Shark

Hive SQL on Spark

Michael Armbrust

```
Stage 0: Map-Shuffle-Reduce

Mapper(row) {
  fields = row.split("\t")
  emit(fields[0], fields[1]);
}

Reducer(key, values) {
  sum = 0;
  for (value in values) {
    sum += value;
  }
  emit(key, sum);
}
```

```
Stage 1: Map-Shuffle

Mapper(row) {
  ...
  emit(page_views, page_name);
}

... shuffle

Stage 2: Local

data = open("stage1.out")
for (i in 0 to 10) {
  print(data.getNext())
}
```

```
SELECT page_name, SUM(page_views) views
FROM wikistats GROUP BY page_name
ORDER BY views DESC LIMIT 10;
```

```
Stage 0: Map-Shuffle-Reduce

Mapper(row) {
  fields = row.split("\t")
  emit(fields[0], fields[1]);
}

Reducer(key, values) {
  page_views = 0;
  for (page_views in values) {
    sum += value;
  }
  emit(key, sum);
}
```

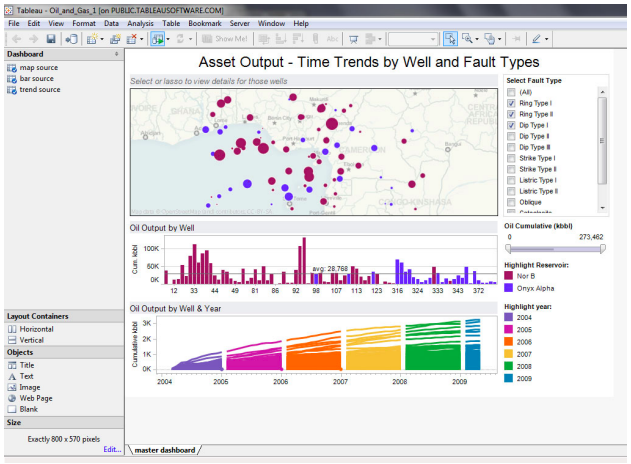
```
Stage 1: Map-Shuffle

Mapper(row) {
  ...
  emit(page_views, page_name);
}

... shuffle sorts the data

Stage 2: Local

data = open("stage1.out")
for (i in 0 to 10) {
  print(data.getNext())
}
```

Why SQL?

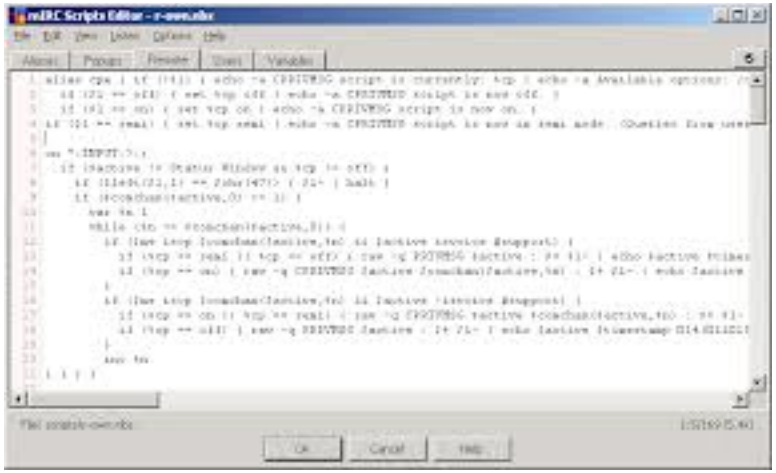
Easy to write queries that filter, join, aggregate data.

Provides *data independence*.

Huge ecosystem of SQL tools.


Asset Output - Time Trends by Well and Fault Types
Select or lasso to view details for those wells
Oil Output by Well
Oil Output by Well & Year
Dashboard
map source
bar source
trend source
Layout Containers
Horizontal
Vertical
Objects
Title
Text
Image
Web Page
Blank
Size
master dashboard


Multi-Channel Customer Analysis


mIRC Scripts Editor

Outline

Hive and Shark

Usage

Under the hood

Apache Hive

Puts structure/schema onto HDFS data

Compiles HiveQL queries into MapReduce jobs

Very popular: 90+% of Facebook Hadoop jobs generated by Hive

Initially developed by Facebook

OLAP vs OLTP

Hive is NOT for online transaction processing (OTLP)

Focuses on **scalability** and **extensibility** for data warehouses / online analytical processing (OLAP)

