



www.hops.io

 @hopshadoop

Hops, the world's fastest Hadoop distribution

Jim Dowling



Assoc Prof

@ KTH

Senior Researcher @ RISE SICS

CEO

@ Logical Clocks AB

JFokus, Feb 6th 2018

Logical Clocks AB*

- Seed Funded
 - 12 FTEs



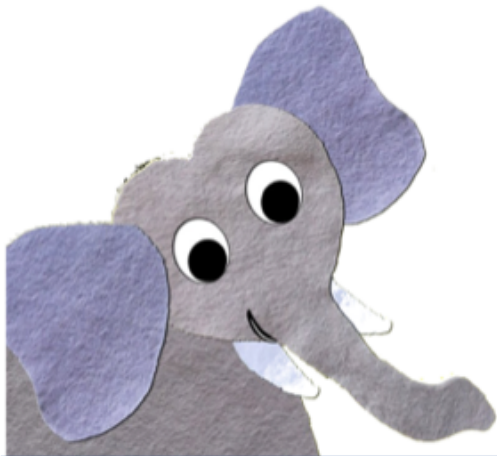
* Logical Clocks are needed to build reliable distributed systems. There is no wall-clock time in distributed systems, so we use a Logical Clock to capture causality (happened-before) between processes.

Evolution of Hadoop

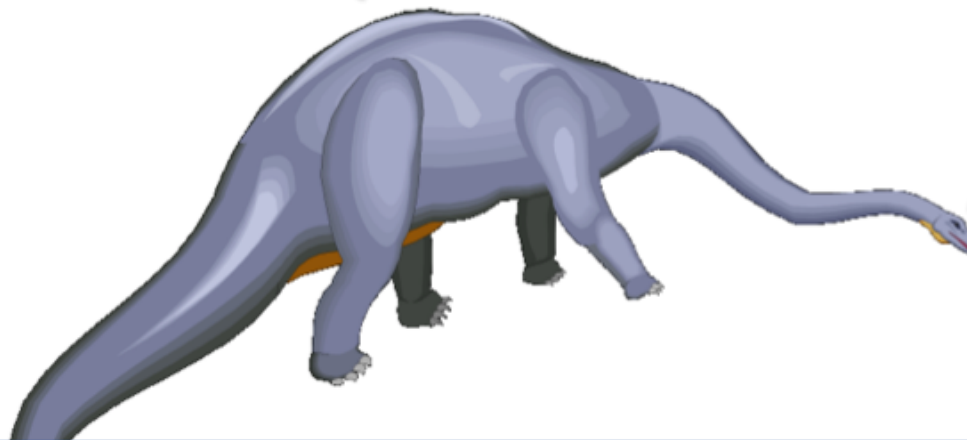
Huge Body (DataNodes)

Tiny Brain
(NameNode, ResourceMgr)

?

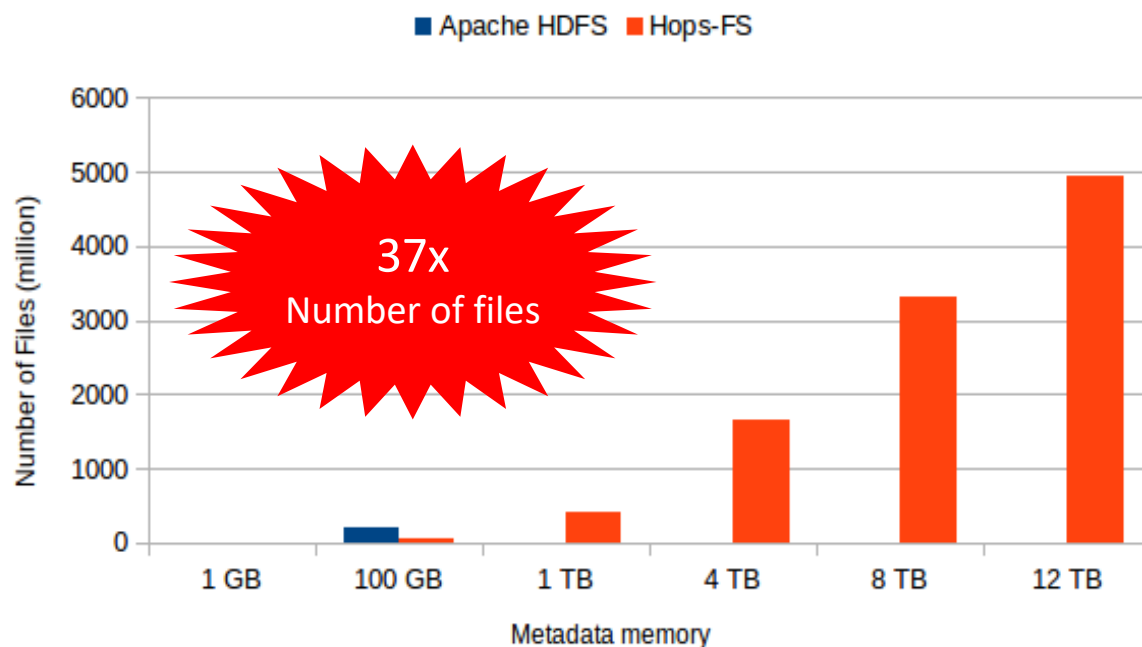


2009

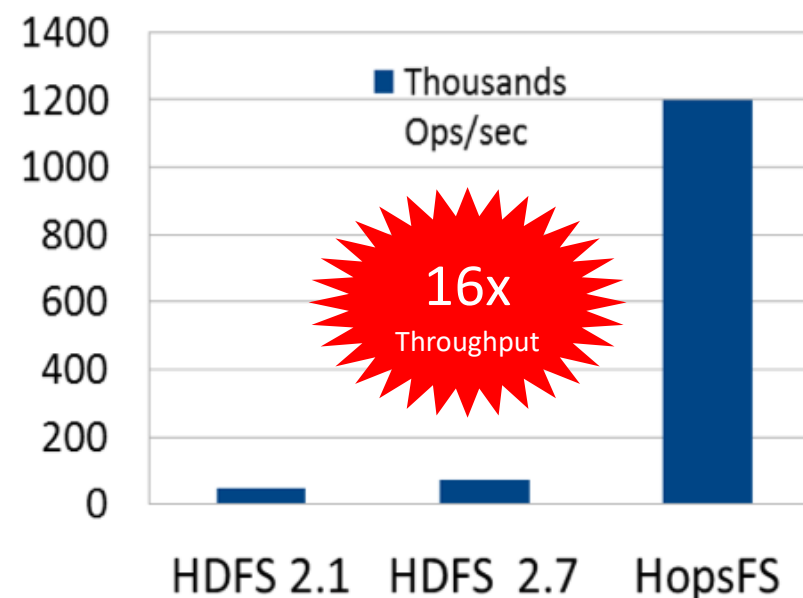


2018

HopsFS: Next Generation HDFS*



Bigger



Faster



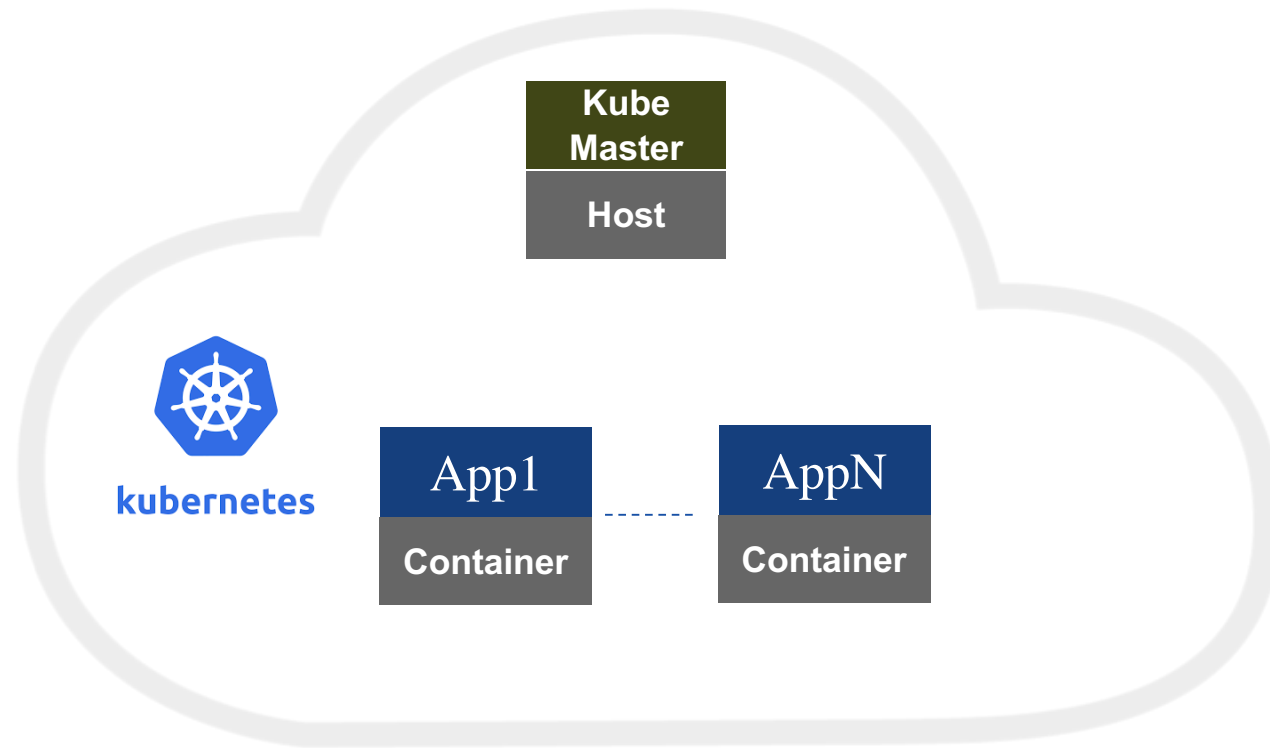
Scale Challenge Winner (2017)

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

NOBODY CARES!



What about our Cloud-Native Future?



Object Stores, as they are today, are not the Future!

The Problems with Object Stores*

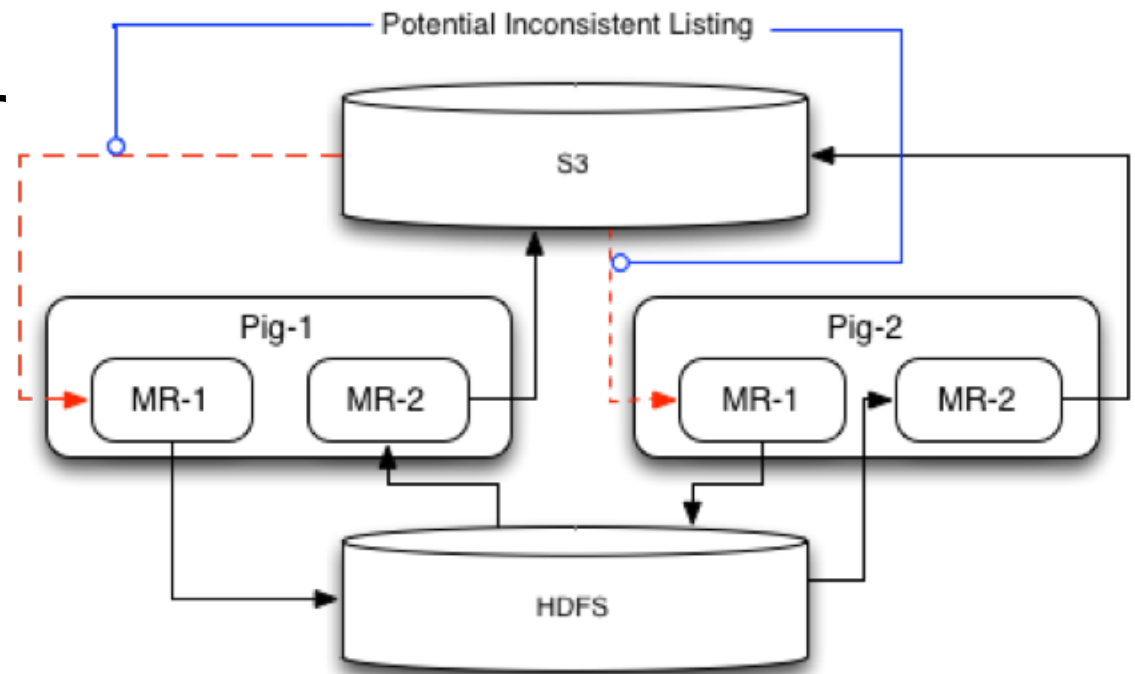
- Object Stores are inferior to hierarchical filesystems
 - False equivalence*
- Object stores can scale better, but at what cost?
 - Read-your-writes existing objs on S3 <#>
 - Write, then list
 - No Hierarchical namespace properties
 - Quotas, permissions
 - Slow rename operations

*False Equivalence**: “unfairly equating a minor failing to a major failing.”

How Bad are existing Object Stores?

Big Companies are implementing their own eventually consistent metadata store to mirror the (unobservable) metadata in S3/GCS

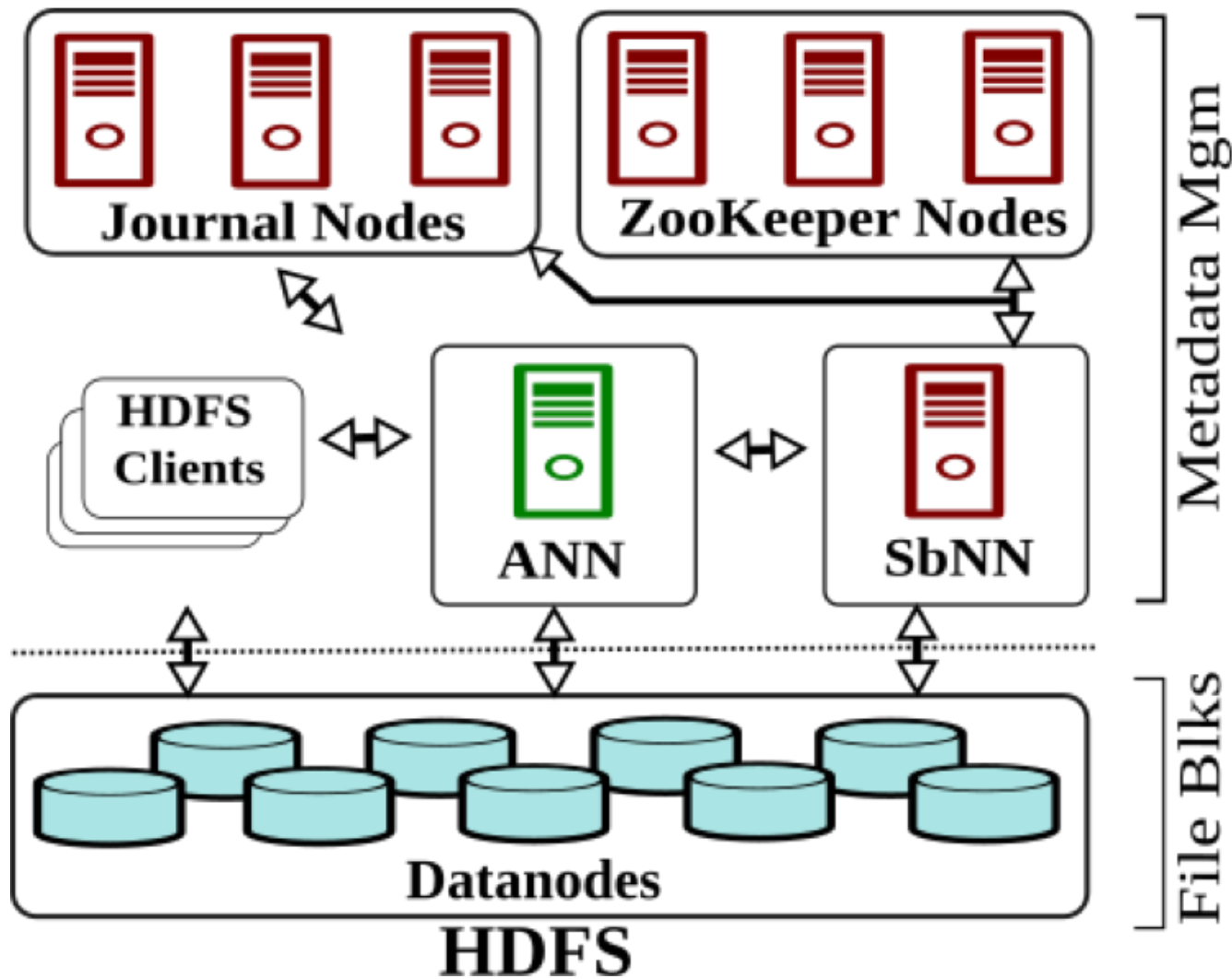
- Netflix for AWS
 - s3mpr
- Databricks DBFS
- EmrFS



