**ASSIGNMENT 12.2**

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

HDFS(Hadoop Distributed File System) is a distributed file system abstracted on top of the local file system by the hadoop which does not provide tabular form of storage while the HBASE is a distributed column oriented database running on top of HDFS that provide tabular form of storage of real-time semi and unstructured bigdata.

a)HDFS is meant for storing massive amounts of data across a distributed system.

b)HBase is a non-relational database that can run on top of Hadoop and provides you random data access/querying capabilities. HDFS, by itself has no support for reads/writes at random location.

c)HBase stores data as key/value pairs as in a column database (something similar to Cassandra DB) while data, in HDFS is stored as flat files.

**2. List and explain components of HBASE.**

The 3 important components of HBASE are:

The Hadoop DataNode stores the data that the Region Server is managing.

The NameNode maintains metadata information for all the physical data blocks.

HMaster

Region Server

ZooKeeper.

* HMaster

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

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 Servers

Region servers serve data for reads and writes.

Regions are assigned to the Regoin Servers.A region server can serve about 1,000 regions.These are the worker nodes which handle read, write, update, and delete requests from clients. Region Server process, runs on every node in the hadoop cluster. Region Server runs on HDFS DataNode and consists of the following components –

Block Cache – This is the read cache. Most frequently read data is stored in the read cache and whenever the block cache is full, recently used data is evicted.

MemStore- This is the write cache and stores new data that is not yet written to the disk. Every column family in a region has a MemStore.

Write Ahead Log (WAL) is a file that stores new data that is not persisted to permanent storage.

HFile is the actual storage file that stores the rows as sorted key values on a disk.

* Zookeeper

Zookeeper maintains a live cluster state.

HBase uses ZooKeeper as 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.

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.

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

HBASE should be used when the data has:

A variable schema Where data is stored in the form of collections

If the application demands key based access to data while retrieving.

HBase is an ideal platform with ACID compliance properties making it a perfect choice for high-scale, real-time applications.

It does not require a fixed schema, so developers have the provision to add new data as and when required without having to conform to a predefined model.

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

HBASE can run in two different modes

Standalone mode

Distributed mode

By default HBASE is in standalone mode to run in distributed mode we need to edit the configuration in conf directory.

* Standalone HBase

This is the default mode.

In standalone mode, HBase does not use HDFS, it uses the local filesystem instead and it runs all HBase daemons and a local ZooKeeper all up in the same JVM.

Zookeeper binds to a well known port so clients may talk to HBase.

* Distributed HBase

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

A pseudo-distributed mode is simply a distributed mode run on a single host. This configuration is used for testing and prototyping on HBase.

A fully-distributed mode can be run on more than one host, using the following configurations. In hbase-site.xml, add the property hbase.cluster.distributed and set it to true and point the HBase hbase.rootdir at the appropriate HDFS NameNode and location in HDFS where we would like HBase to write data.

**5. Why is zookeeper needed in Hbase?**

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.

Because it guarantees

Sequential Consistency - Updates from a client will be applied in the order that they were sent.

Atomicity - Updates either succeed or fail. No partial results.

Single System Image - A client will see the same view of the service regardless of the server that it connects to.

Reliability - Once an update has been applied, it will persist from that time forward until a client overwrites the update.

Timeliness - The clients view of the system is guaranteed to be up-to-date within a certain time bound.

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

Hbase doesnot have fixed column specification we define only column families as a key value pair , therfore Hbase is a schemaless database.This means that 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.

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

Every Hbase table must have a minimum of one column family.

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

With connection pooling we can pre-create a connection for applications which require high-end multithreaded access.

**9. What is the difference between memstore and hfile in HBase?**

* MemStore: It is the write cache. It stores new data which has not yet been written to disk. It is sorted before writing to disk.
* HFiles: Stores the rows as sorted KeyValues on disk. File format for hbase.
* A file of sorted key/value pairs.
* Both keys and values are byte arrays.
* When the MemStore accumulates enough data, the entire sorted KeyValue set is written to a new HFile in HDFS.
* This is a sequential write.It is very fast, as it avoids moving the disk drive head.

**10.Describe compactions in HBase**

There are 2 type of compaction in HBase

* Minor Compaction:

HBase will automatically pick some smaller HFiles and rewrite them into fewer bigger Hfiles.

Minor compaction reduces the number of storage files by rewriting smaller files into fewer but larger ones, performing a merge sort.

* Major Compaction:

Major compaction merges and rewrites all the HFiles in a region to one HFile per column family, and in

the process, drops deleted or expired cells.

This improves read performance; however, since major compaction rewrites all of the files, lot of disk I/O

and network traffic might occur during the process.

**11.List and explain the logical entities in HBase.**

\*Table - HBase organizes data into Tables.

\*Row - Within a table, data is stored according to its row.

\*RowKey - Rows are identified uniquely by their row key

\*Column Family - Data within a row is grouped by column family

\*Column Name - Data within a column family is addressed via its column name.

\*Cell - A combination of row key, column family, and column qualifier uniquely identifies a cell

\*TimeStamp - Values within a cell are versioned.

**12.What will happen if we do not create a row key while inserting the data?**

It is not possible to insert without rowkey.

**13.How can filters be applied in HBase and what are the benefits?**

\*KeyOnlyFilter - Returns the key component of each key-value.

\*FirstKeyOnlyFilter - Returns the first key-value from each row.

\*Prefixfilter

Returns those key-values present in a row that starts with the specified row prefix

\*ColumnPrefixFilter - Returns those key-values present in a column that starts with the specified column prefix.

\*ColumnCountGetFilter - Returns the first limit number of columns in the table.

\*PageFilter - Returns page size number of rows from the table.

\*InclusiveStopFilter - Returns all key-values present in rows up to and including the specified row.

\*Family Filter - Compares each qualifier name with the comparator using the compare operator and if the comparison returns true, it returns all the key-values in that column.

\*ValueFilter - Compares each value with the comparator using the compare operator and if the comparison returns true, it returns that key-value.

**14.What are the data model operations in hBase?**

The data model operations in HBase are:

* Get - Get returns attributes for a specified row.
* Put –Put  either adds new rows to a table (if the key is new) or can update existing rows (if the key already exists).
* Scan - [Scan](http://hbase.apache.org/apidocs/org/apache/hadoop/hbase/client/Scan.html) allow iteration over multiple rows for specified attributes.
* Delete - [Delete](http://hbase.apache.org/apidocs/org/apache/hadoop/hbase/client/Delete.html) removes a row from a table.

**15.How MapReduce can be used with HBase?**

We can run mapreduce jobs that uses HBase by adding the HBase and Zookeeper JAR files to the Hadoop Java classpath.

HBase provides a TableInputFormat, to which you provided a table scan, that splits the rows resulting from the table scan into the regions in which those rows reside.

The map process is passed an ImmutableBytesWritable that contains the row key for a row and a Result that contains the columns for that row.

The map process outputs its key/value pair based on its business logic in whatever form makes sense to your application.

The reduce process builds its results but emits the row key as an ImmutableBytesWritable and a Put command to store the results back to HBase.

Finally, the results are stored in HBase by the HBase MapReduce infrastructure.

**16.What is regionserver?**

Region servers serve data for reads and writes.

Regions are assigned to the Regoin Servers.A region server can serve about 1,000 regions.

These are the worker nodes which handle read, write, update, and delete requests from clients. Region Server process, runs on every node in the hadoop cluster. Region Server runs on HDFS DataNode and consists of components namely block cache and memstore