58369

Copy No.

Number of Book II nd used

TRIBHUVAN UNIVERSITY
INSTITUTE OF SCIENCE & TECHNOLOGY
EXAMINATION BOARD
KIRTIPUR

Code No.:

Book I

Code No.:

Students are required to write their answers on both sides & a margin of about 1¼ inches should be left on each page.

MARKS OBTAINED

1st Q		11th Q	
2nd Q		12th Q	
3rd Q		13th Q	
4th Q		14 th Q	
5th Q		15th Q	
6th Q		16th Q	
7th Q		17th Q	
8th Q		18th Q	
9th Q		19th Q	
10thQ		20th Q	

TOTAL 40

INSTRUCTIONS TO CANDIDATES

[1] Each Candidate will write legibly on the title page his or her ROLL NO. REGISTERED NO. AND THE SCRIPT in which answers are written but not his or her name and the name of the campus from which he or she appears. This should be done in each answer - book used before beginning to write inside.

[2] No loose papers will be provided for scribbling and no other paper should be brought in for this purpose. Any candidate found with loose paper in his or her possession WILL BE EXPELLED. All work must be done in the book provided and pages MUST NOT BE TORN OUT. The book provided CANNOT BE REPLACED BY ANOTHER but, if necessary, an additional book will be given. All work intended for the examiner must be written ON BOTH SIDES of the paper. Anything cancelled will not be looked into. Should a torn leaf be discovered inside an answer book, it should not be removed but crossed out and folded after bringing it to the notice of the invigilator.

[3] Candidate is forbidden to write answers or anything else on the question - paper.

[4] No Candidate will be allowed to leave the room until one hour has passed from the time when the papers are distributed.

[5] Candidate who uses two or more answer-books should see, before handing over to the invigilator, that they are properly stitched together.

INSTRUCTIONS TO THE EXAMINER

1. Aggregate marks of each question should be placed in the given appropriate box.
2. Marks for parts of a qusetion should be totalled under the question inside the answer- book.

Level MDS

Year/Part/Sem. I/II/III

Centre SMS TU

Roll No. 06

Roll No. In Words Six

Arpan Saphota

Reg. No. .....

Subject Techniques for Big Data

Paper .....

Script English

Date 2081/02/04

.....
Signature of Scrutiny Board

.....
Signature of Examiner

Group 'A'

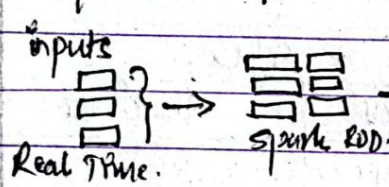
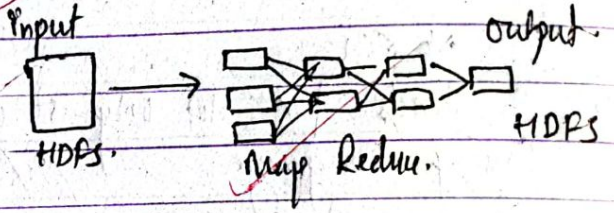
1. NoSQL stands for Not only SQL which means, SQL which can only works with relational and structured Language was the limitation overcomed by NoSQL. Unstructured, semi-structured data can be processed with NoSQL and it has high scalability.

In SQL the five V' are fixed and the complex relationship can be handled by the Joins, However those are not possible in NoSQL which are the main key difference between SQL and NoSQL. The other general differences are:

