# Cassandra

Alpesh Vasant, G Academy

Introduction To Cassandra

- It's a Wide Column NoSQL database
- Open Source- So no paid support
- Distributed – Data stored in clusters, data centers and on nodes
- Handles large amount of data may be spreaded physically across world
- Hence provides services with very high available period

Introduction To Cassandra

- **Apple** uses 100,000 Cassandra nodes
- **Netflix** uses Cassandra as their back-end database for their streaming services
- **SoundCloud** uses Cassandra to store the dashboard of their users
- **Uber** uses Cassandra to store around 10000 features in their daily updated company-wide Features
- **Cisco**'s **Webex** uses Cassandra to store user feed and activity in near real time
- **Discord** switched to Cassandra to store billions of messages from MongoDB in November 2005

Architecture Of Cassandra
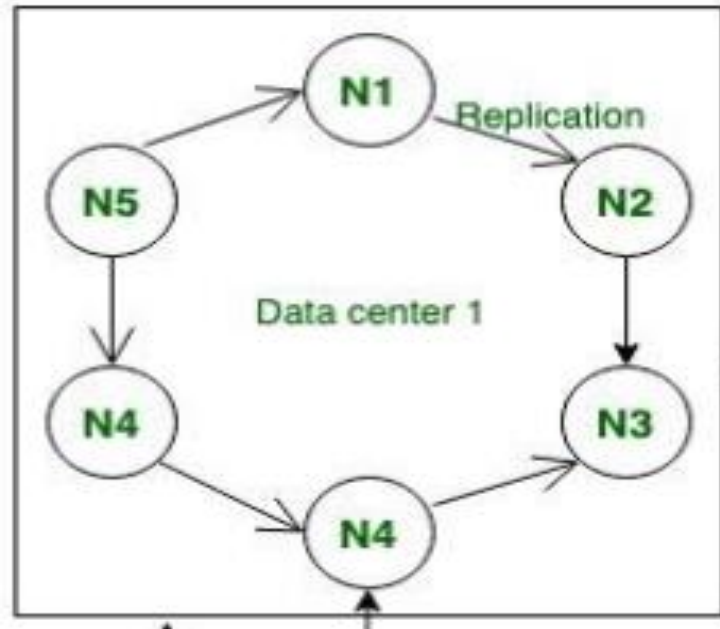
- Token ring like architecture with no master no slave

- Nodes are logically distributed like ring. Hash values of the keys are used to distribute the data among nodes in the cluster.

- Automatic data distribution across all nodes

Architecture Of Cassandra
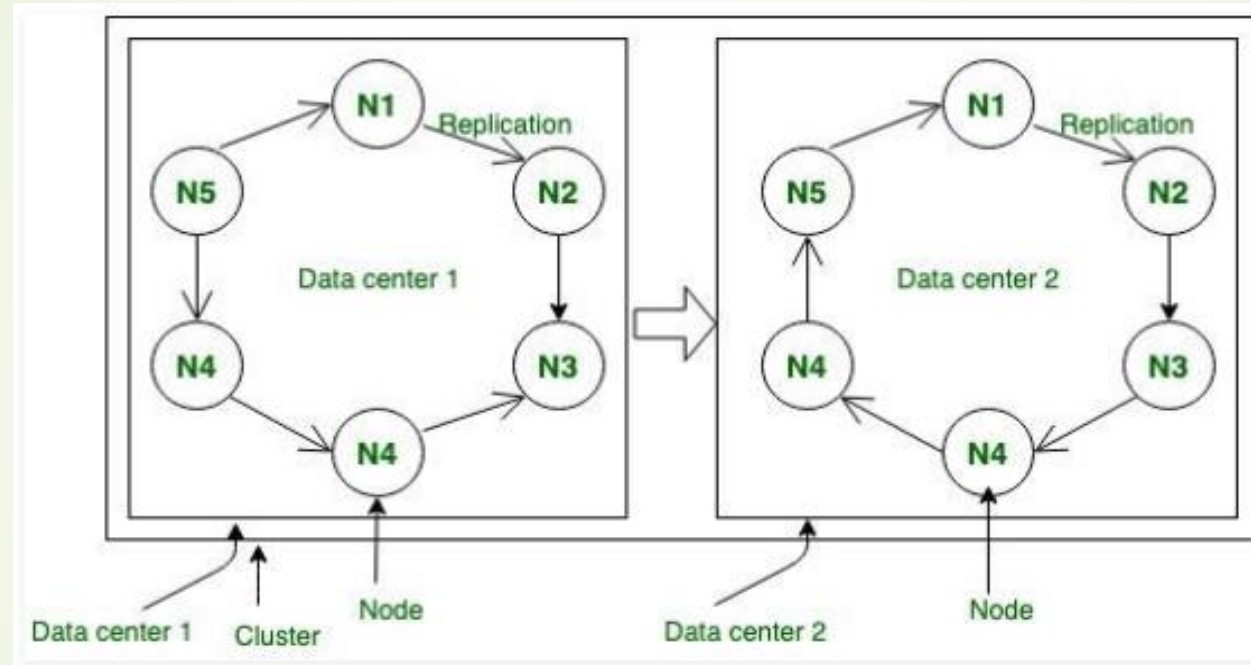
- Node: Base component. Data is stored on Node
- Data Centre: Collection of Nodes is called Data Centers
- Cluster: Collections of many data Centers
- Architecture supports multiple data centers
- Data can be replicated across data centers

