Hadoop and MySQL for Big Data

Alexander Rubin

September 28, 2013

About Me

Alexander Rubin, Principal Consultant, Percona

- Working with MySQL for over 10 years
  - Started at MySQL AB, Sun Microsystems, Oracle (MySQL Consulting)
  - Worked at Hortonworks
  - Joined Percona in 2013

Agenda

- Hadoop Use Cases
- Inside Hadoop
- Big Data with Hadoop
- MySQL and Hadoop Integration
- Star Schema benchmark

Hadoop: when it makes sense

BIG DATA

Big Data

"3Vs" of big data

- Volume
  - Petabytes
- Variety
  - Any type of data – usually unstructured/raw data
  - No normalization
- Velocity
  - Data is collected at a high rate

http://en.wikipedia.org/wiki/Big_data#Definition

Hadoop Use Cases

Clickstream: a data trail for a user visiting website

Task: store and analyze

Hadoop Use Cases

Risk modeling: fraud prevention (banks, credit cards)

- Store all customer activity
- Run analysis to identify credit card fraud

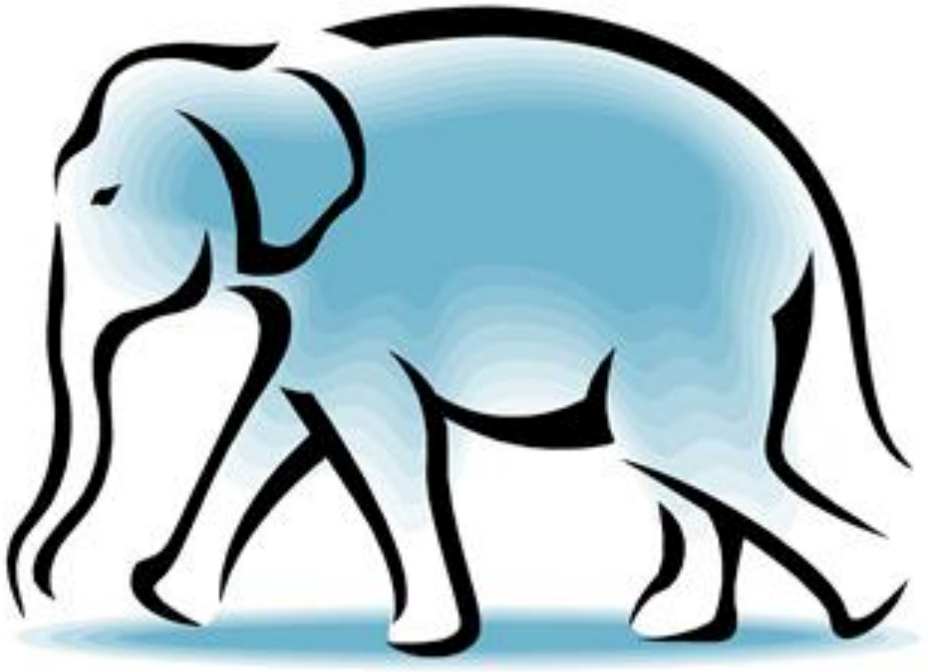
Hadoop Use Cases

Point-of-Sale Analysis

Hadoop Use Cases

Stocks / Historical prices analysis

Inside Hadoop



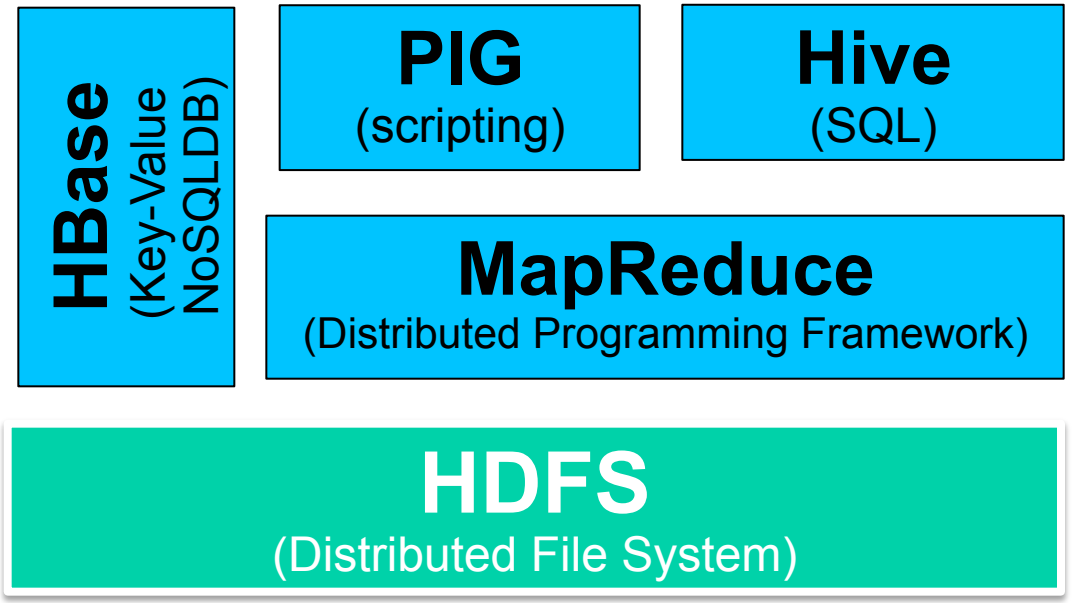
Where is my data?

Data Nodes

HDFS
(Distributed File System)

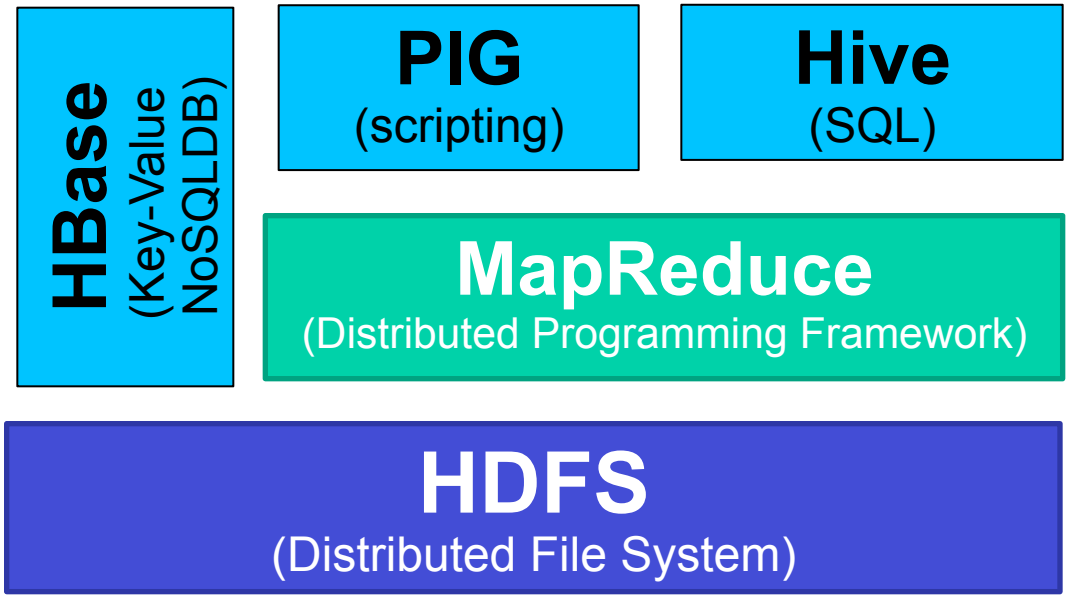
- HDFS = Data is spread between MANY machines