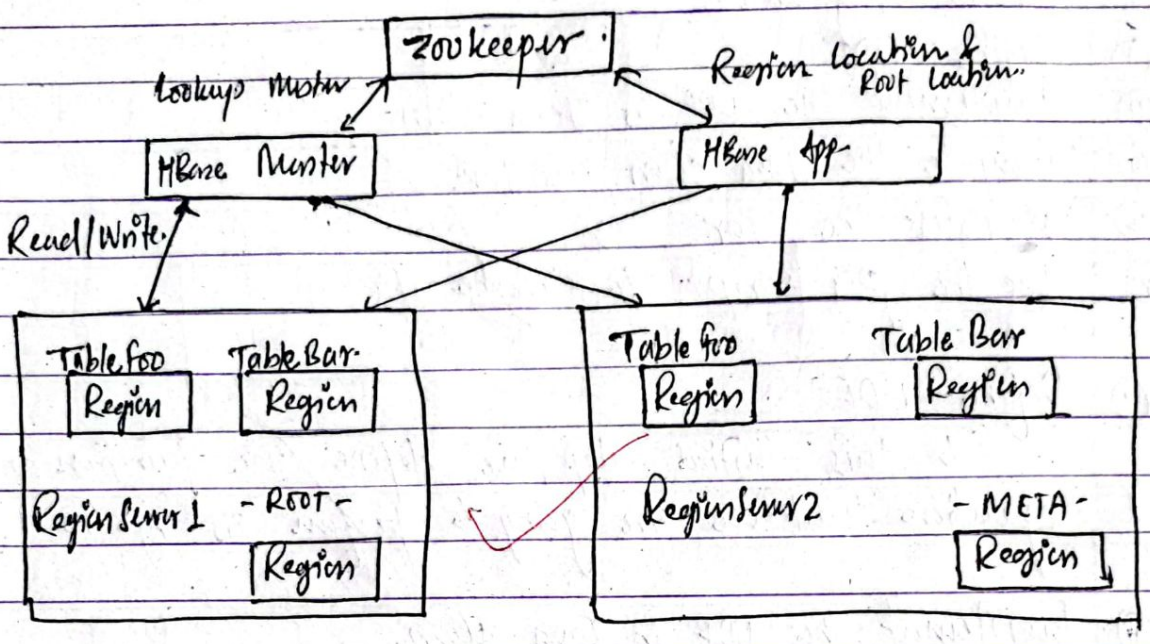
No SQL	SQL
(i) It stands for Not Only Structural Query Language.	It stands for Structural Query Language.
(ii) It works on non-relational databases.	It works only on the Relational databases.
(iii) NoSQL are highly scalable.	SQL are hard to scale.
(iv) Horizontal partitioning are done in NoSQL	Vertical partioning are done in SQL
(v) It works on Unstructured and semi-structured data	It works on structured data only.
(vi) Example: MongoDB, HBase, Cassandra, etc.	Example: Oracle, MySQL, Ms SQL, etc.

2. Apache Spark is different from Hadoop Mapreduce.

Apache Spark is the real time stream processing in big data technology where the real time data like: Live video streams, server logs, sensor data are processed in real time. Where as Hadoop Map reduce is the batch processing in big data where the stored data are processed in batches & jobs.

Apache Spark	Hadoop Map reduce.
(i) 	
(ii) It works on Real time data	It works on stored batches of data.
(iii) Apache spark processes data quickly with the help of RDD	Hadoop Mapreduce process data very slow.
(iv) It does catching in memory.	There is no catching in Hadoop mapreduce.
(v) RDD is the heart (core) component of Apache Spark.	Mapreduce framework is the core component of Hadoop.
(vi) No disk is involved here	Disk is involved here so it is slow.
(vii) Spark can be implemented in Python, Java &	It is implemented only in Java.

4. HBase Architecture.



2.5

In HBase Architecture zookeeper is the main component which acts as the master and look after the Region Servers. Here master assign job to the Region Servers. HBase is built on top of Hadoop Distributed File System (HDFS). It has following core components:

(i) Master
(ii) Region
(iii) Region Server

Master assign jobs to the slaves which are Regions in this case. Regions are the tables in the region servers which acts as slave. Master perform the read write operations on the Region Servers.

5. The process to create UDFs (User Defined functions) in Apache pig are:

(i) Define UDF
(ii) Implement the UDF in Java class.
(iii) Compile the Java class into Jar.
(iv) Register the Jar
(v) Use the pig script to execute it

(i) Define UDF

In this initial step, we define the function or the operation that we are going to perform in Apache Pig.

