APACHE KUDU

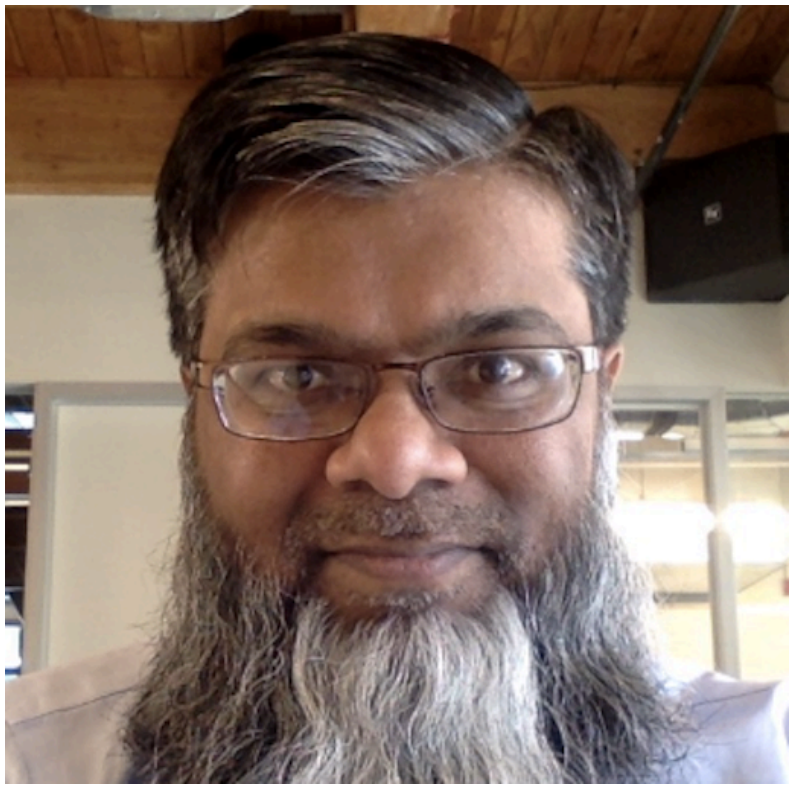
ASIM JALIS

GALVANIZE

INTRO

ASIM JALIS

- Galvanize/Zipfian, Data Engineering
- Cloudera, Microsoft, Salesforce
- MS in Computer Science from University of Virginia

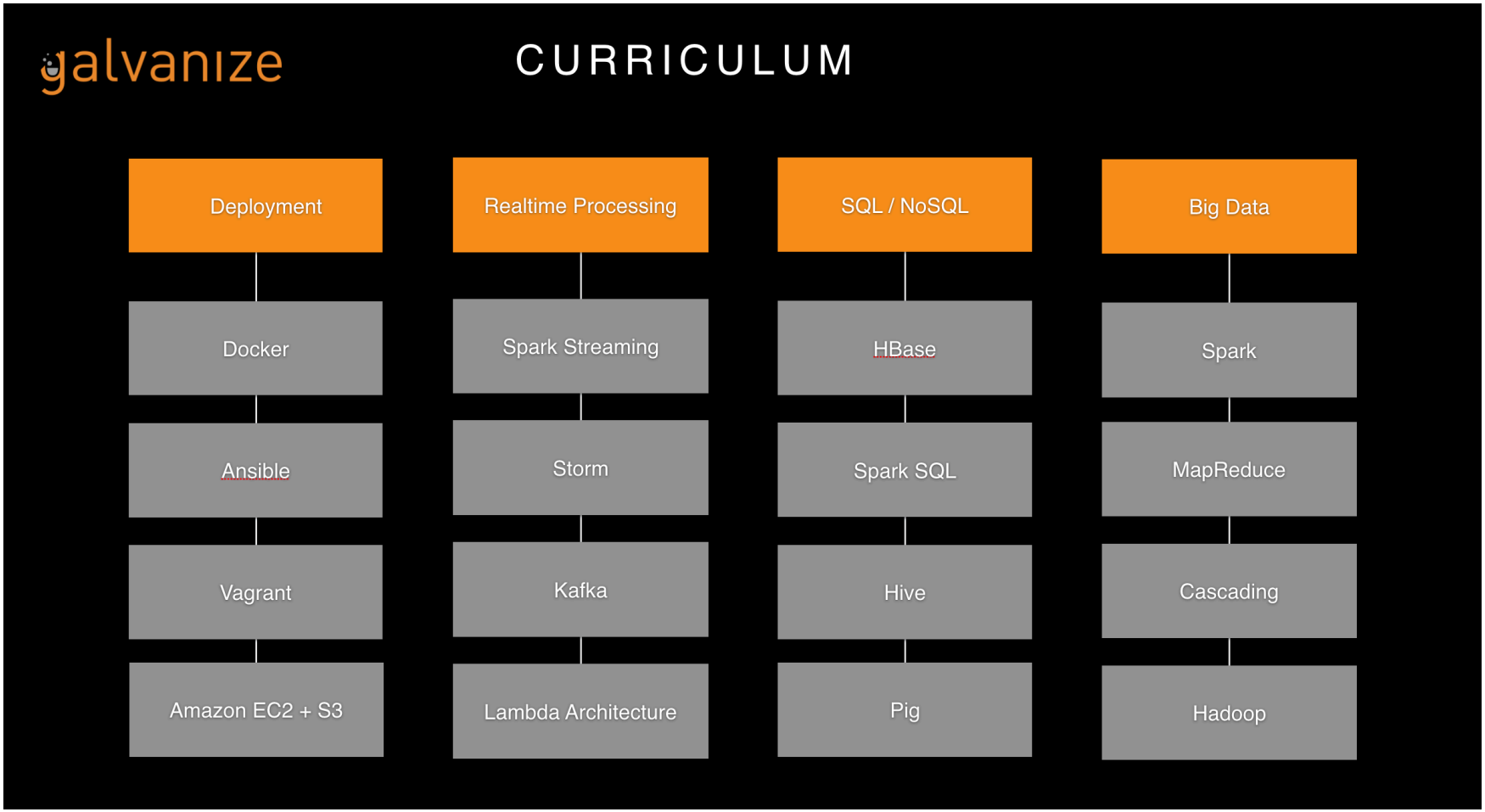


WHAT IS GALVANIZE'S DATA ENGINEERING IMMERSIVE?

- Immersive Peer Learning Environment
- Master High-Demand Skills and Technologies
- Heart of San Francisco in SOMA

YOU GET TO . . .

- Play with Terabytes of Data
- Spark, Hadoop, Hive, Kafka, Storm, HBase
- Data Science at Scale
- Level UP your Career


galvanize
CURRICULUM
Deployment
Docker
Ansible
Vagrant
Amazon EC2 + S3
Realtime Processing
Spark Streaming
Storm
Kafka
Lambda Architecture
SQL / NoSQL
HBase
Spark SQL
Hive
Pig
Big Data
Spark
MapReduce
Cascading
Hadoop

FOR MORE INFORMATION

- Check out http://galvanize.com
- asim.jalis@galvanize.com

TALK OVERVIEW

WHAT IS THIS TALK ABOUT?

- What is Kudu?
- How can I use it to simplify my Big Data architecture?
- How can I use it as an alternative to HBase?
- How does it work?
- Demo

HOW MANY PEOPLE HERE ARE FAMILIAR WITH HDFS?

