cloudera®

Introduction to the Hadoop Ecosystem

Sean Mackrory | Rocky Mountain Oracle Users Group

Big Data Special Interest Group

About Me

- Software Engineer @ Cloudera, Integration / Cloud Platform teams
  - Cloudera QuickStart VM
  - Cloudera Live
- PMC member @ Apache Bigtop
- PPMC member @ Apache Sentry (incubating)

cloudera

About You

cloudera

“That original MapReduce paper from Google… was like receiving knowledge from an alien race.”

—James Mickens

The “Original” Hadoop

- Apache Hadoop was originally HDFS + MapReduce – highly scalable, but…
  - Multiple single points of failure
  - Exclusively for high-latency, high-bandwidth batch jobs
  - Required an advanced user and a lot of boiler-plate

The Enterprise Data Hub (Bigger than a data lake)

- A reliable, scalable and performant system for all data, with:
  - Storage for almost all cases
  - Computation for almost all cases
  - Integration with other systems for other cases
  - Security and management capabilities

cloudera

I am your host, join me for a tour of the stack…

- Virtually everything in Cloudera's stack is explained here
- Some things here are not part of Cloudera's stack, but included for context
- This is roughly organized by layers, from the “bottom” of the stack up

cloudera

Cluster coordination

Apache ZooKeeper

cloudera

Apache ZooKeeper

- A replicated, hierarchical file system for small data (< 1 MB)
- Nodes can have data & children, can be ephemeral & sequential
- Writes are synchronous and versioned
- Use it for master election, node discovery, distributed locks, etc.

File system

Apache Hadoop: HDFS

Apache Hadoop: HDFS

- Breaks files into blocks (e.g. 512 MB), replicated to DataNodes (e.g. 3)
- NameNodes store metadata, real data stream directly to / from DataNodes
- “Write Once, Read Many”, “Create Read Append”

File Formats and Compression

Apache Avro, Apache Parquet, Snappy, LZO

cloudera

A Word on Schema

- Since this is an Oracle User Group, let's clarify that Hadoop is “schema-on-read”
- You can store text files, images, whatever you want in HDFS
- All these things have some structure, it just might be a loose structure
- The trick is being able to deal with evolving or inconsistent schemas

“If your data is schema-less, what that really means is you don't know what the schema is, and that's not good.”

—Gwen Shapira

Apache Avro

- Binary file format, stores schema with data, similar to JSON
- “Splittable” - records aren't cut in half when processed in Hadoop

cloudera

Apache Parquet

- Stores in column-major format instead of row-major
- Optimizes for compression and analytic I/O patterns
- Similar ideas found in (O)RCFile & Avro's “Trevni” format

Compression

- The ideal compression for Hadoop is 'splittable' like the file formats
  - Blocks must be compressed independently
- Snappy is recommended
- LZO can perform better (GPL Extras is an optional add-on to CDH)

