cloudera

Kudu: Storage for Fast Analytics on Fast Data

Ippokratis Pandis | ippokratis@cloudera.com

HPTS 2015

Motivation

- One of the biggest challenges for storing data in HDFS is it's immutability
- Many use-cases require constantly changing fact or dimension tables
- Today: HBase or Cassandra with periodic roll-over to HDFS
  - Complicated workflows
  - Latency / lack of freshness

Fast analytics on fast data.

cloudera

Kudu

- **(Almost) best of both worlds**
  - Nearly as fast as raw HDFS for scans
  - Nearly as fast as HBase for random access
- **High CPU Performance**
  - Single-column scan rate 10-100x faster than HBase
  - Needed to take advantage of RAM and Flash
- **High IO efficiency**
  - True column store with type-specific encodings
  - Allow efficient analytics when only certain columns are accessed
- Expressive and evolvable data model
- An architecture that supports multi-data center operation

cloudera

Kudu

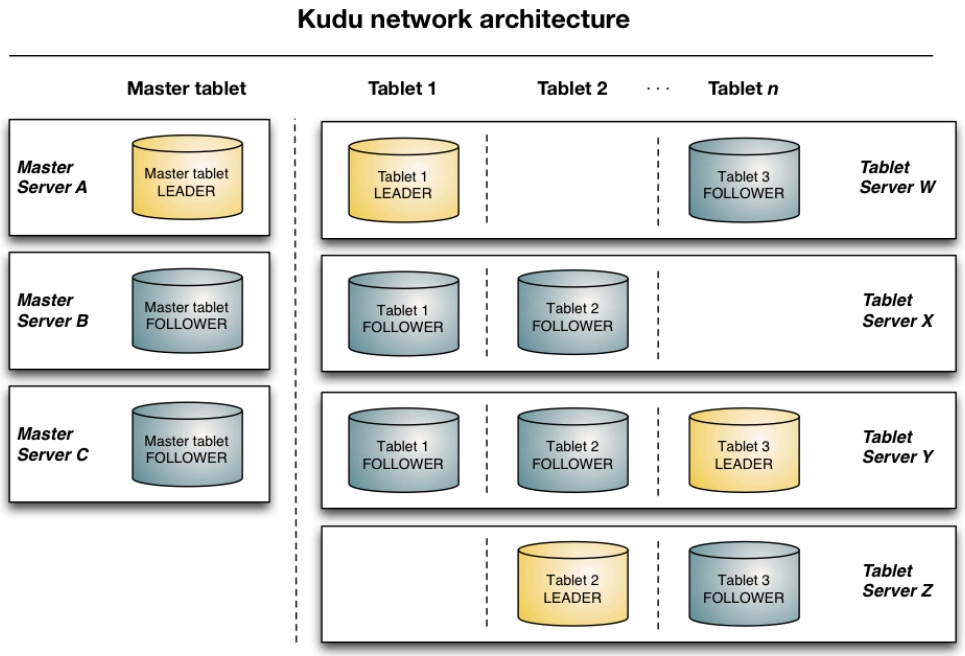
- Announced today at Hadoop World NYC
  - Available as a beta
- Fully integrated into Cloudera Manager infrastructure
- Impala integration part of the beta program

