**HBase**

HBase is an open source, distributed database, developed by[Apache](https://www.guru99.com/apache.html)Software foundation.

Initially, it was **Google Big Table** which was re-named as HBase and is primarily written in Java language.

Storage capacity of data is massive from terabytes to petabytes.

HBase is a column-oriented storage, so it provides fast querying, fetching of results and high amount of data storage.

HBase is one of the commonly used NoSQL databases (Some of the NoSQL models present in the market are **Cassandra, MongoDB** and **CouchDB**).

**HBase Unique Features**

* HBase is built for **low latency operations** i.e., operations experiencing small delay times.
* HBase is used extensively for random read and write operations.
* HBase stores large amount of data in terms of tables.
* Provides linear and modular scalability over cluster environment
* Strictly consistent to read and write operations

If we want to search particular row from a huge amount of data (i.e., web application consisting of billions of rows), HBase is the ideal choice as query fetch time in less.

**Where is HBase used?**

* Telecom Industry
* Banking Industry
* Whenever there is a need to write heavy applications.
* Performing online log analytics and to generate compliance reports.

**HBase - Architecture**

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

**Master Server**

* Assigns regions to the region servers with the help of Apache ZooKeeper.
* Handles load balancing of the regions across region 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.

**Region server**

The region servers have regions (nothing but tables that are split up and spread across the region servers) 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 (part of region server) contains memory store (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 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

## HBase Shell (same as above defined):

* HBase contains a shell using which you can communicate with HBase. HBase uses the Hadoop File System (HDFS) to store its data. It will have a master server and region servers.
* The data storage will be in the form of regions (tables). These regions will be split up and stored in region servers.
* The master server manages these region servers and all these tasks take place on HDFS. Given below are some of the commands supported by HBase Shell.

**RDBMS vs HBase**

* Schema/Database in RDBMS can be compared to namespace in Hbase.
* A table in RDBMS can be compared to column family in Hbase.
* A record (after table joins) in RDBMS can be compared to a record in Hbase.
* A collection of tables in RDBMS can be compared to a table in Hbase.

**HBase Interview questions:**

1. What are the different commands used in HBase operations?

There are 5 atomic commands which carry out different operations by HBase.

Get, Put, Delete, Scan and Increment.

1. How to connect to HBase?

A connection to HBase is established through HBase Shell which is a Java API.

1. What is the role of Master server in HBase?

The Master server assigns regions to region servers and handles load balancing in the cluster.

1. What is the role of Zookeeper in HBase?

The zookeeper maintains configuration information, provides distributed synchronization, and also maintains the communication between clients and region servers.

1. When do we need to disable a table in HBase?

In HBase a table is disabled to allow it to be modified or change its settings. .When a table is disabled it cannot be accessed through the scan command.

1. Give a command to check if a table is disabled.

HBase > is\_disabled “table name”

1. What are the different types of filters used in HBase?

Filters are used to get specific data form an HBase table rather than all the records.

They are of the following types.

* Column Value Filter
* Column Value comparators
* KeyValue Metadata filters.
* RowKey filters.

1. Name three disadvantages HBase has as compared to RDBMS?

* HBase does not have in-built authentication/permission mechanism
* The indexes can be created only on a key column, but in RDBMS it can be done in any column.
* With one HMaster node there is a single point of failure.

1. What are catalog tables in HBase?

The catalog tables in HBase maintain the metadata information. They are named as −ROOT− and .META. The −ROOT− table stores information about location of .META> table and the .META> table holds information about all regions and their locations.

1. Is HBase a scale out or scale up process?

HBase runs on top of Hadoop which is a distributed system. Hadoop can only scale out as and when required by adding more machines on the fly. So HBase is a scale out process.

1. What are the step in writing something into HBase by a client?

In HBase the client does not write directly into the HFile. The client first writes to WAL (Write Access Log), which then is accessed by Memstore. The Memstore flushes the data into permanent memory from time to time.

1. What is compaction in HBase?

As more and more data is written to HBase, many HFiles get created. Compaction is the process of merging these HFiles to one file and after the merged file is created successfully, discard the old file.

1. What are the different compaction types in HBase?

There are two types of compaction. Major and Minor compaction. In minor compaction, the adjacent small HFiles are merged to create a single HFile without removing the deleted HFiles. Files to be merged are chosen randomly.

In Major compaction, all the HFiles of a column are emerged and a single HFiles is created. The deleted HFiles are discarded and it is generally triggered manually.

1. What is the difference between the commands delete column and delete family?

The Delete column command deletes all versions of a column but the delete family deletes all columns of a particular family.

1. What is a cell in HBase?

A cell in HBase is the smallest unit of a HBase table which holds a piece of data in the form of a tuple {row, column, version}

1. What is the role of the class HColumn Descriptor in HBase?

This class is used to store information about a column family such as the number of versions, compression settings, etc. It is used as input when creating a table or adding a column.

1. What is the lower bound of versions in HBase?

The lower bound of versions indicates the minimum number of versions to be stored in HBase for a column. For example If the value is set to 3 then three latest version wil be maintained and the older ones will be removed.

1. What is TTL (Time to live) in HBase?

TTL is a data retention technique using which the version of a cell can be preserved till a specific time period. Once that timestamp is reached the specific version will be removed.

1. Does HBase support table joins?

HBase does not support table joins. But using a mapreduce job we can specify join queries to retrieve data from multiple HBase tables.

1. What is a rowkey in HBase?

Each row in HBase is identified by a unique byte of array called row key.

1. What are the two ways in which you can access data from HBase?

The data in HBase can be accessed in two ways.

* Using the rowkey and table scan for a range of row key values.
* Using mapreduce in a batch manner.

1. What are the two types of table design approach in HBase?

They are − (i) Short and Wide (ii) Tall and Thin

1. In which scenario should we consider creating a short and wide HBase table?

The short and wide table design is considered when there is

* There is a small number of columns
* There is a large number of rows

1. In Which scenario should we consider a Tall-thin table design?

The tall and thin table design is considered when there is

* There is a large number of columns
* There is a small number of rows

1. How does HBase support Bulk data loading?

There are two main steps to do a data bulk load in HBase.

* Generate HBase data file (StoreFile) using a custom mapreduce job) from the data source. The StoreFile is created in HBase internal format which can be efficiently loaded.
* The prepared file is imported using another tool like complete bulk load to import data into a running cluster. Each file gets loaded to one specific region.

1. How does HBase provide high availability?

HBase uses a feature called region replication. In this feature for each region of a table, there will be multiple replicas that are opened in different RegionServers. The Load Balancer ensures that the region replicas are not co-hosted in the same region servers.

1. What is HMaster?

The HMaster is the Master server responsible for monitoring all RegionServer instances in the cluster and it is the interface for all metadata changes. In a distributed cluster, it runs on the NameNode.

1. What is HRegion Server in HBase?

HRegion Server is the Region Server implementation. It is responsible for serving and managing regions. In a distributed cluster, a Region Server runs on a DataNode.

1. How does WAL help when a RegionServer crashes?

The Write Ahead Log (WAL) records all changes to data in HBase, to file-based storage. if a RegionServer crashes or becomes unavailable before the MemStore is flushed, the WAL ensures that the changes to the data can be replayed.

1. Why MultiWAL is needed?

With a single WAL per RegionServer, the RegionServer must write to the WAL serially, because HDFS files must be sequential. This causes the WAL to be a performance bottleneck.