A Tutorial on Apache Spark

A Practical Perspective

By Harold Mitchell

World Changers Shaped Here

SMU

The Goal

Learning Outcomes

World Changers Shaped Here

SMU®

The Goal

Learning Outcomes

- NOTE: The setup, installation, and examples assume Windows user
- Learn the following:
  - General knowledge of the Spark tool
  - Build a simple application using PySpark and Jupyter
  - *Good understanding of RDD’s (primary emphasis)*
  - Familiarity with Spark libraries
  - Use cases for Spark

The Goal

Topics

- What is Apache Spark?
- Getting Apache Spark
- Main Components
- A Closer Look at RDDs
- Putting it All Together
- Returning to the Use Case Argument

What is Apache Spark?

An In-depth View

World Changers Shaped Here

SMU®

What is Apache Spark?

Overview

- Apache Spark is considered to be a unified engine for big data processing
- It is further described

" … a unified engine for distributed data processing. Spark has a programming model similar to MapReduce but tends it with a data-sharing abstraction called Resilient Distributed Datasets, or RDDs. … Spark can capture a wide range of processing that previously needed separate engines …"

[Communications of the ACM]

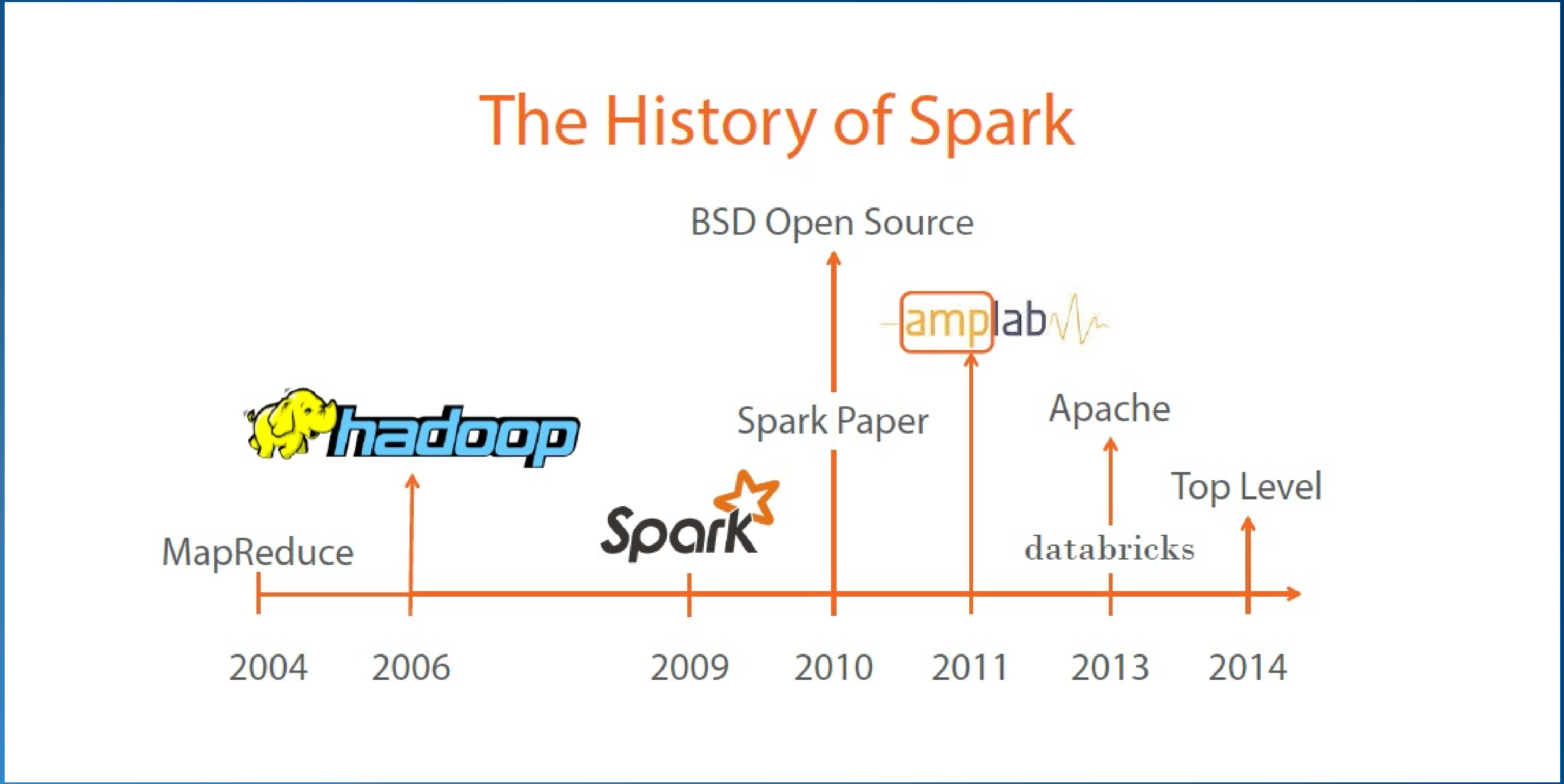
What is Apache Spark?