<http://techblog.netflix.com/2014/01/s3mpr-consistency-in-cloud.html>

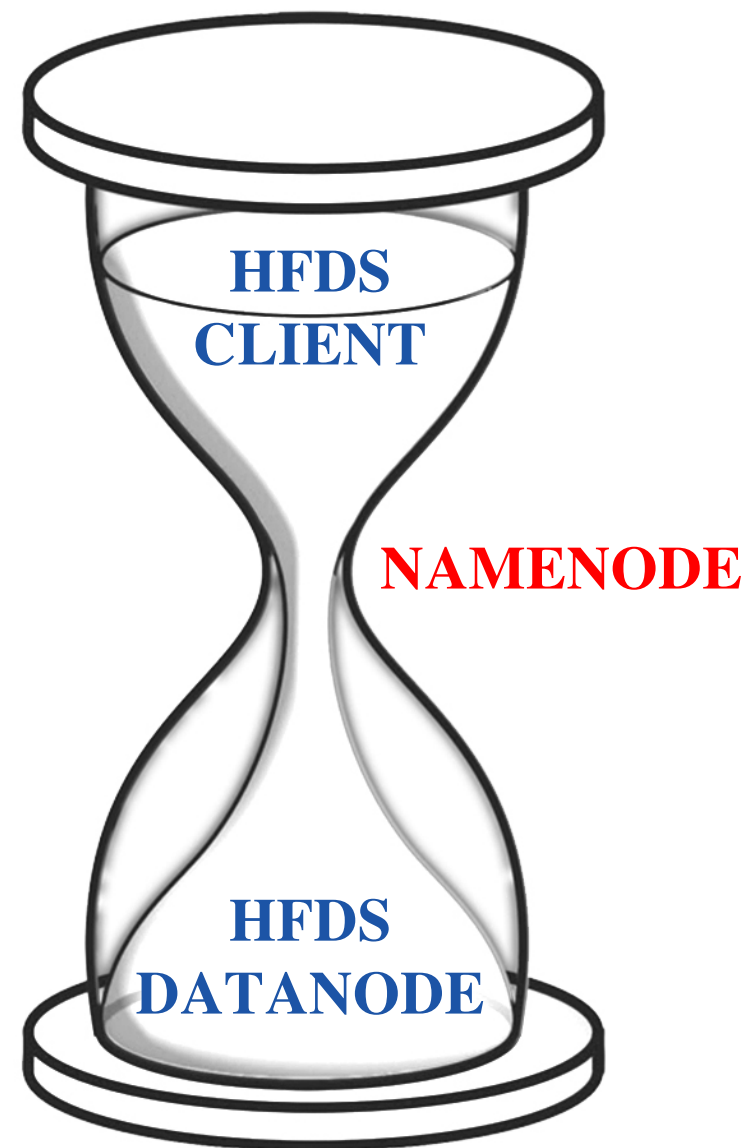
HDFS: A POSIX-Style Distributed FS

Apache HDFS



HDFS Scalability Limitations

- Limited namespace
 - JVM Heap (~300 GB)
- Limited concurrency
 - Single global lock
(single-writer, multiple readers)



NameNode in the Tragedy of the Commons



facebook

Single-Host
NameNode



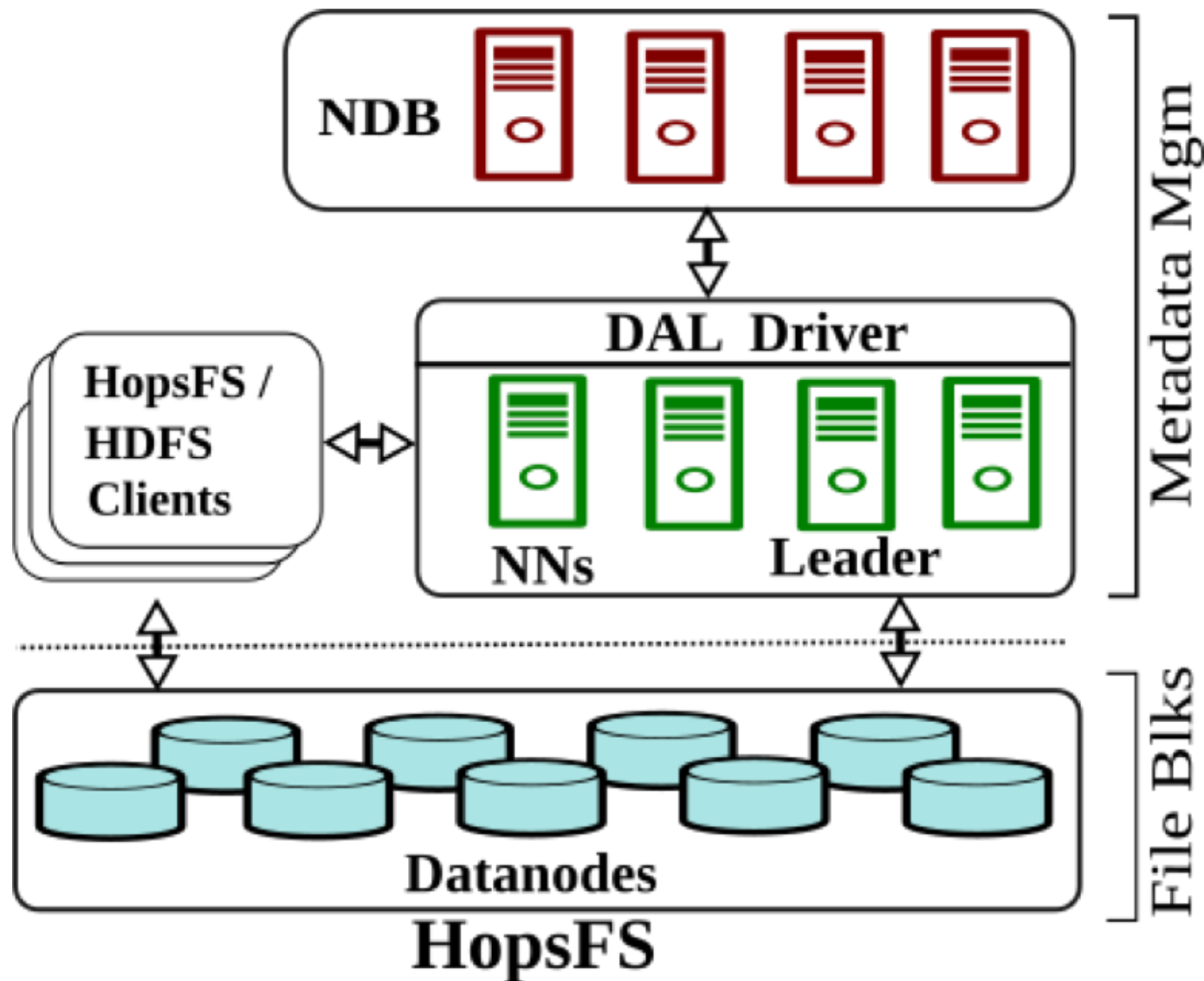
cloudera

YAHOO!

HopsFS

- Scale-out Metadata
 - Metadata in an in-memory distributed database
 - Multiple stateless Namenodes
- Remove the Global Metadata Lock
 - Multiple concurrent read and write operations

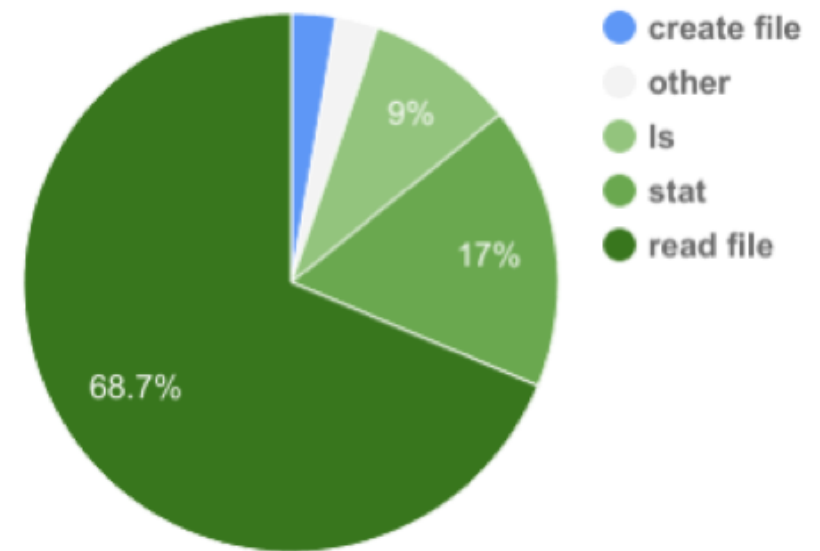
HopsFS Architecture



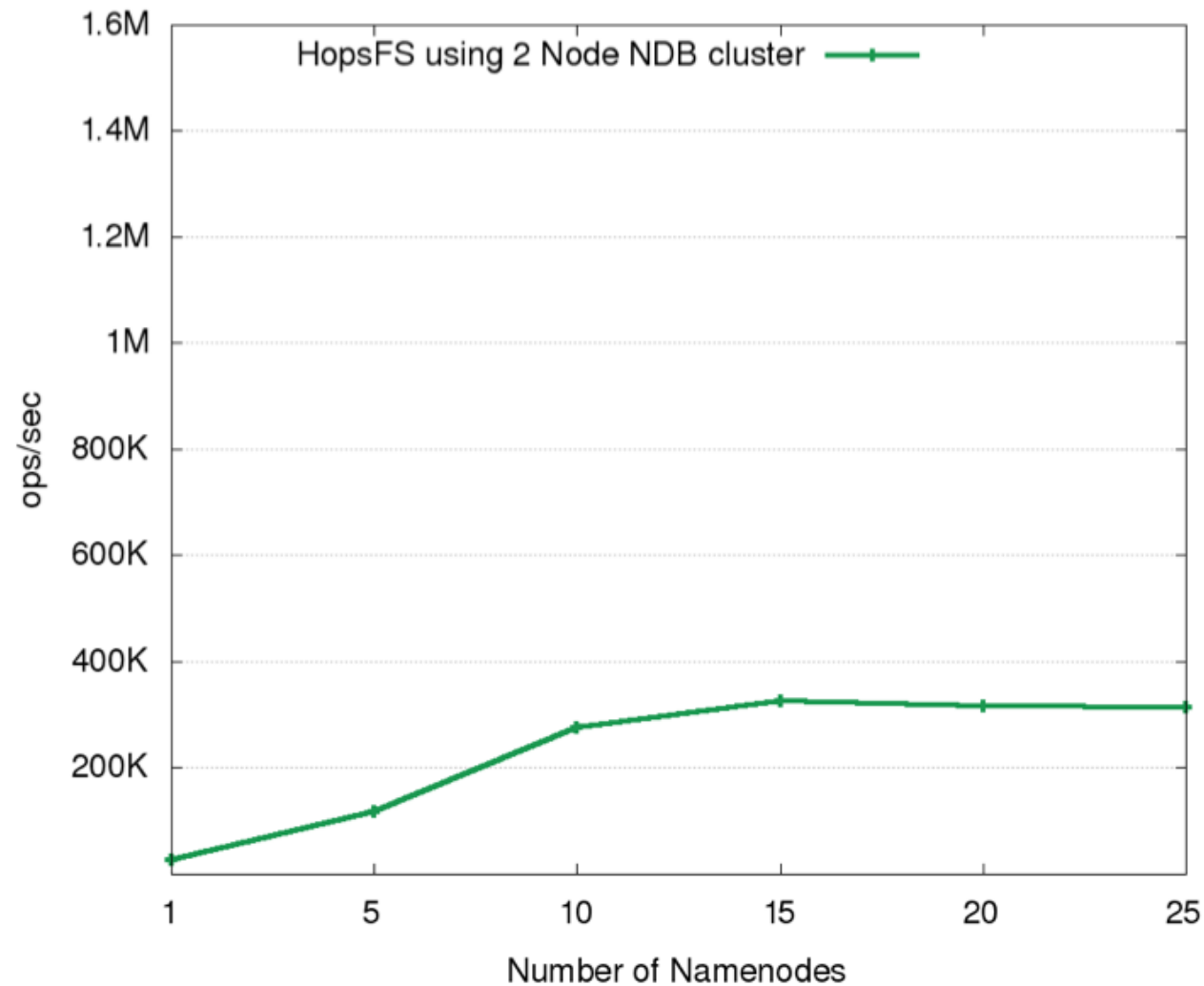
HopsFS Evaluation

- On Premise
 - Dual Intel® Xeon® E5-2620 v3 @2.40GHz
 - 256 GB RAM, 4 TB Disks
- 10 GbE
 - 0.1 ms ping latency between nodes

