCKS

Differential Evolution in Tensorboard (1/4)



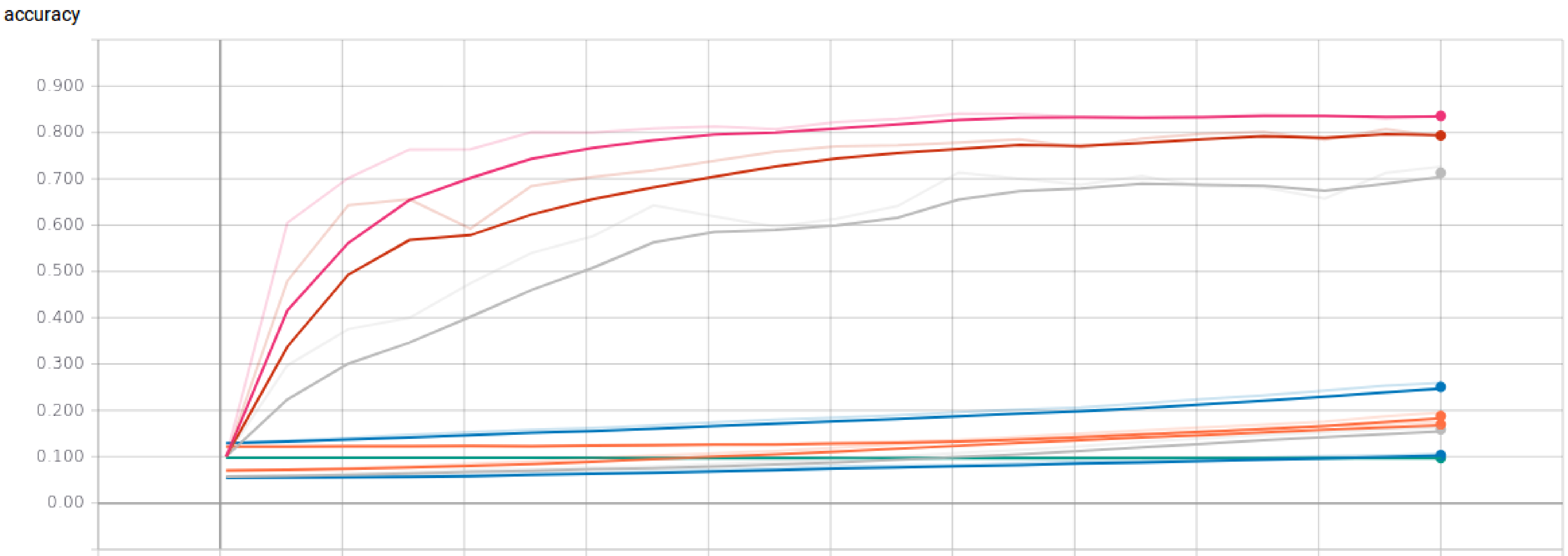
CIFAR-10

Differential Evolution in Tensorboard (2/4)