Inside Hadoop



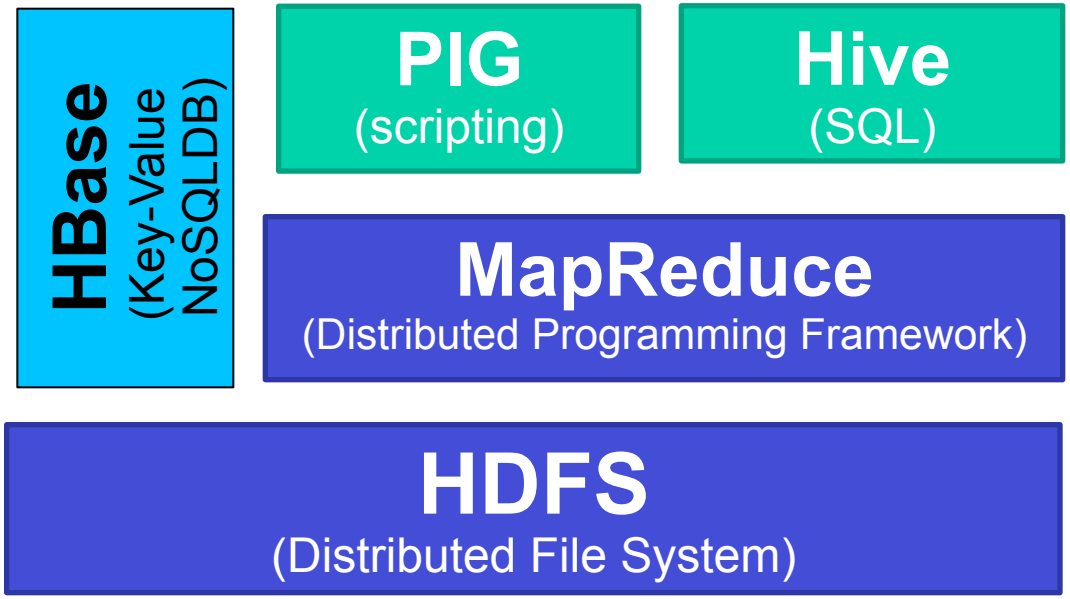
- HDFS = Data is spread between MANY machines
- Write files, “append-only” mode

Inside Hadoop



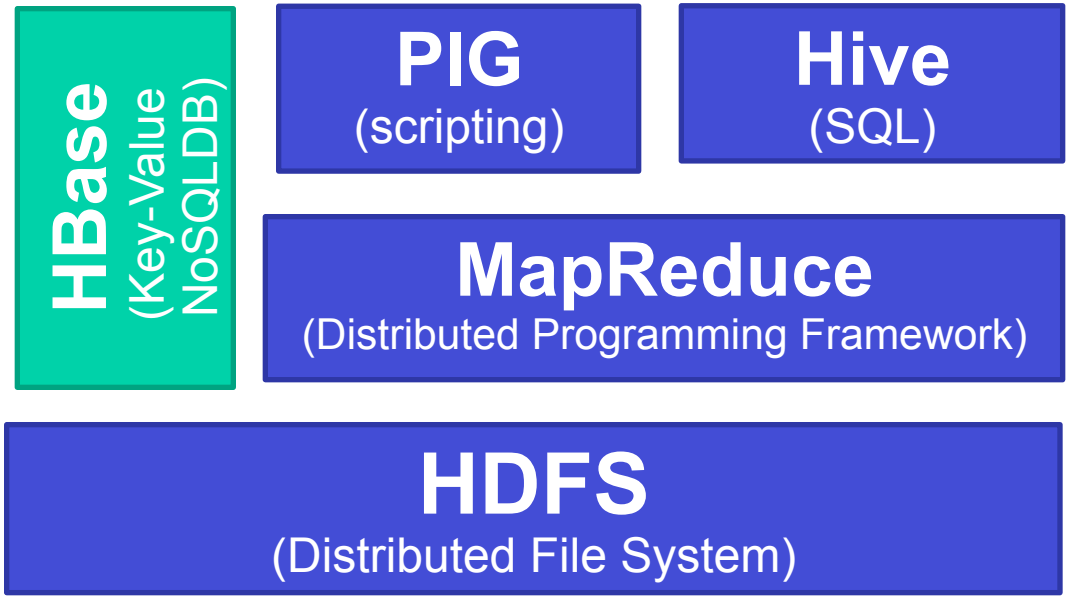
- MapReduce: Reads and Process data
- Code is executed on the DATA NODES inside HDFS

Inside Hadoop



- Hive = executes SQL and translates into MapReduce job
- PIG = scripting language, translates into MapReduce job

Inside Hadoop



- Hbase = Key/Value store, No-SQL

Hive

- SQL level for Hadoop
- ***Translates SQL to Map-Reduce jobs***
- Schema on Read – does not check the data on load

