

# **Toronto Apache Spark (TAS)**

Building Enterprise Data Lake Solution using  
Spark and SequoiaDB

Tao Wang



2015-09-30 (TAS)

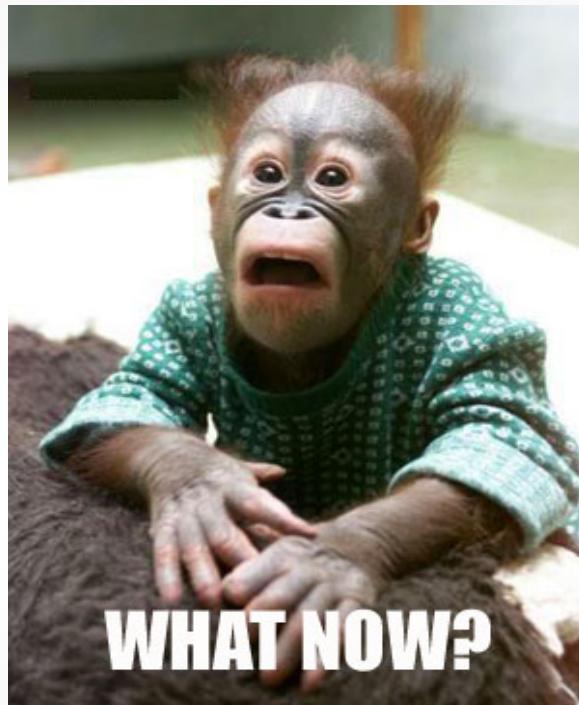
# **Outline**

What

Why

How

Where

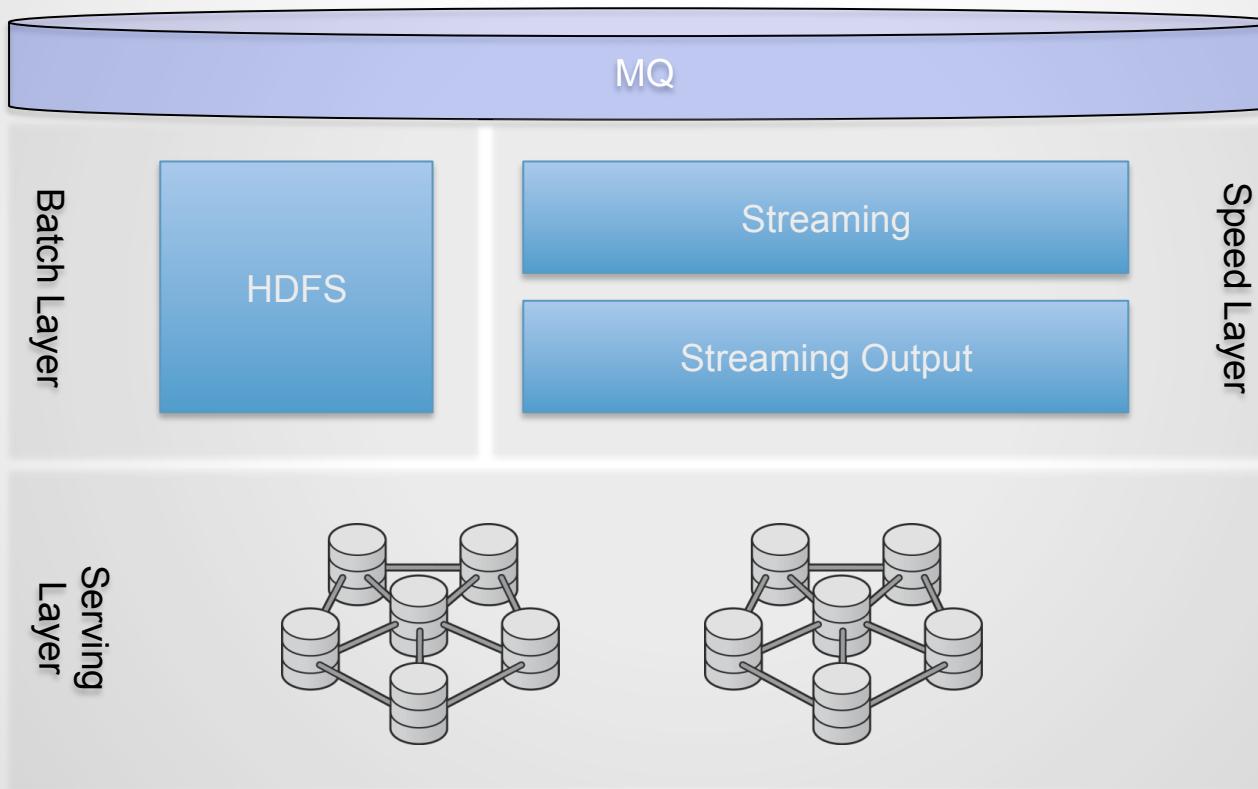


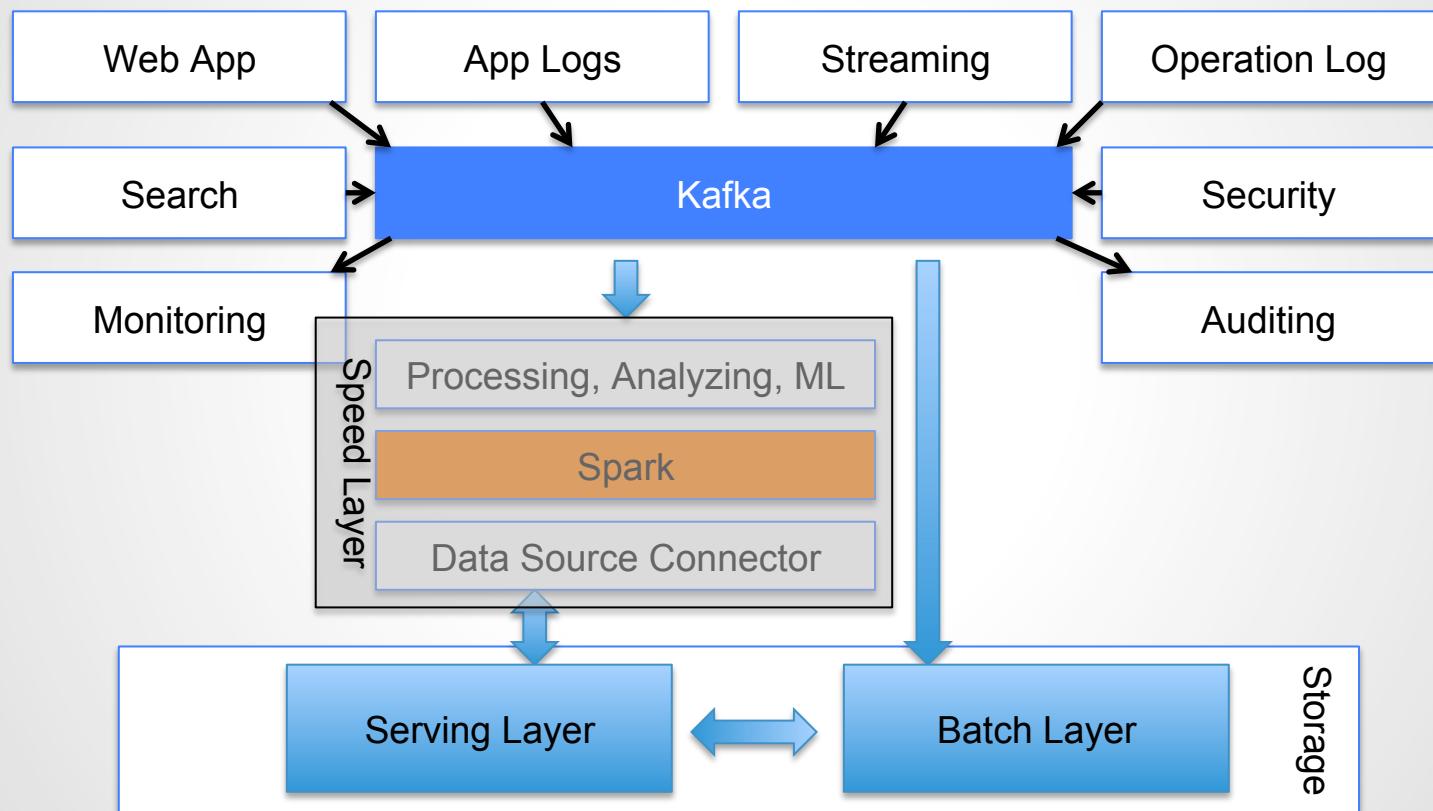
**WHAT NOW?**

# Lambda Architecture

A data-processing architecture designed to handle *massive quantities* of data by taking advantage of both batch and stream processing methods. (wikipedia)

