**PySpark Assessment 2 :** - Abhiram Basa

****PySpark**** is an Apache Spark library written in Python.

* PySpark is a Python API which is an analytical processing engine for large-scale powerful distributed data processing
* ****Apache Spark is an open-source**** unified analytics engine used for large-scale data processing.
* PySpark is used to its efficient processing of large datasets.
* Pyspark can be used in pycharm, jupyter notebook and anaconda idle.

**Apache Spark:**

* Apache Spark is an open-source unified analytics engine for large-scale data processing.
* ApacheSpark is general purpose cluster computing system.
* It provides high-level API in Java, Scala, Python, and R.

It uses HDFS and Map reduce techniques.

HDFS: Hadoop Distributed File System.

Map reduce: Even big data works on Map reduce. It consists of key-value pairing .

It organizes the data in the form of key\_value pairs.

Map reduce is one of the efficient way of data engineering.

**Spark Architecture:**

* The Spark follows the master-slave architecture. Its cluster consists of a single master and multiple slaves.
* The Spark architecture depends upon two abstractions:

**1.** **Resilient Distributed Dataset (RDD)**

**2. Directed Acyclic Graph (DAG)**

**Resilient Distributed Datasets (RDD):**

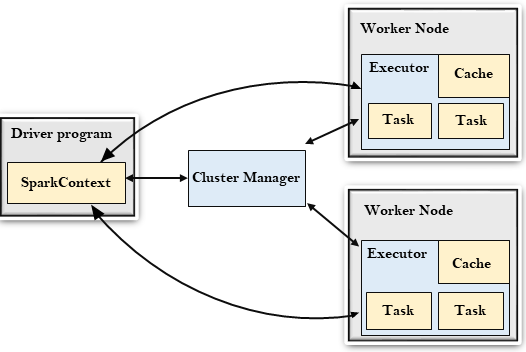
The Resilient Distributed Datasets are the group of data items that can be stored in-memory on worker nodes. Here,

* Resilient: Restore the data on failure.
* Distributed: Data is distributed among different nodes.
* Data set: Group of data.

## Directed Acyclic Graph (DAG):

Directed Acyclic Graph is a finite direct graph that performs a sequence of computations on data. Each node is an RDD partition, and the edge is a transformation on top of data.

**Spark architecture:**



## Driver Program:

The Driver Program is a process that runs the main() function of the application and creates the **SparkContext** object. The purpose of **SparkContext** is to coordinate the spark applications, running as independent sets of processes on a cluster.

To run on a cluster, the **SparkContext** connects to a different type of cluster managers and then perform the following tasks: -

* It acquires executors on nodes in the cluster.
* Then, it sends your application code to the executors. Here, the application code can be defined by JAR or Python files passed to the SparkContext.
* At last, the SparkContext sends tasks to the executors to run.

## Cluster Manager:

* The role of the cluster manager is to allocate resources across applications. The Spark is capable enough of running on a large number of clusters.
* It consists of various types of cluster managers such as Hadoop YARN, Apache Mesos and Standalone Scheduler.
* Here, the Standalone Scheduler is a standalone spark cluster manager that facilitates to install Spark on an empty set of machines.

### **Worker Node:**

* The worker node is a slave node
* Its role is to run the application code in the cluster.

### **Executor:**

* An executor is a process launched for an application on a worker node.
* It runs tasks and keeps data in memory or disk storage across them.
* It read and write data to the external sources.
* Every application contains its executor.

### **Task:**

* A unit of work that will be sent to one executor.

Spark code in Command Prompt:.

**SparkSession.builder.getOrCreate() :** It initializes the spark session of if note present it will create one.

**Obj.show():** It will print the output to the screen.

