DEEP LEARNING AND APACHE SPARK

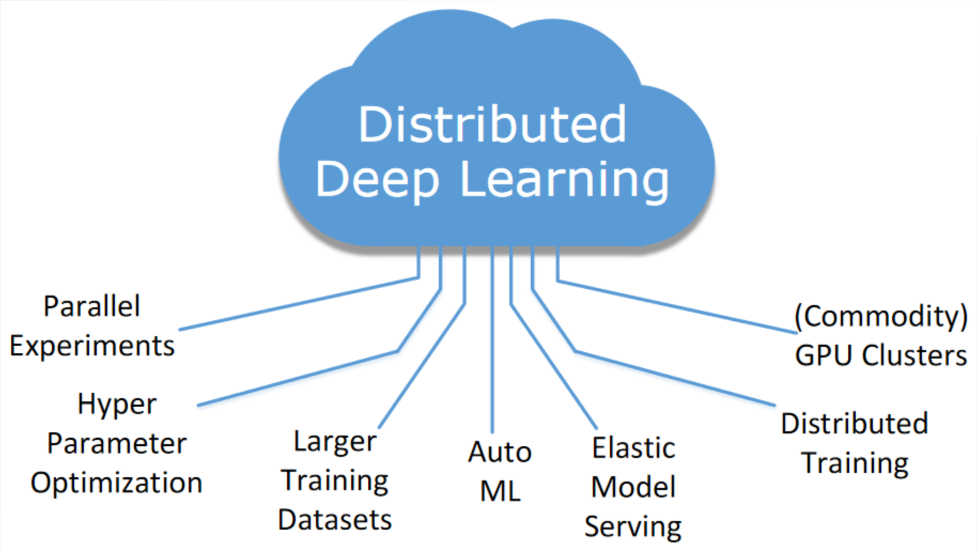
Lecturer: Zeinab Zali

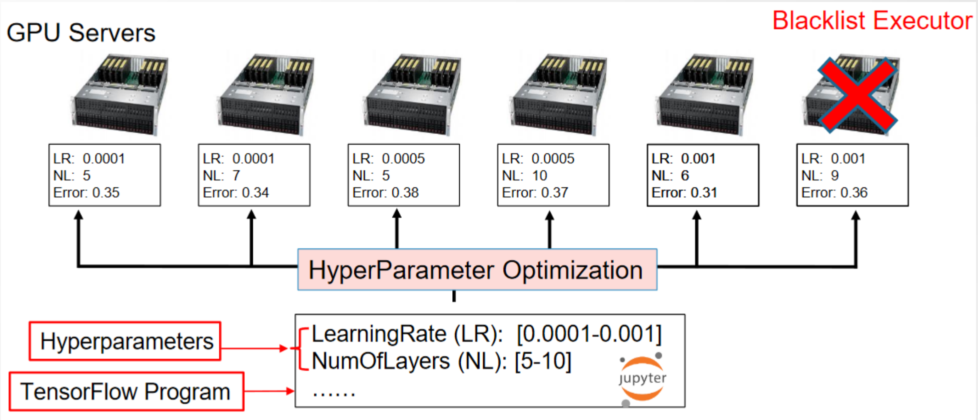
Presenter: Alireza Salehi Hossein Abadi

Why?[1]

- Apache Spark is an amazing framework for distributing computations in a cluster in a easy and declarative way. Is becoming an standard across industries so it would be great to add the amazing advances of Deep Learning to it.
- There are parts of Deep Learning that are computationally heavy, very heavy! Distributing these processes may be the solution to this an other problems, and Apache Spark is the easiest way I could think to distribute them.

