A Study on Cassandra: Data Model

Sayali Amrutkar
MCA Student
School of Computer Science
Faculty of Science
MIT World Peace University
Pune, India

Sayali Patil
MCA Student
School of Computer Science
Faculty of Science
MIT World Peace University
Pune, India

Abhilasha Kumari
MCA Student
School of Computer Science
Faculty of Science
MIT World Peace University
Pune, India

Jigneshkumar Mahadik
MCA Student
School of Computer Science
Faculty of Science
MIT World Peace University
Pune, India

Dr. C.H.Patil
Head of School
School of Computer Science
Faculty of Science
MIT World Peace University
Pune, India

Abstract- **Cassandra is used for handling huge volume of structured data which are expanded in different servers. It has feature like no single point of failure. Cassandra handles constant phase of these failures which maintains reliability and scalability of the systems. Cassandra is a data model which is very easy to understand that helps to keep dynamic control on data layout. It does not support relational data model. Cassandra was designed to handle write throughput without sacrificing read efficiency.**

Keywords- cassandra, scalability, reliability.

I. INTRODUCTION

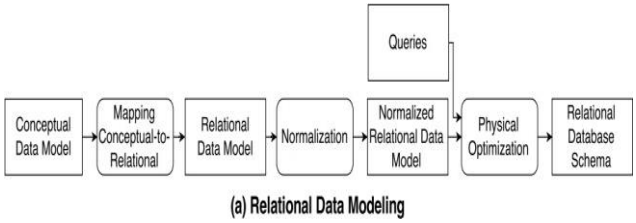
Day by day, the amount of data has been increased at a great rate. Multiple sources produce huge amount of data and its representation becomes a difficult task. So there is a need to handle such huge data. NoSQL is used to deal with this problem which allows managing and storing large datasets designed to scale horizontally. NoSQL provides reliability in favour of scope and segregate resistance which gives a appreciable enhancement in performance and scalability as compared to ACID property(Atomicity, Consistency, Isolation, Durability).These database are known as NoSQL because it does not follow the Relational Database Management System (RDBMS). NoSQL can work with denormalized data. Cassandra is big data ready and highly scalable NoSQL database.

The big data applications adopt Cassandra widely; it is fault tolerant and scalable peer-to-peer architecture. This data model is flexible and versatile which is derived from Big Table. Cassandra uses CQL (Cassandra Query Language) which is very easy to understand, it scales to thousand of transactions per milliseconds and manage failure with ease which is adopted in large volume of data applications

II. TRADITIONAL DATA MODELING

The relational databases use traditional data modelling methodology, which defines well-established steps. Fig.(a)[1] depicts the database schema design workflow which is typically followed by a database designer. Fig.(a)[1] defines relational data modeling, it depicts normalize relations and apply physical optimization to create a systematic relational database schema. By avoiding data duplication and minimizing data redundancy, this process primarily focuses on ability to understand and arrange the data systematically into relations. Secondary role is played by queries in schema design.

The Structured Query Language (SQL) mainly supports nested queries, relational joins. At initial design stage the query analysis is excluded oftenly due to articulation of SQL(Structured Query Language). SQL supports various features in order to optimize the most constantly executed queries.



(a) Relational Data Modeling

III. CASSANDRA DATA MODELING

Cassandra is a non-structured database schema which describes a top-level namespace as keyspace which have a no. of CQL tables to store data objects .

The traditional data modeling methodology and Cassandra data modeling technique differ from each other. It is totally dependent on conceptual data model. Some features like data aggregation and join are not supported by CQL. It supports denormalization and able to answer all queries by a single table. As described in fig. (b)[1] conceptual data model and application window works together which align conceptual data into logical data model after that by using physical optimization process it converts into physical data model.

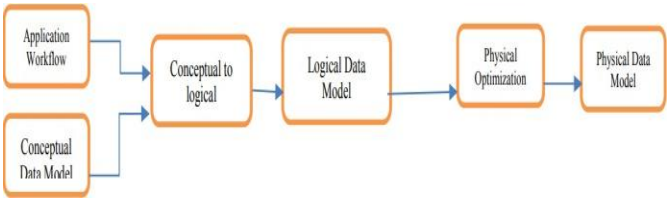


Fig. (b) Cassandra Data Modeling

Column: It is the atomic unit of detailed data. Syntax: key: value.

Super Column: By using Super columns we can form complex data types easily. Super columns are grouped together with a common name.

Row: It is the exclusively distinguishable data which is formed by columns and super columns. Every unique keys in row are important for distributed hash table implementation.

Column Family: It groups together super column and column of huge structured data which is a part of abstraction containing keyed rows. It doesn't described schemas of column names and its types. Application layer holds all logic regarding data interpretation. It acts as contrast to the relational data model. Every column names and its values are represented by 64-bit long integer types or UTF-8 strings and are saved as unlimited size of bytes.

