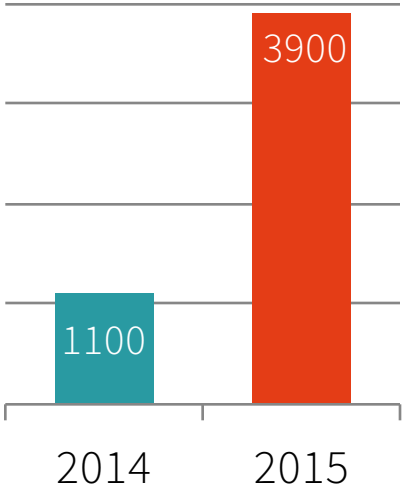
Spark 2.0

Matei Zaharia

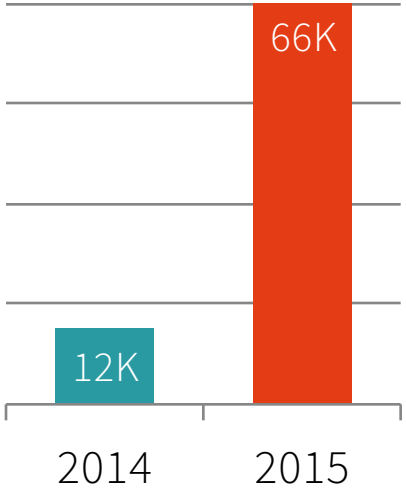
February 17, 2016

2015: A Great Year for Spark

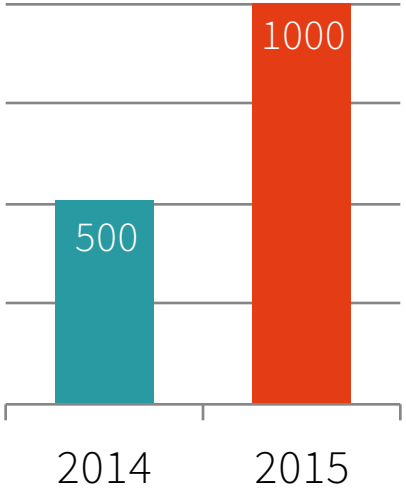
Summit Attendees



Meetup Members



Total Contributors



Meetup Groups: January 2015

source: meetup.com

Meetup Groups: January 2016

source: meetup.com

New Components

DataFrames

Project Tungsten

ML Pipelines

SparkR

Streaming ML

Debug UI

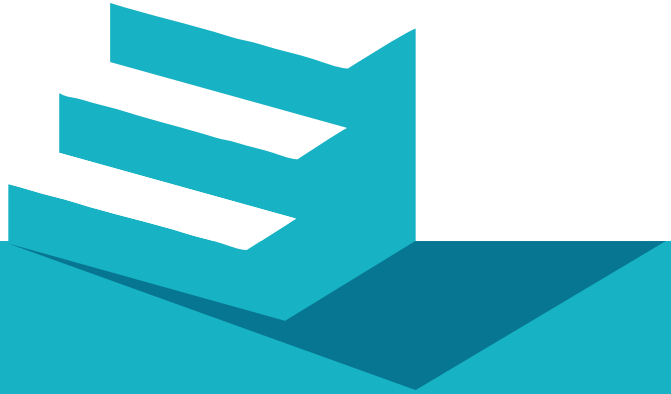
Data Sources

Kafka Connector

Dataset API

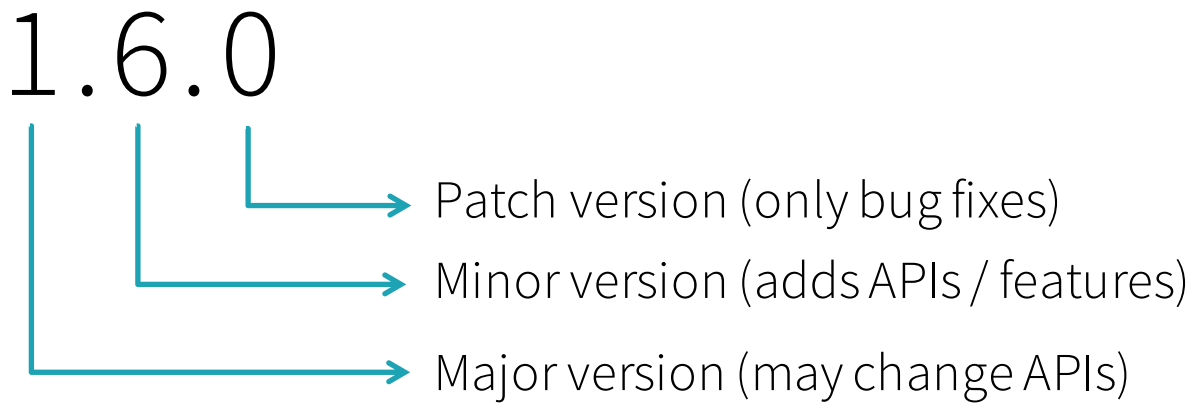
Spark 2.0

Next major release, coming in April / May



Builds on all we learned in past 2 years

Versioning in Spark



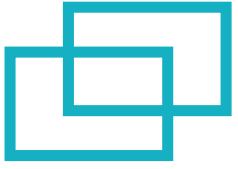
In reality, we hate breaking APIs!

Will not do so except for some dependency conflicts (e.g. Guava)

Major Features in 2.0

Tungsten Phase 2
speedups of 5-10x

Structured Streaming
real-time engine
on SQL/DataFrames