# Lambda Architecture





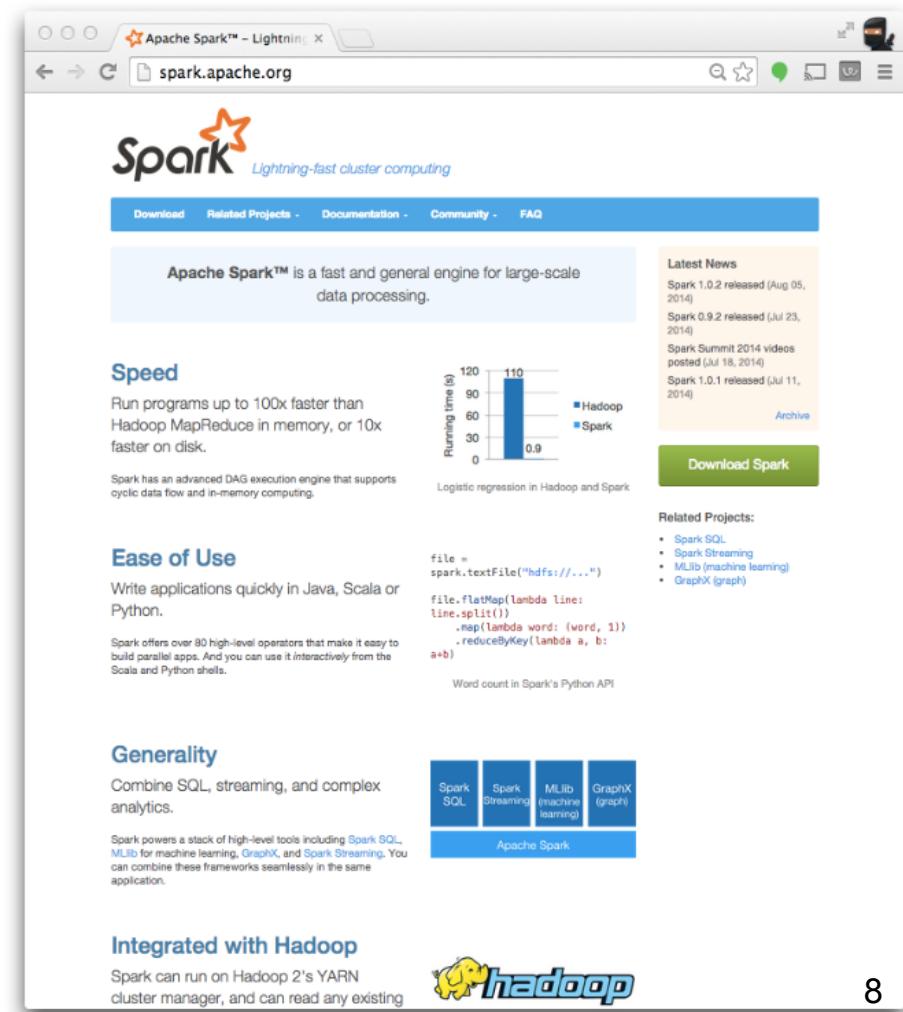


# kafka

- High Throughput Distributed Messaging
- Decouples Data Pipelines
- Handles Massive Data Load
- Support Massive Number of Consumers
- Distribution & partitioning across cluster nodes
- Automatic recovery from broker failures



- Fast, distributed, scalable and fault tolerant cluster compute system
- Enables Low-latency with complex analytics
- Developed in 2009 at UC Berkeley AMPLab, open sourced in 2010, and became a top-level Apache project in February, 2014



The screenshot shows the official Apache Spark website at spark.apache.org. The header features the Spark logo and the tagline "Lightning-fast cluster computing". A navigation bar includes links for Download, Related Projects, Documentation, Community, and FAQ. A "Latest News" sidebar lists recent releases: Spark 1.0.2 (Aug 05, 2014), Spark 0.9.2 (Jul 23, 2014), Spark Summit 2014 videos (Jul 18, 2014), and Spark 1.0.1 (Jul 11, 2014). A large "Download Spark" button is prominent. The main content area highlights "Speed" with a chart comparing Hadoop and Spark's running times for logistic regression, showing Spark is significantly faster. It also covers "Ease of Use" with Scala/Python code snippets and "Generality" by showing how it integrates with other Apache projects like Hadoop, MLlib, and GraphX. The footer features the Hadoop logo.

Apache Spark™ is a fast and general engine for large-scale data processing.

### Speed

Run programs up to 100x faster than Hadoop MapReduce in memory, or 10x faster on disk.

Spark has an advanced DAG execution engine that supports cyclic data flow and in-memory computing.

Running time (s)

System	Running time (s)
Hadoop	110
Spark	0.9

### Ease of Use

Write applications quickly in Java, Scala or Python.

```
file = spark.textFile("hdfs://...")  
file.flatMap(lambda line:  
    line.split()  
    .map(lambda word: (word, 1))  
    .reduceByKey(lambda a, b:  
        a+b))
```

Spark offers over 80 high-level operators that make it easy to build parallel apps. And you can use it interactively from the Scala and Python shells.

### Generality

Combine SQL, streaming, and complex analytics.

Spark powers a stack of high-level tools including [Spark SQL](#), [MLlib](#) for machine learning, [GraphX](#), and [Spark Streaming](#). You can combine these frameworks seamlessly in the same application.

Apache Spark

Apache

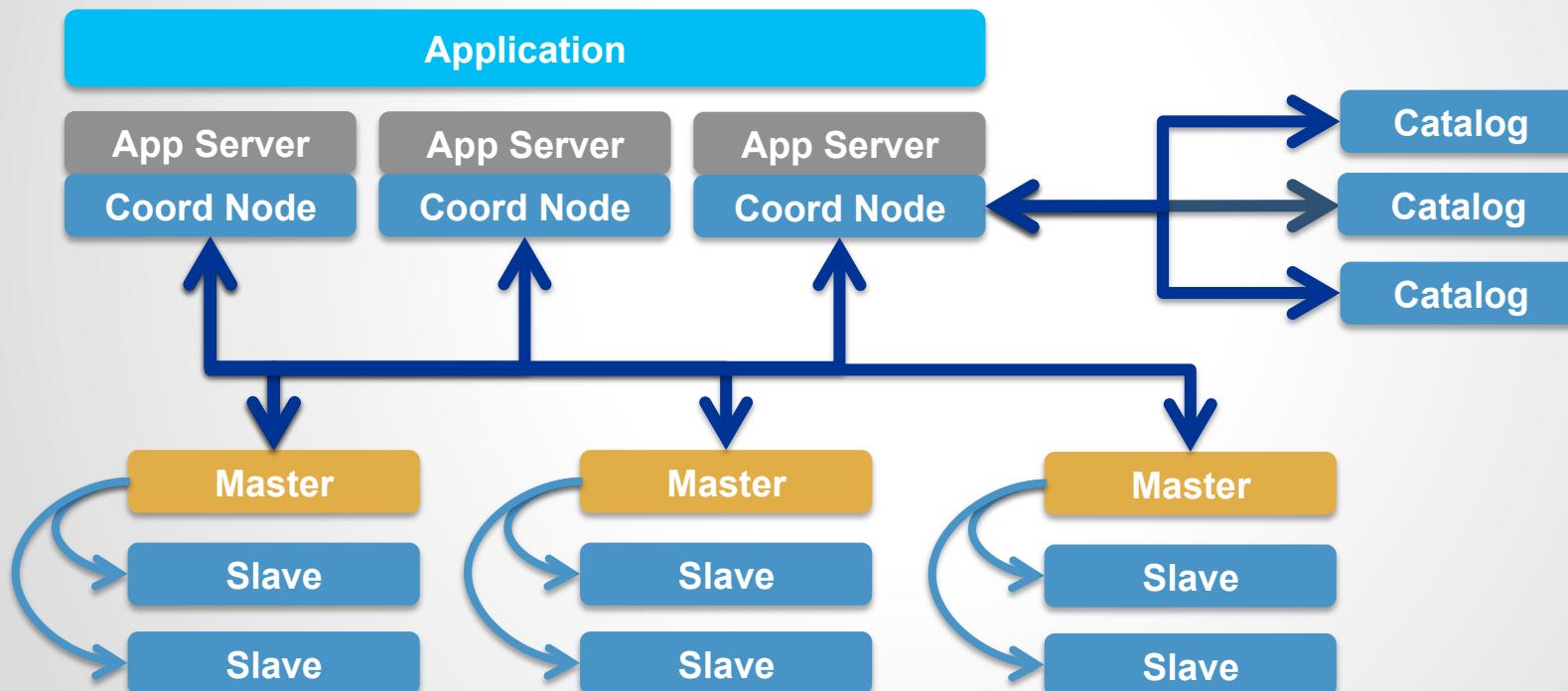
hadoop

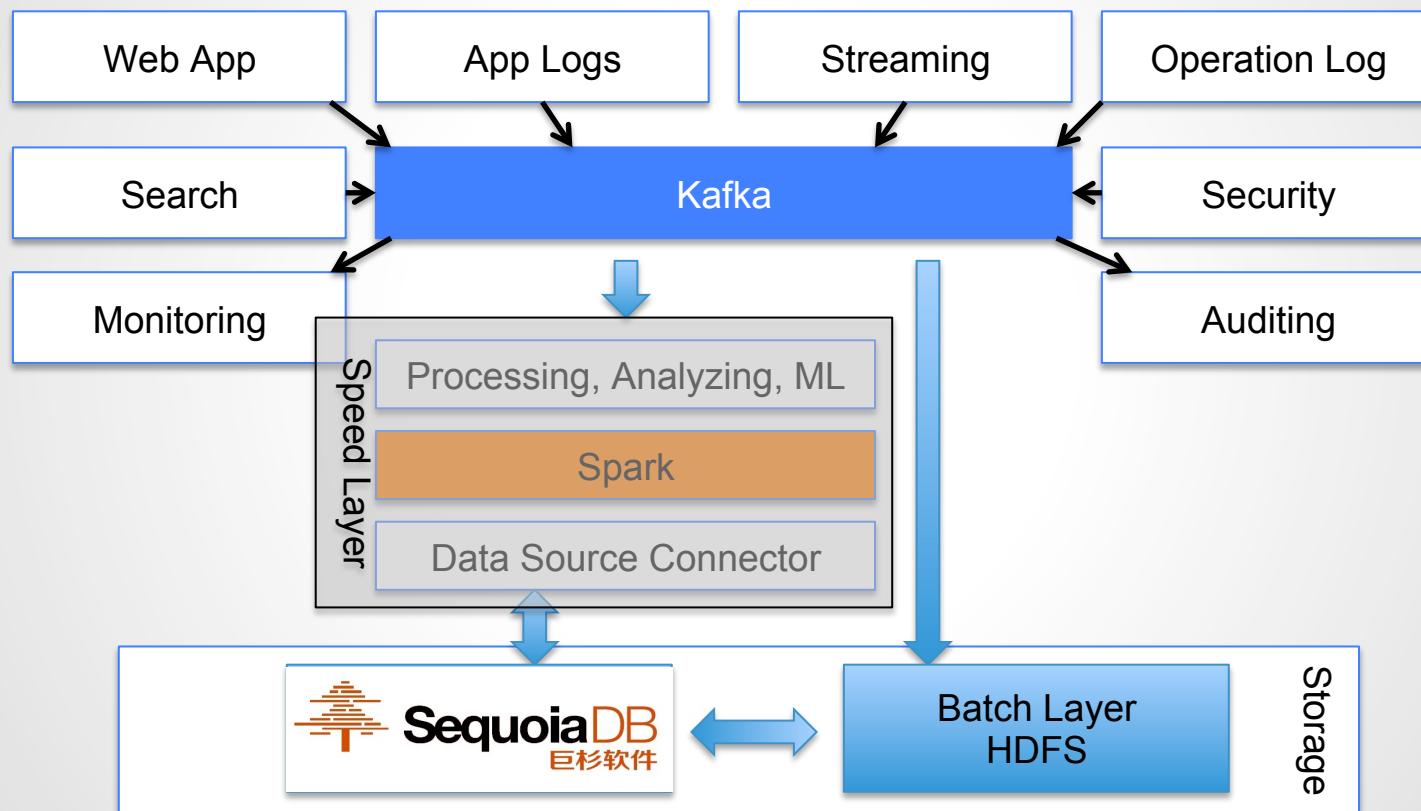


SequoiaDB is a document-oriented NOSQL database that supports JSON transaction processing and SQL. The database is horizontally scalable and provides high performance.