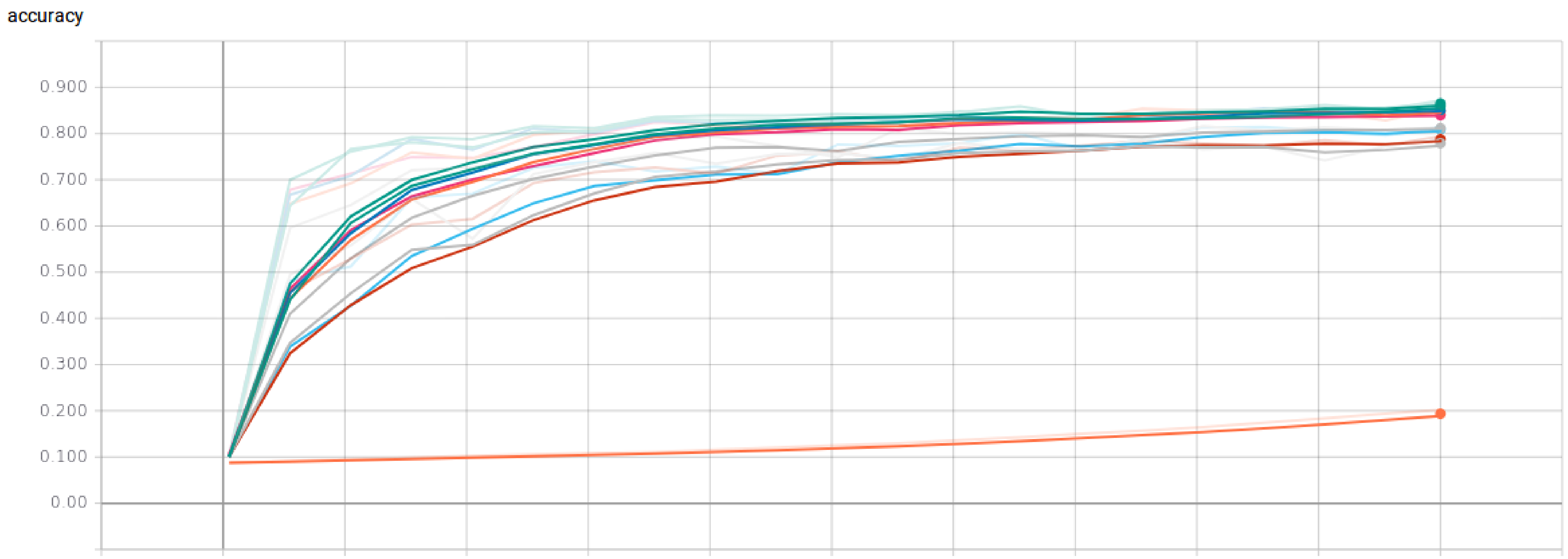
CIFAR-10

Differential Evolution in Tensorboard (3/4)



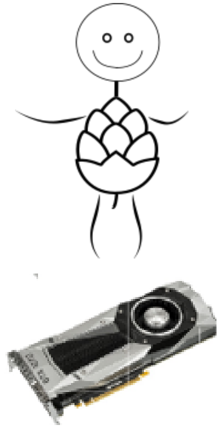
CIFAR-10

Differential Evolution in Tensorboard (4/4)

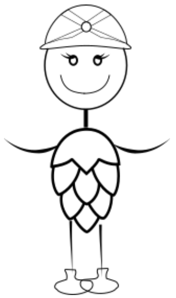


CIFAR-10

Distributed Training



Weeks



Mins

Hops

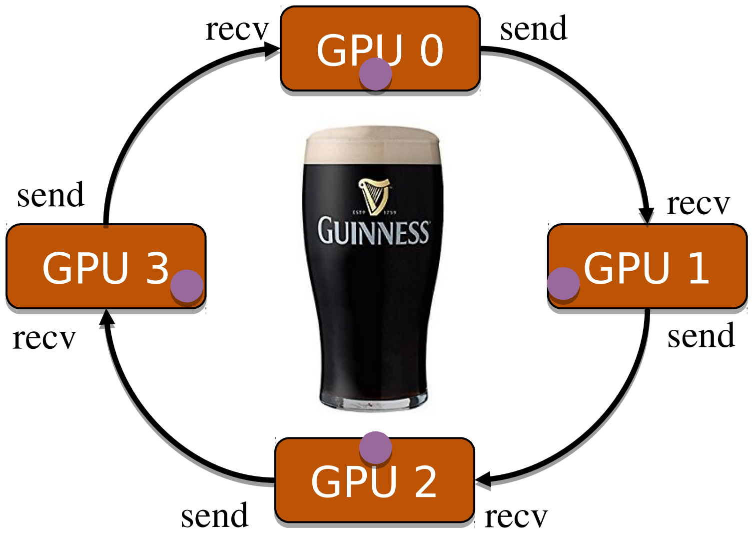
The Inner Loop (training):
" All these experiments took a lot of computation — we used hundreds of GPUs/TPUs for days. Much like a single modern computer can outperform thousands of decades-old machines, we hope that in the future these experiments will become household."
[Google SoTA ImageNet, Cifar-10, March18]

