Data Processing + Analysis using Spark SQL and DataFrame

Disclaimer

* All data and information in this presentation were from public resource
* This is a vendor-independent talk that expresses teacher own opinions
*
* **Please DO NOT record video**

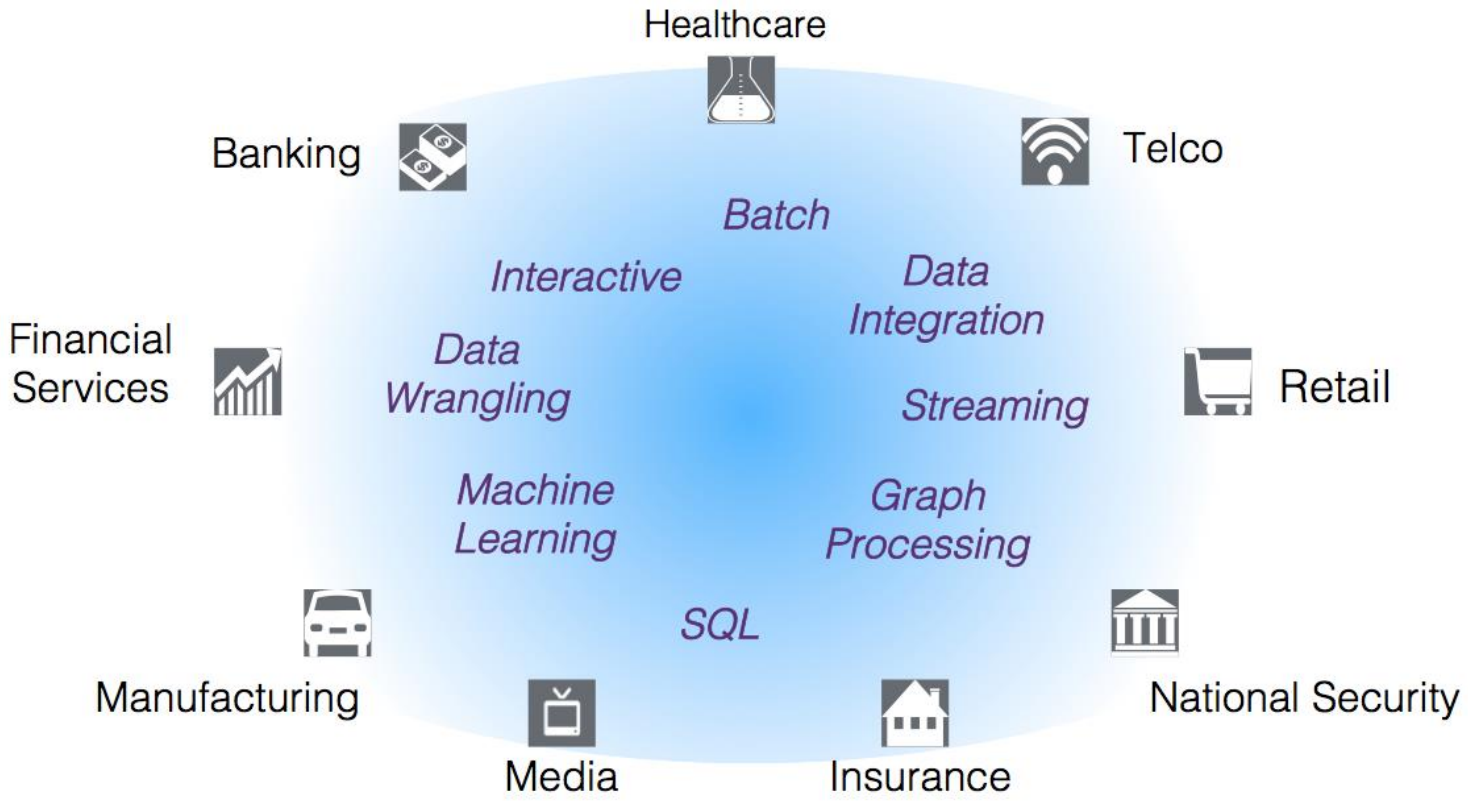
Data Application Lab

Agenda

* Spark Introduction
* Spark SQL and DataFrame
* Spark for Data Analytics
* Demo and Practices

Data Application Lab

Why we need to learn it?
- Spark in the real world



Data Application Lab

What is Spark?