(ii) Implement the UDF in Java class.

We then implement the defined function into a Java class and methods.

(iii) Compile the Java class into Jar

The implemented Java program is compiled into the Jar file.

(iv) Register the Jar

Jar file is registered for the data flow operation in pig. Here, REGISTER, DEFINE, LOAD, and DUMP operation are performed.

(v) Use the pig script to execute.

The registered program is executed in the pig command-line interface.

3. The different execution modes of Apache Pig are:-

(i) Grunt shell Execution.
(ii) CLI Execution
(iii) UI Execution.

(i) Shell Execution

Pig script can be executed with its shell which is known as Grunt shell. We give commands to this shell and pig script is executed.

(ii) CLI Execution.

Pig script can be executed with Command Line Interface as shuttler to shell execution.

(iii) UI Execution.

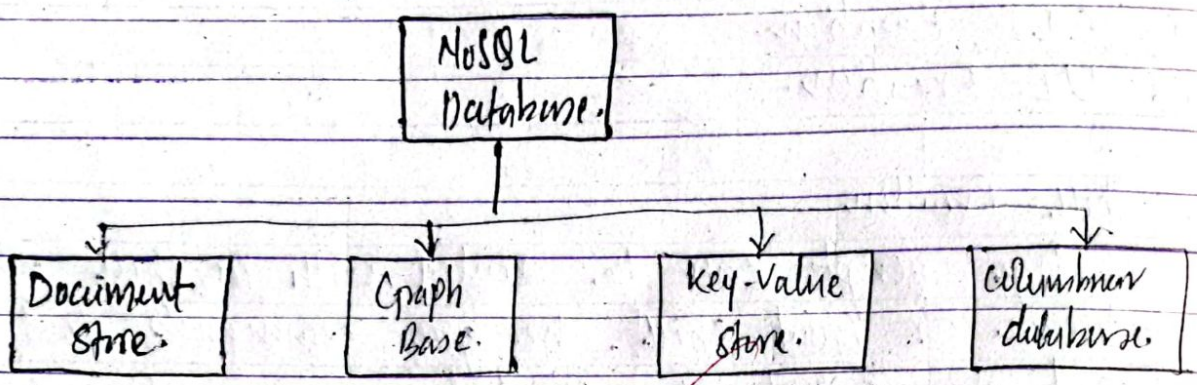
pig script can also be executed in User Interface mode where we give input script in the UI and get output in the UI.

In order to Execute the Pig we must have to follow the following patterns.

REGISTER → Register Java class
DEFINE → Define the pig class.
LOAD → Load the data.
DUMP. → Displaying the result.

Group 'B'

6. The different types of NoSQL databases are:-



(i) Document Store.

Document store are the those type of NoSQL database where the documents i.e. records are stored in a key value pair structure like in a JSON file.
MongoDB is the example of NoSQL Database.
Data stored in document store looks like:-

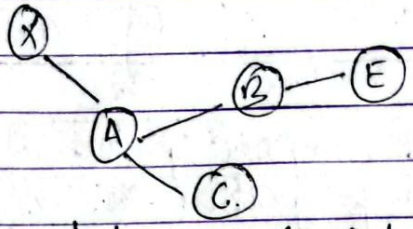
```
{ "Name" : "Arpan"
  "Address" : "Khumaltar"
  "website" : "www.arpansapkota.com.np"
}
```

MongoDB, CouchDB, etc are the Document Store NoSQL databases.

(ii). Graph Base.

Graph Base is the another type of NoSQL database which is used to store the data in the graphical structures. Data are stored on the vertices of the graph.

Example:-



Graph base NoSQL Database.

Neo4j is the example of Graph Base NoSQL database.

(iii) key-value store.

key-value store are the those type of NoSQL database where the data are stored on the basis of key and values on the table. key represents the unique identifier in this database and value is the actual data pointed by the key.

Example.

key	Value.
1	Ram
2	shyam
3	Ghanashyam.