Overview (cont.)

- The four main engines mentioned previously (coincide with the libraries) include:
  - SQL
  - Streaming
  - Machine Learning
  - Graph Processing

What is Apache Spark

Historical Timeline



What is Apache Spark

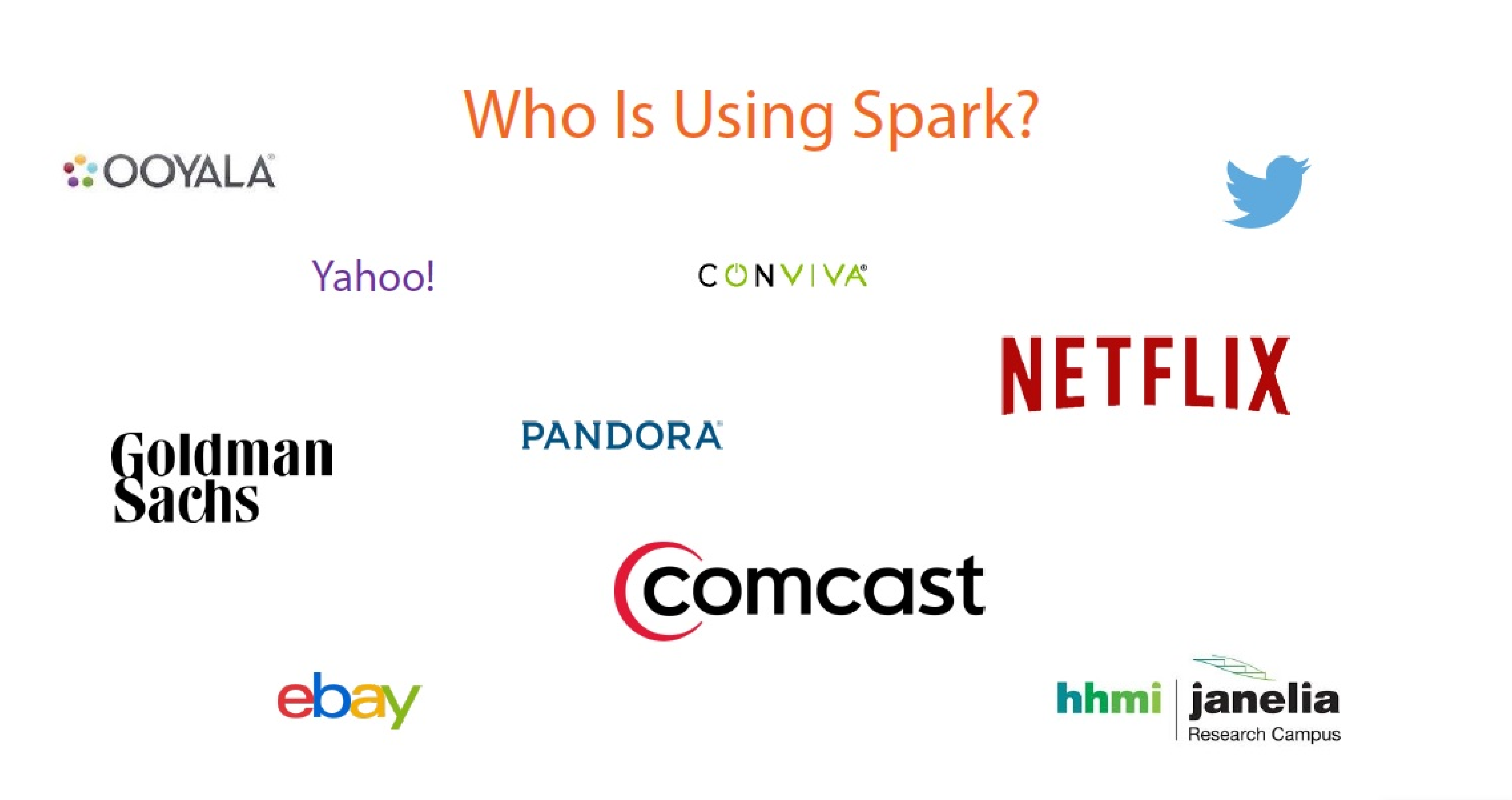
Supported Programming Languages

Spark Languages

Scala
Java
python
R
pluralsight

What is Apache Spark

Who is Using?



Who Is Using Spark?

Getting Apache Spark

Installation Steps for Python Developers

World Changers Shaped Here SMU.