Hive Example

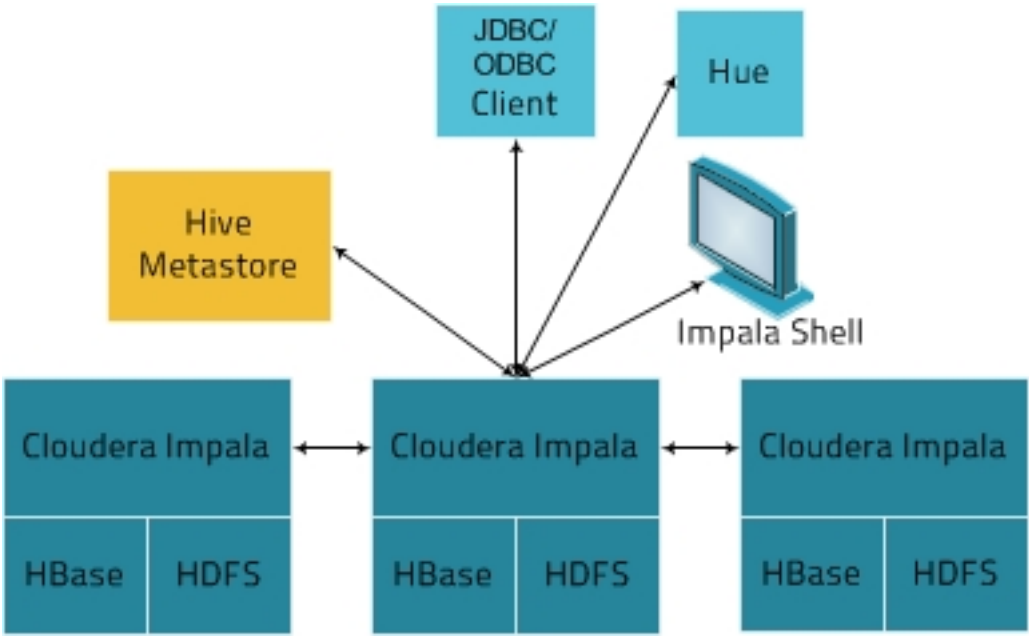
```
hive> create table lineitem (
l_orderkey int, l_partkey int, l_suppkey int,
l_linenumber int, l_quantity double,
l_extendedprice double, l_discount double, l_tax
double, l_returnflag string,
l_linestatus string, l_shipdate string,
l_commitdate string, l_receiptdate string,
l_shipinstruct string, l_shipmode string,
l_comment string)
row format delimited fields terminated by '|';
```

Hive Example

```
hive> create external table lineitem (
l_orderkey int, l_partkey int, l_suppkey int,
l_linenumber int, l_quantity double,
l_extendedprice double, l_discount double, l_tax
double, l_returnflag string,
l_linestatus string, l_shipdate string,
l_commitdate string, l_receiptdate string,
l_shipinstruct string, l_shipmode string,
l_comment string)
row format delimited fields terminated by '|'
location '/ssb/lineorder/';
```

Impala: Faster SQL

Not based on Map-Reduce, directly get data from HDFS



http://www.cloudera.com/content/cloudera-content/cloudera-docs/Impala/latest/Installing-and-Using-Impala/Installing-and-Using-Impala.html

Hadoop vs MySQL

Hadoop vs. MySQL for BigData

- Indexes
- Partitioning
- Sharding

- Full table scan
- Partitioning
- Map/Reduce

Hadoop (vs. MySQL)

- No indexes
  - All processing is full scan
  - BUT: distributed and parallel
- No transactions
- High latency (usually)

Indexes for Big Data challenge

- Creating an index for Petabytes of data?
- Updating an index for Petabytes of data?
- Reading a terabyte index?
- Random read of Petabyte?

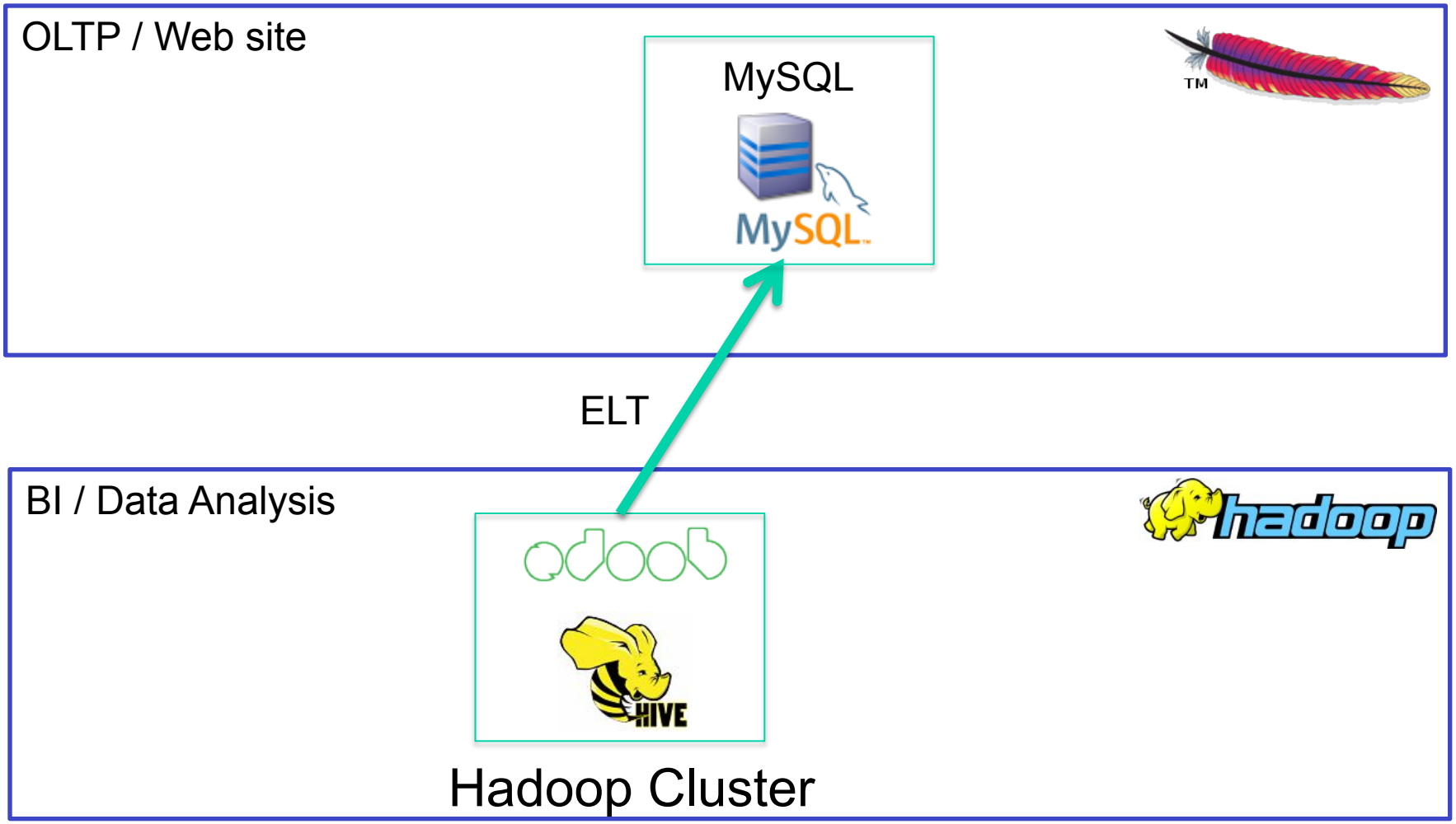
Full scan in parallel is better for big data