5

MongoDB, cassandra, etc are the key-value store databases.

(iv) Columbnar Databases.

Columbnar database are those database where the data are stored on the column families with row key as key value.

Example:

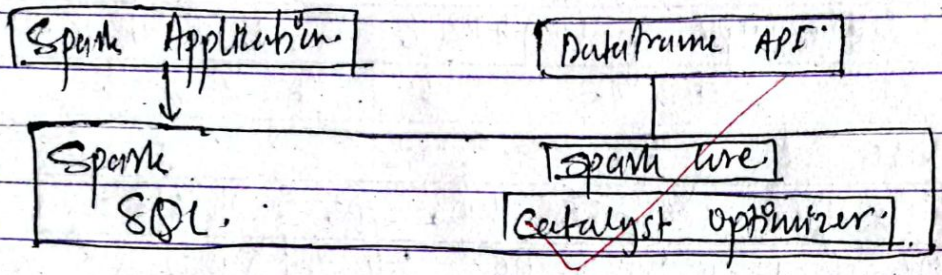
key ↓	column family - 1		column family - 2.	
	column 1	column 2	column 1.	
Row 1				3 cell.
Row 2				

Example: HBase is the columbnar Database.

7.

(a) Spark SQL.

Spark SQL is the one of the component of the Spark Ecosystem which is used to perform the query operation on the Spark.



In Spark SQL, we have spark Application which is the client side of the sql where client can perform the query operation with the spark Application which goes into the spark SQL Engine in the spark core.

Catalyst optimizer perform the optimization task on the query plan. The logical query are optimized into physical query with this optimizer.

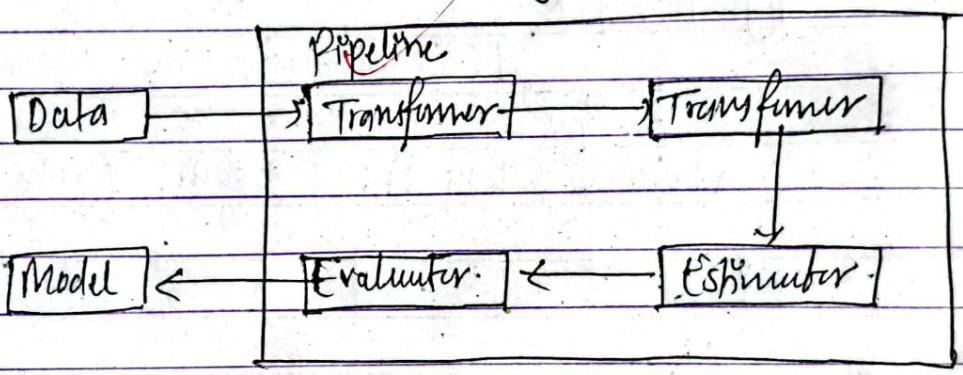
DataFrame API is an interface where the spark Query engine can be interegated into the different data sources like: HDFS, HBase, MongoDB, etc.

Spark Core is the SQL engine of the spark which perform the query operation as according to the input. In this way with the combined efforts of all component spark SQL is operated.

(5). Spark Mlib.

Spark Mlib is the another component of spark Ecosystem which is used to perform the Machine learning task in spark. It has growing set of Machine learning Algorithms like: Page Rank, Tringle count algorithms

Spark Mlib has the following pipeline



Data is provided as input to the transformer which clean and loads the data into spark and the feature extraction are done in another transformer. Once the features are extracted then, it is fitted in the Estimator, Those are then sends to the evaluator which evaluates the performance of the Machine learning algorithm and then finally it sends an output which is the Model.

We have various algorithms in spark Mlib and some are:-

(i) Page Rank
(ii) SVD++
(iii) Tringle counting. etc

10. db.order.find({})

Use order

(9) Maximum price per customer

```
db.order.aggregate([
  {
    $group: {
      _id: "$customer", price: "$price"
    },
    max price: {$max: {$sum: "$price"}}
  }.])
```

(1). Total revenue per day.