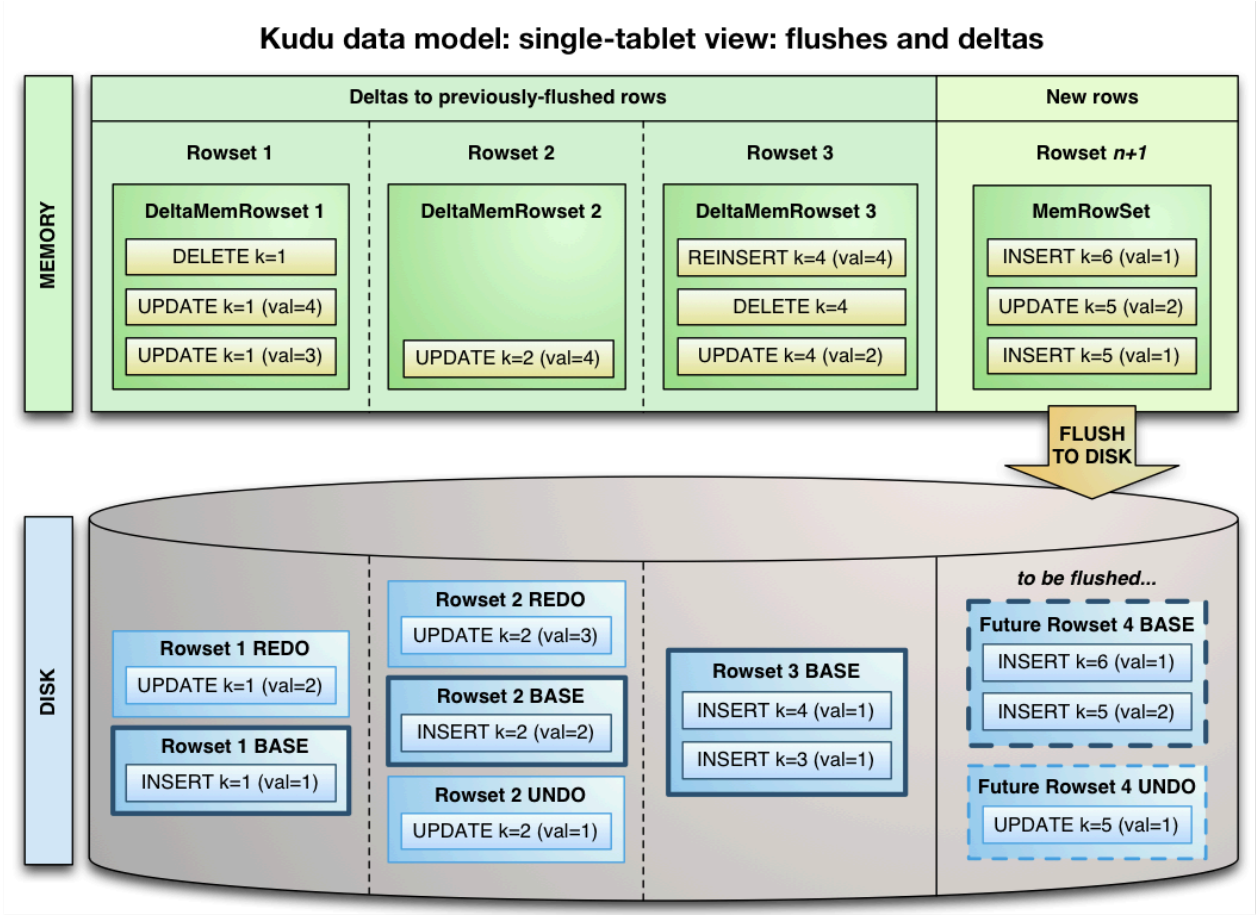
- Written in C++ from scratch
- APIs C++/Java/Python

getkudu.io

cloudera

Architecture





cloudera

Deep Integration with Impala

- Can query Kudu- and HDFS-stored data
- Syntax extensions for **UPDATE** and **DELETE**
- **CREATE TABLE** support for KUDU