HOW MANY PEOPLE HERE ARE FAMILIAR WITH HBASE?

HOW MANY PEOPLE HERE ARE FAMILIAR WITH KUDU?

WHY KUDU

WHAT IS A KUDU?

- Kudus are a kind of antelope
- Found in eastern and southern Africa

WHAT PROBLEM DOES KUDU SOLVE?

WHAT IS LATENCY?

- How long it takes to start
- How long it takes to setup

WHICH HAS HIGHER LATENCY: AIRPLANES OR CARS?

WHICH HAS HIGHER LATENCY: AIRPLANES OR CARS?

- Airplanes have high latency
- Cars have lower latency
- Winner: Cars

WHAT IS THROUGHPUT?

- Operations per second/minute/hour
- How much you get done in unit time

WHICH HAS HIGHER THROUGHPUT: AIRPLANES OR CARS?

WHICH HAS HIGHER THROUGHPUT: AIRPLANES OR CARS?

- Airplanes have high throughput
- Cars have lower throughput
- Winner: Airplanes

WHICH IS BETTER?

	Low throughput	**High throughput**
Low latency	Cars	Jet-Packs
High latency	Horses	Planes

- It depends, usually there is a tradeoff
- Going to Oakland: Drive
- Going to Florida: Fly

WHY DOES HBASE EXIST?

- HDFS immutable, append-only
- HBase mutable, random access read/write
- Fast random access
- Low latency (good)

WHY DOES PARQUET EXIST?

- Main idea: columnar storage
- Fast scan, fast batch processing
- High throughput (good)
- High latency (bad)

WHY DOES KUDU EXIST?

	Low throughput	High throughput
Low latency	HBase	Kudu
High latency	JSON on HDFS	Parquet on HDFS

WHY KUDU?

- Kudu is the child of Parquet and HBase
- HBase with columnar storage
- Mutable Parquet with fast random access for read/write
- Parquet-like HBase
- HBase-like Parquet

WHAT IS THE USE CASE FOR KUDU?

- Use HBase-like features to store data as it arrives in real-time
- Use Parquet-like features to run analytic workloads on real-time data
- In queries easily combine long-term historical data and short-term real-time data

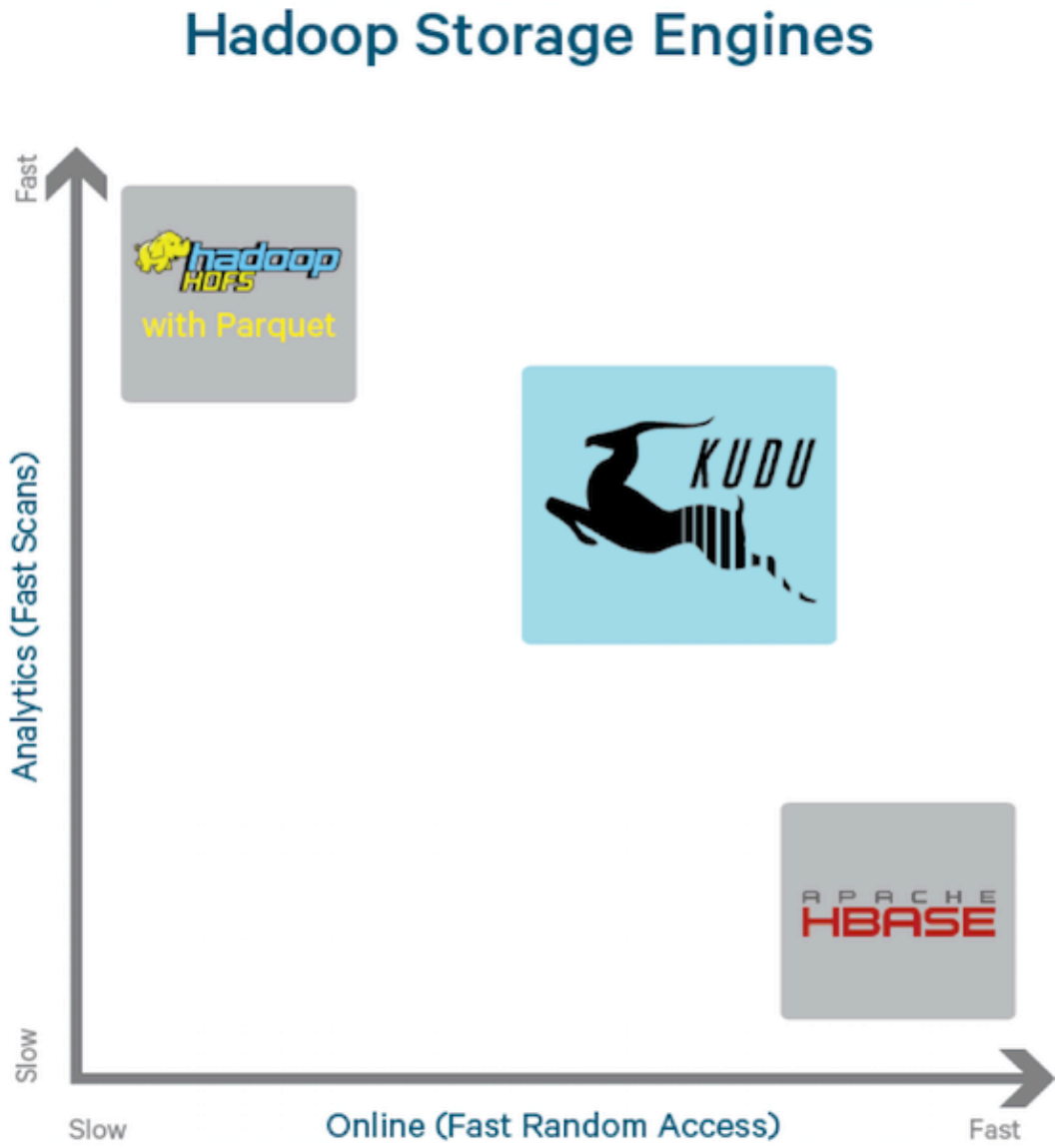
WHAT IS THE BIG IDEA OF KUDU?

- HBase uses in-memory store for low-latency reads/writes
- Parquet on HDFS uses columnar layout for high-throughput scans
- Kudu idea: in-memory store + columnar layout

KUDU MOTIVATION

Goal	Competing With
Fast columnar scans	Parquet/HDFS
Low-latency random updates	HBase
Consistent performance	-

KUDU, HBASE, HDFS



WHAT LANGUAGE IS KUDU WRITTEN IN?