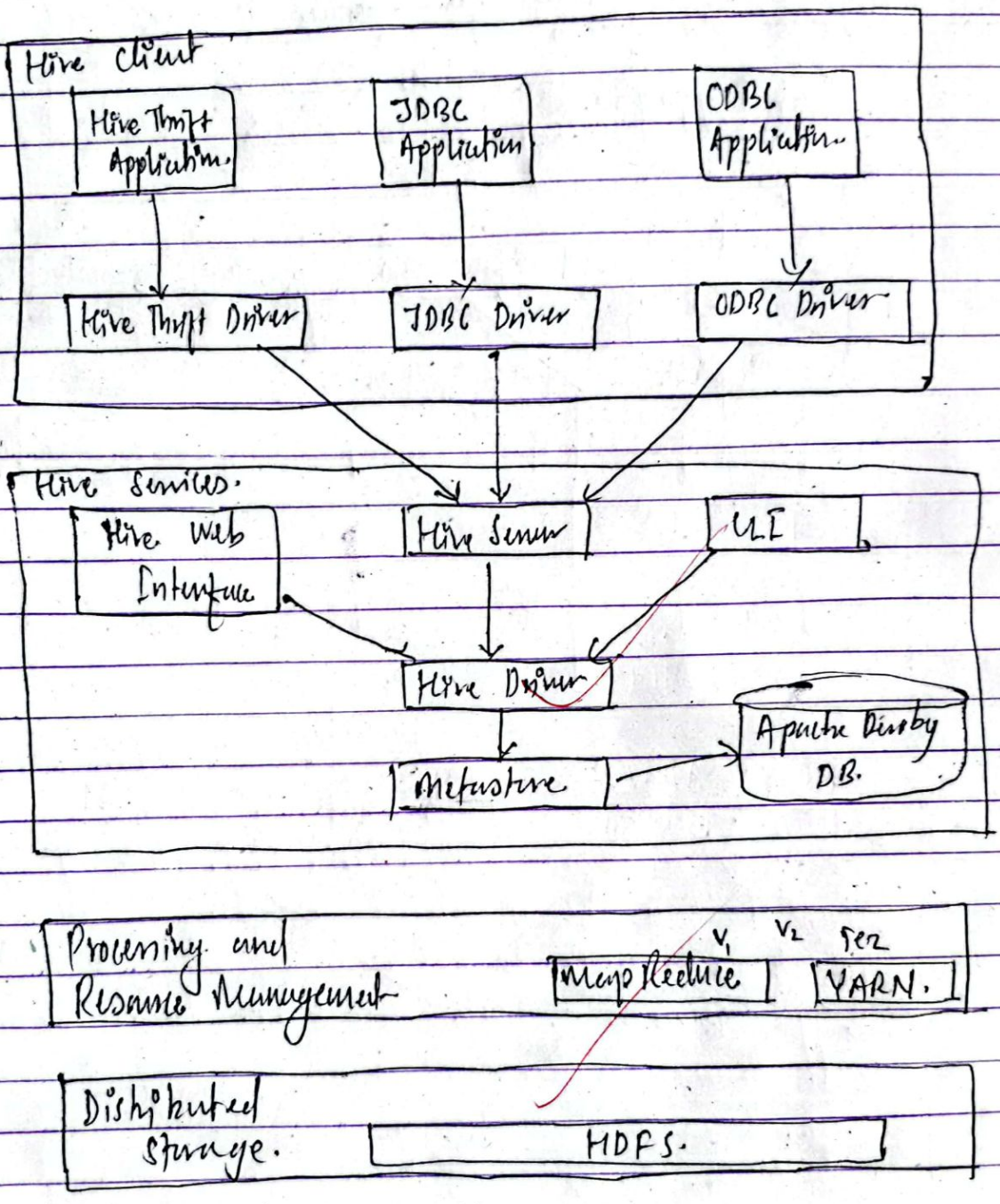
```
db.order.aggregate([
  {
    $group: {
      day: "$date", price: "$price"
    },
    TotalRevenue: {$sum: 1}.
  } ]).
```

@Adrack day

(c). Average price per customer per product.

```
db.order.aggregate([
  {
    $group:{
        _id:"$customer", product:"$product", price:"$price"
      },
    AvgPrice:{$avg:"$price"}
  }
])
```

9. Hive Architecture



Hive architecture has the following major components:

(i) Hive client
(ii) Hive Services.
(iii) Processing and Resource Management.
(iv) Distributed Storage.

