Spark as a Platform to Support Multi-Tenancy and Many Kinds of Data Applications

Kelvin Chu @ Uber

About Myself

- Started with Spark 0.7
- Co-created Spark Job Server at Ooyala
- Working at Uber since 2014 August

About Uber

- Found in 2010
- One Tap to Request a Ride
- Build Software Platform for Driver Partners and Riders

- 311 Cities
- 58 Countries
- Hundreds of thousands of driver partners
- Millions of riders
- 1+ million trips around the world everyday

Data Platform Team

- Second Engineer
- Part of Data Engineering
- Members with diverse background from Hadoop, HBase, Oozie, Spark, Voldemort, YARN, etc.

Data Lake

Sqoop on Spark for Data Ingestion

5:45pm Today

Room 3

Veena Basavaraj (Uber)

Vinoth Chandar (Uber)

Challenges

- Shared by Many Teams
  - Different technical background
  - Producers
  - Consumers
- Many Use Cases
- Different SLAs

Spark

YARN

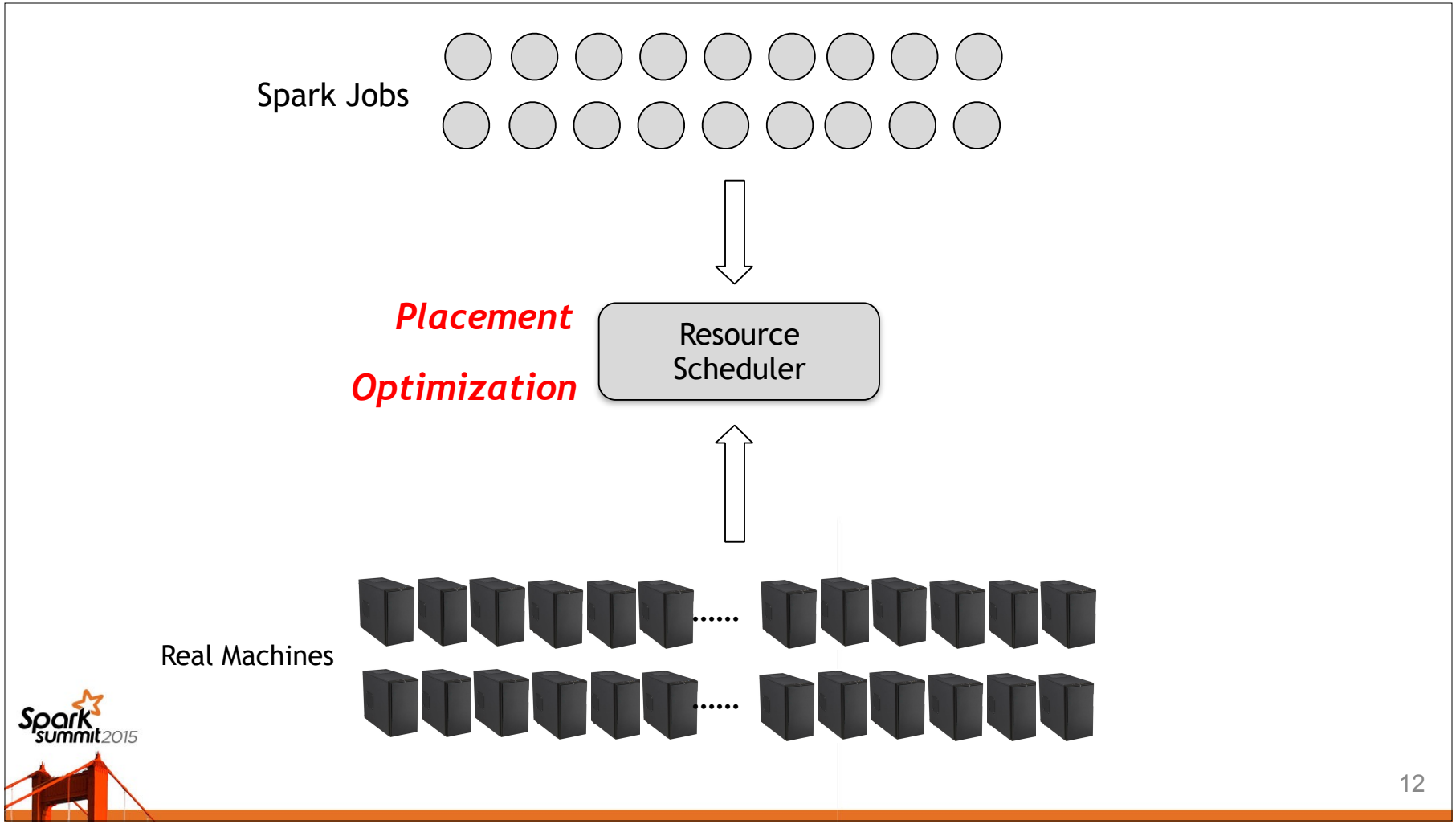
Parquet

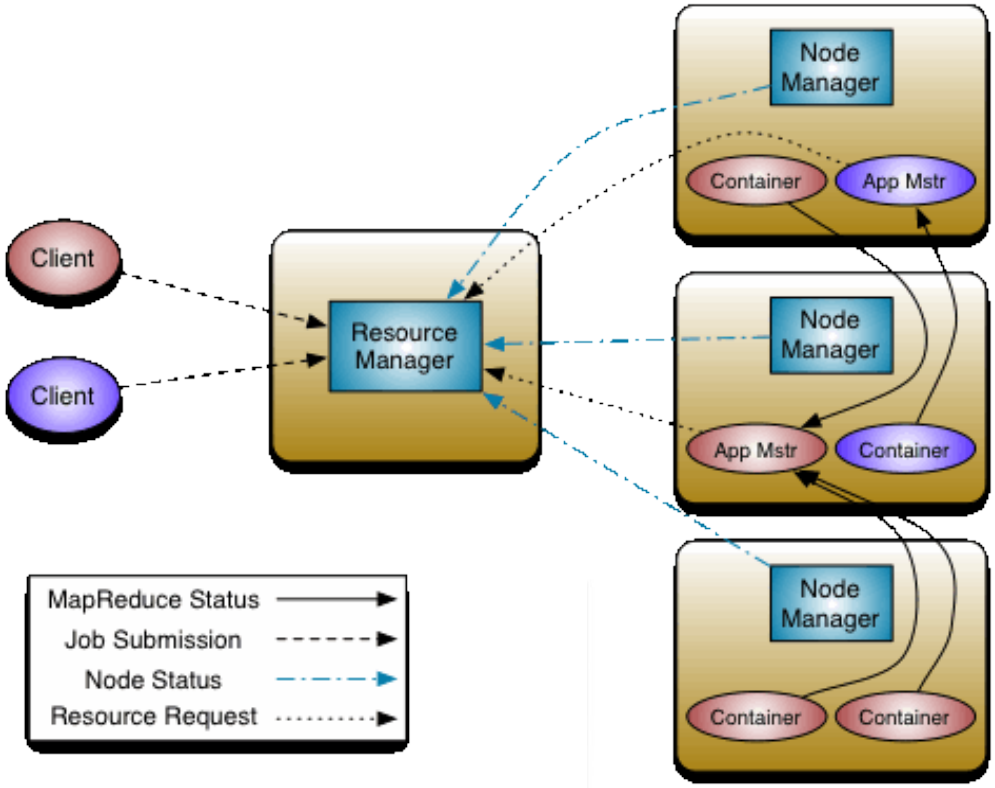
Why Spark?

- Easy to Use
- Ecosystem
  - Batch jobs
  - SparkSQL
  - MLlib

YARN

- Resource Management
  - Allocation
  - Teams/Jobs Isolation
  - Cluster Optimization
- Hadoop Kerberos Security

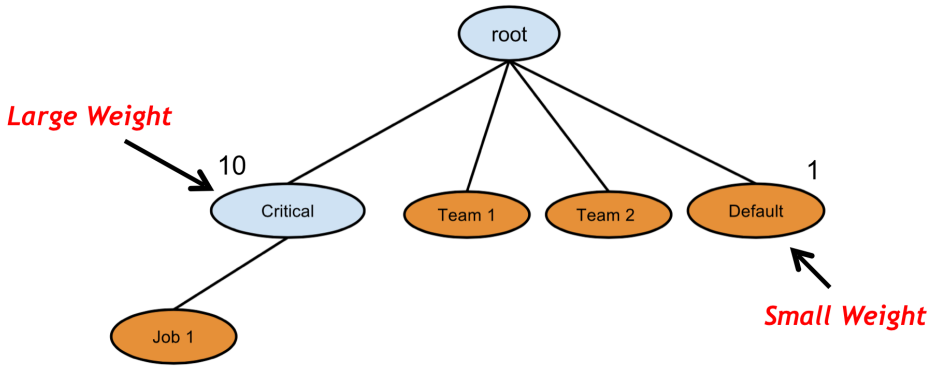

Spark Jobs
Placement
Optimization
Resource Scheduler
Real Machines



http://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html

Resource Queues

- Resource Isolation
- CPU & Memory
  - I/O in the future
- Hierarchical queues
- Priorities as Weights
- Allocate different teams and users to queues
- Queue placement policies



