**Note: 7th Question is missing in the assignment, so it in not answered!**

## Give a brief difference between HBASE and HDFS.

Ans:

|  |  |
| --- | --- |
| HBASE | HDFS |
| Hbase is an open source, distributed, column oriented, No SQL database which run on top of HDFS for providing structural data models. | HDFS is distributed file system to store huge files in redundant storage space. |
| Hbase system comprises a set of tables which contains rows and columns which can process real time datasets. | HDFS consists of files which are broken into blocks and distributed across nodes in the cluster. |
| HBase stores data in the form of key value pair in a columnar fashion. | It is optimized for streaming access of very large files. |
| It gives ability to do random read/writes on your data. | HDFS is optimized for use cases where write once and read many times. It is not for random read/writes. |
| HBase is ideal for random access queries, update and inserts but not good for analytics. | HDFS is for high speed writes and scans but updates are slow and cumbersome. |

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

Ans: HBase has three major components: the client library, a master server, and region servers.

Master server : Assigns regions to the region servers with the help of Apache Zookeeper. It coordinates with region servers, assign regions on start up, reassign regions for recovery, and monitor all regionserver instances in the cluster. It handles load balancing of the regions across region servers. It unloads the busy servers and shifts the regions to less occupied servers. So, it maintains the state of the cluster by negotiating the load balancing. It is responsible for metadata operations, region assignment, DDL operations such as creation of tables and column families.

Region server: It serves data for read and write. For accessing data, client communicates with region servers directly. Region server are the nodes that consists of region that contains all rows in the table. It decided the size of the regions by following the region size thresholds.

Zookeeper: Zookeeper is the coordinator. ZooKeeper is 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.

1. **Does Hbase support sql?**

Ans: Yes Hbase support sql through apache Phoenix. Apache Phoenix takes your SQL query, compiles it into a series of HBase scans, and orchestrates the running of those scans to produce regular JDBC result sets. Direct use of the HBase API, along with coprocessors and custom filters, results in performance on the order of milliseconds for small queries, or seconds for tens of millions of rows. All standard SQL query constructs are supported, including SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY, etc. It also supports a full set of DML commands as well as table creation and versioned incremental alterations through our DDL commands.

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

Ans: We should use HBase in following scenarios:

1. When we require high write throughput.
2. When we require good random read performance compared to other databases.
3. When we need horizontal scalability.
4. When we require automatic failover recovery.
5. When we require strong consistency for read/write operations.
6. When we require benefits of HDFS like distributed storage, fault tolerance, scalability, checksums, mapreduce.
7. When your application has a variable schema where each row is slightly different like when you want to add columns fast enough and many of them are null in rows.
8. When your data is stored in collections like message data or binary data.
9. If you need key key based access to data when storing and retrieving.
10. When you need real time processing. It supports block cache and bloom filters, so real time query processing is easy.
11. When we require API support. HBase support JAVA APIs so clients can access it easily.
12. When data volume is huge like petabytes of data to process.
13. Your application do not require RDBMS features like transaction, triggers, complex query, complex joins.

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

Ans: HBase has two run modes – Standalone and Distributed.

Standalone is the default mode. In it HBase does not use HDFS instead it uses the local filesystem. It runs all HBase daemons and a local Zookeeper in the same JVM. Zookeeper binds to a well-known port so clients may talk to HBase.

Distributed mode can be subdivided into pseudo distributed and fully- distributed.

Pseudo distributed – All daemons run in a single node. A pseudo-distributed mode is simply a distributed mode run on a single host.

Fully-distributed – Daemons are spread across all nodes in the cluster.

Distributed mode requires an instance of the HDFS.

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

Ans: Zookeeper is the coordinator. It is used as a distributed coordination service to maintain server state in the cluster. Zookeeper maintains which server is alive and available, and provides server failure notification. It uses consensus to guarantee common shared state. It is used to coordinate shared state information for members of distributed systems. Region servers and the active HMaster connect with a session to Zookeeper. The Zooker maintains ephemeral nodes for active sessions via heartbeats. Each region server creates an ephemeral node. Zookeeper determines the ephemeral node to make sure only one master is active. The active HMaster sends heartbeat to Zookeeper and inactive HMaster listen for notifications of active HMaster failure.

**8 Hbase is a schema less database, what does it mean?**

Ans: Schema is stored with the record, not the table. In RDBMS, the schema is defined and that table has the schema. In HBase, data is labeled with its types. It doesn't have the concept of fixed columns schema; defines only column families. The column names can be completely variable and the number of columns can vary by row – so you could have a table with billions of rows and could have rows with 5 or 5 million columns.

**9. What is the minimum number of column family every Hbase table should have?**

Ans: One column family. Only introduce a second and third column family in the case where data access is usually column scoped; i.e. you query one column family or the other but usually not both at the one time.

**10 What is the benefit of using connection pool in Hbase?**

Connection Pooling : For applications which require high-end multithreaded access (e.g., web-servers or application servers that may serve many application threads in a single JVM), you can pre-create a Connection, as shown in the following example:

Example : Pre-Creating a Connection

// Create a connection to the cluster.

Configuration conf = HBaseConfiguration.create();

try (Connection connection = ConnectionFactory.createConnection(conf)) {

try (Table table = connection.getTable(TableName.valueOf(tablename)) {

// use table as needed, the table returned is lightweight

}

}

Constructing HTableInterface implementation is very lightweight and resources are controlled.