Unifying Datasets
and DataFrames

Tungsten Phase 2

Background on Project Tungsten

CPU speeds have not kept up with I/O in past 5 years

Bring Spark performance closer to bare metal, through:

- Native memory management
- Runtime code generation

Tungsten So Far

Spark 1.4–1.6 added binary storage and basic code gen

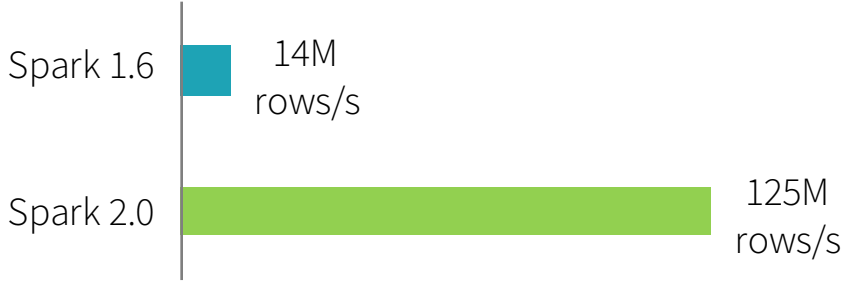
DataFrame + Dataset APIs enable Tungsten in user programs

- Also used under Spark SQL + parts of MLlib

New in 2.0

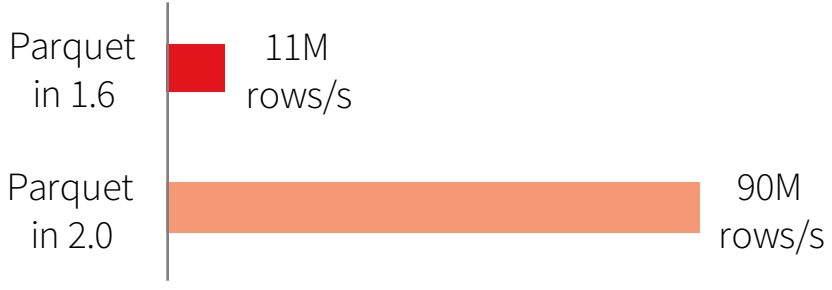
Whole-stage code generation

- Remove expensive iterator calls
- Fuse across multiple operators



Optimized input / output

- Parquet + built-in cache



Automatically applies to SQL, DataFrames, Datasets

Structured Streaming

Background

Real-time processing is increasingly important

Most apps need to combine it with batch & interactive queries

- Track state using a stream, then run SQL queries
- Train an ML model offline, then update it

