Answers

**1. Give a brief difference between HBASE and HDFS.**

Ans:

**HDFS:**

* Distributed file system abstracted on top of local file system by hadoop, suitable for storing huge files, it does not provide facility of tabular form of storage as such.
* HDFS is just a file system.
* Optimized for streaming access of large files.
* Follows write-once read-many ideology.
* Doesn't support random read/write.

**HBase:**

* It is column oriented distributed (on top of Hadoop) data store which runs on top of hdfs for providing structural data models. stores data in table row column
* It is now built on top of hdfs, you could compare this
* Stores key/value pairs in columnar fashion (columns are clubbed together as column families).
* Provides low latency access to small amounts of data from within a large data set.
* Provides flexible data model.

**2. List the main components of HBASE.**

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

**MasterServer**

* Assigns regions to the region servers and takes the help of Apache ZooKeeper 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.

**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.

When we take a deeper look into the region server, it contains regions and stores as shown below:



The store contains memory store 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.

**3. Does Hbase support sql?**

**Ans:** HBase does not support a structured query language like SQL.

**4. When should we use HBASE, list some of the scenarios for the same.**

**Ans:** Hbase is a distributed, scalable, big data store, modelled after Google's BigTable. It stores data as key/value pairs. It's basically a database, a **NoSQL database** and like any other database it's biggest advantage is that it provides you random read/write capabilities. Hadoop is not very good for your real time needs, so you can use Hbase to serve that purpose. If you have some data which you want to access real time, you could store it in Hbase. Hbase has got it's own set of very good API which could be used to push/pull the data. Not only this, Hbase can be seamlessly integrated with MapReduce so that you can do bulk operation, like indexing, analytics etc etc.

**5. What are the different modes in which HBase can be run?**

### Ans: The different mode are : - 1) Standalone HBase 2) Distributed

**6. Why is zookeeper needed in HBase?**

**Ans:**

**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.

**7 . HBase is a schema less database, what does it mean?**

**Ans:** Schema less database mean the "schema" is stored with the record, not the table. In a RDBMS, the schema is defined and that table has the schema. In HBase (and other BigTable implementations) data is labeled with its types.

**8. What is the minimum number of column family every HBase table should have?**

**Ans:** There is a limit to the number of column families in HBase. The more you add column families there will be more Memstore created and Memstore flush will be more frequent. It will degrade the performance.

**9. What is the benefit of using connection pool in HBase?**

**Ans:**