ETL vs ELT

1. **E**xtract data from external source
2. **T**ransform before loading
3. **L**oad data into MySQL

1. **E**xtract data from external source
2. **L**oad data into Hadoop
3. **T**ransform data/ Analyze data/ Visualize data;

Hadoop Example: Load Data I



Archiving to Hadoop

OLTP / Web site

Goal: keep 100G

MySQL

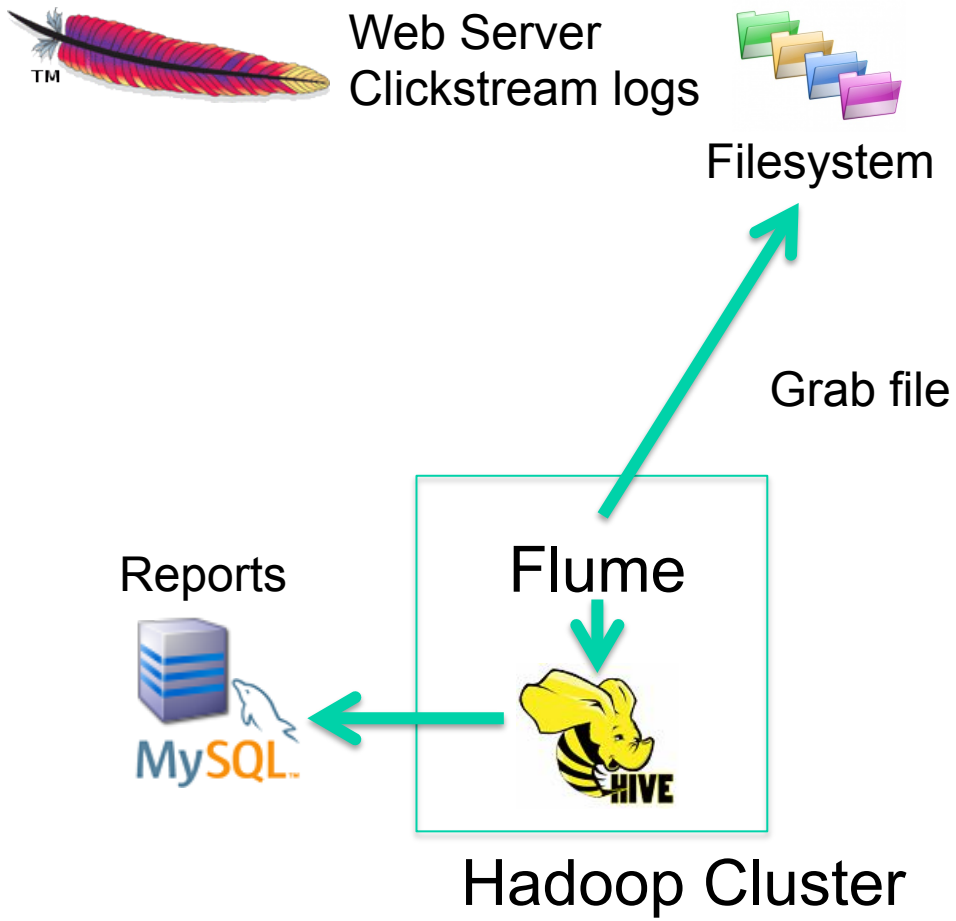
ELT

BI / Data Analysis

Hadoop Cluster

Can store Petabytes for archiving

Hadoop Example: Load Data II



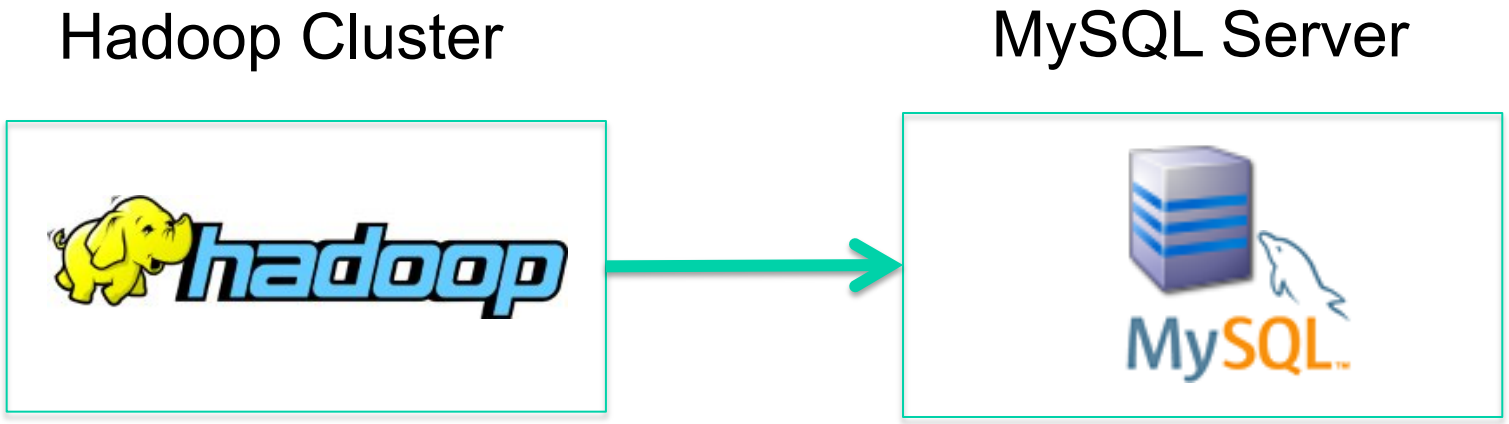
Hadoop and MySQL

- Apache Sqoop: http://sqoop.apache.org/
- Used to import and export data between Hadoop and MySQL

