Spark Community Update

Matei Zaharia & Patrick Wendell

June 15th, 2015

A Great Year for Spark

Most active open source project in data processing

New language: R

Many new features & community projects

Community Growth

	June 2014
total contributors	255
contributors/month	75
lines of code	175,000

Community Growth

	June 2014	June 2015
total contributors	255	730
contributors/month	75	135
lines of code	175,000	400,000

Mostly in libraries

Users

1000+ companies

. . .

Distributors + Apps

50+ companies

. . .

Large-Scale Usage

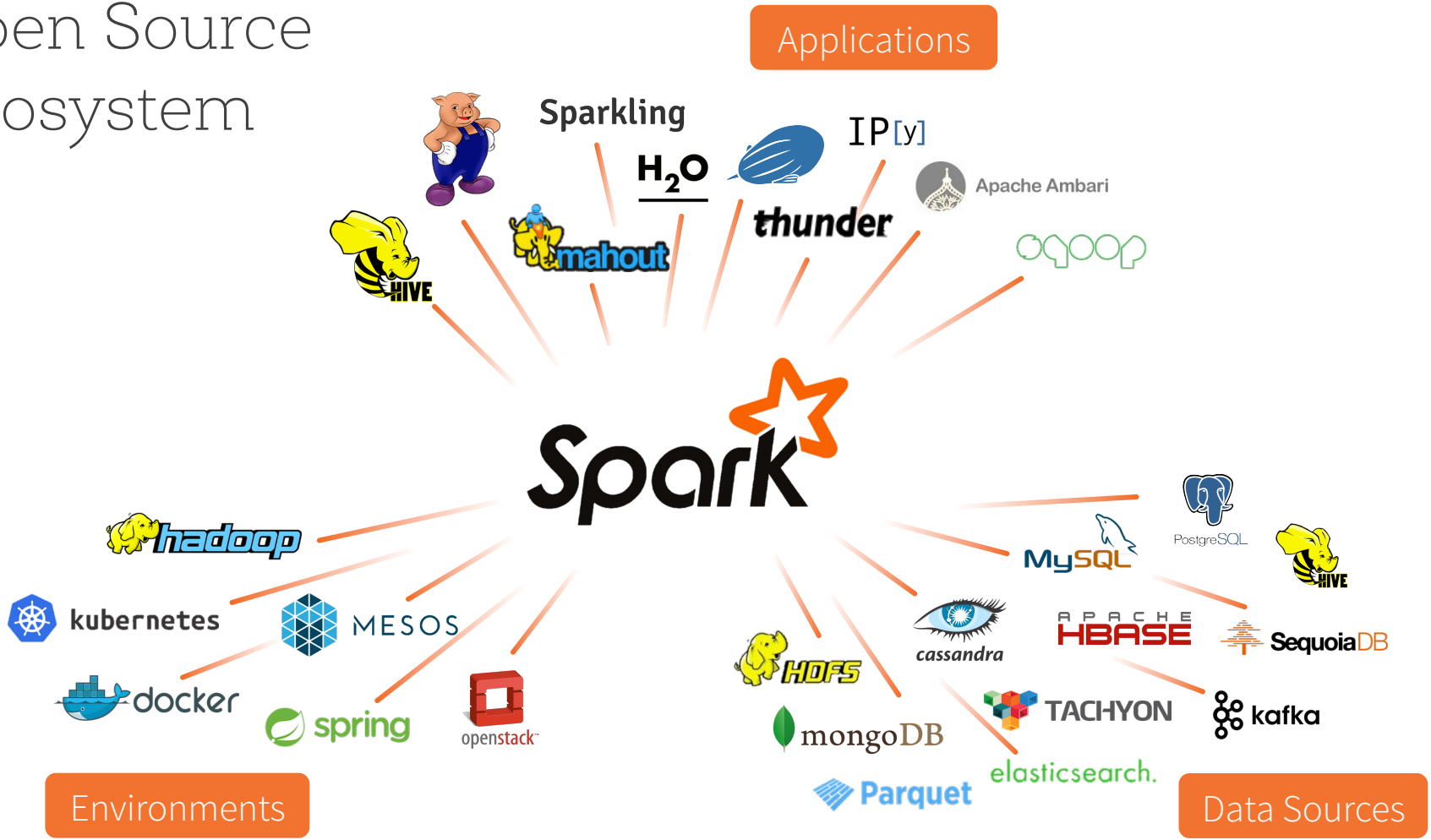
Largest cluster: 8000 nodes 