Time

Distributed Training: Theory and Practice

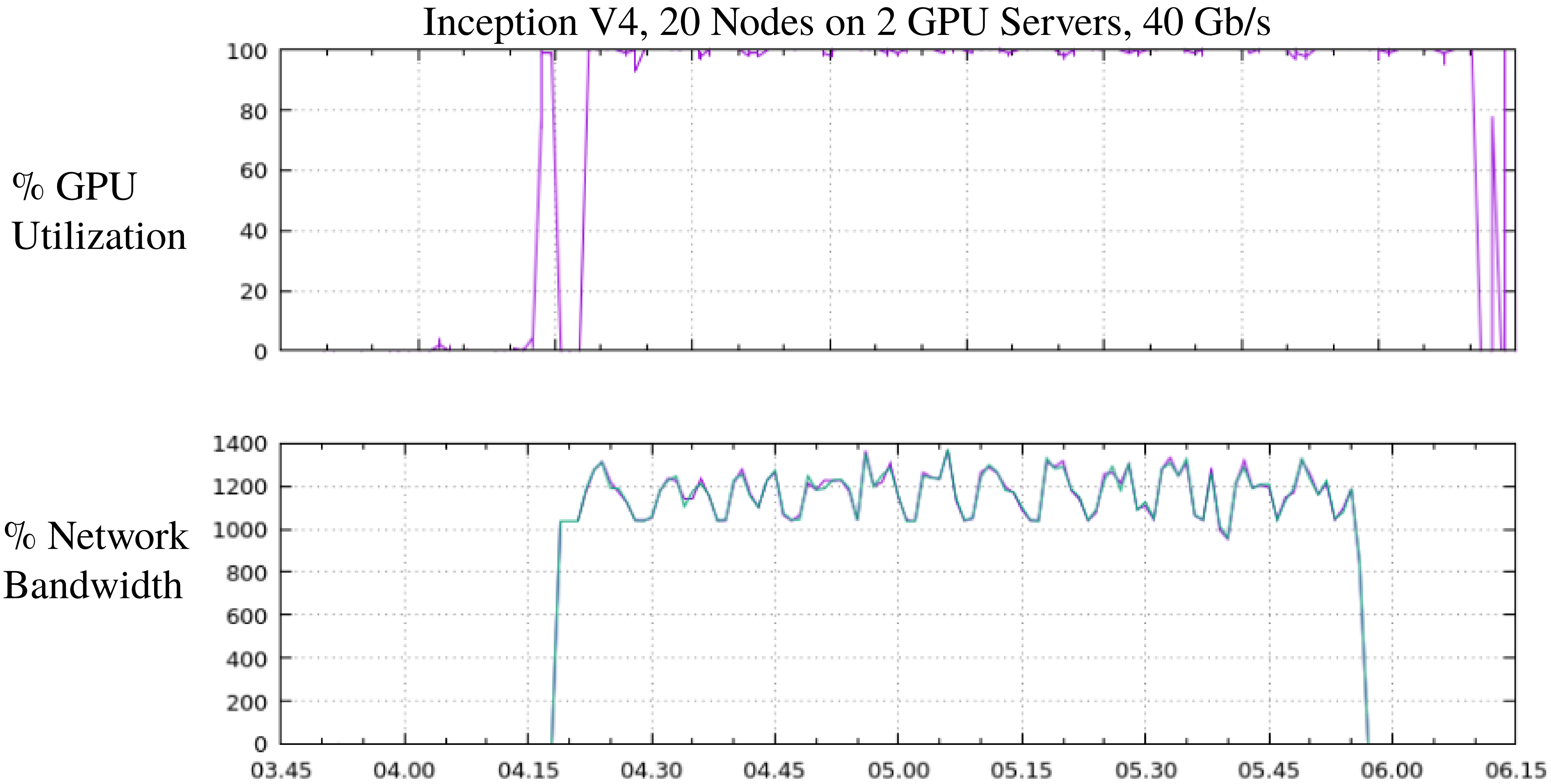
Image from @hardmaru on Twitter.

