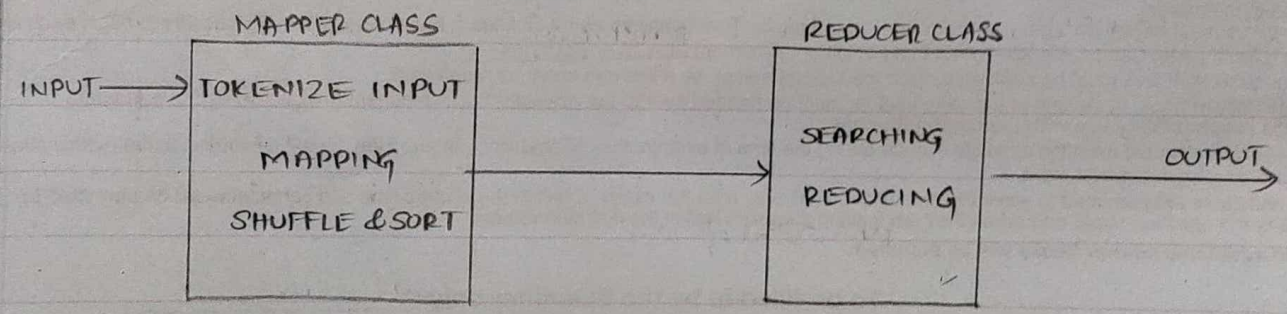
1. Explain in detail with neat diagram about executing Map phase-shuffling and sorting and Reducing phase.
2. Explain architecture of pig and write its advantages.
3. Explain about parameter substitution with examples.
4. Write about HIVE and its architecture.
5. Explain the following with examples: a. loading data into HIVE tables, b. Managed Tables.

1. -The Map Reduce algorithm contains two important tasks, namely Map and Reduce.
-The Map task is done by means of Mapper class.
-The reduce task is done by means of reducer class.