Largest single job: 1 petabyte 

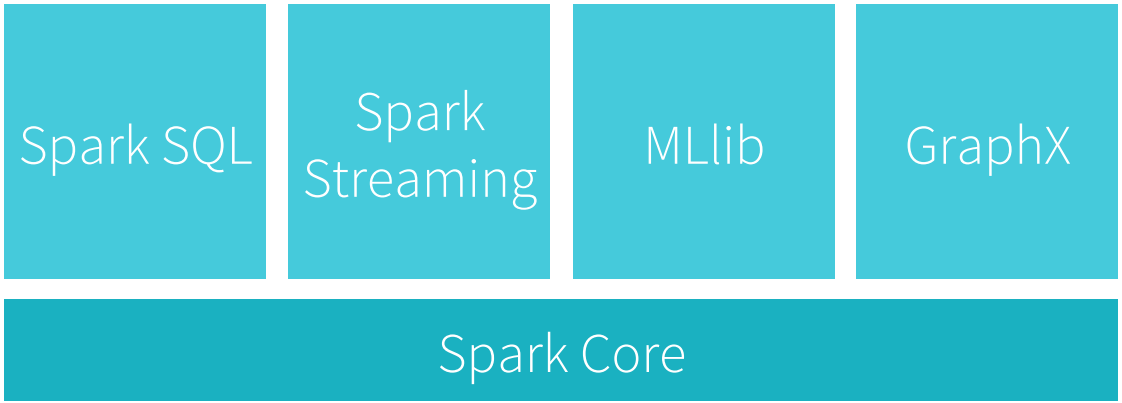
Top streaming intake: 1 TB/hour 

2014 on-disk sort record

Open Source Ecosystem



Current Spark Components



Unified engine across diverse workloads & environments

Major Directions in 2015

Data Science

Similar interfaces to single-node tools

Platform APIs

Growing the ecosystem

Data Science

DataFrames: popular API for data transformation 

Machine Learning Pipelines: inspired by scikit-learn 

R Language 

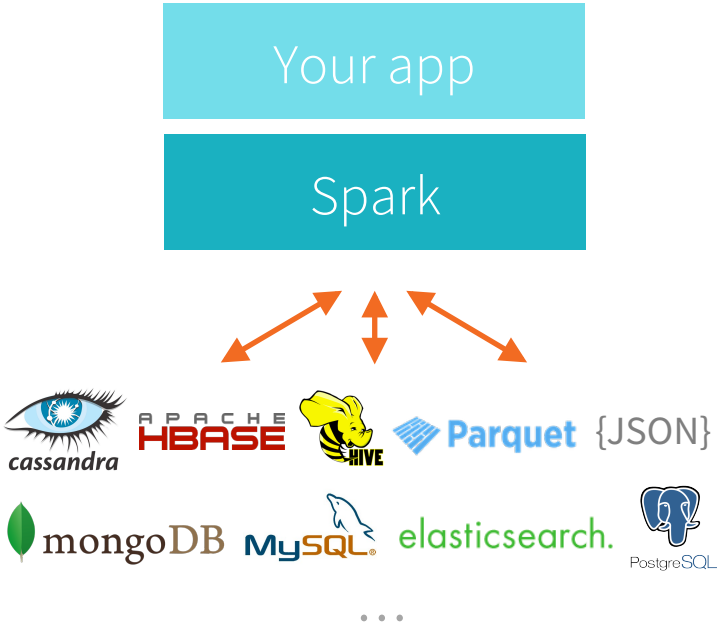
Platform APIs

Data Sources

- Uniform interface to diverse sources (DataFrames + SQL)

