Assignment\_17.1

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

NoSQL and Not Only SQL describe an approach to database design that implements a key-value store, document store, column store or graph format for data. It is an alternative to the Structured Query Language (SQL) database prevalent beginning in the 1980s. NoSQL contrasts to databases that adhere to SQL's relational methods, where data are placed in tables and data schema are carefully designed before the database is built. NoSQL databases especially target large sets of distributed data.

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

Key/Value Databases- In this model, data is represented as a collection of key-value pairs. Eg: Voldemort, Redis, Scalaris

Columnar Databases- Stores columns of data together than rows. Query performance is often increased as a result, particularly in very large data sets. eg -HBase, Hypertable, Cassandra

Document Databases -Document-oriented databases are inherently a subclass of the key-value store. A document can have any number of key pair values or key array values. The difference lies in the way the data is processed; in a key-value store the data is considered to be inherently opaque to the database, whereas a document-oriented system relies on internal structure in the document in order to extract metadata that the database engine uses for further optimization. eg -MongoDb, CouchDB

Graph Databases- data whose relations are well represented as a graph consisting of elements interconnected with a finite number of relations between them. The type of data could be social relations, public transport links, road maps or network topologies. eg - InfoGrid, Neo4j

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

Columns in Apache HBase are grouped into column families. All column members of a column family have the same prefix. For example, the columns courses:history and courses:math are both members of the courses column family. The colon character (:) delimits the column family from the . The column family prefix must be composed of printable characters. The qualifying tail, the column family qualifier, can be made of any arbitrary bytes. Column families must be declared up front at schema definition time whereas columns do not need to be defined at schema time but can be conjured on the fly while the table is up an running.

Physically, all column family members are stored together on the filesystem. Because tunings and storage specifications are done at the column family level, it is advised that all column family members have the same general access pattern and size characteristics

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

There are no limits to number of columns. Developers can choose either

1) more rows with less columns

2) more columns with less rows

hbase.table.max.rowsize

Description

Maximum size of single row in bytes (default is 1 Gb) for Get’ting or Scan’ning without in-row scan flag set. If row size exceeds this limit RowTooBigException is thrown to client.

Default

1073741824

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

HBase tables are created with column families which generally is not changed. The column keys are specified after the table is up and running. Operations such as tunings and storage specifications are done at the column family level and are hence stored in separate HFiles.

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

Hbase is natively supported on Hadoop and it is the subject of this tutorial. The main characteristics that make Hbase an excellent data management platform are fault tolerance, speed and usability. Fault tolerance is provided by automatic fail-over, automatically sharded and load balanced tables, strong consistency in row level operations and replication. Speed is provided by almost real time lookups, in memory caching and server side processing. Usability is provided by a flexible data model that allows many uses, a simple Java API and ability to export metrics.

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

To write data to HBase, you use methods of the HTableInterface class. You can use the Java API directly, or use HBase Shell, Thrift API, REST API, or another client which uses the Java API indirectly. When you issue a Put, the coordinates of the data are the row, the column, and the timestamp. The timestamp is unique per version of the cell, and can be generated automatically or specified programmatically by your application, and must be a long integer