High Availability

- Cluster Mode
- Spark Context in Application Master
- Automatic Retry
  - Default: Once
- Executor failure handled by Spark

HA Tests Passed

- Kill active YARN Resource Manager
- Kill YARN Node Manager
- Kill the job Application Master
- Kill random Spark executors
- Kill YARN history server
- Kill Spark history server
- Results:
  - Existing spark jobs finished
  - New jobs can be submitted

SPARK-6751

use version 1.3+

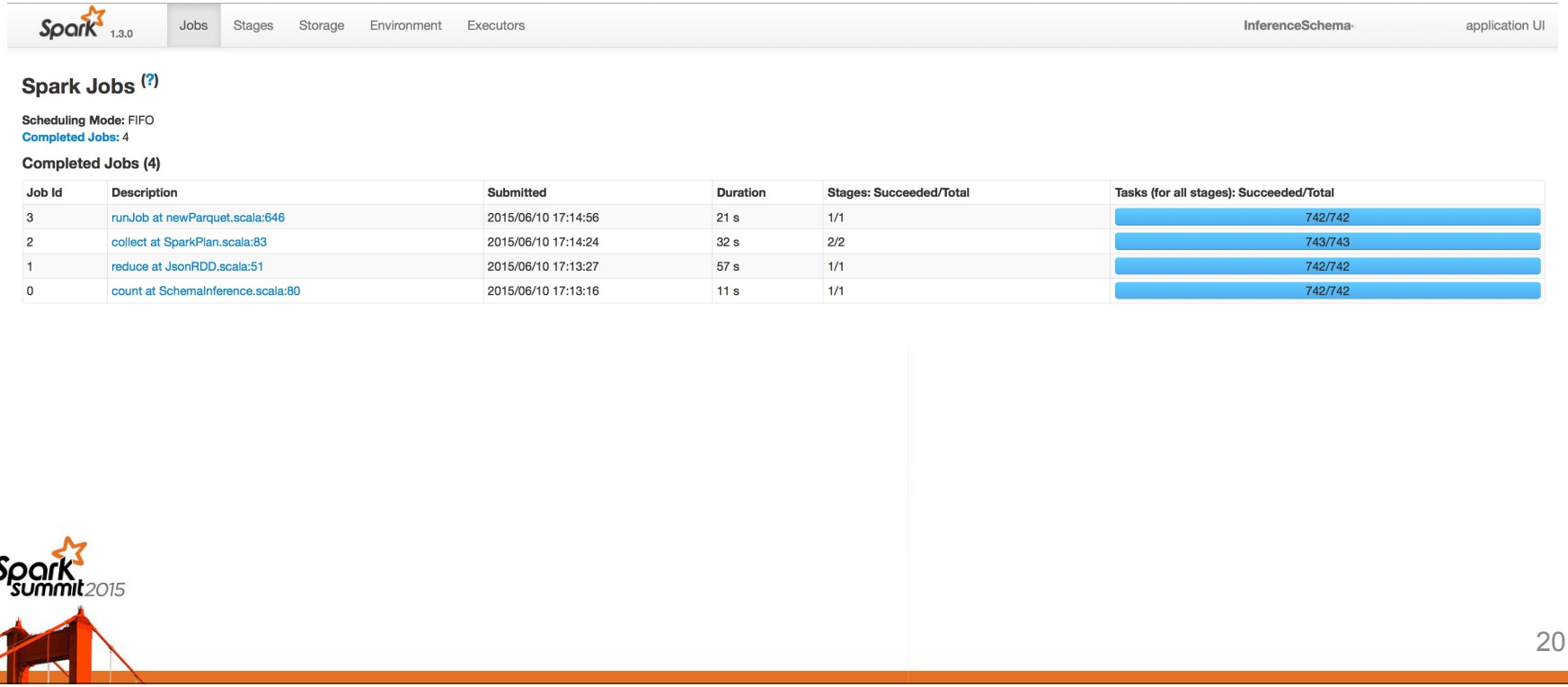
or set the flag

spark.eventLog.overwrite

Security

- Critical in Multi-Tenancy
- Only cluster manager
  - Hadoop Kerberos Security
- Authentication
- Authorization handled by HDFS