cloudera

NoSQL Storage

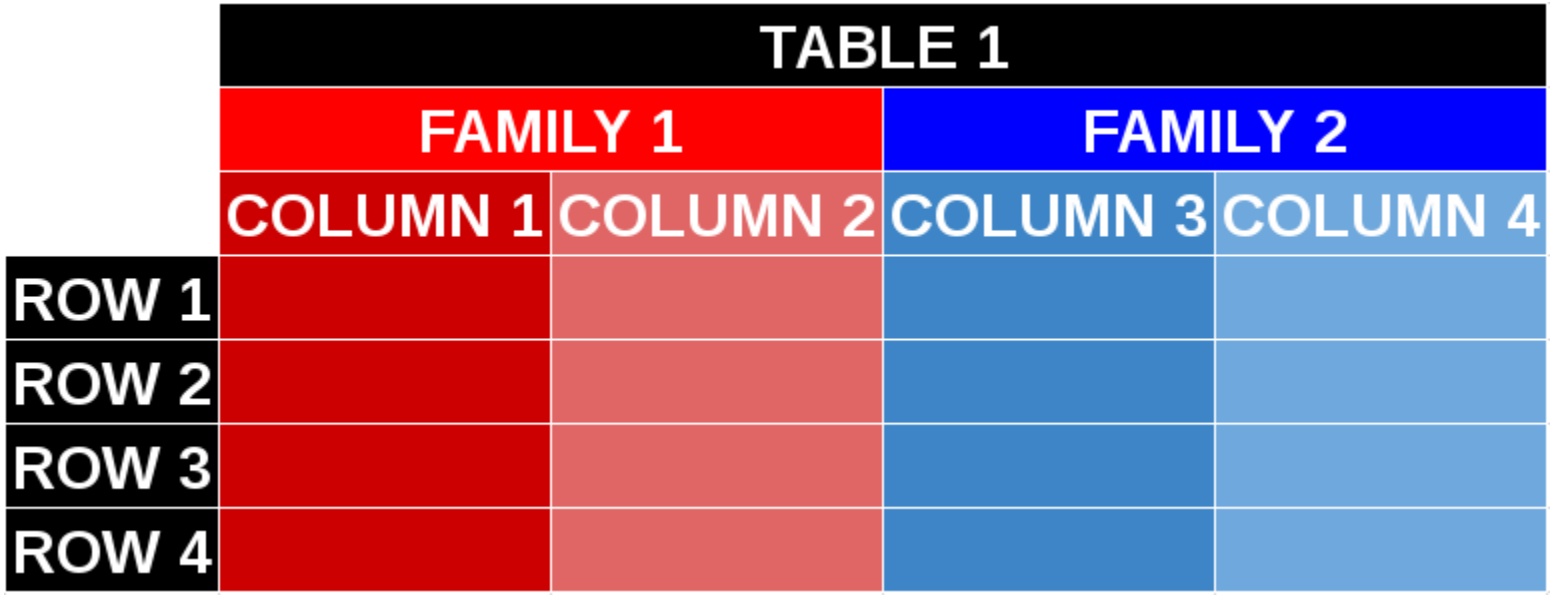
Apache HBase, Apache Accumulo

cloudera

Apache HBase

- Random, read-write access to billions of rows, millions of columns (in families)
- Mutations are logged, compacted, all stored in HDFS (cells are versioned)
- Can be used as an alternate storage layer directly in Hadoop
- Richer data model than a key-value store

Apache HBase



Apache Accumulo

- Based on Google BigTable, like HBase
- First to have cell-level access controls: popular with government users
- HBase RegionServers vs. Accumulo TabletServers

cloudera

Resource Management

Apache Hadoop: YARN

cloudera

Apache Hadoop: YARN

- Separates the scheduling and management from the MR framework
- Cluster has one or more ResourceManagers and NodeManagers
- YARN allocates and manages containers for ApplicationMasters and workers
- Critical for multi-tenancy

cloudera

Execution Engines

Apache Hadoop: MapReduce, Apache Spark

cloudera

Apache Hadoop: MapReduce “mappers”

- Map phase: (K^1,V^2) -> (K^2,V^2)
  - Brings the computation to the data, “embarrassingly parallel”
  - Extract needed values, apply record-level predicates, etc.

Apache Hadoop: MapReduce “reducers”