(i) Hive client:

It is the client side component in hive which has the applications like Thrift applications, JDBC Application, ODBC Application and the respective driver. These are the database connection application to connect to the Server to process the data in Hive Server.

(ii) Hive Services

Hive Services are the another main component in the server side which has the following services:

→ Web Interfaces
→ Hive Server:
→ Command line Interface (CLI)
→ Metastore and Apache DB.

When client sends request to hive services then the processing happens with the help of these services.

(iii) Processing and Resource Management

Mapreduce, YARN are used for the processing and resource management task in apache hive.

(iv) Distributed Storage.

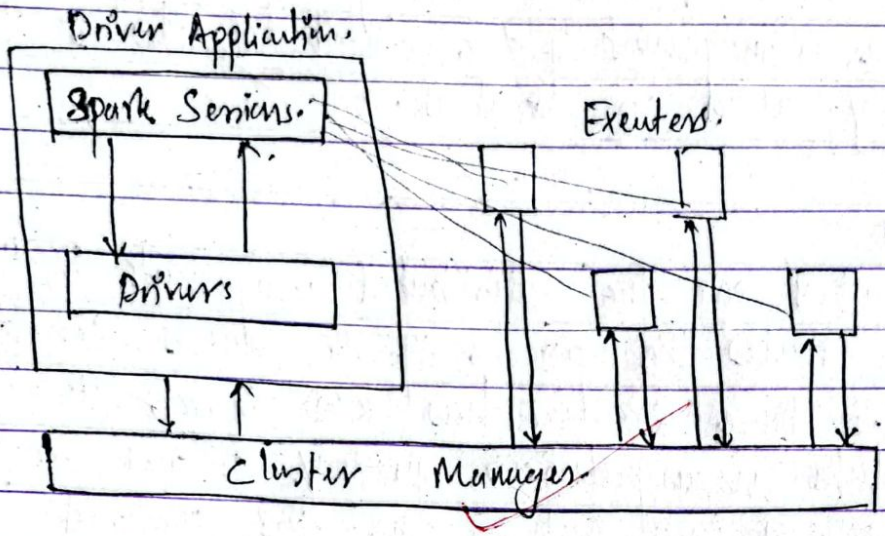
For storing the files, HDFS is used in Hive.

In this way, Apache Hive runs with the client request on Apache Hive client side. It is normally a Data Warehouse.

Comparing Apache Hive with Apache Pig-

Apache Pig.	Apache Hive.
(i) It was developed by Yahoo.	It was developed by Facebook.
(ii) It is used for Data Flow programing.	It is used for Research and Analysis.
(iii) It does not have fixed schema.	It has schema during Insertion.
(iv) Pig Latin is the Language used in Apache Pig.	HiveQL is the Language used in Apache Hive.

8. Architecture of Apache Spark.



Architecture of Apache spark consists of following core components.

(i) Driver Application.
(ii) cluster Manager
(iii) Executers.

(i). Driver Application.

Driver application is the spark application where the spark sessions and drivers are contained. This is simply the user session. When any real time data needs to be processed, first the user submits the job from this component of the Apache spark architecture. Spark Sessions are handled by the drivers.

(ii) cluster Manager.

