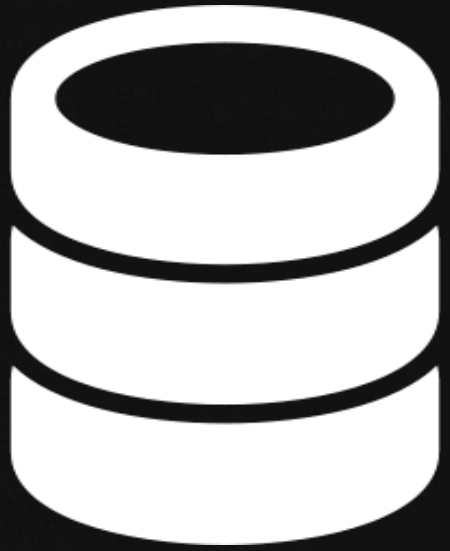


Shark:

SQL and Rich Analytics at Scale

http://bit.ly/cmu-db-fall2013

@andy_pavlo

SQL on MR

- Writing Java code to analyze data sets is akin to CODASYL.
- Alternatives:
  - *SQL→MapReduce*
  - *Pig→MapReduce*

Why is SQL on MR slow?

- Intermediate output on disk.
- Inferior data format and layout
- Naïve execution strategies.
- Task scheduling overhead.

Shark Overview

- Data warehouse on top of Spark.
- Scalable + fault-tolerant.
- Low-latency, interactive queries through in-memory computation.
- Compatible with Apache Hive.

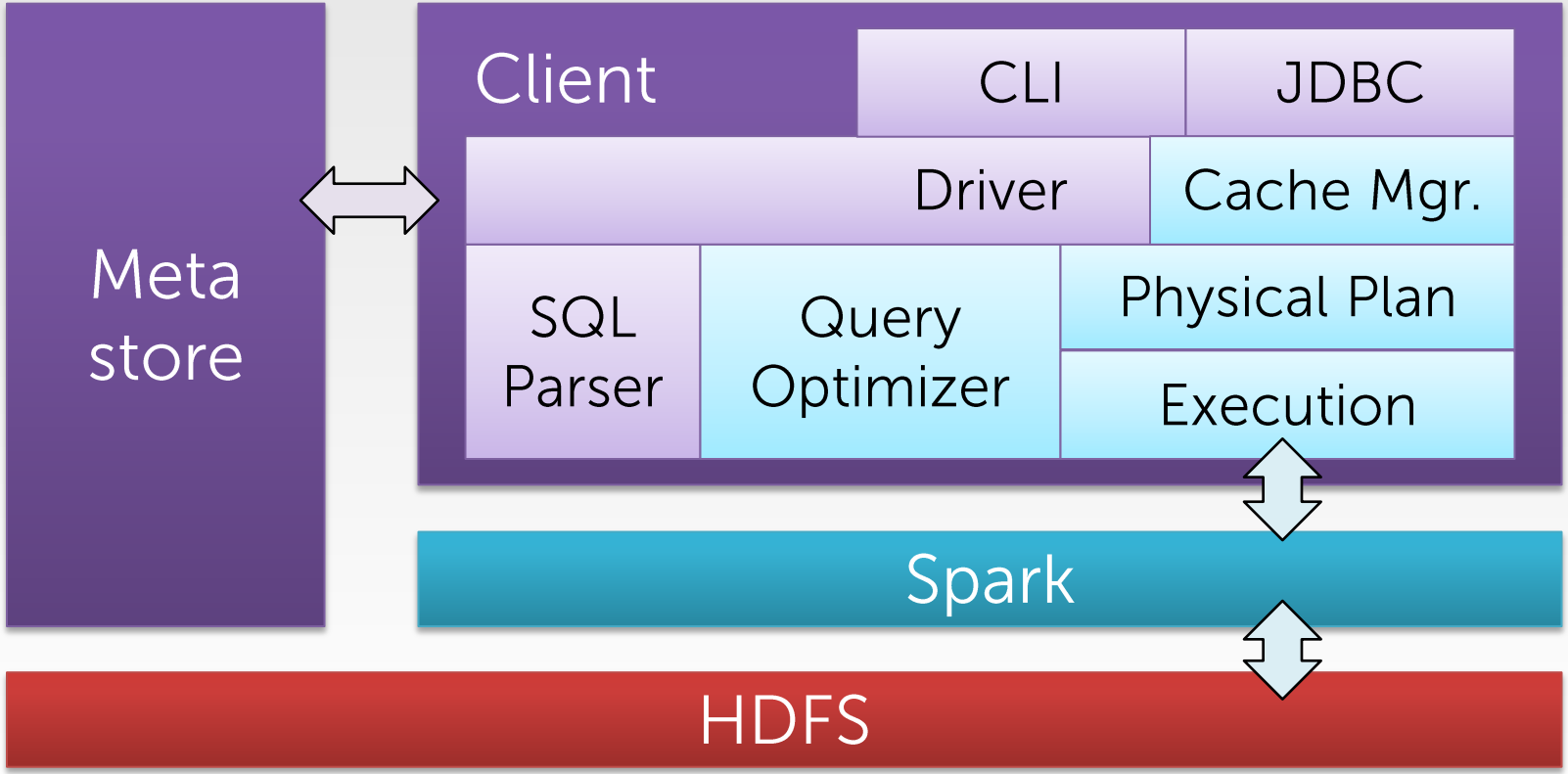
Spark Overview

- General purpose distributed computing framework.
- Work with distributed collections as you would with local ones.
- Resilient distributed datasets.

