Apache Spark

Introduction:-

First

Distributed Processing :- It involves the use of multiple computing resources such as computers or servers to solve a single problem. Basically, multiple machines can handle large datasets.

where as

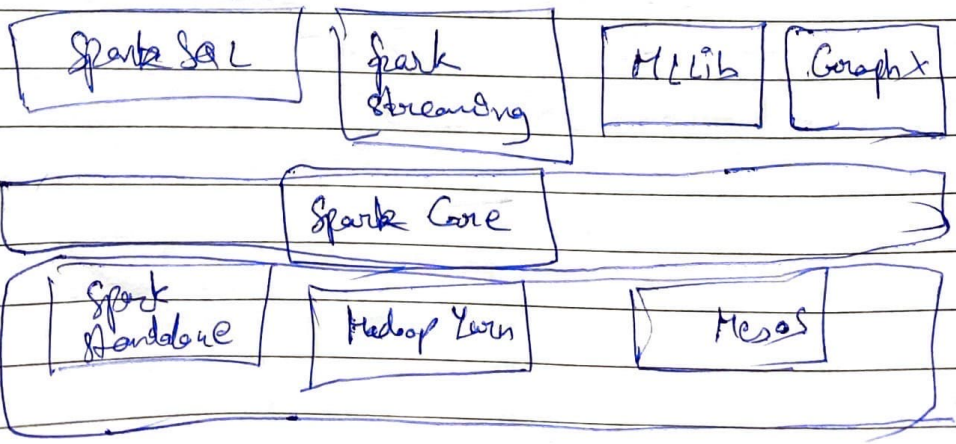
Parallel Processing :- In this one big task is divided into smaller subtasks that can be executed simultaneously.

Spark Components :-

Spark features :-

1. It is written in Scala prog. language and runs in JVM
2. It uses API's like: Scala, Java, Python, R.
3. Interactive shell available for Spark :- Scala and Python.
4. Data Sources :- SQL, NoSQL, HDFS etc.
5. It is good for ML algorithms.

Spark Components



classmate
Date
Page

1) What is Spark :-

- It is basically an open-source, distributed computing system that provides a fast and general-purpose cluster-computing framework for big data processing.
- mainly used for machine learning, graph, processing and large-scale data processing.

- Spark Core key features :-
  - Incharge of essential I/O functionalities.
  - Important in observing the role of the spark cluster
  - Task dispatching
  - Fault Recovery
  - It overcomes the lag of mapReduce by using in-memo

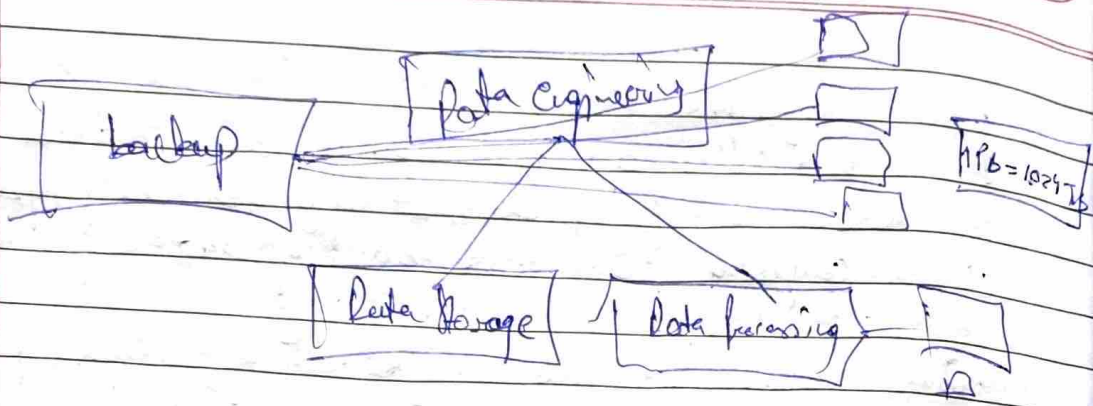
- Spark Core is embedded with a special collection called RDD (Resilient distributed datasets).

RDDs basically handles data partitioning across all the clusters.