Hadoop and MySQL Together

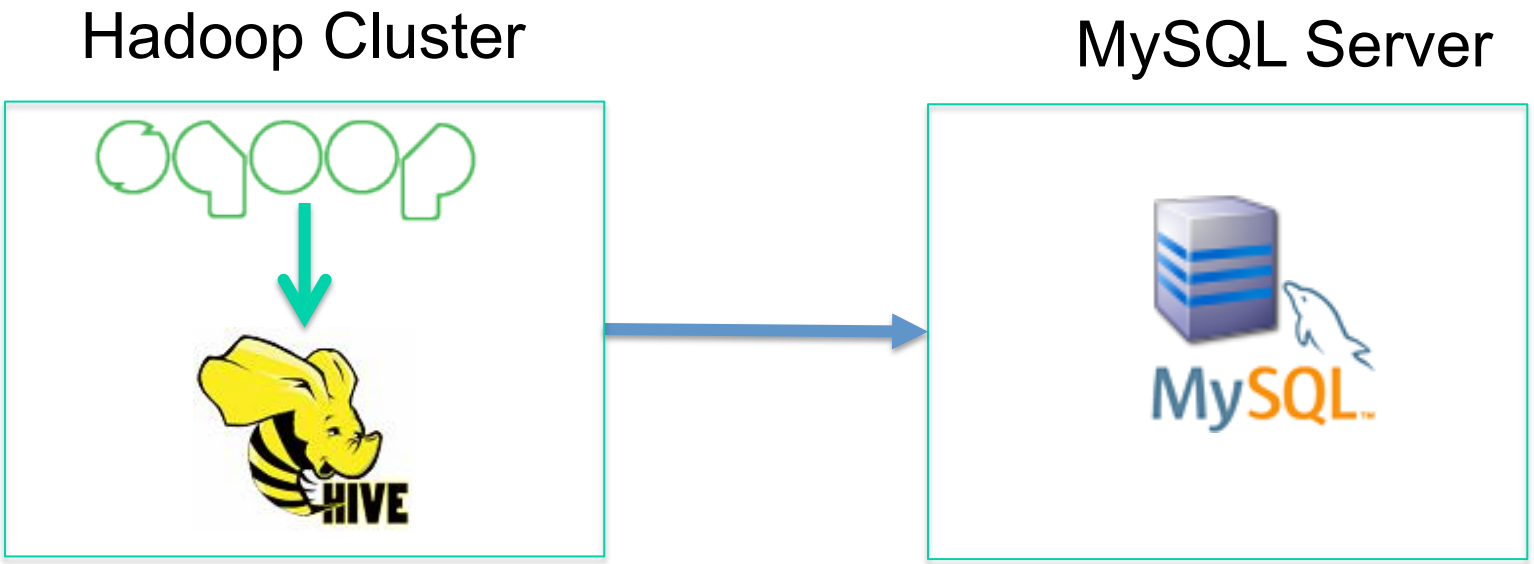
Integrating MySQL and Hadoop

Integration: Hadoop -> MySQL



Hadoop -> MySQL: Sqoop

```
$ sqoop export
--connect jdbc:mysql://mysql_host/db_name
--table orders
--export-dir /results/
```



Hadoop -> MySQL: Sqoop

Not real-time: run from a cronjob

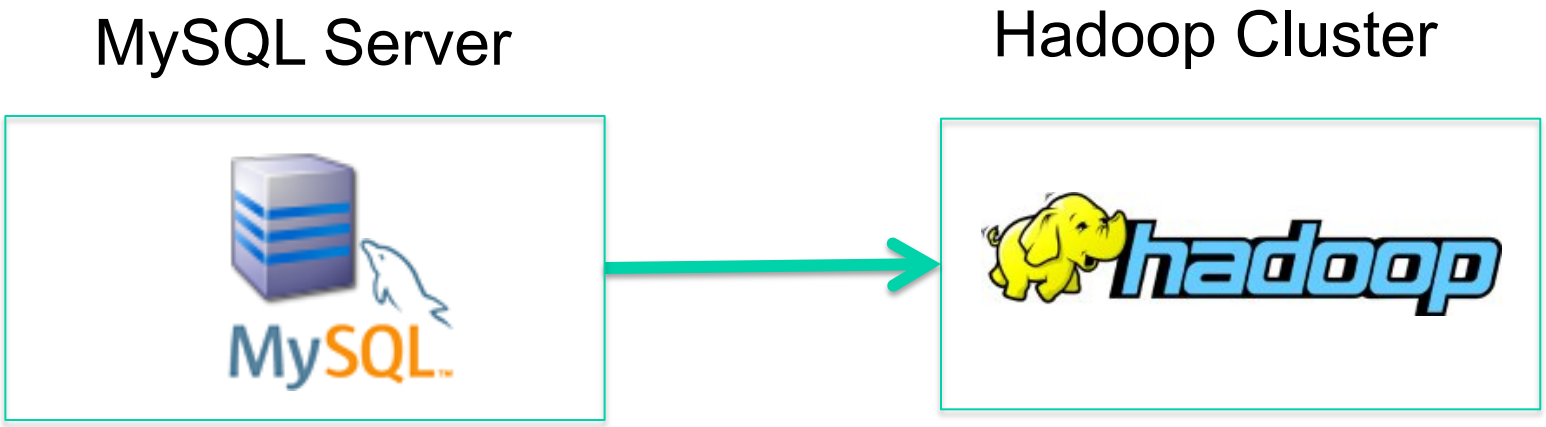
- Export from files on HDFS to MySQL
- To export *from HIVE*

  Hive tables = files inside HDFS

```
$ sqoop export
--connect jdbc:mysql://mysql_host/db_name
--table orders
--export-dir /user/hive/warehouse/orders.db
--input-fields-terminated-by '\001'
```

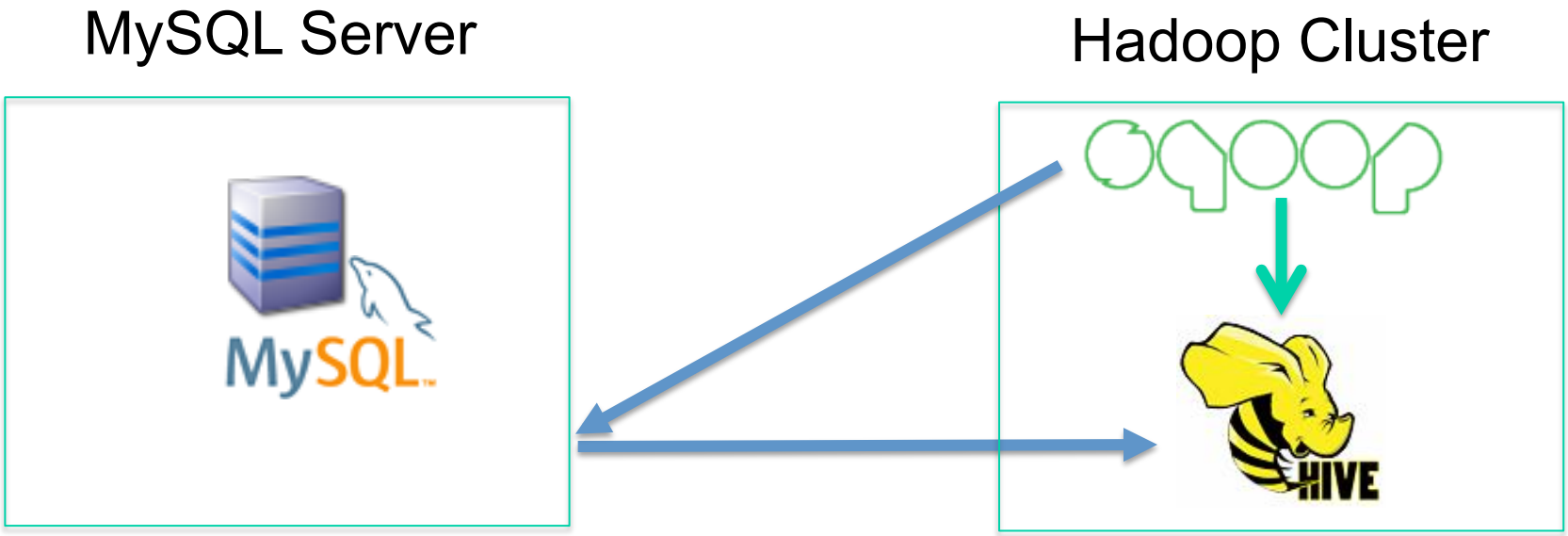
  - Hive uses ^A (\001) as a field delimiter