Why? (continue)[5]



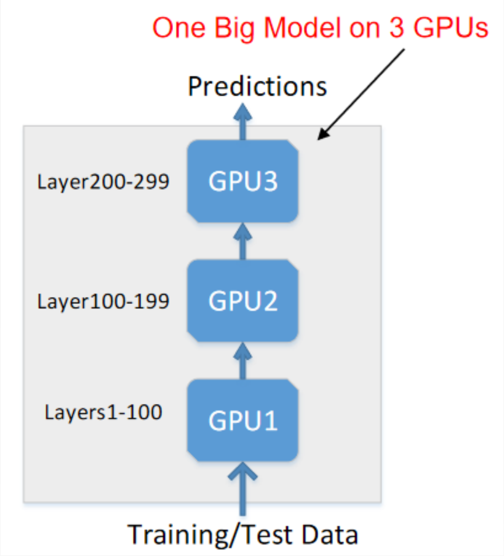


DECLARATIVE/API APPROACH[5]

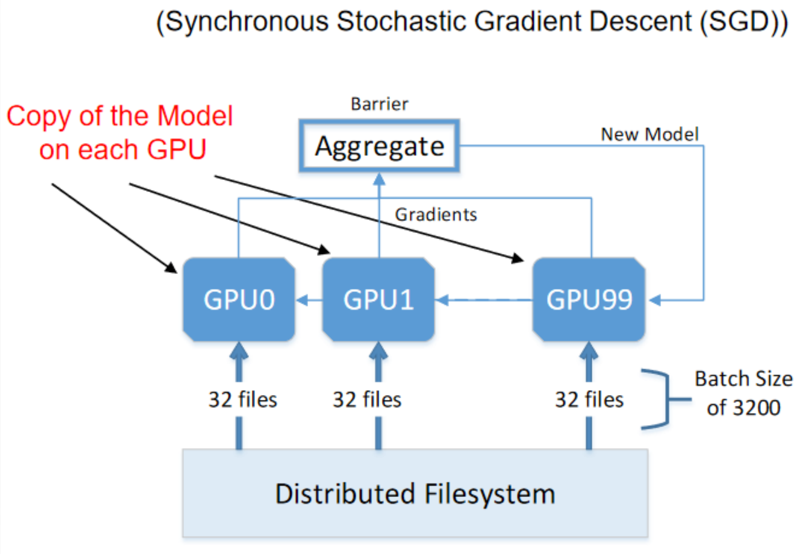
- Declarative Hyperparameters in external files
  - Vizier/CloudML (yaml)
  - Sagemaker (json)*
- API-Driven
  - Databrick’s MLFlow
  - HopsML

DISTRIBUTED TRAINING[5]