Scalability

Massive scale out and fault tolerance capabilities on commodity hardware

Can handle petabytes of data

Easy to provision (because of scale-out)

Extensibility

Data types: primitive types and complex types

User-defined functions

Scripts

Serializer/Deserializer: text, binary, JSON…

Storage: HDFS, Hbase, S3…

But slow…

Takes 20+ seconds even for simple queries

"A good day is when I can run 6 Hive queries”
- @mtraverso

Shark

Analytic query engine compatible with Hive
- » Supports Hive QL, UDFs, SerDes, scripts, types
- » A few esoteric features not yet supported

Makes Hive queries run much faster
- » Builds on top of Spark, a fast compute engine
- » Allows (optionally) caching data in a cluster's memory
- » Various other performance optimizations

Integrates with Spark for machine learning ops

Spark User Meetup

Home Members Sponsors Photos Pages Discussions More Group tools My profile

Spark

San Francisco, CA
Founded Jan 20, 2012

About us...

Spark Enthusiasts	1,135
Group reviews	18
Past Meetups	14
Our calendar	

We're about:

Spark · Big Data · Machine Learning · hadoop · Data Analytics · Open Source · Scala · Cloud Computing · Functional Programming · MapReduce · Hive · Artificial Intelligence

Organizers:

Copy this Meetup

Shark in Yahoo's Advertising Data Platforms

2 days ago · 6:30 PM
ClassRoom 4/5 Building C, Yahoo!

upload

This is use case talk from the folks at Yahoo!. In Yahoo!'s advertising and dat... see all

How was the Meetup?

avg:

Ask a question, share something, or leave a comment...

Notifications Post

Grega Kešpret

Tools

241 attended

Reynold Xin
CO-ORGANIZER
EVENT HOST

Ram Sriharsha
EVENT HOST
Good to see you

Henry Fang
Good to see you

Markus Anderle
Good to see you

Minh Do
Good to see you

Use cases

Interactive query & BI (e.g. Tableau)

Reduce reporting turn-around time

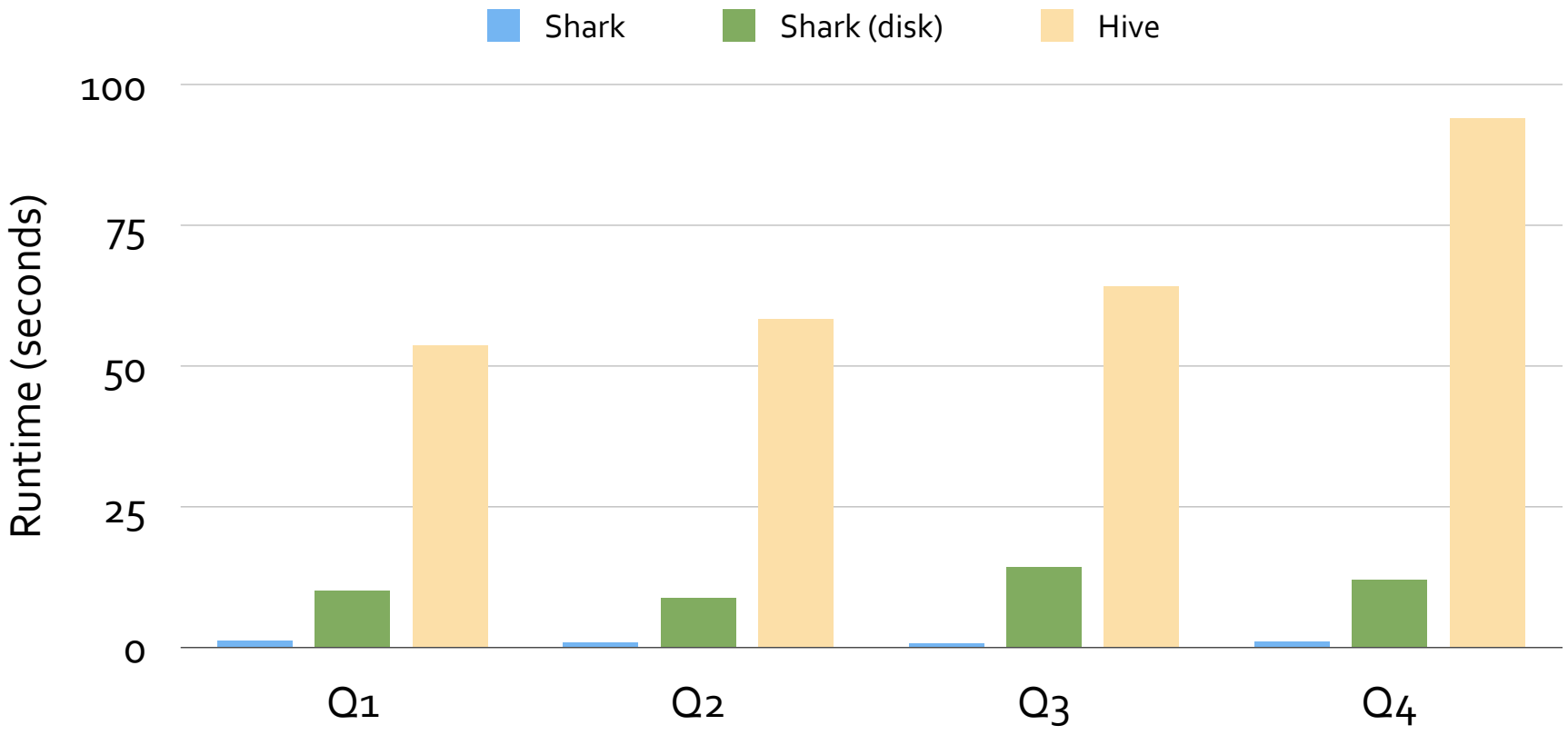
Integration of SQL and machine learning pipeline