Integration: MySQL -> Hadoop



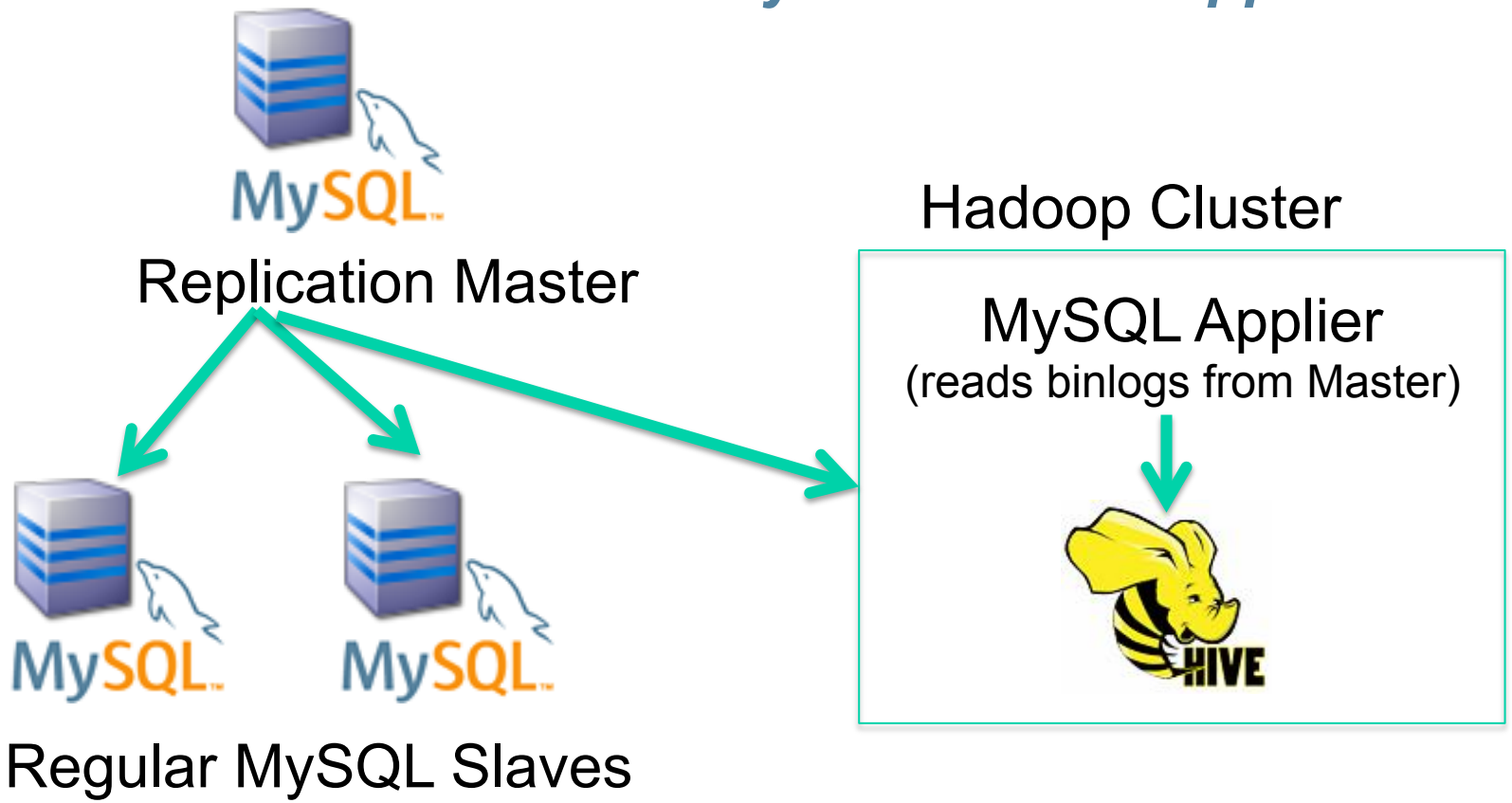
MySQL -> Hadoop: Sqoop

```
$ sqoop import
--connect jdbc:mysql://mysql_host/db_name
--table ORDERS
--hive-import
```