Spark is very well-suited to do this

Structured Streaming

High-level streaming API built on Spark SQL engine

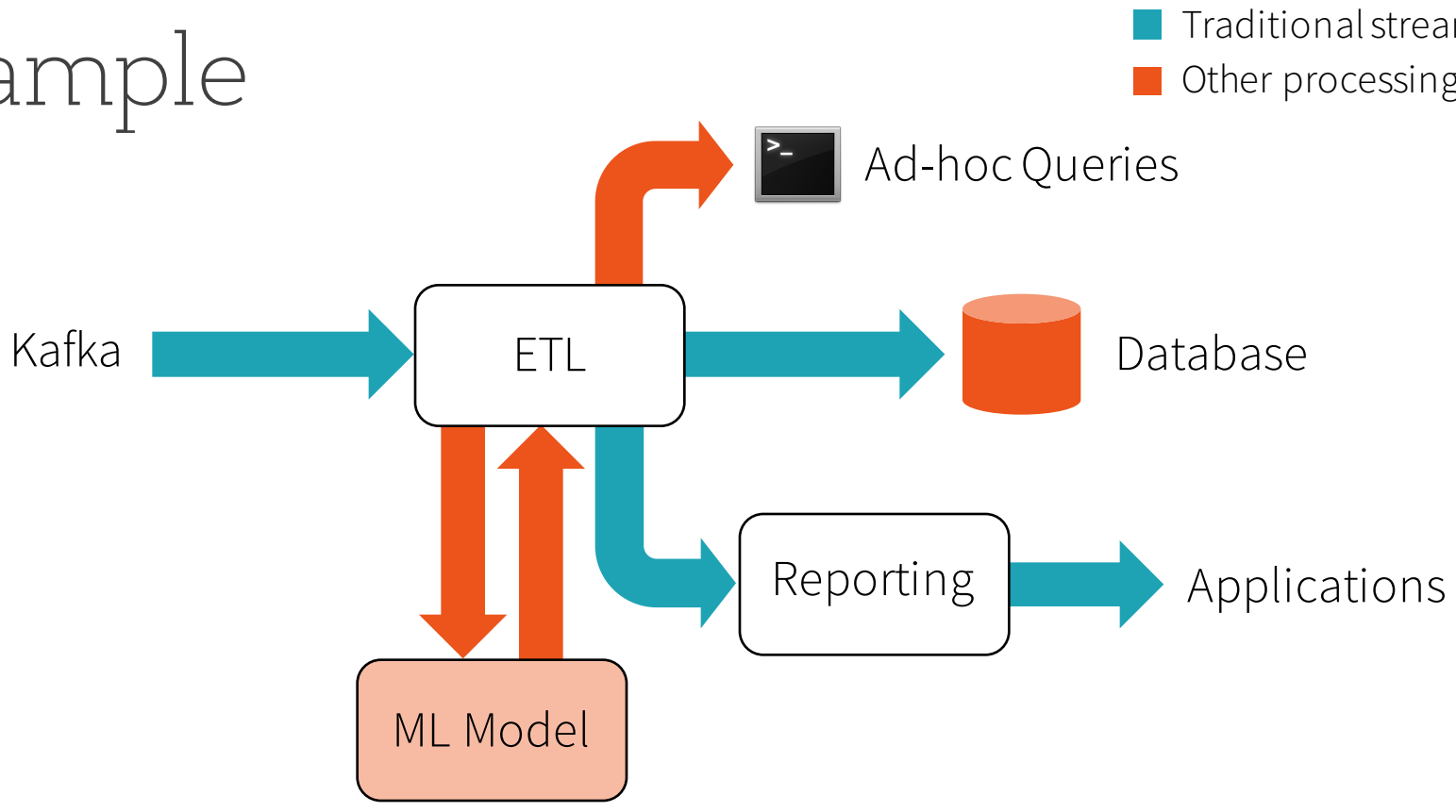
- Declarative API that extends DataFrames / Datasets
- Event time, windowing, sessions, sources & sinks

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"

Example



Goal: end-to-end continuous applications

Details on Structured Streaming

Spark 2.0 will have a first version focused on ETL [SPARK-8360]

Later versions will add more operators & libraries

See Reynold's keynote tomorrow for a deep dive!

Datasets & DataFrames

Datasets and DataFrames

In 2015, we added DataFrames & Datasets as structured data APIs

- DataFrames are collections of rows with a schema
- Datasets add static types, e.g. Dataset[Person]
- Both run on Tungsten

Spark 2.0 will merge these APIs: DataFrame = Dataset[Row]

Example

```
case class User(name: String, id: Int)
case class Message(user: User, text: String)

dataframe = sqlContext.read.json("log.json")      // DataFrame, i.e. Dataset[Row]
messages = dataframe.as[Message]                  // Dataset[Message]

users = messages.filter(m => m.text.contains("Spark"))
                .map(m => m.user)                 // Dataset[User]

pipeline.train(users)                  // MLlib takes either DataFrames or Datasets
```

Benefits

Simpler to understand

- Only kept Dataset separate to keep binary compatibility in 1.x

Libraries can take data of both forms

With Streaming, same API will also work on streams

Long-Term

RDD will remain the low-level API in Spark

Datasets & DataFrames give richer semantics and optimizations

- New libraries will increasingly use these as interchange format
- Examples: Structured Streaming, MLlib, GraphFrames

Thank you!

Enjoy Spark Summit