**ASSIGNMENT - 31.1**

**Problem Statement:**

Explain in brief

● Differences between HBASE and HDFS.

● List and explain the main components of HBASE.

● Does Hbase support sql

**Solution:**

* **Differences between HBASE and HDFS:**
* **HBase:**

HBase is an open source, distributed, versioned, column-oriented, No-SQL / Non-relational database management system that runs on the top of Hadoop.

It adds transactional capability to hadoop, allowing users to update data records. Hadoop is designed for batch processing of large dataset, but with HBase on the top of Hadoop we can process real time dataset.

In HBase a master node manages the cluster and region servers store portions of the tables and perform the work on the data.

An HBase system comprises a set of tables. Each table contains rows and columns, much like a traditional database.

Each table must have an element defined as a Primary Key, and all access attempts to HBase tables must use this Primary Key. An HBase column represents an attribute of an object.

* **HDFS:**

HDFS is a distributed file system which provides redundant storage space for storing files which are very huge in sizes; files which are in the range of Terabytes and Petabytes.

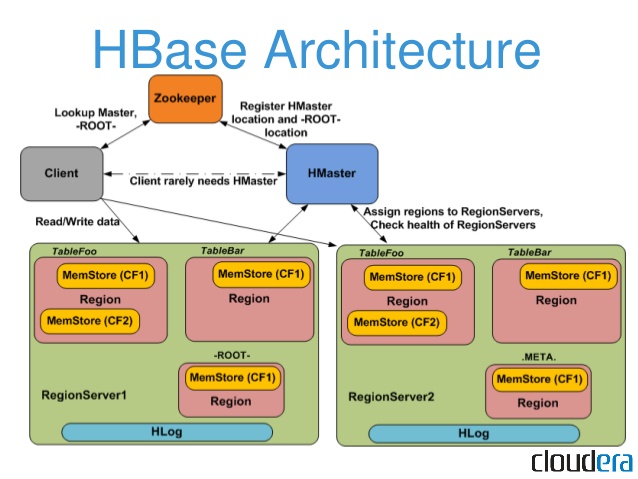
In HDFS data is stored reliably. Files are broken into blocks and distributed across nodes in a cluster.

After that each block are replicated, means copies of blocks are created on different machines. Hence if a machine goes down or get crashed, then also we can easily retrieve and access our data from different machines.

By default 3 copies of a file are created on different machines. Hence it is highly fault tolerant. HDFS provides faster file read and write mechanism, as data is stored in different nodes in a cluster.

Hence user can easily access the data from any machine in a cluster. Hence HDFS is highly used as a platform for storing huge volume and different varieties of data worldwide.

* **Components of Apache HBase Architecture:**

****

HBase architecture has 3 important components-

* HMaster
* Region Server and
* ZooKeeper
* **HMaster:**

HBase HMaster is a lightweight process that assigns regions to region servers in the Hadoop cluster for load balancing.

Responsibilities of HMaster –

* Manages and Monitors the Hadoop Cluster
* Performs Administration (Interface for creating, updating and deleting tables.)
* Controlling the failover
* DDL operations are handled by the HMaster
* Whenever a client wants to change the schema and change any of the metadata operations, HMaster is responsible for all these operations.
* **Region Server:**

These are the worker nodes which handle read, write, update, and delete requests from clients. Region Server process, runs on every node in the hadoop cluster. Region Server runs on HDFS DataNode and consists of the following components –

* Block Cache – This is the read cache. Most frequently read data is stored in the read cache and whenever the block cache is full, recently used data is evicted.
* MemStore- This is the write cache and stores new data that is not yet written to the disk. Every column family in a region has a MemStore.
* Write Ahead Log (WAL) is a file that stores new data that is not persisted to permanent storage.
* HFile is the actual storage file that stores the rows as sorted key values on a disk.
* **Zookeeper:**

HBase uses ZooKeeper as a distributed coordination service for region assignments and to recover any region server crashes by loading them onto other region servers that are functioning.

ZooKeeper is a centralized monitoring server that maintains configuration information and provides distributed synchronization. Whenever a client wants to communicate with regions, they have to approach Zookeeper first.

HMaster and Region servers are registered with ZooKeeper service, client needs to access ZooKeeper quorum in order to connect with region servers and HMaster. In case of node failure within an HBase cluster, ZKquoram will trigger error messages and start repairing failed nodes.

ZooKeeper service keeps track of all the region servers that are there in an HBase cluster- tracking information about how many region servers are there and which region servers are holding which DataNode.

HMaster contacts ZooKeeper to get the details of region servers. Various services that Zookeeper provides include –

* Establishing client communication with region servers.
* Tracking server failure and network partitions.
* Maintain Configuration Information
* Provides ephemeral nodes, which represent different region servers.
* **Does Hbase support sql?**

HBase does not support sql.

HBase is a column-oriented database management system that runs on top of hdfs. It is well suited for sparse data sets, which are common in many big data use cases.

Unlike relational database systems, HBase does not support a structured query language like SQL; in fact, HBase isn’t a relational data store at all. HBase applications are written in Java much like a typical MapReduce application. HBase does support writing applications in Avro, REST, and Thrift.

An HBase system comprises a set of tables. Each table contains rows and columns, much like a traditional database. Each table must have an element defined as a Primary Key, and all access attempts to HBase tables must use this Primary Key. An HBase column represents an attribute of an object;

For example, if the table is storing diagnostic logs from servers in your environment, where each row might be a log record, a typical column in such a table would be the timestamp of when the log record was written, or perhaps the server name where the record originated. In fact, HBase allows for many attributes to be grouped together into what are known as column families, such that the elements of a column family are all stored together.

This is different from a row-oriented relational database, where all the columns of a given row are stored together. With HBase you must predefine the table schema and specify the column families.

However, it’s very flexible in that new columns can be added to families at any time, making the schema flexible and therefore able to adapt to changing application requirements.

Just as HDFS has a NameNode and slave nodes, and MapReduce has JobTracker and TaskTracker slaves, HBase is built on similar concepts. In HBase a master node manages the cluster and region servers store portions of the tables and perform the work on the data.