

Experiment - 5

Aim: To execute HBase commands in Hadoop Hortonworks Sandbox

Theory:

Features of HBase

- i) HBase is a distributed column-oriented database built on top of the HDFS file system.
- ii) It is an open-source project and is horizontally scalable.
- iii) It leverages the fault tolerance provided by the Hadoop File System (HDFS).
- iv) It is a part of the Hadoop ecosystem that provides random real-time read/write access to data in the Hadoop File System.
- v) One can store the data in HDFS either directly or through HBase.
- vi) Data consumer reads/accesses the data in HDFS randomly using HBase.
- vi) HBase sits on top of the Hadoop File System and provides read and write access.

HBase Architecture

HBase has three major components:

- i) HMaster or NameNode
- It is the master node that handles all region servers within HBase.
- The HMaster is responsible for assigning regions to region servers and performs create, delete and update operations.
- It also tries to distribute the load across all region servers with the help of zookeeper.

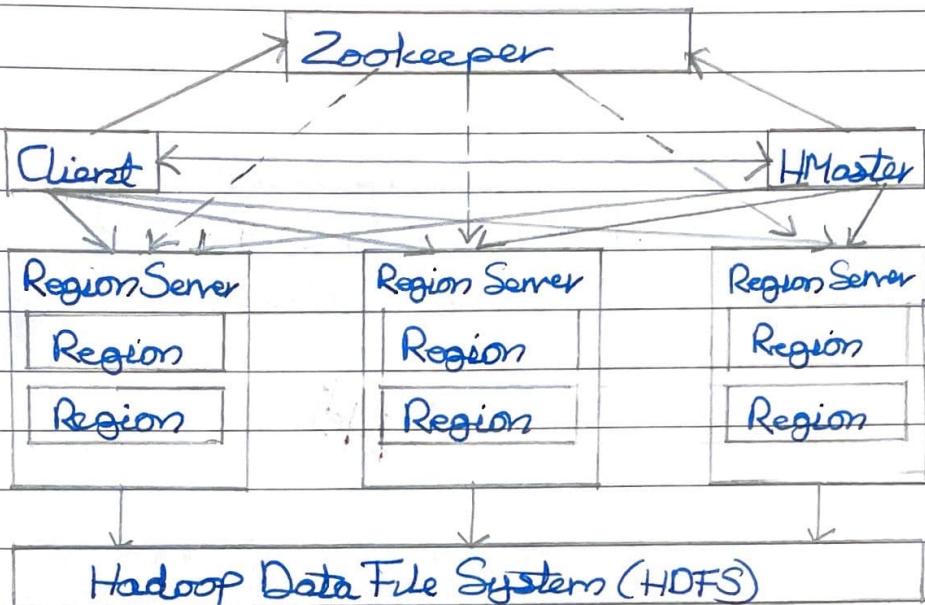
- If any of the region servers are down, it is responsible for load balancing and recovery actions.

ii) Regions

- As HBase is a distributed database, region servers are responsible for distributing table keys and managing read/write operation in particular regions.
- Within one region server, there can be multiple regions.
- Table data is split into multiple regions horizontally.
- Each region contains certain ranges of keys, & keys in each region are sorted.
- The default size of each region is 256 MB, which can be further configured according to requirement.

iii) Zookeeper

- It manages the health of all the region servers and the HMaster by maintaining sessions.
- Zookeeper listens to the heartbeat of the HMaster and region servers continuously. By doing so, it makes regular checks for the health of all nodes. In turn, the master node can distribute multiple requests in different region servers.
- HBase cluster maintains two types of master nodes: i) Active HMaster and ii) Inactive HMaster.



HBase Commands

- i) **Create**: The 'create' command is used to create a table within HBase Shell.
eg - create 'table_name', 'column_family1', 'column_family2'.
- ii) **List**: This command is used to list all tables within the shell
eg - list
- iii) **put**: Used to add or update only one column of one rowkey
eg - put 'table_name', 'row_key', 'column_family', 'column', 'VALUE'
- iv) **scan**: Scan applies to all the rowkey and in turn all columns within the specified table.
eg - scan 'table_name'
- v) **get**: This is used to fetch data associated with a particular rowkey.
eg - get 'table_name', 'row_key'
- vi) **LIMIT**: If we want to get first n row's data, LIMIT fits in perfectly. This is applicable only for SCAN.
eg - scan 'table_name', {LIMIT=>n}
- vii) **DELETE TABLE**: First we need to disable a table only after that can we delete a table.

eg - disable 'table-name'
delete 'table-name'

Conclusion: In conclusion we have implemented various HBase queries and commands and learnt about the features and architecture of HBase.

Ques)