Getting Apache Spark
Prerequisites

- NOTE: These instructions apply to Windows users; however, one is encouraged to use Linux or Mac OS X.
- Prerequisites List
    - Java 7 or above
    - Anaconda (includes Jupyter notebook)
    - Gnu on Windows installed

Getting Apache Spark

Helpful Links

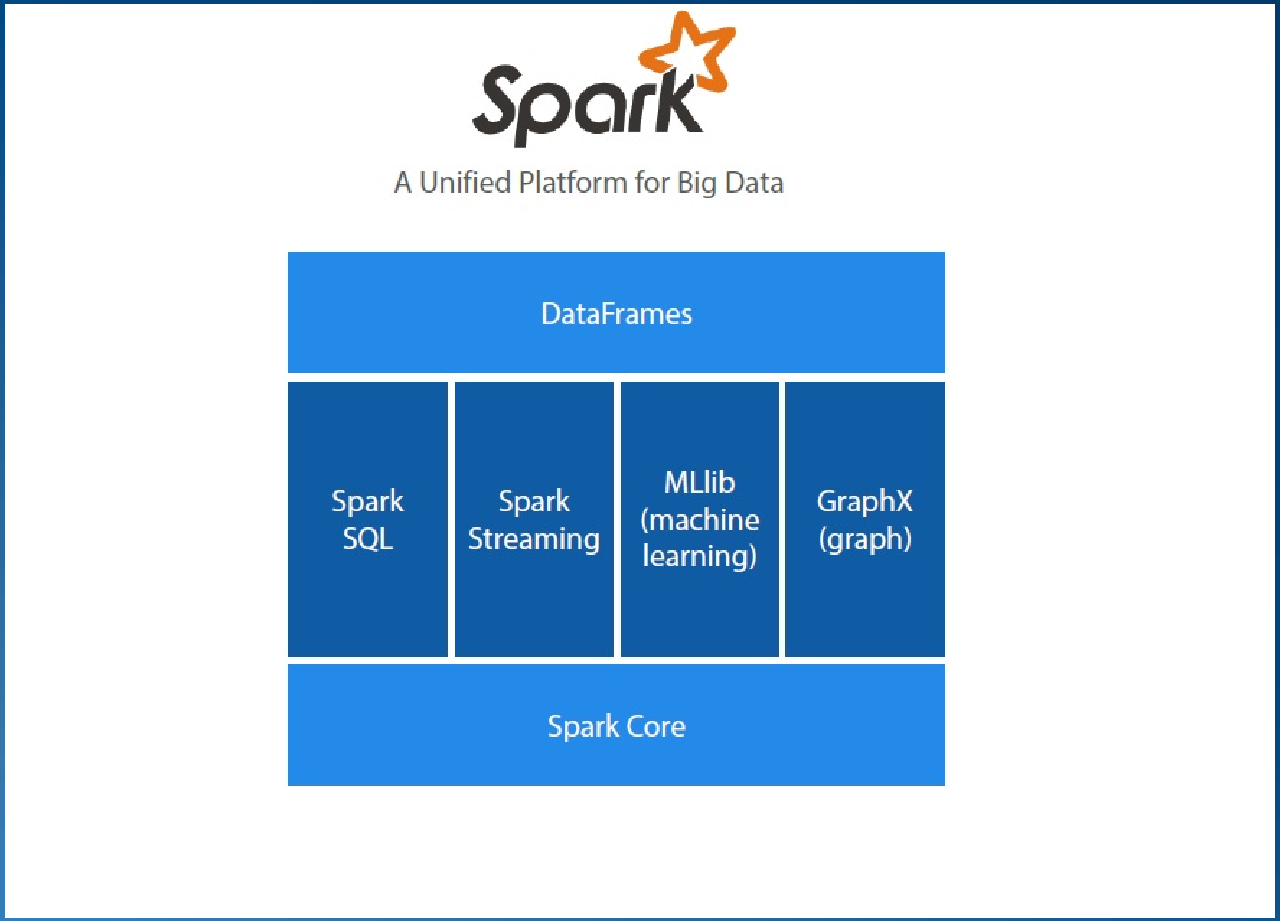
- The Links Below assume Windows installation
  - Anaconda download
  - Anaconda, Gnu Install and Setup
  - Spark, Java Install and Setup
- Good Starting Point for Spark Post-Installation
  - Spark Programming Guide
  - Apache Spark Github

Main Components

DataFrames, Libraries, and The Core

World Changers Shaped Here SMU®

Main Components – High Level



Spark

A Unified Platform for Big Data

DataFrames

Spark SQL

Spark Streaming

MLlib (machine learning)

GraphX (graph)

Spark Core

Main Components

DataFrames

- Similar to DataFrames in both Python and R
- A basic data transformation
  - RDDs of records with a known schema
- Based on relational algebra
- Parallelize and optimize automatically using Spark’s SQL query planner