2 operations performed on RDDs :-

1) Transformation

2) Action



2) Apache Spark SQL

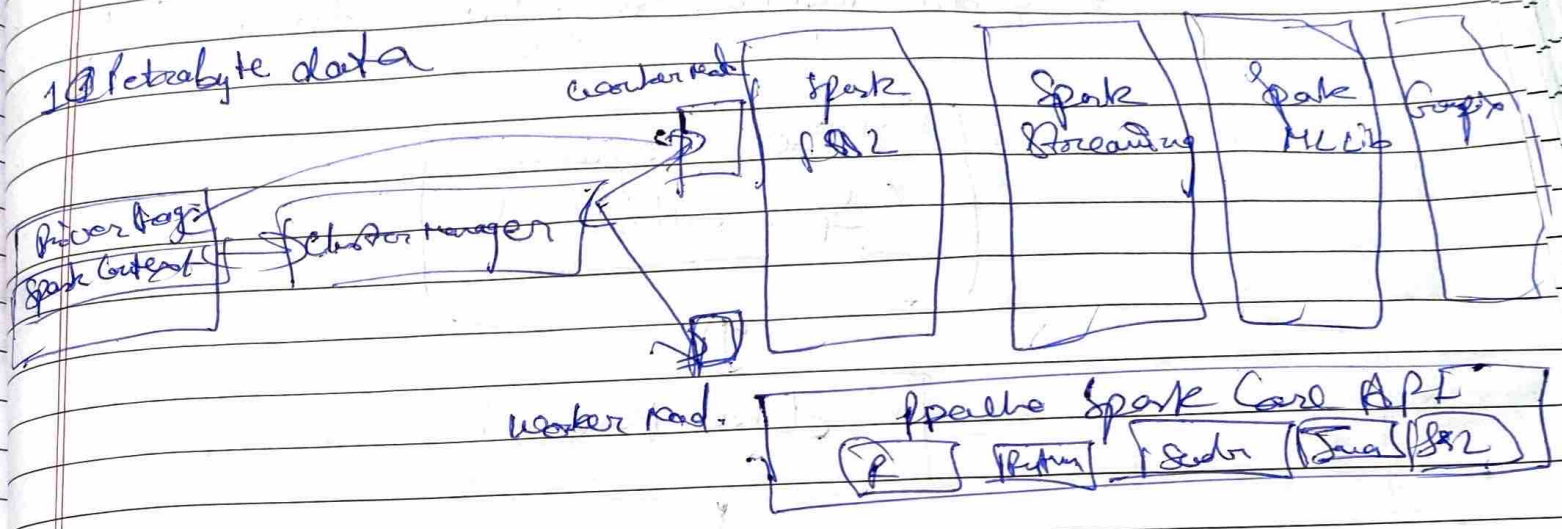
- Spark SQL is a distributed framework for structured data processing.
- It uses same execution engine while computing an output. It does not depend on API/language to express the computation.
- It work to access structured and semi-structured information.

Features :-

1. Cost based optimizer
2. Full compatibility with existing HIVE data.
3. Data frames and SQL provide a common way to access a variety of data sources.

HIVE ⇒ Data warehousing and SQL like query lang. system built on Hadoop.

