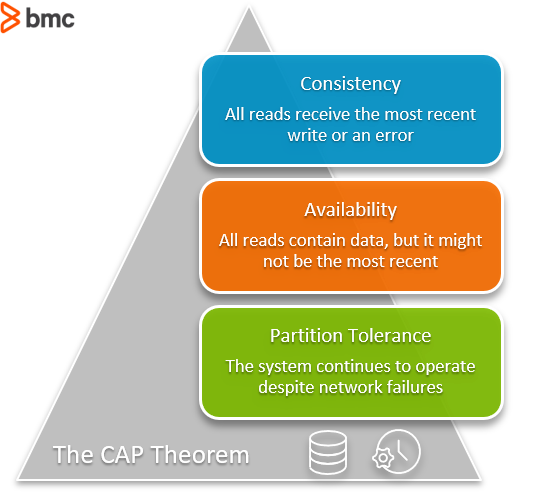
Cassandra

It is a NoSQL distributed database with columns stored in a separate file.

The distributed system should have any two of these three properties(known as the CAP theorem)

1. Consistent – if we want to change some data, then this data should be changed on every node/server/computer. The data should be consistent. One query should retrieve similar results on all the nodes.
2. Available – The data should always be available, even if one cluster/server fails, data should be retrieved from other servers.
3. Partition Tolerant – The system continues to operate despite network failures.



In Cassandra, it is impossible to perform joins, and because of this its read/write operation is efficient. It uses a query-first approach instead of a relational model approach.

Relational data is mostly preferred for small applications which store small amounts of data Whereas Cassandra is used in application which store huge amount of data.

Cassandra Architecture

It doesn’t follow the leader-follower approach because in the leader-follower approach if the leader goes down, then it takes some time to connect to followers and for that span of time application data won’t be available.

It is based on a peer-to-peer architecture which consists of a network of nodes and all the nodes in the network are considered the same i.e there is neither leader nor follower. If data needs to be written then it is written on all the nodes selected for data storage, also there is a coordination node that receives the request for storing/retrieving data from these networks.

There might also be a possibility of network failure, in this case data is written to the nodes which are part of the network currently and the remaining nodes are updated once they are available.

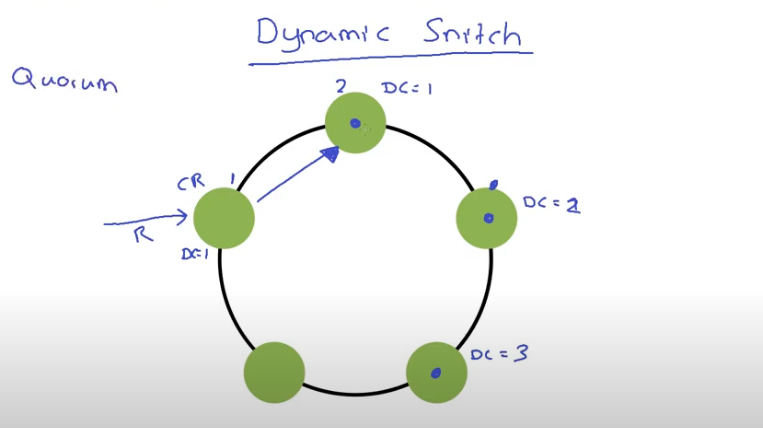
Snitch

In Cassandra, the snitch has two functions:

* it teaches Cassandra enough about your network topology to route requests efficiently i.e it keep records of different nodes (in which data center/rack data is stored on to the server). It maintins a property file for it(Property file snitch).
* it allows Cassandra to spread replicas around your cluster to avoid correlated failures. It does this by grouping machines into "datacenters" and "racks." Cassandra will do its best not to have more than one replica on the same "rack" (which may not actually be a physical location).

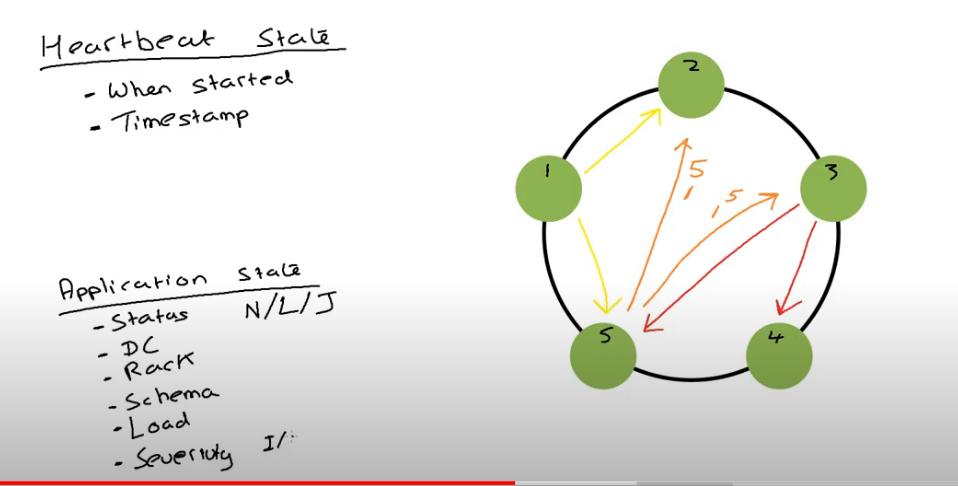
Dynamic Snitching

It keeps records about the performance of different nodes in the cluster.