Much faster?

100X faster with in-memory data

2 - 10X faster with on-disk data

Performance (1.7TB on 100 EC2 nodes)



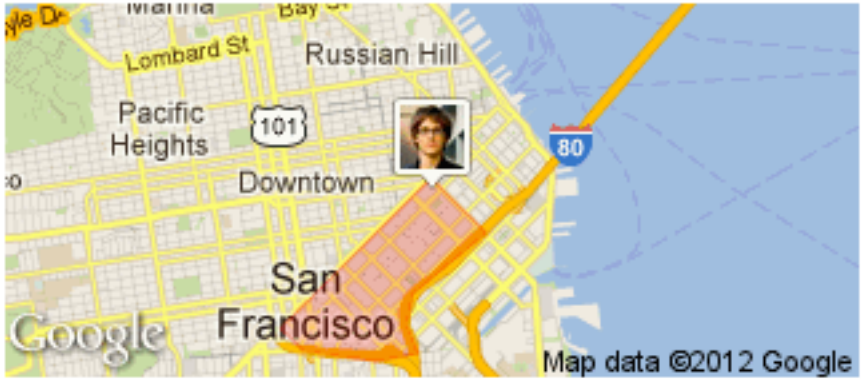
Jure Leskovec
@jure

Following

Median Hadoop job input data size at Microsoft, Yahoo and Facebook is only about 15gb!
research.microsoft.com/pubs/163083/ho...

Reply Retweeted Favorite Pocket

36 RETWEETS **23** FAVORITES



from SoMa
San Francisco, CA

4:33 PM - 9 Jul 12 via Twitter for iPhone · Embed this Tweet

Outline

Hive and Shark

Usage

Under the hood

Data Model

Tables: unit of data with the same schema

Partitions: e.g. range-partition tables by date

Data Types

Primitive types

- TINYINT, SMALLINT, INT, BIGINT
- BOOLEAN
- FLOAT, DOUBLE
- STRING
- TIMESTAMP

Complex types

- Structs: STRUCT {a INT; b INT}
- Arrays: ['a', 'b', 'c']
- Maps (key-value pairs): M['key']

Hive QL

Subset of SQL

- » Projection, selection
- » Group-by and aggregations
- » Sort by and order by
- » Joins
- » Sub-queries, unions

Hive-specific

- » Supports custom map/reduce scripts (TRANSFORM)
- » Hints for performance optimizations

Analyzing Data

```
CREATE EXTERNAL TABLE wiki
(id BIGINT, title STRING, last_modified STRING, xml
STRING, text STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
LOCATION 's3n://spark-data/wikipedia-sample/';
```

```
SELECT COUNT(*) FROM wiki WHERE TEXT LIKE '%Berkeley%';
```

Caching Data in Shark

```
CREATE TABLE mytable_cached AS SELECT * FROM
mytable WHERE count > 10;
```

Creates a table cached in a cluster's memory using RDD.cache ()