RDDs

- Immutable collections of objects.
- Built through parallel transformations (e.g., map, filter).
- Automatically rebuilt on failure
- Controllable persistence

Shark Architecture



Shark Optimizations

- Dynamic Query Optimization.
- Columnar Memory Store.
- Co-partitioning & Co-location.
- Machine Learning Integration.

Query Optimization

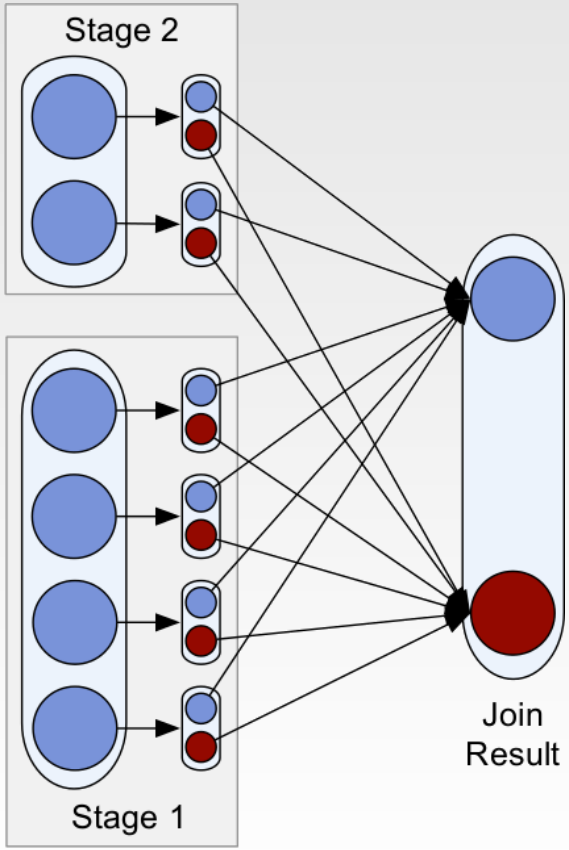
- No stats for non-loaded data.
- UDFs are difficult to optimize.

```
SELECT * FROM table1 AS t1
  JOIN table2 AS t2 ON t1.key = t2.key
WHERE assclownUDF(t1.field, t2.field)=true;
```

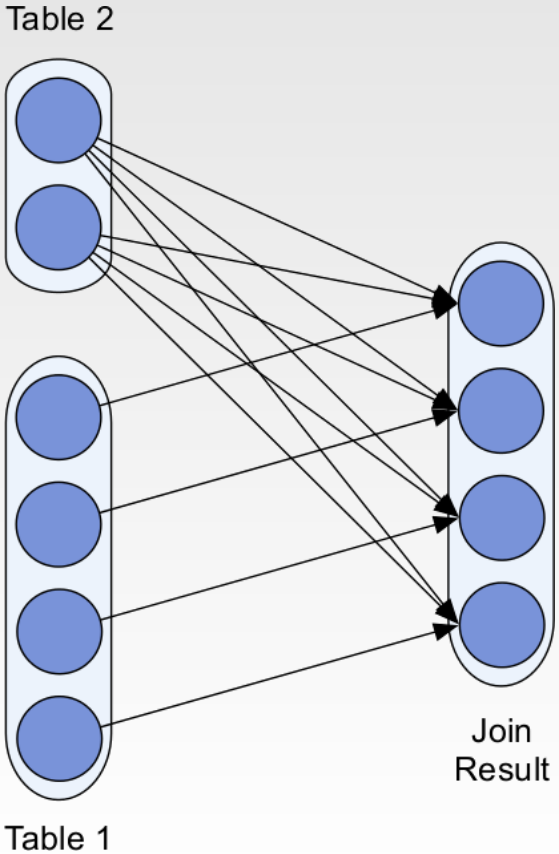
Partial DAG Execution

- Gather statistics per partition when materializing outputs.
- Allows for dynamic alternation of query plans based on statistics collected at runtime.

Shuffle Join



Map Join



Column Store

- Column-oriented storage using arrays of primitive types.
- Per-partition compression.