- SPARK-5342
  - Delegation tokens expire in 7 days
  - Spark Streaming
  - Resolved in v1.4
- SPARK-5111
  - HiveContext


Spark 1.3.0
Jobs
Stages
Storage
Environment
Executors
InferenceSchema
application UI
Spark Jobs (?)
Scheduling Mode: FIFO
Completed Jobs: 4
Completed Jobs (4)
Job Id | Description | Submitted | Duration | Stages: Succeeded/Total | Tasks (for all stages): Succeeded/Total
3 | runJob at newParquet.scala:646 | 2015/06/10 17:14:56 | 21 s | 1/1 | 742/742
2 | collect at SparkPlan.scala:83 | 2015/06/10 17:14:24 | 32 s | 2/2 | 743/743
1 | reduce at JsonRDD.scala:51 | 2015/06/10 17:13:27 | 57 s | 1/1 | 742/742
0 | count at SchemaInference.scala:80 | 2015/06/10 17:13:16 | 11 s | 1/1 | 742/742

History Server

Event log directory: hdfs://nameservice1/user/spark/applicationHistory

Showing 1-20 of 8247

App ID	App Name	Started
application_1433440453721_144531	SparkSQLJob	2015/06/12 04:31:05
application_1433440453721_144523	SparkSQLJob	2015/06/12 04:21:42
application_1433440453721_144345	Merge Application	2015/06/12 04:06:22
application_1433440453721_143900	Merge Application	2015/06/12 03:56:12
application_1433440453721_143895	Merge Application	2015/06/12 03:54:24
application_1433440453721_143840	Merge Application	2015/06/12 03:06:21
application_1433440453721_143580	Merge Application	2015/06/12 02:55:25
application_1433440453721_143574	Merge Application	2015/06/12 02:54:05
application_1433440453721_143507	Merge Application	2015/06/12 02:05:55
application_1433440453721_143254	Merge Application	2015/06/12 01:55:09
application_1433440453721_143247	Merge Application	2015/06/12 01:53:55
application_1433440453721_143242	SparkSQLJob	2015/06/12 01:37:22
application_1433440453721_143236	SparkSQLJob	2015/06/12 01:30:57
application_1433440453721_143231	SparkSQLJob	2015/06/12 01:27:00
application_1433440453721_142661	ChangelogCompactionTest	2015/06/12 00:16:03
application_1433440453721_143213	Merge Application	2015/06/12 01:06:45
application_1433440453721_143101	Merge Application	2015/06/12 00:55:10
application_1433440453721_143095	Merge Application	2015/06/12 00:54:05
application_1433440453721_143089	SparkSQLJob	2015/06/12 00:47:23
application_1433440453721_142431	Kafka Processor - Json topics: ssl	2015/06/12 00:16:17

Show incomplete applications

Data Locality

- Executors are started before data
  - No Data Locality
- Pass data locations to SparkContext

```
val locations = InputFormatInfo
  .computePreferredLocations(Seq(new InputFormatInfo(new Configuration(),
                                                     classOf[ParquetInputFormat],
                                                     new Path("...")))

val sc = new SparkContext(conf, locations)
```

Second Argument

Parquet

- Schema
- Columnar file format
  - Column pruning
  - Filter predicate push down
- Strong Spark support
  - SparkSQL
  - ParquetInputFormat

Schema

- Contract
  - Multiple teams
  - Producers
  - Consumers
- Data to persist in a typed manner
  - Analytics
- Serve as documentation
  - Develop new applications faster
- Prevent a lot of bugs

Schema Evolution

- Schema merging in Spark v1.3
  - SparkSQL
- Schema evolution
  - Merge old and new compatible versions
- No “Alter table …”

Schema Tools

- Big Investment
- Services
  - Creating and retrieving schema
  - Validating schema evolution
- Libraries for producers and consumers
  - Multiple languages

Speed

2 to 4 times FASTER

- Columnar file format
- Column pruning
  - Wide Table
- Filter predicate push down
- Compression

Spark UDK

- Uber Development Kit
  - Specific to Uber Environment
- Help users get their jobs up and running quickly.
- UDK doesn't wrap Spark API.
  - We embrace it!