- C++ for performance
- Java and Python wrappers

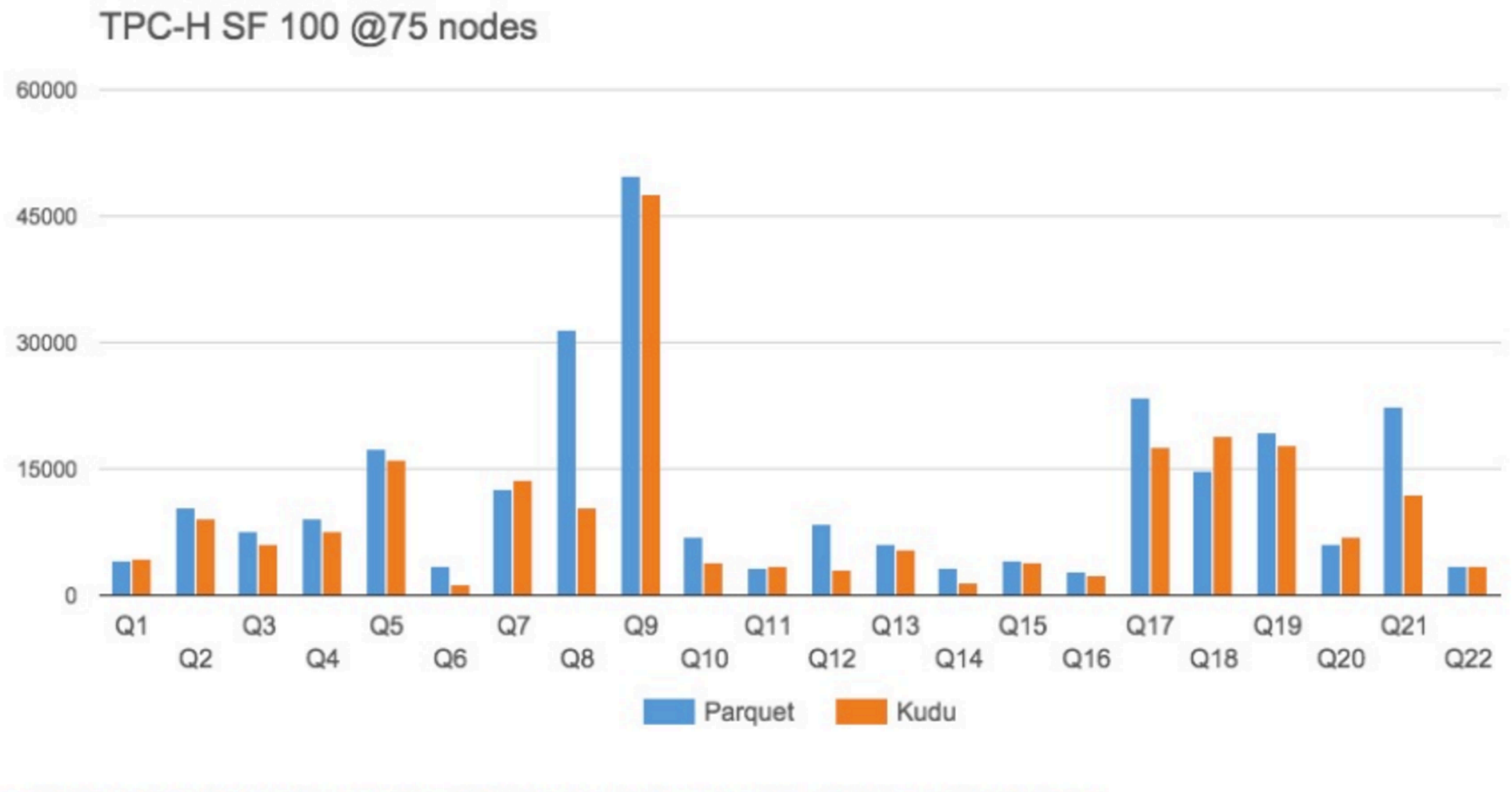
HOW CAN I INTERACT WITH KUDU?

- Impala is Kudu's defacto shell
- C++, Java, or Python client
- MapReduce
- Spark (beta)
- MapReduce and Spark read-only access (currently)

BENCHMARKS

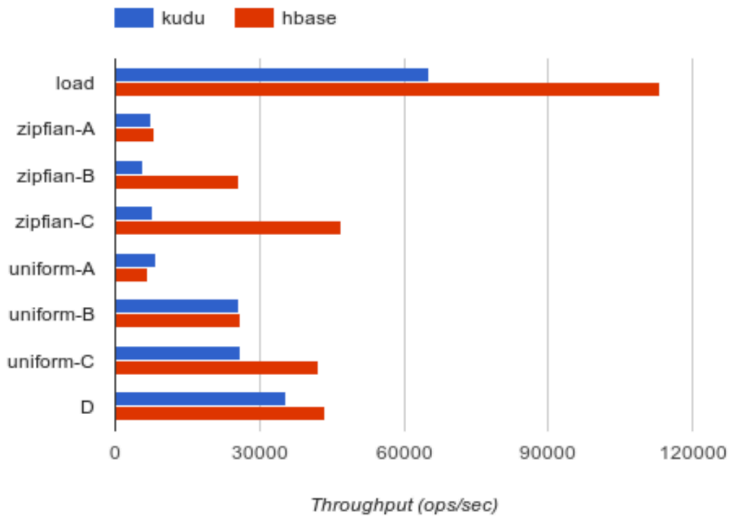
KUDU VS PARQUET ON HDFS

- TPC-H: Business-oriented queries/updates
- Latency in ms: lower is better

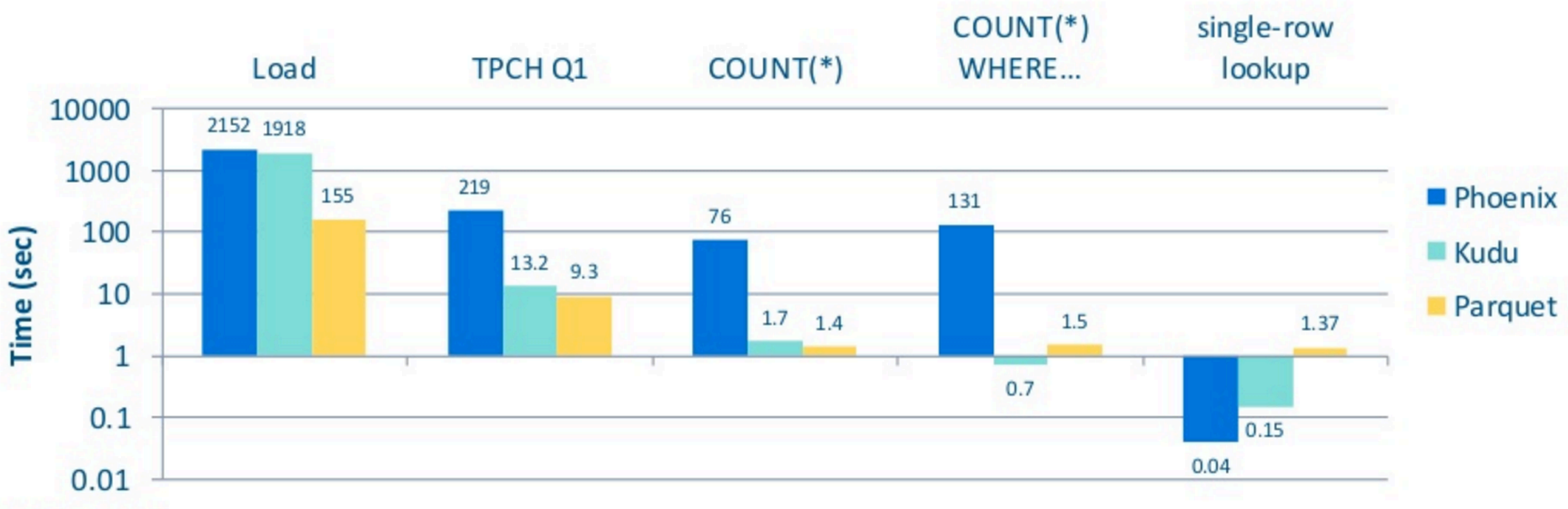


KUDU VS HBASE

- Yahoo! Cloud System Benchmark (YCSB)
- Evaluates key-value and cloud serving stores
- Random acccess workload
- Throughput: higher is better



