Assignment Submission- Session 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 design database that can accommodate a wide variety of data models, including key-value, document, columnar and graph formats. It is an alternative to 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 sets of distributed data and store structured, semi-structured or 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 a "key" together with its value. The value is a blob that the data store just stores, without caring or knowing what's inside; it's the responsibility of the application to understand what was stored. Since key-value stores always use primary-key access, they generally have great performance and are highly scalable for session management and caching in web applications. The key-value database uses a 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 give great performance and can be very easily scaled as per business needs.

Document store NoSQL database:

Document store NoSQL databases are similar to key-value databases in that there's a key and a value. Data is stored as a value. Its associated key is the unique identifier for that value. The difference 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 and can be in XML, JSON or BSON format.

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 a virtually unlimited number of columns that can be created at runtime or while defining the schema. Read and write is done using columns rather than rows. Column families are groups of similar data that is usually accessed together.

● CAP Theorem

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

CAP theorem can help in choosing the system as per our 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 two of these three properties.

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

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

Partition Tolerance - This means that the system 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 somewhat similar to how map-reduce works. It has 3 components in a master-slave setup.
Region server sits where datanode exists, and serve 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, providing distributed synchronization between the components.

● HBase vs RDBMS

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.

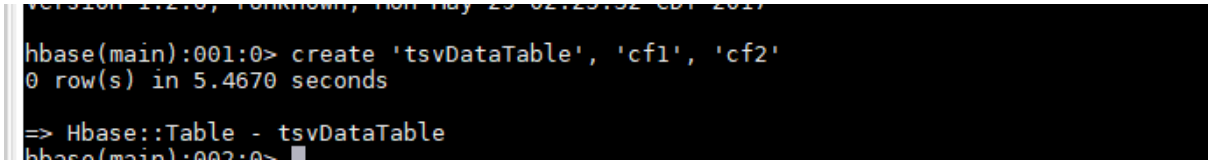
Task2
Execute blog present in below link:

https://acadgild.com/blog/importtsv-data-from-hdfs-into-hbase/

Solution:

1. Creating hbase table:

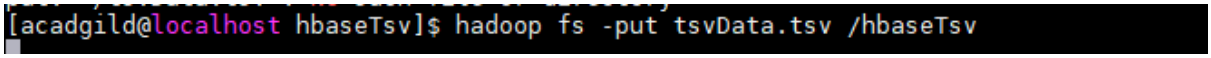
create 'tsvDataTable', 'cf1', 'cf2'



2. Creating hbase dir in HDFS

hadoop fs -mkdir /hbaseTsv

3. Putting tsv file in HDFS



4. Running import command

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