Module 4: Column-Family Database

Text chapters 9, 10, & 11

University of Illinois
at Springfield

What is Column Family Database

- Column family databases originated with Google's BigTable.
- The following are important features of BigTable:
  - Developers have dynamic control over columns.
  - Data values are indexed by row identifier, column name, and a time stamp.
  - Data modelers and developers have control over location of data.
  - Reads and writes of a row are atomic.
  - Rows are maintained in a sorted order.

What is Column Family Database

- Column families are organized into groups of data items that are frequently used together.
- Column families for a single row may or may not be near each other when stored on disk, but columns within a column family are kept together.
- The use of column families and dynamic columns enables database modelers to define broad, course-grained structures (that is, column families) without anticipating all possible fine-grained variations in attributes.

What is Column Family Database

- A column name uniquely identifies a column in a table.
  - The time stamp orders versions of the column value.
- Column families store columns together in persistent storage, making it more likely that reading a single data block can satisfy a query.
- In column family databases, all read and write operations are atomic regardless of the number of columns read or written.
  - This means that as you read a set of columns, you will be able to read all the columns needed or none of them.
  - No partial results are allowed with atomic operations.

What is Column Family Database

- Column families are analogous to keyspaces in key-value databases.
  - Developers are free to add keys and values in key-value databases just as they are free to add columns and values to column families.
- Column family databases, like document databases, do not require all columns in all rows.
  - Some rows in a column family database may have values for all columns, whereas others will have values for only some columns in some column families.
- In both column family and document databases, columns or fields can be added as needed by developers.

What is Column Family Database

- Column family databases have some features that are similar to features in relational databases and others that are superficially similar but different in implementation.
- Both column family databases and relational databases use unique identifiers for rows of data.
  - These are known as row keys in column family databases and as primary keys in relational databases.
- Both row keys and primary keys are indexed for rapid retrieval.

What is Column Family Database

- Both types of databases can be thought of as storing tabular data, at least at some level of abstraction.
  - The actual storage model varies, even between relational databases.
- Column family databases use the concept of maps (also known as dictionaries or associative arrays).
  - A column key maps from a column name to a column value.
  - A column family is a map/dictionary/associative array that points to a map/dictionary/associative array.
- Column family databases do not support the concept of a typed column.
  - Column values can be seen as a series of bytes that are interpreted by an application, not the database.

What is Column Family Database

- Although you can expect to find atomic reads and writes with respect to a single row, column family databases such as Cassandra do not support multi-row transactions.
- If you need to have two or more operations performed as a transaction, it is best to find a way to implement that operation using a single row of data.
  - This might require some changes to your data model and is one of the considerations you should take into account when designing and implementing column families.

What is Column Family Database

- There should be minimal need for joins and subqueries in a column family database.
- Column families promote denormalization and that eliminates, or at least reduces, the need for joins.

Column Family Database Architecture

- Two common types of architectures are used with distributed databases:
  - multiple node type
  - peer-to-peer type.
- Multiple node type architectures have at least two types of nodes, although there may be more.
  - HBase is built on Hadoop and uses various Hadoop nodes, including name nodes, data nodes, and a centralized server for maintaining configuration data about the cluster.
- Peer-to-peer type architectures have only one type of node.
  - Any node can assume responsibility for any service or task that must be run in the cluster.

Column Family Database Architecture

- Gossip protocols are used to share information about the state of data on different servers.
- With gossip protocols, each server updates another server about itself as well as all the servers it knows about.
- Those servers can then share what they know with a second set of other servers.
- The second set, which might receive information from a few different servers, can pass on all the status information it has been sent instead of just passing on its own information.

Column Family Database Architecture: Entropy

- Distributed database designers must address information entropy.
- Information entropy increases when data is inconsistent in the database.
- Many Distributed Architectures use an anti-entropy algorithm—that is, one that increases order—to correct inconsistencies between replicas.

Column Family Database Architecture: Entropy

- When a server initiates an anti-entropy session with another server, it sends a hash data structure, known as a Merkle or hash tree, derived from the data in a column family.
- The receiving server calculates a hash data structure from its copy of the column family.
  - If they do not match, the servers determine which of the two has the latest information and update the server with the outdated data.