* A fast, general engine for large-scale data processing and analysis
* A technology that interoperates well with Hadoop
* A big data ecosystem

Data Application Lab

What is Spark?

* Like Hadoop, it distributes data across cluster, and process in parallel
* Created by AMPLab now Databricks
* Interoperable with Hadoop
* Written in Scala

Data Application Lab

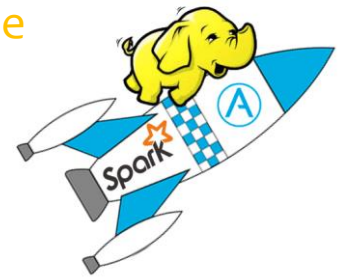
Why Spark is better?

* Improves efficiency through:
  * In-memory computing primitives
  * General computation graphs

Up to 100x faster
(2-10x on disk)

* Improves usability through:
  * Rich APIs in Scala, Java, Python
  * Interactive shell

2-5x less code



Data Application Lab

Not just for In-Memory Data

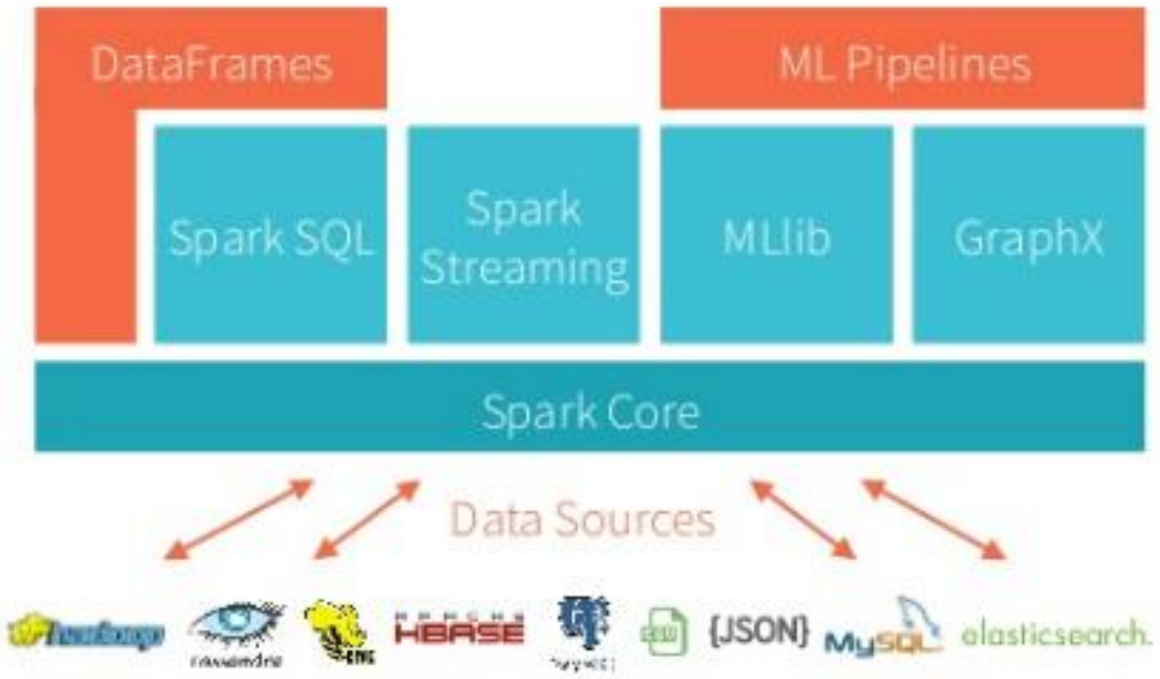
WIRED

Startup Crunches 100 Terabytes of Data in a Record 23 Minutes

	Hadoop Record	Spark 100TB	Spark 1PB
Data Size	102.5TB	100TB	1000TB
Time	72 min	23 min	234 min
# Cores	50400	6592	6080
Rate	1.42 TB/min	4.27 TB/min	4.27 TB/min
Environment	Dedicate	Cloud (EC2)	Cloud (EC2)

Data Application Lab

Spark Stack


DataFrames
ML Pipelines
Spark SQL
Spark Streaming
MLlib
GraphX
Spark Core
Data Sources
HBASE
{JSON}
MySQL
elasticsearch

Spark SQL and DataFrame

* Spark SQL
  * SQL engine with an optimizer
  * Not ANSI compliant
  * Operates on DataFrames(RDD with Schema)
  * Primary goal: Write less code
* DataFrame API
  * Programmatically API to build SQL-like constructs
  * Share machinery with SQL engine

Data Application Lab

Obviously,

Is about more than SQL.

