computing based on java.
The MapReduce algorithm contains two important tasks: namely Map & Reduce. Map takes a set of data & converts it into another set of data, where individual elements are broken down into tuples (key/value pairs).
Secondly, reduce task, which takes the output from a map as an input & combines into those data tuples into a smaller set of tuples. As the sequence of name MapReduce implies, the reduce task is always performed after the map job.

The major advantage of MapReduce is that it is easy to scale data processing over multiple computing nodes.
Under the MapReduce model, the data processing primitives are called mappers & reducers.
Decomposing a data processing application into mappers & reducers is sometimes nontrivial.
But, once we write an application in the MapReduce form, scaling the application to run over hundreds, thousands, or even tens of thousands of machines. in a cluster is merely a configuration change.
This simple scalability is what has attracted many programmers to use the MapReduce model.

* The Algorithm :-

MapReduce program executes in three stages, namely map, stage, shuffle stage & reduce stage.

i) Mapstage : The map or mapper's job is to process the input data. Generally the input data is in the form of file or directory & is stored in Hadoop file system (HDFS). The input file is passed to the mapper function line by line. The mapper processes the data & creates several small chunks of data.

ii) Shufflestage : This stage combines all values associated to an identical key. For eg, (Are, 1) is there three times in the input file. So after the shuffling phase, the output will be like (Are, [1,1,1]).

iii) Reduce stage : This stage is combination of shuffle stage & Reduce stage. The Reducer's job is to process the data that comes from the mapper. After processing, it produces a new set of output, which will be stored in HDFS.

* Inserting Data into HDFS :

- The MapReduce framework operates on <key, value> pairs that is, the framework views the input to the job as a set of <key, value> pairs & produces a set of <key, value> pairs as the output of the job, conceivably of different types.
- The key & the value classes should be in serialized manner by the framework & hence, need to implement the Writable interface. Additionally, the key classes have to implement the Writable-Comparable interface to facilitate sorting by the framework.

- Input & Output types of MapReduce job :
(Input) < k1 , v1 > → map → < k2, v2 > → reduce → < k3, v3 >
(output).

CONCLUSION : Thus we have learnt how to design a distributed application using MapReduce & process a log file of a system.

DS & BDA Lab
Group A - Experiment 3
TITLE: Write an application using HiveQL for flight information system.

OBJECTIVE:

1. To learn NoSQL Databases (Open Source) such as Hive / Hbase.
2. To study detailed concept HIVE.

SOFTWARE REQUIREMENTS:

1. Ubuntu 14.04 / 14.10
2. GNU C Compiler.
3. Hadoop
4. Java
5. HIVE.

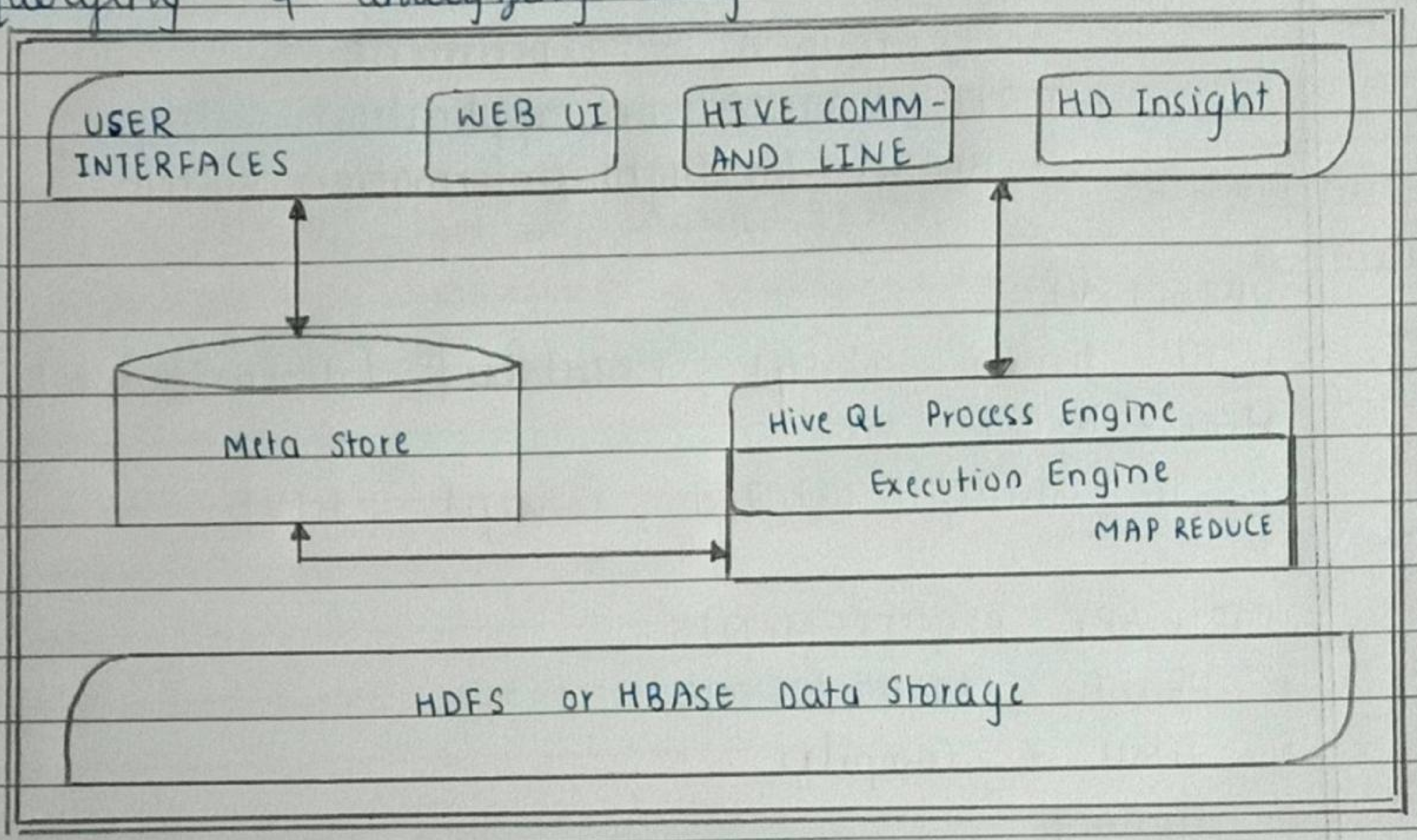
PROBLEM STATEMENT: Write an application using HiveQL for flight information system which will include:

1) Creating, Dropping & altering Database tables.
2) Creating an external Hive table to connect to the HBase for Customer Information Table.
3) Load table with data, insert new values & field in the table, Join tables with Hive.
4) Create Index on Flight Information Table.
5) Find the average departure delay per day in 2008.

THEORY:

*. Hive is a data warehouse infrastructure tool to process structured data in Hadoop. It resides on top of Hadoop to summarize Big data, & makes

querying & analyzing easy.



The following describes each unit :

i) User Interface : Hive supports HiveWeb UI, Hive command line, & HIVE HD Insight (In Windows server).

ii) Meta store : Stores tables, databases, columns in table, their data types & HDFS mapping.

iii) HiveQL Process Engine : Instead of MapReduce in Java, write a query for MapReduce job & process it.

iv) Execution Engine : Process query & generates results same as MapReduce results.

v) HDFS or HBase : To store data into file system.

Steps :-

i) Verifying Hadoop Installation

ii) Installing Hive

iii) Extracting & verifying Hive Archive.

iv) Copying files to /urs/ local / hive directory.

v) Setting up environment for Hive.

vi) Configuring Hive.

vii) Creating Database:

- can use SCHEMA in place of Database.
- Drop Database is a statement that drops all tables & deletes the database
- Create Database is statement used to create a database in Hive.
- Load Data statement: In Hive, we can insert data using LOAD statement.

• LOCAL - identifies to specify local path.
• OVERWRITE - overwrite data in table.
• PARTITION - optional in database.

- Alter Table statement: It is used to alter tables in Hive.

Attributes are :-

• RENAME
• ADD COLUMNS
• DROP
• CHANGE
• REPLACE COLUMNS

*. The Hive Query Language (Hive QL) is a query language for Hive to process & analyze structured data in a Metastore. SELECT statement is used to retrieve the data from the table. WHERE clause works similar to a condition. It filters the data using the condition & gives a finite result. The built-in operators & functions generate an expression, which fulfills the condition.

CONCLUSION : Thus we have learnt how to design a application HBase & HiveQL for flight information system.