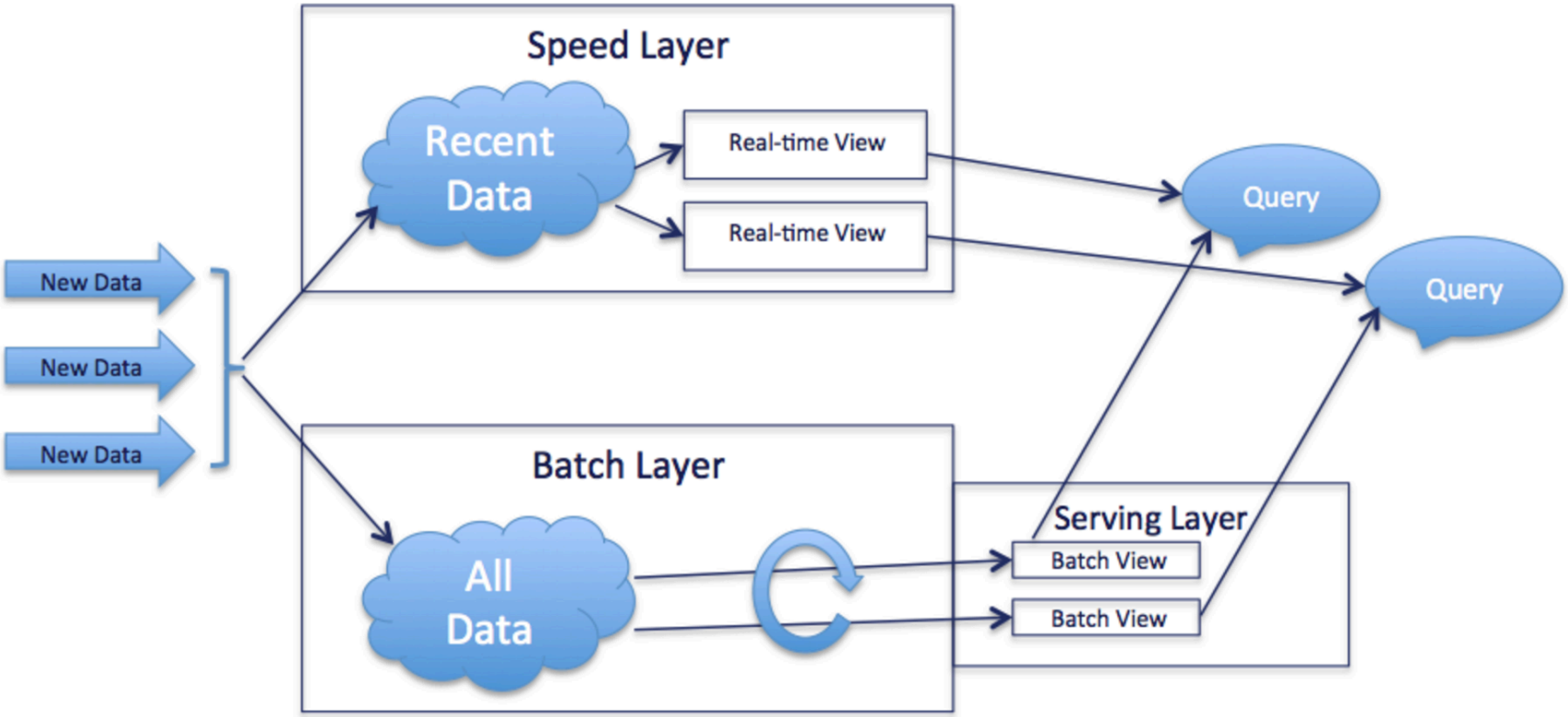
KUDU VS PHOENIX VS PARQUET



- SQL analytic workload
- TPC-H LINEITEM table only
- Phoenix best-of-breed SQL on HBase

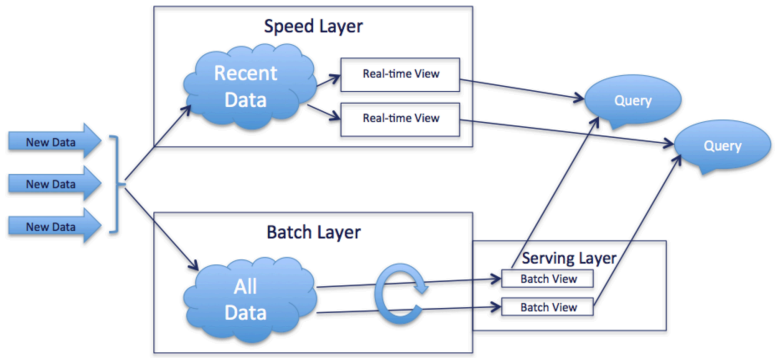
LAMBDA ARCHITECTURE

KUDU USE CASE: LAMBDA ARCHITECTURE



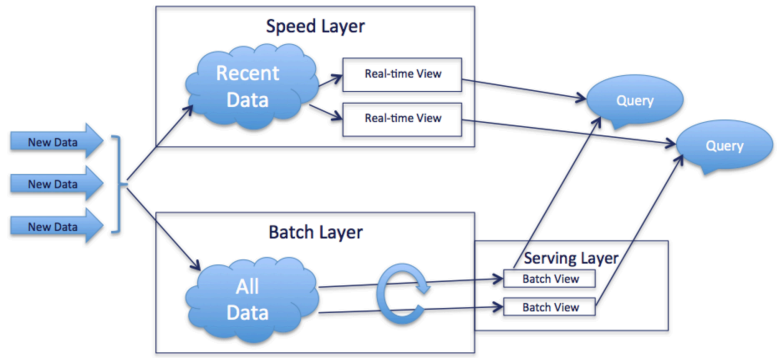
LAMBDA ARCHITECTURE

- Real-time data process by Speed Layer
- Historical data processed by Batch Layer
- Speed Layer results saved in HBase
- New and old data joined in Spark Streaming



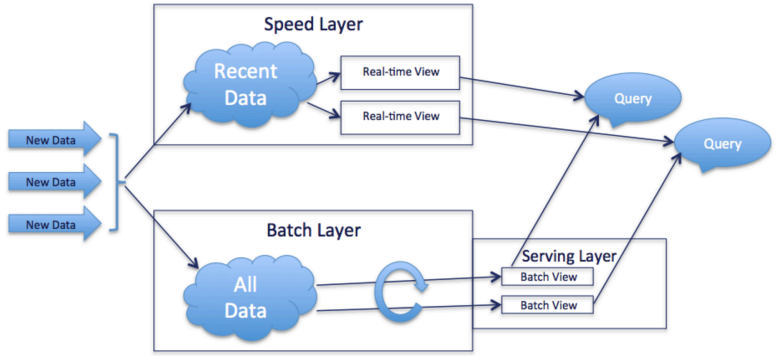
ONLINE RETAILER USE CASE

- List popular items top of front page
- What is best selling item:
  - this hour
  - today
  - this week
- Keep inventory numbers updated