Ring-AllReduce vs Parameter Server



Network Bandwidth is the Bottleneck for Distributed Training

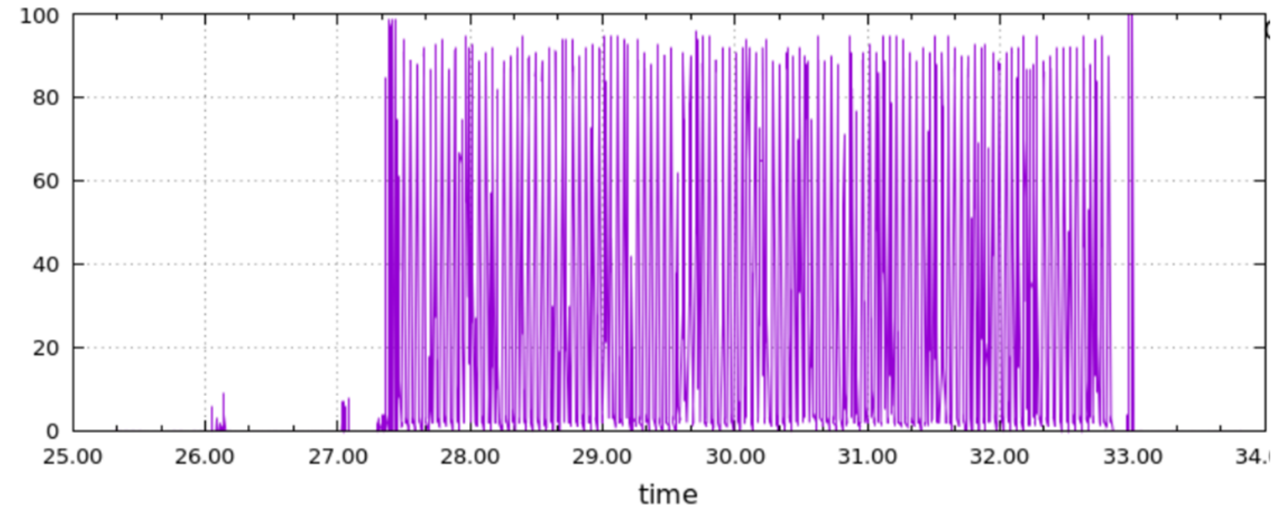
Horovod - AllReduce Inception V4 Performance