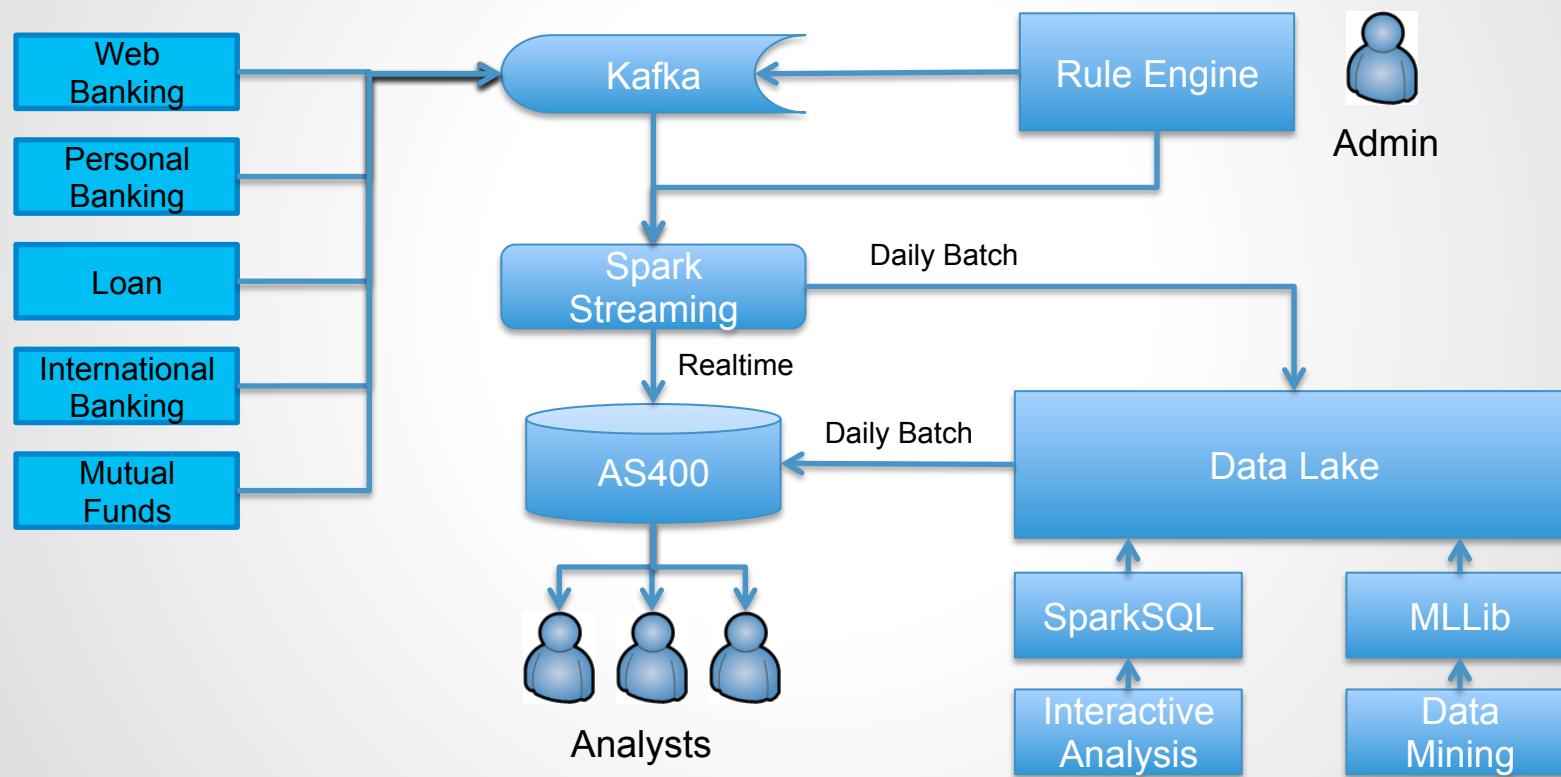


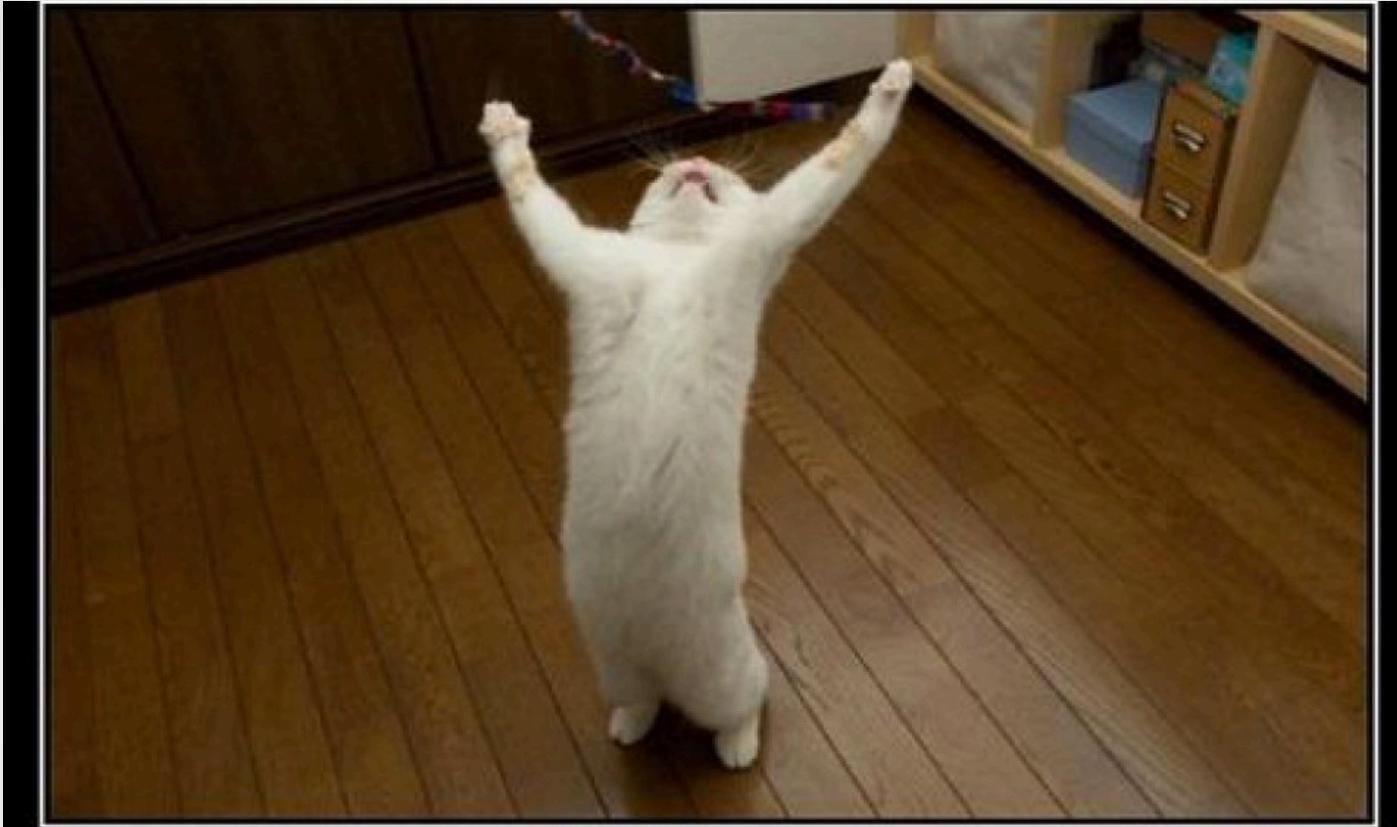
# Storage Layer: SequoiaDB





# Real-time Fraud Detection

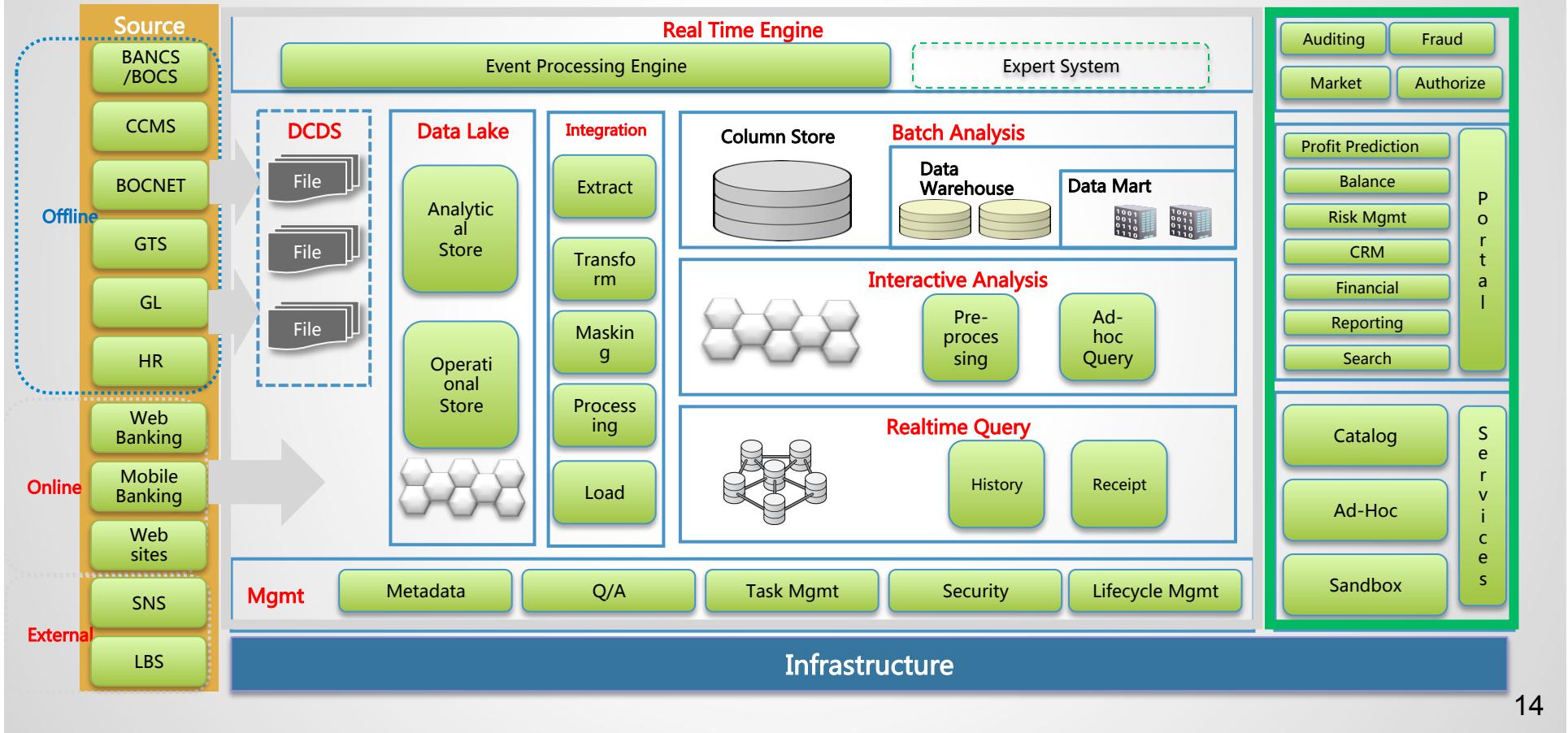




WHY???

WHY GOD WHY!!!

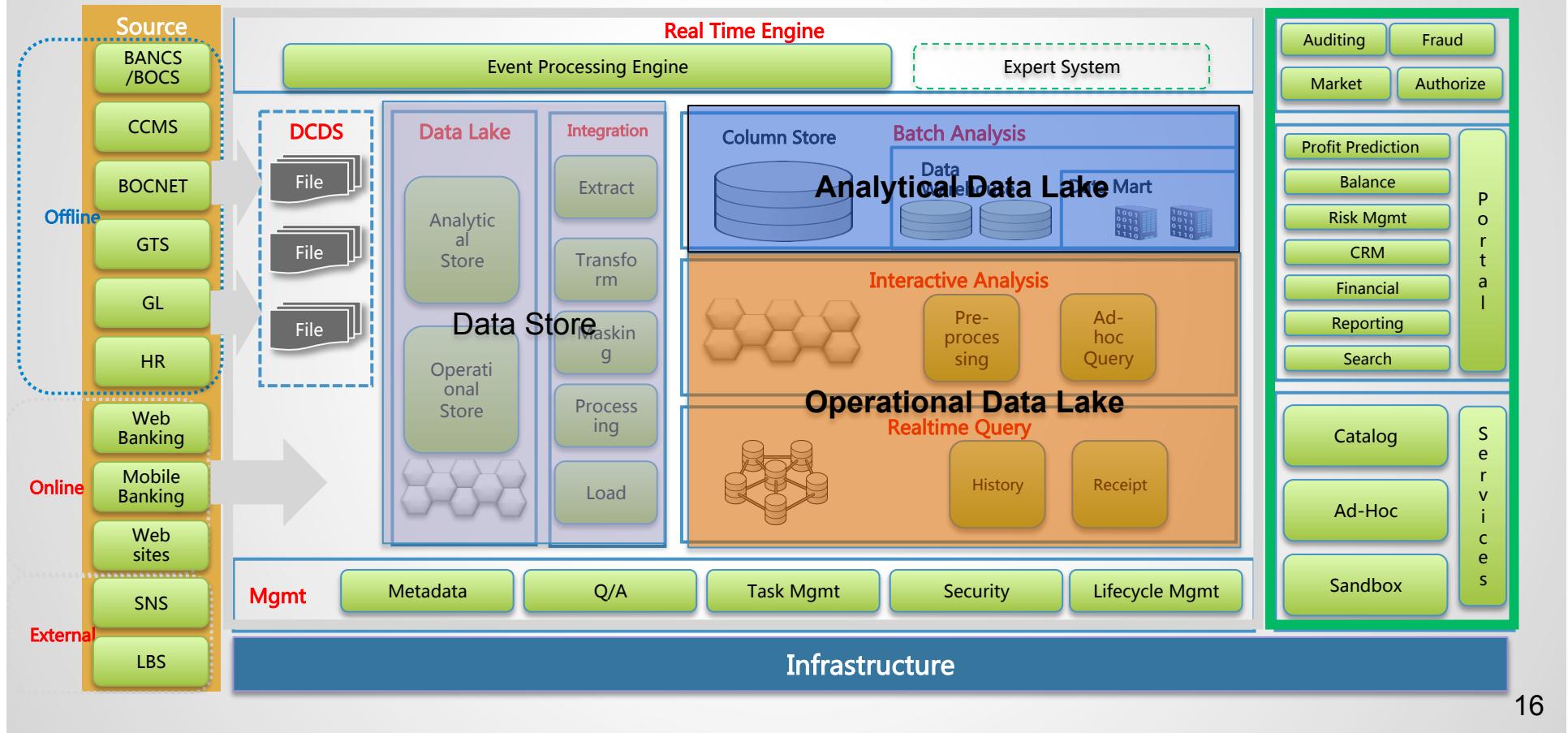