Spark SQL Overview

- Spark's module for working with Structured data (tables with rows/columns)

Spark SQL

Integrated （集成的）

Seamlessly mix SQL queries with Spark programs.

Spark SQL lets you query structured data inside Spark programs, using either SQL or a familiar DataFrame API. Usable in Java, Scala, Python and R.

```
context = HiveContext(sc)

results = context.sql(
  "SELECT * FROM people")

names = results.map(lambda p: p.name)
```

Apply functions to results of SQL queries.

Data Application Lab

Spark SQL

Uniform Data Access

Connect to any data source the same way.

DataFrames and SQL provide a common way to access a variety of data sources, including Hive, Avro, Parquet, ORC, JSON, and JDBC. You can even join data across these sources.

```
context.jsonFile("s3n://...")
  .registerTempTable("json")

results = context.sql(
  """SELECT *
     FROM people
     JOIN json ..."""")
```

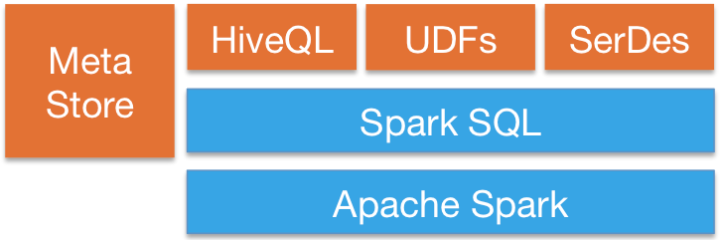
Query and join different data sources.

Data Application Lab

Spark SQL

Hive Compatibility

Run unmodified Hive queries on existing data.



Spark SQL can use existing Hive metastores, SerDes, and UDFs.

Spark SQL reuses the Hive frontend and metastore, giving you full compatibility with existing Hive data, queries, and UDFs. Simply install it alongside Hive.

Data Application Lab

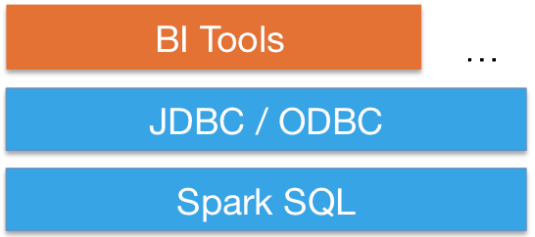
Spark SQL

Spark SQL

Standard Connectivity

Connect through JDBC or ODBC.

A server mode provides industry standard JDBC and ODBC connectivity for business intelligence tools.



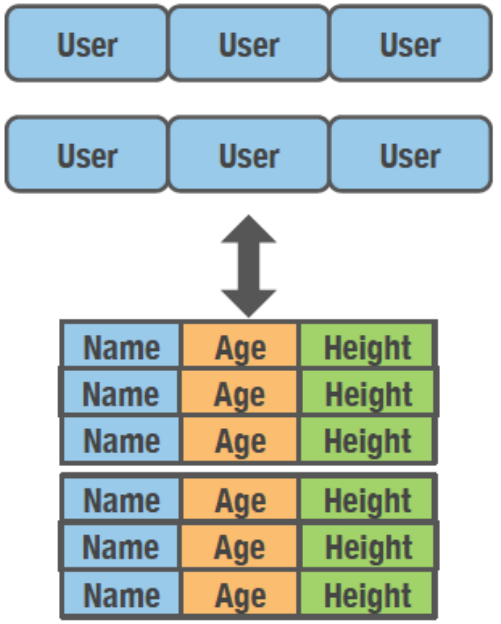
Use your existing BI tools to query big data.

