# National University of Computer and Emerging Sciences

## Laboratory Manual-08

*for*

## Fundamentals of Big Data Lab

Course Instructor: Isbah

Lab Instructors: Rida Mahmood, Mr. Raja Muzammil

Semester: Spring 2024

## Department of Computer Science

FAST-NU, Lahore, Pakistan

## Big Data processing systems

**Hadoop/MapReduce**:

Scalable and fault tolerant framework written in Java

Open source

Batch processing

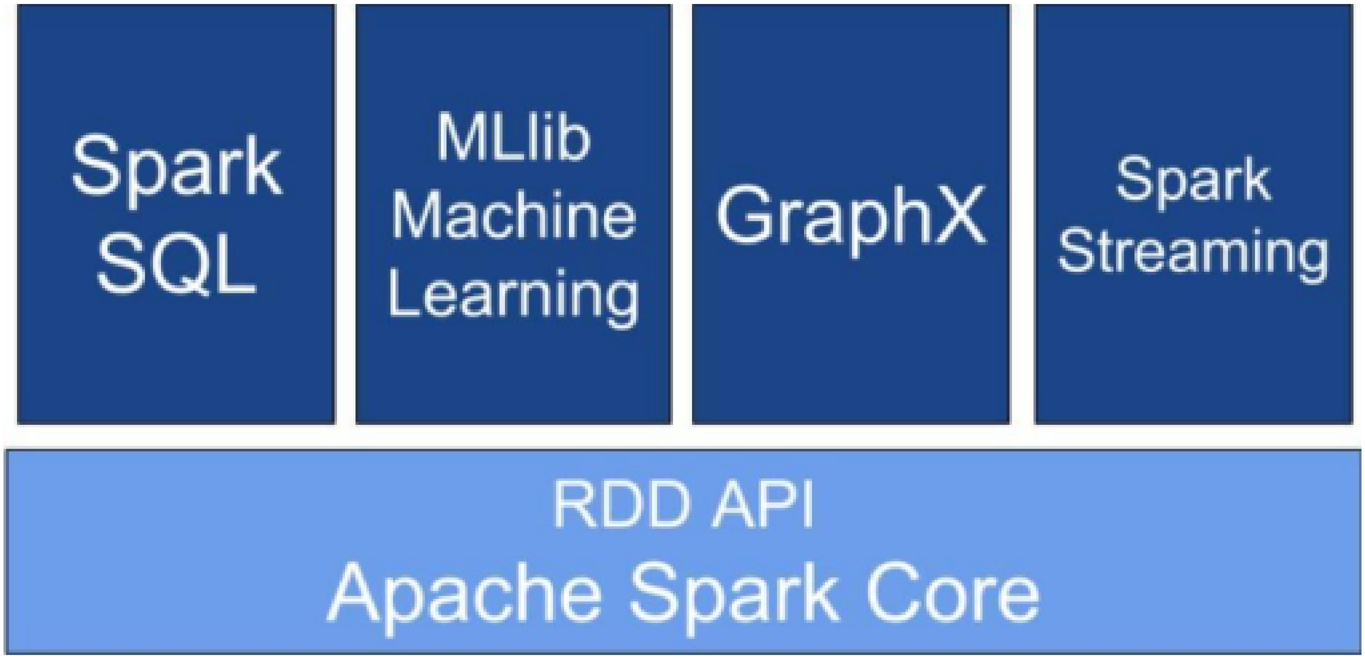
**Apache Spark:**

General purpose and lightning fast cluster computing system

Open source

Both batch and real-time data processing

## Apache Spark Components



## Spark modes of deployment

**Local mode:** Single machine such as your laptop.

Local model convenient for testing, debugging and demonstration

Cluster mode: Set of pre-defined machines

Good for production

**Overview of PySpark**

Apache Spark is written in Scala

To support Python with Spark, Apache Spark Community released PySpark

Similar computation speed and power as Scala

PySpark APIs are similar to Pandas and Scikit-learn

PySpark Documentation Link : https://spark.apache.org/docs/3.3.2/
Pyspark RDD Documentation Link: https://spark.apache.org/docs/latest/rdd-programming-guide.html

**Note:** Use google colab or jupyter notebook for PySpark

## Configuration of PySpark in System

Install pyspark using the line: !pip install pyspark

Import the following library:

```
from pyspark import SparkContext, SparkConf
```

Configure the PySaprk and start the session:

```
conf = SparkConf().setAppName(appName).setMaster(master)
sc = SparkContext(conf=conf)
```

where appName is your name of your project/lab and "local[*]" is your master if you are working locally.

## Understanding SparkContext

A SparkContext represents the entry point to Spark functionality. It's like a key to your car. When we run any Spark application, a driver program starts, which has the main function and your SparkContext gets initiated here.

## LAB TASKS

## Practice - Spark Data Frames

***SPARK DATAFRAMES***

You are provided dataset "Movies.csv" that contains information about 1600 movies with properties such as year, length, main actor and actress, director and popularity.

***Load the given dataset into Spark Data-Frames and answer the following queries using Data Frame functions only. You are not allowed to write the SparkSQL queries.***

1. Find the title, year, and director of action films that won an award.
2. For each award-winning actor, find the movies he acted it. Print the names of the movies and the director of the movie.

3. Find the top 10 most popular movies that did not win an award.
4. Find the 10 least popular movies that were released before 1980.
5. Find the average length of the movies of each genre.
6. Find the actor and actress pair who has acted in more than three Comedies together.
7. Find the names of actors who acted in movies of both 'Comedy' **and** 'Drama' Genre.
8. Find the names of actors who acted in movies of both 'Comedy' or 'Drama' Genre.
9. Find the names of actors who did not act in any 'Comedy'.
10. Find each actor, find the mean, max, and min ranking of his movies.
11. List the number of movies released in each decade starting from the 1960's.
12. Find the number of movies released each year.
13. Find the number of movies released in each year of each genre. Consider only the movies with a length greater than 100 minutes.
14. Sort the movie's release before 1990 by the title.
15. Find the movies with long titles. A movie title is considered long if it is greater than 50 alphabets.