# Enterprise Data Lake Architecture



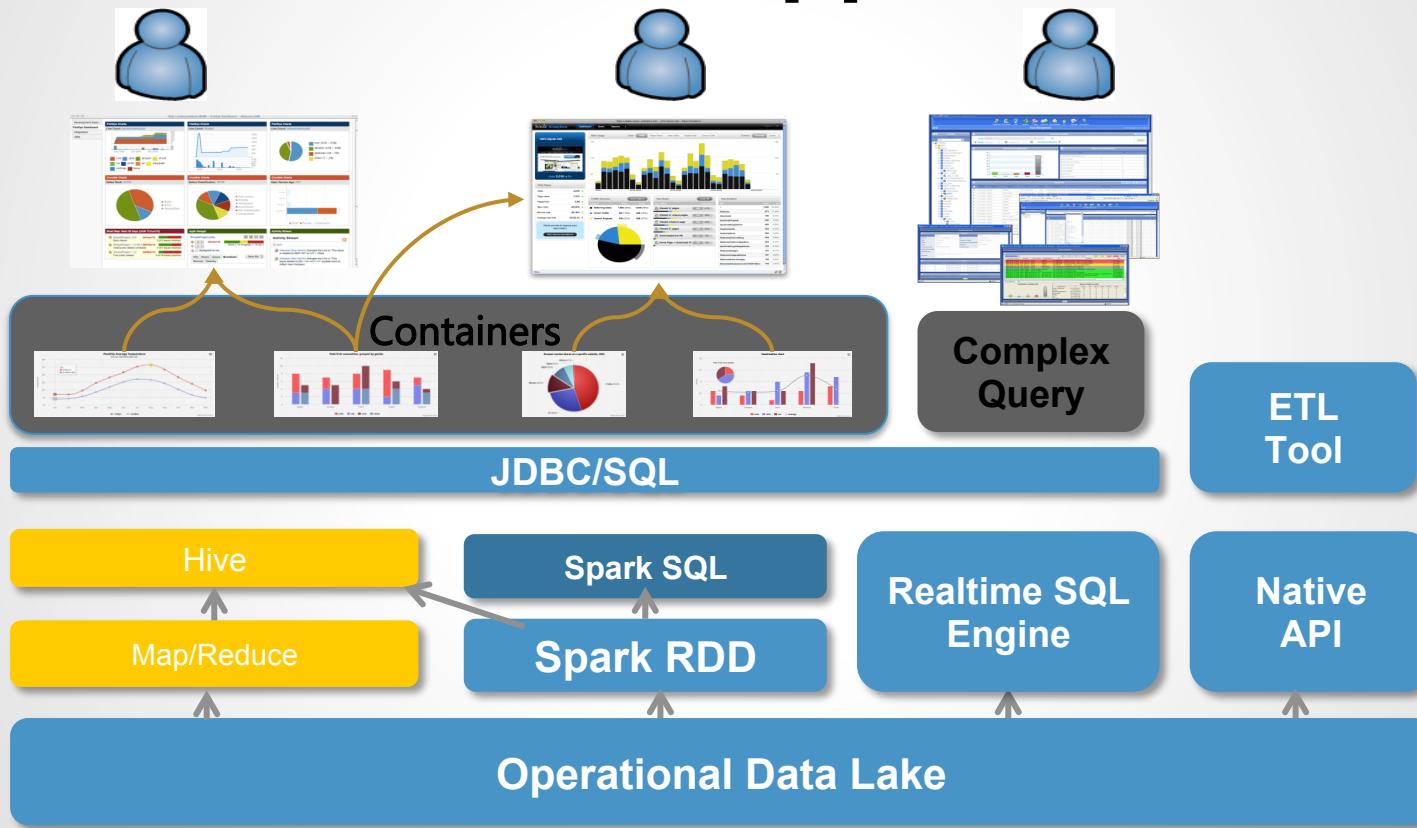
# Data Lake Requirement

- Joint solution of real-time and batch jobs
- Support reporting, OLAP and ad-hoc queries
- HDFS is for reporting system ONLY
- Distributed Database is mandatory for Operational Data Store that supports mixed workload