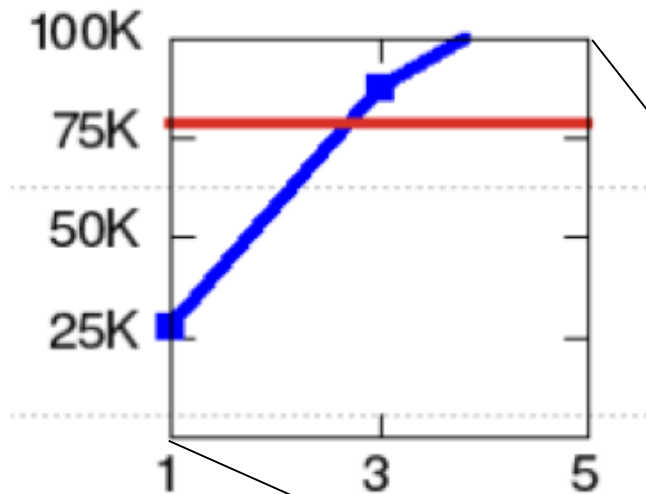
Spotify's HDFS Workload



Evaluation: Spotify Workload

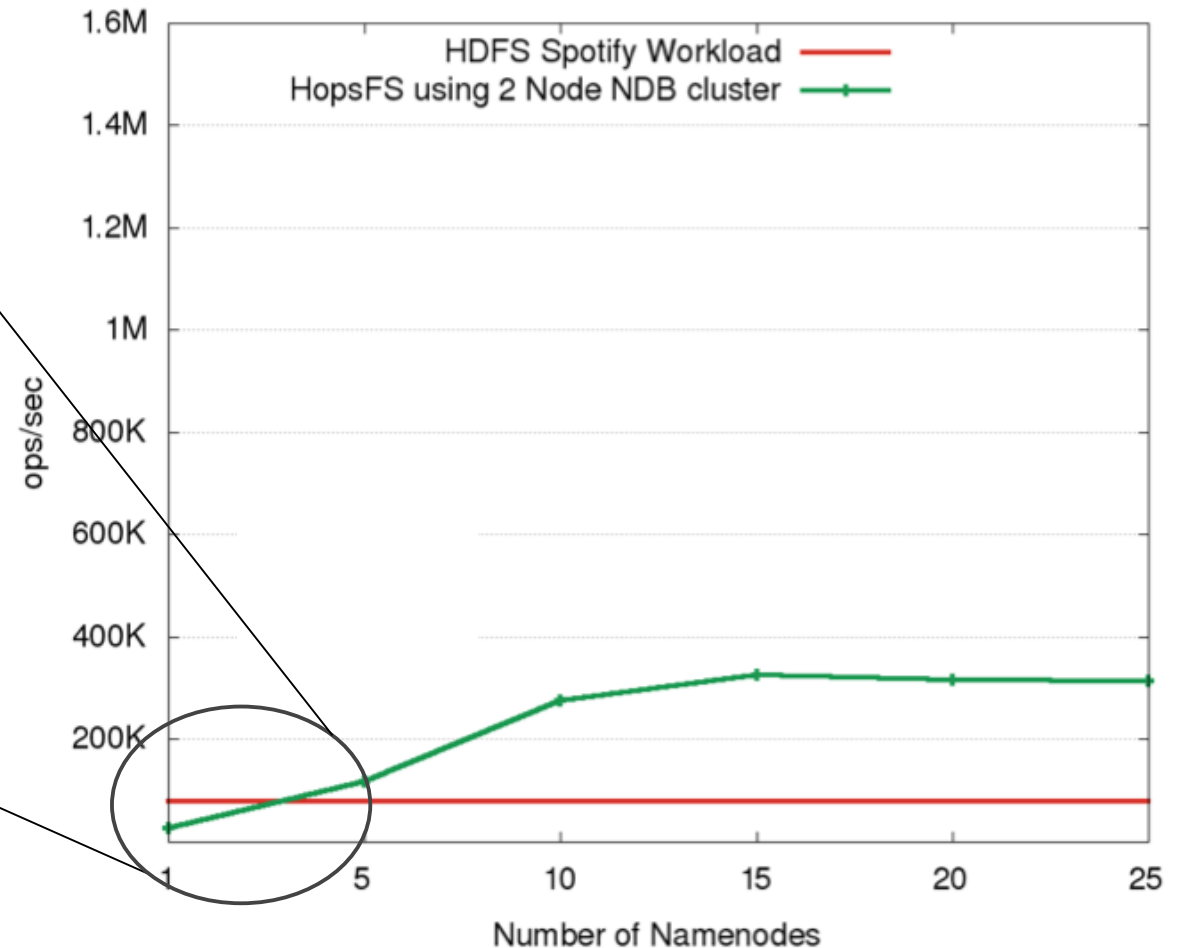


HopsFS Higher Throughput with Same Hardware

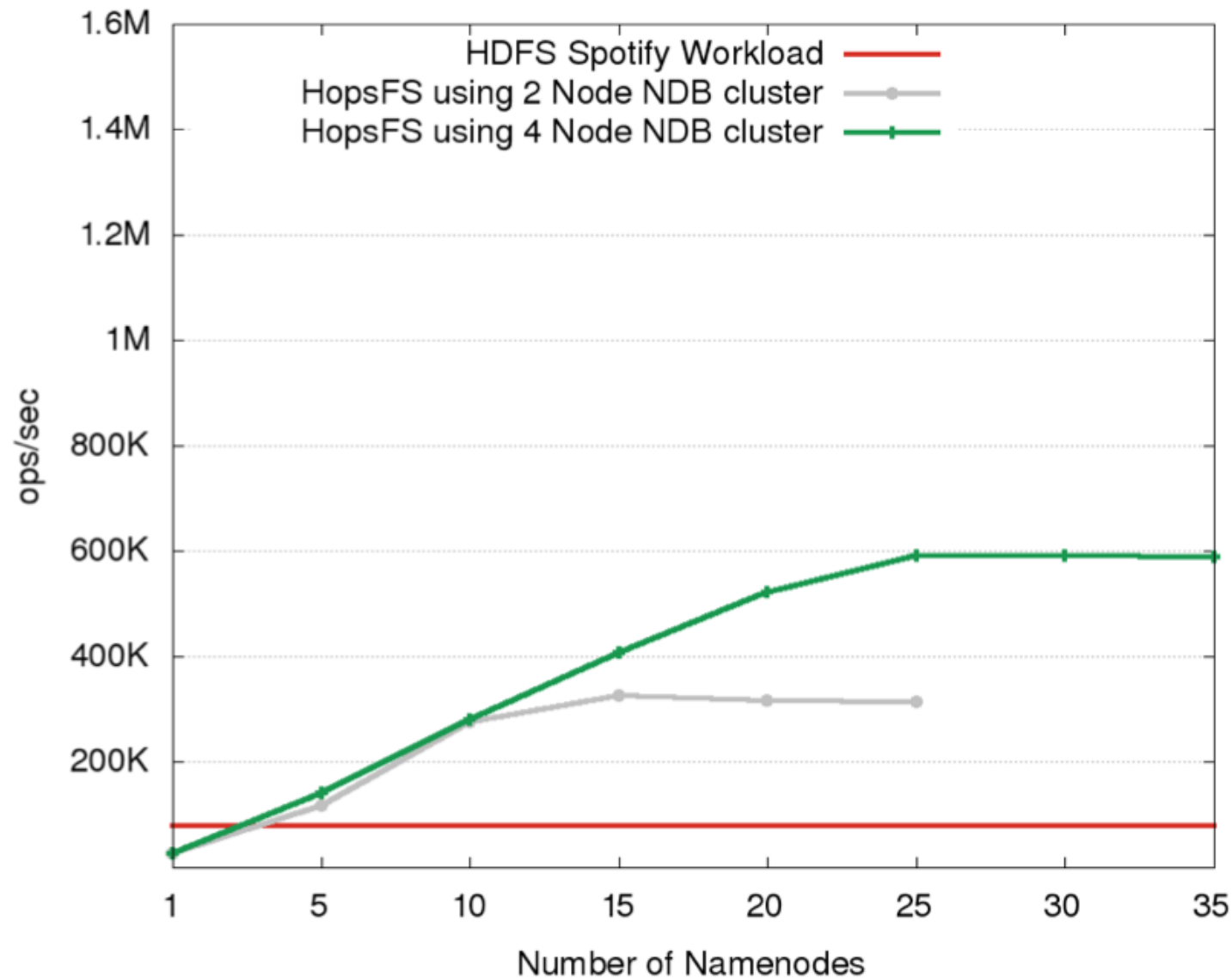


HopsFS outperforms with equivalent hardware: HA-HDFS with **Five** Servers

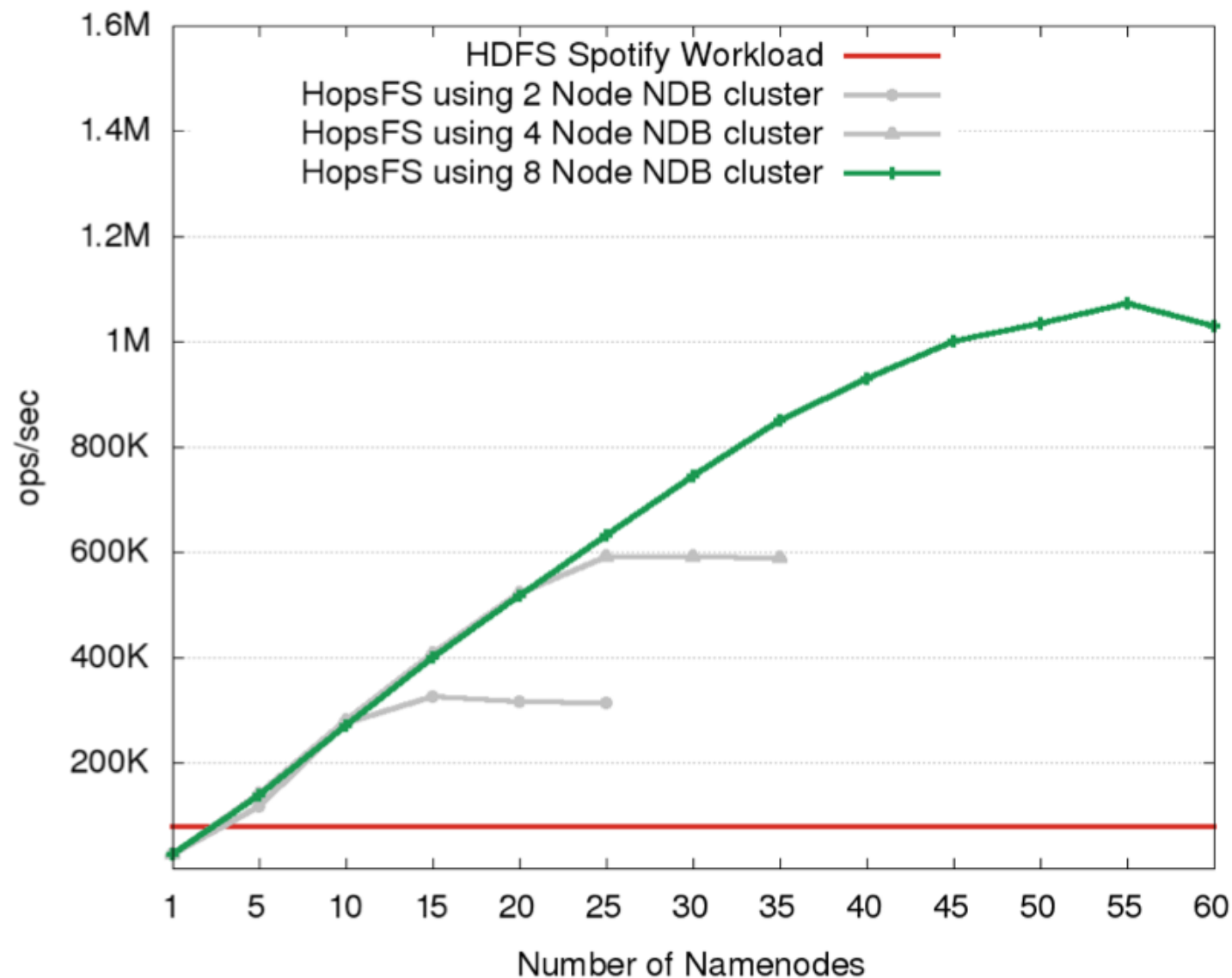
- 1 Active Namenode
- 1 Standby NameNode
- 3 Servers
 - Journal Nodes
 - ZooKeeper Nodes



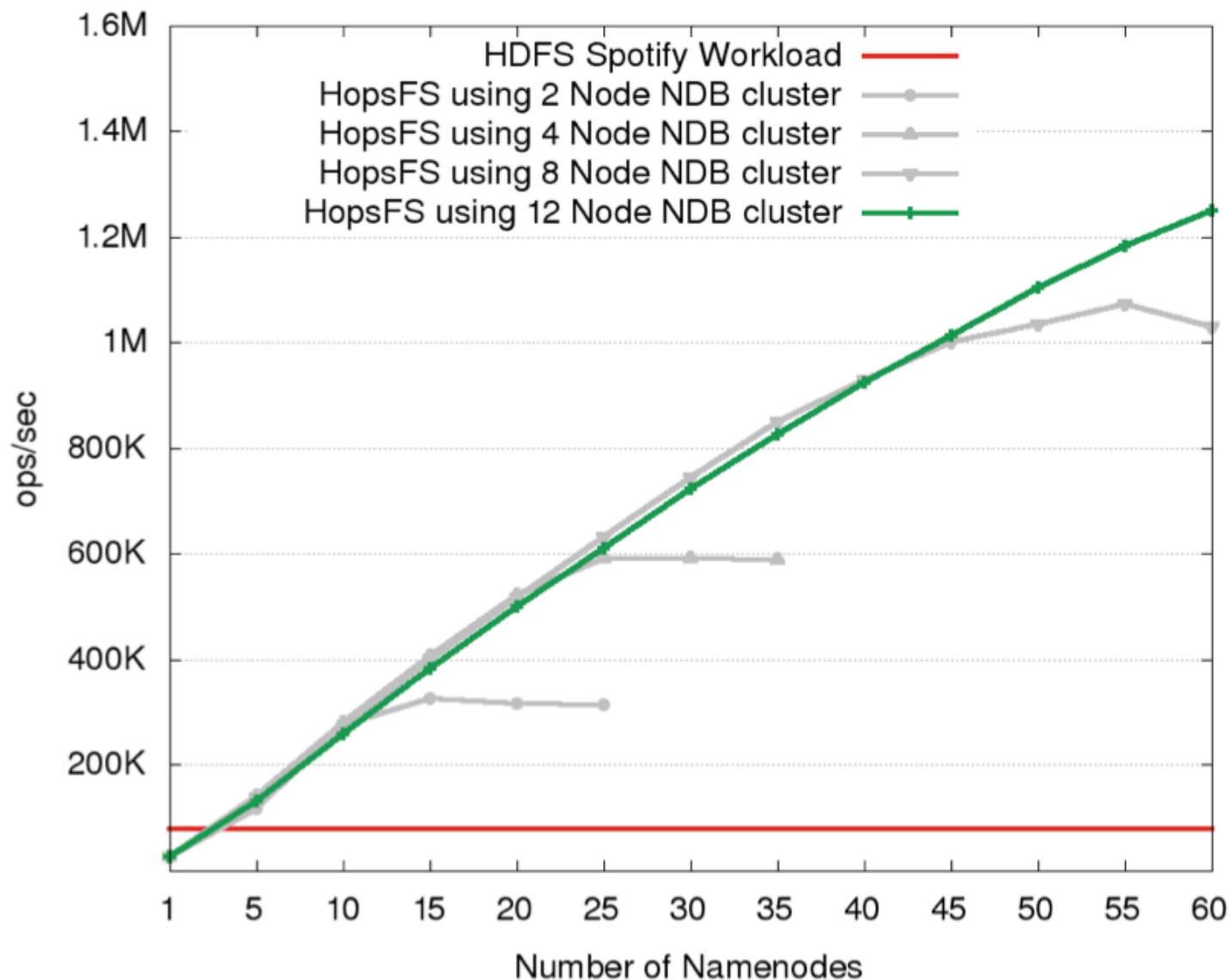
Evaluation: Spotify Workload (contd.)



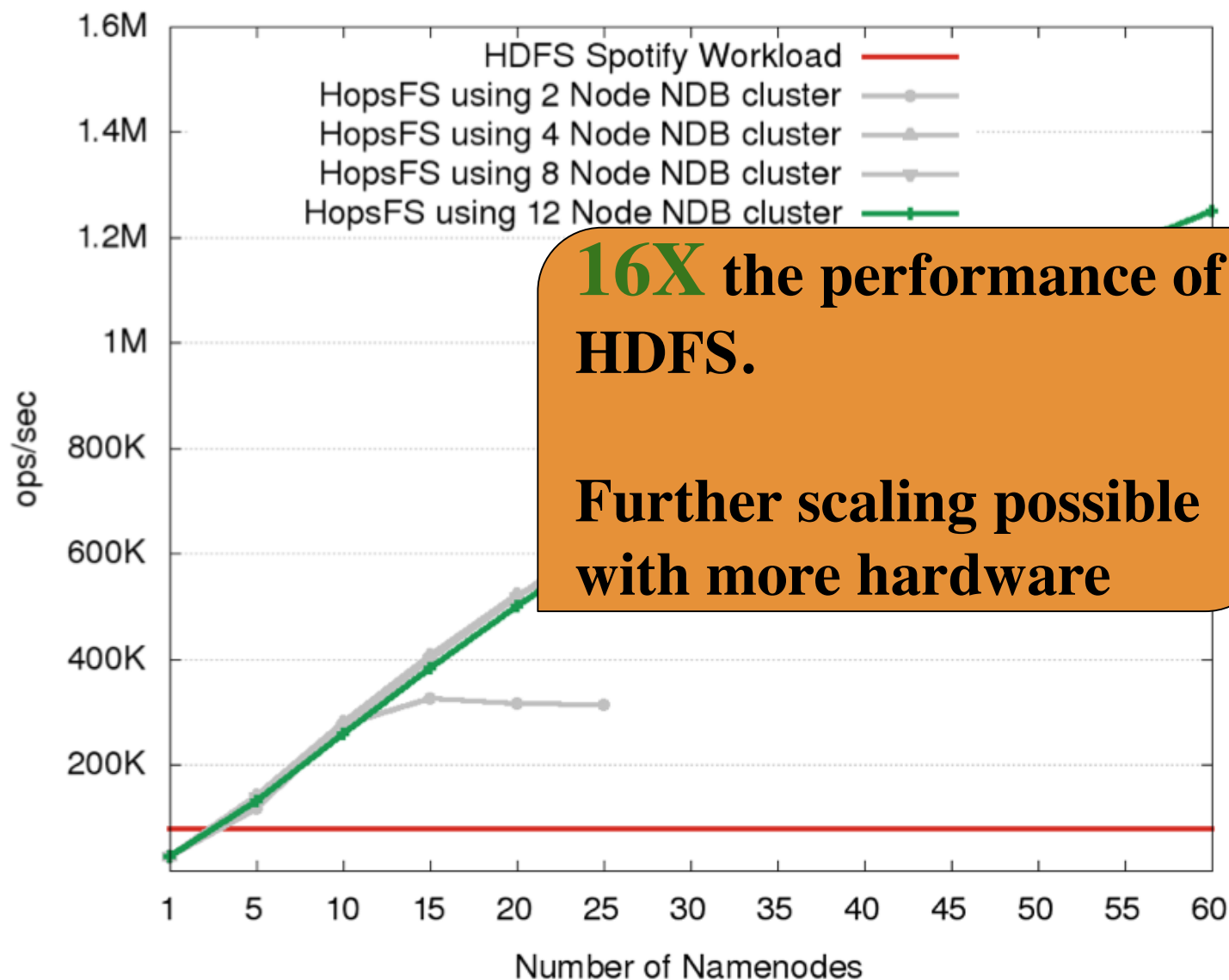
Evaluation: Spotify Workload (contd.)



Evaluation: Spotify Workload (contd.)



Evaluation: Spotify Workload (contd.)



Write Intensive workloads



Workloads	HopsFS ops/sec	HDFS ops/sec	Scaling Factor
Synthetic Workload (5.0% File Writes)	1.19 M	53.6 K	22
Synthetic Workload (10% File Writes)	1.04 M	35.2 K	30
Synthetic Workload (20% File Writes)	0.748 M	19.9 K	37

Scalability of HopsFS and HDFS for write intensive workloads

Write Intensive workloads

Workloads	HopsFS ops/sec	HDFS ops/sec	Scaling Factor
Synthetic Workload (5.0% File Writes)	1.19 M	53.6 K	22
Synthetic Workload (10% File Writes)	1.04 M	35.2 K	30
 Synthetic Workload (20% File Writes)	0.748 M	19.9 K	37