Main Components

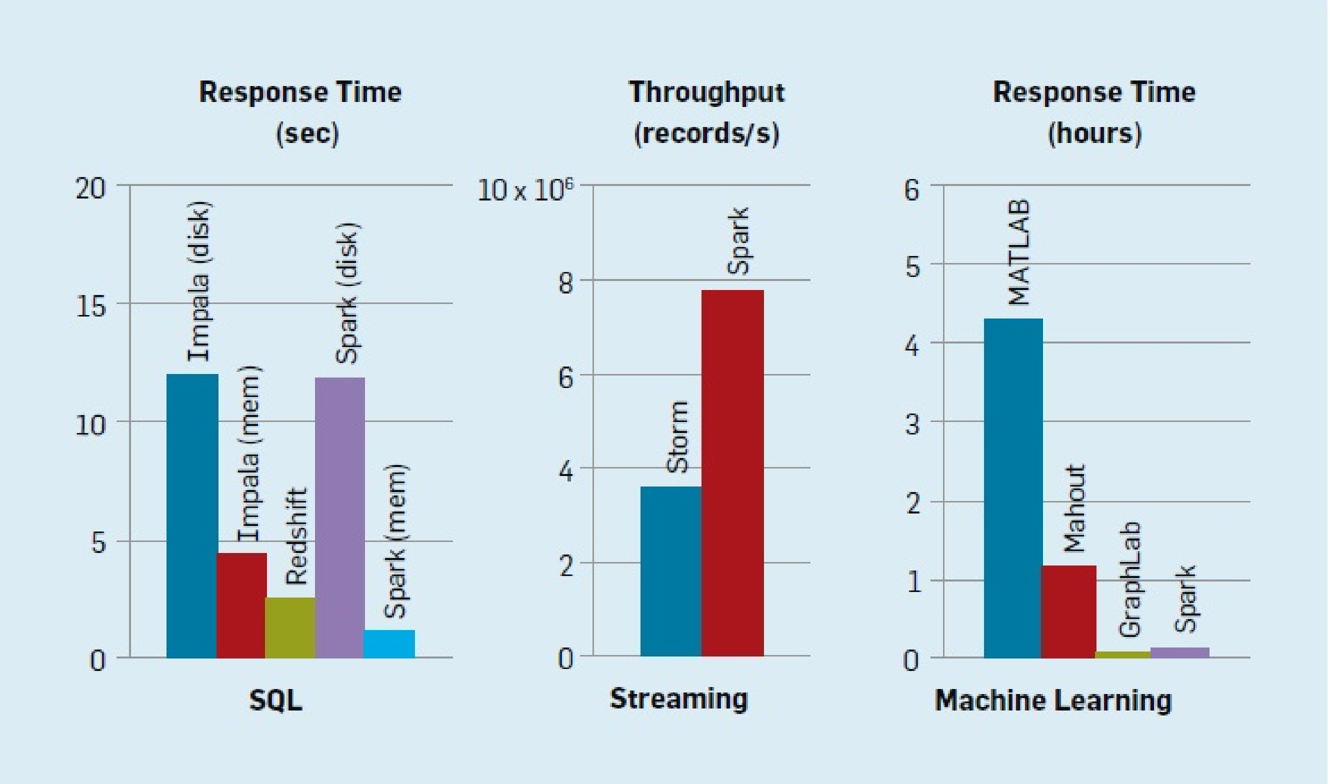
Spark Libraries

- SparkSQL
  - Implements relational queries
  - Supports columnar storage, cost-based optimizations, and code generation for code execution
  - Data sources supported: JSON, HIVE, Avro, Parquet, Amazon Redshift, CSV
- Streaming
  - Implements incremental streaming
  - Uses discretized streams, input data split into to small batches (usu. 200 milliseconds)
- Mllib
  - Machine learning library
  - Implements 50+ common algorithms
- GraphX
  - Graph computation interface
  - Implements vertex partitioning schemes