# Data Lake Architecture



# Mixed Workload Support



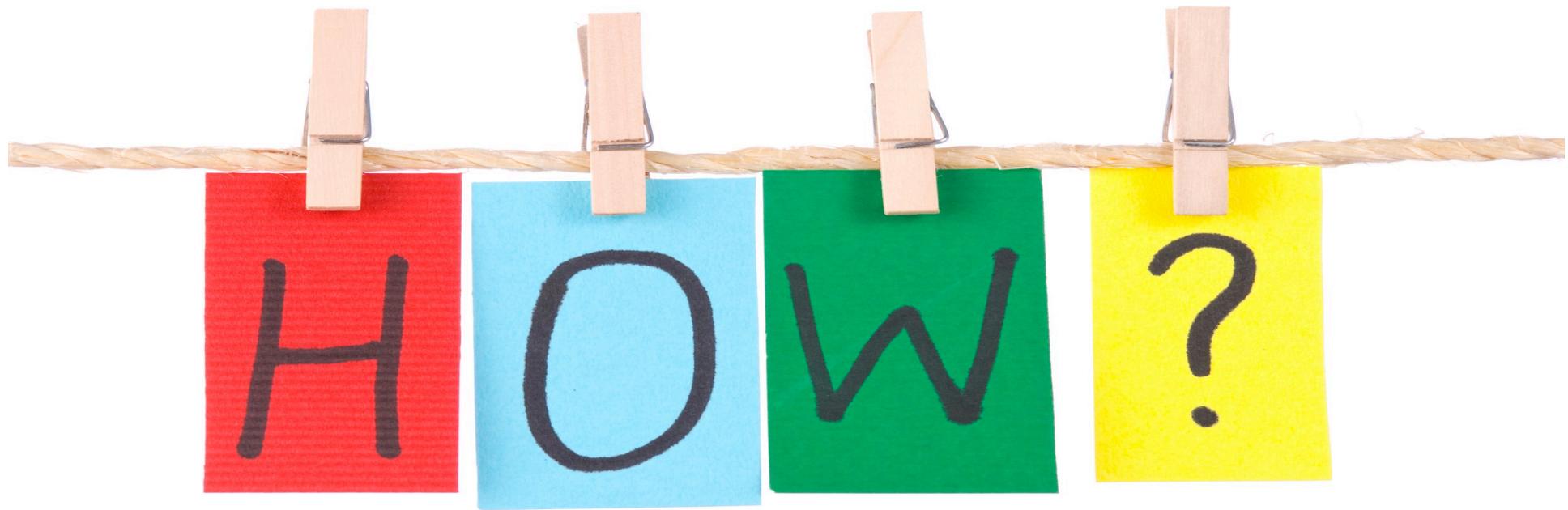
# **Why Spark**

Everybody likes SQL

Streaming is critical for Lambda architecture

Embedded libs/utilities like MLlib and GraphX

Able to handle unstructured data



A string of four colored cards spelling "HOW?" hangs from wooden clothespins. The cards are red, blue, green, and yellow. The word "HOW" is written in black capital letters, and a question mark is written in black on the yellow card.

HOW?

# Spark-SequoiaDB Connector



PRODUCT SPARK SOLUTIONS CUSTOMERS COMPANY BLOG RESOURCES

Training Careers

## COMPANY

All Posts



## Guest blog: SequoiaDB Connector for Apache Spark

August 3, 2015 | by Tao Wang



Partners

Events

Press Releases

## DEVELOPER

All Posts

Spark

Spark SQL

Spark Streaming

MLlib

Spark Summit

Search Blog



Subscribe



This is a guest blog from Tao Wang at [SequoiaDB](#). He is the co-founder and CTO of SequoiaDB, leading its long-term technology vision, and is responsible for the leadership of advanced technology incubations. SequoiaDB is a JSON document-oriented transactional database.

## Why We Chose Spark

SequoiaDB is a NoSQL database that has the capability to replicate data on different physical nodes and allows users to specify which “copy of data” that the application should access. It is capable of running analytical and operational workloads simultaneously on the same cluster with minimal I/O or CPU contention.

The joint solution of Apache Spark and SequoiaDB allows users to build a single platform such that a wide variety of workloads (e.g., interactive SQL and streaming) can run together on the same physical cluster.

## Making SequoiaDB Work with Spark: The Spark-SequoiaDB Connector

# Spark-SequoiaDB Connector

The screenshot shows the GitHub repository page for "SequoiaDB / spark-sequoiadb". The repository has 19 commits, 1 branch, 0 releases, and 1 contributor. The latest commit was made by wangzhonnew on March 31, 2014. The repository has 5 stars and 1 fork. The repository page includes links for Code, Issues, Pull requests, Wiki, Pulse, Graphs, and Settings. It also provides an SSH clone URL.

Spark-SequoiaDB is a library that allows users to read/write data with Spark SQL from/into SequoiaDB collections. <http://www.sequoiadb.com> — Edit

19 commits · 1 branch · 0 releases · 1 contributor

Branch: master

spark-sequoiadb / +

change readme

wangzhonnew authored on 31 Mar

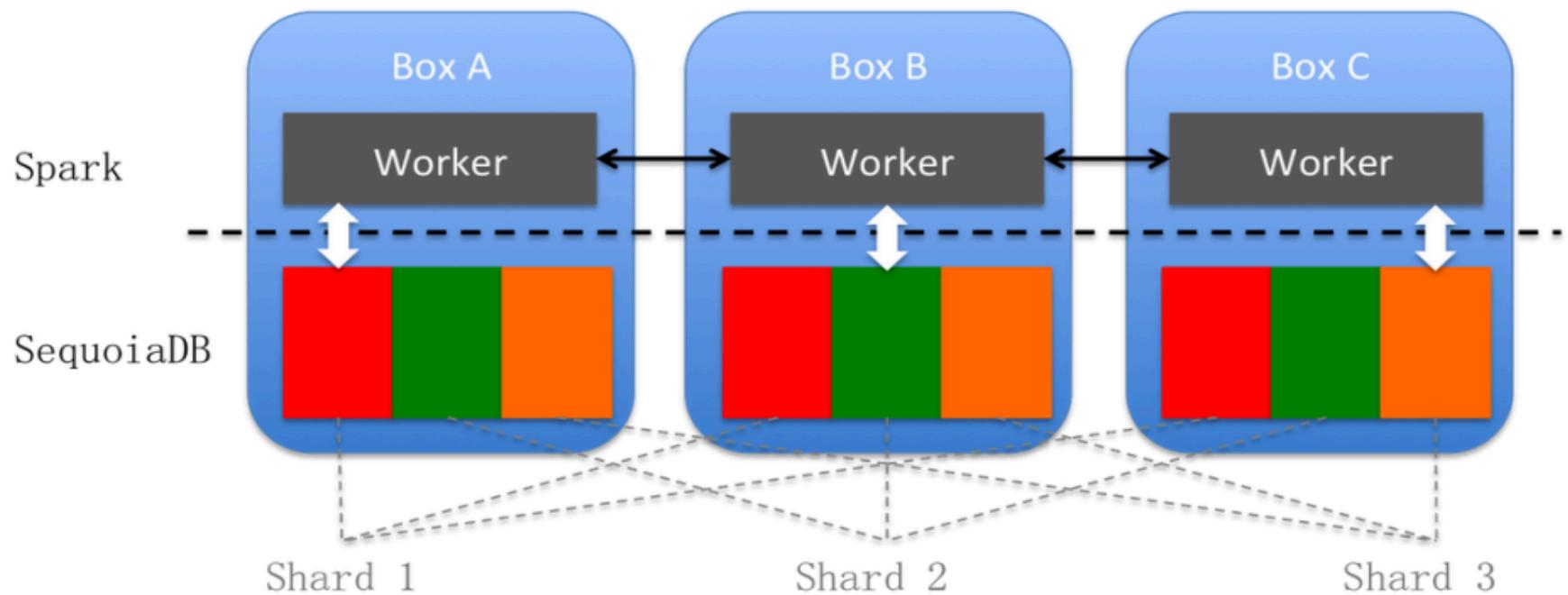
latest commit db09c9eb89

File	Commit Message	Date
project	add sbt-spark-package plugin	5 months ago
sbt	remove part	5 months ago
src	use explain	5 months ago
LICENSE	initial commit	5 months ago
README.md	change readme	5 months ago
build.sbt	remove duplicate license	5 months ago
pom.xml	add scala version	5 months ago