Scalability of HopsFS and HDFS for write intensive workloads

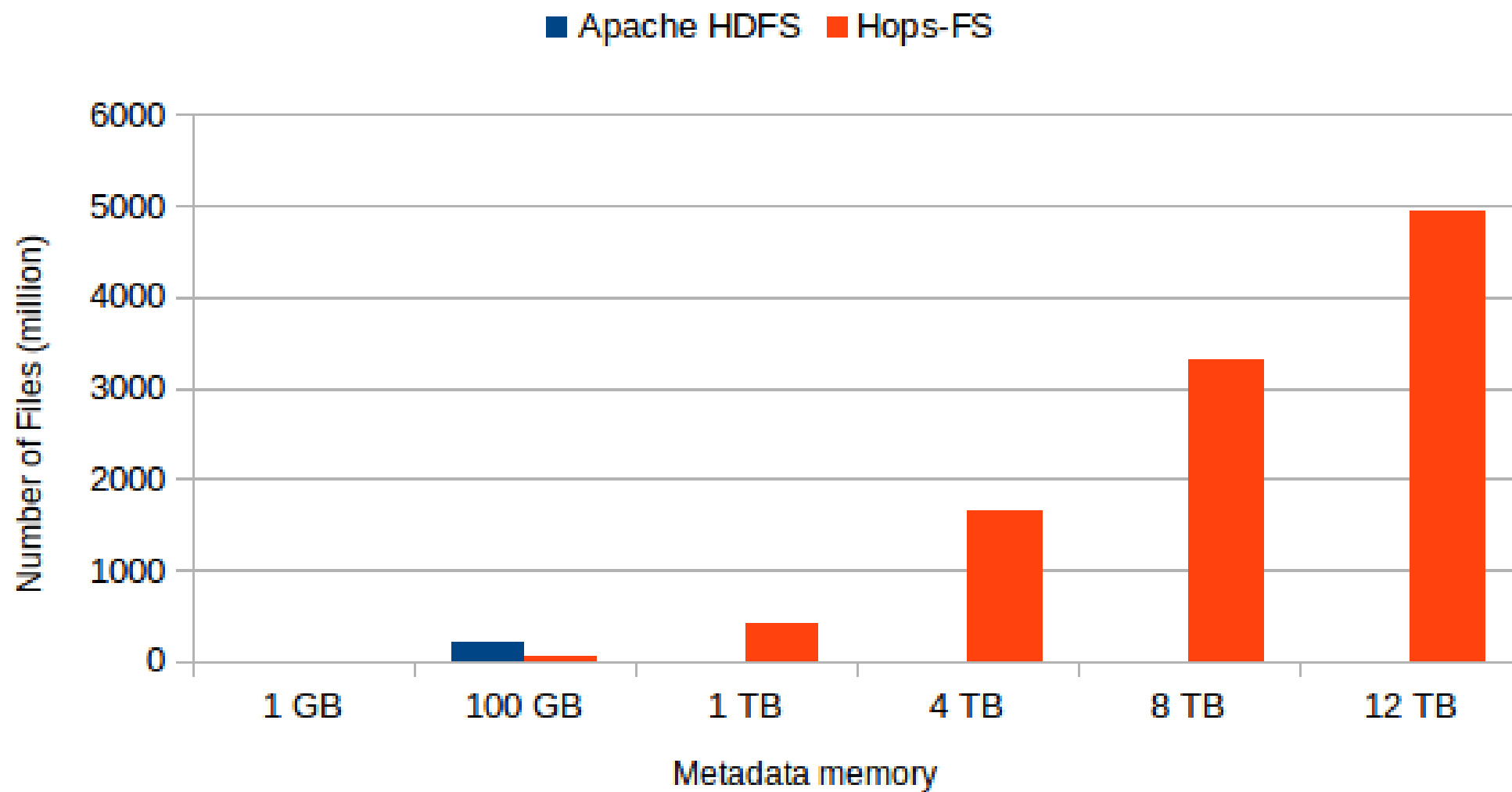
Metadata Scalability

HDFS Metadata **200 GB** → Max **500 million** files/directories

HopsFS Metadata **24 TB** → **Max 17 Billion** files/directories

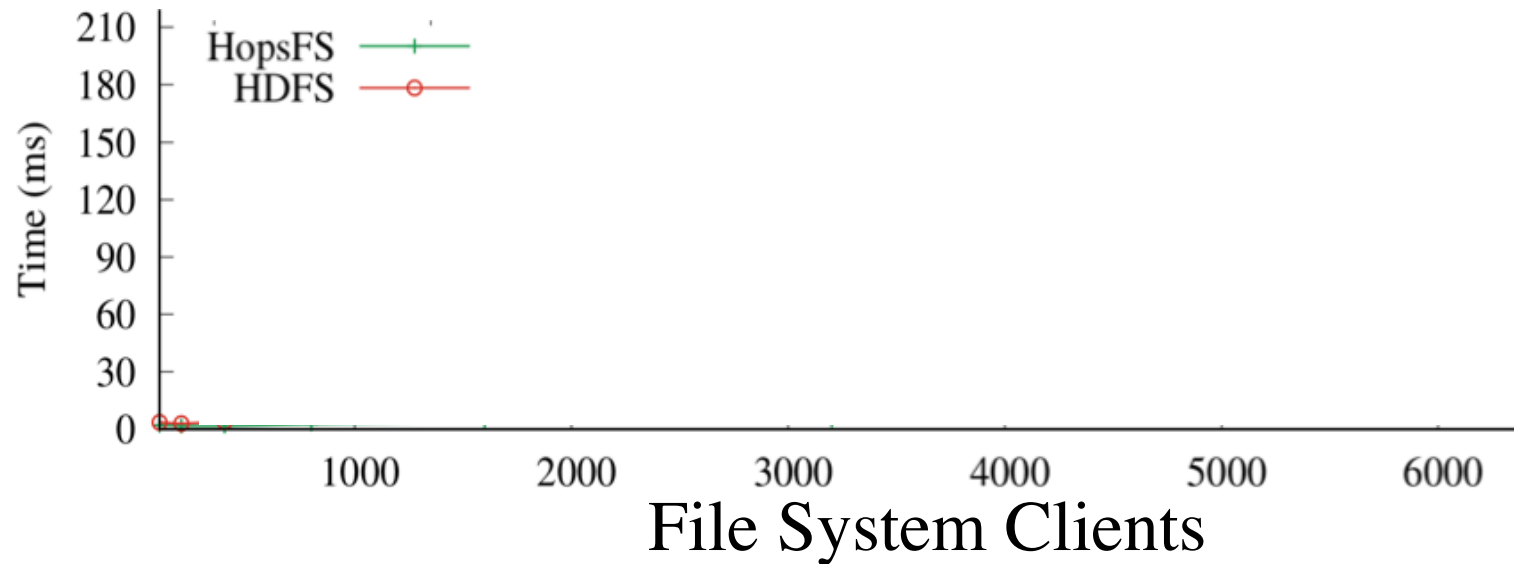
□ **37 times** more files than HDFS

HopsFS Metadata Capacity



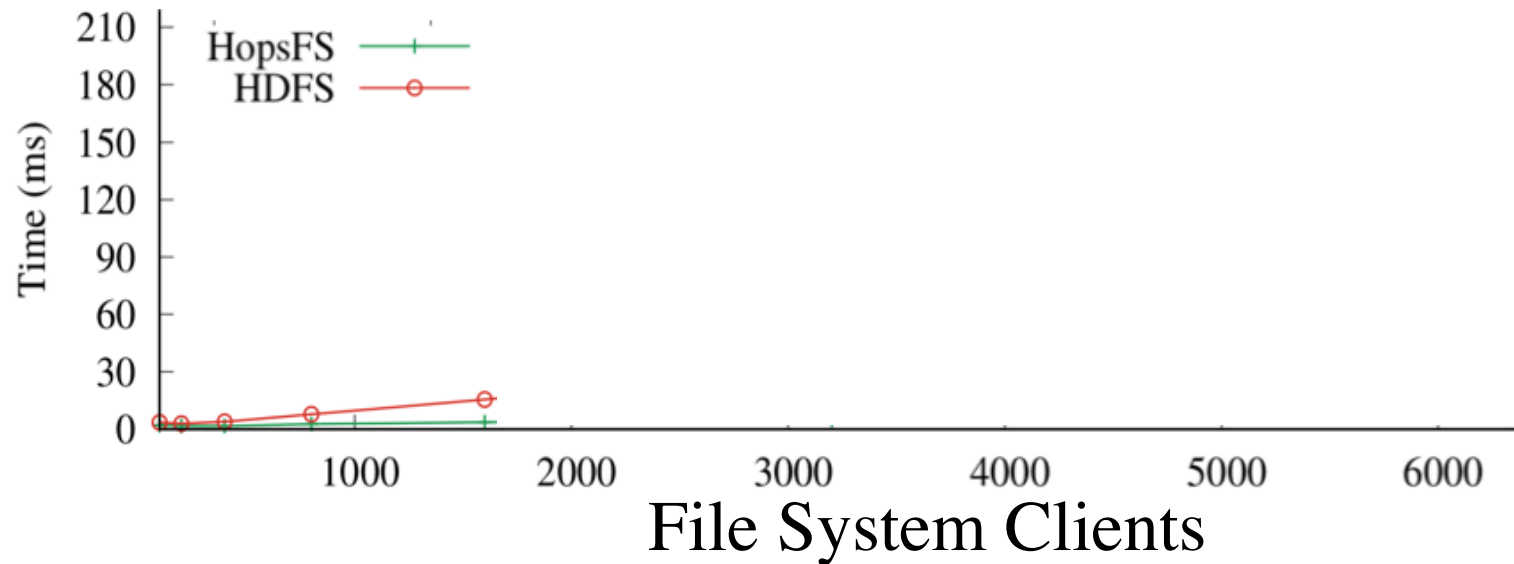
Assuming 256MB Block Size, 100 GB JVM Heap for Apache Hadoop

Operational Latency



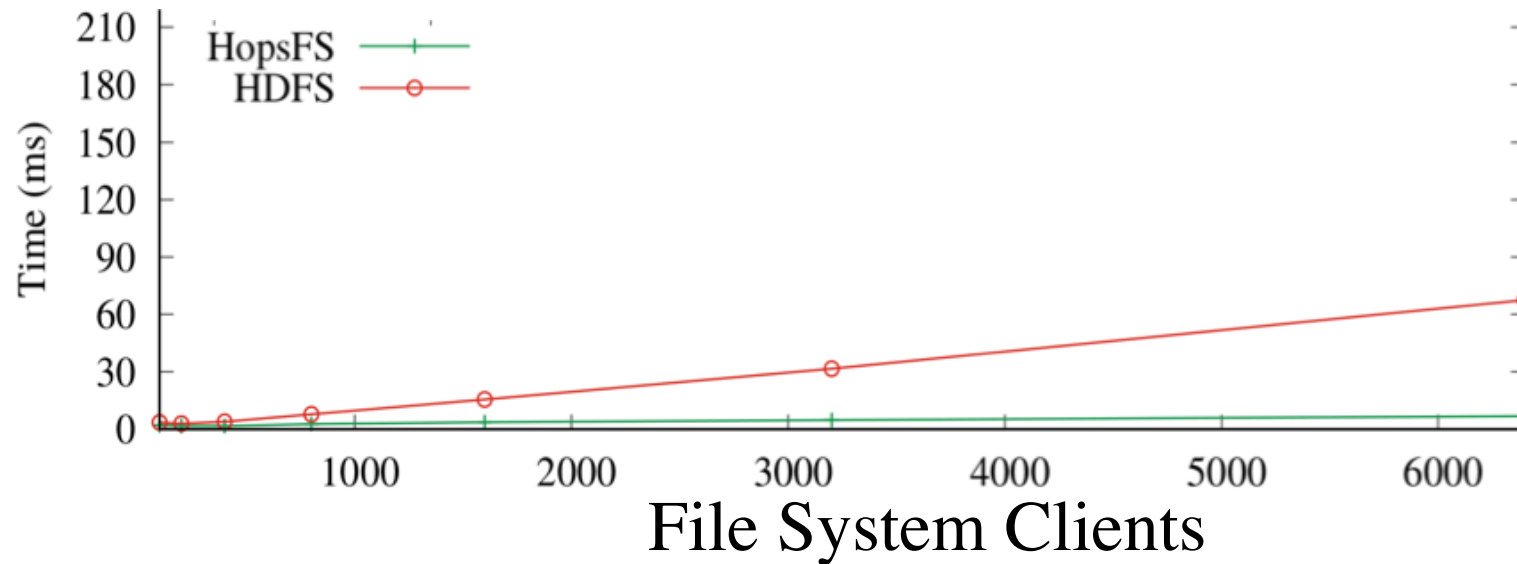
No of Clients	HopsFS Latency	HDFS Latency
50	3.0	3.1

Operational Latency



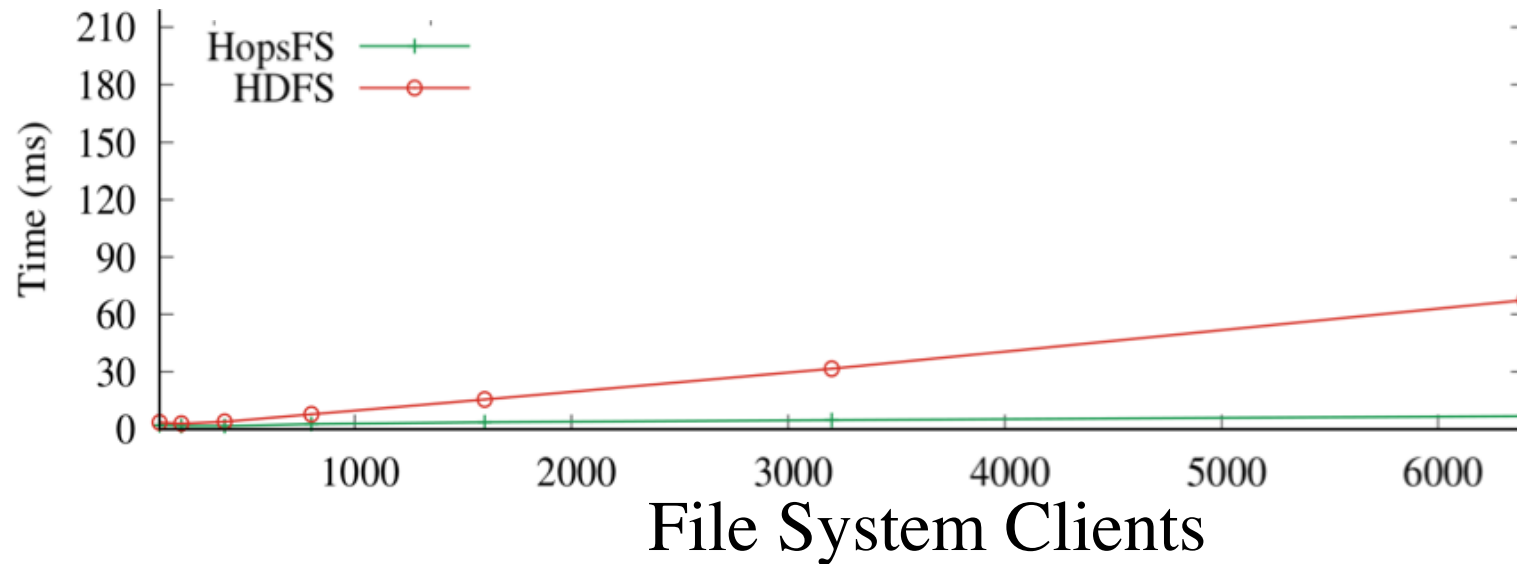
No of Clients	HopsFS Latency	HDFS Latency
50	3.0	3.1
1500	3.7	15.5

Operational Latency



No of Clients	HopsFS Latency	HDFS Latency
50	3.0	3.1
1500	3.7	15.5
6500	6.8	67.4

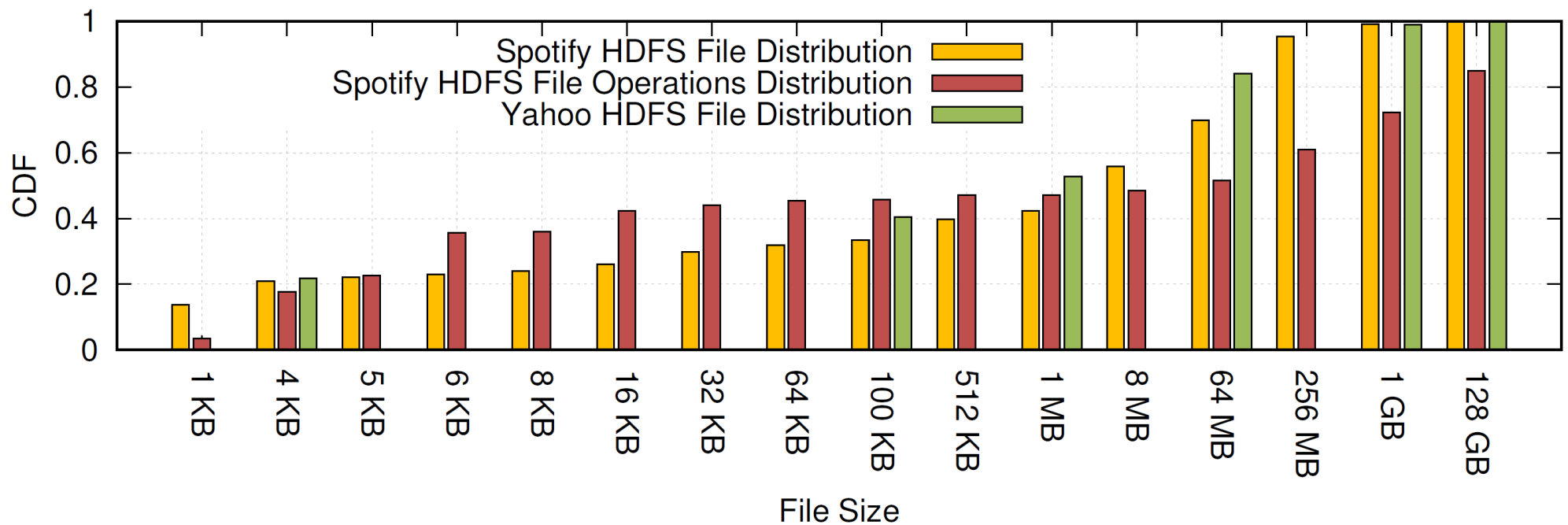
Operational Latency



For 6500 clients HopsFS has **10 times** lower latency than HDFS

Small Files in Hadoop

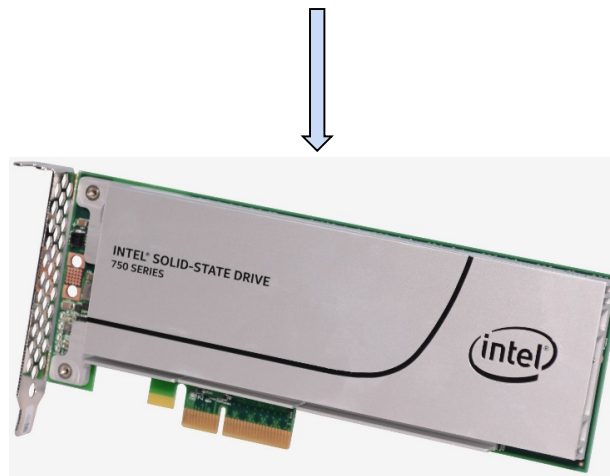
- In both Spotify and Yahoo 20% of the files are ≤ 4 KB



Small Files in HopsFS*

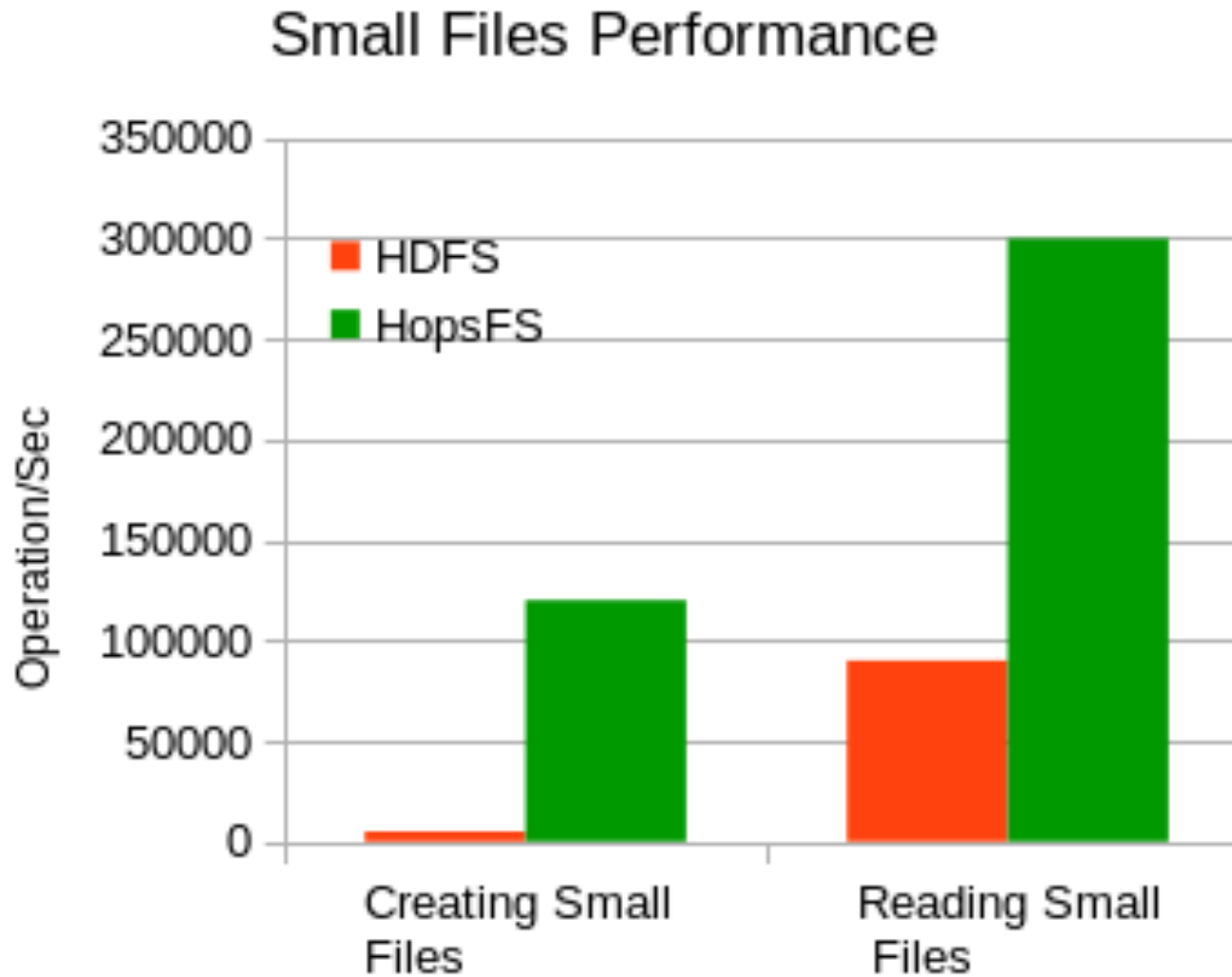
- In HopsFS, we can store small files co-located with the metadata in MySQL Cluster as on-disk data.