Unified table naming (in Shark 0.8.1):

```
CACHE mytable;  UNCACHE mytable;
```

Spark Integration

From Scala:

```
val points = sc.runSql[Double, Double](
  “select latitude, longitude from historic_tweets”)

val model = KMeans.train(points, 10)

sc.twitterStream(...)
  .map(t => (model.closestCenter(t.location), 1))
  .reduceByWindow(“5s”, _ + _)
```

Spark Integration

From SQL (in Shark 0.8.1):

```
GENERATE KMeans(tweet_locations) AS TABLE tweet_clusters
```

```
// Scala table generating function (TGF):
object KMeans {
  @Schema(spec = “x double, y double, cluster int”)
  def apply(points: RDD[(Double, Double)]) = {
    ...
  }
}
```

Tuning Degree of Parallelism

SET mapred.reduce.tasks=50;

Shark relies on Spark to infer the number of map tasks (automatically based on input size)

Number of “reduce” tasks needs to be specified

Out of memory error on slaves if too small

We are working on automating this!

Outline

Hive and Shark

Data Model

Under the hood

How?

A better execution engine

- » Hadoop MR is ill-suited for short running SQL

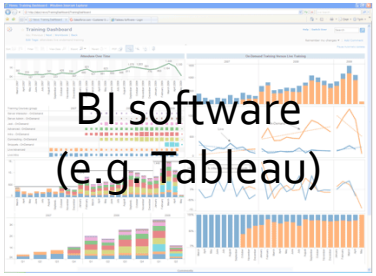
Optimized storage format

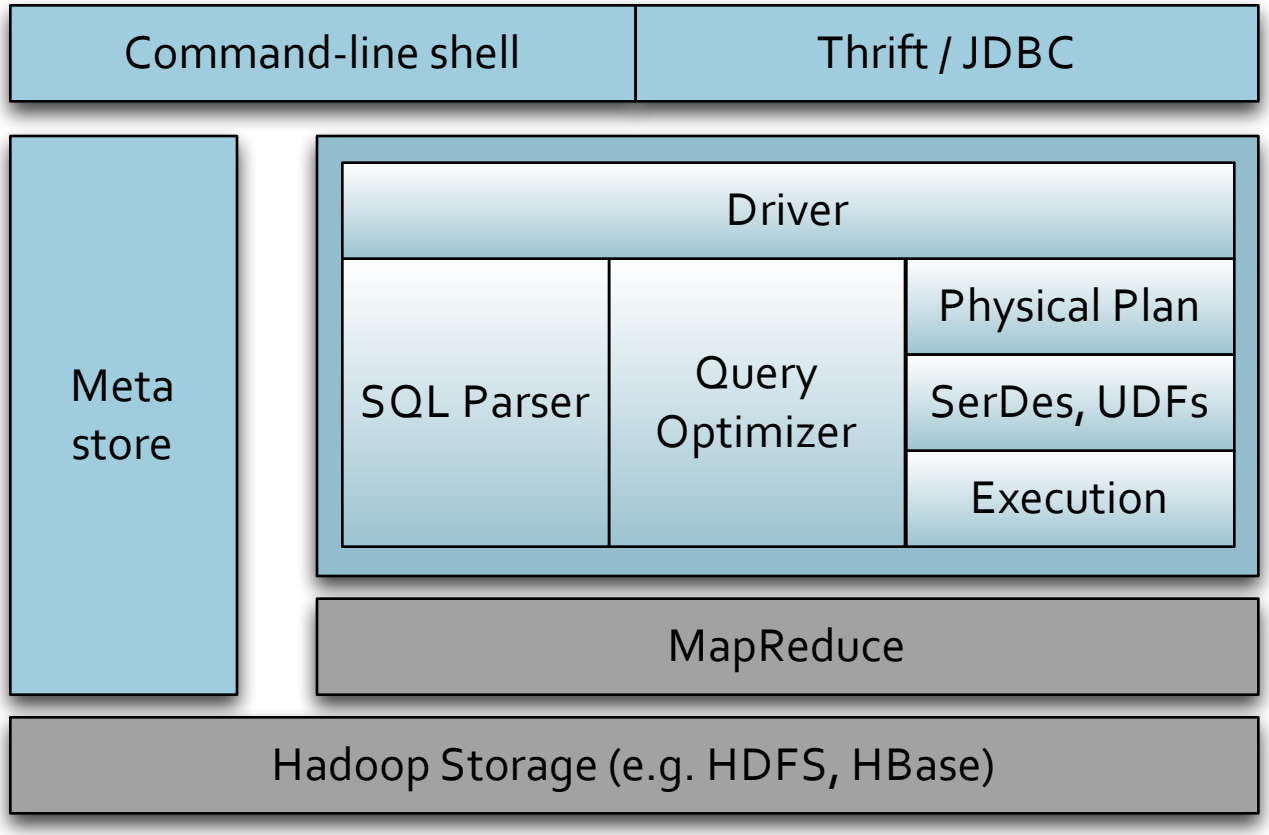
- » Columnar memory store

Various other optimizations

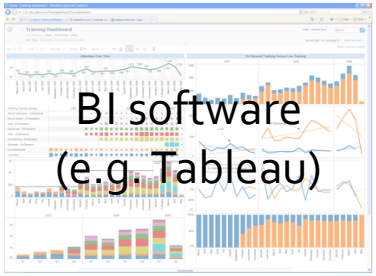
- » Fully distributed sort, data co-partitioning, partition pruning, etc

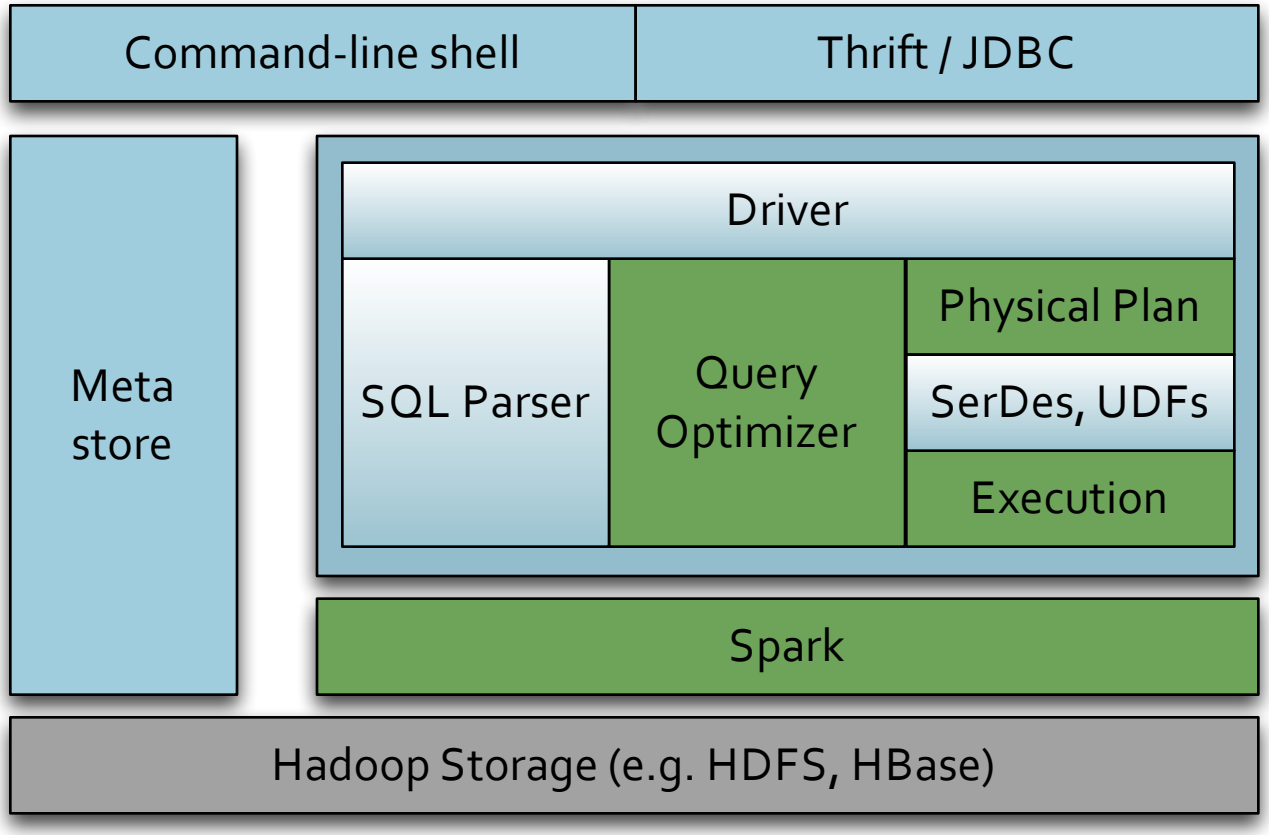
Hive Architecture





Shark Architecture





Why is Spark a better engine?