Main Components

By the Numbers

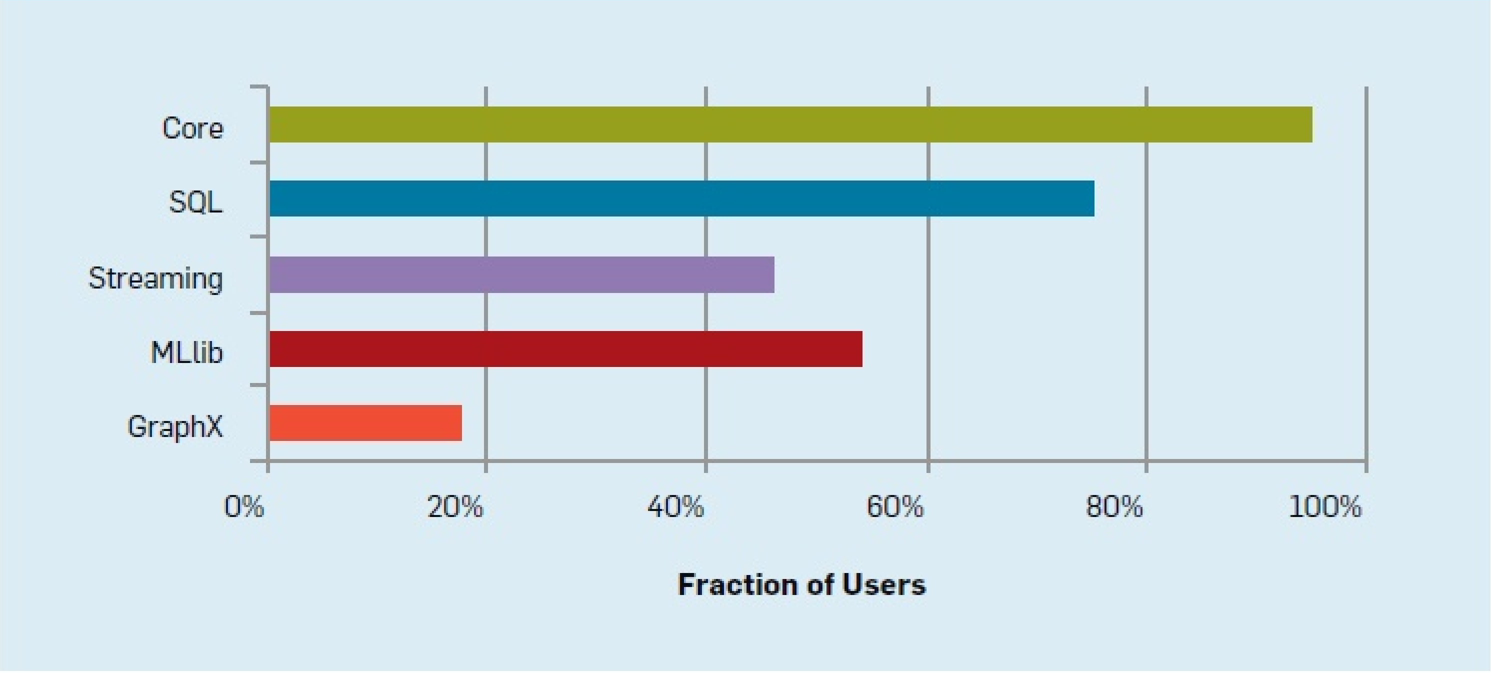
Figure 6. Comparing Spark's performance with several widely used specialized systems for SQL, streaming, and machine learning. Data is from Zaharia[24] (SQL query and streaming word count) and Sparks et al.[17] (alternating least squares matrix factorization).



Main Components

More Numbers

Figure 9. Percent of organizations using each Spark component, from the Databricks 2015 Spark survey; https://databricks.com/blog/2015/09/24/.



Main Components

The Core

- The computing engine for Spark
- Needs and interfaces with
  - A storage system
    - Local file system
    - HDFS
  - A cluster manager
    - Built-in
    - YARN