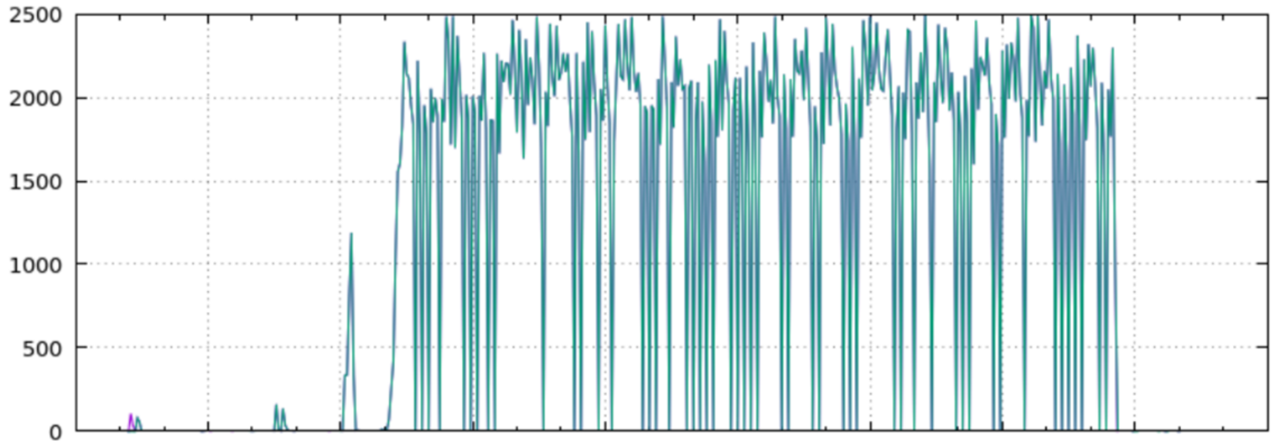
Parameter Server – Inception V4 Performance