inode_id	varbinary (on-disk column)
32123432	/tmp/smallFile.txt



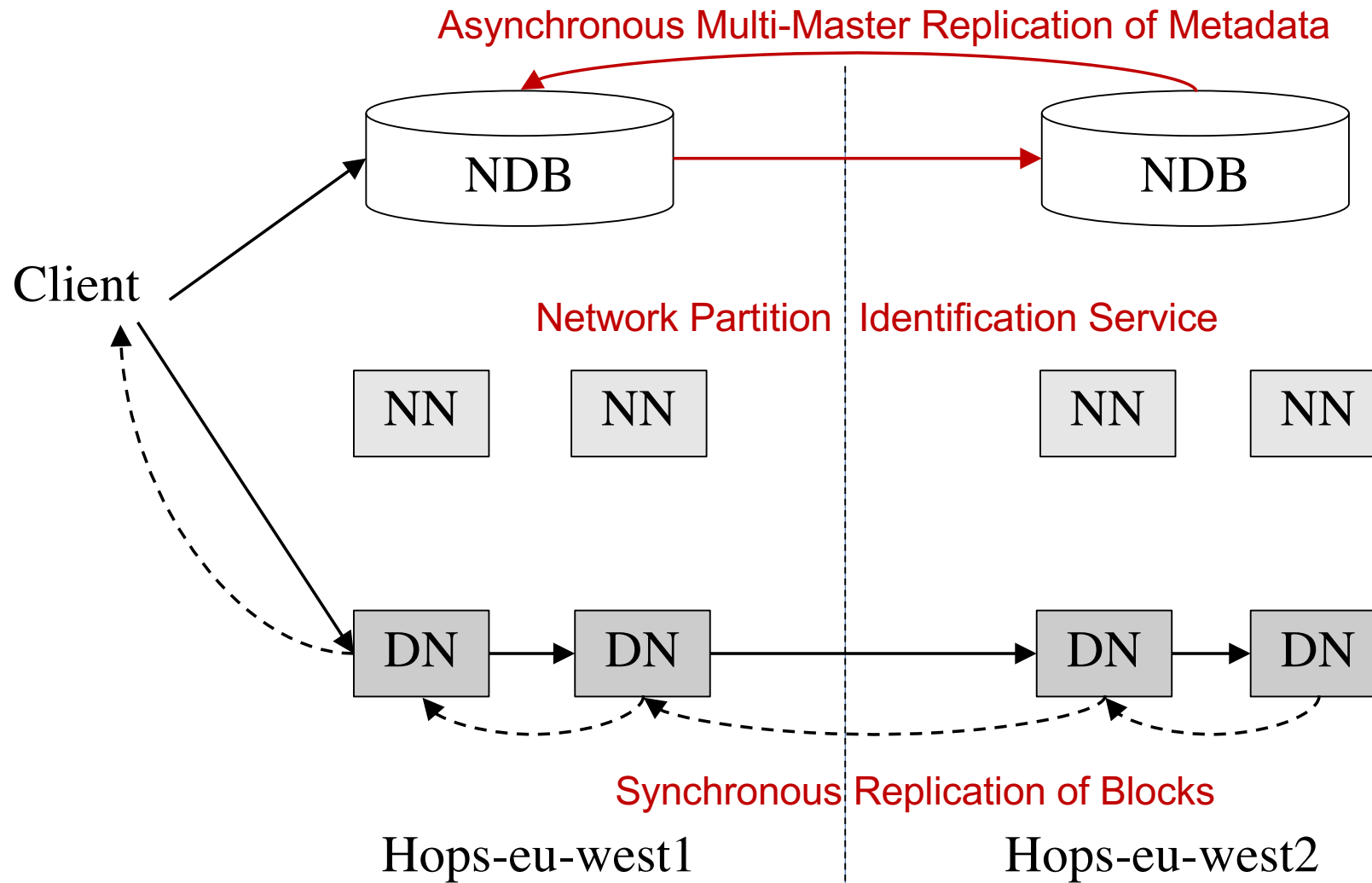
*Niazi et al, Size Matters: Improving the Performance of Small Files in HDFS, Poster at Eurosys 2017

HopsFS Small Files Performance



30 namenodes/datanodes and 6 NDB nodes were used. Small file size was 4 KB. HopsFs files were stored on Intel 750 Series SSDs

Ongoing work: Multi-Data-Center HopsFS



What do I do with all this Metadata?

Extending Metadata in Hops

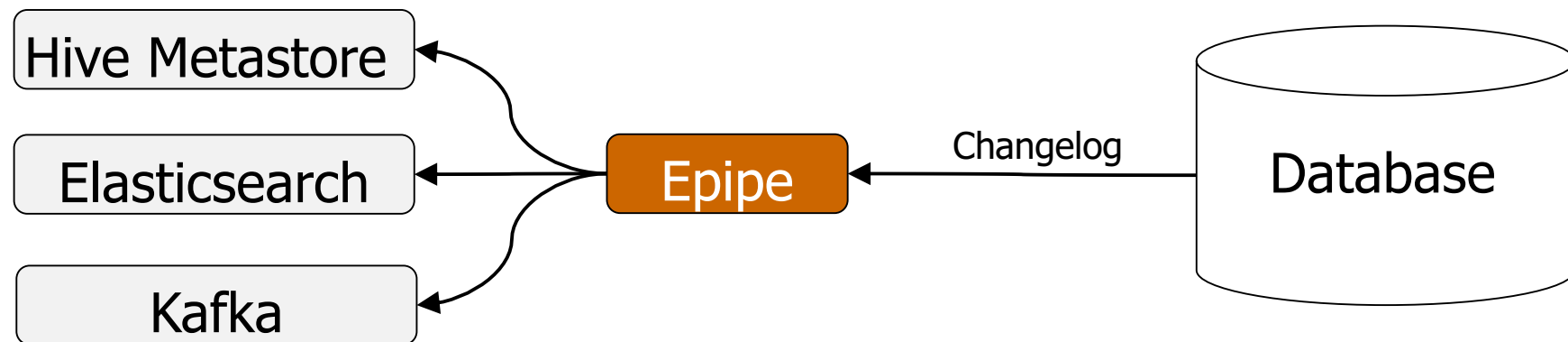
Metadata API

```
public void attachMetadata(Json obj, String pathToFileorDir)
public Json getMetadata(String pathToFileorDir)
public void removeMetadata(Json obj, String pathToFileorDir)
```

DIY Extended Metadata

- Add your own tables in the database
 - Foreign keys + transactions ensure metadata consistency

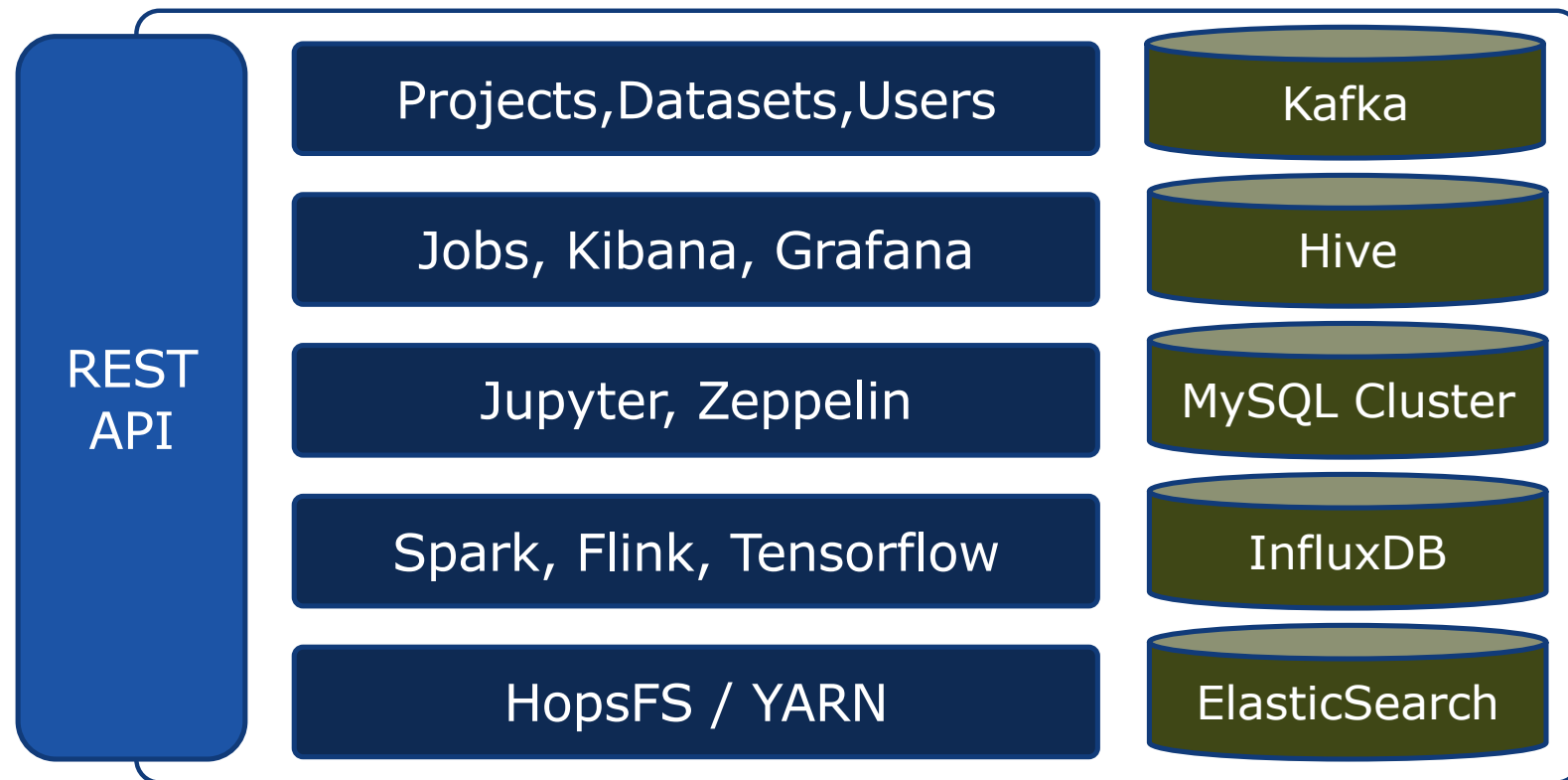
Strong Eventually Consistent Metadata



**Free-Text Search for Files/Dirs in
the HopsFS Namespace**

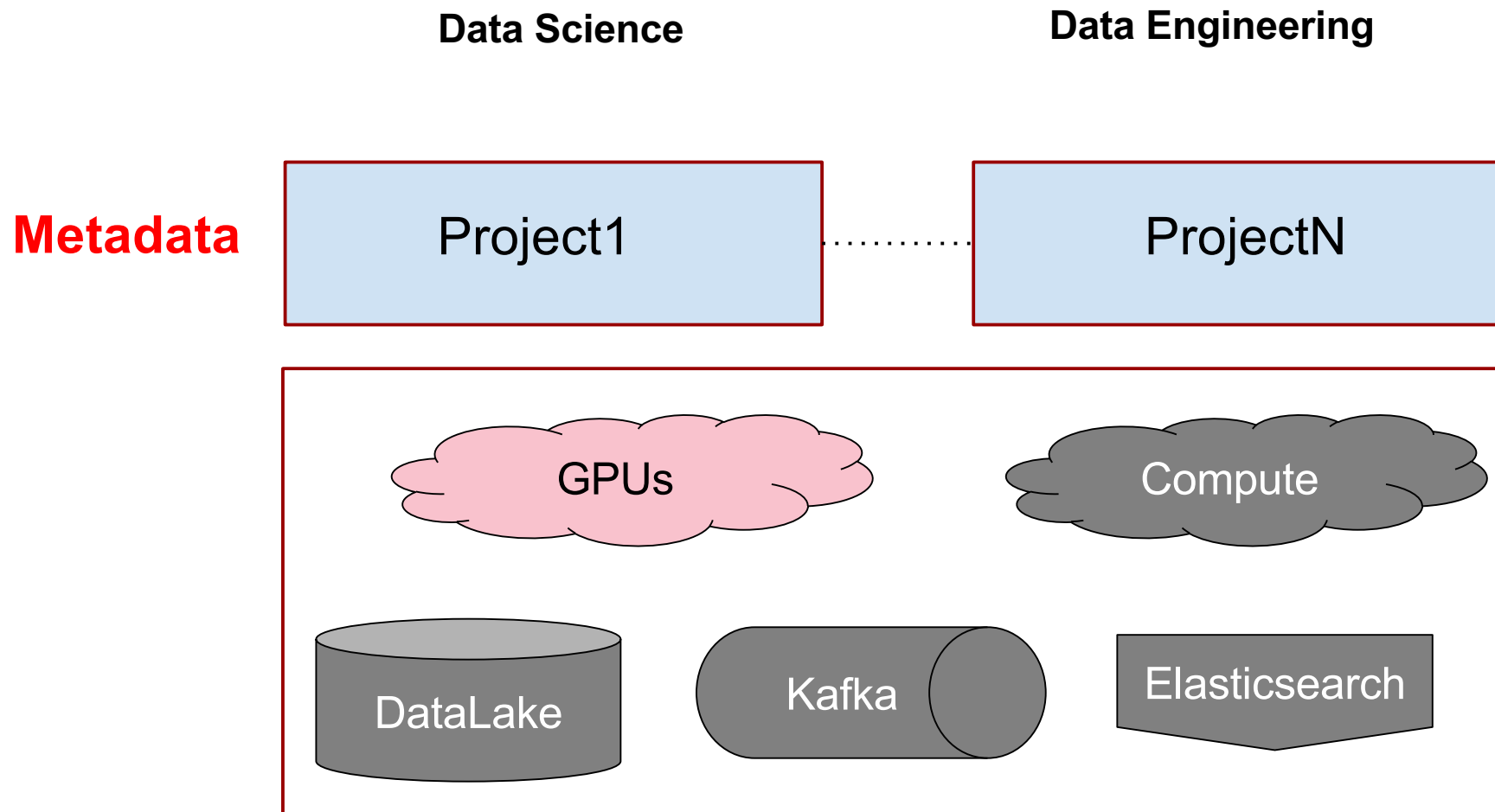
Re-imagine Hadoop with Big Metadata

Hopsworks = Big Data Science Platform



Java EE Application (AngularJS FrontEnd)