Row Storage

1	john	4.1
2	mike	3.5
3	sally	6.4

Column Storage

1	2	3
john	mike	sally
4.1	3.5	6.4

Data Co-Partitioning

- Store table partitions together in storage layer using pre-defined join attribute.

```
CREATE TABLE ol
AS SELECT * FROM lineitem
DISTRIBUTE BY L_ORDERKEY;
```

```
CREATE TABLE o TBLPROPERTIES
("copartition"="ol")
AS SELECT * FROM order
DISTRIBUTE BY O_ORDERKEY;
```

Machine Learning

- Unified system for query processing & machine learning.
- Query processing & ML share the same set of workers and caches.

Machine Learning

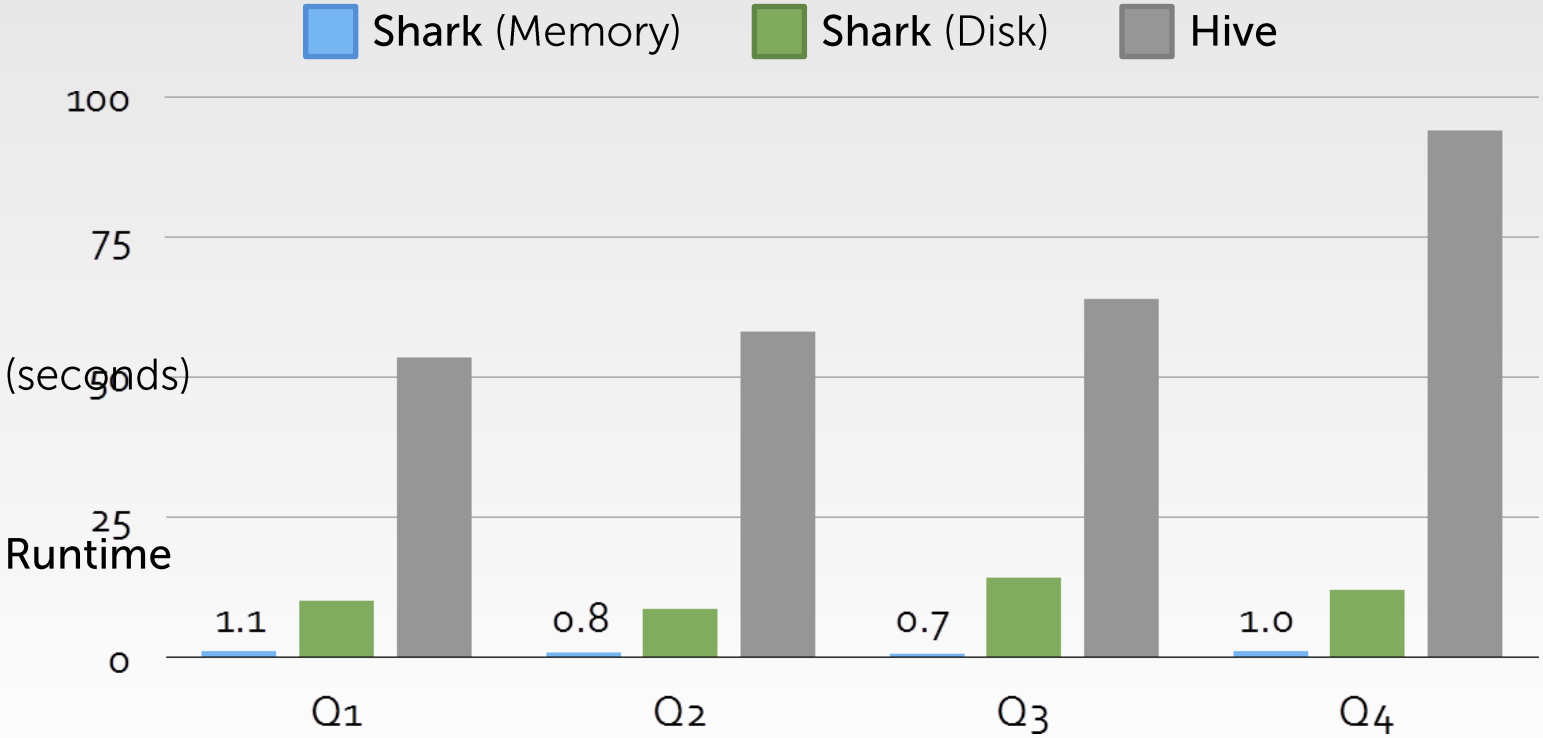
```
users = sql2rdd("SELECT * FROM user u
                 JOIN comment c ON c.uid=u.uid")

features = users.mapRows { row =>
           new Vector(extractFeat1(row.getInt("age")),
                      extractFeat2(row.getStr("name")),
                      ...)}
trainedVector = logRegressUDF(features.cache())
```

Experimental Analysis

- Comparison between Shark and Hive using HDFS.
- Data set from Conviva.
  - *100 EC2 nodes.*
  - *1.7 TB of video viewing sessions.*
  - *Four different queries.*

Hive Comparison



Conclusion

- Shark is a scalable, fault-tolerant data warehouse.
- Relies on Spark for distributed execution, but also provides “big data” optimizations.

Sources

- Reynold Xin (Berkeley)