MySQL to Hadoop: Hadoop Applier

Only inserts are supported



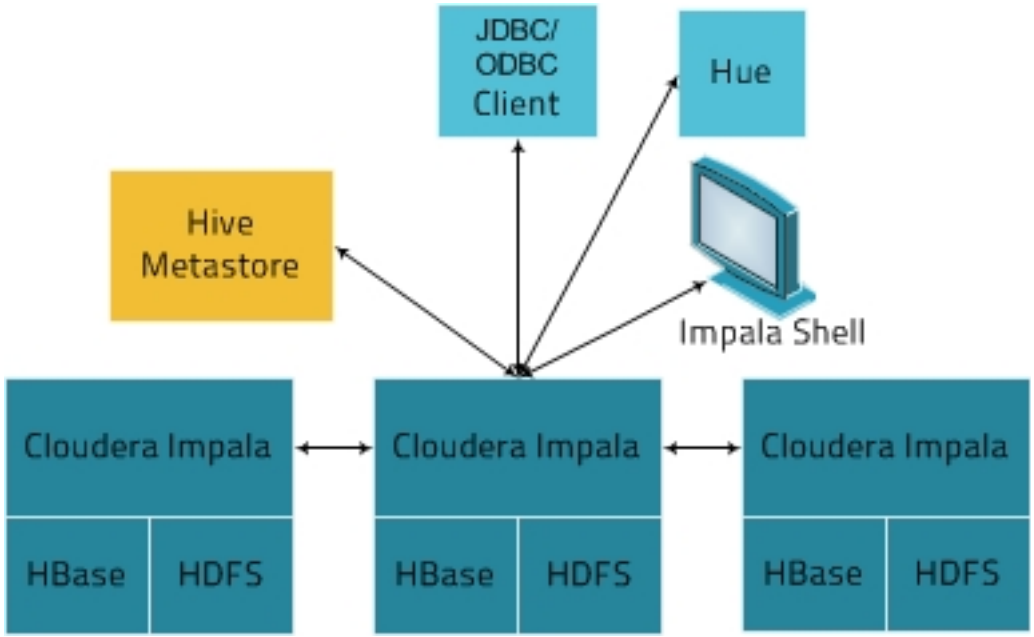
MySQL to Hadoop: Hadoop Applier

- Download from: http://labs.mysql.com/
- Alpha version right now
- We need to write code how to process data

Star Schema benchmark

Variation of TPC-H with star-schema database design
http://www.percona.com/docs/wiki/benchmark:ssb:start

Cloudera Impala



Impala and Columnar Storage

Parquet: Columnar Storage for Hadoop

- Column-oriented storage
- Supports compression
- Star Schema Benchmark example
  - RAW data: 990GB
  - Parquet data: 240GB

http://parquet.io/

http://www.cloudera.com/content/cloudera-content/cloudera-docs/Impala/latest/Installing-and-Using-Impala/ciiu_parquet.html

Impala Benchmark: Table

1. Load data into HDFS

```
$ hdfs dfs –put /data/ssb/lineorder* /ssb/lineorder/
```

http://www.percona.com/docs/wiki/benchmark:ssb:start

Impala Benchmark: Table

2. Create external table

```
CREATE EXTERNAL TABLE lineorder_text
(
    LO_OrderKey bigint ,
    LO_LineNumber tinyint ,
    LO_CustKey int,
    LO_PartKey int,
    ...
) ROW FORMAT DELIMITED FIELDS TERMINATED BY '|' LINES TERMINATED
BY '\n' STORED AS TEXTFILE LOCATION '/ssb/lineorder';
```

http://www.percona.com/docs/wiki/benchmark:ssb:start

Impala Benchmark: Table

3. Convert to Parquet

```
Impala-shell> create table lineorder
like lineorder_text
STORED AS PARQUETFILE;
Impala-shell> insert into lineorder
select * from lineorder_text;
```

http://www.percona.com/docs/wiki/benchmark:ssb:start

Impala Benchmark: SQLs

Q1.1

```
select
    sum(lo_extendedprice*lo_discount) as revenue
from
    lineorder, dates
where
    lo_orderdate = d_datekey
    and d_year = 1993
    and lo_discount between 1 and 3
    and lo_quantity < 25;
```

http://www.percona.com/docs/wiki/benchmark:ssb:start

Impala Benchmark: SQLs

Q3.1

```
select  c_nation, s_nation, d_year, sum(lo_revenue) as revenue
from
    customer, lineorder, supplier, dates
where
    lo_custkey = c_custkey and lo_suppkey = s_suppkey
    and lo_orderdate = d_datekey
    and c_region = 'ASIA' and s_region = 'ASIA'
    and d_year >= 1992 and d_year <= 1997
group by c_nation, s_nation, d_year
order by d_year asc, revenue desc;
```

Partitioning for Impala / Hive

Partitioning by Year

```
CREATE TABLE lineorder
(
    LO_OrderKey bigint ,
    LO_LineNumber tinyint ,
    LO_CustKey int,
    LO_PartKey int,
    ...
) partitioned by (year int);
```

Will create a "virtual" field on "year"

```
select * from lineorder where … year = '2013'
```

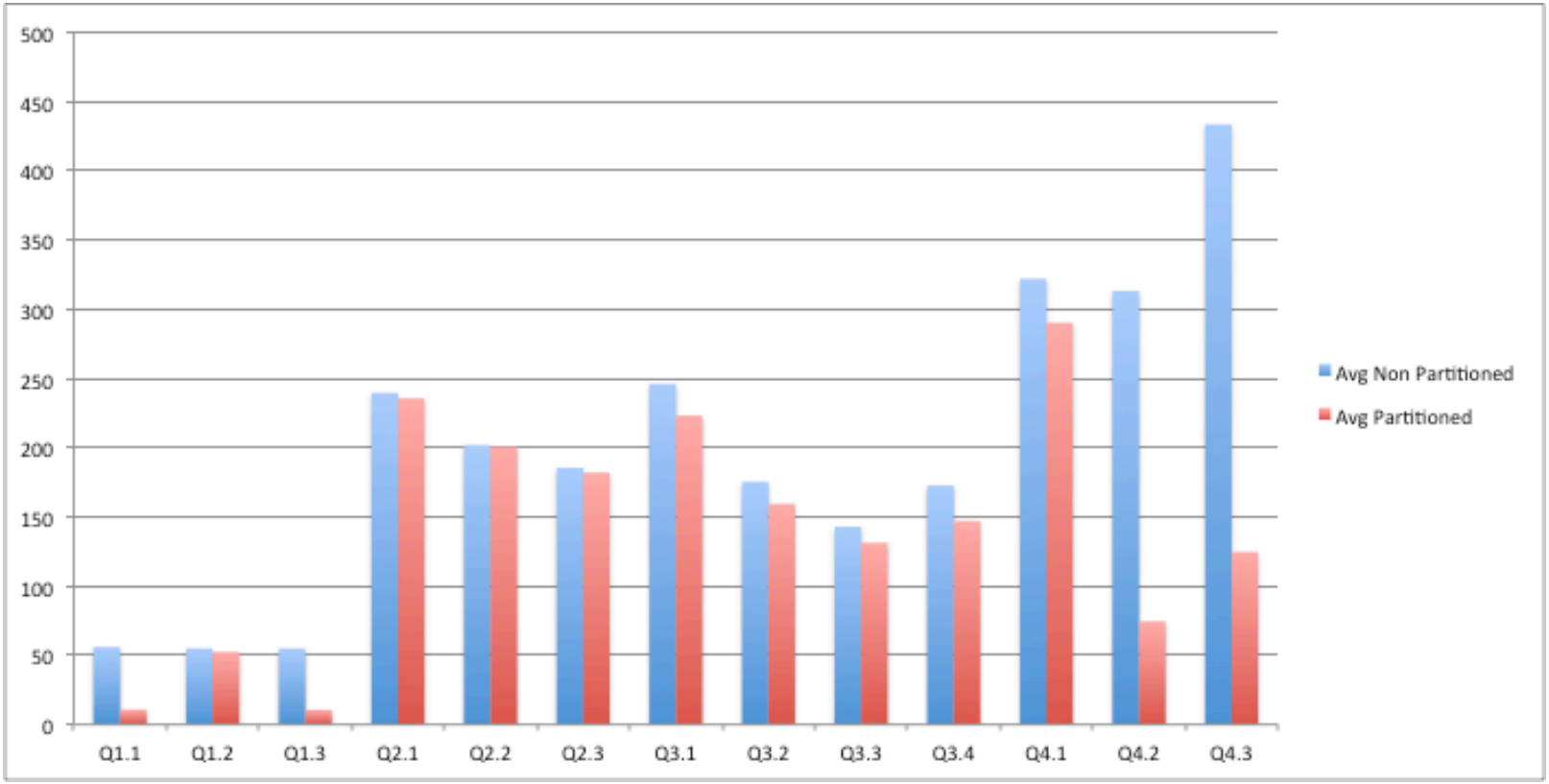
… will only read the "2013" partition (much less disk IOs!)