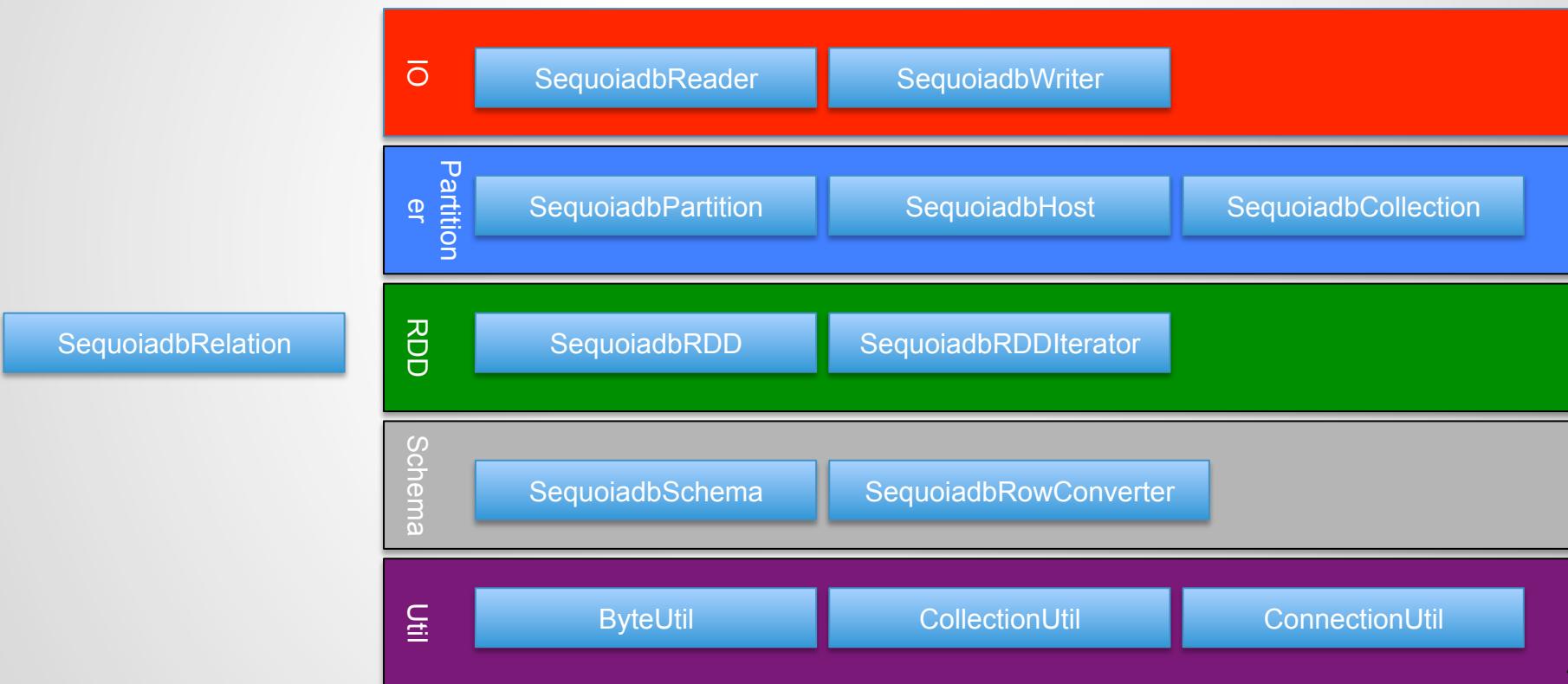
SSH clone URL  
git@github.com:Se...

You can clone with [HTTPS](#), [SSH](#), or [Subversion](#).

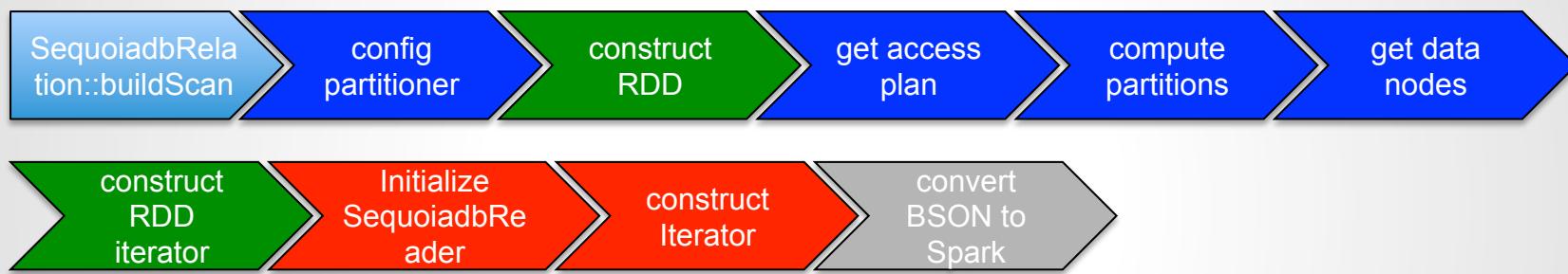
# Spark/SequoiaDB Deployment



# Connector Architecture



# Query Process



# SparkSQL + Connector

```
sqlContext.sql("CREATE temporary table org_department ( deptno string, deptname s
tring, mgrno string, admrdept string, location string ) using com.sequoiadb.spark
  OPTIONS ( host 'host-60-0-16-2:50000', collectionspace 'org', collection 'depart
ment', username 'sdbreader', password 'sdb_reader_pwd')")
res2: org.apache.spark.sql.DataFrame = []

sqlContext.sql("CREATE temporary table org_employee ( empno int, firstnme string,
  midinit string, lastname string, workdept string, phoneno string, hiredate date,
  job string, edlevel int, sex string, birthdate date, salary int, bonus int, comm
  int ) using com.sequoiadb.spark OPTIONS ( host 'host-60-0-16-2:50000', collectio
nspace 'org', collection 'employee', username 'sdb_reader', password 'sdb_reader_
pwd')")
res3: org.apache.spark.sql.DataFrame = []

sqlContext.sql("select * from org_department a, org_employee b where a.deptno='D1
1'").collect().take(3).foreach(println)
[D11,MANUFACTURING SYSTEMS,000060,D01,null,10,CHRISTINE,I,HAAS,A00,3978,null,PRES
,18,F,null,152750,1000,4220]
[D11,MANUFACTURING SYSTEMS,000060,D01,null,20,MICHAEL,L,THOMPSON,B01,3476,null,MA
NAGER,18,M,null,94250,800,3300]
[D11,MANUFACTURING SYSTEMS,000060,D01,null,30,SALLY,A,KWAN,C01,4738,null,MANAGER,
20,F,null,98250,800,3060]
```

# WHERE?

# Real-time fraud detection

## Before

- Hard coded data processing architecture
- Difficult to manage
- Lack of ability to horizontally scale

## Now

- Horizontally scale out
- Dynamic schema for data model changes
- Support ad-hoc and real time queries
- Streaming input