Gossip Protocol

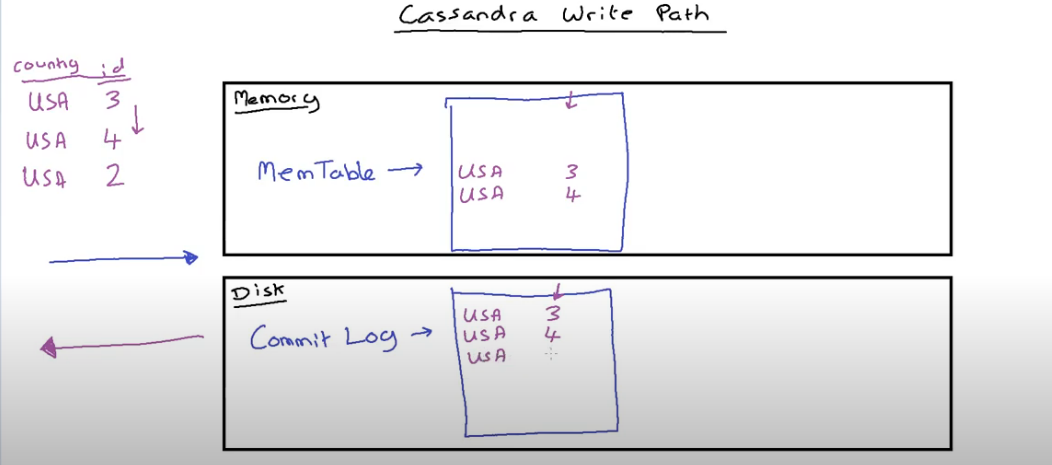
Cassandra uses a protocol called gossip to discover location and state information about the other nodes participating in a Cassandra cluster. Gossip is a peer-to-peer communication protocol in which nodes periodically exchange state information about themselves and about other nodes they know about.



Cassandra Write Path

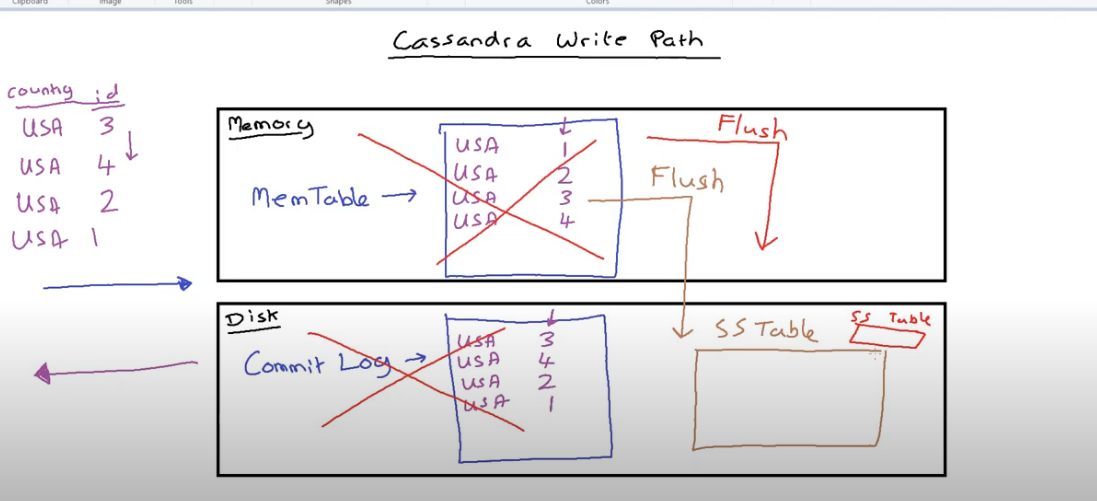
There are 2 places in which Cassandra can store data i.e

1. memory
2. Disk



First data is stored in the commit log in the disk then it is stored in the MemTable of memory. Commit log appends the data it does not store data in a particular order. MemTable stores data in a particular order.