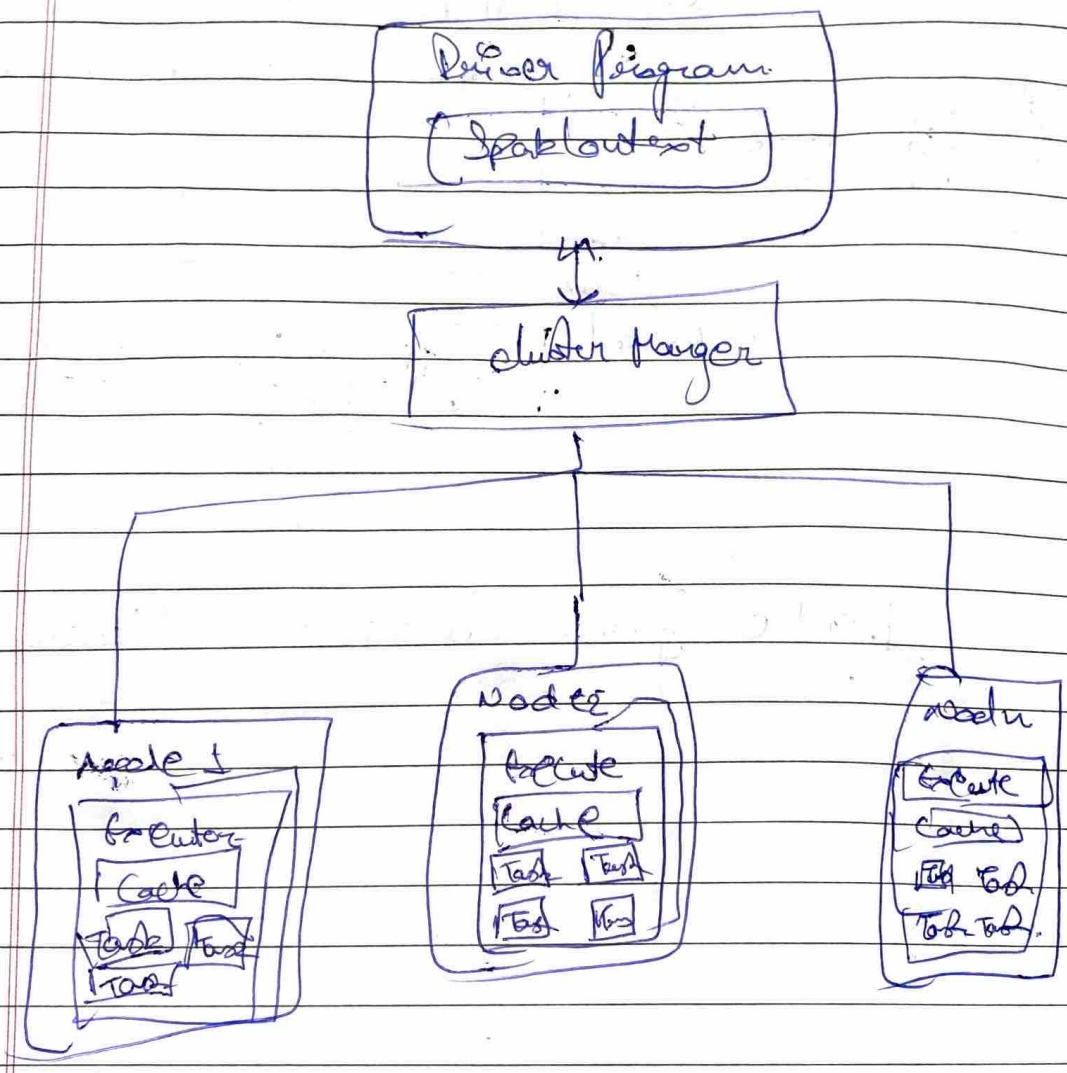
Apache Spark Ecosystem



Apache Spark Cluster Architecture.

=> Apache Spark Streaming -

Cluster -> Group of Machines
=> Can create in databrick or spark standalone
=> work together to perform distributed processing



Spark application will load on cluster, so it becomes spark cluster.

classmate
Date
Page

like
Streaming occurs on platforms like netflix, youtube, hotstar etc.

2 types of data :- Streaming - live data
Storage - already stored.

=> Apache Spark Streaming.

- It is an add onto core spark API, which allows scalable, high throughput, fault tolerant stream processing of live data streams. Spark can access data from sources like netflix, youtube etc.

=> Working of spark Streaming.

1) Gathering :- Gathering of data from diff sources like kafka, flume, kinesis, file systems and socket connections etc.

2) Processing :- Gathered data processed using complex algorithms.

3) Data Storage :- Processed data is pushed out to file systems, databases and live dashboards.

~~Other~~ DStream :- It signifies continuous flow of data. It is basically a sequence of RDDs.

⇒ Apache Spark MLlib

- It is a scalable ML library that discusses high quality algos. and high speed.
- Purpose for MLlib Creation is to make ML scalable and easy.
- Helps in implementation of clustering, regression, classification and collaborative filtering.

⇒ Core Concepts

① Job :- Peice of code which takes I/P from HDFS or local, performs some computation on the data.

② Stages :- Jobs are divided into two stages :- Map & Reduce stages.

③ Tasks :- Each Stage has some task one task per partition.

④ DAG :- Directed Acyclic Graph.

⑤ Executor :- Process responsible for executing a Task.

⑥ Master :- The machine on which the driver prog. runs.

⑦ Slave - Machine on which Executor prog. runs.

(1) Spark Context :- Represents connection to a spark cluster and can be used to create RDDs, etc

(2) DAG Scheduler :- Computes a DAG of stages for each jobs and submits them to Task Scheduler determines preffered location for tasks

(3) DAG - Directed ~~Analysis~~ Acyclic Graph.
Collection Tasks together.

(4) Task Scheduler :- Responsible for sending tasks to cluster, running them and retrying in case of failure.

(5) Scheduler Backend - Backend interface for scheduling system that allows ~~all~~ plugging in different implementations (Mesos, YARN, Local).

d) How Spark works?

1. Every layer in spark interpreters Spark uses Scala interpreter.
2. Code entered in spark console creates a operator graph.
3. DAG scheduler divides operator graph into map and reduce stages.
4. Each stage is comprised of tasks based on partition of i/p data.
5. DAG scheduler pipelines operator together to optimize the graph.
6. Finally the DAG scheduler passes the stages to the task scheduler which is launched via cluster manager. (Spark Standalone/ Yarn/Mesos)