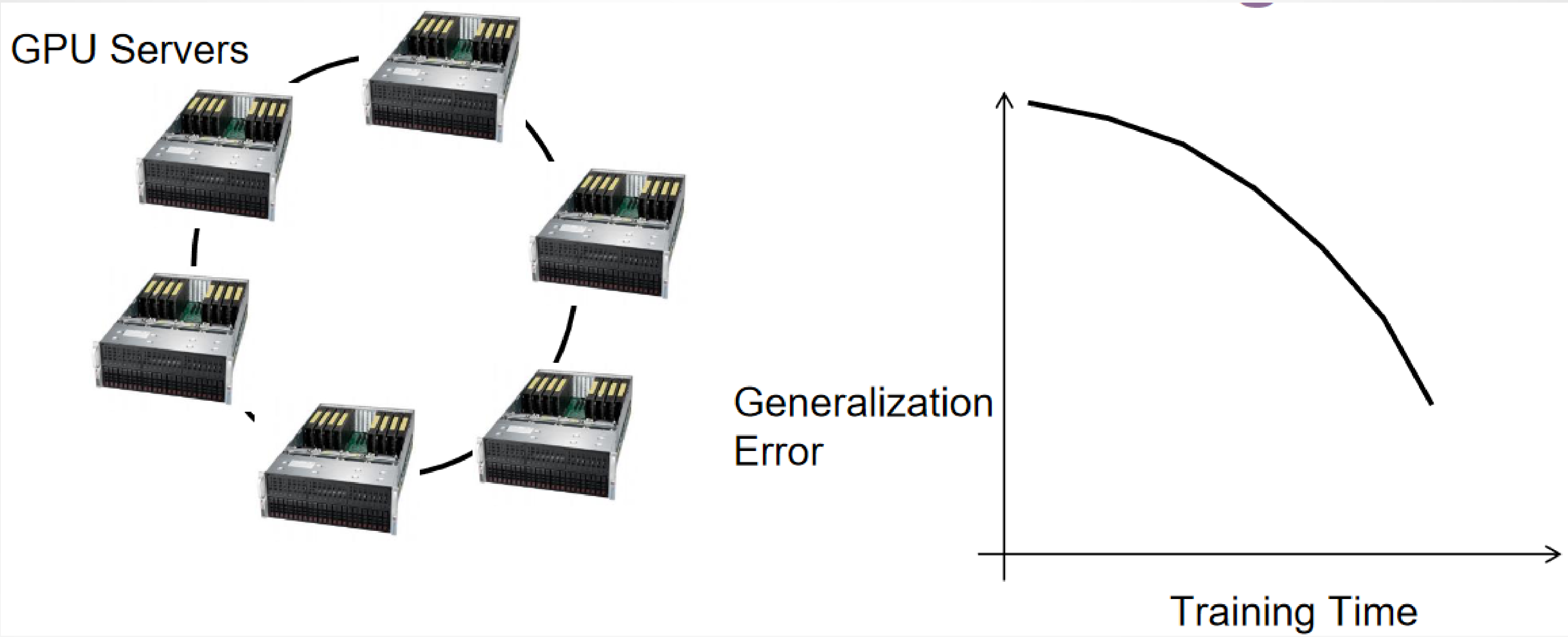
MODEL PARALLELISM



DATA PARALLELISM



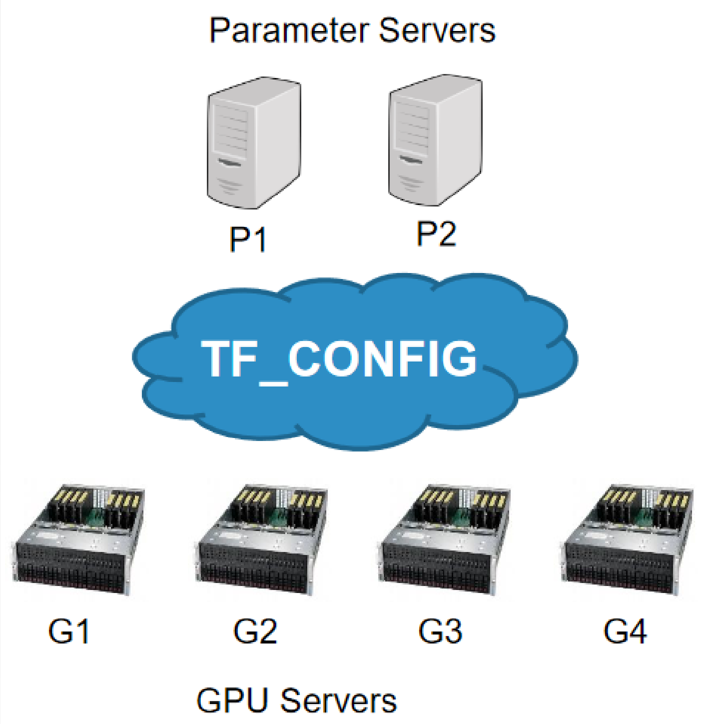
DATA PARALLEL DISTRIBUTED TRAINING



Ways[1]

- Elephas: Distributed DL with Keras & PySpark
- Yahoo! Inc.: TensorFlowOnSpark
- CERN Distributed Keras (Keras + Spark)
- Qubole (Keras + Spark)
- Intel Corporation: BigDL (Distributed Deep Learning Library for Apache Spark)[2]

DISTRIBUTED TENSORFLOW / TFONSPARK[5]

