

Session 10:

HBASE BASICS

Assignment 1

Task 1

Answer in your own words with example.

1)What is NoSQL data base?

NoSQL is a class of database management systems (DBMS) that do not follow all of the rules of a relational DBMS and cannot use traditional SQL to query data. The term is somewhat misleading when interpreted as "No SQL," and most translate it as "Not Only SQL," as this type of database is not generally a replacement but, rather, a complementary addition to RDBMSs and SQL.

NoSQL-based systems are typically used in very large databases, which are particularly prone to performance problems caused by the limitations of SQL and the relational model of databases. Many think of NoSQL as the modern database of choice that scales with Web requirements. Some notable implementations of NoSQL are Facebook's Cassandra database, Google's BigTable and Amazon's SimpleDB and Dynamo.

2.How does data get stored in NoSQL database?

Several different varieties of NoSQL databases have been created to support specific needs and use cases. These fall into four main categories:

- Key-value data stores: Key-value NoSQL databases emphasize simplicity and are very useful in accelerating an application to support high-speed read and write processing of non-transactional data. Stored values can be any type of binary object (text, video, JSON document, etc.) and are accessed via a key. The application has complete control over what is stored in the value, making this the most flexible NoSQL model. Data is partitioned and replicated across a cluster to get scalability and availability. For this reason, key value stores often do not support transactions. However, they are highly effective at scaling applications that deal with high-velocity, non-transactional data.
- Document stores: Document databases typically store self-describing JSON, XML, and BSON documents. They are similar to key-value stores, but in this case, a value is a single document that stores all data related to a specific key. Popular fields in the

document can be indexed to provide fast retrieval without knowing the key. Each document can have the same or a different structure.

- Wide-column stores: Wide-column NoSQL databases store data in tables with rows and columns similar to RDBMS, but names and formats of columns can vary from row to row across the table. Wide-column databases group columns of related data together. A query can retrieve related data in a single operation because only the columns associated with the query are retrieved. In an RDBMS, the data would be in different rows stored in different places on disk, requiring multiple disk operations for retrieval.
- Graph stores: A graph database uses graph structures to store, map, and query relationships. They provide index-free adjacency, so that adjacent elements are linked together without using an index.

3.What is a column family in HBase?

Column families are the base storage mechanism in HBase. A HBase table is comprised of one or more column families, each of which is stored in a separate set of regionfiles sharing a common key.

To express it in terms of an RDBMS, a column family is roughly analogous to a RDBMS table with the rowkey as a clustered primary key index. A HBase table would then be a view which does a full outer join on a set of RDBMS tables which all share the same primary key (thus having a 1:1 relationship). In this analogy, HBase region files map to pages in an RDBMS.

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

HBase currently does not do well with anything above two or three column families so keep the number of column families in your schema low.

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

Column qualifiers are dynamic and can be defined at write time. They are stored as byte so you can even put data in them.

6.How does data get managed in HBase?

HBase allows for many attributes to be grouped together into what are known as column families, such that the elements of a column family are all stored together. This is different from a row-oriented relational database, where all the columns of a given row are stored together. With HBase you must predefine the table schema and specify the column families. However, it's very flexible in that new columns can be added to families at any time, making the schema flexible and therefore able to adapt to changing application requirements. Data in Hbase is organized into tables. Any characters that are legal in file paths are used to name tables. Tables are further organized into rows that store data. Each row is identified

by a unique row key which does not belong to any data type but is stored as a bytearray. Column families are further used to group data in rows. Column families define the physical structure of data so they are defined upfront and their modification is difficult. Each row in a table has same column families. Data in a column family is addressed using a column qualifier. It is not necessary to specify column qualifiers in advance and there is no consistency requirement between rows. No data types are specified for column qualifiers, as such they are just stored as bytearrays. A unique combination of row key, column family and column qualifier forms a cell. Data contained in a cell is referred to as cell value. There is no concept of data type when referring to cell values and they are stored as bytearrays. Versioning happens to cell values using a timestamp of when the cell was written.

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

When the client issues a Put request, the first step is to write the data to the write-ahead log, the WAL:

- Edits are appended to the end of the WAL file that is stored on disk.
- The WAL is used to recover not-yet-persisted data in case a server crashes.

Once the data is written to the WAL, it is placed in the MemStore. Then, the put request acknowledgement returns to the client.

The MemStore stores updates in memory as sorted KeyValues, the same as it would be stored in an HFile. There is one MemStore per column family. The updates are sorted per column family.

When the MemStore accumulates enough data, the entire sorted set is written to a new HFile in HDFS. HBase uses multiple HFiles per column family, which contain the actual cells, or KeyValue instances. These files are created over time as KeyValue edits sorted in the MemStores are flushed as files to disk.

Note that this is one reason why there is a limit to the number of column families in HBase. There is one MemStore per CF; when one is full, they all flush. It also saves the last written sequence number so the system knows what was persisted so far.

The highest sequence number is stored as a meta field in each HFile, to reflect where persisting has ended and where to continue. On region startup, the sequence number is read, and the highest is used as the sequence number for new edits.

Data is stored in an HFile which contains sorted key/values. When the MemStore accumulates enough data, the entire sorted KeyValue set is written to a new HFile in HDFS. This is a sequential write. It is very fast, as it avoids moving the disk drive head.

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.