**APACHE CASSANDRA**

1. **USE CASES:**

Nearly 1500 organizations use Cassandra for different purposes such as Collecting, Processing and Analyzing data from Sensors (CERN, Internet of Things, Smart Meter), Recommendation to the users from data collected from online websites (Netflix, IMDB, GrubHub), Fraud Detection from multiple resources (Barracuda Networks), Location Based Services (HERE Maps, Google Maps, Uber, Lyft, Hailo) and Segregating data for providing financial advice through Financial Products (Intuit).

1. **Advantages:**

* **High Throughput –** Amount of data getting traversed over the network is very high
* **Massive Scalability –** As the engine is distributed system, scalability is massive which means the need for additional storage systems can be scaled easily based on the amount of incoming data
* **High Availability –** As the data is distributed across different nodes in the cluster, when one node goes down, the other node can be picked up to respond to the write or read requests. Also, the cluster can be distributed across different data centers. Once a data center goes down, the cluster can be accessed from other data centers.

1. **Architecture:**

All the nodes present in a single network is called a Cluster. Every node present in a cluster, has the same functionality as the others.

NODE 1

NODE 4 CLUSTER NODE 2

NODE 3

As Cassandra is a distributed database system, data is distributed across all nodes and multiple servers which gives doorway to horizontal scalability.

Each node exchanges information with other nodes across the cluster.

After every write, commit log ensures data durability by capturing the write activity.

Data to be written to the node are first indexed and written to in-memory table called **MemTable** (can be called as “Write Back Cache”).

During, the process of write the data into the MemTable, when the MemTable is full, data is then written to the disk called **SSTable**.

All writes are automatically partitioned and distributed across all nodes throughout the cluster.

Cassandra periodically consolidates SS Table by flushing out unwanted data using a process called **“Compaction”**. **Tombstone** is marker in a row that indicates a column was deleted. Hence, Cassandra is a row oriented database.

An **Authorized** person can connect to any node and access data using **CQL**.

In Cassandra, typically a cluster has **One Keyspace** per application. When a client gets connected to a Node, that Node serves as a coordinator for that client operation. A coordinator typically decides what nodes to be requested in the ring for fetching the data based on the partitioning of data and placement of replicas.