

CptS415 Big Data : Assignment 4

Md Muhtasim Billah

12/6/2020

Question 1.

1(a)

Answer:

NoSQL Database:

The NoSQL database refers to a non-relational database where the data format is different from the conventional tabulated, relations. In such database, data can be stored in a nested manner within a single data structure. Unstructured as well as semi-structured data can be stored in a NoSQL database. NoSQL database uses its very own distinctive data model for data query and for analyzing complex relationships. Its scalability, flexibility, and reliability features can sometimes offer better performance compared to a SQL database.

1(b)

Answer:

Types of NoSQL systems:

There are the 4 major types of NoSQL systems. These are briefly described below with examples.

i. Document Database:

There are NoSQL database systems that only store the data as documents. The most common types of document formats are JSON, XML and BSON. These file formats contain the information in certain **fields** and **values**. The values can be of different data types such as **numerical**, **string** and **Boolean** etc. In a document database, the documents are allowed to be stored and then retrieved as data objects to be used within an application. Sometimes, there is a built in scale-out type architecture in the database which can facilitate scalability for both data traffic and data volumes. This type of database is commonly used in ecommerce and trading platforms. One of the most popular document database systems these days is MongoDB.

ii. Key-value Store Database:

Key-value store database is the most simplistic type of NoSQL database where each data point is stored in the form of key-value pairs. The **keys** are the attributes of the database and the **values** corresponds to the **keys**. The basic data storage format has two columns- one for key and one for value and the columns stand in parallel. Most common types of applications for such database is the online shopping cart, personal user profile etc. One of the most popular key-value store database these days is the DynanoDB.

iii. Column-oriented Database:

For column-oriented database, the data are stored in a set of organized columns in the database. Here, all rows are required to have the same number of columns and thus offer significant advantages and flexibility over a SQL database. Moreover, for such database, the columns are often very similar to each other which facilitates efficient data compression and results in faster data query. Such databases are commonly used for internet of things (IOT), few other types of analytics etc. The most popular database of this category is **Cassandra**.

iv. Graph Database:

Graph database is the best option when the relationship between data elements is the primary focus. In a graph database, the data elements are stored as **nodes**. The connection between the nodes are established using the links which are called the **edges**. The **nodes** are stored directly in the database and then the **edges** are used to express the relationship between these **nodes**. The most common uses of such database systems are in the social networking websites, creation of knowledge graphs etc. One of the commonly used graph database is **Neo4j**.

1(c)

Answer:

As explained in 1(b), graph database systems are most suitable for social networking website and one classic example to explain its benefits can be **Facebook**. **Facebook** is world's most popular and fastest growing social network till date with 1.69 billion users as of 2020. All the users are connected to their friends on **Facebook** and they interact among each other. Each user can also interact with one or multiple pages, groups and advertisements etc. Each **Facebook** user can be considered as a node and their interactions with other users, pages, groups etc. can be considered as edges. These edges are the links that contain all the information about how each user is interacting with other nodes in the system. Thus, for **Facebook**, the store all these information, graph database can be the best option.

On the other side, for ecommerce and trading platforms, the informations about the clients, the stocks and the purchase etc. are mostly contained in documents and for such case, a graph database system will not be suitable for dealing with the data. Rather, a document database can serve better in this case.

Question 2.

2(a)

Answer:

CAP Theory:

CAP theory states that it is impossible for a distributed data store to simultaneously provide more than two out of following three guarantees

- Consistency
- Availability
- Partition tolerance

To put the theory into perspective, a cluster can be considered that contains two servers **S1** and **S2**. The CAP theory implies that if there is network partition, one has to choose between consistency and availability. When network partitions occur, a single cluster usually prioritizes data consistency over availability. On the other hand, a pair of unsynchronously replicated clusters prioritizes service availability over data consistency when network partitions occur.

2(b)

Answer:

Using the provided example and necessary scenarios, it can be shown when ACID (Atomicity, Consistency, Isolation and Durability) can be violated.

Atomicity: From a user's perspective, a transaction is either completed entirely (i.e., all relevant relations are updated) or not at all, there cannot be anything in between. If an error or interruption occurs, all changes made up to that point are backed out.