tableau SOFTWARE ZOOMDATA Qlik

Data Application Lab

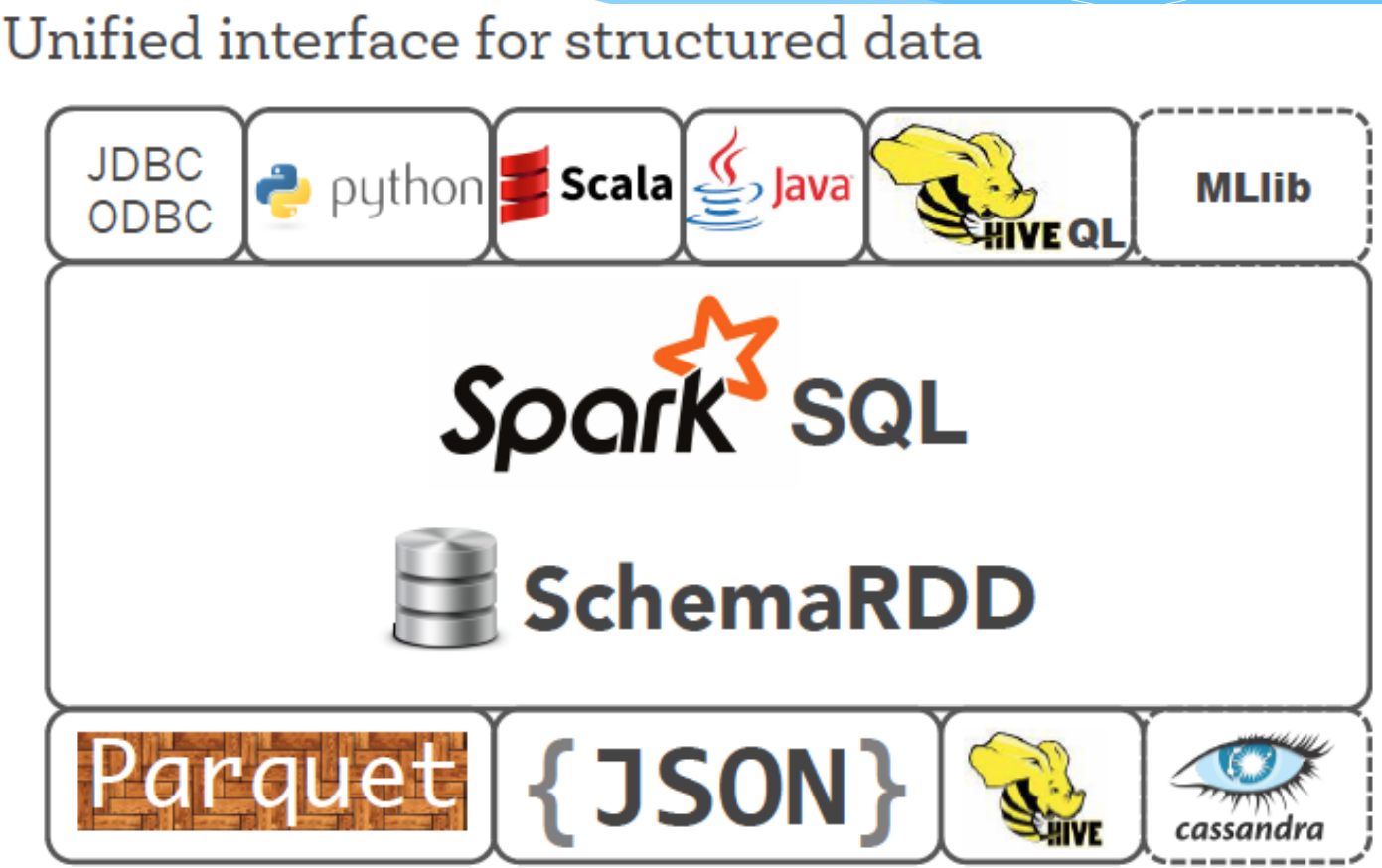
RDD and DataFrame

* Spark + RDD
  * Functional transformations on partitioned collections of objects.
* SQL + DataFrame
  * Declarative transformations on partitioned collections of tuples.



