Vijay's Assignment – Hbase1

Task 1

Answer in your own words with example.

1.What is NoSQL data base?

NoSQL DB are non relational databases provide mechanism to store and retireve data that is different from the traditional tabular format used by RDBMS. NoSQL is designed to cater to the fast expanding structured, unstructured and hybrid data from various sources like mobile, social media etc that cannot be stored in RDBMS efficiently. NoSQL DB are designed to be implemented in cheap low cost commoditiy hardware like cloud. NoSQL DBs vary from traditional RDBMS (SQL DBs) on these factors

a) Many different types of DB in NoSQL compared to one SQL DB
b) Dynamic schema in NoSQL compared to static schema in SQL DB
c) NoSQL is highly scalable to dynamic workloads
d) NoSQL DBs provide excellent support for unstrcutured data compared to tranditional RDBMS

Ex: Mongo DB, Cassandra, Hbase etc

2.How does data get stored in NoSQl database?

In NoSQL data gets stored based on the type of database

a) Key-value: Data is stored as a collection of key and value pairs
b) Document store: Data is stored in some standard formats like XML, JSON etc. Documents are addressed with a unique key within the DB
c) Graph: Stores data which are interconnected among themselves with finite relations between them
d) Wide colum: Stores columns of data together instead of rows

3.What is a column family in HBase?

Column family in HBase is a collection of columns and their values. Hbase stores data in column family for performance reasons. Each column family has set of storage properties such as data cache in memory, data compression etc. Each row in the table has same column family although it may be empty

4.How many maximum number of columns can be added to HBase table?

There is actually no limit on the number of columns in a HBase table. Column family is limited to only three.

5.Why columns are not defined at the time of table creation in HBase?

Data within a row in HBase is grouped by Column family. Column family also influence the physical arrangement of data stored in HBase. For this reason, they must be defined up front and are not easily modified. Columns within a column family can be specified at run time as it does not influence the way data is stored in HBase.

6.How does data get managed in HBase?

The HBase Physical Architecture consists of servers in a Master-Slave relationship.Typically, the HBase cluster has one Master node, called HMaster and multiple Region Servers called HRegionServer. Each Region Server contains multiple Regions – HRegions.

Data in HBase is stored in Tables. These tables are stored in multiple regions based on its size. Table data is stored only in HRegions within HFiles. The mapping of Region server to Region is kept in a table called .META. present with the zookeeper. Clients access this META table to know the region server to work with and directly communicates with it.

HBase is a columnar type DB. Hence each row in a table is a collection of column families. Each column family could have multiple columns and each row need not have value for all the column families. Each row in HBase has a unique row key and that is used while retrieving the data. A cell is a unique combination of row key, column family, columns and its values. Each cell is also versioned.

7.What happens internally when new data gets inserted into HBase table?

Data is HFile is always in a sorted order. Hence random inserts cannot be allowed in HFile. HBase tackles this problem with the implementation of Memstore and Write Ahead Logs. Following is the process in short

a) New data from the client is written first to the write ahead log of that region server
b) Then the data is copied over to memstore
c) Data in the memstore is sorted in the same order as HFile
d) Once memstore has enough data it is stored in HFile of that region.
e) An acknolwedgement of write task completion is sent back to the client

Task 2

1. Create an HBase table named 'clicks' with a column family 'hits' such that it should be

able to store last 5 values of qualifiers inside 'hits' column family.

```
hbase(main):043:0> create 'clicks', {NAME=>'hits', VERSIONS=>'5'}
0 row(s) in 1.2340 seconds
```

```
hbase(main):062:0> describe 'clicks'
Table clicks is ENABLED
clicks
COLUMN FAMILIES DESCRIPTION
{NAME => 'hits', BLOOMFILTER => 'ROW', VERSIONS => '5', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCOD
ING => 'NONE', TTL => 'FOREVER', COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPL
ICATION_SCOPE => '0'}
1 row(s) in 0.1100 seconds
```

2. Add few records in the table and update some of them. Use IP Address as row-key. Scan the table to view if all the previous versions are getting displayed.

Inserts

```
=> Hbase::Table - clicks
hbase(main):044:0> put 'clicks', '10.255.10.1', 'hits:address', '91011'
0 row(s) in 0.0290 seconds

hbase(main):045:0> put 'clicks', '10.255.10.1', 'hits:count', '3'
0 row(s) in 0.0130 seconds

hbase(main):046:0> put 'clicks', '10.255.10.2', 'hits:address', '91012'
0 row(s) in 0.0130 seconds

hbase(main):047:0> put 'clicks', '10.255.10.2', 'hits:count', '2'
0 row(s) in 0.0260 seconds

hbase(main):048:0> put 'clicks', '10.255.10.3', 'hits:count', '4'
0 row(s) in 0.0150 seconds

hbase(main):049:0> put 'clicks', '10.255.10.3', 'hits:address', '91010'
0 row(s) in 0.0190 seconds
```

Updates

```
hbase(main):054:0> put 'clicks', '10.255.10.3', 'hits:count', '6'
0 row(s) in 0.0160 seconds

hbase(main):055:0> put 'clicks', '10.255.10.3', 'hits:address', '91015'
0 row(s) in 0.0130 seconds

hbase(main):056:0> put 'clicks', '10.255.10.2', 'hits:count', '9'
0 row(s) in 0.0170 seconds

hbase(main):057:0> put 'clicks', '10.255.10.1', 'hits:count', '10'
0 row(s) in 0.0140 seconds
```

Scan with version

```
hbase(main):063:0> scan 'clicks', {VERSIONS=>2}
ROW                           COLUMN+CELL
 10.255.10.1                  column=hits:address, timestamp=1546973648762, value=91011
 10.255.10.1                  column=hits:count, timestamp=1546973792592, value=10
 10.255.10.1                  column=hits:count, timestamp=1546973658941, value=3
 10.255.10.2                  column=hits:address, timestamp=1546973668708, value=91012
 10.255.10.2                  column=hits:count, timestamp=1546973783957, value=9
 10.255.10.2                  column=hits:count, timestamp=1546973679601, value=2
 10.255.10.3                  column=hits:address, timestamp=1546973776984, value=91015
 10.255.10.3                  column=hits:address, timestamp=1546973698585, value=91010
 10.255.10.3                  column=hits:count, timestamp=1546973766811, value=6
 10.255.10.3                  column=hits:count, timestamp=1546973686893, value=4
3 row(s) in 0.0910 seconds
```