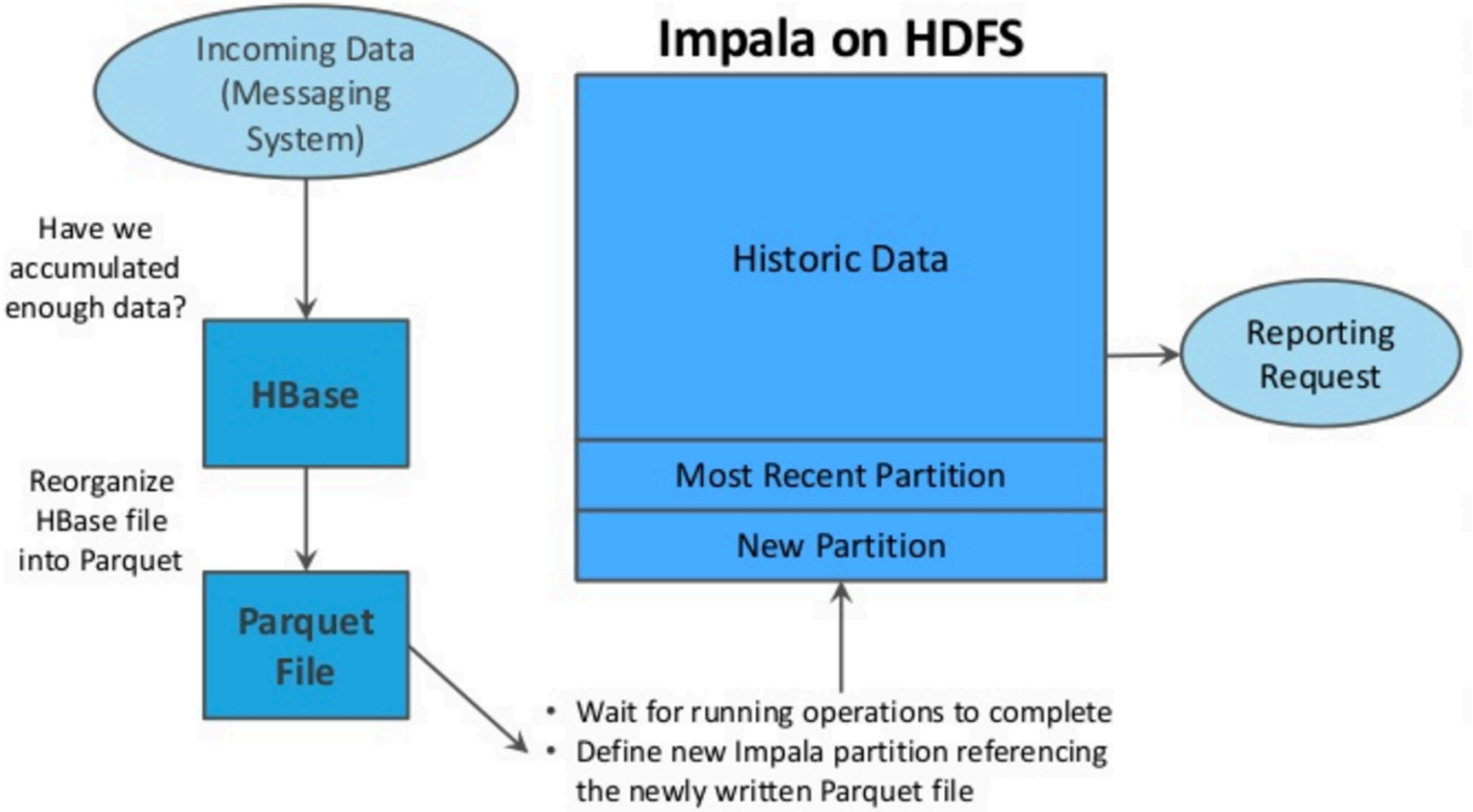


KUDU

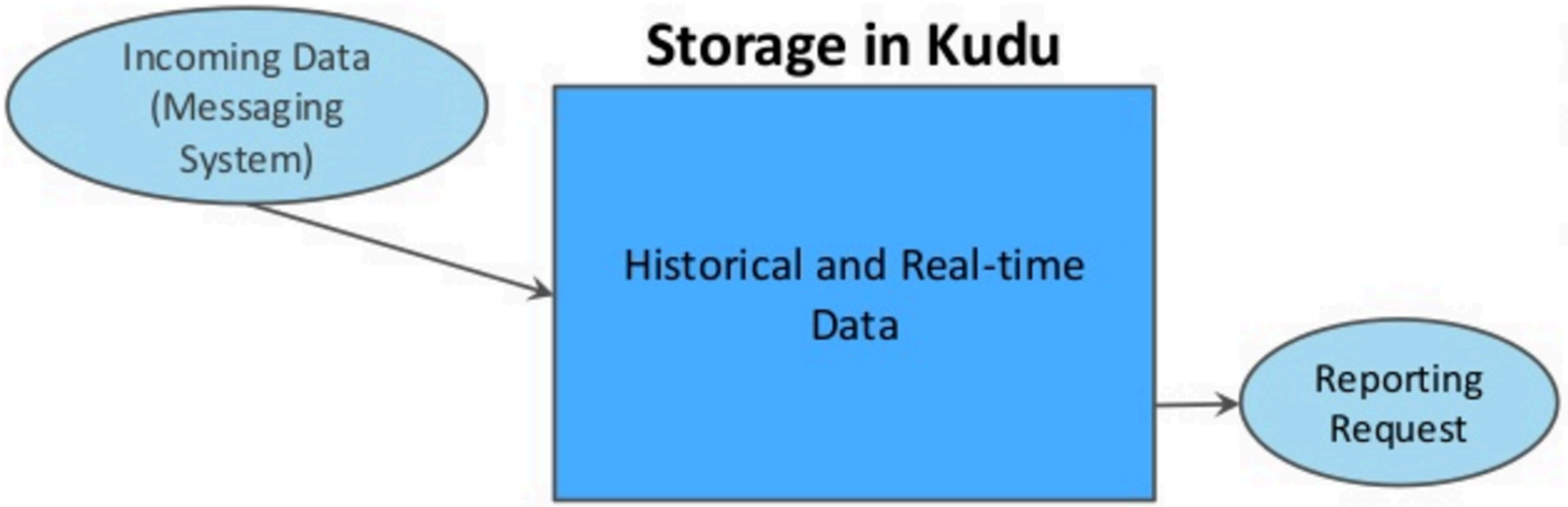
- Using Kudu
- Speed layer queries can include analytics