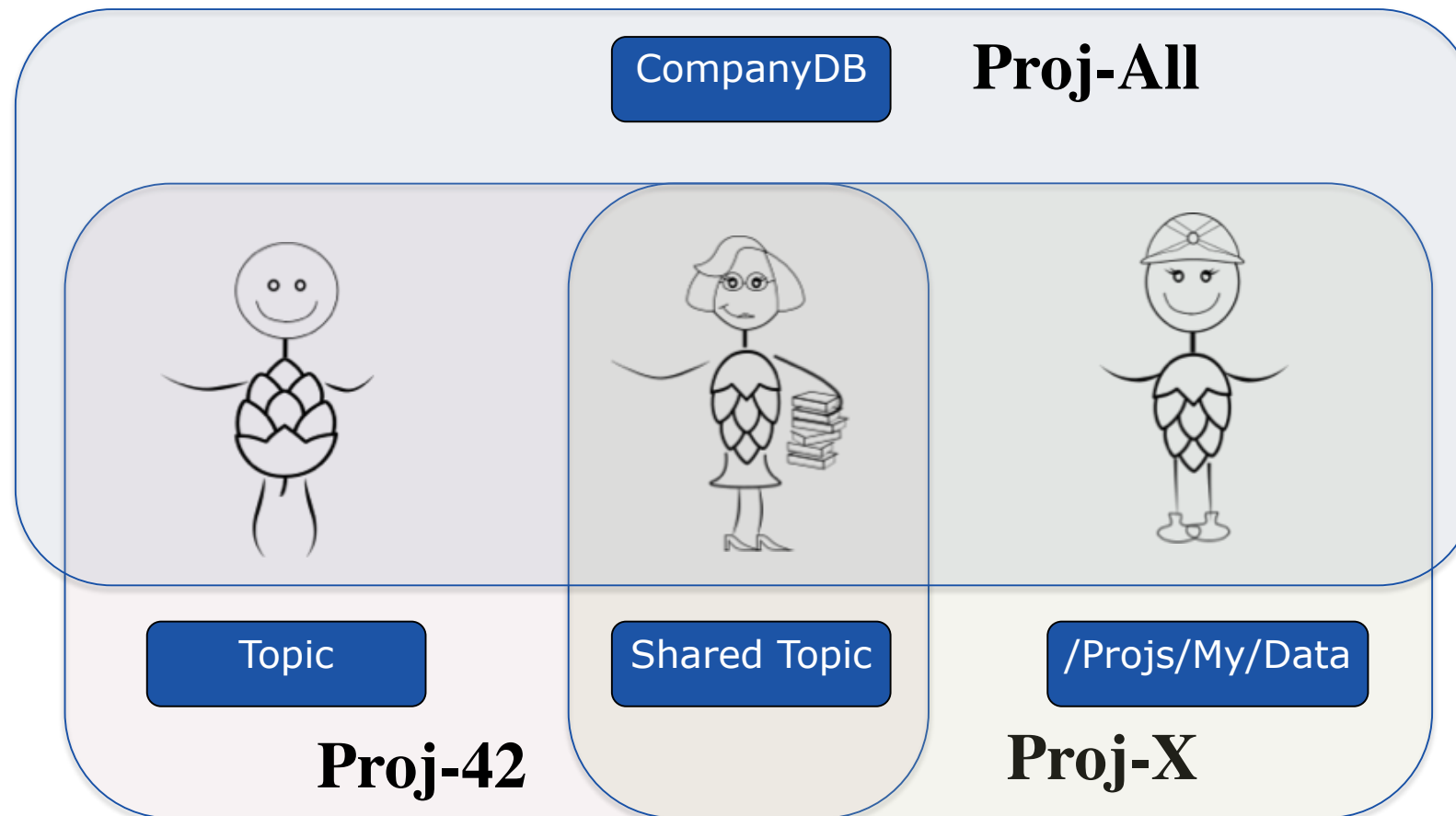
Projects in Hopsworks



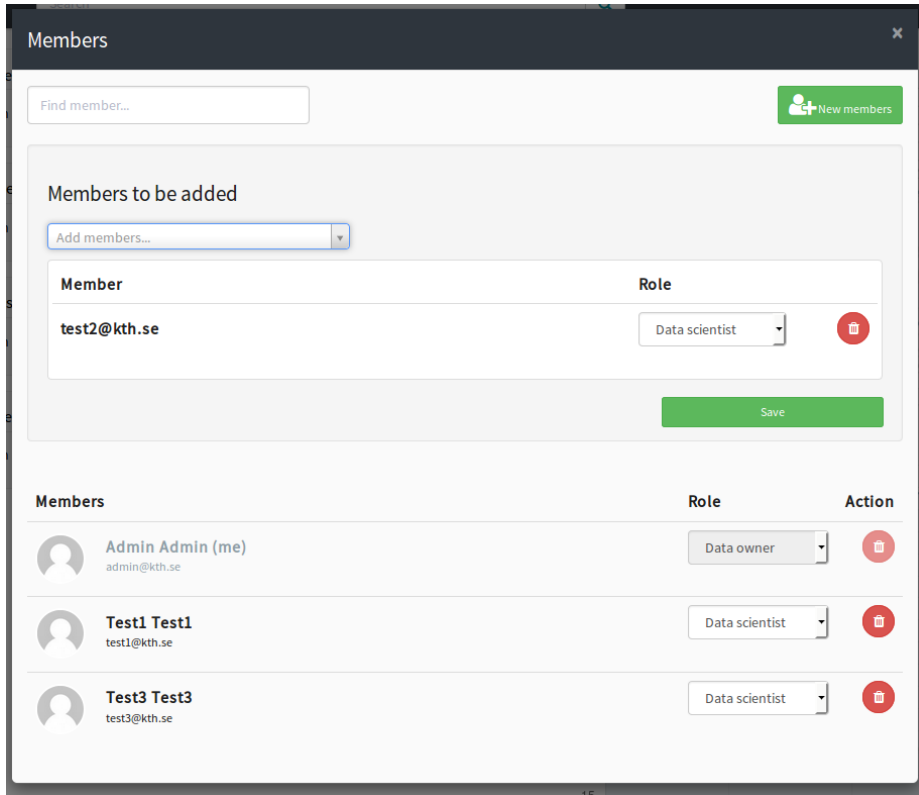
Only Hadoop Platform with Support for GPUs (Feb 2018)

Projects for Software-as-a-Service

A **Project** is a Grouping of **Users** and **Data**



Project Roles



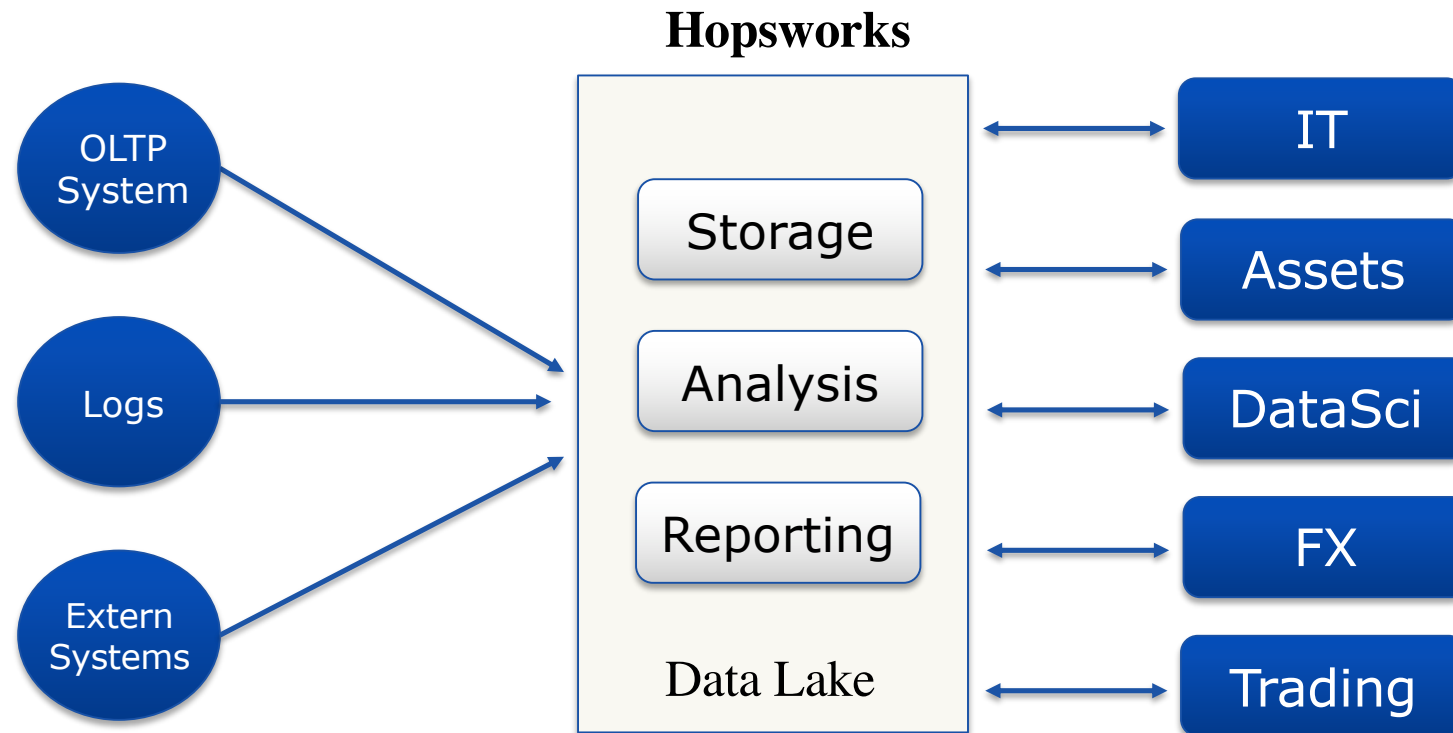
The screenshot shows a web interface for managing project members. At the top, there's a 'Find member...' search bar and a 'New members' button. Below this is a section titled 'Members to be added' containing an 'Add members...' dropdown. A form below the dropdown has a 'Member' field with the value 'test2@kth.se' and a 'Role' dropdown with the value 'Data scientist'. A 'Save' button is at the bottom right of this form. Below the form is a table of existing members.

Members	Role	Action
Admin Admin (me) admin@kth.se	Data owner	
Test1 Test1 test1@kth.se	Data scientist	
Test3 Test3 test3@kth.se	Data scientist	

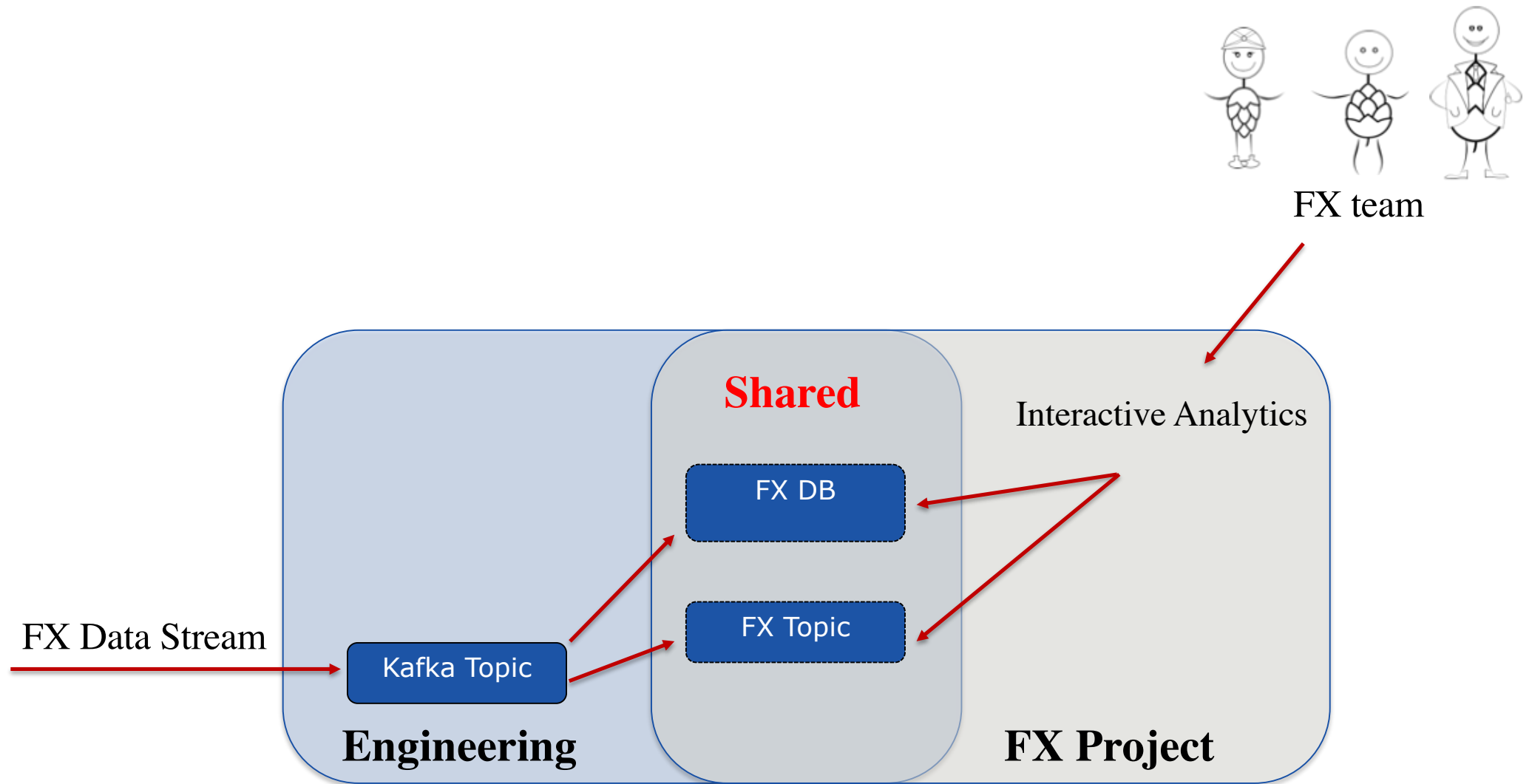
- Data Owner Privileges
 - Import/Export data
 - Manage Membership
 - Share DataSets, Topics
- Data Scientist Privileges
 - Write and Run code

