

What is Spark (It is a framework)

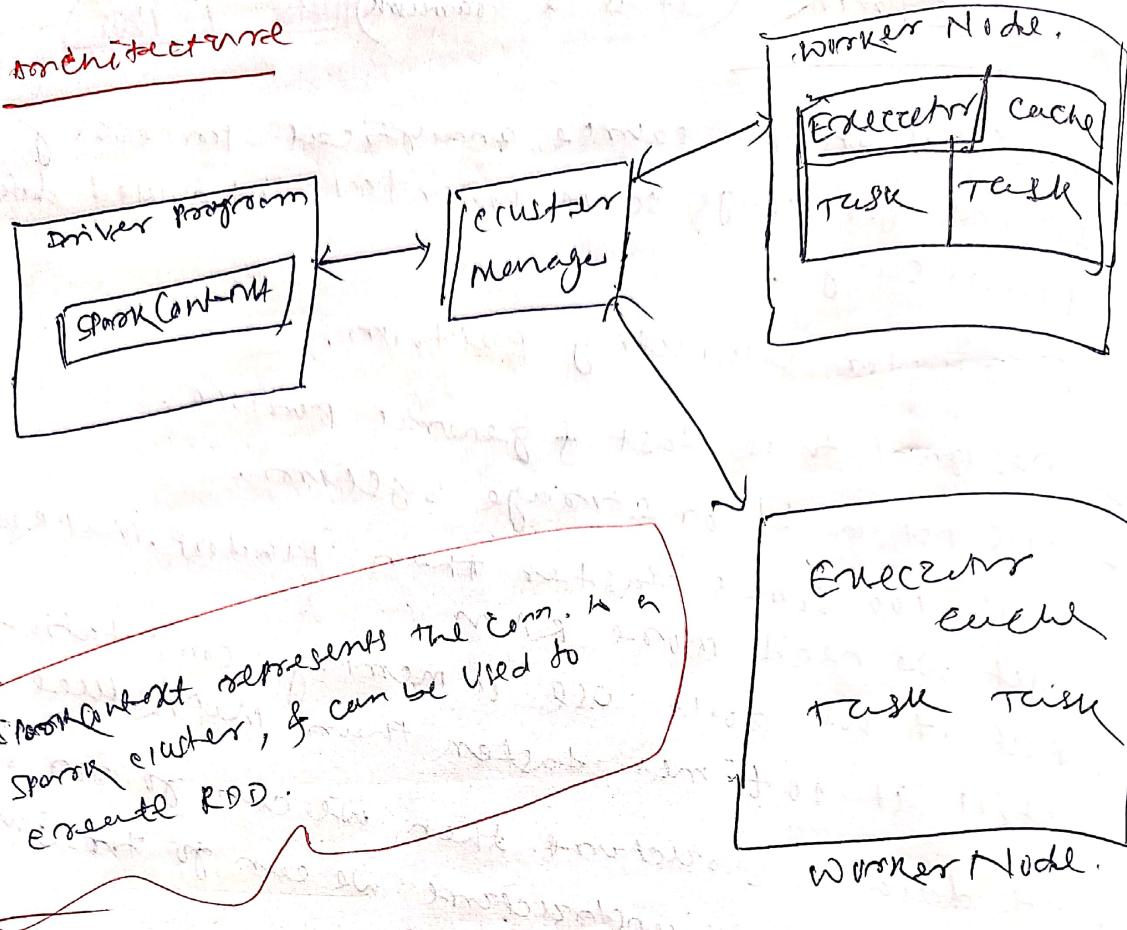
Mangal

Date: 1/12/21

- spark is an open source analytical processing engine for large scale powerful distributed data processing.
- A cluster computing platform.
- Designed to be fast & general purpose.
- It's not a db or storage system.
- It is 100 times faster than Hadoop: MapReduce.
 - But we need more RAM.
 - But if we do it all in memory computation.
 - still it 10 times faster than MapReduce.
- If data is structured then we can go for SQL.
- If data is unstructured we can go for java, python, Scala, spark.
- We can run spark without Hadoop.

Features of spark

- In memory computation
- Distributed processing using parallelize
- Can be used with multiple cluster manager (spark, yarn, mesos etc) , Kubernetes
- Fault tolerance
- Immutable
- Lazy evaluation
- cache & partition
- Inbuilt optimization when using DF
- Support ANSI SQL .

