**Differences between HBASE and HDFS.**

|  |  |
| --- | --- |
| **HBASE** | **HDFS** |
| HBase - open source, distributed, versioned, column-oriented,  No-SQL / Non-relational database management system is needed to run on top of Hadoop. | HDFS –is called as distributed file system which will distribute data over cluster of machines and will take care of the redundancy. |
| HBase will store data as key/value pairs as in a column database.  Each key will hold a value. | HDFS is used to store data in the form of flat files. |
| HBASE has the ability of random read/writes of data. | Instead HDFS will not support random reads. |
| HBase will provide flexible data model**.** | HDFS in contrast will not provide with any flexible data model. |
| HDFS files will be written once and read many times.  And there is no option for random write or read. | Data will be indexed by row key  Will be provided with very flexible data model  data will stored inside the hashed table and will be accessed in a random manner |
| HDFS will never support fast individual record lookups | Whereas HDFS will support faster data lookup in the tables |

**List and explain the main components of HBASE**

HBase is composed of three types of servers in a master slave type of architecture.

• **Region servers** serve data for reads and writes.

• **HBase Master** process handles the Region assignment, DDL (create, delete tables) operations

• **Zookeeper** maintains a live cluster state.

**REGIONS:**

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

**REGION SERVER**:

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.

The store contains memstore 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 memory store is flushed.

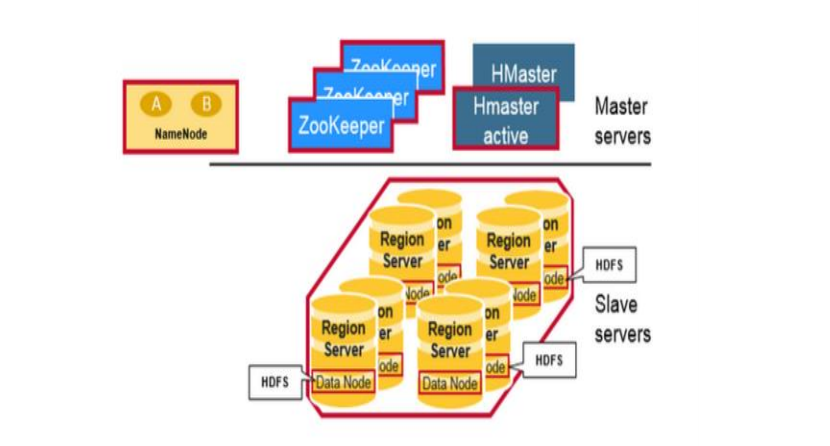
**MASTER SERVER:**

The master server

* Assigns regions to the region servers and takes the help of Apache Zoo Keeper 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.
* Maintains the state of the cluster by negotiating the load balancing.
* Is responsible for schema changes and other metadata operations such as creation of tables and column families.

## 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.
* HBase uses Zoo Keeper as a distributed coordination service to maintain server state in the cluster.
* Zookeeper maintains which servers are alive and available, and provides server failure notification.
* Zookeeper uses consensus to guarantee common shared state. Note that there should be three or five machines for consensus.

****

**HBASE ARCHITECTURE DIAGRAM**

**Does Hbase support sql**

HBase is a column-oriented database management system that runs on top of [HDFS](http://www.ibm.com/software/data/infosphere/hadoop/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](http://www.ibm.com/software/data/infosphere/hadoop/mapreduce/) application.

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.

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.