We delegate administration of privileges to users

Multi-Tenant Project-Based Platform

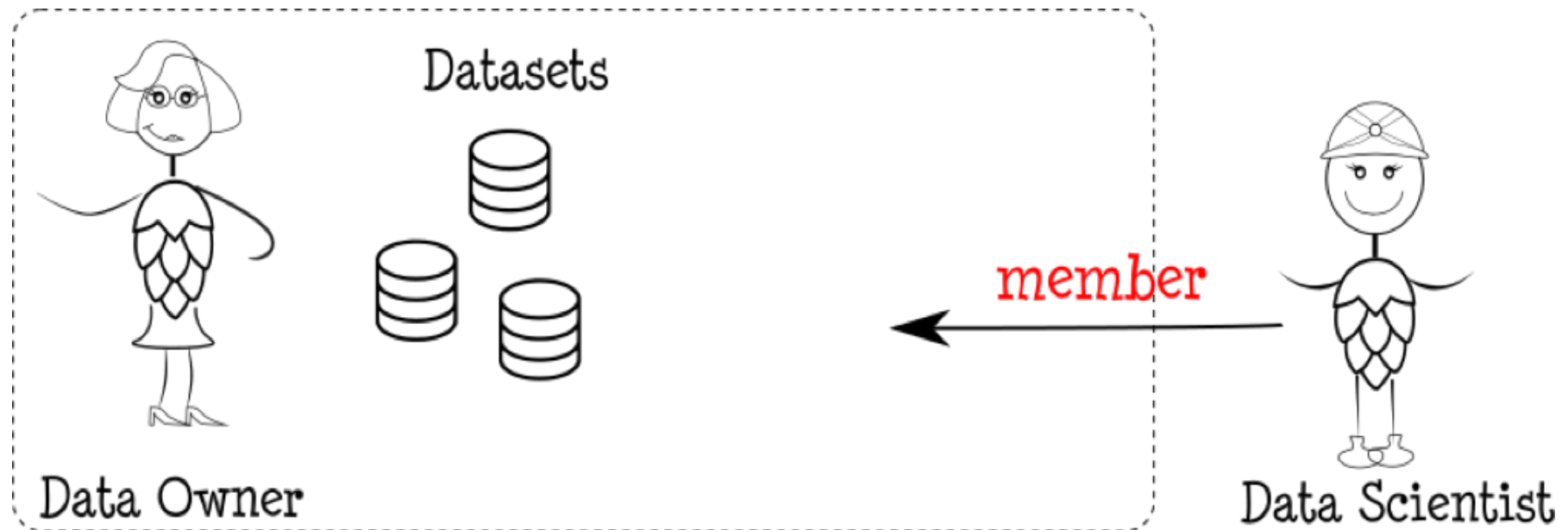


Hopsworks Projects in Practice

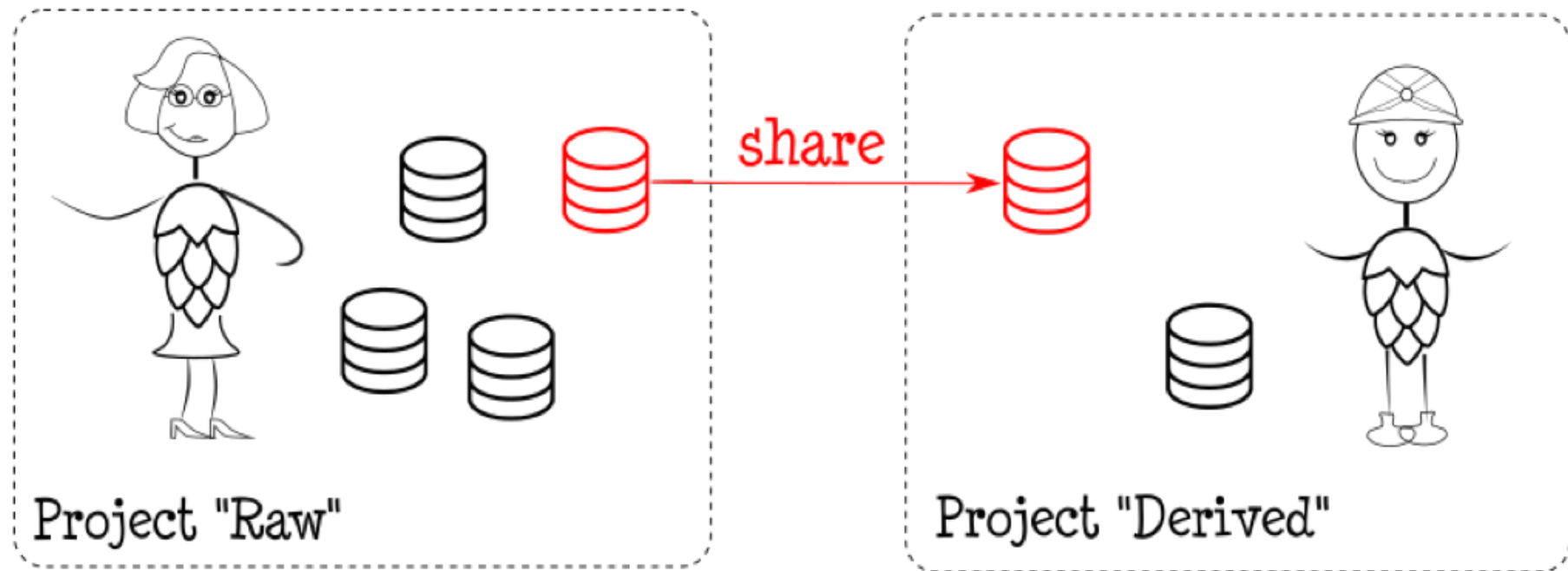


Hopsworks: Hadoop for Humans

Manage Projects like GitHub

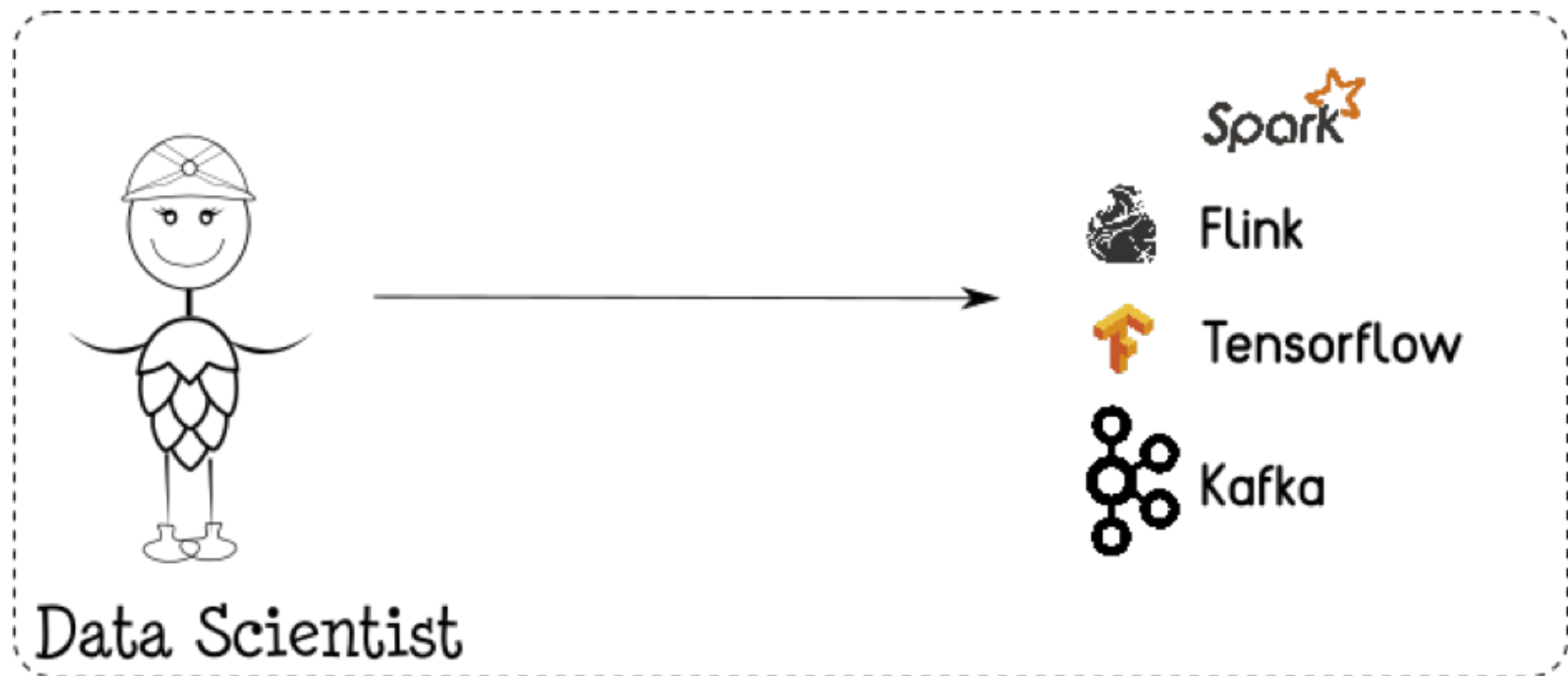


Share like in Dropbox

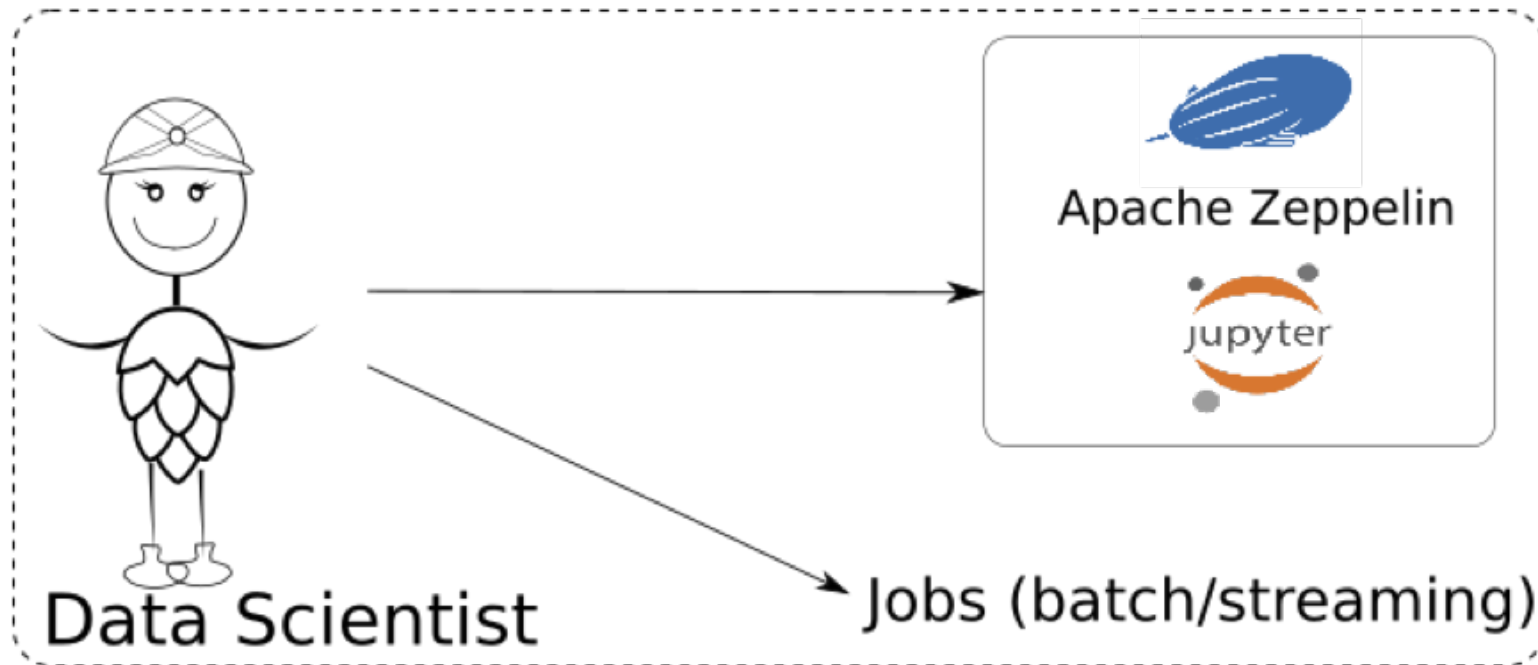


Share any Data Source/Sink: HDFS Datasets, Kafka Topics, etc

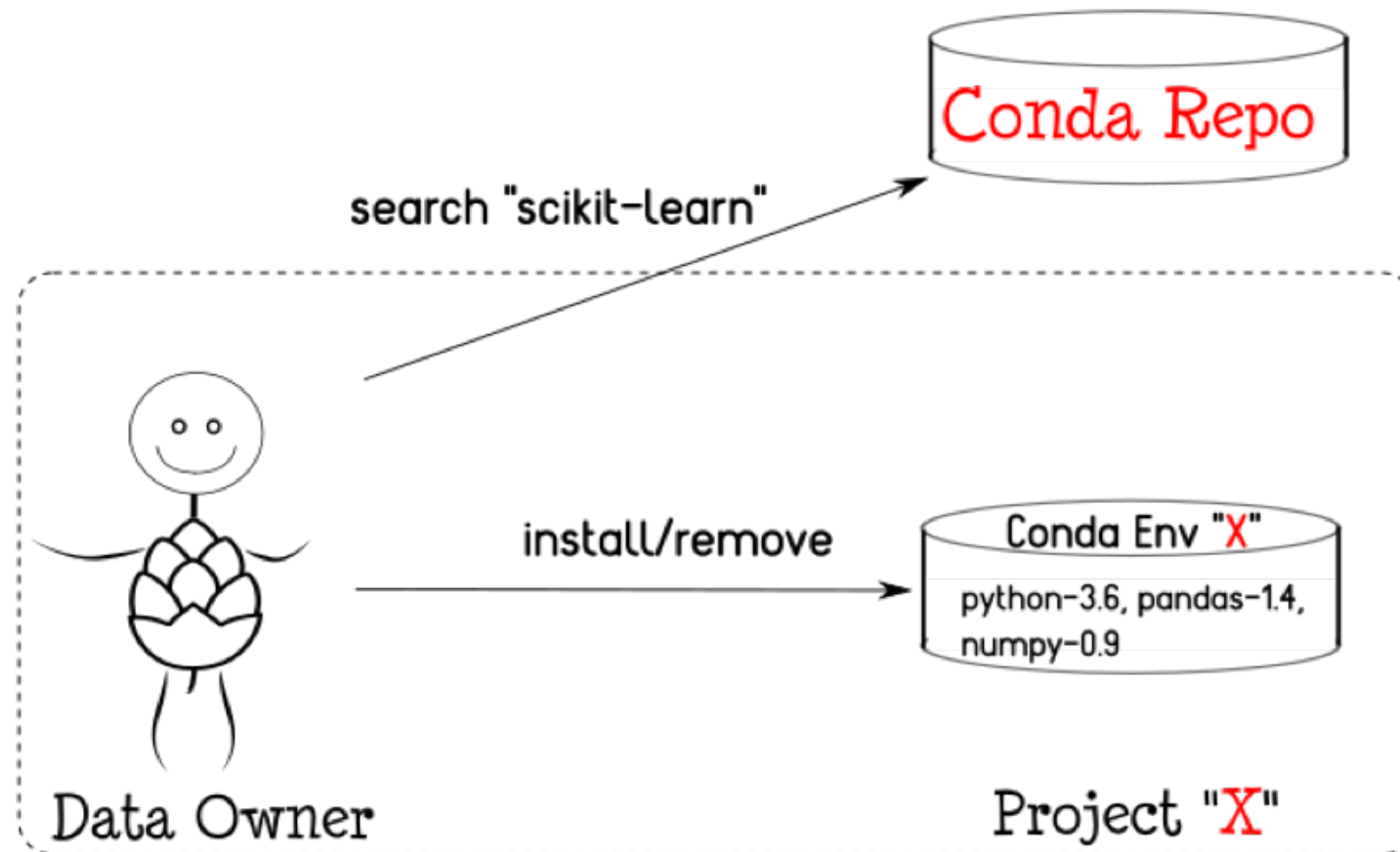
Modern Data Parallel Platforms



Workflow/Jobs and Notebook Support



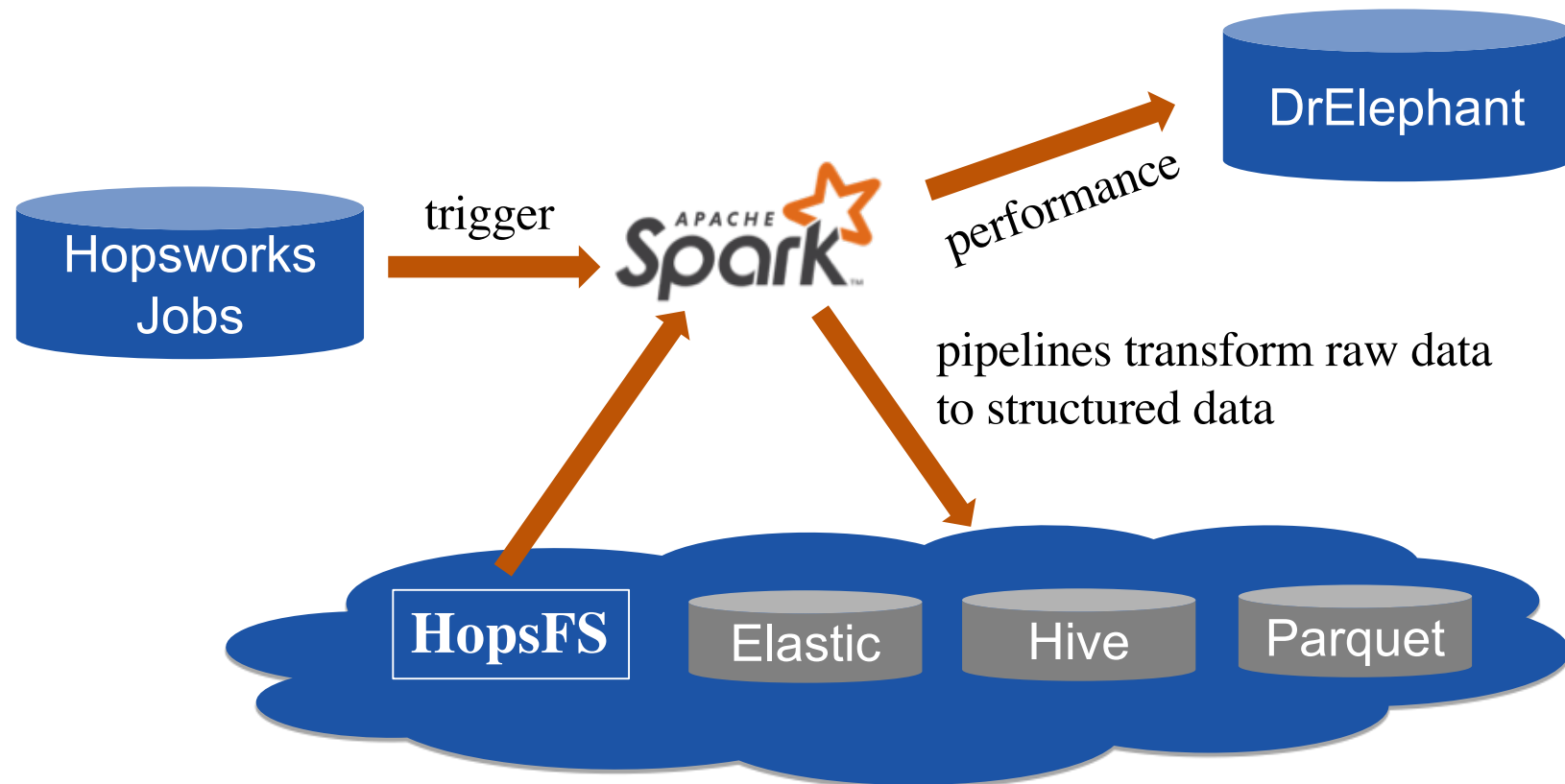
Custom Python Environments with Conda



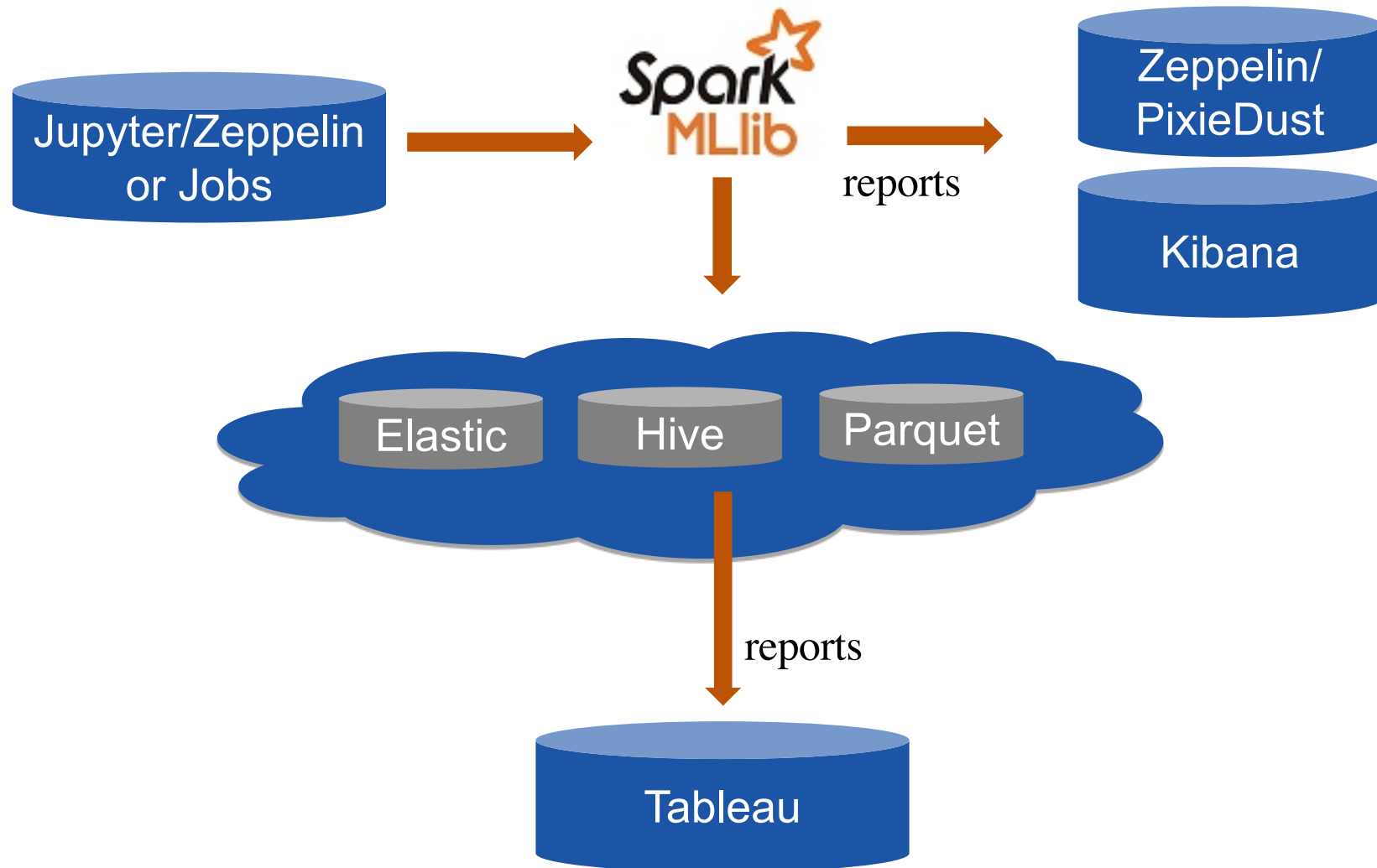
Python libraries are usable by Spark/Tensorflow

How is Hopsworks Used?

