

# Apache Spark 2.0

Matei Zaharia

@matei\_zaharia



# Apache Spark 2.0

Next major release, coming out this month

- Unstable preview release at [spark.apache.org](https://spark.apache.org)

Remains highly compatible with Apache Spark 1.X

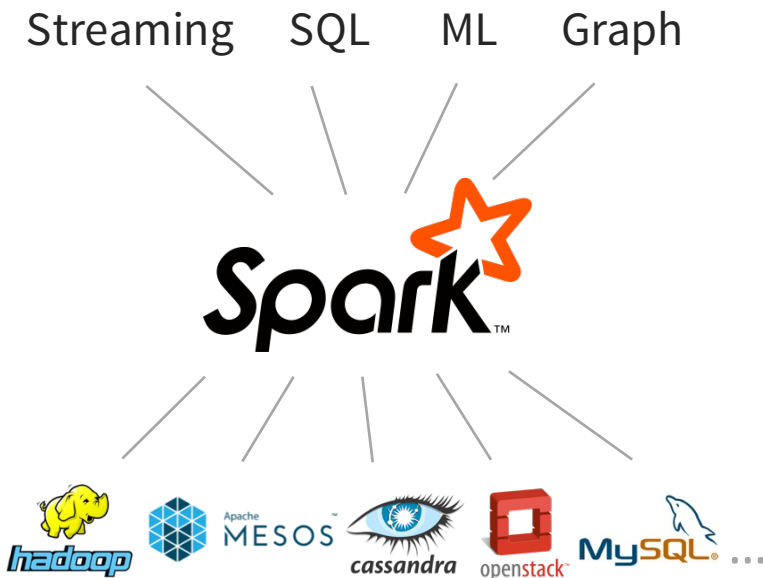
Over 2000 patches from 280 contributors!

# Apache Spark Philosophy

1 Unified engine  
Support end-to-end applications

2 High-level APIs  
Easy to use, rich optimizations

3 Integrate broadly  
Storage systems, libraries, etc



# New in 2.0

Structured API improvements  
(DataFrame, Dataset, SparkSession)

Structured Streaming

MLlib model export

MLlib R bindings

SQL 2003 support

Scala 2.12 support

# Broader Community

Deep learning libraries  
(Baidu, Yahoo!, Berkeley, Databricks)

GraphFrames

PyData integration

Reactive streams

C# bindings: Mobius

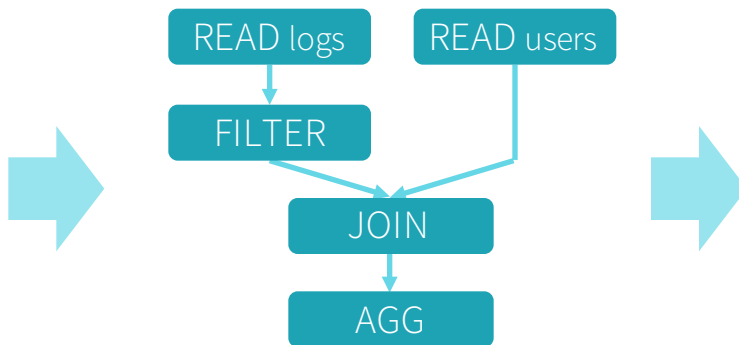
JS bindings: EclairJS

Build on common interface of RDDs & DataFrames

# Deep Dive: Structured APIs

```
events =  
  sc.read.json("/logs")  
  
stats =  
  events.join(users)  
    .groupBy("loc", "status")  
    .avg("duration")  
  
errors = stats.where(  
  stats.status == "ERR")
```

DataFrame API



Optimized Plan

```
while(logs.hasNext) {  
  e = logs.next  
  if(e.status == "ERR") {  
    u = users.get(e.uid)  
    key = (u.loc, e.status)  
    sum(key) += e.duration  
    count(key) += 1  
  }  
}  
...
```

Specialized Code

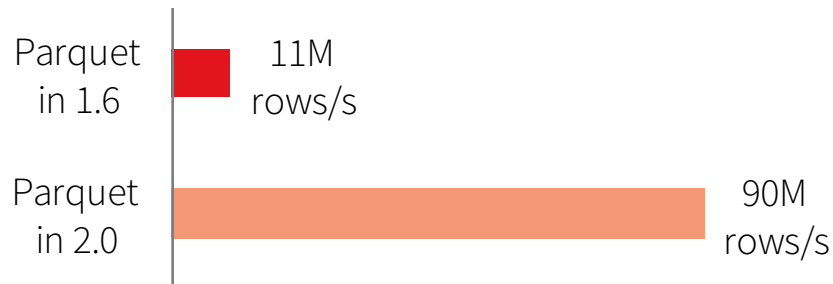
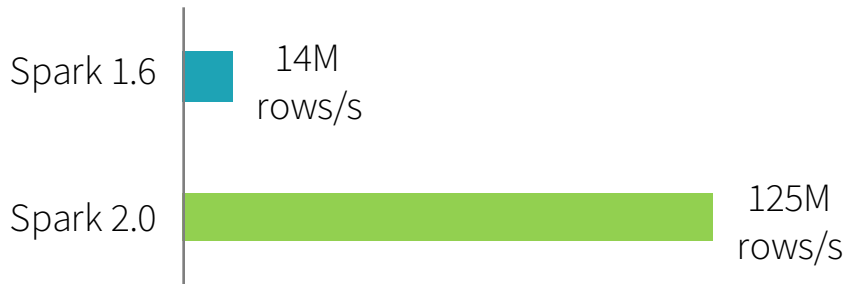
# New in 2.0

## Whole-stage code generation

- Fuse across multiple operators

## Optimized input / output

- Apache Parquet + built-in cache



# Structured Streaming

High-level streaming API built on DataFrames

- Event time, windowing, sessions, sources & sinks

Also supports interactive & batch queries

- Aggregate data in a stream, then serve using JDBC
- Change queries at runtime
- Build and apply ML models

Not just streaming, but  
“continuous applications”

# Structured Streaming API

# Apache Spark 1.X: Static DataFrames

[illegible]

# Apache Spark 2.0: Infinite DataFrames

[illegible]

# Single API



# Example: Batch App

```
logs = ctx.read.format("json").open("s3://logs")
```

```
logs.groupBy("userid", "hour").avg("latency")  
    .write.format("jdbc")  
    .save("jdbc:mysql://...")
```

# Example: Continuous App

```
logs = ctx.read.format("json").stream("s3://logs")
```

```
logs.groupBy("userid", "hour").avg("latency")  
    .write.format("jdbc")  
    .startStream("jdbc:mysql//...")
```

# More Details in Conference

**Engine:** Structuring Spark, Structured Streaming, deep dives

**ML:** SparkR, MLlib 2.0, new algorithms

**Other:** deep learning, GraphFrames, Solr, Cassandra, ...

Try 2.0-preview at [spark.apache.org](https://spark.apache.org)



# Growing the Community

New initiatives from Databricks

# The largest challenge in applying big data is the **skills gap**.

## V. Top Paying Tech

Top Paying Tech in US

Top Paying Tech Worldwide



# Databricks Community Edition



Free version of Databricks with:

- Interactive tutorials
- Apache Spark and popular data science libraries
- Visualization & debug tools



GA Today!

[databricks.com/ce](https://databricks.com/ce)

# Massive Open Online Courses

Free 5-course series on big data with Apache Spark



[dbricks.co/mooc16](https://databricks.co/mooc16)





# Demo

Michael Armbrust