Spark Packages

- Community site with 70+ libraries
- spark-packages.org



Ongoing Engine Improvements

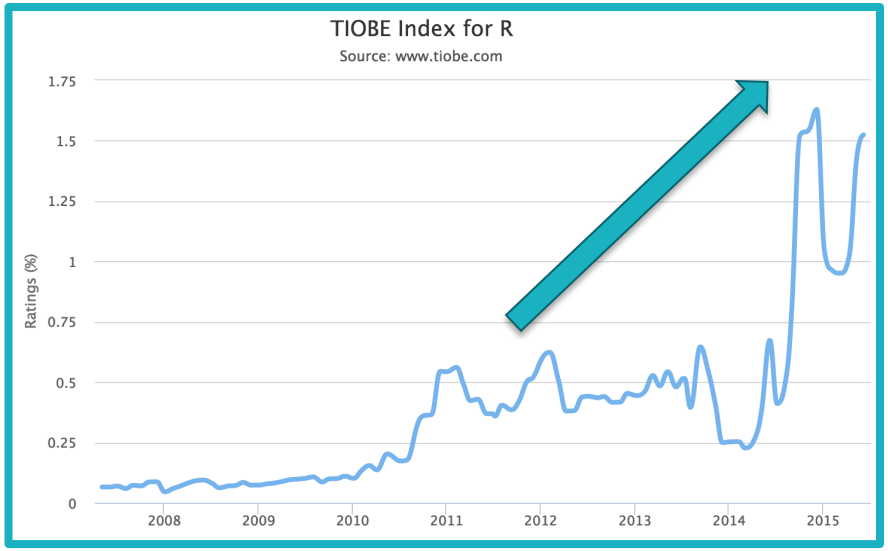
Project Tungsten

- Code generation, binary processing, off-heap memory

DAG visualization & debugging tools

Spark's 1.4 Release

R Language Support



R API based on Spark's DataFrames

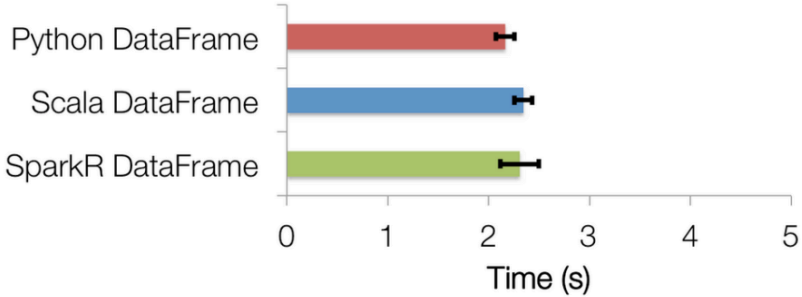
An R Runtime for Big Data

Spark's scale

Thousands of machines and cores

Spark's performance

Runtime optimizer, code generation, memory management



Access to Spark's I/O Packages

```
# Dataframe from JSON
> people <- read.df("people.json", "json")

# ... from MySQL
> people <- read.df("jdbc:mysql://sql01", "jdbc")

# ... from Hive
> people <- read.table("orders")
```