Horovod, Inception V4, 10 workers, 2 PSes, 40 Gb/s, Nvidia 1080Ti

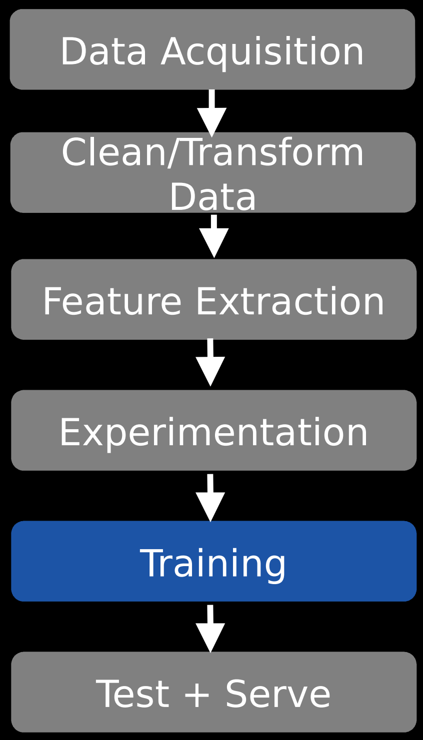
% GPU Utilization



% Network Bandwidth

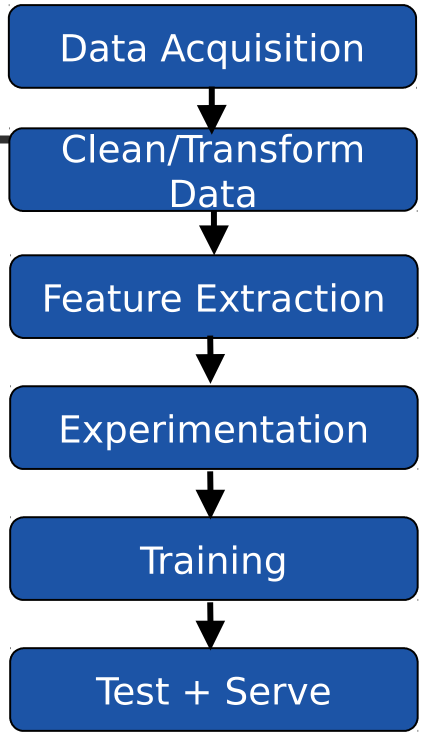


Distributed Training with Horovod on Hops



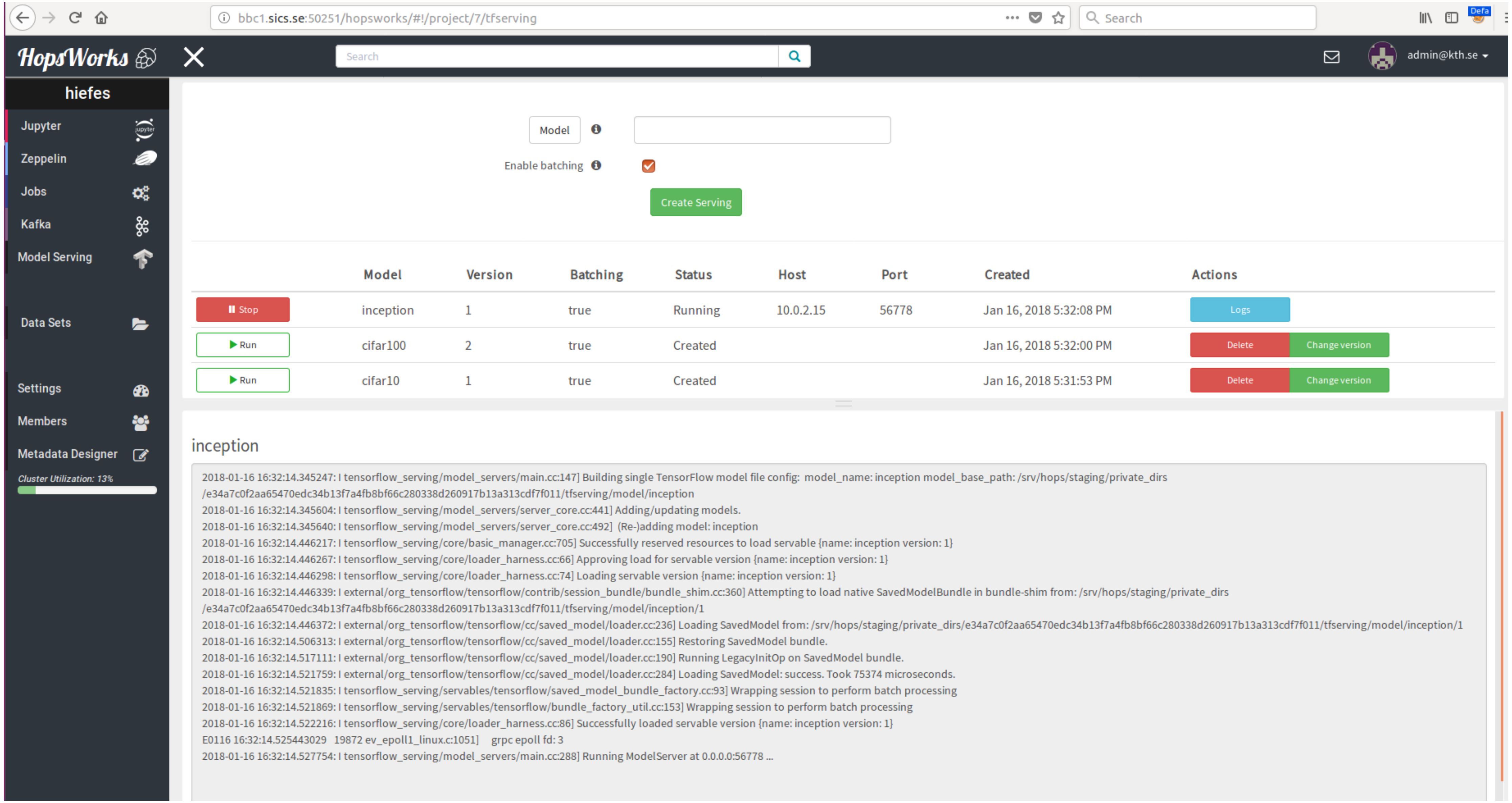

```
hvd.init()
opt = hvd.DistributedOptimizer(opt)
if hvd.local_rank()==0:
    [TensorFlow Code here]
.....
else:
    [TensorFlow Code here]
.....
```