PRE-KUDU ARCH

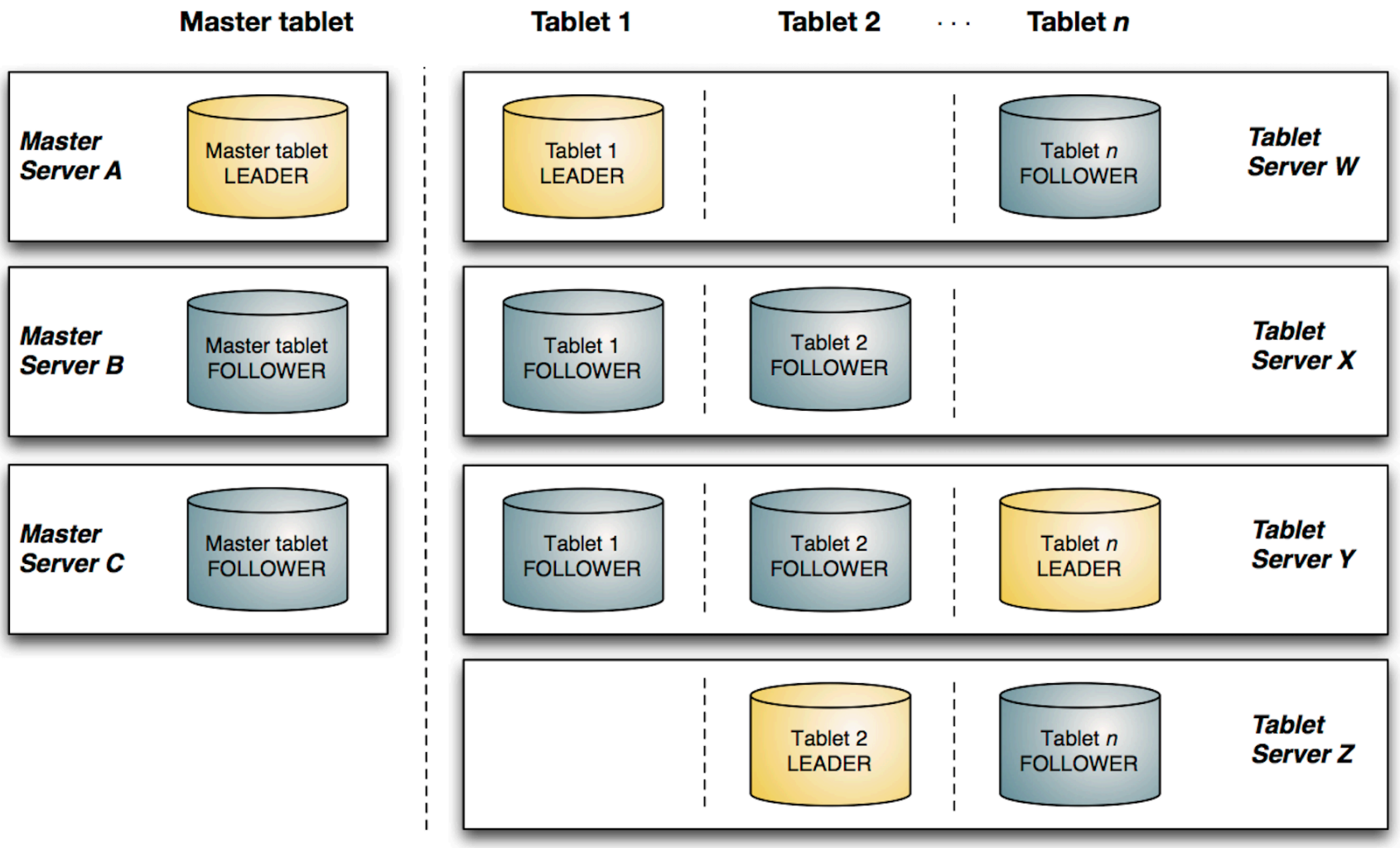


POST-KUDU ARCH



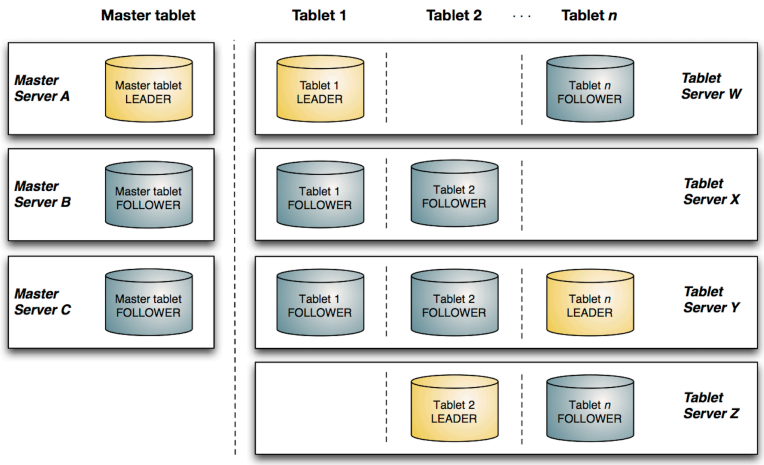
KUDU INTERNALS

KUDU ARCHITECTURE



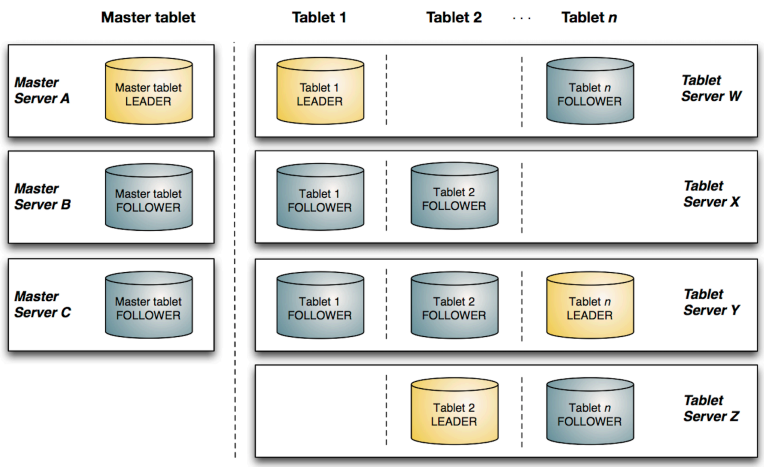
KUDU MASTER

- Fault tolerance
- Failover to backup masters
- Raft used for electing new leaders
- Only leader serves client requests



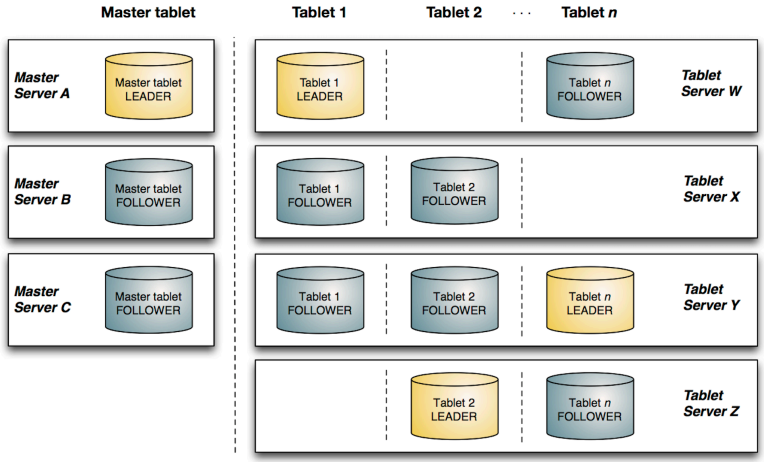
KUDU MASTER ROLES

- Catalog Manager
  - Metadata: schema, replication
  - Master Tablet stores metadata
- Cluster Coordinator
  - Who is alive: redistribute data on death
- Tablet directory
  - Which tablet is where