# Requirement

# of stocks: 50k

# of accounts: 300m

# of active accounts / day: 50m

# of transactions / day: 400m

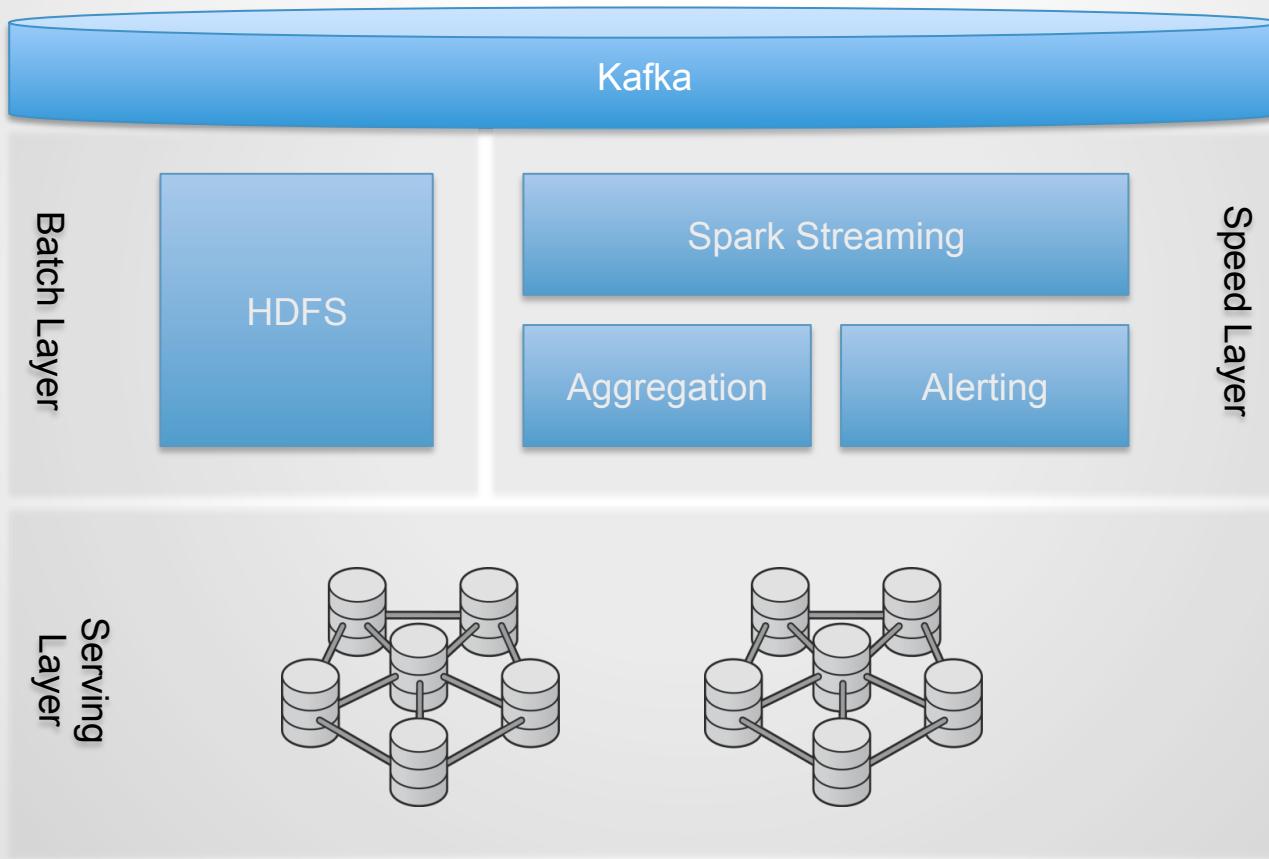
# of trading office: 10k

# of transactions / sec during spike: 1m

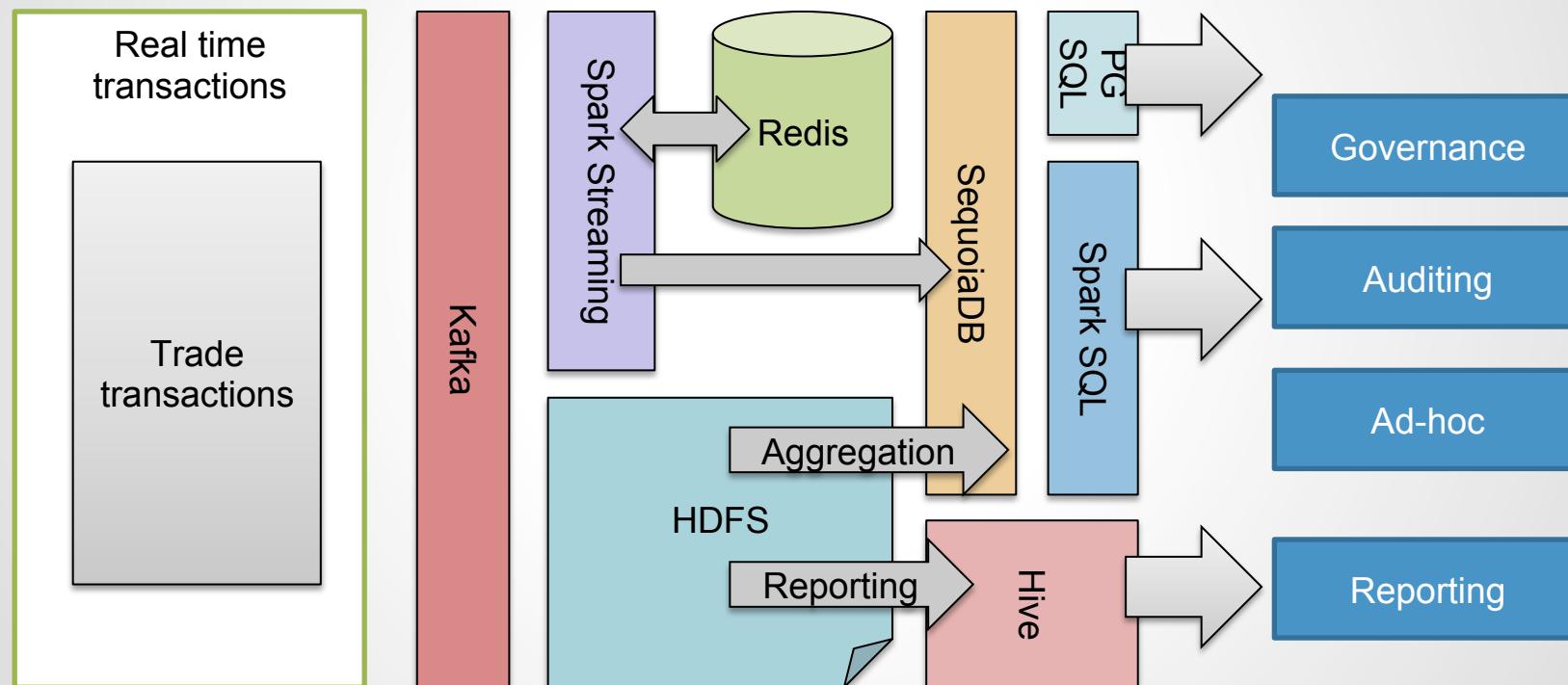
fraud alarm interval: 10 seconds

detection window: 3 min

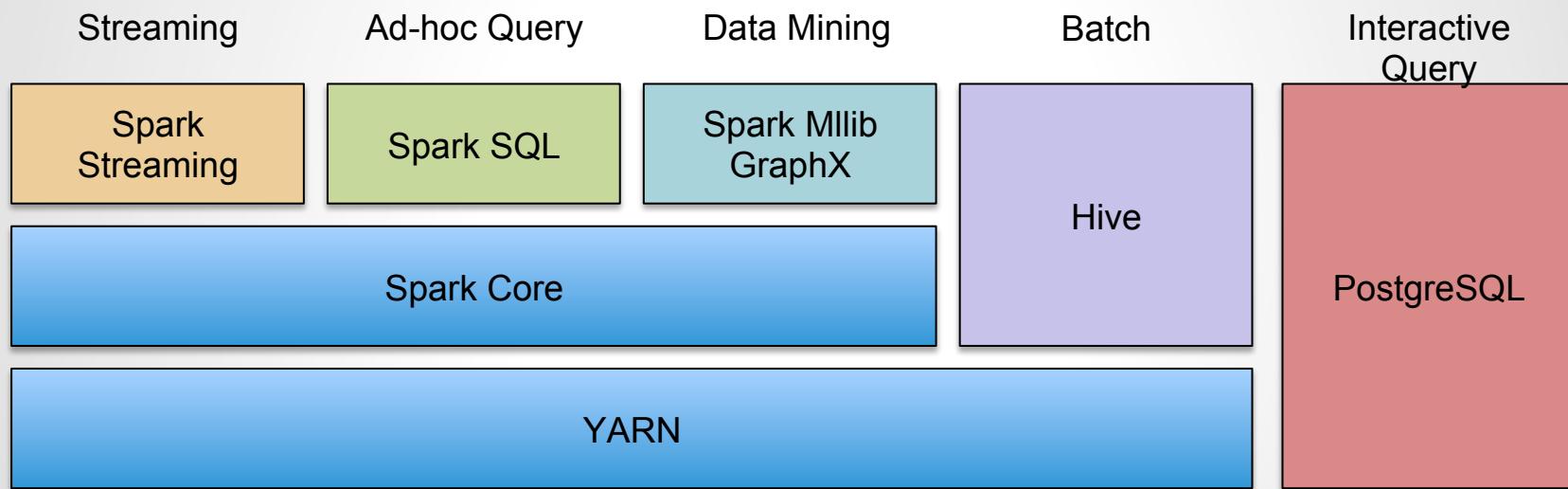
# Architecture View



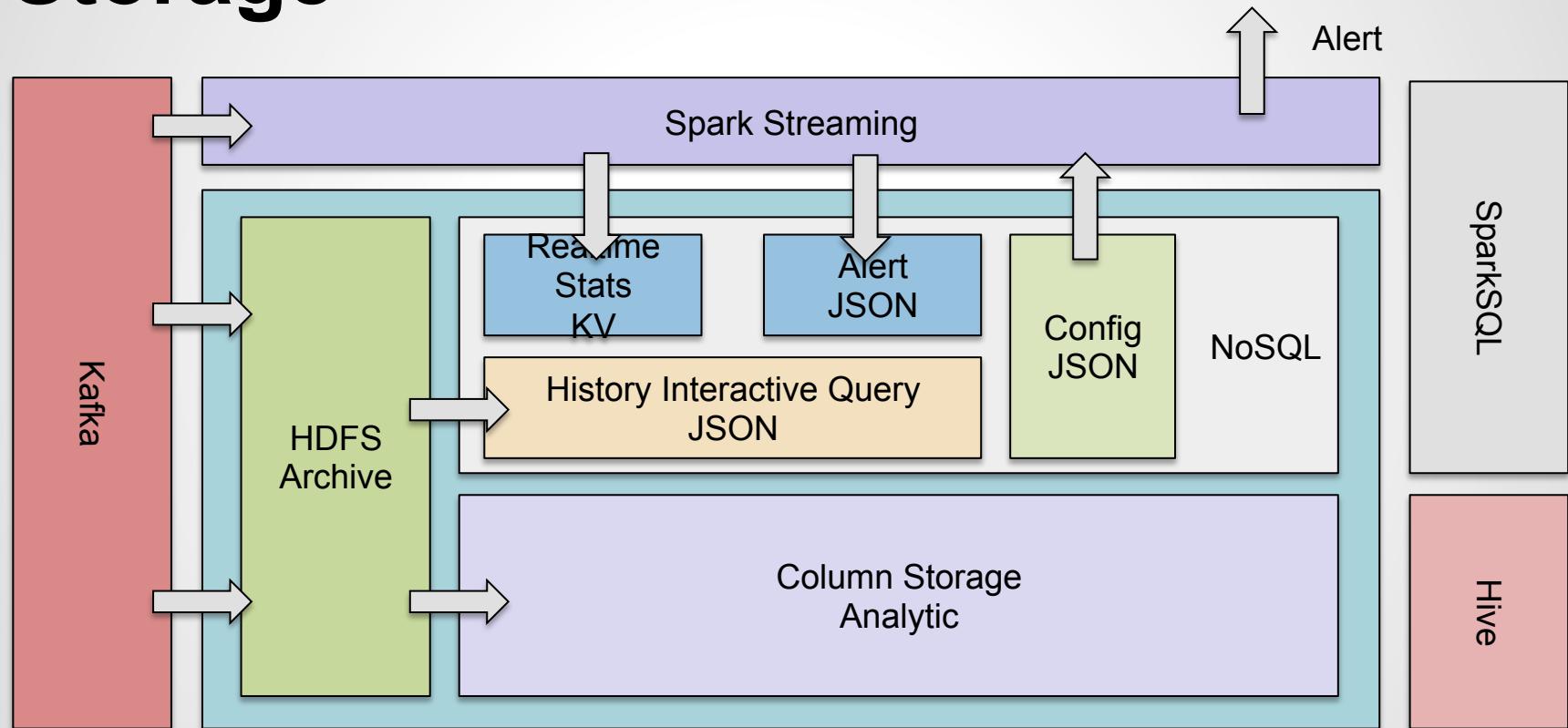
# Architecture



# Computing

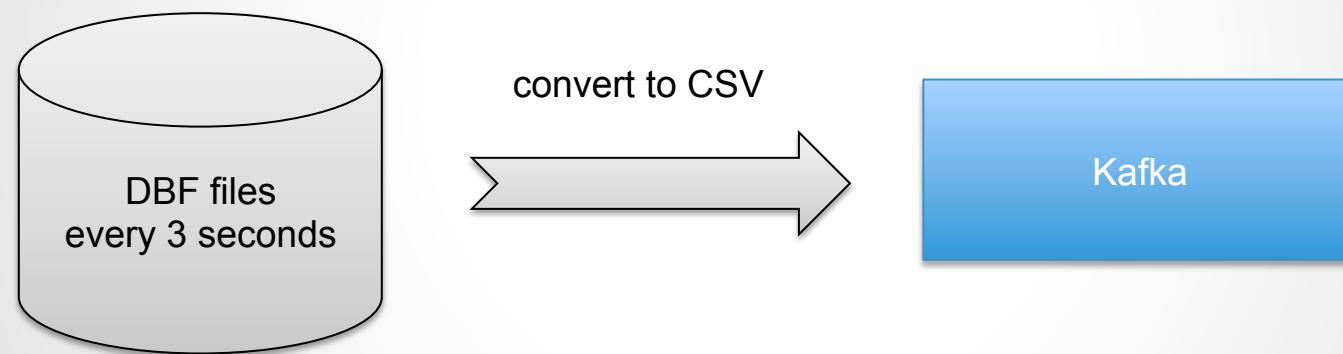


# Storage



# **Deployment**

# 1: Data Input



## 2: Cluster Planning

- Storage: Sizing, Schema, Logical Design
- Computing: Sizing, # of zones, # of workers