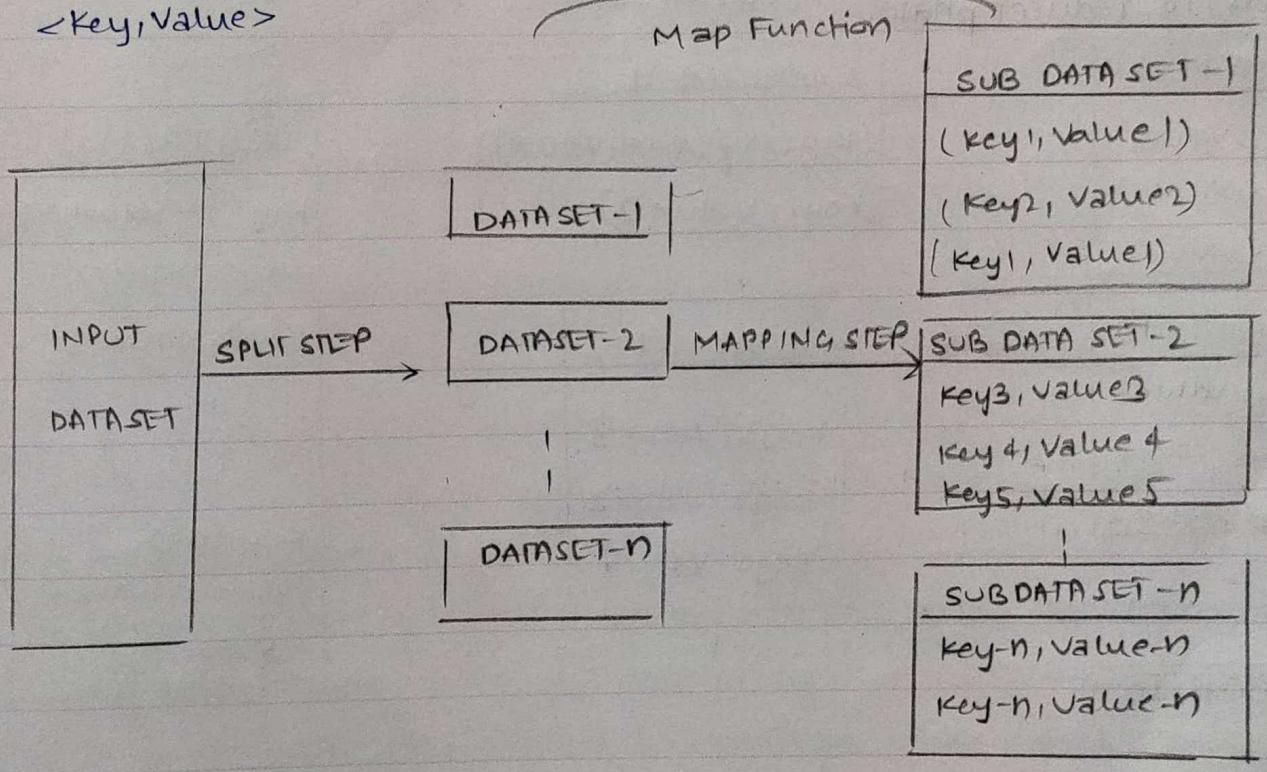
1. MAP PHASE:-

-Map Function is the first step in map-reduce algorithm.
-Map phase will work on key & value pairs input. It takes input tasks and divides them into smaller sub-tasks and then perform required computation on each sub-task in parallel.
-In map phase, key & value is in the form of byte offset values. A list of data elements is provided to mapper function called map().
-Map() transforms input data to an intermediate output data element.

- Mapper output will be displayed in the form of (K,V) pairs.
- Map phase performs the following two sub-steps:

1. Splitting:- Takes input datasets from source and divide into smaller sub-datasets.
2. Mapping:- Takes the smaller sub-datasets as an input and perform required action or computation on each sub-dataset.