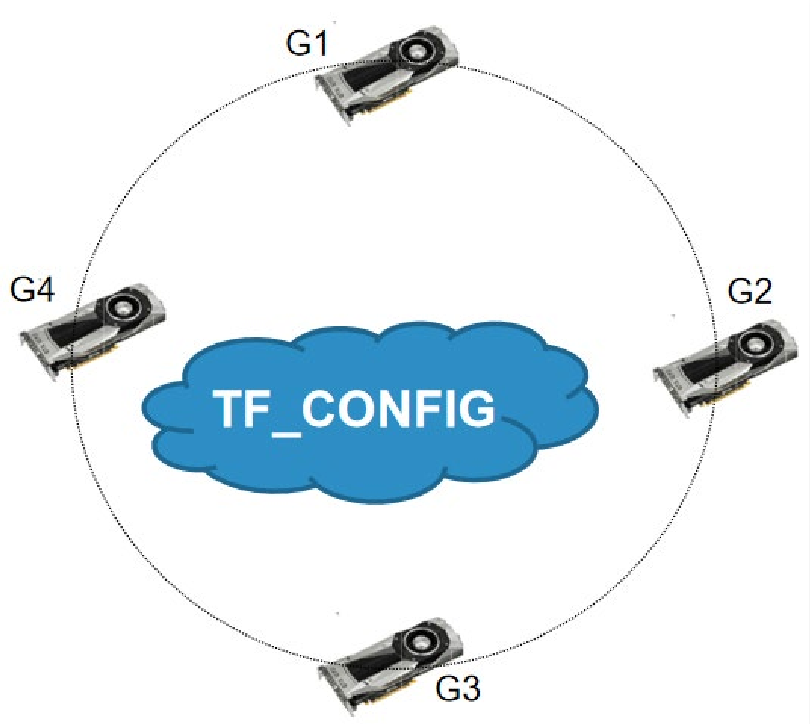


- TF_CONFIG
  Bring your own Distribution!
  - Start all processes for P1,P2, G1-G4 yourself
  - Collect all IP addresses in TF_CONFIG along with GPU device IDs.

RING ALL REDUCE (HOROVOD)[5]

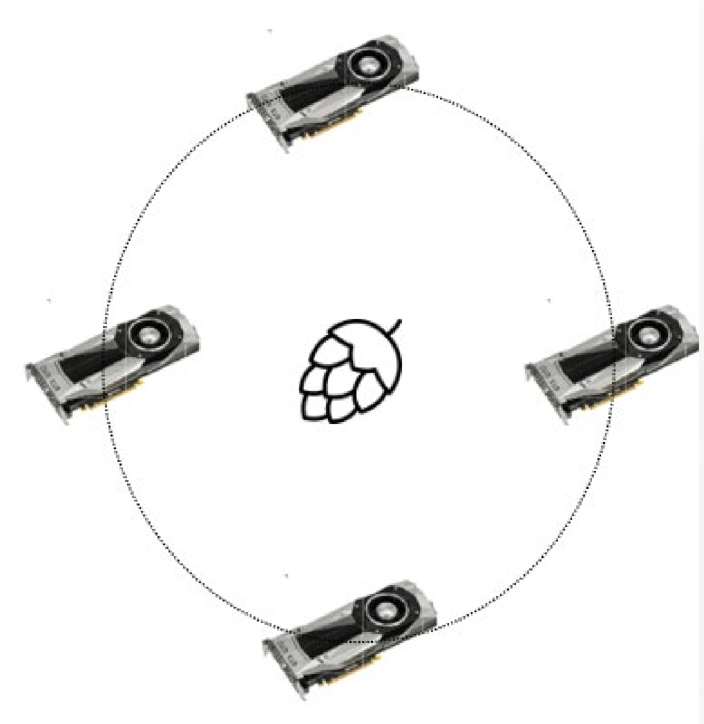
- Bandwidth optimal
- Automatically builds the ring (MPI)
- Supported by HopsML and Databricks' HorovodEstimator

TF COLLECTIVEALLREDUCESTRATEGY[5]



- TF_CONFIG
  Bring your own Distribution!
  - Start all processes for G1-G4 yourself
  - Collect all IP addresses in TF_CONFIG along with GPU device IDs.