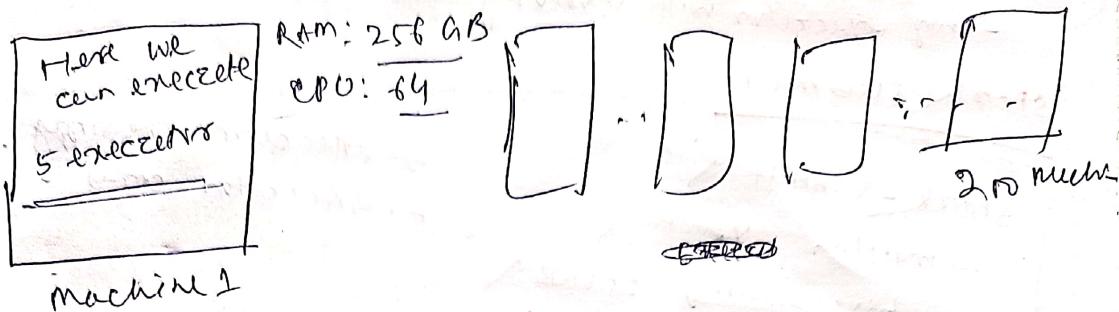
Architectural

- Spark API runs on independent sets of processes of a cluster.
- Spark Context object ~~is~~ the main program (called driver program).
- Job will execute with the help of Executors.
- In MapReduce in every machine map will be with the help of Job Tracker to process the data.
- But in spark Let's say 2 executors are there. Then some of the data can locally available or some of the data can remotly.

Executor = logical unit created out of certain amount of memory & certain amount of cores of each machine (RAM).

Ex:- 10 executors run on 100 machine.

- we can run 2 executors on same machine.

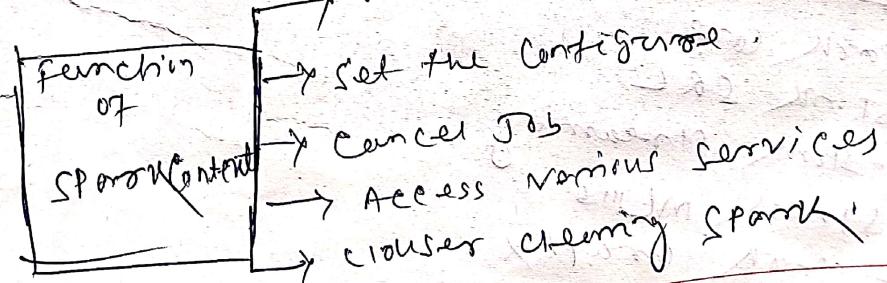


$$\text{RAM} = 256 \times 2^0 = 512 \text{ GB}$$

$$\text{CPU} = 64 \times 2^0 = 128$$

Each executor will execute where it is decided by the cluster manager.

- A Spark "driver" is an API that creates a spark context for executing 1 or more jobs in the spark cluster.



Specifically, to run on a cluster, the spark context can communicate to several type of cluster manager.

- once connected, spark connects to the nodes in cluster.
- next, it sends your API code to the executors. (job file goes to every executor).

- Finally spark context sends tasks to the executors to run.
- only executor will involving the processing of data.

spark submit

spark-submit >

--master yarn >

--driver-memory 256MB >

--jar mysql-connector.jar >

--class demo.ReadPartitions >

/JdbcAPItest.jar

--executor-memory

--executor-cores

History Server

http://localhost:1080

version	App-Id	App Name	start Date	Completed At	Duration
---------	--------	----------	------------	--------------	----------

Spark Module

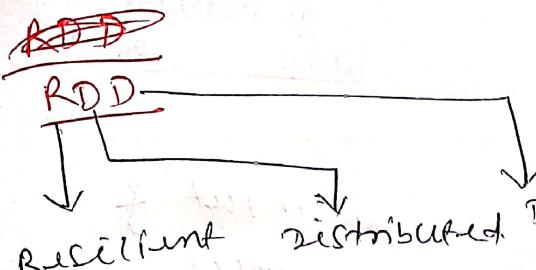
- Spark Core

- Spark SQL

- Spark Streaming

- Spark MLlib

- Spark GraphX



- RDD is immutable.

Resilient

Fault-tolerance with the help of RDD lineage graph (DAG) & so able to ~~recompute~~ recompute missing or damaged partitions due to node failure.

Distributed

- Since data are stored in multiple machine nodes.

Dataset

Represents record of data you work with.
The user can load the dataset externally which can be either JSON, CSV, text file or database via JDBC with no specific datastructure.

- Spark can read the data from any distributed file system (HDFS or Amazon S3)
- RDD do 2 operations.
 1. Transformation
 2. Action

Transformations

- Func that take an RDD as an input & produce 1 or more RDD as the output.
- Transformation are lazy ops?