- The output of the map function is a set of key and value pairs as <Key, Value>

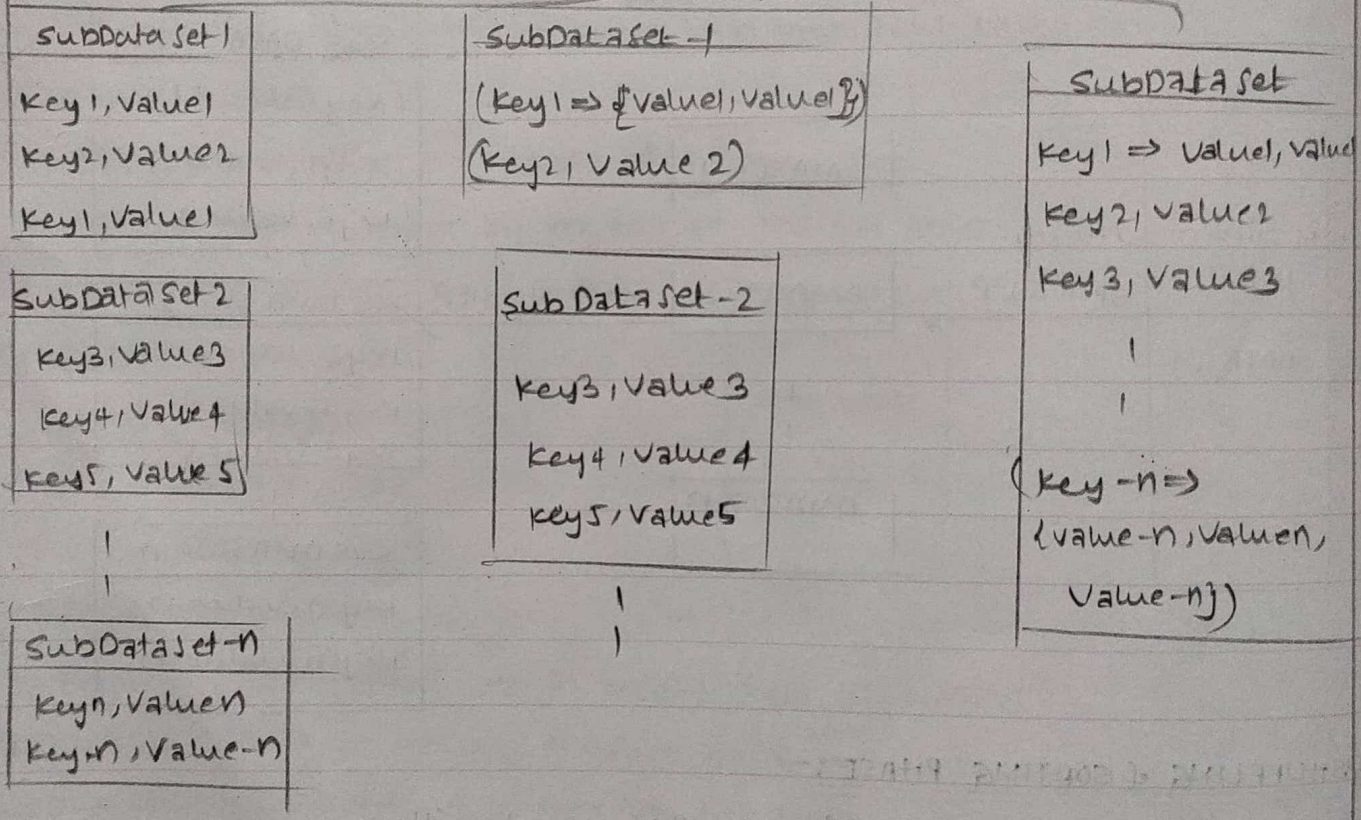


SHUFFLING & SORTING PHASE:-

- This is the second step in Map Reduce algorithm.
- Shuffle Function is also known as 'combine function'.
- Mapper output will be taken as input to sort and shuffle.
- Shuffling is grouping of the data from various nodes based on the key.
- This is a logical phase.