HOPSML COLLECTIVE ALL REDUCE STRATEGY[5]



- Uses Spark/YARN to add distribution to TensorFlow's Collective All Reduce Strategy
  - Automatically builds the ring (Spark/YARN)
- Scale to 10s or 100s of GPUs on Hops
- Generate Tensor board Logs in HopsFS
- Checkpoint to HopsFS
- Save a trained model to HopsFS
- Experiment History
  - Reproducible training

COLLECTIVE ALL REDUCE VS HOROVOD[5]

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

COLLECTIVE ALL REDUCE VS HOROVOD[5]

TensorFlow:	1.11		TensorFlow:	1.7	
Model:	**VGG19**		Model:	**VGG19**	
Dataset:	imagenet (synthetic)		Dataset:	imagenet (synthetic)	
Batch size:	256 global, 32.0 per device		Batch size:	256 global, 32.0 per device	
Num batches:	100		Num batches	100	
Optimizer	Momemtum		Optimizer	Momemtum	
Num GPUs:	8		Num GPUs:	8	
AllReduce:	**collective**		AllReduce:	**horovod**	
Step	Img/sec	total_loss	Step	Img/sec	total_loss
1	images/sec: 634.4 +/- 0.0		1	images/sec: 583.01 +/- 0.0	
10	images/sec: 635.2 +/- 0.8		10	images/sec: 582.22 +/- 0.1	
100	images/sec: 635.0 +/- 0.5		100	images/sec: 583.61 +/- 0.2	

Big Model

total images/sec: **634.80**

total images/sec: **583.61**

REDUCTION IN LOC FOR DIST TRAINING[5]

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

BIGDL[2]

- a distributed deep learning library for Apache Spark
- implements distributed, data-parallel training directly on top of the functional compute model using the core Spark features of copy-on-write and coarse-grained operations
- has been referenced in applications as diverse as transfer learning-based image classification, object detection and feature extraction, sequence-to-sequence prediction for precipitation nowcasting, neural collaborative filtering for recommendations, and more
- Contributors and users include a wide range of industries including Mastercard, World Bank, Cray, Talroo, University of California San Francisco (UCSF), JD, UnionPay, Telefonica, GigaSpaces[3]

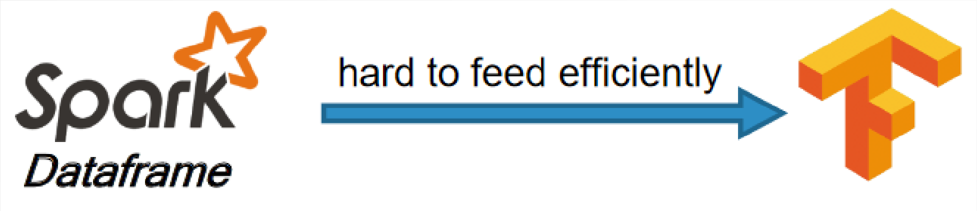
H20.AI – SPARKLING WATER FOR SPARK[2]

- fast, scalable, open-source machine learning, and deep learning for smarter applications
- cover a wide range of useful machine learning techniques
- Cover only fully connected MLPs for deep learning
- enterprises like PayPal, Nielsen Catalina, Cisco, and others can use all their data without sampling to get accurate predictions faster[4]

Deep learning pipelines[1]

