Introduction to CassandraDB

Apache CassandraDB

- Apache Cassandra is a highly scalable, high-performance distributed database designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure.
- It is a type of NoSQL database.

NoSQL vs. RDBMS

Relational Database	NoSql Database
Supports powerful query language.	Supports very simple query language.
It has a fixed schema.	No fixed schema.
Follows ACID (Atomicity, Consistency, Isolation, and Durability).	It is only "eventually consistent".
Supports transactions.	Does not support transactions.

Popular NoSQL Databases

- Apache HBase:
  - HBase is an open source, non-relational, distributed database modeled after Google's BigTable and is written in Java. It is developed as a part of Apache Hadoop project and runs on top of HDFS, providing BigTable-like capabilities for Hadoop.
- MongoDB:
  - MongoDB is a cross-platform document-oriented database system that avoids using the traditional table-based relational database structure in favor of JSON-like documents with dynamic schemas making the integration of data in certain types of applications easier and faster.

Features of Cassandra

- Elastic scalability: Cassandra is highly scalable; it allows to add more hardware to accommodate more customers and more data as per requirement.
- Always on architecture: Cassandra has no single point of failure and it is continuously available for business-critical applications that cannot afford a failure.
- Fast linear-scale performance: Cassandra is linearly scalable, i.e., it increases your throughput as you increase the number of nodes in the cluster. Therefore it maintains a quick response time.
- Flexible data storage: Cassandra accommodates all possible data formats including: structured, semi-structured, and unstructured. It can dynamically accommodate changes to your data structures according to your need.

Features of Cassandra

- Flexible data storage: Cassandra accommodates all possible data formats including: structured, semi-structured, and unstructured. It can dynamically accommodate changes to your data structures according to your need.
- Easy data distribution: Cassandra provides the flexibility to distribute data where you need by replicating data across multiple datacenters.
- Transaction support: Cassandra supports properties like Atomicity, Consistency, Isolation, and Durability (ACID).
- Fast writes: Cassandra was designed to run on cheap commodity hardware. It performs blazingly fast writes and can store hundreds of terabytes of data, without sacrificing the read efficiency.

