|  |
| --- |
| 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 realtime semi and unstructed bigdata. |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| 2. components of HBASE. |
|  |

|  |
| --- |
| The components of HBASE are: |
|  |

|  |
| --- |
| Region servers serve data for reads and writes. |
|  |

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

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

|  |
| --- |
| Zookeeper maintains a live cluster state. |
|  |

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

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

|  |
| --- |
|  |
|  |

|  |
| --- |
| 3. use of HBASE and its scenarios |
|  |

|  |
| --- |
| HBASE should be used when the data has: |
|  |

|  |
| --- |
| A variable schema |
|  |

|  |
| --- |
| When data is stored in the form of collections |
|  |

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

|  |
| --- |
|  |
|  |

|  |
| --- |
| 4. different modes of Hbase |
|  |

|  |
| --- |
| 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. |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| 5.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. |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| 6. Hbase - a schema less database |
|  |

|  |
| --- |
| Hbase doesnot have fixed column specification we define only column families as a key value pair , therfore |
|  |

|  |
| --- |
| Hbase is a schemaless database. |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| 7. minimum number of column family |
|  |

|  |
| --- |
| Every Hbase table must have atleast 1 column family. |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| 8. 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. 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. |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| 10.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. 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. not creating a row key while inserting the data |
|  |

|  |
| --- |
| It is not possible to insert without rowkey. |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| 13.filters applied in HBase and their 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.data model operations in hBase |
|  |

|  |
| --- |
| The data model operations in HBase are: |
|  |

|  |
| --- |
| Get |
|  |

|  |
| --- |
| Put |
|  |

|  |
| --- |
| Scan |
|  |

|  |
| --- |
| Delete |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| 15. MapReduce used with HBase |
|  |

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

|  |
| --- |
|  |
|  |

|  |
| --- |
| 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.