RANGE and HASH based distribution

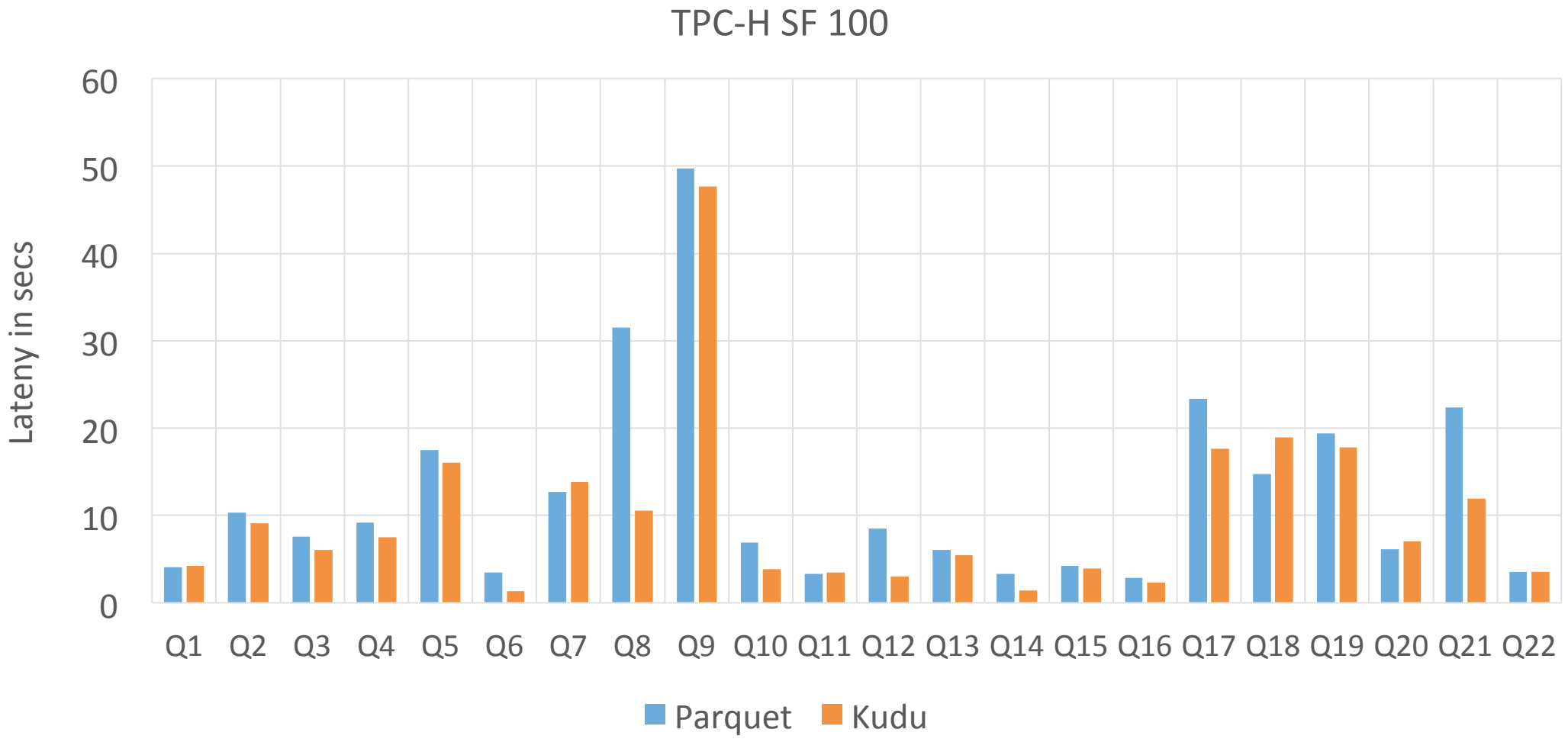
```
CREATE TABLE partsupp
DISTRIBUTE BY HASH INTO 75 BUCKETS
TBLPROPERTIES (
 'storage_handler' = 'com.cloudera.kudu.hive.KuduStorageHandler',
 'kudu.table_name' = 'tpch_100_kudu_partsupp',
 'kudu.num_tablet_replicas' = '3',
 'kudu.master_addresses' = 'a1111.halxg.cloudera.com',
 'kudu.key_columns' = 'ps_partkey,ps_suppkey' )
AS SELECT * FROM tpch_100_parquet.partsupp;
```

REPLICAS for performance and fault-tolerance

Real (compound) primary-keys.

cloudera

Performance Evaluation – TPC-H SF 100



cloudera

Thank you!

+

cloudera