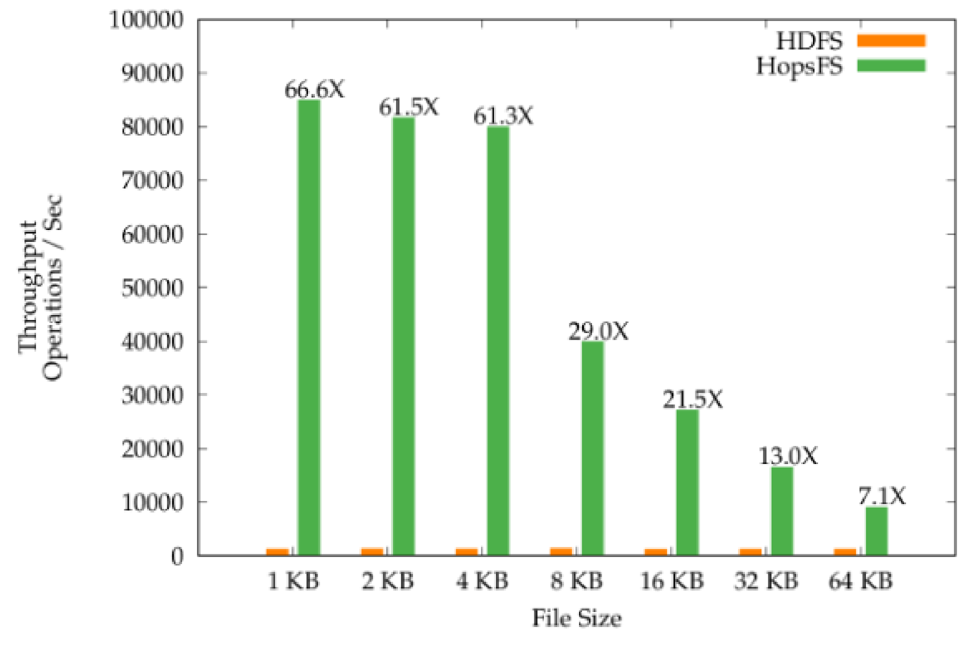
- an open source library created by Databricks
- provides high-level APIs for scalable deep learning in Python with Apache Spark
- won’t be long until is merged into the official API
- Some of the advantages of this library compared to the ones:
  - In the spirit of Spark and Spark MLlib, it provides easy-to-use APIs that enable deep learning in very few lines of code.
  - It focuses on ease of use and integration, without sacrificing performace.
  - It’s build by the creators of Apache Spark (which are also the main contributors) so it’s more likely for it to be merged as an official API than others.
  - It is written in Python, so it will integrate with all of its famous libraries, and right now it uses the power of TensorFlow and Keras, the two main libraries of the moment to do DL.

The Data Layer[5]



- FEED_DICT is single threaded (Python GIL)
- TensorFlow Dataset API does not support DFs
- Petastorm (Uber) for Parquet->TensorFlow training
- What about Datafiles (.csv, images, txt)?

HOPSFS[5]

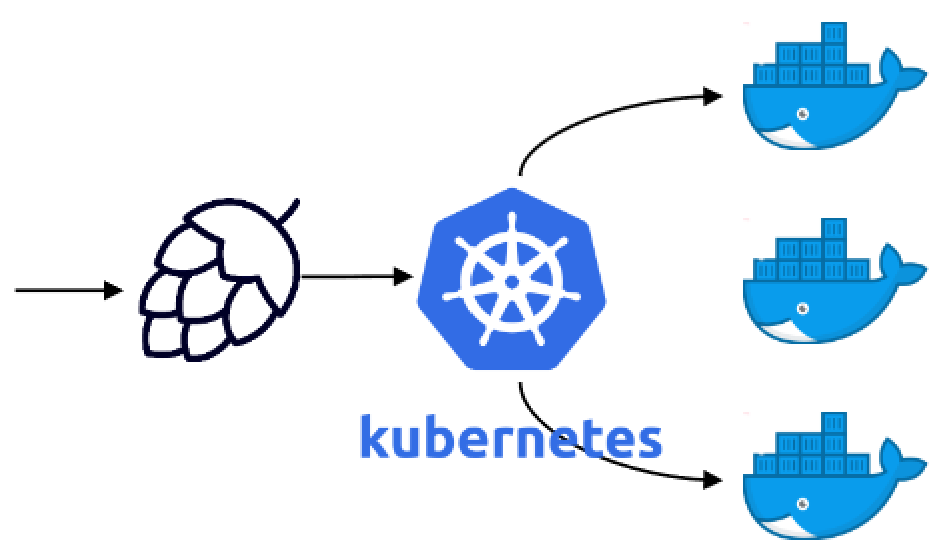


a. File Write Performance

- HDFS derivative with Distributed Metadata
  - 16X HDFS throughput.
  - Winner IEEE Scale Prize 2017
- Integrates NVMe disks transparently*
  - Store small files (replicated) on NVMe hardware