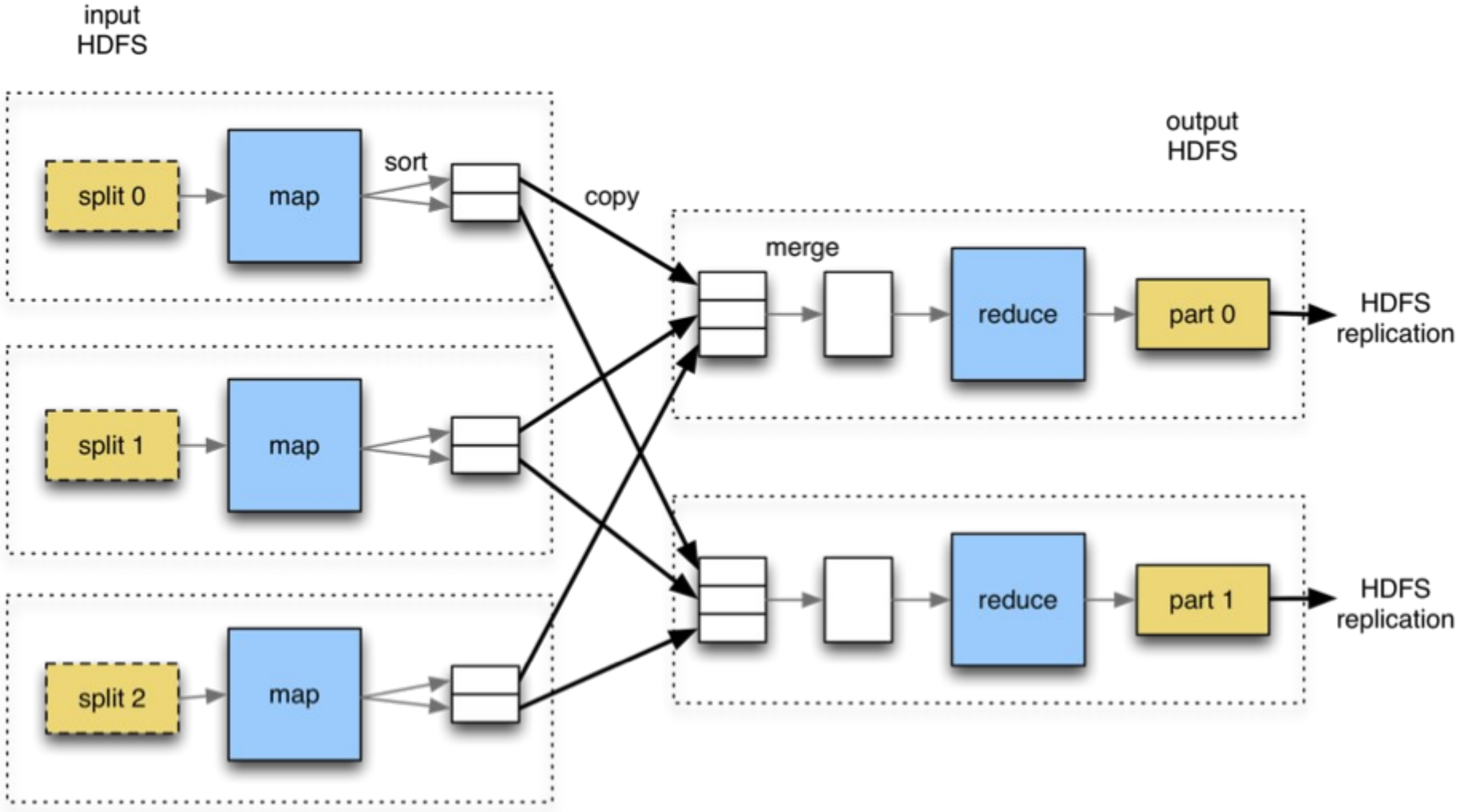
- Shuffle phase sorts by key and groups into other partitions

- Reducers: $(K^1,V^1,V^2...V^N)$ -> (K^2,V^2)

  - Aggregations, grouping, other predicates

cloudera

Apache Hadoop: MapReduce



http://blog.cloudera.com/wp-content/uploads/2014/03/ssd1.png

Apache Spark

- RDDs: Datasets stored in memory, along with DAGs for how they are produced
- Higher-level APIs, but still uses the “map” and “reduce” concepts heavily
- Use directly from Scala, Java, or Python

SQL

Apache Hive, Impala (+ Llama), Apache Sqoop, Apache Phoenix and more…

cloudera

Apache Hive

- Compiles SQL queries into MapReduce jobs: very flexible for batch
- Stores table metadata / schema in “MetaStore”