Architecture Of Cassandra
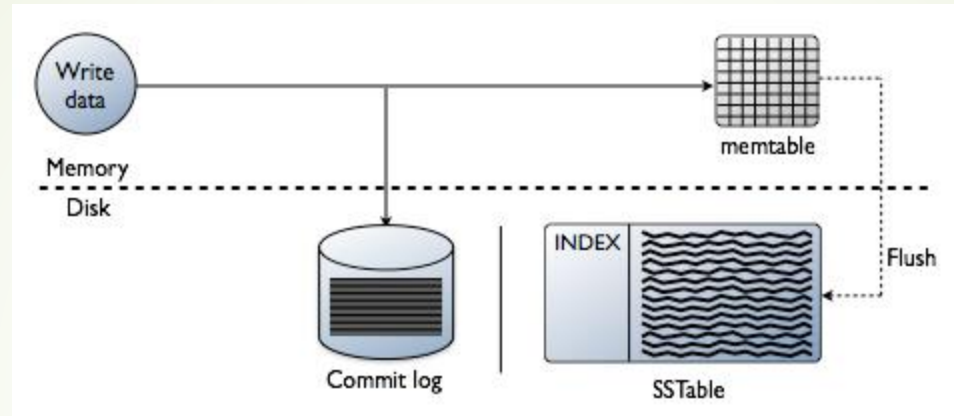
- Commit Log: Every WRITE operation is written here. Used for crash recovery

- Mem Table: Post data written in commit log, data is also written in Mem table but temporally

- SS Table: Once the Mem Table reaches to certain preset threshold, data is shifted to SS Table (disk file)

Comparison between Cassandra and MongoDB

MongoDB	Cassandra
Document based NoSQL database	**Wide Column** NoSQL database
Data is stored in documents in the form of **Key-Value pair**	Stores data in the form of traditional **rows and columns**
Explicit data type is not required to declare while storing yet it stores large type of data including: strings, numbers, geo data, dates, arrays, decimal, nested objects, and binary data	**Each column has a specific data type** that it can store that has to be specified at the time of table creation
Compare to Cassandra **availability is bit compromised** due to master slave model	Cassandra provides very **high availability** due to token ring type of architecture
JSON is the language for query	**CQL** is language for query
Has **built in aggregation framework** to run ETL pipeline	**No built in aggregation framework**. If needed, it used **external tools** like **Apache Hadoop** or **Apache Spark**

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 is continuously available for business-critical applications that cannot afford a failure.

- **Fast linear-scale performance** − As you increase the number of nodes in the cluster, you handle more simultaneous requests by customers. 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

- **Easy data distribution** – Cassandra provides the flexibility to distribute data where you need by replicating data across multiple data centers.

- **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.