Big Data analytics tools

Week 1
Day 1
Apache Spark SQL

YORK
UNIVERSITÉ
UNIVERSITY

What is Apache Spark

What is Apache Spark

- Apache Spark is a unified computing engine and a set of libraries for parallel data processing on computer clusters.
- The computing engine allows you the process any size dataset distributed or non-distributed

Unified Computing Engine

- This means that you can use spark to perform a wide range of analytic task using the same APIs.
  - Data loading
  - SQL
  - Machine learning
  - Data Streaming
  - Batch processing

Spark Components

The following diagram shows the major components of the spark framework

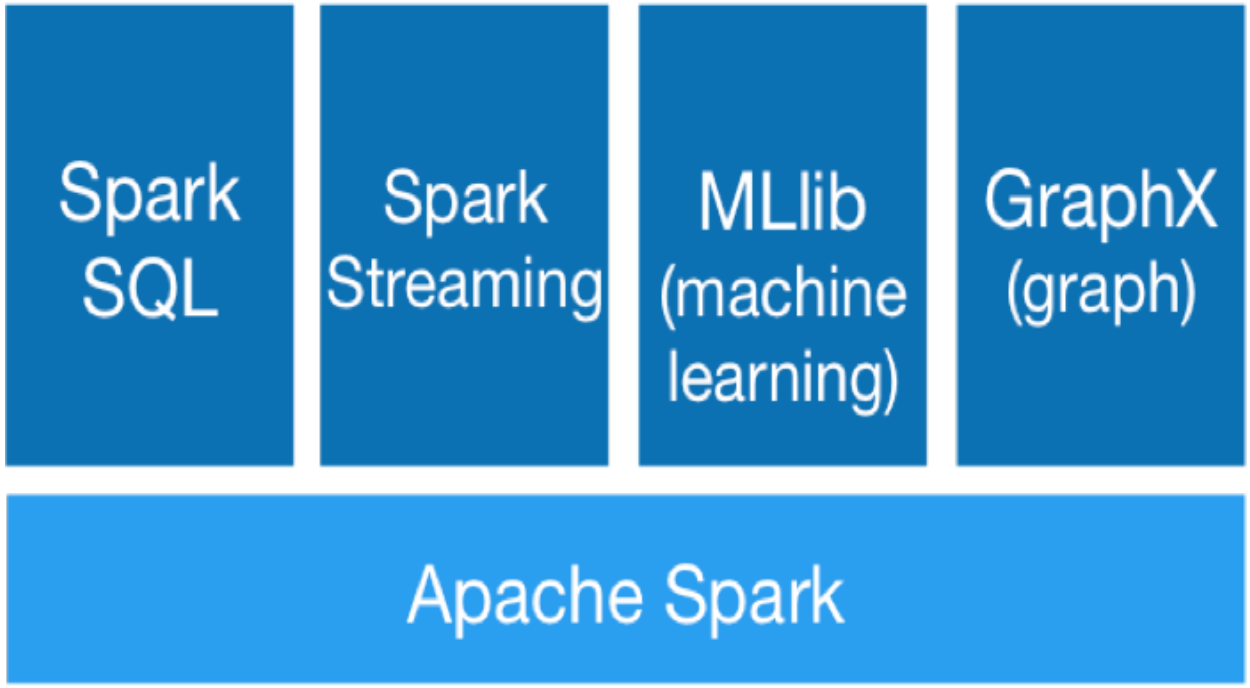

Image source: spark definite guide

Spark SQL

- Spark SQL is used for processing with structured data
- It provides Uniform Data Access to various data sources for example Hive, Avro, Parquet, ORC, JSON, and JDBC.
- It allows you to use ODBC or JDBC to connect to data stored in relational databases or data store in hadoop

MLlib

- MLlib is Apache Spark's scalable machine learning library
- Runs on existing Hadoop clusters and data.
- It can also run standalone, in the cloud or on HDFS.

SQL

What Is SQL?

- SQL or Structured Query Language is a domain-specific language for expressing relational operations over data. It is used in all relational databases, and many "NoSQL" databases.
- Spark implements a subset of the ANSI SQL:2003 Standard.
- Spark SQL supports both ANSI-SQL as well as Hive QL queries.

Spark Dataframes and Datasets

- Dataframes and Datasets are (distributed) table-like collections with well-defined rows and columns. Each column must have the same number of rows as all the other columns
- You can think of spark Dataframes and datasets as being equivalent to tables in a relational database

Schemas

- A schema defines the column names and types of a Dataframe. Schemas can be specified manually or be inferred from a data source .
- Schemas are used to specify the type of data that is stored in the columns of the Dataframe i.e. Integer, String, Date etc.

Working with Structured data

- You can create a Dataframe by using the following snippet of code:

```
val events_df =
spark.read.format("csv").option("delimiter", "\t").
.load("/gdelt/data/raw/events.csv")
```

- You can view the schema of this Dataframe by using :

```
events_df.printSchema()
```

- This will show the name and types of columns in the Dataframe

DataFrame Transformations

- In order for Dataframes to be useful they must all transformations. The main dataframe transformations are:
  - Creating Dataframes
  - Removing columns or Rows
  - Add rows or columns
  - Sort data by values in rows

Creating Dataframes

- The code listing below shows how to create a dataframe when working with csv files.

```
val events_df = spark.read.format("csv")
.option("delimiter", "\t")
.load("/gdelt/data/raw/events/events.csv")

events_df.createOrReplaceTempView("events_table")
```

- You can also create Dataframes from jdbc, odbc and many other data sources.

Removing Columns and Rows

- It is not possible to remove a row or column from a Dataframe because Dataframes are immutable i.e. they can be changed after they are created.
- However you can use select queries to create a new Dataframe without the row or column you which to remove. This is shown below

```
new_df = events_df.select("DEST_COUNTRY_NAME",
"ORIGIN_COUNTRY_NAME")
```

Removing Columns

- Rows are removed by using where clause or by filtering. The code listing below shows how this is done:

```
new_df = events_df.where(col("count") <
2).where(col("ORIGIN_COUNTRY_NAME") =!= "Croatia")
.show(2)
```

Adding Columns

- You can add columns to and existing Dataframe using the command below. This will add a new column called numberOne to the events_df dataframe.

```
new_df = events_df.withColumn("numberOne", lit(1))
```

Adding Columns

- You can add columns to and existing Dataframe using the command below. This will add a new column called numberOne to the events_df dataframe.

```
new_df = events_df.withColumn("numberOne", lit(1))
```