Impala Benchmark

CDP Hadoop with Impala

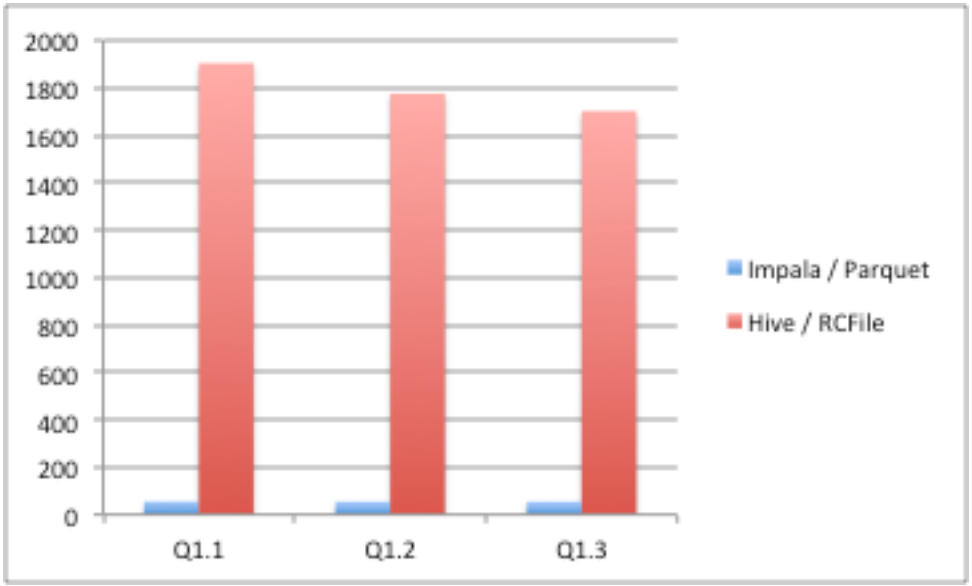
- 10 m1.xlarge Amazon EC2
- RAW data: 990GB
- Parquet data: 240GB

Impala Benchmark



Scanning 1Tb of row data 50-200 seconds
10 xlarge nodes on Amazon EC2

Impala vs Hive Benchmark



10 xlarge nodes on Amazon EC2
Impala and Hive

Amazon Elastic Map Reduce

Hive on Amazon S3

```
hive> create external table lineitem (
l_orderkey int, l_partkey int, l_suppkey int,
l_linenumber int, l_quantity double,
l_extendedprice double, l_discount double, l_tax
double, l_returnflag string,
l_linestatus string, l_shipdate string,
l_commitdate string, l_receiptdate string,
l_shipinstruct string, l_shipmode string,
l_comment string)
row format delimited fields terminated by '|'
location 's3n://data.s3ndemo.hive/tpch/lineitem';
```

Amazon Elastic Map Reduce

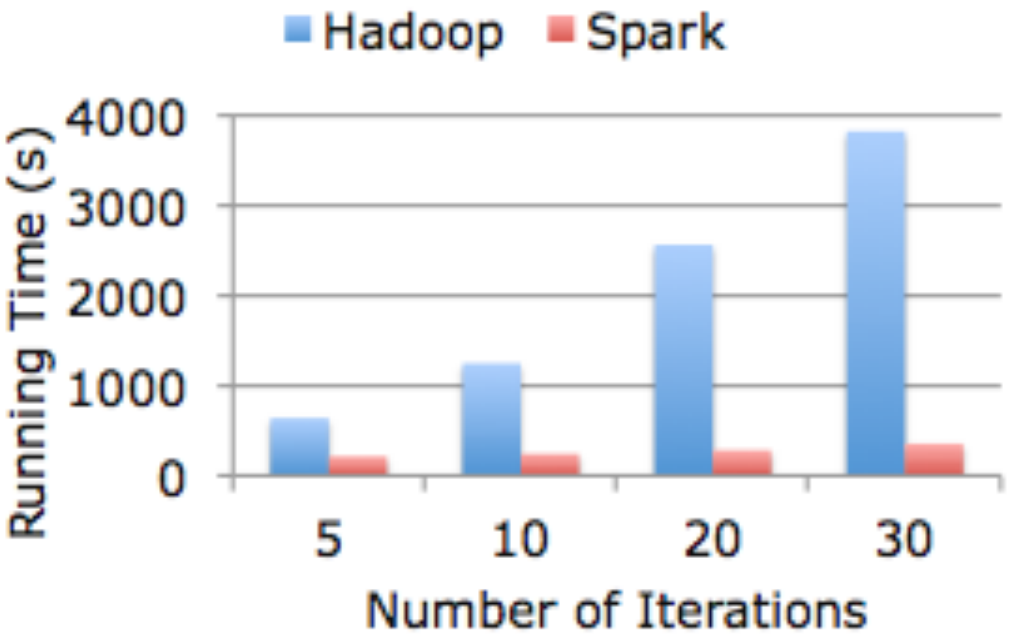
- Store data on S3
- Prepare SQL file (create table, select, etc)
- Run Elastic Map Reduce
  - Will start N boxes then stop them
- Results loaded to S3

Amazon Elastic Map Reduce

DEMO

Spark: Faster Hadoop, beta

Faster, utilize memory for processing data



http://spark.incubator.apache.org/

Questions?

Thank you!

Blog: http://www.arubin.org