Assignment 11

Task1:

Explain the below concepts with an example in brief.

NoSQL Databases:

Ans: NoSQL stands for Not only SQL. It is an approach to to design database that can accommodate wide variety of data models, including key-value, document, columnar and graph formats. It is an alternative for traditional relational databases in which data is placed in tables and data schema is carefully designed before the database is built. NoSQL databases are especially useful for working with large datasets of distributed data and store structured, semi-structured and unstructured data. Ex: Hbase, MongoDB, Cassandra etc.

Types of NoSQL databases:

Ans: NoSQL database are of following types

- Key-value database:
  Key-value database are the simplest NoSQL database to use. Every item in the database is stored as "key" along with its value. The value is the blob that the data store just stores without caring or knowing what's inside; it is the responsibility of an application to understand what was stored. Since key-value stores always use primary key access, they generally have great performance and highly scalable for session management and caching in web-applications. The key-value database uses the hash-table to store unique keys and pointers with respect to each data value it stores. There are no column type relations in the database; hence, its implementation is easy. Key-value databases gives great performance and can be very easily scaled as per the business needs.

  Ex:

KEY	**VALUE**
"India"	{"C-101, Sector 123, New Delhi"}
"America"	{"1-2-345, ABC colony, XYZ Area, Mumbai, Maharashtra"}

- Document Store NoSQL database:
  Document store NoSQL databases are similar to the Key-value databases. In Key-value NoSQL databases there's a key and a value where data is stored as a value. And its associated key is the unique identifier for that value. The only difference between these 2 databases is that, in a

document database, the value contains structured or semi structured data. This structured/semi structured value is referred to as a document or it can be in XML, JSON or BSON format.

Couchbase and *MongoDB* are the most popular document store NoSQL databases

Ex:

KEY	VALUE
Ravi@gmail.com	{Name:”Ravi”, age:30, city:”Hyd” }
Miller@outlook.com	{Name:”Miller”, age:16, city:New York” }

- Column store NoSQL database:
  In column-oriented NoSQL databases, data is stored in cells grouped in columns of data rather than as rows of data. Columns are logically grouped into column families.

  Column families can contain virtually unlimited number of columns that can be created at the run time or while defining the schema. Read and write is done using columns rather than rows.

  Column families are the groups of similar data that is usually accessed together.

  Ex:

  Ctable --------------->tablename

Key	**value**
101	name:”Sumanth”
101	sal:30000
101	city:”Hyd”
102	name:”Santhosh”
102	city:”pune”
102	desig:”Manager”

- Graph Base NoSQL database:
  In a graph base NoSQL database, you will not find rigid format of SQL or the tables and columns representation, a flexible graphical representation is instead used which is perfect to address scalability concerns. Graph structures are used with edges, nodes and properties which provides index-free adjacency. Data can be easily transformed from one model to another using a Graph base NoSQL database.

Ex: facebook friends

User Friend

Ravi----->Vani

Vani----->Veni

Veni----->Sony

Sony----->Ravi

CAP Theorem

Ans: In CAP ‘C’ stands for “Consistency”, ‘A’ stands for “Availability” and ‘P’ stands for “Partition Tolerance”.

CAP can help in choosing the system as per your requirement. You cannot build a general data store that is continually available, sequentially consistent and tolerant to any partition pattern. You can build one that has any of these three properties.

Consistency- This means the data in the database remains consistent after the execution of the operation. For example after the partition update operation, all clients see the same data. (Duplication is not maintained)

Availability- This means the system is always on (service guarantee availability), no downtime (reduces consistency).

Partition Tolerance- This means the system means to continues to function even if the communication among the servers is unreliable i.e. the servers may be partitioned into multiple groups that cannot communicate with one another. (Result may not be available for some records in case of failure).

Hbase relies on “AP” of CAP.

Hbase Architecture:

Ans: Hbase architecture is partially similar to that of map-reduce. It has 3 components in a master-slave setup.

Region Server sits where data node exists, and server data read write functionality/availability. Datanode stores the data managed by it on HDFS.

HBase Master handles regions assignment, creation/deletion/updation of tables, databases. It assigns region to the new data created, coordinates between the region servers and monitor all region servers as “Namenode does for datanodes”

Zookeper helps in maintaining the live state of the cluster. It is a centralized service for maintaining configuration information, naming, and providing distributed synchronization between the components.

<u>Hbase vs RDBMS:</u>

Ans: RDBMS is a row oriented database, where Hbase is a column oriented one which uses unique row-key to store data.

In RDBMS, the schema for the table is fixed, there is no fixed schema in Hbase, the number of columns and types of columns need not to be same for all row-keys.

RDBMS strictly follows ACID properties whereas Hbase promises consistency and partition tolerance. RDBMS uses SQL to query the data where Hbase uses java client API for the same.

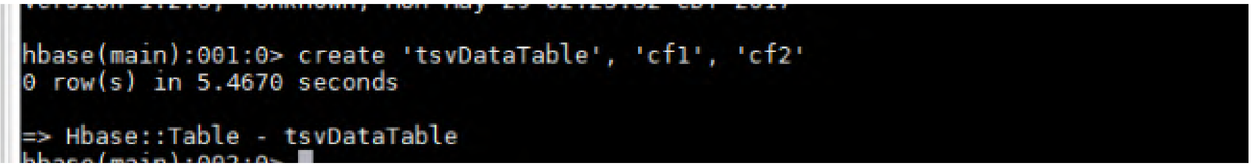
<u>Task2:</u>

Execute blog present in below link:
https://acadgild.com/blog/importtsv-data-from-hdfs-into-hbase/

Solution:

1. Creating hbase table:

create 'tsvDataTable', 'cf1', 'cf2'

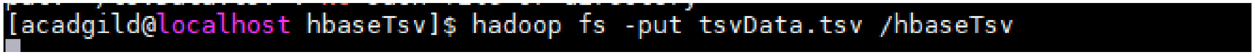


```
hbase(main):001:0> create 'tsvDataTable', 'cf1', 'cf2'
0 row(s) in 5.4670 seconds

=> Hbase::Table - tsvDataTable
hbase(main):002:0>
```

2. Creating hbase dir in HDFS
hadoop fs -mkdir /hbaseTsv

3. Putting tsv file in HDFS



```
[acadgild@localhost hbaseTsv]$ hadoop fs -put tsvData.tsv /hbaseTsv
```

4. Running import command

hbase org.apache.hadoop.hbase.mapreduce.ImportTsv –Dimporttsv.columns=HBASE_ROW_KEY,cf1:name,cf2:id tsvDataTable /hbaseTsv/tsvData.tsv