Data Application Lab

DataFrames: More than SQL



Data Application Lab

Getting started: Spark SQL

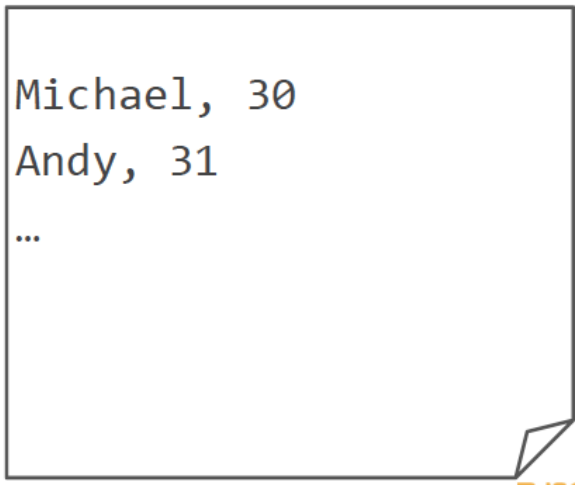
* SQLContext/HiveContext
  * Entry point for all SQL functionality
  * Wraps/extends existing spark context

```
from pyspark.sql import SQLContext
sqlCtx = SQLContext(sc)
```

Data Application Lab

Example Dataset

* A text file filled with people's names and ages:



Data Application Lab

DataFrames as Relations(Python)

```
# sc is an existing SparkContext.
 from pyspark.sql import SQLContext, Row
sqlContext = SQLContext(sc)

# Load a text file and convert each line to a Row.
lines = sc.textFile("examples/src/main/resources/people.txt")
parts = lines.map(lambda l: l.split(","))
people = parts.map(lambda p: Row(name=p[0], age=int(p[1])))

# Infer the schema, and register the DataFrame as a table.
schemaPeople = sqlContext.inferSchema(people)
schemaPeople.registerTempTable("people")
```

Data Application Lab

Querying Using SQL

 # SQL can be run over DataFrames that have been registered as a table.

```
teenagers = sqlContext.sql("SELECT name FROM people WHERE age >= 13
AND age <= 19")
```

 # The results of SQL queries are RDDs and support all the normal RDD operations.

```
teenNames = teenagers.map(lambda p: "Name: " + p.name)
for teenName in teenNames.collect(): print teenName
```

Data Application Lab

Reading Data Store in Hive

```
from pyspark.sql import HiveContext
hiveCtx = HiveContext(sc)

hiveCtx.sql("""CREATE TABLE IF NOT EXISTS src (key INT, value
STRING)""")

hiveCtx.sql("""LOAD DATA LOCAL INPATH 'examples/…/kv1.txt' INTO
TABLE src""")

#Queries can be expressed in HiveQL.
results = hiveCtx.sql("FROM src SELECT key, value").collect()
```

Data Application Lab

DataFrame

* A distributed collection of rows organized into name columns
* Equivalent to a table in a relational database, or a data frame in R/Python
* An abstraction for selecting, filtering aggregating and plotting structured data(R, Pandas)
* Archaic: Previously SchemaRDD(Spark < 1.3)

Data Application Lab

Write Less Code: Input & Output

Unified interface to reading/writing data in a variety of formats

```
df = sqlContext.read \
        .format("json") \
        .option("samplingRatio", "0.1") \
        .load("/home/michael/data.json")

df.write \
    .format("parquet") \
    .mode("append") \
    .partitionBy("year") \
    .saveAsTable("fasterData")
```

Data Application Lab

Spark SQL CSV practice

* Demo - NYC Uber trip data Analysis

Write Less Code: Input & Output

* Spark SQL's Data Source API can read and write Dataframes using a variety of formats.

Built-In

External

and more…


Data Application Lab

Write Less Code: Input & Output

* Spark SQL's Data Source API can read and write Dataframes using a variety of formats.

Using RDDs

```
data = sc.textFile(...).split("\t")
data.map(lambda x: (x[0], [int(x[1]), 1])) \
    .reduceByKey(lambda x, y: [x[0] + y[0], x[1] + y[1]]) \
    .map(lambda x: [x[0], x[1][0] / x[1][1]]) \
    .collect()
```