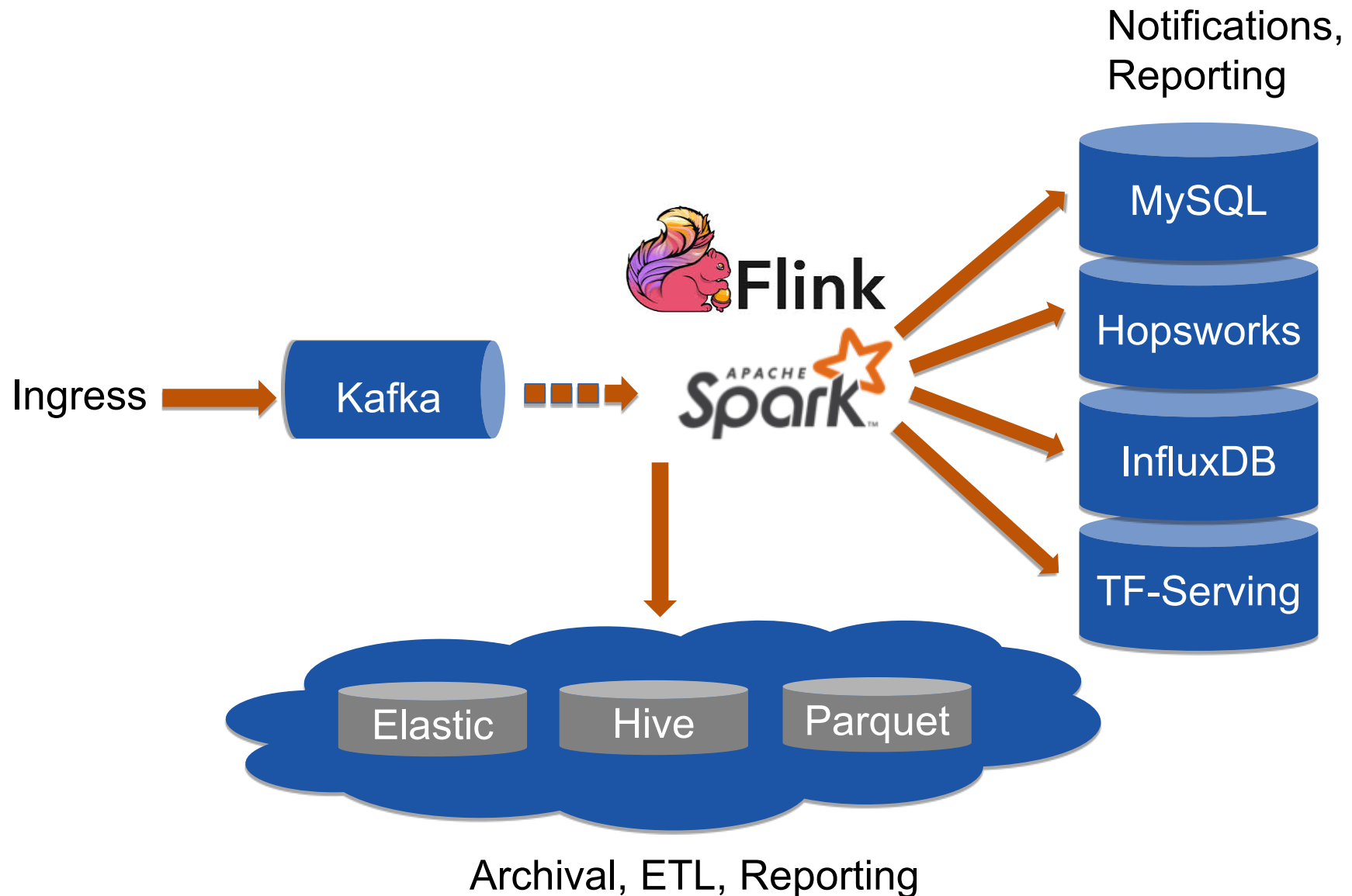
ETL Workloads



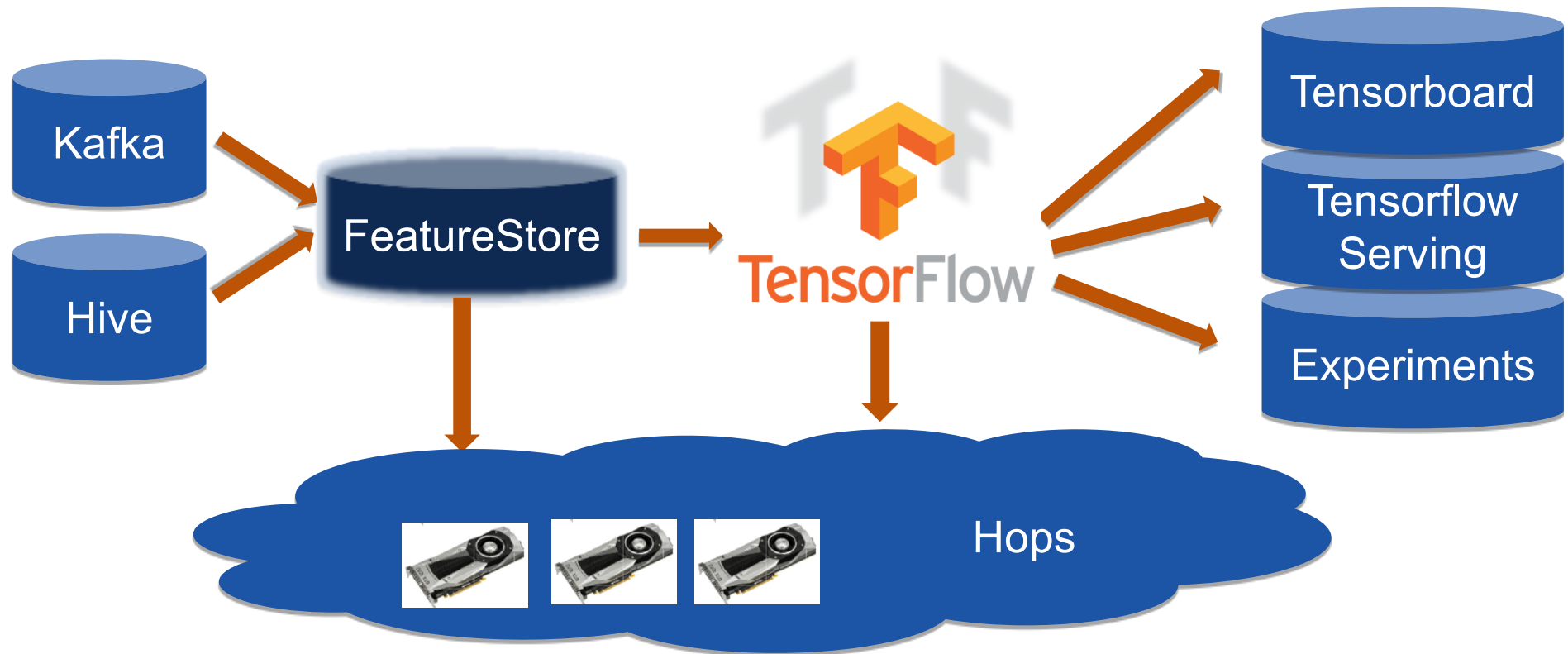
Business Intelligence Workloads



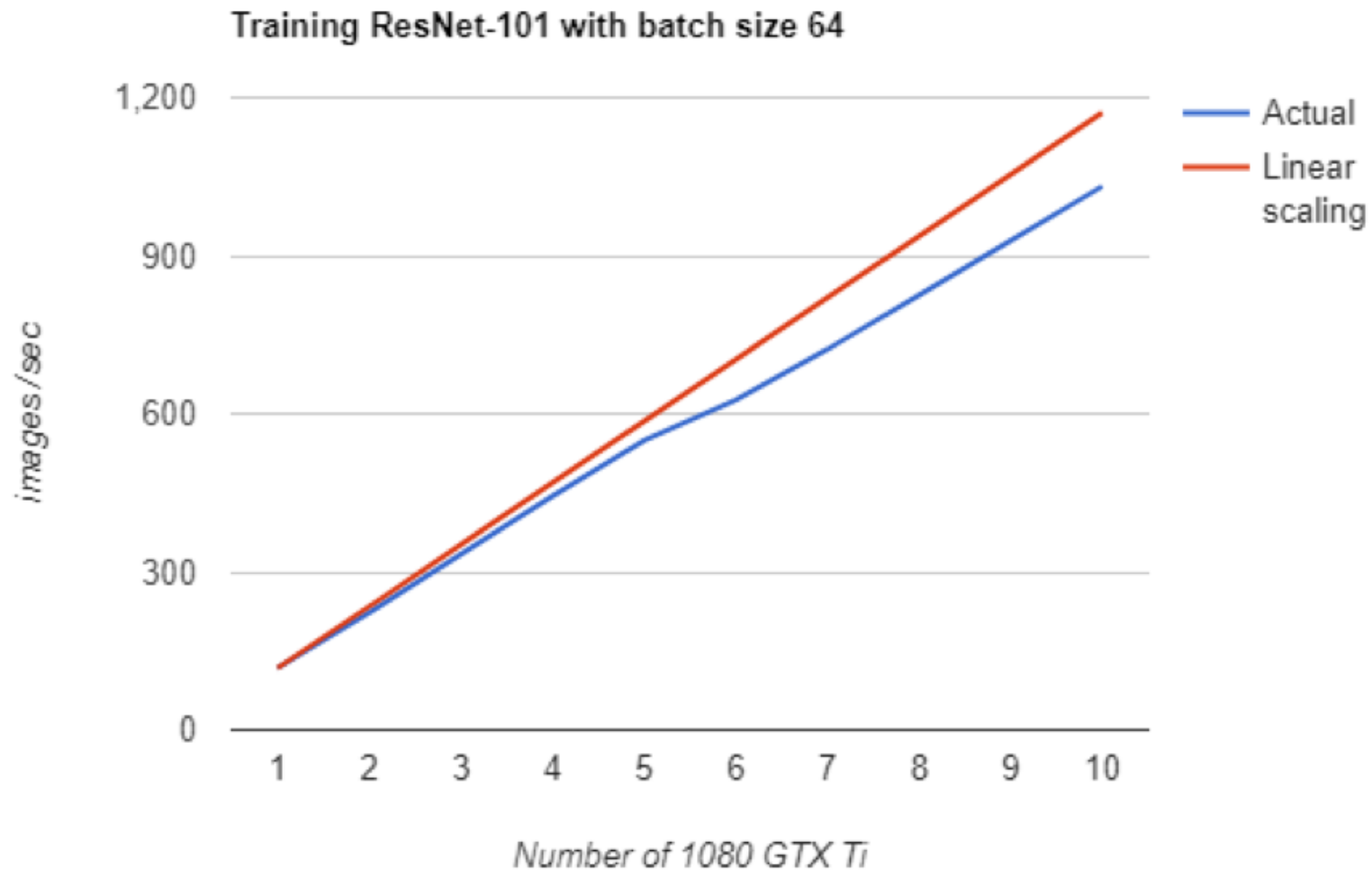
Streaming Analytics in Hopsworks



DeepLearning-as-a-Service



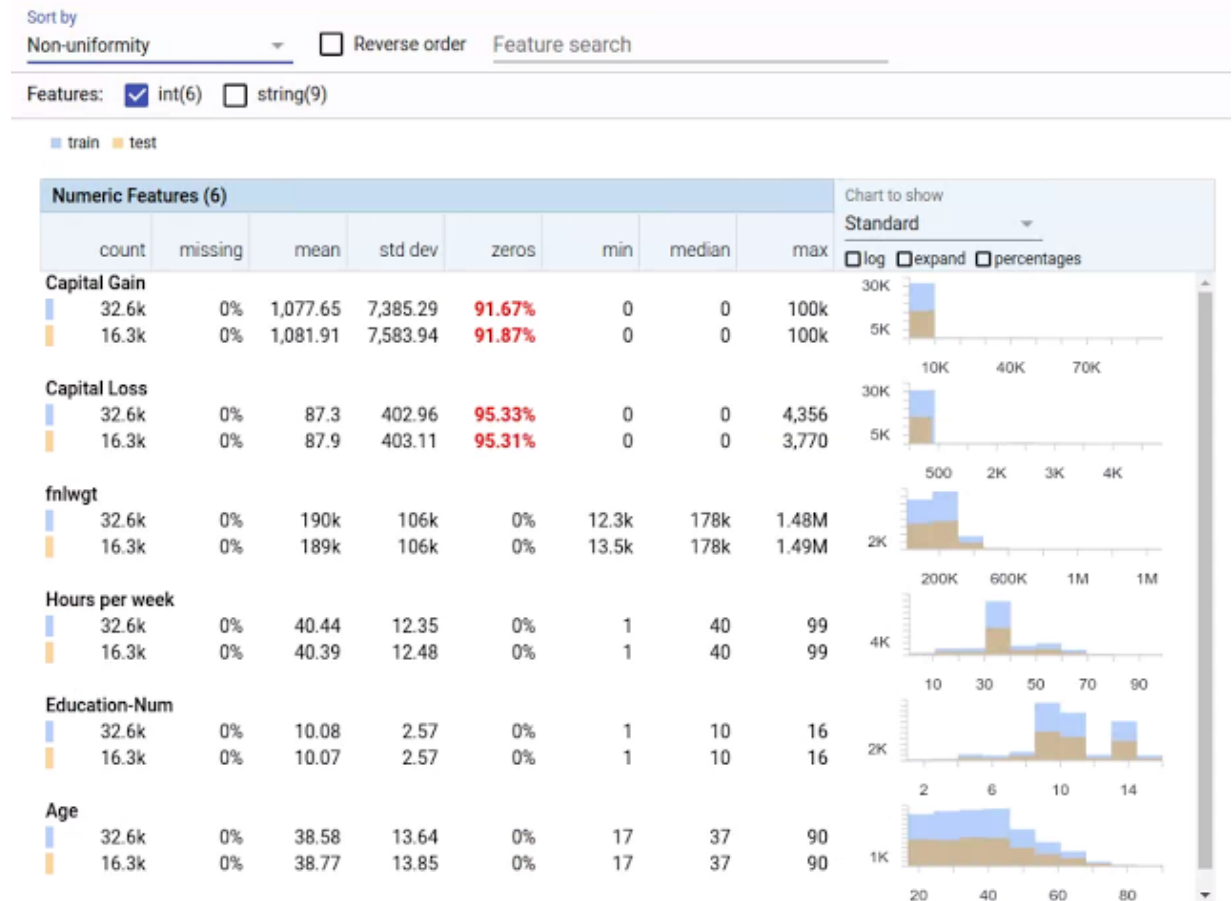
TensorFlow/Hops on 10 1080Ti GPUs



<https://www.oreilly.com/ideas/distributed-tensorflow>

Data Wrangling on Hopsworks

- Google Facets/Dive
- Visualize data distributions
 - Quick Understanding of underlying data
- Min/max/mean/media values for features
- Missing values in columns



Making Spark/TensorFlow Easier to Program

HopsUtil simplifies Secure Spark/Kafka

```
Properties props = new Properties();
props.put(ProducerConfig.BootstrapServersConfig, brokerList);
props.put(SCHEMA_REGISTRY_URL, restApp.restConnect);
props.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG,
org.apache.kafka.common.serialization.StringSerializer.class);
props.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG,
io.confluent.kafka.serializers.KafkaAvroSerializer.class);
props.put("producer.type", "sync");
props.put("serializer.class", "kafka.serializer.StringEncoder");
props.put("request.required.acks", "1");
props.put("ssl.keystore.location", "/var/ssl/kafka.client.keystore.jks");
props.put("ssl.keystore.password", "test1234");
props.put("ssl.key.password", "test1234");
ProducerConfig config = new ProducerConfig(props);
String userSchema = "{\"namespace\": \"example.avro\", \"type\": \"record\", \"name\": \"User\", \"fields\": [{\"name\": \"name\", \"type\": \"string\"}]}";
Schema.Parser parser = new Schema.Parser();
Schema schema = parser.parse(userSchema);
GenericRecord avroRecord = new GenericData.Record(schema);
avroRecord.put("name", "testUser");
Producer<String, String> producer = new Producer<String, String>(config);
ProducerRecord<String, Object> message = new ProducerRecord<>("topicName", avroRecord);
producer.send(data);
```

**SparkProducer producer =
HopsUtil.getSparkProducer();**



Parallel Experiments on Spark/TensorFlow

```
def model_fn(learning_rate, dropout):
```

```
... .
```

Launches 3 Executors with 3 different Hyperparameter settings. Each Executor can have 1-N GPUs.



```
from hops import tflauncher
```

```
args_dict = {'learning_rate': [0.001, 0.005, 0.01],  
             'dropout': [0.5, 0.6, 0.7]}
```

```
tflauncher.launch(spark, model_fn, args_dict)
```

Hadoop-as-a-Service in Sweden

- **Hops**
 - Spark/Flink/Kafka/TensorFlow/Hadoop-as-a-service
 - www.hops.site
- **RISE SICS ICE**
 - 250 kW Datacenter, ~1000 servers
 - Research and test environment



Summary

- Europe's Only Hadoop Distribution – Hops Hadoop
 - Fully Open-Source
 - Larger/faster Hadoop Clusters
- Hopsworks is a new Data Platform built on HopsFS with first-class support for Streaming and ML
 - Spark or Flink
 - Tensorflow
 - Support services for Streaming Analytics and ML

The Team

Active:

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



www.hops.io

 @hopshadoop

Alumni:

Vasileios Giannokostas, Johan Svedlund Nordström, Rizvi Hasan, Paul Mälzer, Bram Leenders, Juan Roca, Misganu Dessalegn, K "Sri" Srijeanthan, 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, Filotas Siskos.





Hops

Thank You.

Follow us:

@hopshadoop

Star us:

<http://github.com/hopshadoop/hopsworks>

Join us:

<http://www.hops.io>