Column Family Database Architecture: Handoff

- A hinted handoff entails storing information about the write operation on a proxy node and periodically checking the status of the unavailable node.
- When that node becomes available again, the node with the write information sends, or “hands off,” the write request to the recently recovered node.

When to use a Column Family Database

- Column family databases are appropriate choices for large-scale database deployments that require high levels of write performance, a large number of servers, or multi-data center availability.
- Column family databases are also appropriate when a large number of servers are required to meet expected workloads.

Column Family Database Vs RDMS

- Column family and relational databases share terminology, such as columns, rows, and tables.
  - These terms are equivalent at a logical level but they differ at a physical implementation level.
- Column families use maps and maps of maps to organize data. Relational databases store data in more row- or column-oriented formats.

Column Family Database Vs RDMS

- **Column family databases support some level of transactions, but they are not as robust as transactions in relational databases.**
  - **For example, a read or write operation on a row in a column family database is atomic. Relational databases support transactions across multiple tables and rows.**
- **Column family databases are designed for large, data-intensive applications.**
  - **They probably are not the best option for applications that can run on a single server.**

Column Family Database Terminology: KEYSPACE

- **Keyspace: The top-level data structure in a column family database.**
- **It is top level in the sense that all other data structures you would create as a database designer are contained within a keyspace.**
  - **A keyspace is analogous to a schema in a relational database.**
- **Row key: A column that uniquely identifies a row in a column family.**
  - **It serves some of the same purposes as a primary key in a relational database.**

Column Family Database Terminology: COLUMN

- **Column: The data structure for storing a single value in a database.**
- **Column families: Collections of related columns.**
  - **Columns that are frequently used together should be grouped into the same column family.**

Column Family Database Terminology

- **Cluster: A set of servers configured to function together.**
  - **Servers sometimes have differentiated functions, and sometimes they do not.**
- **Partition: A logical subset of a database.**
  - **Partitions are usually used to store a set of data based on some attribute of the data.**
- **Commit logs: Append-only files that always write data to the end of the file.**
  - **Used to save copies of data written to tables. Enables data recovery.**

Column Family Database Terminology

- **Bloom filter: A probabilistic filter that tests whether an element is a member of a set.**
  - **Might return a false positive but never returns a false negative.**
- **Consistency level: The consistency between copies of data on different replicas.**
  - **In the strictest sense, data is consistent only if all replicas have the same data.**
- **Replication: The process of determining where to place replicas and how to keep them up to date.**

Column Family Database Terminology

- **Anti-entropy: The process of detecting differences in replicas.**
  - **From a performance perspective, it is important to detect and resolve inconsistencies with a minimum amount of data exchange.**
- **Gossip protocol: A protocol for sharing information between nodes in a cluster that enables nodes to share information about themselves and other clusters for which they have up-to-date information.**
- **Hinted handoff: A protocol for collecting write information when a server is unavailable.**
  - **Another server in the clusters saves the write data and passes it to the target server when it becomes available again.**

Column Family Database Design

- Like other NoSQL databases, queries drive the design.
- Queries provide information needed to effectively design column family databases. The information include
  - entities,
  - attributes of entities,
  - query criteria,
  - derived values.

Column Family Database Design

- Entities represent things that can range from concrete things, such as customers and products, to abstractions, such as service-level agreements or a credit score history.
- Entities are modeled as rows in column family databases.
  - A single row should correspond to a single entity.
  - Rows are uniquely identified by row keys.
- Attributes of entities are modeled using columns.
- Queries include references to columns to specify criteria for selecting entities and to specify a set of attributes to return.