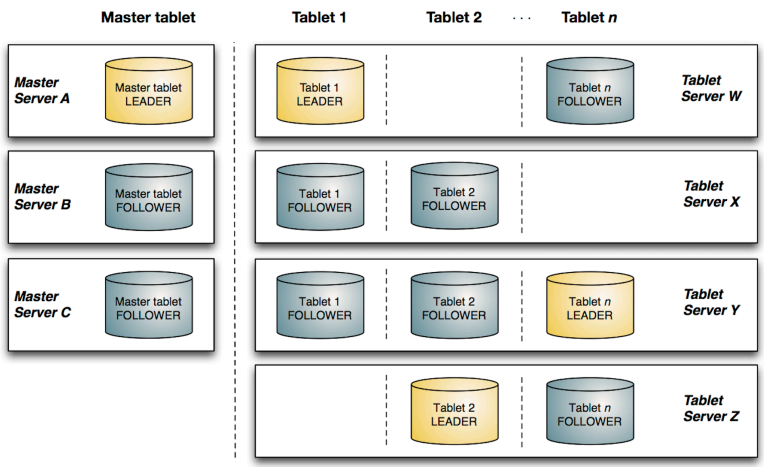
TABLETS

- Tablets are similar to HBase Regions
- Tables statically partitioned into tablets
- Tablets can be replicated to 3 or 5
- Replication uses Raft's Leader/Follower pattern for consensus
- Data stored on local file system, not HDFS



TABLET REPLICATION

- Writes must be done on leader tablet
- Reads can be on leader or followers
- Write log is replicated
- Log replication driven by leader
- Uses Raft consensus



THICK CLIENT

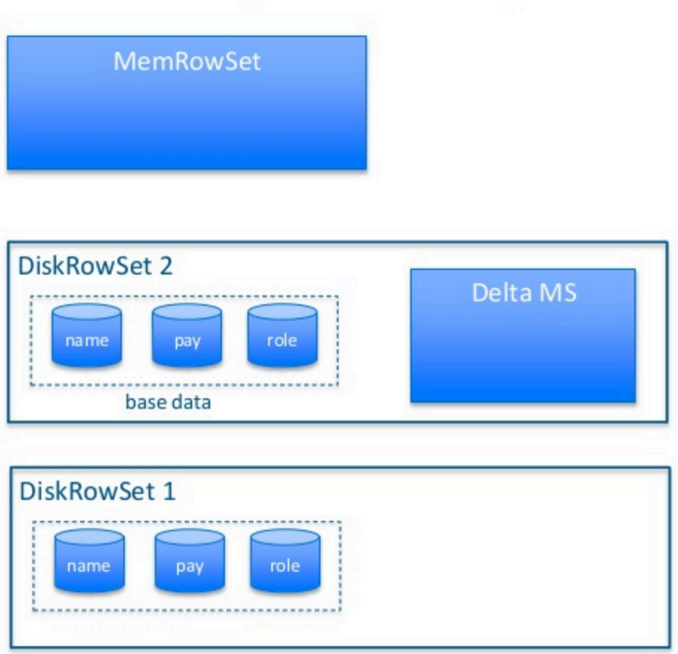
- Client caches tablet metadata
- Uses metadata to figure out leader to write to
- Failure on write to deposed leader
- Write failure forces metadata refresh

TABLET INTERNALS

Component	Description
MemRowSet	In-memory writes, stored row-wise
DiskRowSet	Disk-based data store, columnar, 32 KB
DeltaMemStores	In-memory update store
DeltaFile	Disk-based update store

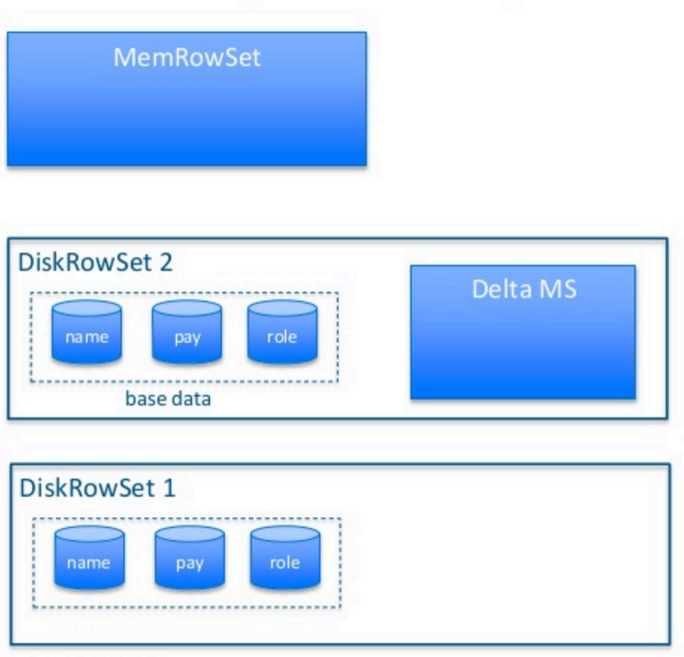
MEM ROW SET

- Writes for new keys are written in MemRowSets.
- Then flushed out to DiskRowSets.
- Kudu uses Bloom filters to determine if a key is in DiskRowSet.



DELTA MEM STORES

- Updates are written to DeltaMemStores.
- The flushed to DeltaFiles.



TABLET COUNT

- Tablet count based on partitions
- Partitioning function maps row to tablet
- Partitioning defined at table creation time
- Partitions cannot be redefined dynamically

PARTITIONING

- Hash Partitioning
  - Subset of the primary key columns and number of buckets
  - For example

    ```
    DISTRIBUTE BY HASH(col1,col2) INTO 16 BUC
    ```
- Range Partitioning
  - Ordered subset of primary key columns
  - Maps tuples into binary strings by concatenating values of specified columns using order-preserving encoding.

DOES KUDU REQUIRE HADOOP?

- It does not depend on HDFS.
- Has no required dependency on Hadoop, HDFS, MapReduce, Spark.
- However, in practice accessed most easily through Impala.

DO KUDU TABLETSERVERS SHARE DISK SPACE WITH HDFS?

- Kudu TabletServers and HDFS DataNodes can run on the machines.
- Kudu's InputFormat enables data locality.
- Data locality: MapReduce and Spark tasks likely to run on machines containing data.

KUDU SCHEMA

WHAT DATA TYPES DOES KUDU SUPPORT?

- Boolean
- 8-bit signed integer
- 16-bit signed integer
- 32-bit signed integer
- 64-bit signed integer
- Timestamp
- 32-bit floating-point
- 64-bit floating-point
- String
- Binary

HOW LARGE CAN VALUES BE IN KUDU?

- Values in the 10s of KB and above are not recommended
- Poor performance
- Stability issues in current release
- Not intended for big blobs or images

ENCODING TYPES

Column Type	Encoding
integer, timestamp	plain, bitshuffle, run length
float	plain, bitshuffle
bool	plain, dictionary, run length
string, binary	plain, prefix, dictionary

WHAT IS PLAIN ENCODING?

- Data in its natural format.
- E.g. int32 stored as 32-bit little-endian integers.

WHAT IS BITSHUFFLE ENCODING?