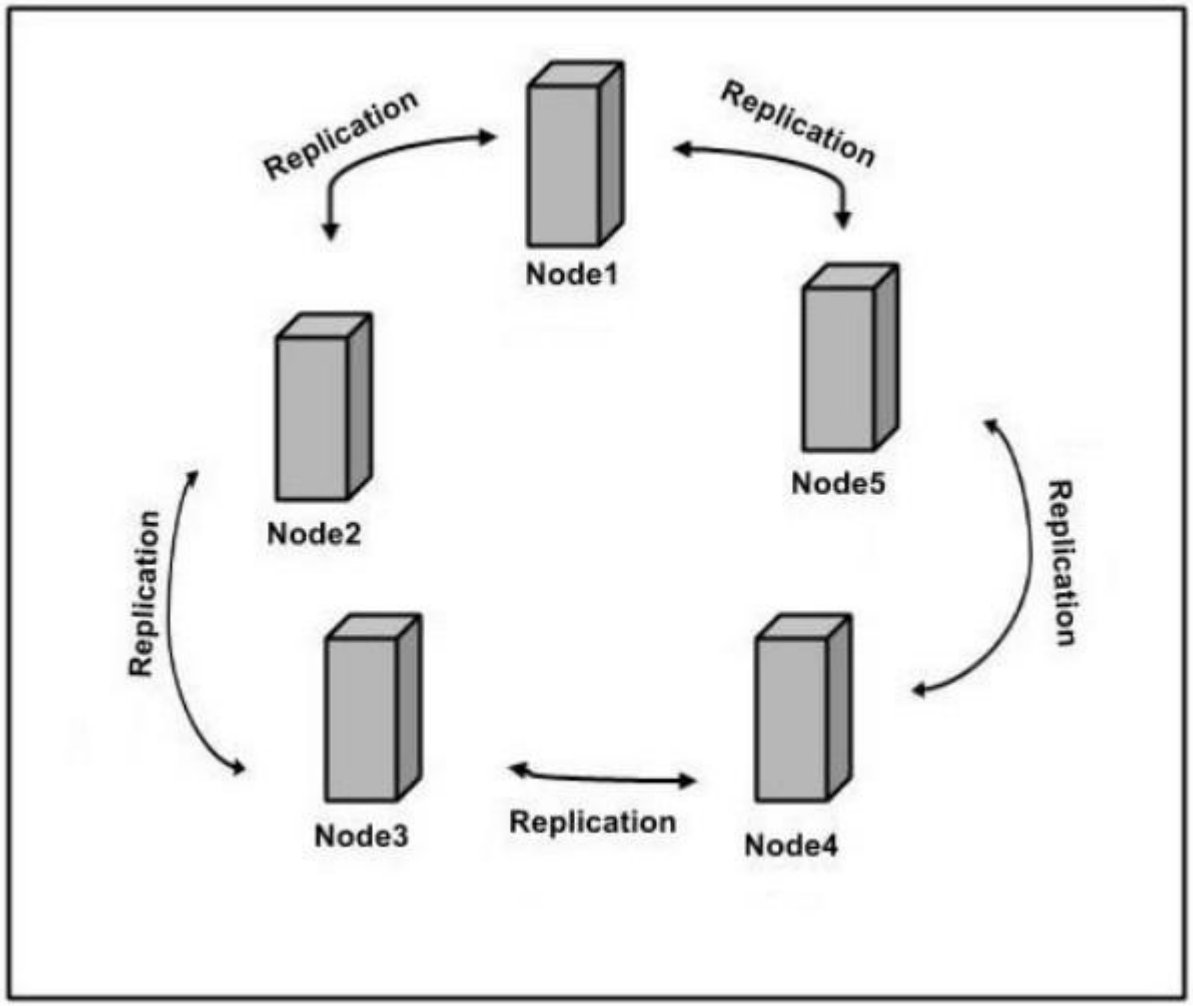
History of Cassandra

- Cassandra was developed at Facebook for inbox search.
- It was open-sourced by Facebook in July 2008.
- Cassandra was accepted into Apache Incubator in March 2009.
- It was made an Apache top-level project since February 2010.

Data replication in Cassandra

- In Cassandra, one or more of the nodes in a cluster act as replicas for a given piece of data.
- If it is detected that some of the nodes responded with an out-of-date value, Cassandra will return the most recent value to the client.
- After returning the most recent value, Cassandra performs a read repair in the background to update the stale values.

Data replication in Cassandra



Cassandra QL

- Users can access Cassandra through its nodes using Cassandra Query Language (CQL). CQL treats the database (Keyspace) as a container of tables.
- Programmers use cqlsh: a prompt to work with CQL or separate application language drivers.
- Clients approach any of the nodes for their read-write operations. That node (coordinator) plays a proxy between the client and the nodes holding the data.

Data Model

- The data model of Cassandra is significantly different from what we normally see in an RDBMS.
- Cassandra database is distributed over several machines that operate together.
- The outermost container is known as the Cluster. For failure handling, every node contains a replica, and in case of a failure, the replica takes charge.
- Cassandra arranges the nodes in a cluster, in a ring format, and assigns data to them.

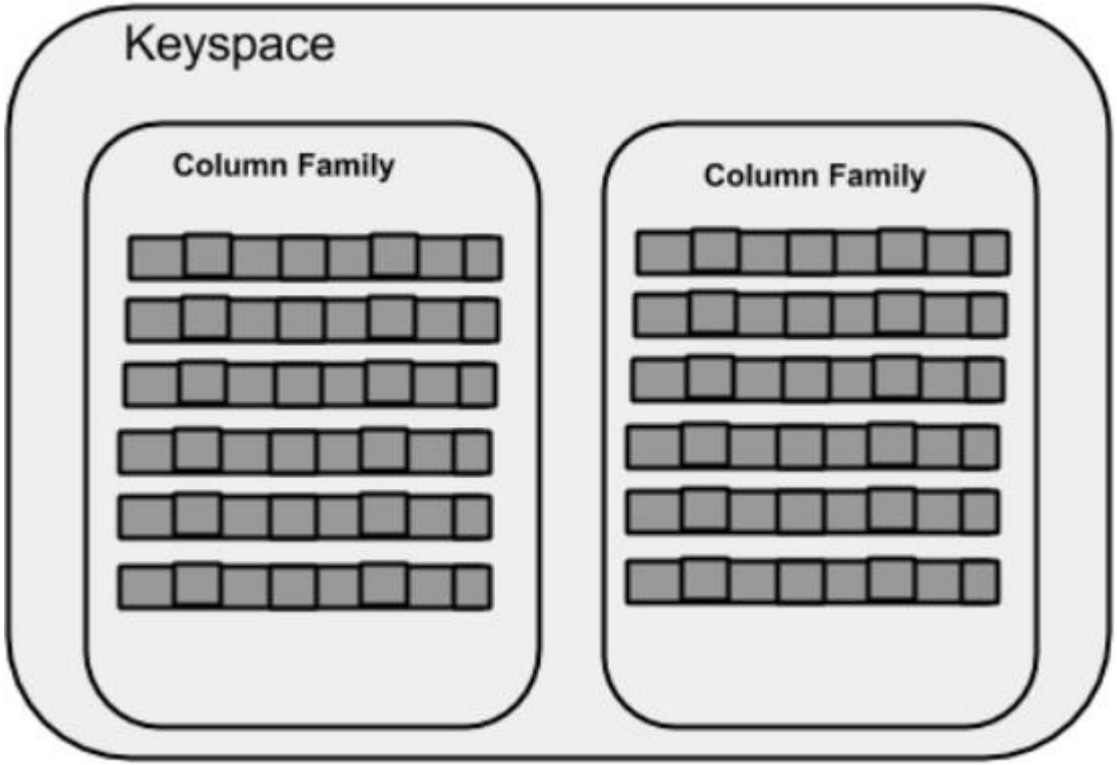
Data Model

- Keyspace is the outermost container for data in Cassandra. The basic attributes of a Keyspace in Cassandra are:
- Replication factor:
  - It is the number of machines in the cluster that will receive copies of the same data.
- Replica placement strategy:
  - It is nothing but the strategy to place replicas in the ring. We have strategies such as simple strategy (rack-aware strategy), old network topology strategy (rack-aware strategy), and network topology strategy (datacenter-shared strategy).
- Column families:
  - Keyspace is a container for a list of one or more column families. A column family, in turn, is a container of a collection of rows. Each row contains ordered columns. Column families represent the structure of your data. Each keyspace has at least one and often many column families.