~~map(), flatmap()~~~~filter()~~~~map()~~~~flatmap()~~~~mapPartitions()~~~~filter()~~~~sample()~~~~union()~~Wide

intersection

distinct

reduceByKey

groupByKey

join

Cartesian

partition

Actions

- The action in Spark returns final result of RDD computation.
- Trigger execution using lineage graph to load the data into original RDD, carryout all the intermediate transformation & return final results to driver program to write it over the system.

`first()`, `take()`, `reduce()`, `collect()`,
`count()`

15-06-22

Streaming

Project Name = sample

myStream → DirectKafkaCount. Scala

Here we will create StreamContext object.

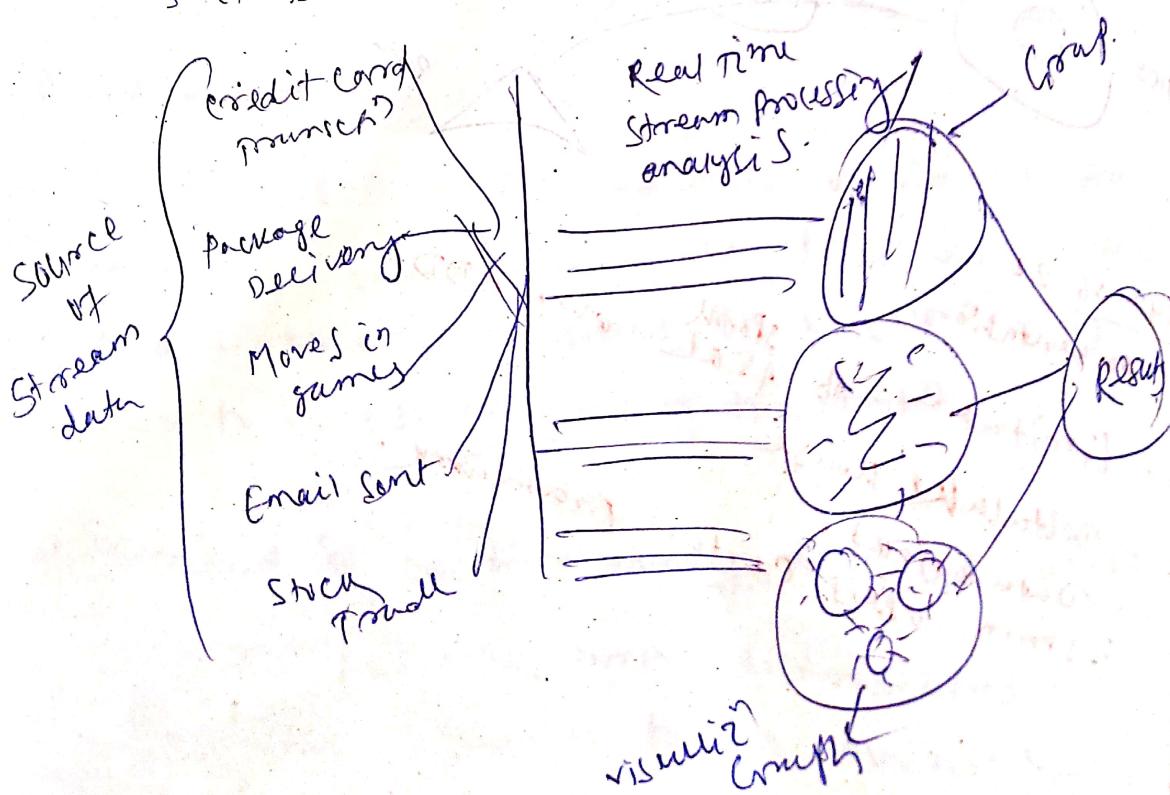
val ssc = new StreamContext(sparkConf, sec)

val KafkaConsumer

- The death is coming from Kafka Consumer.
- but it is not my stream.
- there may be fit of death can coming, but it is bad design.

What is streaming processing

- Stream processing refers to the ongoing processing of one or more event streams as it is produced or received.