- Bitwise columnar encoding.
- MSB stored first, then second-MSB, etc.
- Result LZ4 compressed.
- Works well when values repeat or change by small amounts.

```
11010010       1111 1111
11010011 -->   0000 1111
11010100       0000 0011
11010101       1100 1010
```

WHAT IS RUN LENGTH ENCODING?

- Repeated values (runs) are stored as value and count.
- Works well for denormalized tables with consecutive repeated values when sorted by primary key.

```
54.231.184.7
54.231.184.7
54.231.184.7  --> 54.231.184.7 | 5
54.231.184.7
54.231.184.7
```

WHAT IS DICTIONARY ENCODING?

- Dictionary of unique values.
- Value encoded as index in dictionary.
- Works well if column has small set of unique values.
- If there are too many values Kudu falls back to plain encoding.

WHAT IS PREFIX ENCODING?

- Common prefixes compressed in consecutive column values.
- Works well when values share common prefixes.

COLUMN COMPRESSION

- Per-column compression using LZ4, Snappy, or ZLib.
- By default, columns stored uncompressed.

WHAT IMPACT WILL COMPRESSION HAVE ON SCAN PERFORMANCE AND ON SPACE?

- It will reduce storage space.
- It will reduce scan performance.

DEMO

ADMIN UI VM PORT SETUP

```
Virtual Box > Settings > Network > Adapter 2 > Port Forwarding > +

Enter:
    Name: Rule 1
    Protocol: TCP
    Host IP:
    Host Port: 8051
    Guest IP:
    Guest Port: 8051

Open browser:
http://quickstart.cloudera:8051
```

CONNECT TO VM

```
ssh demo@quickstart.cloudera
```

START IMPALA SHELL

```
impala-shell
```

CREATE A TABLE

```
CREATE TABLE sales (
  state STRING,
  id INTEGER,
  sale_date STRING,
  store INTEGER,
  product INTEGER,
  amount DOUBLE
)
DISTRIBUTE BY RANGE(state)
SPLIT ROWS(('CA'),('WA'),('OR'))
TBLPROPERTIES(
  'storage_handler' = 'com.cloudera.kudu.hive.KuduStorageHandler',
  'kudu.table_name' = 'sales',
  'kudu.master_addresses' = 'quickstart.cloudera:7051',
  'kudu.key_columns' = 'state,id'
);
```

INSERT SOME DATA INTO IT

```
INSERT INTO sales
(state, id, sale_date, store, product, amount)
VALUES
('WA',101,'2014-11-13',100,331,300.00),
('OR',104,'2014-11-18',700,329,450.00),
('CA',102,'2014-11-15',203,321,200.00),
('CA',106,'2014-11-19',202,331,330.00),
('WA',103,'2014-11-17',101,373,750.00),
('CA',105,'2014-11-19',202,321,200.00);
```

TRY SQL

```
SELECT COUNT(*) FROM sales;
SELECT STATE,COUNT(*) FROM sales GROUP BY STATE;
```

VIEW TABLE IN ADMIN UI

- Go to http://quickstart.cloudera:8051/

REFERENCES

READINGS

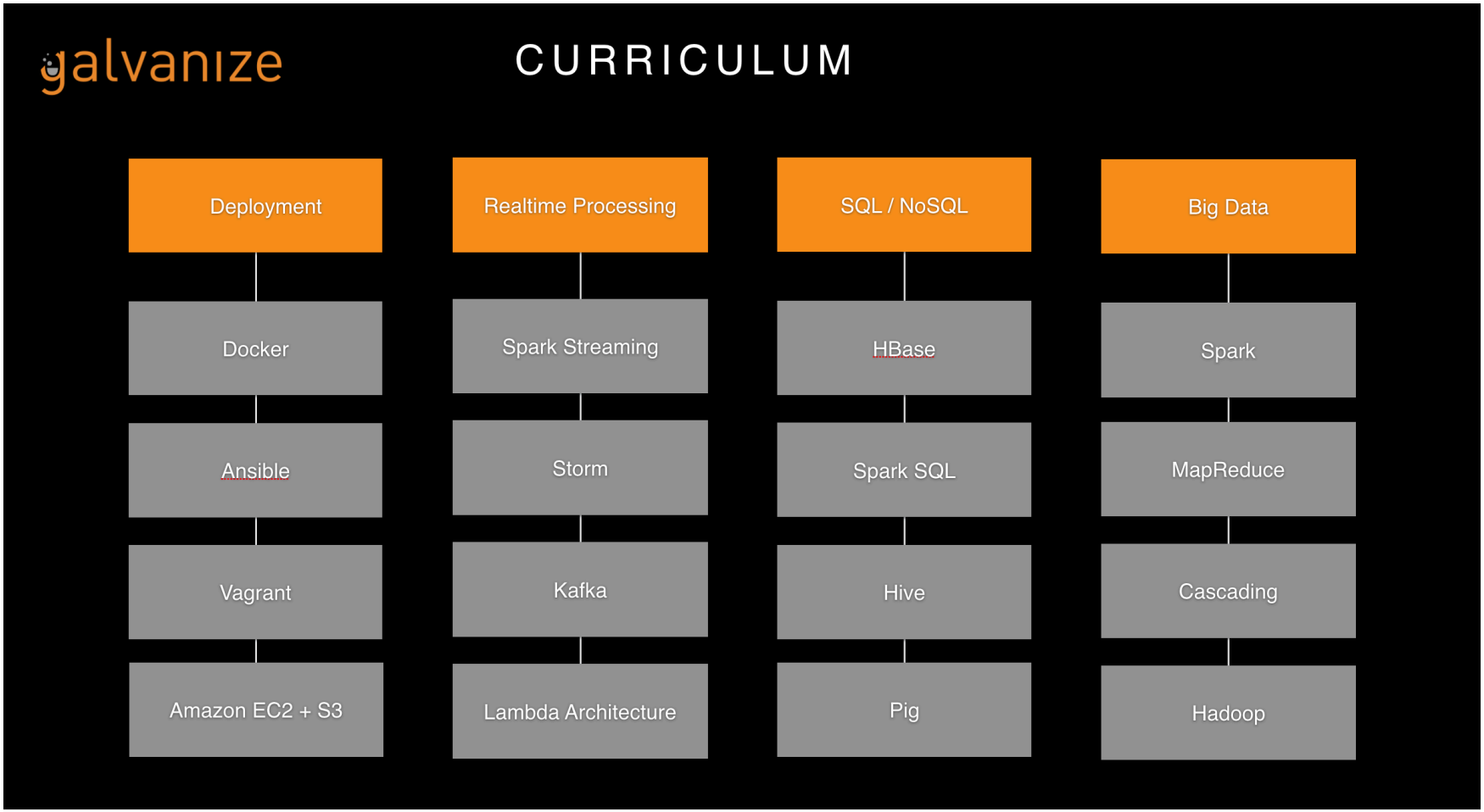
Kudu Whitepaper
http://getkudu.io/kudu.pdf

Quickstart VM
http://getkudu.io/docs/quickstart.html

Schema
http://getkudu.io/docs/schema_design.html

Kudu Impala Guide
http://www.cloudera.com/documentation/betas/kudu/0-5-0/topics/kudu_impala.html

GALVANIZE DATA ENGINEERING



QUESTIONS