Main Components

SQL, machine learning, and streaming libraries code

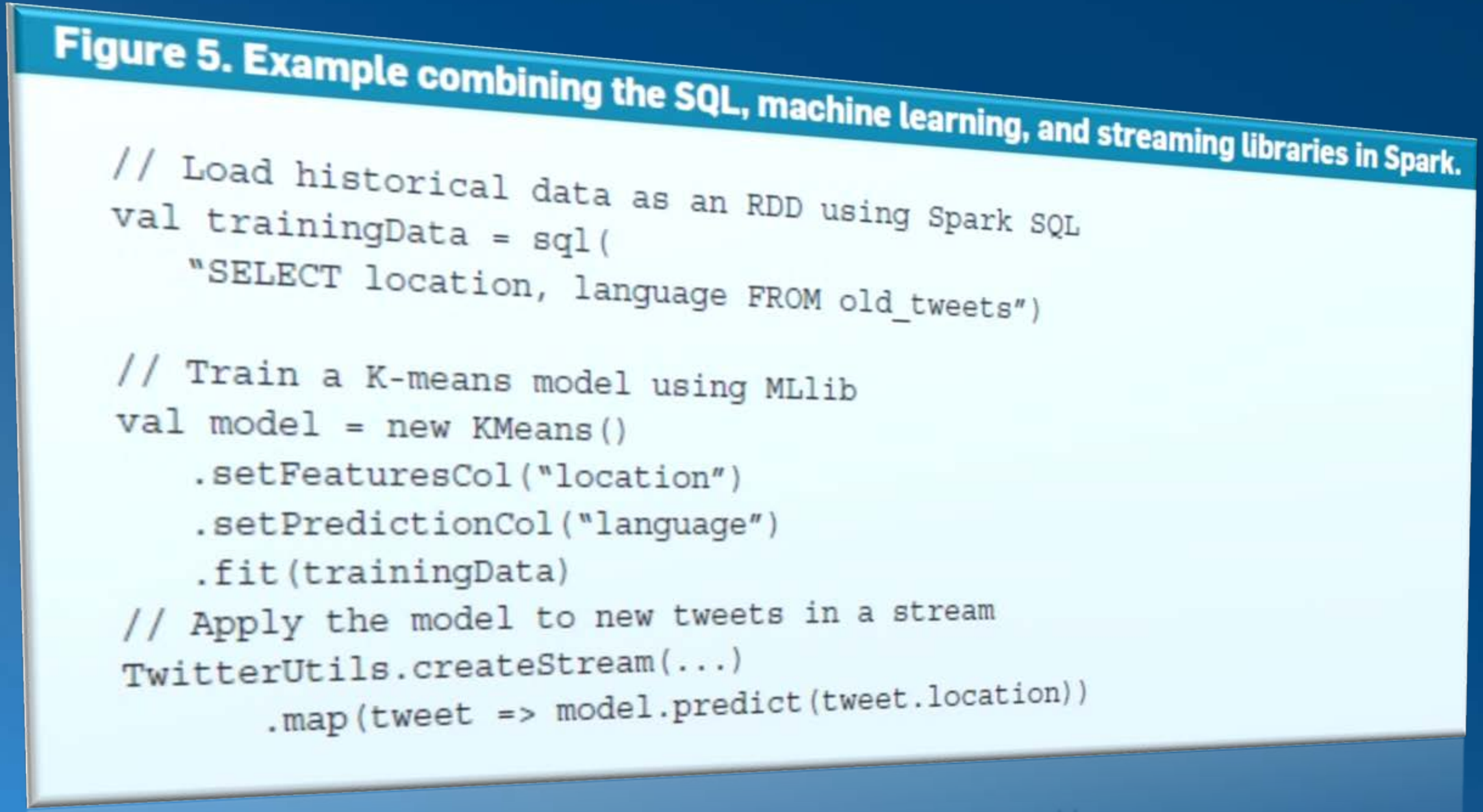


Figure 5. Example combining the SQL, machine learning, and streaming libraries in Spark.

```
// Load historical data as an RDD using Spark SQL
val trainingData = sql(
   "SELECT location, language FROM old_tweets")

// Train a K-means model using MLlib
val model = new KMeans()
   .setFeaturesCol("location")
   .setPredictionCol("language")
   .fit(trainingData)
// Apply the model to new tweets in a stream
TwitterUtils.createStream(...)
      .map(tweet => model.predict(tweet.location))
```

A Closer Look at RDDs

Resilient Distributed Datasets

World Changers Shaped Here SMU

A Closer Look at RDDs

- Spark’s main programming abstraction
- In-memory collection of objects, yet resilient
- Can process billions of rows of data
- APIs for Scala, Java, Python, and R
- Fault-tolerant
- Can be partitioned across clusters and run in parallel
- Read-only, immutable

A Closer Look at RDDs

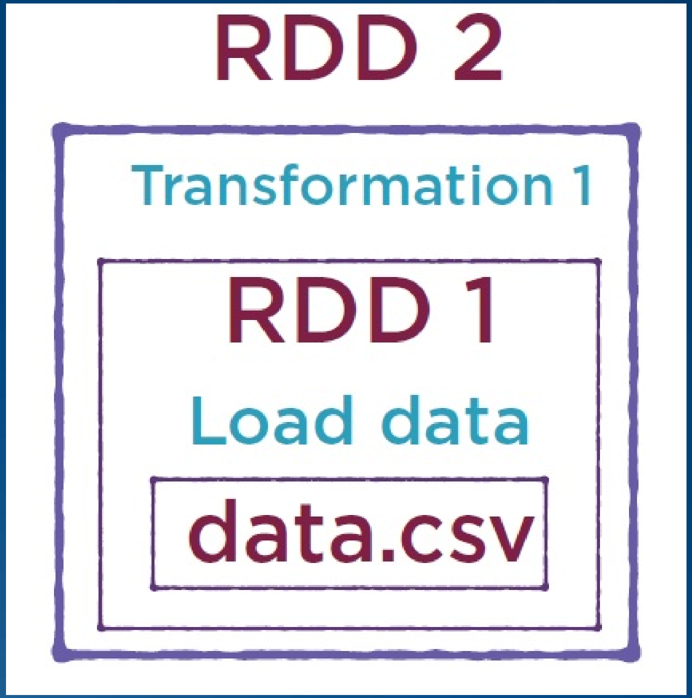
Concepts

- Important Concepts
  - Transformation
    - Applying operations on the data
    - Examples: map, filter, and groupBy
      - Deeper example: 1) Load data, 2) pick only 2nd column, 3) sort the values
  - Actions
    - Requesting a result from the data using an action
    - Data is processed only when user requests a result
    - Examples: 1) Load 1st 10 rows 2) Count the rows 3) Calculate the sum of the rows

A Closer Look at RDDs

Concepts (cont.)