- Sort is used to list the shuffled inputs in sorted order.
- It performs the following two steps.
  1. Merging
  2. Sorting.
- It takes output coming from Map Function and performs these two sub-steps on each and every key-value pair.
- Finally, shuffle function returns a list of <key, List<Value>> sorted pairs to reducer phase.

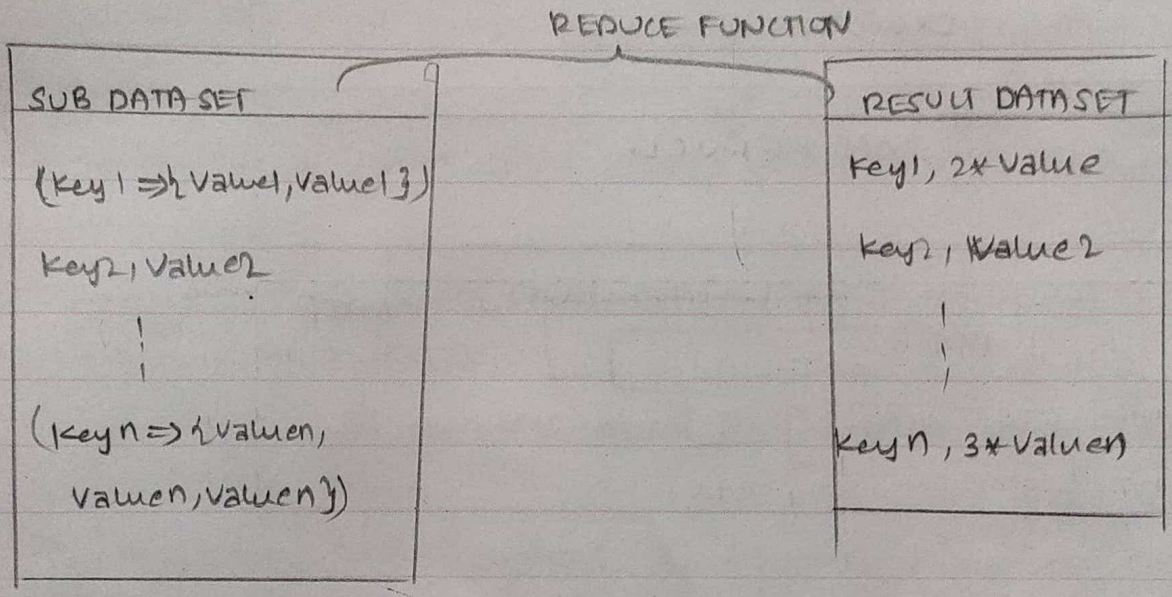
Shuffle Function.