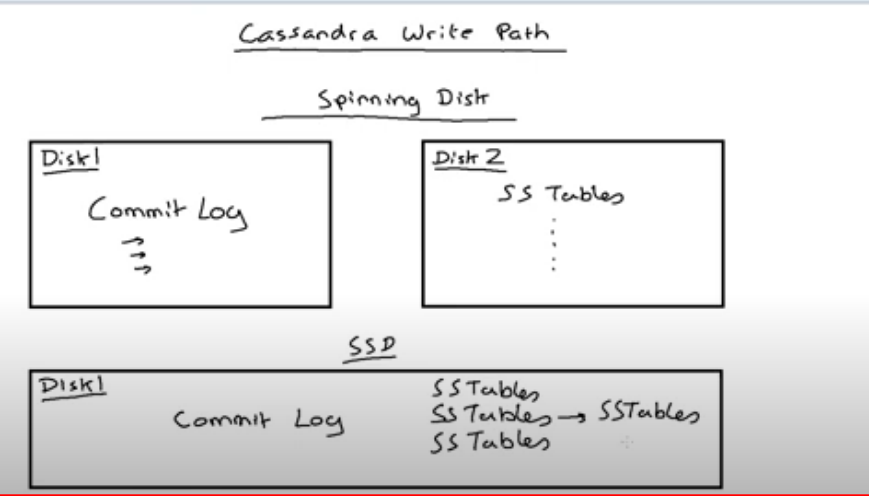
After all the data is stored in MemTable and commit log , then flush is performed and this stores data in SS table and new MemTable and commit log is created.



SS tables are immutable.

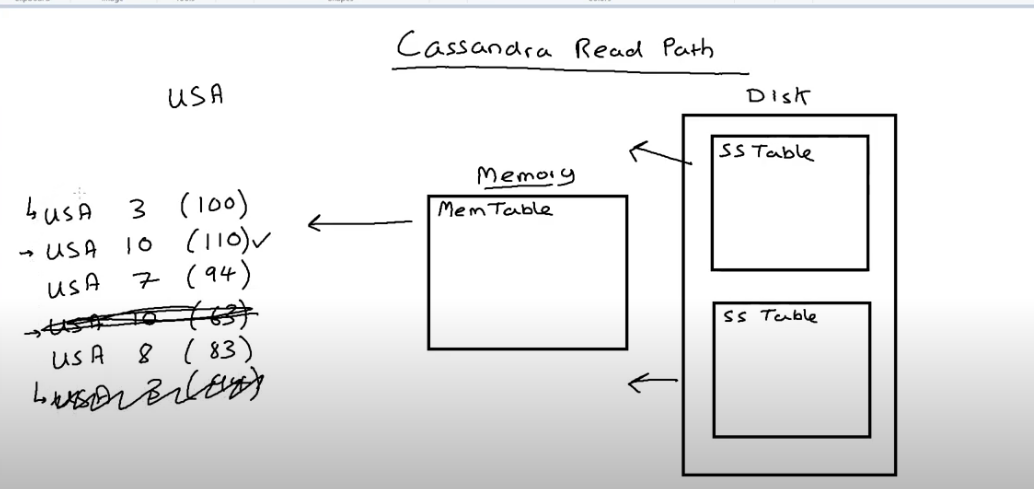
In case of spinning disk, commit log can be created on one disk and SS tables can be stored on another disk.

In case of SSD both can be created on a single disk.

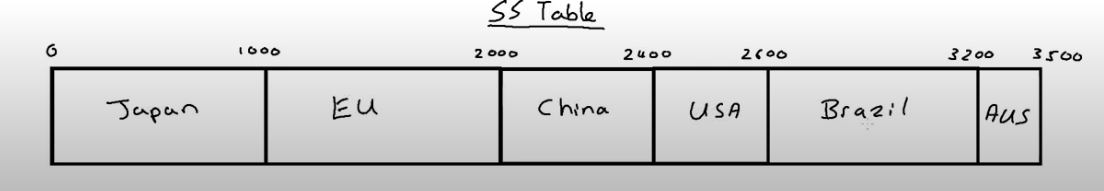


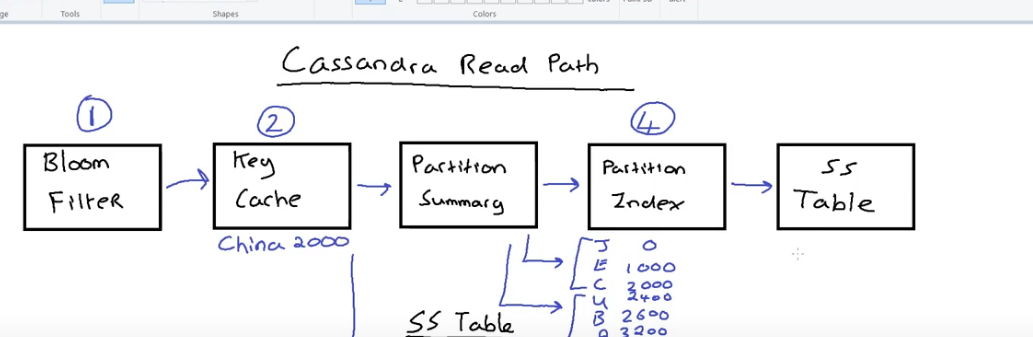
Cassandra Read Path

While reading from Cassandra data may be read from MemTable or SS Table, but mostly it is read from MemTable as it contain updated data.



Cassandra stores data in the SS Table in an ordered manner. We generally cluster the data according to the country.





<https://docs.datastax.com/en/cassandra-oss/3.x/cassandra/dml/dmlAboutReads.html>