KUBERNETES MODEL SERVING[5]

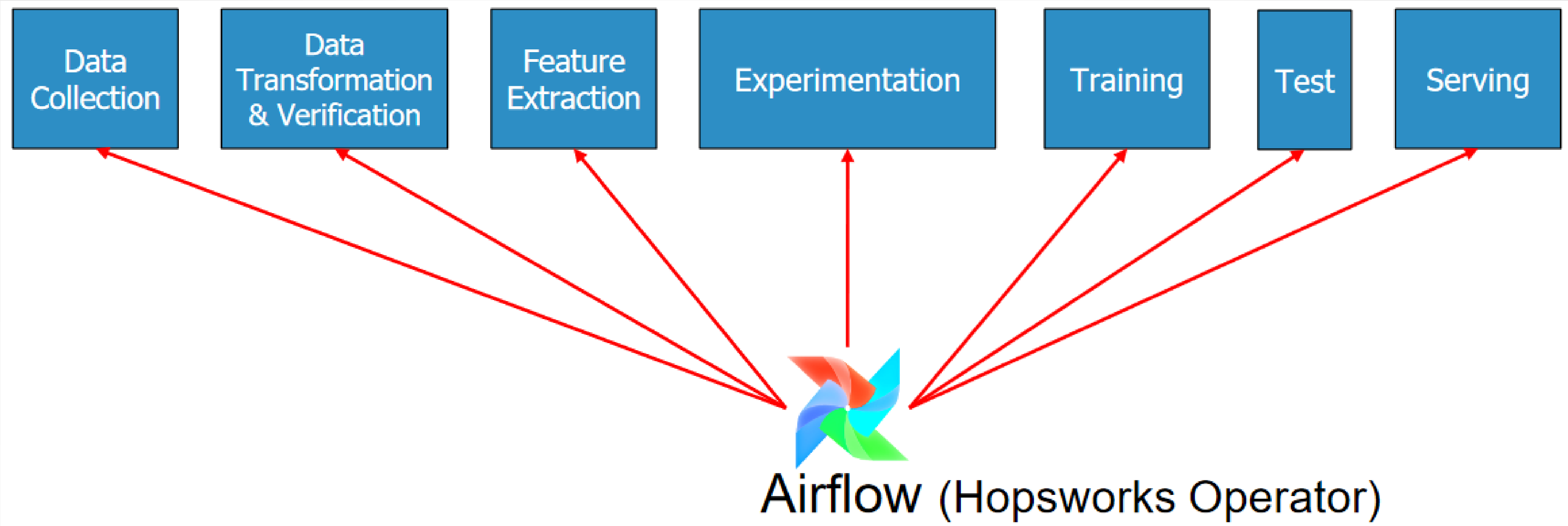
- Elastic scaling for model serving
- Supports:
  - Fault tolerance
  - Rolling release new models
  - Autoscaling



MODEL MONITORING WITH SPARK STREAMING[5]

- Model Monitoring with Spark Streaming Log model inference requests/results to Kafka
- Spark monitors model performance and input data
- 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.

ORCHESTRATING HOPSML WORKFLOWS[5]



More Information

- Deep Learning With Apache Spark — Part 2
- Databricks Machine Learning guide

REFRENCES

[1] Deep Learning With Apache Spark — Part 1

[2] Deep Learning on Spark is Getting Interesting

[3] Dai, J. J., Wang, Y., Qiu, X., Ding, D., Zhang, Y., Wang, Y., ... & Wang, J. (2019, November). Bigdl: A distributed deep learning framework for big data. In Proceedings of the ACM Symposium on Cloud Computing (pp. 50-60).

[4] Candel, A., Parmar, V., LeDell, E., and Arora, A. (Apr 2020). Deep Learning with H2O https://www.h2o.ai/wp-content/themes/h2o2016/images/resources/DeepLearningBooklet.pdf

[5] Distributed Deep Learning with Apache Spark and TensorFlow, Jim Dowling, Logical Clocks AB