Cassandra is a peer-to-peer distributed database that runs on a cluster of homogeneous nodes. Cassandra has been architected from the ground up to handle large volumes of data while providing high availability. Cassandra provides high write and read throughput.  A Cassandra cluster has no special nodes i.e. the cluster has no masters, no slaves or elected leaders. This enables Cassandra to be highly available while having no single point of failure.

In Cassandra does not support joins, group by, OR clause, aggregations

Data denormalization and data duplication are defacto of Cassandra

**Cassandra Architecture & Replication Factor Strategy**

Cassandra is designed to handle big data. Cassandra’s main feature is to store data on multiple nodes with no single point of failure.

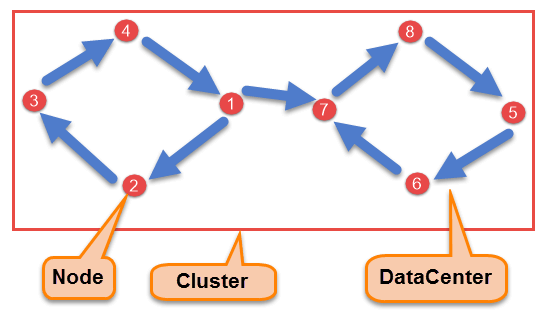
The reason for this kind of Cassandra’s architecture was that the hardware failure can occur at any time. Any node can be down. In case of failure data stored in another node can be used. Hence, Cassandra is designed with its distributed architecture.

Cassandra stores data on different nodes with a peer to peer distributed fashion architecture.

All the nodes exchange information with each other using**Gossip protocol**. Gossip is a protocol in Cassandra by which nodes can communicate with each other.

## Components of Cassandra

There are following components in the Cassandra;

[](https://www.guru99.com/images/cassandra/021116_0524_CassandraAr1.png)

Cassandra Architecture Diagram

* **Node**

Node is the place where data is stored. It is the basic component of Cassandra.

* **Data Center**

A collection of nodes are called data center. Many nodes are categorized as a data center.

* **Cluster**

The cluster is the collection of many data centers.

* **Commit Log**

Every write operation is written to Commit Log. Commit log is used for crash recovery.

* **Mem-table**

After data written in Commit log, data is written in Mem-table. Data is written in Mem-table temporarily.

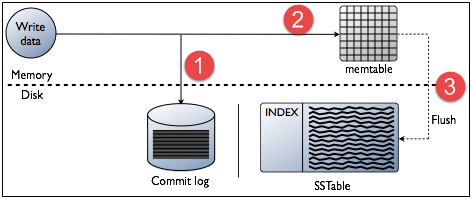
* **SSTable**

When Mem-table reaches a certain threshold, data is flushed to an SSTable disk file.

* **SSTable** - A Sorted String Table (SSTable) ordered immutable key value map. It is basically an efficient way of storing large sorted data segments in a file.

## **Write Operations**

Every write activity of nodes is captured by the commit logs written in the nodes. Later the data will be captured and stored in the mem-table. Whenever the mem-table is full, data will be written into the SStable data file. All writes are automatically partitioned and replicated throughout the cluster. Cassandra periodically consolidates the SSTables, discarding unnecessary data.



## Data Replication

As hardware problem can occur or link can be down at any time during data process, a solution is required to provide a backup when the problem has occurred. So data is replicated for assuring no single point of failure.

Cassandra places replicas of data on different nodes based on these two factors.

* Where to place next replica is determined by the **Replication Strategy**.
* While the total number of replicas placed on different nodes is determined by the **Replication Factor**.

One Replication factor means that there is only a single copy of data while three replication factor means that there are three copies of the data on three different nodes.

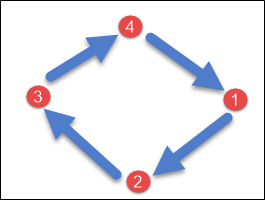
For ensuring there is no single point of failure, **replication factor must be three.**

There are two kinds of replication strategies in Cassandra.

**SimpleStrategy**

SimpleStrategy is used when you have just one data center. SimpleStrategy places the first replica on the node selected by the partitioner. After that, remaining replicas are placed in clockwise direction in the Node ring.

Here is the pictorial representation of the SimpleStrategy.

[](https://www.guru99.com/images/cassandra/021116_0524_CassandraAr2.png)

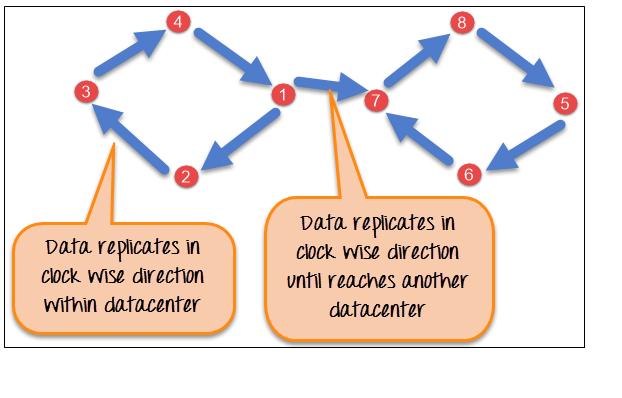
**NetworkTopologyStrategy**

NetworkTopologyStrategy is used when you have more than two data centers.

In NetworkTopologyStrategy, replicas are set for each data center separately. NetworkTopologyStrategy places replicas in the clockwise direction in the ring until reaches the first node in another rack.

This strategy tries to place replicas on different racks in the same data center. This is due to the reason that sometimes failure or problem can occur in the rack. Then replicas on other nodes can provide data.

Here is the pictorial representation of the Network topology strategy

[](https://www.guru99.com/images/cassandra/021116_0524_CassandraAr3.png)

## **Different components of Cassandra Keyspace**

**Strategy:** There are two types of strategy declaration in Cassandra syntax:

* **Simple Strategy:** Simple strategy is used in the case of one data center. In this strategy, the first replica is placed on the selected node and the remaining nodes are placed in clockwise direction in the ring without considering rack or node location.
* **Network Topology Strategy:** This strategy is used in the case of more than one data centers. In this strategy, you have to provide replication factor for each data center separately.

**Replication Factor:** Replication factor is the number of replicas of data placed on different nodes. More than two replication factor are good to attain no single point of failure. So, 3 is good replication factor.

### **What are the differences between a node, a cluster, and datacenter in Cassandra?**

**Node:** A node is a single machine running Cassandra.

**Cluster:** A cluster is a collection of nodes that contains similar types of data together.

**Datacenter:** A datacenter is a useful component when serving customers in different geographical areas. Different nodes of a cluster can be grouped into different data centers.

### **What is the use of Bloom Filter in Cassandra?**

On a request of a data, before doing any disk I/O Bloom filter checks whether the requested data exist in the row of SSTable.

### **What is Kundera in Cassandra?**

In Cassandra, Kundera is an object-relational mapping (ORM) implementation which is written using Java annotations.

Interview

<https://www.edureka.co/blog/interview-questions/cassandra-interview-questions/>

<https://www.javatpoint.com/cassandra-interview-questions-and-answers>