Creating keyspace

- ```
  CREATE KEYSPACE Keyspace name WITH
  replication = {'class':
  'SimpleStrategy',
  'replication_factor' : 3};
  ```
  ```

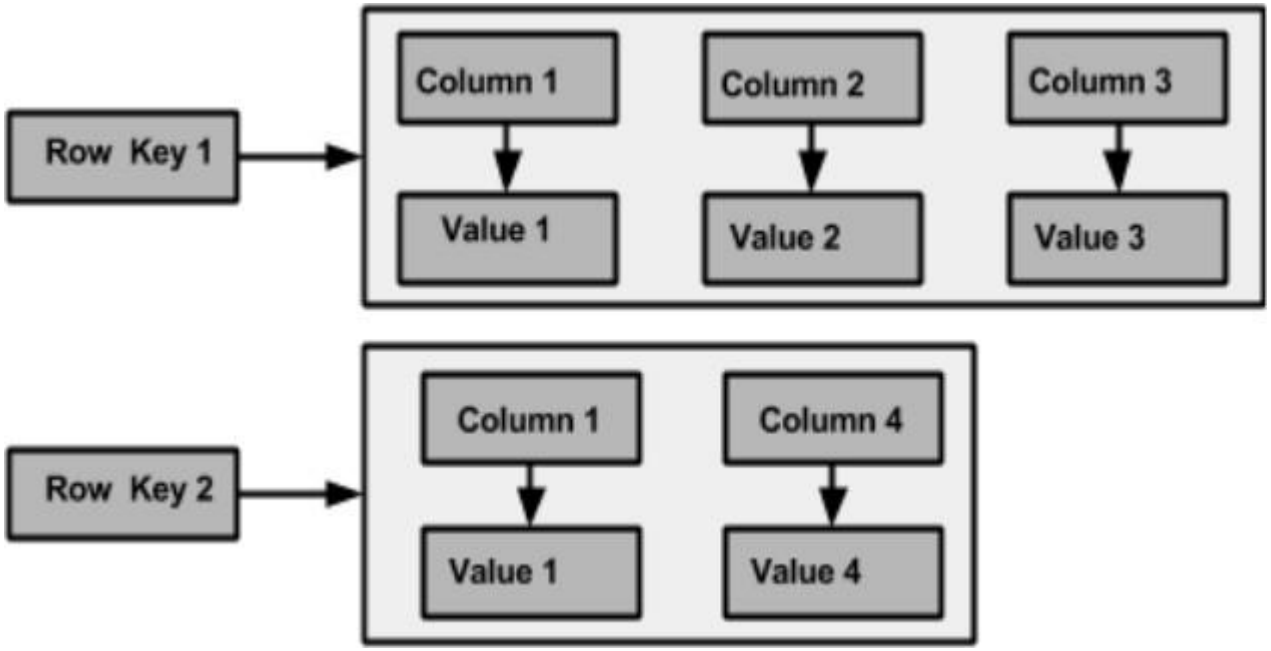
Keyspace



Column family

- A column family is a container for an ordered collection of rows. Each row, in turn, is an ordered collection of columns.
- A Cassandra column family has the following attributes:
  - keys_cached It represents the number of locations to keep cached per SSTable.
  - rows_cached It represents the number of rows whose entire contents will be cached in memory.
  - preload_row_cache: It specifies whether you want to pre-populate the row cache.

Column family



RDBMS vs. Cassandra

RDBMS	Cassandra
RDBMS deals with structured data.	Cassandra deals with unstructured data.
It has a fixed schema.	Cassandra has a flexible schema.
In RDBMS, a table is an array of arrays. (ROW x COLUMN)	In Cassandra, a table is a list of "nested key-value pairs". (ROW x COLUMN key x COLUMN value)
Database is the outermost container that contains data corresponding to an application.	Keyspace is the outermost container that contains data corresponding to an application.
Tables are the entities of a database.	Tables or column families are the entity of a keyspace.
Row is an individual record in RDBMS.	Row is a unit of replication in Cassandra.
Column represents the attributes of a relation.	Column is a unit of storage in Cassandra.
RDBMS supports the concepts of foreign keys, joins.	Relationships are represented using collections.

- **Web Resources** http://mitu.co.in http://tusharkute.com
- Tushar B. Kute, http://tusharkute.com
- **Blogs** http://digitallocha.blogspot.in
  http://kyamputar.blogspot.in