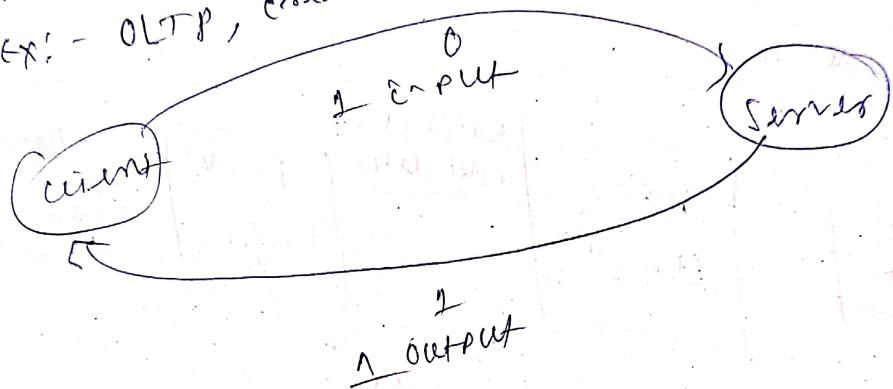


- There are 2 models are there.
- 1 - Request & Response
 - 2 - Batch

Request & Response

This mode of processing is usually blocking - an app sends a request & waits for the ~~response~~ processing system to respond.

Ex:- OLTP, credit card processing, time trace system



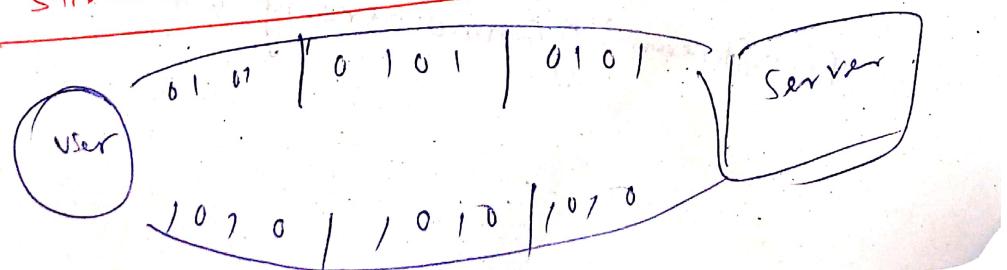
② Batch processing

In the database world, data is loaded in huge batches once a day, reports are generated, if user look at the same reports until the next data load occurs.

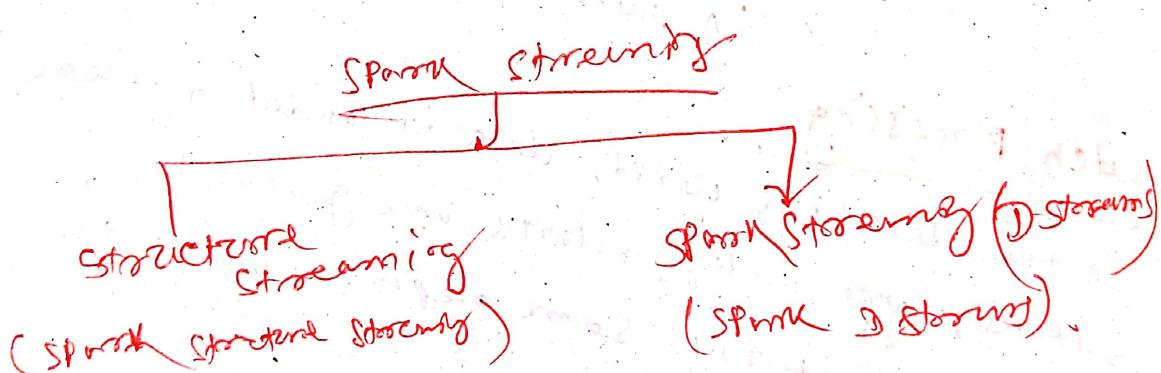
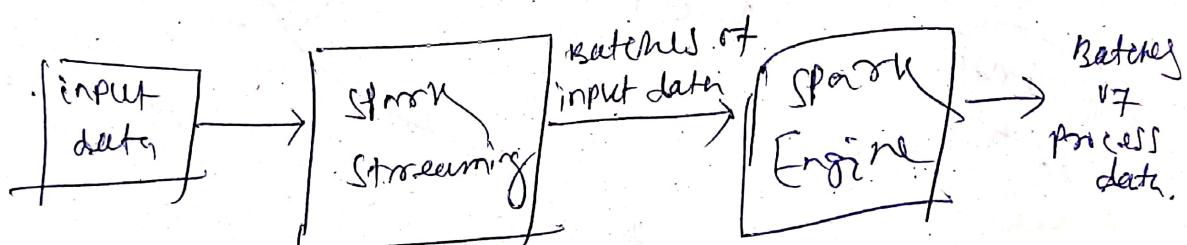
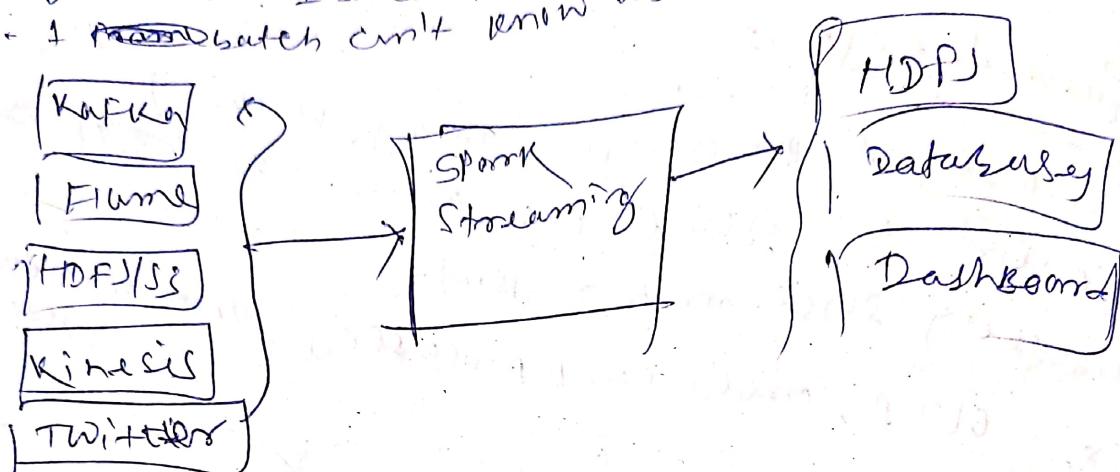
Here we get all the records at a time & process it & then i get response at a time