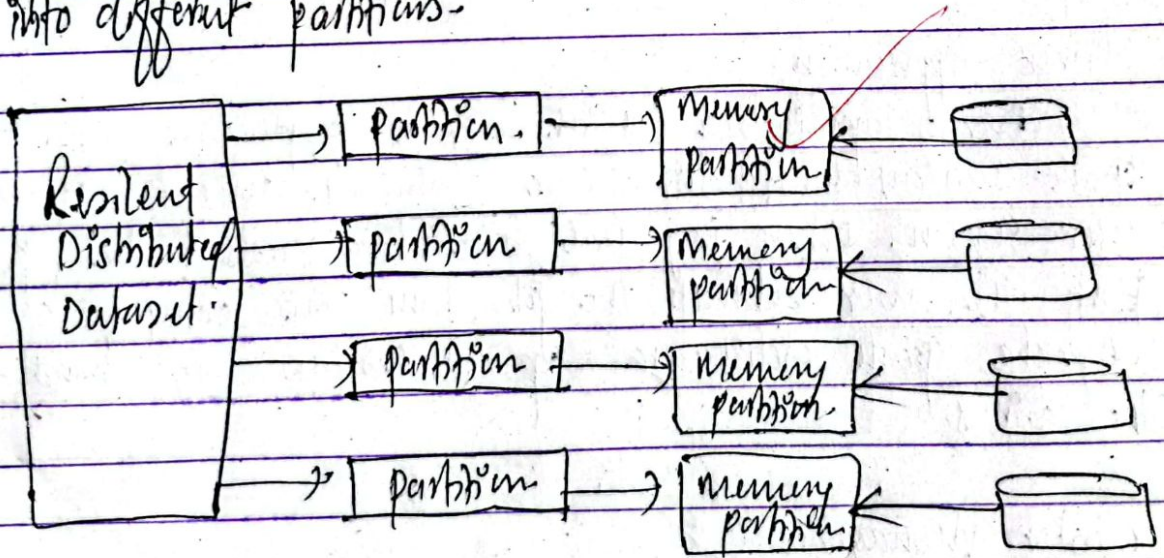
In order to process real time stream of data, there are lots of clusters where the partition are made to process the data. There are chunk of RDD (Resilient

Distributed Datasets) in order to manage these distributed clusters we need cluster manager. This cluster manager does the management of the distributed clusters these are generally called as executors.

(iii) Executors.

Executors are the distributed components in the spark which executes the apache spark actions i.e. data processing actions. These executors has RDD inside it, which performs the data manipulations quickly. It acts as the slaves since executors just perform the jobs assigned by the cluster managers.

RDD (Resilent Distributed Dataset) is the core part of the Apache Spark where the data are immutable and distributed into different partitions.

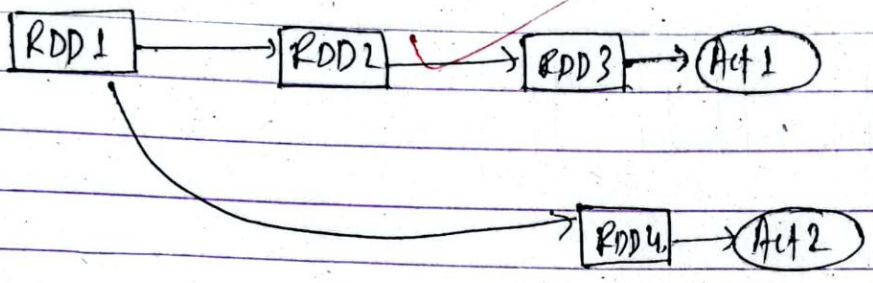


RDD helps to form its chain and the Data processing

becomes very fast in Apache Spark.

Spark transformation and actions.

In Apache Spark. we have chain of RDD's
let us consider an example.



Spark Transformation.

When there needs to form another RDD in the chain then it uses previous RDD and it does not uses the actual data. It just performs the transformation only. The chain formed are just from the transformation of data which is known as spark Transformation.

Example

Spark Action.

Whenever there needs to perform any task then spark will eagerly acts and perform the task which is the spark action.

In Spark Evaluation is slow and task action are fast so it is also called as lazy Evaluator.