ASSIGNMENT(22-12-2023)

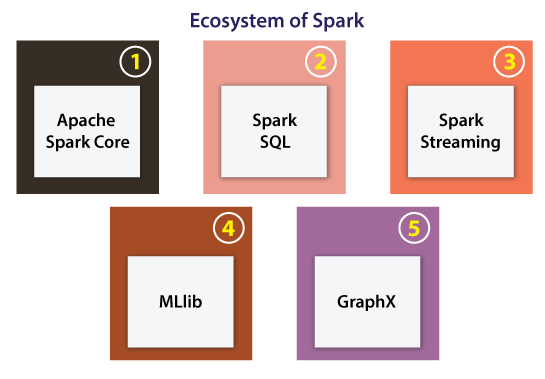
APACHE SPARK

🡪Apache Spark is an open-source, distributed processing system used for big data workloads.

🡪A key advantage of Apache Spark is its speed and performance capabilities.

🡪It is much faster as compared to the previous concepts to implement with Big Data such as classical **MapReduce**.

ECOSYSTEM OF SPARK



Apache Spark Core:

* Apache Spark Core can be defined as an underlying normal execution engine for the platform of Spark.
* It facilitates referencing data sets and in-memory computing within the external storage structures.

Spark SQL:

🡪This component is a module of Apache Spark for operating with many kinds of structured data.

🡪 Various interfaces provided by Spark SQL facilitates Spark along with a lot of information regarding both the computation and data being implemented.

Spark Streaming:

🡪Spark streaming permits Spark for processing streaming data in **real-time**.

Machine Learning Library (MLlib) :

🡪 Apache Spark is armed with a prosperous library called MLlib.

🡪The MLlib includes a wide range of machine learning algorithms collaborative filtering, clustering, regression, and classifications.

GraphX :

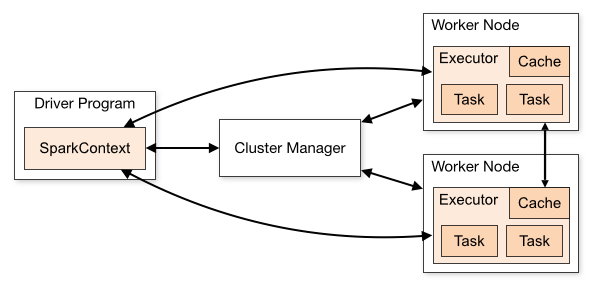
* Apache Spark comes using a library for manipulating graph databases and implement computations known as GraphX.
* This component unifies Extract, Transform, and Load (ETL) process, constant graph computation, and exploratory analysis in an individual system.

RESILIENT DISTRIBUTED DATASET :

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.
* Dataset: Group of data.

PYSPARK 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.

FEATURES OF SPARK:

* Fast Processing
* Flexibity
* In-memory Computing
* Real-time Processing
* Better Analytics