1MB or 1 GB or 1 TB

③ Stream processing



- If we give more record the server will take less record to process.
- Spark streaming is a problem like it can't go below 1 second.
- If ~~one~~ batch can't know about other batch.



- We can streaming files from a directory also.
- In that case we have to use Structured Streaming to monitor the folder.

Spark Streaming from directory - scalar.

- Here checkpoint is there ,
I batch can know the info about other batch.
- so we can modify the checkpoint path.
.option ("checkpoint", "tmp/dath")

Veracity

- It refers to the assurance of credibility/integrity/accuracy of the data.
- Since the data is collected from multiple sources, we need to check the data for accuracy before using it in business.

Value

- Just because we collected lots of data, it's not value unless we garner some insights out of it.
- Value refers to how useful the data in decision making.

DAG

Directed Acyclic Graph.

This is a logical flow of ops.

- This is a logical flow of ops?
- Plan gets created at the time of transform.
- Plan gets created at the time of action.

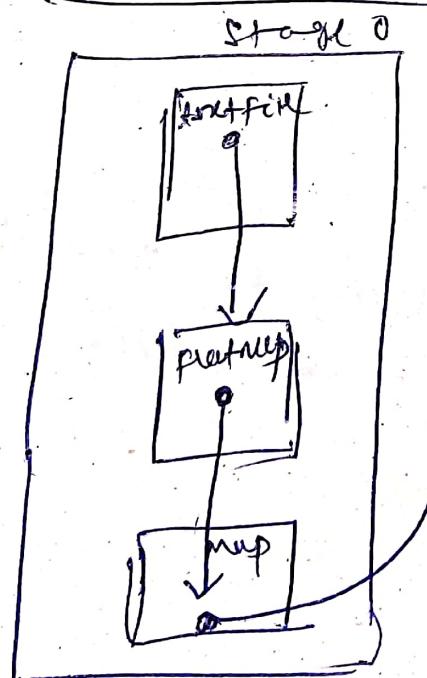
Why DAG

- most optimized plan
- minimize shuffling
- fault tolerance
- RDD recovering

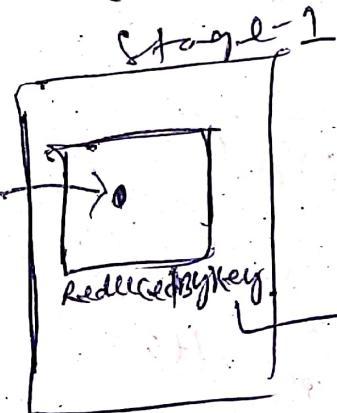
HOW DAGS WORK

- RDD have 2 partitions?
 - Very narrow →
 - wide →
- When an action is encountered
 - 1. Spark creates DAG & submits it to my DAG scheduler.
 - 2. DAG scheduler will split the DAG into multiple stages.
 - 3. These stages are created based on partition.
 - 4. Then DAG scheduler will then submit the stages into task scheduler.

Data visualization



Narrow partition



Wide partition

Jobs → Stages → Task