Extremely fast scheduling
- » ms in Spark vs secs in Hadoop MR

Support for general DAGs
- » Each query is a “job” rather than stages of jobs

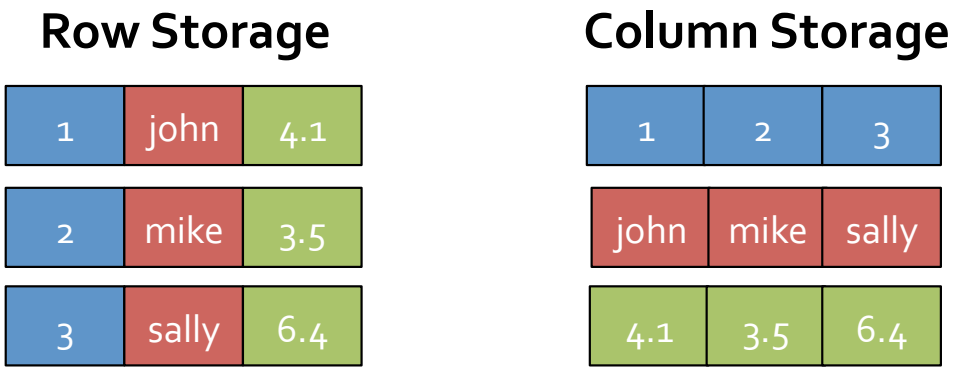
Many more useful primitives
- » Higher level APIs
- » Broadcast variables
- » …

Columnar Memory Store

Column-oriented storage for in-memory tables

Yahoo! contributed CPU-efficient compression (e.g. dictionary encoding, run-length encoding)

3 – 20X reduction in data size



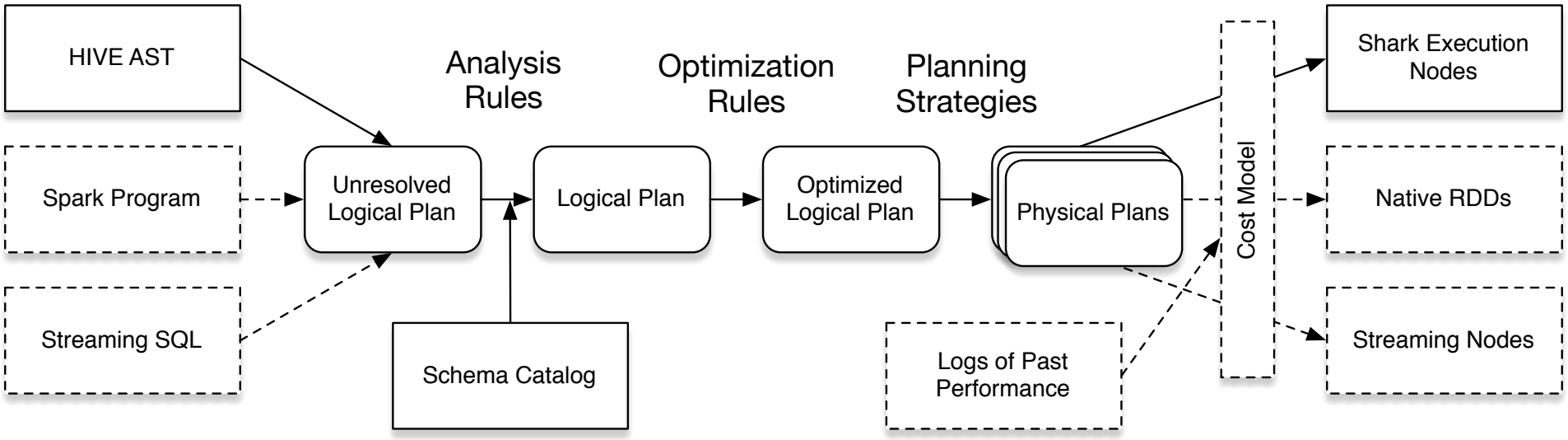
Ongoing Work

- Code generation for query plan (Intel)
- BlinkDB integration (UCB)
- Bloom-filter based pruning (Yahoo!)
- More intelligent optimizer

