ASSIGNMENT NO: 10

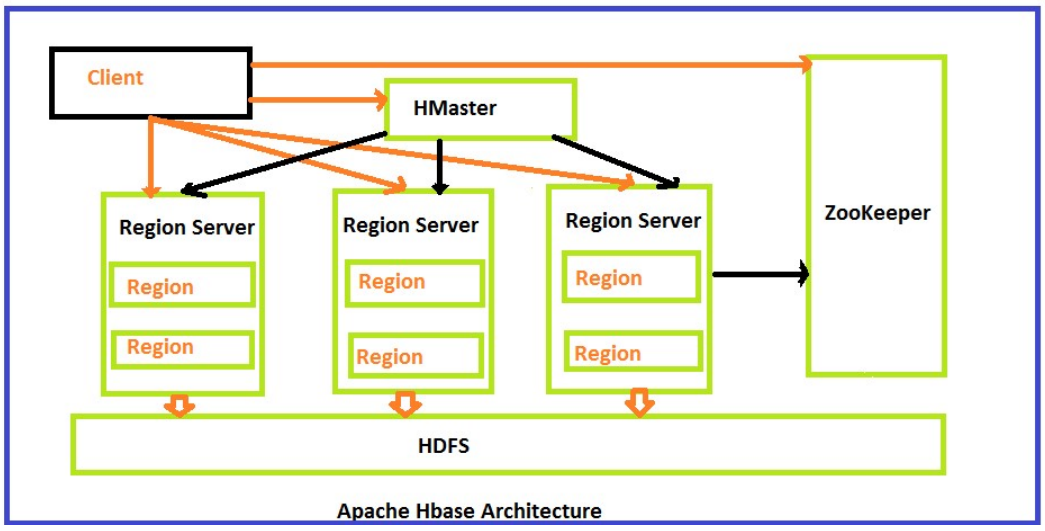
AIM : Using HBase perform following operations

Create a table, Add, Retrieve, Modify, and delete the record(s), Drop the table

INDEX TERMS: Hbase Database Operations, Hbase,

THEORY

In HBase, tables are split into regions and are served by the region servers. Regions are vertically divided by column families into "Stores". Stores are saved as files in HDFS. Shown below is the architecture of HBase.



HBase has three major components: the client library, a master server, and region servers. Region servers can be added or removed as per requirement.

The master server -

Assigns regions to the region servers and takes the help of Apache ZooKeeper for this task. Handles load balancing of the regions across region servers. It unloads the busy servers and shifts the regions to less occupied servers. It Maintains the state of the cluster by negotiating the load balancing. It is responsible for schema changes and other metadata operations such as creation of tables and column families.

Regions

Regions are nothing but tables that are split up and spread across the region servers.

The region servers have regions that -

- Communicate with the client and handle data-related operations.

- Handle read and write requests for all the regions under it.
- Decide the size of the region by following the region size thresholds.

When we take a deeper look into the region server, it contain regions and stores as shown below:

Regional Server

The store contains memory store and HFiles. Memstore is just like a cache memory. Anything that is entered into the HBase is stored here initially. Later, the data is transferred and saved in Hfiles as blocks and the memstore is flushed.

Zookeeper

Zookeeper is an open-source project that provides services like maintaining configuration information, naming, providing distributed synchronization, etc.

Zookeeper has ephemeral nodes representing different region servers. Master servers use these nodes to discover available servers.

In addition to availability, the nodes are also used to track server failures or network partitions.

Clients communicate with region servers via zookeeper.

In pseudo and standalone modes, HBase itself will take care of zookeeper.

Creating a Table using HBase :

You can create a table using the create command, here you must specify the table name and the Column Family name. The syntax to create a table in HBase shell is shown below.

```
create ‘<table name>’,’<column family>’
```

Example

Given below is a sample schema of a table named emp. It has two column families: “personal data” and “professional data”.

You can create this table in HBase shell as shown below.

```
hbase(main):002:0> create 'emp', 'personal data', 'professional data'
```

Inserting Data using HBase

To create data in an HBase table, the following commands and methods are used:

- put command,
- add() method of Put class, and
- put() method of HTable class.

Using put command, you can insert rows into a table. Its syntax is as follows:

```
put ’<table name>’,’row1’,’<colfamily:colname>’,’<value>’
```

```
hbase(main):005:0> put 'emp','1','personal data:name','raju'
0 row(s) in 0.6600 seconds
hbase(main):006:0> put 'emp','1','personal data:city','hyderabad'
0 row(s) in 0.0410 seconds
hbase(main):007:0> put 'emp','1','professional
data:designation','manager'
0 row(s) in 0.0240 seconds
hbase(main):007:0> put 'emp','1','professional data:salary','50000'
0 row(s) in 0.0240 seconds
```

Updating Data using HBase

You can update an existing cell value using the put command. To do so, just follow the same syntax and mention your new value as shown below.

```
put ‘table name’,’row ’,'Column family:column name',’new value’
```

Example :

```
hbase(main):002:0> put 'emp','row1','personal:city','Delhi'
0 row(s) in 0.0400 seconds
```

Reading Data using HBase

The get command and the get() method of HTable class are used to read data from a table in HBase. Using get command, you can get a single row of data at a time. Its syntax is as follows:

```
get ’<table name>’,’row1’
```

Given below is the syntax to read a specific column using the get method.

```
hbase> get 'table name', ‘rowid’, {COLUMN ⇒ ‘column family:column name ’}
```

Deleting a Specific Cell in a Table

Using the delete command, you can delete a specific cell in a table. The syntax of delete command is as follows:

```
delete ‘<table name>’, ‘<row>’, ‘<column name >’, ‘<time stamp>’
```

Example :

```
hbase(main):006:0> delete 'emp', '1', 'personal data:city',
1417521848375
0 row(s) in 0.0060 seconds
```

Deleting All Cells in a Table

Using the “deleteall” command, you can delete all the cells in a row. Given below is the syntax of deleteall command.

```
deleteall ‘<table name>’, ‘<row>’,
```

Scaning using HBase

The **scan** command is used to view the data in HTable. Using the scan command, you can get the table data. Its syntax is as follows:

```
scan ‘<table name>’
```

FAQS:
Q1.What are the advantages of Hbase?
Q2. Which applications uses Hbase?

Conclusion:
Outcome of the experiment is students are able to implement Basic Operation with HBase.