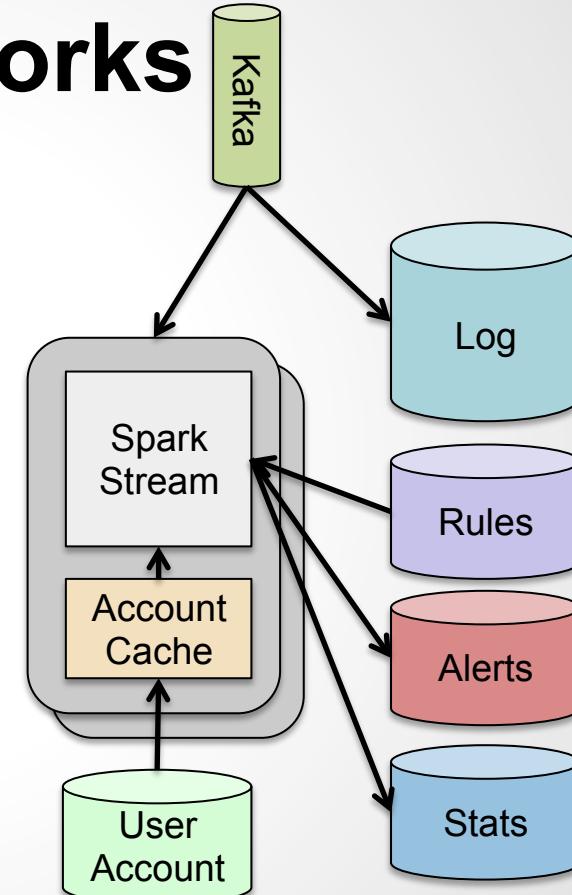
# 3: How Streaming Works

Kafka feeds data to Spark Streaming  
and Log files in HDFS

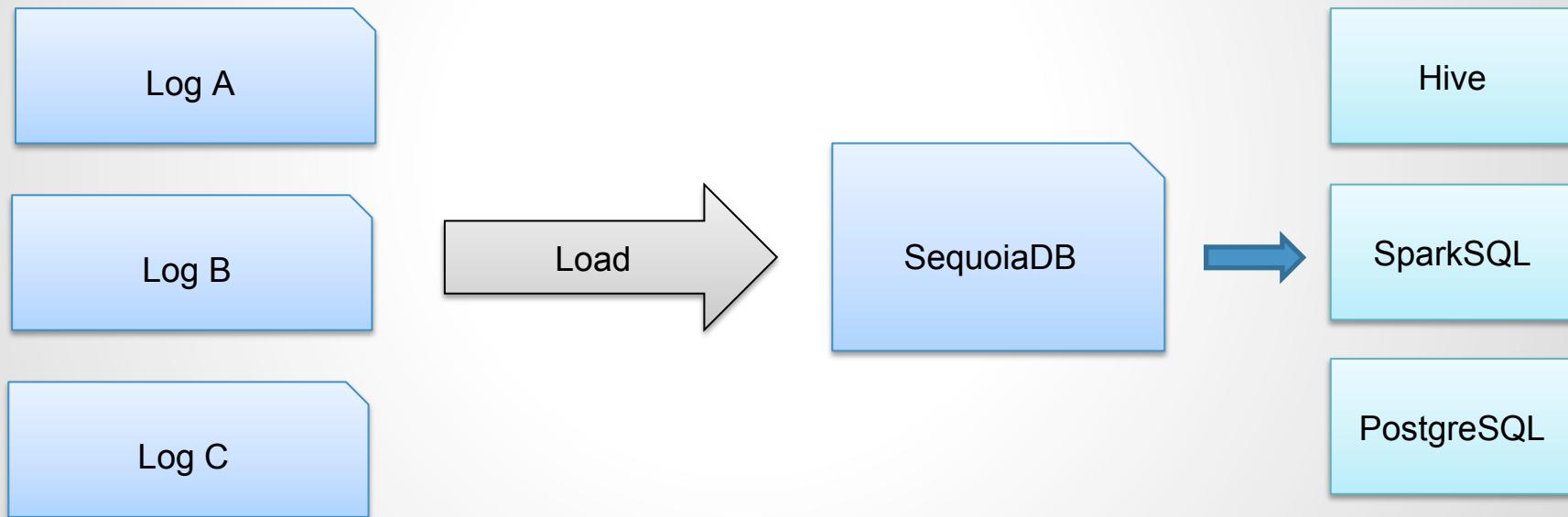
Cache user accounts in every box

Write alerts and aggregated stats to  
NoSQL

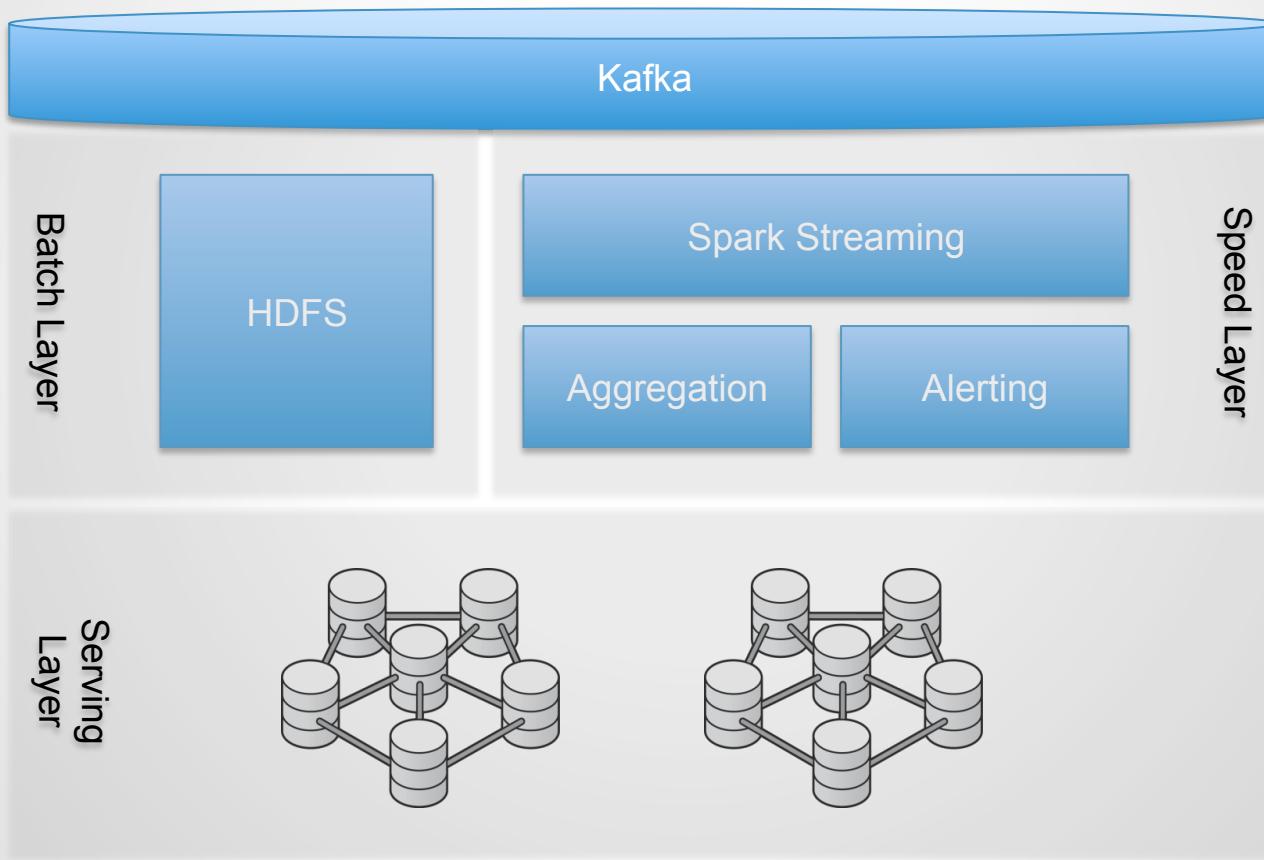
Applications read from aggregated  
results



# 4: Batch Job



# 5. Overall



# **Pitfalls**

# Scheduler

Cannot terminate tasks that created by ThriftServer

Resolved in Spark 1.5

Tasks run in serialized mode in Standalone mode

SET spark.sql.thriftserver.scheduler.pool=accounting;

Need to create many resource pools for parallel jobs

# Functions

OOM caused by beeline java heap size

Script was not able to pass arguments to java process,  
resolved in Spark 1.5

Hostname doesn't work in --master

Use IP address instead

# Monitoring

- Limited monitoring methods, web only
- Number of workers/executors are critical to performance
- SparkSQL is 10 times faster than Hive for big table joins

# **Spark/SequoiaDB Connector**

- Predicate Pushdown is critical to performance
- Milliseconds query if predicates hit index
- Database design is critical for real-time queries
- Specify multiple coordinator nodes in connection string

# We Are Hiring!

## Thanks!

My Contact Info:

- [taoewang@sequoiadb.com](mailto:taoewang@sequoiadb.com)
- [www.sequoiadb.com](http://www.sequoiadb.com)