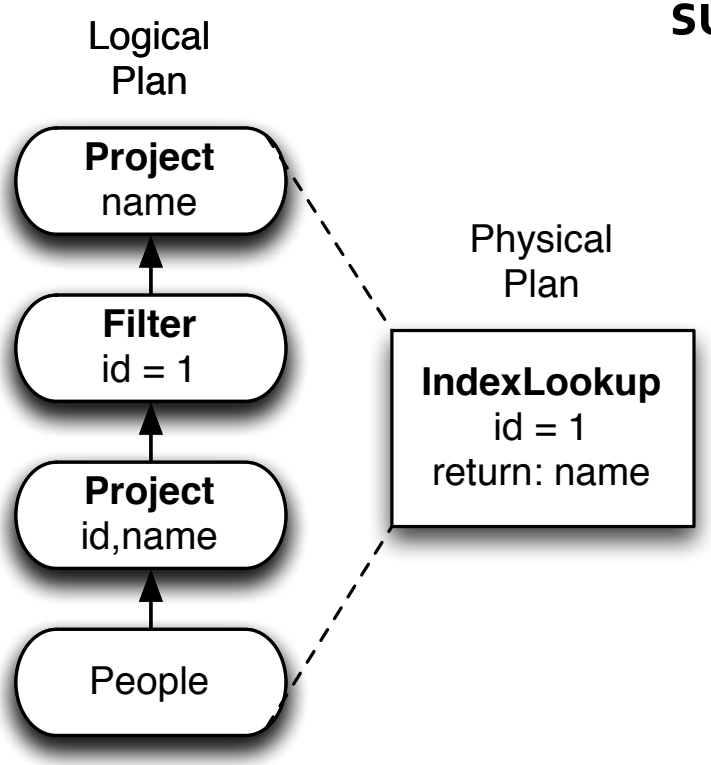
Catalyst

Framework for expressing query optimizations.