Using DataFrames

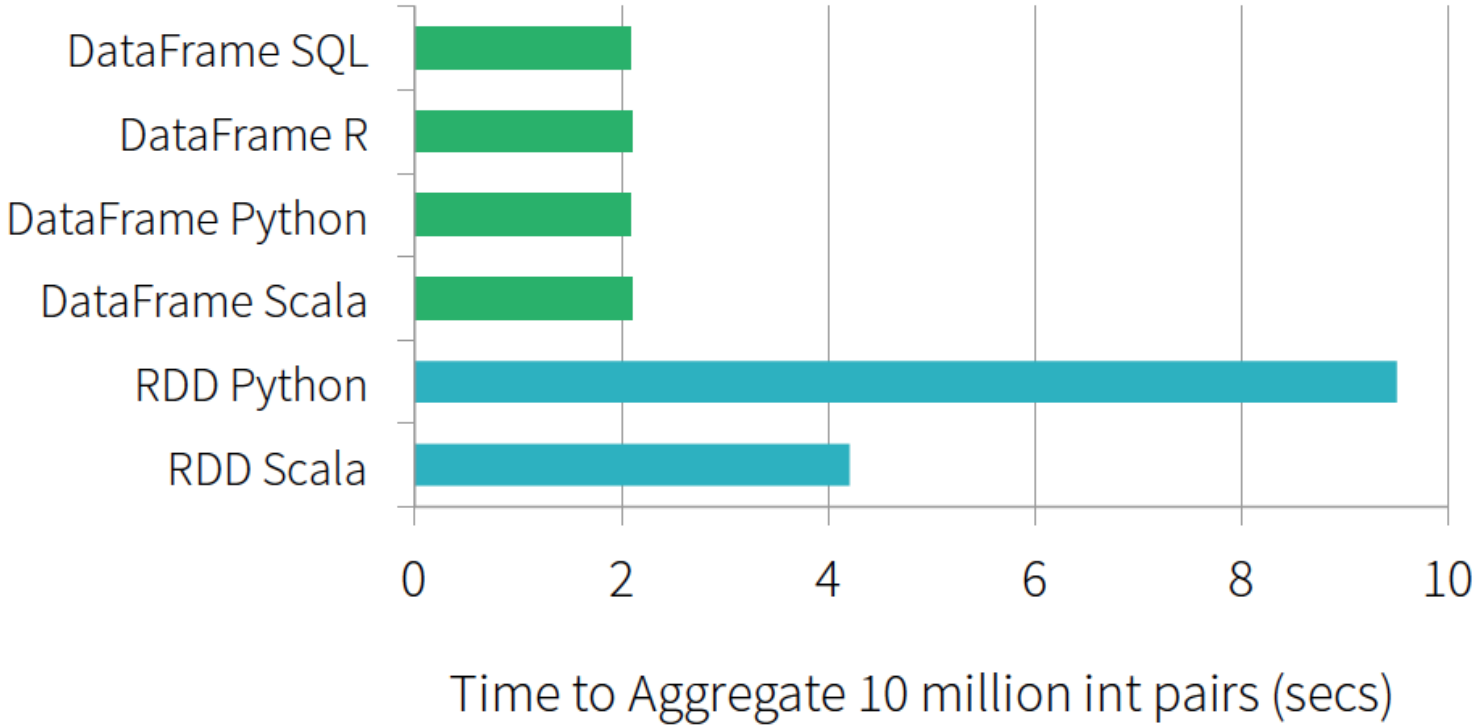
```
sqlCtx.table("people") \
    .groupBy("name") \
    .agg("name", avg("age")) \
    .collect()
```

Full API Docs

- Python
- Scala
- Java
- R

Data Application Lab

Not just less Code: Faster



Data Application Lab

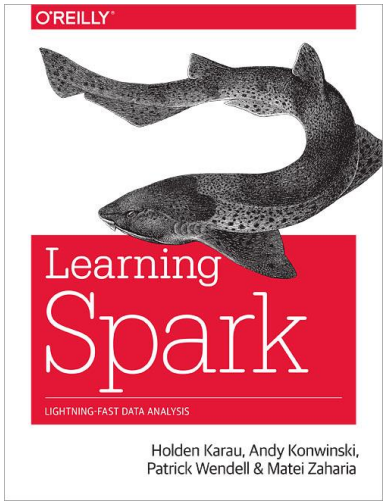
Case Study: Data processing and Analytics

* Analyze Fantasy Sports –NBA insights

Data Application Lab

Recommended Reading

* http://spark.apache.org/



Data Application Lab