Consistency: All integrity conditions in the database are maintained with each transaction, taking the database from one consistent state into another consistent state.

Isolation: Each transaction is isolated from other transactions, and thus, each transaction only accesses data that are part of a consistent database state.

Durability: If a transaction has been reported back to a user as complete, the resulting changes to the database survive subsequent hardware or software failures.

Question 3.

3(a)

Answer:

Column Stores Features:

A column store database is one which stores data using a column-oriented model. The key concept behind the column store is the **keyspace**. The **keyspace** is similar to the schema in the relational model. A column store database has the following features.

- i. Efficient compression: Column stores are highly efficient at data compression and/or partitioning.
- ii. Aggregation queries: Due to their inherent data structure, column store database performs remarkably well with aggregation queries such as **SUM**, **COUNT** and **AVG** etc.
- iii. Superior scalability: Column store database facilitates scalability. They are well suited for massive parallel processing that involves having data spread across a large cluster of machines, sometimes thousands of machine.
- iv. Fast data loading and querying: Column store database can store data extremely fast. A table containing a billion rows can be loaded in a few seconds and querying and analyzing can be started almost immediately.

3(b)

Answer:

The three techniques to optimize column-oriented databases are explained below.

i. Compression:

Data compression makes low information entropy lead to high compression ratio. Each column is stored contiguously at a separate location on a disk and thus, the disk space is saved and less I/O from disk to memory is required. Performance of compression can be further improved if we can perform operation directly on compressed data. Among the available schemes, the **lightweight compression** perform better.

ii. Late Materialization:

For this technique, an optimal set of MVs is created for the given query of workload. The basic idea is to provide the required data to avoid overheads and to get better performance. This is expected to give better performance than the other two techniques in some situations.

iii. Block Iteration:

In block iteration, operator operate on blocks of tuples at once. Here, the iteration runs over blocks instead of tuples. Block iteration is similar to batch processing. Here, the blocks of values from the same columns are sent to an operator in a single function call. If the column width is fixed, then it can be worked on as an array. This can minimize the per-tuple overhead and exploit the potential for parallelism.

Question 4.

4(a)

Answer:

NewSQL System:

NewSQL can be thought of as an alias to the relational database management system (RDBMS). It is derived from the concepts of both the SQL and the NoSQL languages. In case of RDBMS, NewSQL helps to improve the performance in the online transaction processing. In terms of reliability and efficiency, NewSQL offers better performance than the SQL and NoSQL. Another promising advantage of NewSQL is that it is more consistent and also compatible with SQL. The complexity of NewSQL is also less compared to the other systems.

Design Principles of NewSQL:

NewSQL is built on the following design principles:

- NewSQL is derived from the traditional SQL system.
- Inherits the ACID (Atomicity, Consistency, Isolation and Durability) properties.
- It uses the principles of the DBMS (Distributed Databases Management System).
- Utilizes the principles of STM (Scalable Transaction Management).
- NewSQL adopts the principle of data streaming.

4(b)

Answer:

For in-memory DBMS, the set-associative cache for block placement and LRU block replacement policy are described below.

Set associative cache:

Set associative cache is a method for placement of data as blocks for in-memory DBMS. When a block is placed in a restricted set of cache, it's called the set associative cache. Here, the set is a collection of multiple blocks. A specific block is mapped in the set and then that block is placed in all the places where the set is grouped. This is commonly known as the interface between the fully associative cache and the direct associative cache. Where the direct cache can fail in some situations, set associative cache has proven to show better results. Another advantage of set associative cache is that multiple words of memory can be stored under the same address.

LRU block replacement policy:

Least recent used or LRU is a block replacement policy for in-memory DBMS which is based on the locality of the interface. In this scheme, usually the least recent memory item is referenced or updated. In order to indicate how the blocks are used in the LRU policy, the Cache Memory Block (CMB) is used. By the largest block number, the LRU block is replaced in the process and the policy reduces the chance of throwing away information. The block that has not been used for a long period of time, this policy replaces that block.