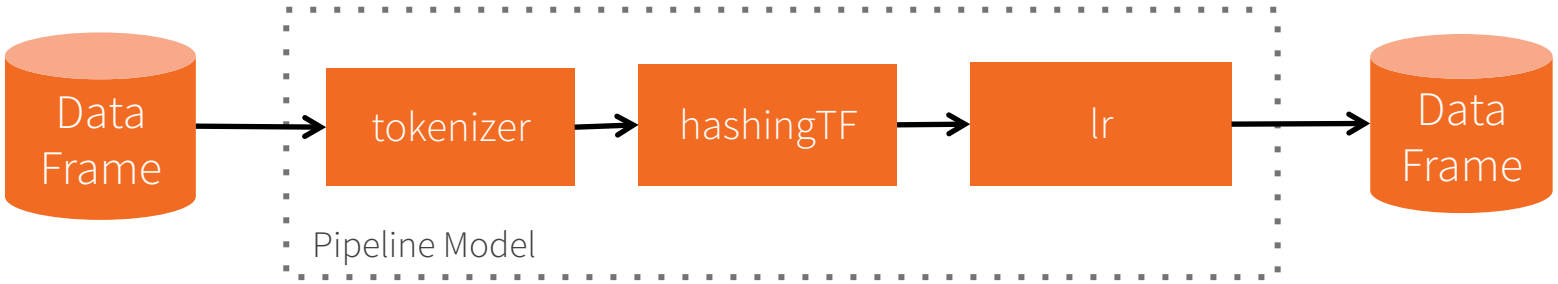
CSV
Avro
Parquet
SequoiaDB
ElasticSearch
Cassandra
MongoDB
HBase

ML Pipelines

```
// create pipeline
tok = Tokenizer(in="text", out="words”)
tf = HashingTF(in=“words”, out="features”)
lr = LogisticRegression(maxIter=10, regParam=0.01)
pipeline = Pipeline(stages=[tok, tf, lr])
```

```
// train pipeline
df = sqlCtx.table(”training”)
model = pipeline.fit(df)

// make predictions
df = sqlCtx.read.json("/path/to/test”)
model.transform(df)
  .select(“id”, “text”, “prediction”)
```



ML Pipelines

Stable API with hooks for third party pipeline components

Feature transformers

VectorAssembler
String/VectorIndexer
OneHotEncoder
PolynomialExpansion
….

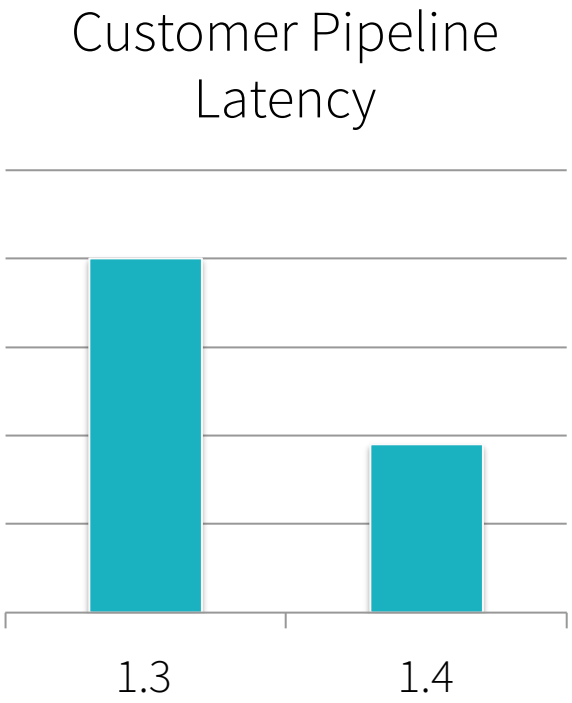
New algorithms

GLM with elastic-net
Tree classifiers
Tree regressors
OneVsRest
…

Performance and Project Tungsten

Managed memory for aggregations

Memory efficient shuffles



What's Coming in Spark 1.5+?

Project Tungsten: Code generation, improved sort + aggregation

Spark Streaming: Flow control, optimized state management

ML: Single machine solvers, scalability to many features

SparkR: Integration with Spark's machine learning APIs

Join Us Today at Office Hours!

	Area
1:00-1:45	Spark Core, YARN Spark Streaming
1:45-2:30	Spark SQL
3:00-3:40	Spark Ops
3:40-4:15	Spark SQL
4:30-5:15	Spark Core, PySpark Spark MLlib
5:15-6:00	Spark MLlib

Databricks booth (A1)

More tomorrow…

Thanks!