Who am I?

Reynold Xin

Spark PMC member

Databricks co-founder & architect

UC Berkeley AMPLab PhD (on leave)
About Databricks

Founded by creators of Spark and is the largest contributor to Spark

Databricks Cloud (in limited availability)
- Fully managed Spark clusters (one-click launch)
- Interactive workspace
- Production data pipelines
- 3rd party applications
Show of Hands!

How familiar are you with Spark?

A. Heard of it, but haven't used it before.
B. Kicked the tires with some basics.
C. Worked or working on a POC.
D. Worked or working on a production deployment.
Fast and general engine for distributed data processing
2014: an Amazing Year for Spark

Total contributors: 150 → 500

Lines of code: 190K → 370K

500+ active production deployments
Contributors per Month to Spark

Most active project in big data
Note: not a scientific comparison.
2014 Focus

Richer libraries

Core engine: stability, performance, and enterprise readiness
Libraries in 2014

Spark SQL

Java 8 closure support

Python streaming

GraphX

Random forests

Streaming MLlib

...
Core Engine in 2014

Revamped “shuffle” and network transport
  – Higher throughput and lower memory usage

Much better memory estimation and management
  – harder to get OutOfMemoryError

Enterprise Readiness
  – YARN integration, security, …
On-Disk Sort Record:
Time to sort 100TB

2013 Record: Hadoop
2100 machines
72 minutes

2014 Record: Spark
207 machines
23 minutes

Source: Daytona GraySort benchmark, sortbenchmark.org
Continuing in 2015

Performance & robustness

Security
  – Option to encrypt all communication (control & data planes)

Usability
  – Visualization and debugging tools
Continuing in 2015

MLlib
  – Many new algorithms

GraphX
  – Java API

Streaming
  – Stronger integration with Kafka, etc

SQL
  – Performance improvements, robustness
New Directions in 2015

Data Science
Making it easier for wider class of users

Platform Interfaces
Scaling the ecosystem
public static class WordCountMapClass extends MapReduceBase implements Mapper<LongWritable, Text, Text, IntWritable> {
    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(LongWritable key, Text value,
                    OutputCollector<Text, IntWritable> output,
                    Reporter reporter) throws IOException {
        String line = value.toString();
        StringTokenizer itr = new StringTokenizer(line);
        while (itr.hasMoreTokens()) {
            word.set(itr.nextToken());
            output.collect(word, one);
        }
    }
}

public static class WorkdCountReduce extends MapReduceBase implements Reducer<Text, IntWritable, Text, IntWritable> {

    public void reduce(Text key, Iterator<IntWritable> values,
                        OutputCollector<Text, IntWritable> output,
                        Reporter reporter) throws IOException {
        int sum = 0;
        while (values.hasNext()) {
            sum += values.next().get();
        }
        output.collect(key, new IntWritable(sum));
    }
}

val file = spark.textFile("hdfs://...")
val counts = file.flatMap(line => line.split(" "))
    .map(word => (word, 1))
    .reduceByKey(_ + _)
counts.saveAsTextFile("hdfs://...")
Beyond Hadoop Users

Early adopters

Users

Understands MapReduce & functional APIs

Data Scientists
Statisticians
R users … PyData
Data Frames

De facto data processing abstraction for data science (R and Python)

Google Trends for “dataframe”
Spark DataFrames

Similar API to data frames in R and Pandas

Automatically optimized via Spark SQL

Out in Spark 1.3

```python
df = jsonFile("tweets.json")

df[df["user"] == "matei"]
  .groupBy("date")
  .sum("retweets")
```
DataFrames and SQL share the same optimization/execution pipeline

Maximize code reuse & share optimization efforts
PySpark RDD vs DataFrame

```python
data.map(lambda x: (x.dept, [x.age, 1])) \n  .reduceByKey(lambda x, y: [x[0] + y[0], x[1] + y[1]]) \n  .map(lambda x: [x[0], x[1][0] / x[1][1]]) \n  .collect()

data.groupBy("dept").avg("age")
```
Machine Learning Pipelines

High-level API inspired by SciKit-Learn

Featurization, evaluation, parameter search

```python
tokenizer = Tokenizer()
tf = HashingTF(numFeatures=1000)
lr = LogisticRegression()

pipe = Pipeline([tokenizer, tf, lr])
model = pipe.fit(df)
```
Spark 1.4 (June)

Exposes DataFrames, and ML library in R

```r
df = jsonFile("tweets.json")

summarize(
    group_by(
        df[df$user == "matei",],
        "date"),
    sum("retweets"))
```
Data Science at Scale