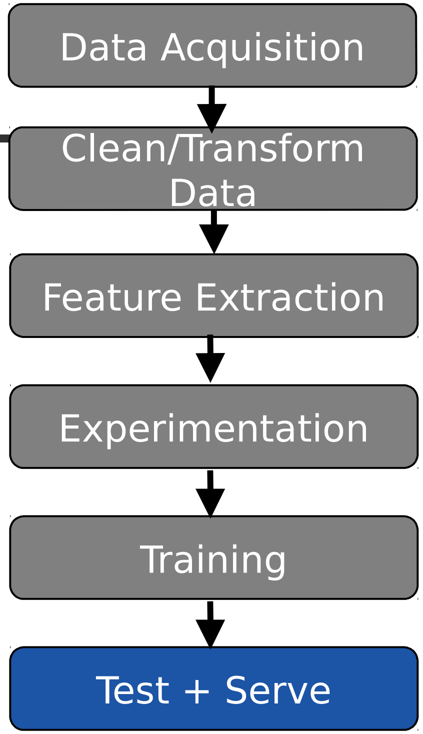
Hops API



- Python (also Java/Scala)
  - Manage tensorboard, Load/save models in HDFS
  - Horovod, TensorFlowOnSpark
  - Parallel experiments
    - Gridsearch
    - Model Architecture Search with Genetic Algorithms
  - Secure Streaming Analytics with Kafka/Spark/Flink
    - SSL/TLS certs, Avro Schema, Endpoints for Kafka/Zookeeper/etc

TensorFlow Model Serving





Training-Serving Skew

- Monitor differences between performance during training and performance during serving.
  - Differences in how you process data in training vs serving.
  - Differences in the training data and live data for serving.
  - A feedback loop between your model and your algorithm.

- When to retrain?
  - If you look at the input data and use **covariant shift** to see when it deviates significantly from the data that was used to train the model on.

http://hopshadoop.com:8080/hopsworks

0. Register an account

1. Create a ’tensorflow_tour’

serving/train_and_export_model.ipynb

Try out other notebooks – tensorflow/cnn/grid_search,

- 2. Create a new project called ’yourname_hotdog’
- a) enable Python 3.6 for the project
  b) search for the dataset ’hotdog’ and import it into ’yourname_hotdog’
  c) download http://hopshadoop.com:8080/hotdog.ipynb to your laptop.
  d) upload **hotdog.ipynb** into the Jupyter dataset in `yourname_hotdog`
  e) install the conda dependencies: matplotlib, pillow, numpy
  f) Start Jupyter and run **hotdog.ipynb**

  [Credit Magnus Pedersson : https://www.youtube.com/watch?v=oxrcZ9uUblI]

Summary

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

- You can do fun stuff on Hops, like ”Hotdog or not”, as well as serious stuff.

The Team

Active:

Jim Dowling, Seif Haridi, Gautier Berthou, Salman Niazi, Mahmoud Ismail, Theofilos Kakantousis, Ermias Gebremeskel, Antonios Kouzoupis, Alex Ormenisan, Fabio Buso, Robin Andersson, August Bonds.

Alumni:

Vasileios Giannokostas, Johan Svedlund Nordström,Rizvi Hasan, Paul Mälzer, Bram Leenders, Juan Roca, Misganu Dessalegn, K “Sri” Srijeyanthan, Jude D’Souza, Alberto Lorente, Andre Moré, Ali Gholami, Davis Jaunzems, Stig Viaene, Hooman Peiro, Evangelos Savvidis, Steffen Grohsschmiedt, Qi Qi, Gayana Chandrasekara, Nikolaos Stanogias, Daniel Bali, Ioannis Kerkinos, Peter Buechler, Pushparaj Motamari, Hamid Afzali, Wasif Malik, Lalith Suresh, Mariano Valles, Ying Lieu, Fanti Machmount Al Samisti, Braulio Grana, Adam Alpire, Zahin Azher Rashid, ArunaKumari Yedurupaka, Tobias Johansson , Roberto Bampi, Roshan Sedar.

www.hops.io

@hopshadoop