Template Class

- Memory
  - executor-memory
  - driver-memory
  - spark.yarn.executor.memoryOverhead
  - spark.yarn.driver.memoryOverhead
  - spark.kryoserializer.buffer.max.mb
  - spark.driver.maxResultSize
- CPU
  - num-executors
  - executor-cores
- High Availability
  - spark.eventLog.overwrite
- spark.serializer to org.apache.spark.serializer.KryoSerializer
- spark.speculation
- parquet.enable.binaryString

- Default for Uber environment
  - e.g. HBase
- Default high performance and failover settings
  - Specific Spark version.
- Data store API
- API for common computation
- UDF
- Logging

Uber Use Cases

Inference | Cleaning | Parquet

- ETL
  - JSON in gzip
  - Avro
- Schema Inference
  - SparkSQL

- Data Cleaning by Inferred Schema
- Conversion to Parquet
- Validation
  - Sampling
  - SparkSQL

Analytics

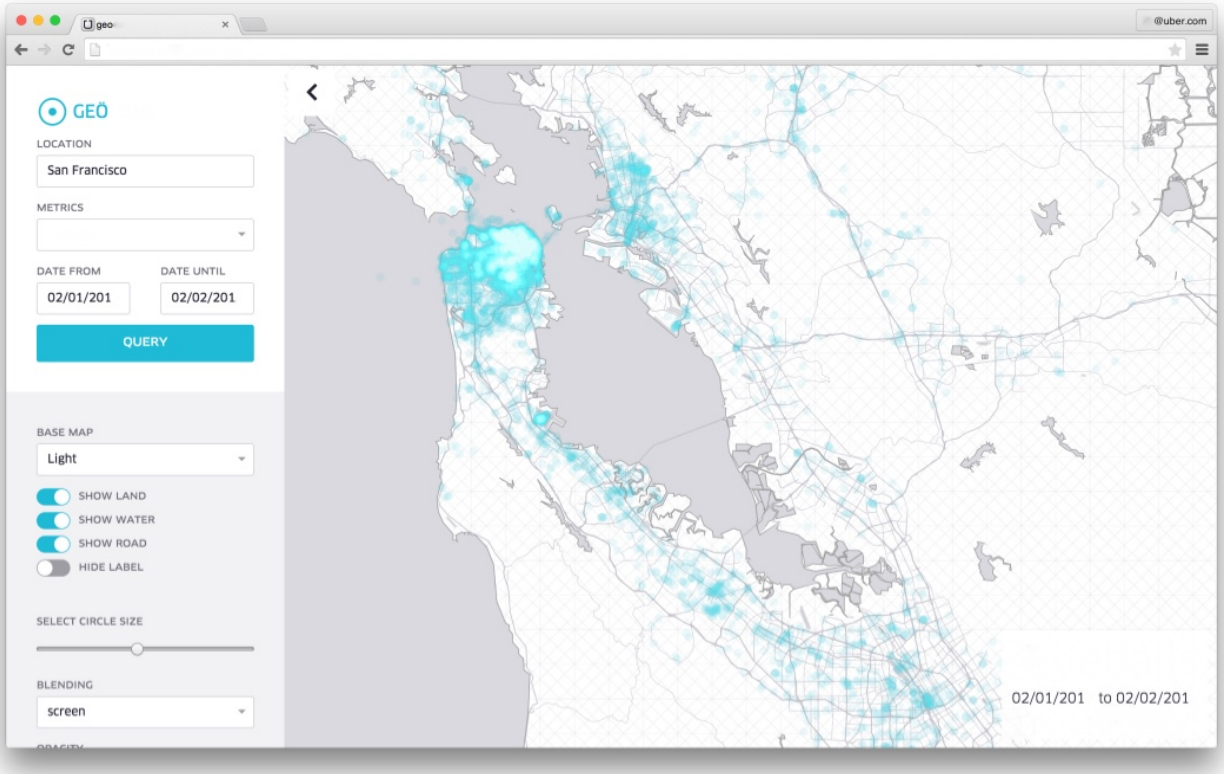
- SparkSQL on Data Lake
- Business metrics
- Data validation
- Spark Job Server
  - Caching for multiple queries via REST

