

**Solution-**

1. **Difference between HBASE and HDFS.**

**HBASE-**

* Apache Hbase is the Hadoop database, a distributed, column oriented, scalable, big data store.
* Use Apache HBase when you need random, realtime read/write access to your Big Data.
* This project's goal is the hosting of very large tables - billions of rows, X millions of columns - atop clusters of commodity hardware.
* Apache HBase is an open-source, distributed, versioned, non-relational database modelled after Google's Bigtable: A Distributed Storage System for Structured Data by Chang et al.
* Just as Bigtable leverages the distributed data storage provided by the Google File System, Apache HBase provides Bigtable-like capabilities on top of Hadoop and HDFS.
* HBase provides you with the following:  
  1. Low latency access to small amounts of data from within a large data set. You can access single rows quickly from a billion row table.  
  2. Flexible data model to work with and data is indexed by the row key.  
  3. Fast scans across tables.  
  4. Scale in terms of writes as well as total volume of data.

**HDFS-**

HDFS is a distributed file system and has the following properties:  
- It is optimized for streaming access of large files. You would typically store files that are in the 100s of MB upwards on HDFS and access them through MapReduce to process them in batch mode.   
- HDFS files are write once files. You can append to files in some of the recent versions but that is not a feature that is very commonly used. Consider HDFS files as write-once and read-many files. There is no concept of random writes.  
- HDFS doesn't do random reads very well.

* Optimized for streaming access of large files.
* Follows write-once read-many ideology.
* Doesn't support random read/write.

 HDFS provides us storage, but in a fault tolerant manner with high throughput and lower risk of data loss(because of the replication).But, being a File System , HDFS lacks random read and write access. This is where HBase comes into picture. It’s a distributed, scalable, big data store, modelled after Google’s BigTable. Cassandra is somewhat similar to hbase.

1. **Components of HBase-**

HBase architecture has 3 important components-

HMaster

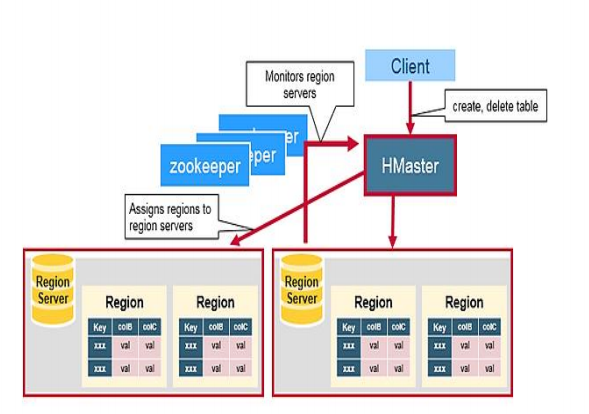
Region Server

ZooKeeper.

1. **HMaster**

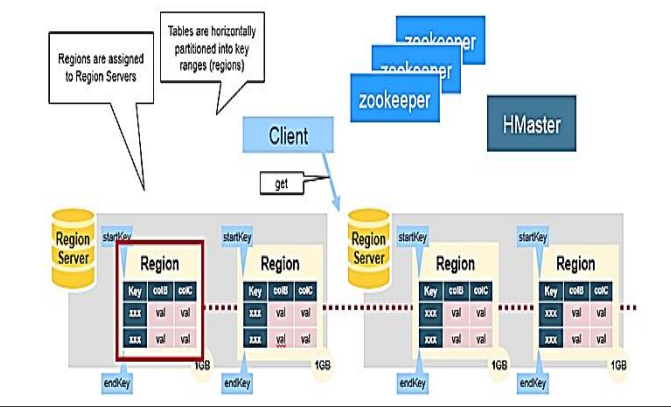
HBase HMaster is a lightweight process that assigns regions to region servers in the Hadoop cluster for load balancing. Responsibilities of HMaster –

* 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 assignment, DDL (create, delete tables) operations are handled by the HBase Master• Coordinating the region servers
* Assigning regions on startup
* Re-assigning regions for recovery or load balancing
* Monitoring all RegionServer instances in the cluster (listens for notifications from zookeeper) Admin functions
* Interface for creating, deleting, updating tables



1. **Region Server**

* HBase Tables are divided horizontally by row key range into “Regions.”
* A region contains all rows in the table between the region’s start key and end key.
* Regions are assigned to the nodes in the cluster, called “Region Servers,” and these serve data for reads and writes.
* A region server can serve about 1,000 regions.

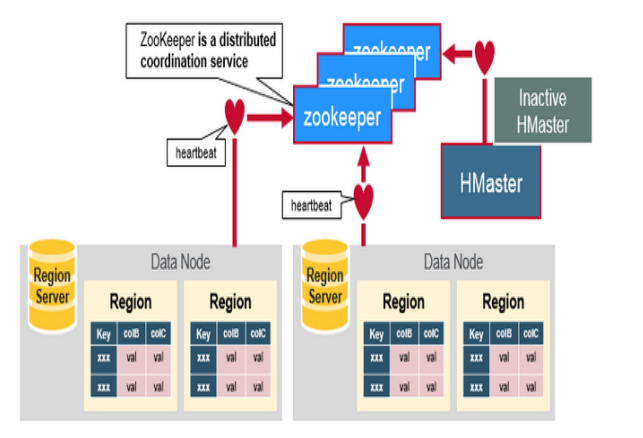


1. **Zookeeper**

* HBase uses ZooKeeper 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.
* ZooKeeper is 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.



1. **Does HBase supports sql.**

* No Hbase do not supports sql. To have sql like queries, you need to have pheoniox on top of HBase.