Column Family Database Design

The following are important characteristics of column family databases:

- **Column family databases are implemented as sparse, multidimensional maps.**
- **Columns can vary between rows.**
- **Columns can be added dynamically.**
- **Joins are not used; data is denormalized instead.**

Column Family Database Design

The following are several guidelines to keep in mind when designing tables:

- **Denormalize instead of join.**
- **Make use of valueless columns.**
- **Use both column names and column values to store data.**
- **Model an entity with a single row.**
- **Keep an appropriate number of column value versions.**
- **Avoid complex data structures in column values.**
- **Avoid hotspotting in row keys.**

Column Family Database Design

- **Distinguishing two kinds of indexes is helpful:**
  - **primary**
  - **secondary**
- **Primary indexes are indexes on the row keys of a table.**
  - **They are automatically maintained by the column family database system.**
- **Secondary indexes are indexes created on one or more column values.**
  - **Secondary indexes can be created and managed by the database system or by your application.**
  - **Not all column family databases provide automatically managed secondary indexes, but you can create and manage tables as secondary indexes in all column family database systems.**

Column Family Database Design

- As a general rule, if you need secondary indexes on column values and the column family database system provides automatically managed secondary indexes, then you should use them.
- The primary advantage of using automatically managed secondary indexes is that they require less code to maintain than the alternative.
- The automatic use of secondary indexes has another major advantage because you do not have to change your code to use the indexes.

Column Family Database Design

- There are times when automatically managed indexes should not be used. Avoid, or at least carefully test, the use of indexes in the following cases:
  - There is a small number of distinct values in a column
  - There are many unique values in a column
  - The column values are sparse

Column Family Database Design

- If your column family database system does not support automatically managed secondary indexes or the column you would like to index has many distinct values, you might benefit from creating and managing your own indexes.
- When using tables as indexes, you will be responsible for maintaining the indexes.

Column Family Database Design

- **When maintaining the indexes, there are two broad options with regard to the timing of updates.**
  - **update the index whenever there is a change to the base tables; for example, when a customer makes a purchase.**
    - Updating index tables at the same time you update the base tables keeps the indexes up to date at all times.
  - **Or you could run a batch job at regular intervals to update the index tables.**
    - Updating index tables with batch jobs has the advantage of not adding additional work to write operations.

Column Family Database and Big Data

Gartner defines Big Data as high velocity, high volume, and/or high variety.

- ***Velocity*** **refers to the speed at which data is generated or changed.**
- ***Volume*** **refers to the size of the data.**
- ***Variety*** **refers to the range of data types and the scope of information included in the data.**

Column Family Database and Big Data

Moving large amounts of data is challenging for several reasons, including

- **Insufficient network throughput for the volume of data**
- **The time required to copy large volumes of data**
- **The potential for corrupting data during transmission**
- **Storing large amounts of data at the source and target**

Column Family Database and Big Data

There are many ways to analyze data, look for patterns, and otherwise extract useful information.

Two broad disciplines are useful here:

- **Statistics**
- **Machine Learning.**

Column Family Database and Big Data

- ***Statistics*** **is the branch of mathematics that studies how to describe large data sets, also known as populations, and how to make inference from data.**
- **Descriptive statistics are particularly useful for understanding the characteristics of your data.**
- **Predictive, or inferential, statistics is the study of methods for making predictions based on data.**

Column Family Database and Big Data

- ***Machine learning*** **incorporates methods from several disciplines, including computer science, artificial intelligence, statistics, linear algebra, and others.**
  - **Many services might be taken for granted, such as getting personal recommendations based on past purchases, analyzing the sentiment in social media posts, fraud detection, and machine translation, but all depend on machine-learning techniques.**

Column Family Database and Big Data

One area of machine learning, called *Unsupervised Learning*, is useful for exploring large data sets.

- **A common unsupervised learning method is clustering.**
- **Clustering algorithms are helpful when you want to find non-apparent structures or common patterns in data.**

Column Family Database and Big Data

- ***Supervised Learning*** **techniques provide the means to learn from examples.**
  - **A credit card company, for example, has large volumes of data on legitimate credit card transactions as well as data on fraudulent transactions. There are many ways to use this data to create classifiers, which are programs that can analyze transactions and classify them as either legitimate or fraudulent.**
- **Commonly used tools for big data analysis include Map Reduce, Spark, R, and Mahout.**
- **Tools for managing big data include Ganglia, Hannibal for HBase, and OpsCenter for Cassandra.**