Cloudera Impala

- MPP query engine intended for interactive and high-performance queries
- Reads directly from the file system using run-time code generation
- Shares metadata / JDBC drivers with Hive
- Llama: Low-Latency Application Master for YARN integration

cloudera

Apache Sqoop

- For moving data between data sources (esp. RDBMS) and Hadoop
- Creates a MapReduce job for each table to transfer splits in parallel

Other Big Data SQL Engines

- Apache Phoenix implements a SQL layer over Hbase
- Apache Drill, Apache Tajo (incubating), Apache MRQL (incubating), Apache Spark SQL (formerly Shark)
- Cluster functionality is increasingly popular in other RDBMS

Search

Apache Solr, NGData HBase Indexer, other integration points

cloudera

Apache Solr

- Full text search, faceted search, range buckets, etc.
- Documents still have schema: fields are stored, indexed, or both
- With Flume + Morphlines, data can be indexed as it streams in
- Datasets can be indexed in batch with MapReduce

NGData HBase Indexer

- Indexes data in HBase for retrieval by Solr
- As data mutates, changes are “replicated” through the indexer and re-indexed

cloudera

Stream Processing

Apache Kafka, Apache Flume, Apache Spark (Spark Streaming)

cloudera

Apache Flume

- Ingesting streams of data into HDFS
  - Log files, network streams, telemetry, etc.
- Integrated with Morphlines to do transformations in-flight

Apache Kafka

- "Distributed commit log" publish / subscribe system
- Messages go into "topics", from "producers", through "brokers", to "consumers"
- Recently got much better integration for ingest into Hadoop

Apache Spark: Streaming

- Processes data in micro-batches (batch size can be 1 if you want)
- "Lambda architecture": share code between batches and streams
  - Batches for high-bandwidth comprehensive views
  - Streams for low-latency up-to-date views
- See also: Apache Storm, Apache S4 (incubator)

Machine Learning

Apache Mahout, Apache Spark: MLLib, Cloudera Oryx

cloudera

Machine Learning

- Clustering, classification, regression algorithms, etc.
- Mahout implements these for MapReduce, MLLib for Spark
- Oryx focuses less on algorithms, more on infrastructure
  - Not just for building models, but updating and querying them in real-time

Graph Processing

Apache Spark + libraries, Apache Giraph

cloudera

Graph Processing

- GraphX (superseded Bagel, based on Pregel) allows graph processing in Spark
  - Algorithms benefit from the in-memory, iterative programming model
- Apache Giraph compiles jobs down to MapReduce

Higher Level Frameworks

Apache Pig, Apache DataFu, Apache Crunch, Apache Tez,
Apache Oozie, Kite SDK

cloudera

Apache Pig & Apache DataFu (incubating)

- Pig Latin is a high-level language for Big Data processing
- Apache DataFu (incubating) is a library of functions for data mining and statistics on Pig and MapReduce

Apache Crunch and Apache Tez

- Apache Crunch: Java library for MapReduce pipelines
- Apache Tez: Framework for DAG of MapReduce-style tasks

Apache Oozie

- “Workflow scheduler” for managing DAGs of tasks

- Tasks can be Hadoop, Hive, Pig Sqoop, etc.

Kite SDK

- Higher-level tools for working with abstract “datasets”
- “Morphlines”: command-based pipelines for ETL

cloudera

User Interface

Cloudera Hue

Cloudera Hue

- Web application for Big Data
- Makes authentication and interaction much easier
- Apps for most of the tools we've talked about

cloudera

Security

Apache Sentry (incubating), KeyTrustee

cloudera

Apache Sentry (incubating)

- Authorization framework (requires authentication, e.g. Kerberos)
- Currently integrates with Hive / Impala, Solr and HDFS

Key Trustee KMS

- Key Management Server that plugs in to Hadoop
- Enables data-at-rest encryption in HDFS

Cloudera Manager

+ Navigator, BDR, Director

cloudera

Cloudera Manager

- CM: service & configuration management, monitoring
- BDR: Backup / disaster recovery
- Navigator: Audit access, trace data lineage, policies, etc.
- Director: self-service cloud deployments