3. REDUCER PHASE:-

- Reducer Phase is the final step in Map Reduce algorithm.
- Reducing is inherently sequential unless processing multiple tasks.
- It takes list of <key, List<Value>> sorted pairs from Shuffle function and perform reduce operations.

-Reducer combines all the values together and provide single output value for the specific keys.
-This phase performs only one step -Reduce step.
-Reduce step <key, value> pairs are different from map step <key, Value> pairs.

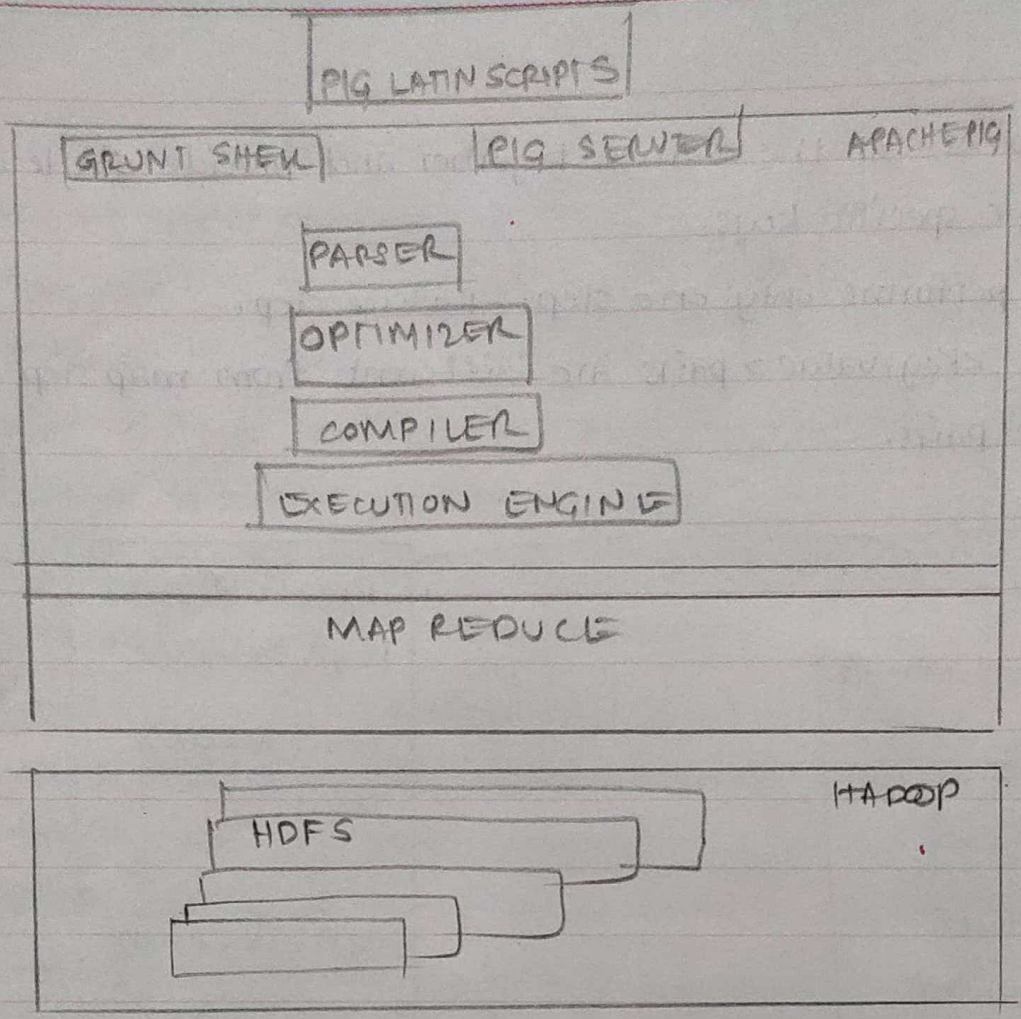


2. PIG ARCHITECTURE:-

-The language used to analyze data in Hadoop using Pig is known as Pig Latin.
-It is a highlevel data processing language which provides a rich set of data types and operators to perform various operations on the data.
-The architecture of Pig is shown below:-

PIG COMPONENTS:-

-There are various components in the Apache pig framework.
-Let us take a look at the major components.



PARSER:-

- Initially, the Pig scripts are handled by the Parser.
- It checks the syntax of the scripts, does type checking and other miscelleaneous checks.
- The output of parser will be a DAG, which represents the Pig Latin statements and logical operators.

OPTIMIZER:-

- The logical plan (DAG) is passed to the logical optimizer, which carries out the logical optimizations.

COMPILER:-

- The compiler compiles the optimized logical plan into a series of MapReduce jobs.

EXECUTION ENGINE:-

- Finally, the mapreduce jobs are submitted to Hadoop in a sorted order.
- Finally, these jobs are executed on hadoop producing the desired results.

3. PARAMETER SUBSTITUTION:-

- Pig Latin has an ophan called param, using this we can write dynamic scripts.
- Assume, we have a file called numbers with below data,
  12, 23, 34, 12, 56, 34, 57, 12
- If we want to list numbers equal to 12, then we write pig latin code like below:

```
Numbers = load '/data/numbers' as (number: int);
SpecificNumber = filter numbers by number == 12;
Dump specific Number;
```

- Usually we write above code in a file. Let us assume we have written it in a file called numbers.pig
- And we write code from file using
  Pig -f /path/to/numbers.pig
- Later if we want to see only numbers = 34, then we change second line to
  SpecificNumber = filter Numbers by number == 34;
- And we re-run the code using same command.
- But it is not a good practice to touch the code in production, so we make use of -param ophion of piglatin.

-Whatever values we want to decide at the time of running we make them dynamic.

-Now we want to decide number to be filtered at the running time we can write second line like below:

