**Session 09: HBase**

**Assignment 1**

1. **What is NoSQL data base?**

A NoSQL database is, simply put, a non-relational and largely distributed database system that enables rapid, ad-hoc organization and analysis of extremely high-volume, disparate data types. NoSQL databases are sometimes referred to as cloud databases, non-relational databases, Big Data databases and were developed in response to the sheer volume of data being generated, stored and analyzed by modern users (user-generated data) and their applications (machine-generated data).

In general, NoSQL databases have become the first alternative to relational databases, with scalability, availability, and fault tolerance being key deciding factors.

1. **How does data get stored in NoSQl database?**

**Columnar store**

-Based on Google's BigTable

-Data model is similar to Relational databases except that data is stored by column types rather than by rows (think of an index)

-Faster queries for aggregation types

-**When to use this kind of database:** When we run a lot of aggregation queries on a few columns of data ,Columnar data compression

-**When not to use this kind of database:** Lots of incremental data loadsOLTPQueries against row data rather than a few column

Ex: Hbase, Cassandra, BigTable

**Key-Value Stores**

-Focus on scaling to huge amounts of Data

-Designed to handle a massive load

-Based on Amazon's Dynamo whitepaper

-Data model is a collection of Key-Value pairs that can be replicated, and partitioned for best performance

**-When to use this kind of database:** Quick storage of non-structured data,Quick access and inserts of any kind of data

-**When not to use this kind of database:** Need for durability and consistency

Ex:Voldemort

**Graph store**

-Focused on modeling interconnectivity between data

- Inspired by the mathematical Graph theory (G = E(V) )

-Data model is a property graph where the relationships and the strength between the nodes is saved in addition to the value stored in the node itself.

**-When to use this kind of database:** Relationship between data needs to be stored and retrieved quickly

**-When not to use this kind of database**: Need for structured data

Ex: Giraph, Neo4j

**Document store**

-Similar to Key-Value stores, but the db knows what the value is

- Data model is a collection of Key-Value pairs

-Value is often specified in the form of Json objects (which can be nested as necessary for complexity)

-Version control for Documents is usually built in

**-When to use this kind of database:** Quick storage and retrieval of quasi/semi-structured data required

-**When not to use this kind of database:** Quick Storage and retrieval of well-formed data

Ex: CouchDB,MongoDB

**3. What is a column family in HBase?**

In the HBase data model columns are grouped into column families, which must be defined up front during table creation. Column families are stored together on disk, which is why HBase is referred to as a column-oriented data store.

When creating a table in HBase, the developer or administrator is required to define one or more column families using printable characters.

Generally, column families remain fixed throughout the lifetime of an HBase table but new column families can be added by using administrative commands. The official recommendation for the number of column families per table is three or less.

In addition, you should store data with similar access patterns in the same column family — you wouldn’t want a customer’s middle name stored in a separate column family from the first or last name because you generally access all name data at the same time.

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

 There is no hard limit to number of columns in HBase , we can have more than 1 million columns but usually three column families are recommended ( not more than three).

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

We cannot provide column-qualifier during table creation. This is what makes HBase schema-less. The syntax for creating tables in HBase is as follows.

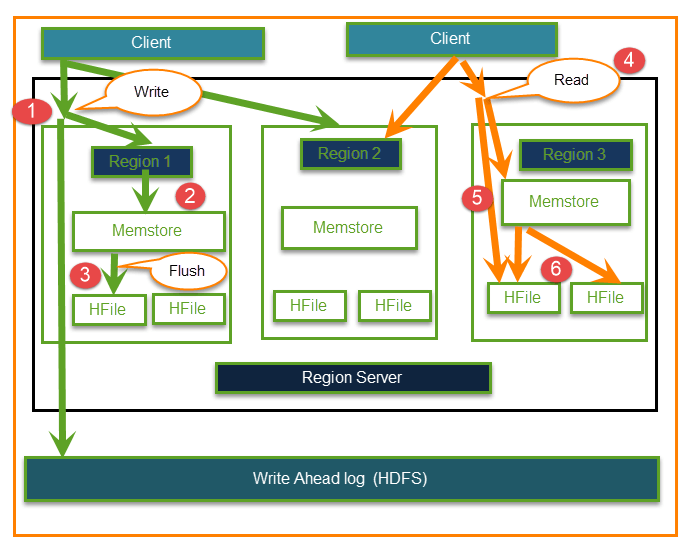
create ‘<table name>’,’<column family>’

Once we have created the table, we can add rows which should belong to at-least one column-family in a table and then can belong to any column-qualifier. If the column-qualifier is not already present, it will be created.

**6. How does data get managed in HBase?**

Data can be managed in HBase by using the **create, get, put and scan** commands from the HBase shell. Data is written to the database by using put and read by using **get**. The **scan** command is used to obtain data from multiple rows in a table. Data can also be managed using the HBase C# API, which provides a client library on top of the HBase REST API. An HBase database can also be queried by using Hive. Co-processors are also available, which allow data processing in the nodes that host the database.

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



**Write operation**

**Step 1)** Client wants to write data and in turn first communicates with Regions server and then regions

**Step 2)** Regions contacting memstore for storing associated with the column family

**Step 3)**First data stores into Memstore, where the data is sorted and after that it flushes into HFile. The main reason for using Memstore is to store data in Distributed file system based on Row Key. Memstore will be placed in Region server main memory while HFiles are written into HDFS.

**Read operation**

**Step 4)**Client wants to read data from Regions

**Step 5)**In turn Client can have direct access to Mem store, and it can request for data.

**Step 6) Client** approaches HFiles to get the data. The data are fetched and retrieved by the Client.