Should greatly improve Sharks ability to optimize queries



Optimized Execution

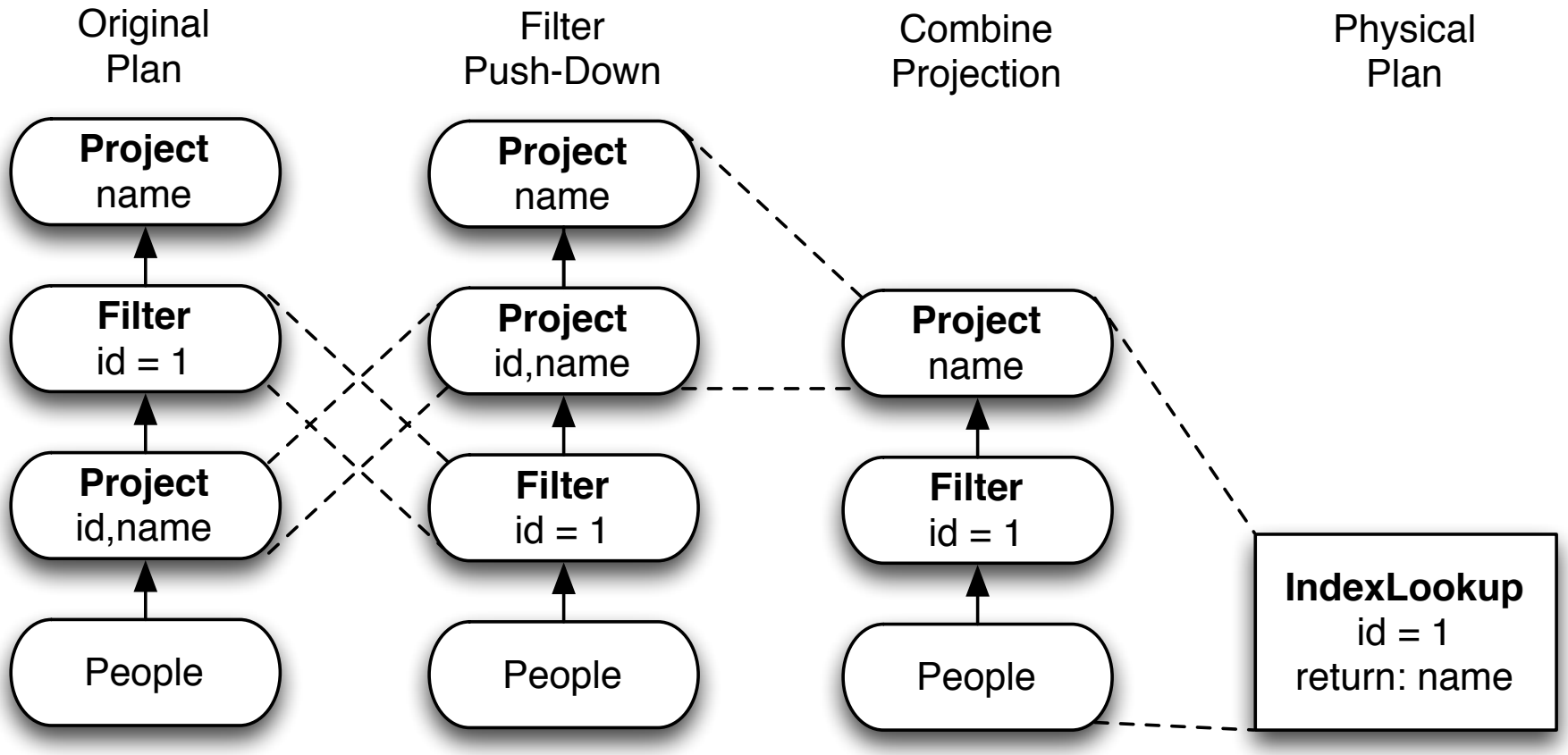


Writing imperative code to optimize such patterns generally is hard.

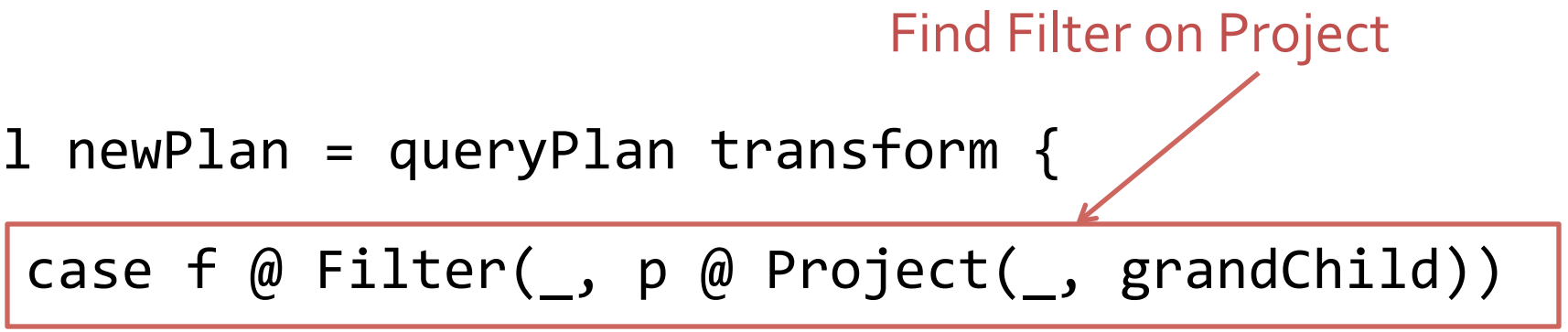
Instead write simple rules:

- Each rule makes one small change
- Run rules many rules together to fixed point.

Optimizing with Rules



Writing Rules as Tree Transformations



Writing Rules as Tree Transformations

```
val newPlan = queryPlan transform {
   case f @ Filter(_, p @ Project(_, grandChild))
     if(f.references subsetOf grandChild.output) =>
   p.copy(child = f.copy(child = grandChild)
}
```

Check that the filter can be evaluated without the result of the project.

Writing Rules as Tree Transformations

```
val newPlan = queryPlan transform {
   case f @ Filter(_, p @ Project(_, grandChild))
     if(f.references subsetOf grandChild.output) =>
   p.copy(child = f.copy(child = grandChild))
}
```

If so, switch the order.

Getting Started

~5 mins to install Shark locally

» https://github.com/amplab/shark/wiki

Spark EC2 AMI comes with Shark installed (in /root/shark)

Also supports Amazon Elastic MapReduce

Use Mesos or Spark standalone cluster for private cloud

Exercises @ Spark Summit

Each on-site audience gets a 4-node EC2 cluster preloaded with Wikipedia traffic statistics data

Live streaming audiences get an AMI preloaded with all software (Mesos, Spark, Shark)

Use Spark and Shark to analyze the data

More Information

Hive resources:

- https://cwiki.apache.org/confluence/display/Hive/GettingStarted
- http://hive.apache.org/docs/r0.10.0/api/

Shark resources:

- http://shark.cs.berkeley.edu
- https://github.com/amplab/shark