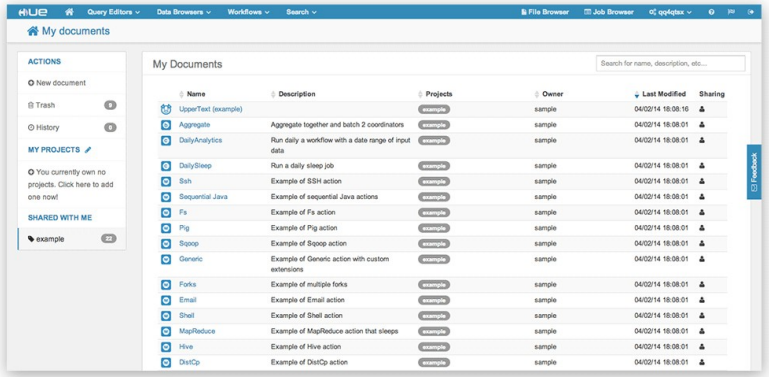
cloudera

Cloudera Live

Try this at home, kids!

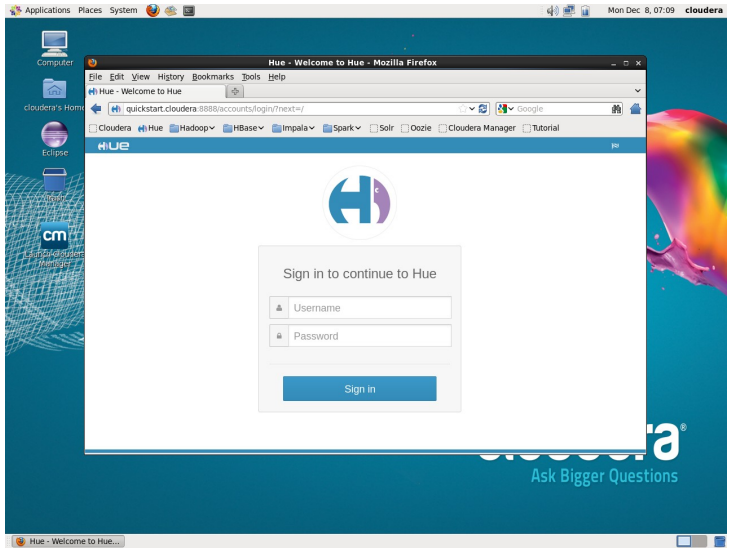
cloudera

live.cloudera.com



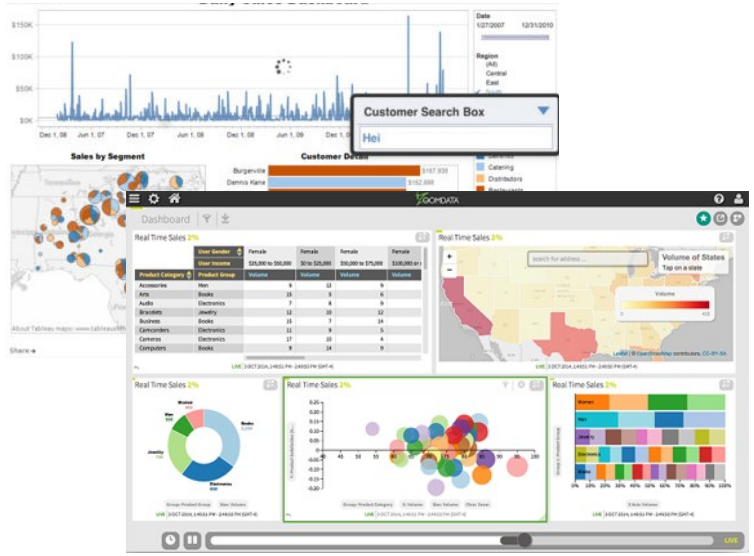
Demo Environment

Read-only access to Hue on a shared cluster