specificNumber = filter numbers by number == $dynanumber

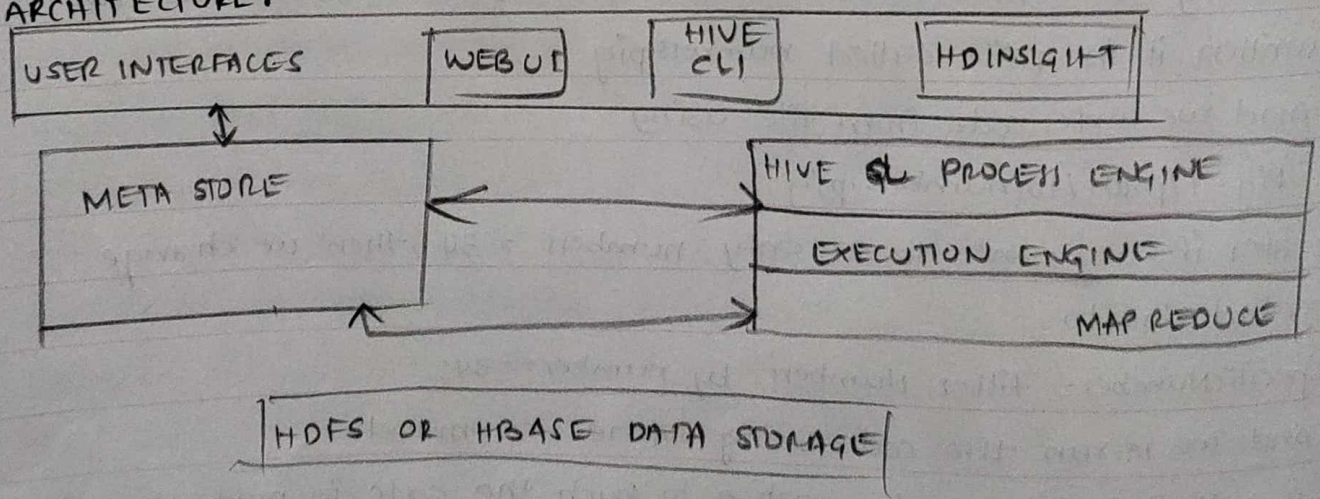
and we run code like below:-

pig -param dynanumber=12 -f numbers.pig.

4. HIVE:-

- Hive is a data warehouse infrastructure tool to process structured data in Hadoop.
- It resides on top of Hadoop to summarize big data, and makes querying and analyzing easy.
- It stores schema in a database and processed data into HDFS.
- It is designed for OLAP.
- It is familiar, fast, scalable and extensible.

ARCHITECTURE:-



USER INTERFACE:-

- Hive is a data warehouse infrastructure software that can create interaction between user and HDFS.
- The user interfaces that Hive supports are Hive web UI, Hive command line and Hive HD insight.

META STORE:-

- Hive chooses respective database servers to store the schema or Metadata of tables, databases, columns in a table, their datatypes and HDFS mapping.

HIVE QL PROCESS ENGINE:-

- Hive QL is similar to SQL for querying on schema info on the Metastore.
- It is one of the replacements of traditional approach for MapReduce program.

EXECUTION ENGINE:-

- The conjunction part of Hive QL process Engine and MapReduce is Hive Execution Engine.
- It processess the query and generates results as same as MapReduce results.

HDFS OR HBASE:-

- Hadoop distributed file system or HBASE are the data storage techniques to store data into the file system.

5.2. LOADING DATA INTO HIVE:-

- Sinve Hive has no row-level insert, update and delete operations, the only way to put data into your table is to use one of the bulk

load operations.

-Alternatively we can write the files in directories looked up by Hive.

1. Use LOAD DATA command.

```
LOAD DATA LOCAL INPATH '/home/hduser/docs/hive/companies/src' INTO
TABLE hiveclass.companies;
```

-Here, we use the LOAD DATA function for that purpose. what this command does is that it will just copy the data from the local file path specified and places it inside the directory that was specified to store particular table data at time of creation of table.

2. Specify a directory as path and not an individual file.

```
LOAD DATA LOCALMPATH '/home/hduser/docs/hive/companies' INTO
TABLE hiveclass.companies.
```

-Now load DATA, you're going to specify that the data stored in the local file system with keyword LOCAL, then using the keyword INPATH you can specify the path of file that is stored and which is the table that you want to load data into

MANAGED TABLES:-

-Managed Tables is also called as internal table. This is the default table in Hive.

-When we create table in HIVE without specifying it as external, by default we will get a Managed table.

- Use default location or explicitly provided location to store data.

-Relevant directory is created.

-Metadata is updated in metastore.

- LOAD DATA statement moves data into appropriate directory created by HIVE.
- When table is dropped, data is removed and metadata is removed from metastore.
- Hive controls the life cycle of table.

```
CREATE TABLE daily (exchangename STRING, symbol STRING, date STRING,
 openclose adj' MAP<STRING, FLOAT>, highlow STRUCT<high:FLOAT, low:FLOAT>,
volume INT)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
COLLECTION ITEMS TERMINATED BY '#'
MAP KEYS TERMINATED BY '*'
LINES TERMINATED BY '\n'

STORED AS TEXTFILE;
```