- Lazy Evaluation
  - Spark keeps a record of the series of transformations requested by the user
  - Groups transformations in an efficient way
- Lineage
  - When RDD created just holds metadata
  - Every RDD knows where it came from
  - See illustration to the right



World Changers Shaped Here SMU

Putting it All Together

Two Simple Application Using PySpark

World Changers Shaped Here

SMU®

Putting it All Together

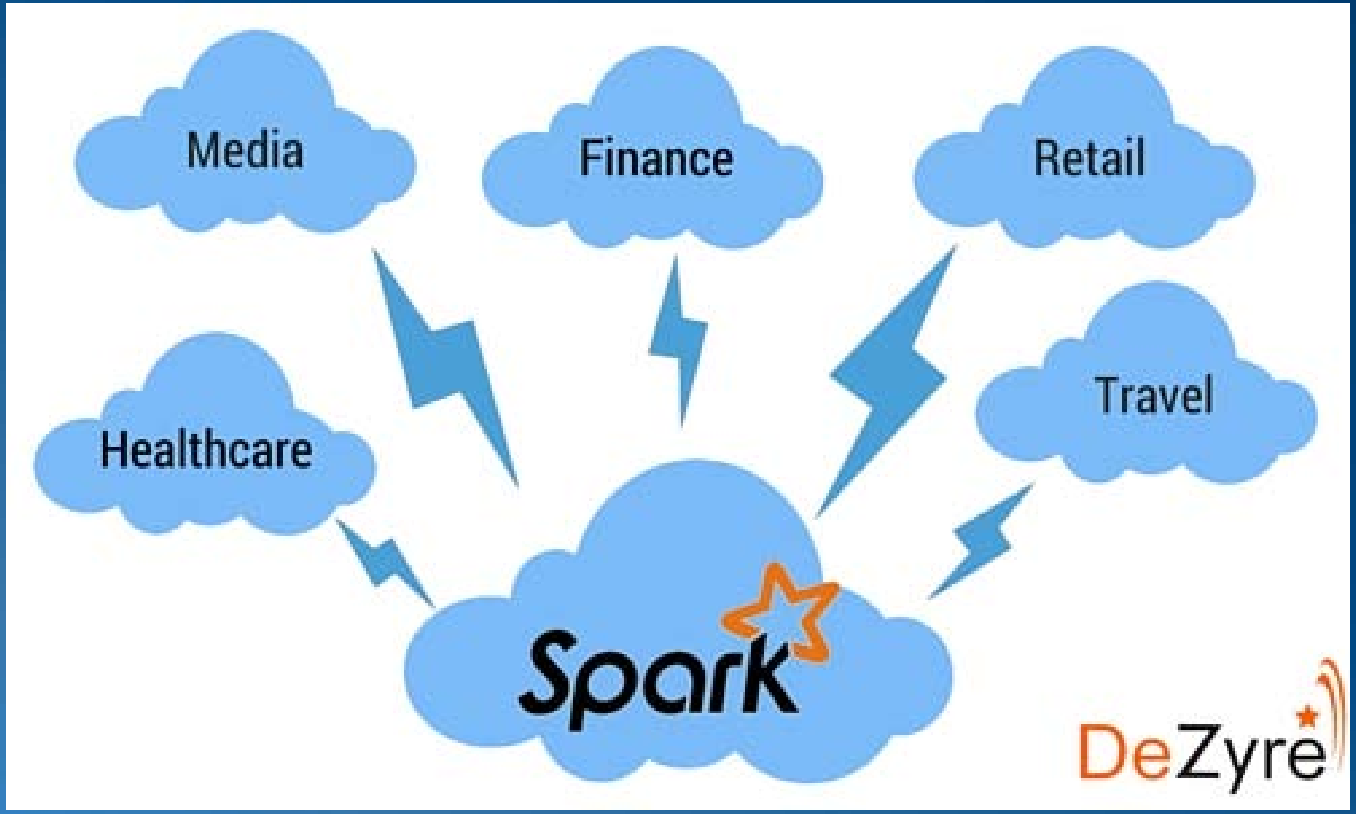
- Talking through demos
  - Pyspark_First_Program.ipynb
  - Spark-HelloWorld.ipynb

The Use Case Argument

Answering What Value is Added

World Changers Shaped Here SMU®

The Use Case Argument
Illustration



The Use Case Argument

Details

- Healthcare
  - MyFitnessPal uses apache spark to clean the data entered by users with the end goal of identifying high quality food items.
- Media
  - Yahoo uses Apache Spark for personalizing its news webpages and for targeted advertising.
- Finance
  - One of the financial institutions that has retail banking and brokerage operations is using Apache Spark to reduce its customer churn by 25%.