QuickStart VMs

Downloadable virtual appliances

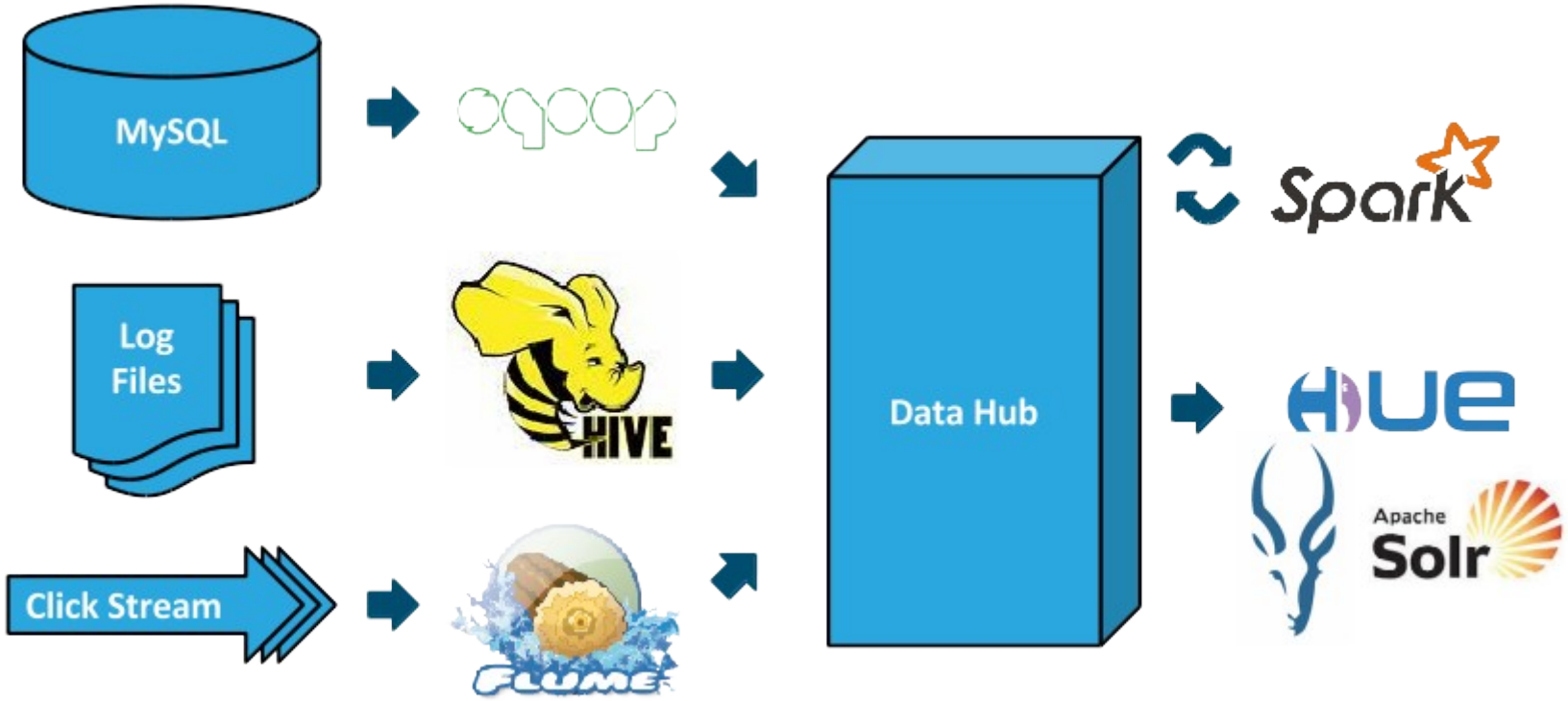


Cloudera Live

Free trial clusters on GoGrid: ZoomData, Tableau, Trifacta

cloudera

The DataCo Story



Thank you

@SeanMackrory
sean@cloudera.com
mackrorysd@apache.org