Higher level interfaces in Scala, Java, Python, R

Drastically easier to program Big Data
– With APIs similar to single-node tools
New Directions in 2015

Data Science
Making it easier for wider class of users

Platform Interfaces
Scaling the ecosystem
Spark Mailing Lists (Mar 2015)

user: 2,577      dev: 456

issues: 4,473    reviews: 9,993
#5539 opened an hour ago by rakeshchalasani

[SPARK-6198][SQL] Support "select current_database()"
#5538 opened 2 hours ago by DoingDone9

[SPARK-6955][NETWORK]Do not let Yarn Shuffle Server retry its server port.
#5537 opened 3 hours ago by SaintBacchus

[SPARK-6954] [YARN] Dynamic allocation: numExecutorsPending in ExecutorAllocationManager should not become negative
#5536 opened 4 hours ago by piaozhxiu

[SPARK-6952] Handle long args when detecting PID reuse
#5535 opened 5 hours ago by punya

[SPARK-6893][ML] default pipeline parameter handling in python
Feature Requests and Patches

Can I add this new API zipWithIndexWithContext to RDD?

I want support for HBase/Cassandra/Accumulo/…

I want this algorithm for machine learning…
Platformization

Standardize interfaces for integration points (so implementations can run in many versions to come)

Make it as easy as possible to consume these implementations that are not in Apache Spark repo
External Data Sources

Platform API to plug smart data sources into Spark

Returns DataFrames usable in Spark apps or SQL

Pushes logic into sources
External Data Sources

Platform API to plug smart data sources into Spark

Returns DataFrames usable in Spark apps or SQL

Pushes logic into sources

```
SELECT * FROM mysql_users u JOIN hive_logs h
WHERE u.lang = "en"
```
/**
 * ::DeveloperApi::
 * A BaseRelation that can eliminate unneeded columns and filter using selected
 * predicates before producing an RDD containing all matching tuples as Row objects.
 *
 * The actual filter should be the conjunction of all `filters`,
 * i.e. they should be "and" together.
 *
 * The pushed down filters are currently purely an optimization as they will all be evaluated
 * again. This means it is safe to use them with methods that produce false positives such
 * as filtering partitions based on a bloom filter.
 */

@DeveloperApi
trait PrunedFilteredScan {
  def buildScan(requiredColumns: Array[String], filters: Array[Filter]): RDD[Row]
}
Existing Data Sources

built-in

- Parquet
- { JSON }
- JDBC
- PostgreSQL
- Hive
- MySQL
- HDFS
- S3
- H2

external

- AVRO
- CSV
dBase
- Apache HBase
- Elasticsearch
- Cassandra
- Amazon Redshift

and more ...
Spark Packages

Community index of third party packages

```
bin/spark-shell --packages databricks/spark-csv:0.2
```

spark-packages.org
Spark SQL  
Spark Streaming  
MLlib  
GraphX  
Spark Core
Data Sources

Spark Core

DataFrames
- Spark SQL
- Spark Streaming

ML Pipelines
- MLlib
- GraphX

Data Sources
Data Sources

DataFrames
- Spark SQL
- Spark Streaming

ML Pipelines
- MLlib
- GraphX

Spark Core

Spark Core

Data Sources
- Hadoop
- Cassandra
- Hive
- Apache HBase
- PostgreSQL
- CSV
- JSON
- MySQL
- Elasticsearch
Goal: unified engine across data sources, workloads and environments
# Day 1 - Spark Summit 2015

<table>
<thead>
<tr>
<th>TIME</th>
<th>06/15/2015</th>
</tr>
</thead>
<tbody>
<tr>
<td>07:00 AM - 09:00 AM</td>
<td>Registration</td>
</tr>
<tr>
<td>09:00 AM - 12:00 PM</td>
<td>Keynotes</td>
</tr>
<tr>
<td>12:00 PM - 12:00 PM</td>
<td>Lunch</td>
</tr>
</tbody>
</table>

## Track A - Developer
- **Beyond SQL: Spark SQL Abstractions For The Common Spark Job**
  - Michael Armbrust (Databricks)
  - 30 min

## Track B - Data Science
- **Large-scale Lasso and Elastic-Net Regularized Generalized Linear Models**
  - DB Tsai (Alpine Data Labs)
  - 30 min

## Track C - Applications
- **Use of Spark MLLib for Predicting the Offlineing of Digital Media**
  - Christopher Burdorf (NBC Universal)
  - 30 min

---

**Sponsors**

**Platinum**
- Databricks
- Intel
- IBM

**Gold**
- Zoomdata
- Platfora

**Silver**
- Impetus
- Sharethrough
Enjoy Spark Forum at ApacheCon!