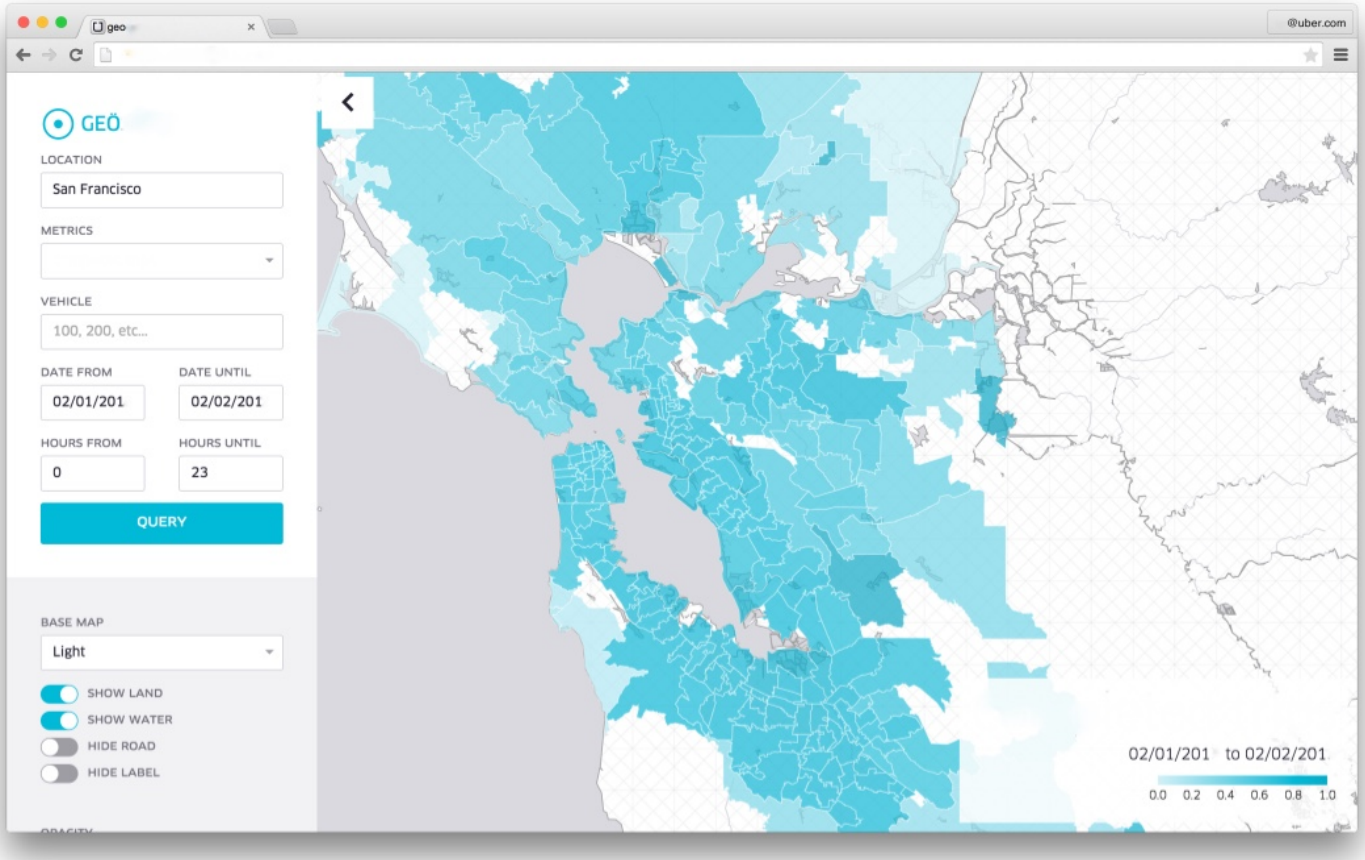
MLlib

- Decision Tree
  - Random Forest
  - Boosting Tree
- K-Mean

- Powerful Algorithms in Many Area
  - API Easy to use
- SPARK-3727: More prediction functionality
- Estimated probability
- Multiple ways of aggregating predictions

Spatial Analysis




geo
@uber.com
GEO
LOCATION
San Francisco
METRICS
VEHICLE
100, 200, etc...
DATE FROM
02/01/201
DATE UNTIL
02/02/201
HOURS FROM
0
HOURS UNTIL
23
QUERY
BASE MAP
Light
SHOW LAND
SHOW WATER
HIDE ROAD
HIDE LABEL
02/01/201 to 02/02/201
0.0 0.2 0.4 0.6 0.8 1.0

Summary

- Motivation
- YARN
- Parquet
- Some Use Cases

Spark Job Server Community Gathering

Today

Welcome to join us!