Node: It is the primary part of Cassandra which stores data in it**.**

Data Center: A group of nodes is called data center**.**

Cluster: Many data centers together form cluster.

Commit Log: It records every write operation.

Mem-table: After writing data in Commit log it is written in Mem-table.

Keyspace: The top level namespace in Cassandra is Keyspace. Keyspace have exactly one subordinate called as column families.

EXAMPLES:

CRUD Operations:

Creating Table:

```
Create table stud (Prn int PRIMARY KEY,
                   name text,
                   email  text,
                   trimester int  );
```

Inserting data into table:

```
Insert into SY_MCA.stud (Prn,name,email,trimester)
values(12098765,'Abhinay','abhinay4044@gmail.com',6);
```

Selecting data from table:

```
Select * from SY_MCA.stud;
```

Updating data into table:

```
Update SY_MCA.stud Set name='Abhishek'
where Prn=12098765;
```

Deleting the data:

```
Delete from SY_MCA.stud where Prn=12098765;
```

IV. ARCHITECTURE

1. Cluster

There are several machines that operate together and Cassandra database is distributed over them. Cluster is known as the outermost container. For failure handling there is feature in Cassandra in which every node holds a replica, which handles its deficiency and takes charge if any deficiency occurs. Cassandra use SimpleStrategy which is like a ring format to arrange nodes in a cluster.

Every node takes read and write request inconsiderating the actual location of data stored in cluster. Many different layers of storage units form the Cassandra Cluster. The Keyspace serves as the outer layer for data in Cassandra.

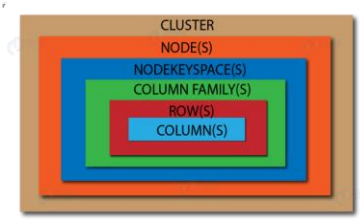
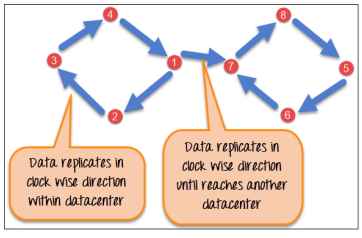


Fig. (c) Cluster

2. Replication

Cassandra replicates data according to the chosen replication strategy. By using replication strategy it resolves the arrangement of copied data. Cassandra uses two main replication methods NetworkTopologyStrategy and the SimpleStrategy. Replication strategy resolves how the adjacent replicas are arranged. Cassandra use SimpleStrategy which is like a ring format to arrange subsequent replicas on the adjacent nodes in a cluster.

When Cassandra is deployed across various data centers then the NetworkTopologyStrategy works in satisfactory manner. The NetworkTopologyStrategy assure that replicas are stored in different racks but in same data center. Cassandra uses various sources to find long-range network topology. Replication technique or replica arrangement strategy is used to travel along inter node requests in its bound.



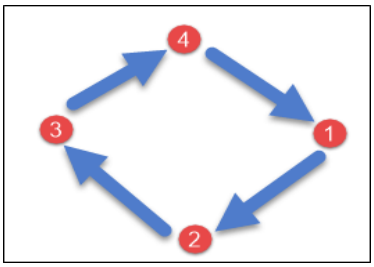


Fig.(d)Network Topology Strategy

CONCLUSION

When there is large volume and various types of data and we need to improve their performance and scalability then we have to make significant changes or improvements to the ways in which data is represented and examined so data can be extracted efficiently with details. Cassandra supports many features such as no single point of failure, high scalability and flexible schema. And It helps in implementation of storing huge amount of data. No matter what the size of data stored, Cassandra provides feature of fast retrieval of data in proper and efficient manner. In this thesis we explained why Cassandra is superior over Relational Data Modeling. Cassandra data modeling defines all its functionality such as, physical data model, physical optimization, Forming concepts of data model and its mapping from conceptual to logical.

REFERENCES

[1] Artem Chebotko, Andrey Kashlev, Shiyong Lu, " A Big Data Modeling Methodology for Apache Cassandra", 2015 IEEE International Congress on Big Data, 2015.

[2] Andre Ribeiro, Afonso Silva, Alberto Rodrigues da Silva, " Data Modeling and Data Analytics: A Survey from a Big Data Perspective", Journal of Software Engineering and Applications, 2015, 8, 617-634. Cattell, R. (2011). Scalable SQL and NoSQL data stores. ACM SIGMOD Record, 39(4), 12-27.

[3] John Berryman, "The CQL3/Cassandra Mapping", OpenSource Connections.

[4] Ramakrishnan, R. and Gehrke, J. (2012) Database Management Systems. 3rd Edition, McGraw-Hill, Inc., New York.