The Use Case Argument
Details

- Travel
  - OpenTable, an online real time reservation service, with about 31000 restaurants and 15 million diners a month, uses Spark for training its recommendation algorithms and for NLP of the restaurant reviews to generate new topic models.
- Retail
  - Shopify has processed 67 million records in minutes, using Apache Spark and has successfully created a list of stores for partnership.

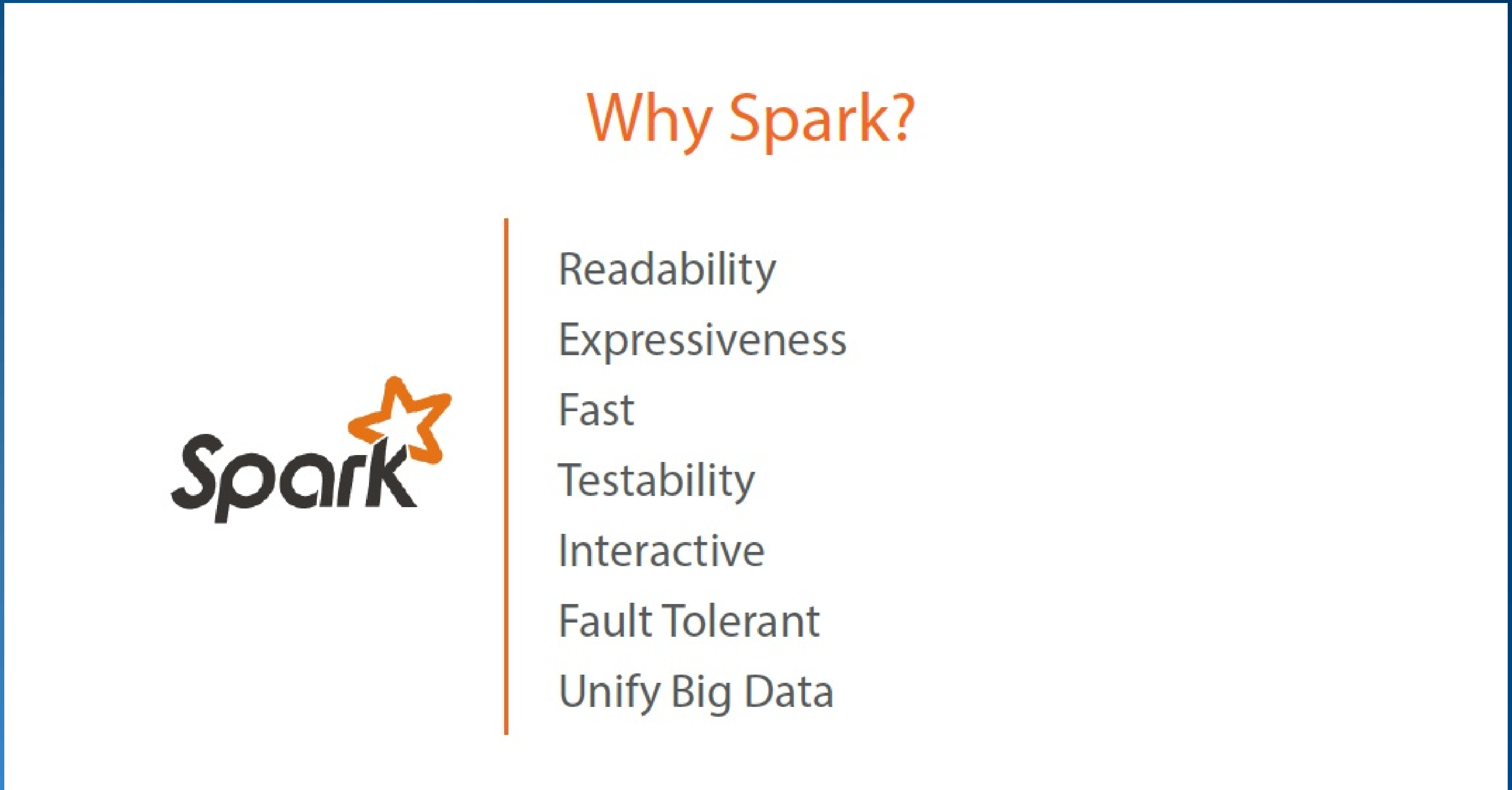
Summary

Let’s Review

World Changers Shaped Here

SMU®

Summary



Why Spark?

- Readability
- Expressiveness
- Fast
- Testability
- Interactive
- Fault Tolerant
- Unify Big Data

The End

References

- Pluralsight Course: Beginning Data Exploration and Analysis with Apache Spark
- Pluralsight Course: Apache Spark Fundamentals
- Communications of the ACM | November 2016 | Vol. 59 | No. 11
- Top 5 Apache Spark Use Cases