Cassandra

Day 1

Day 1 - Overview

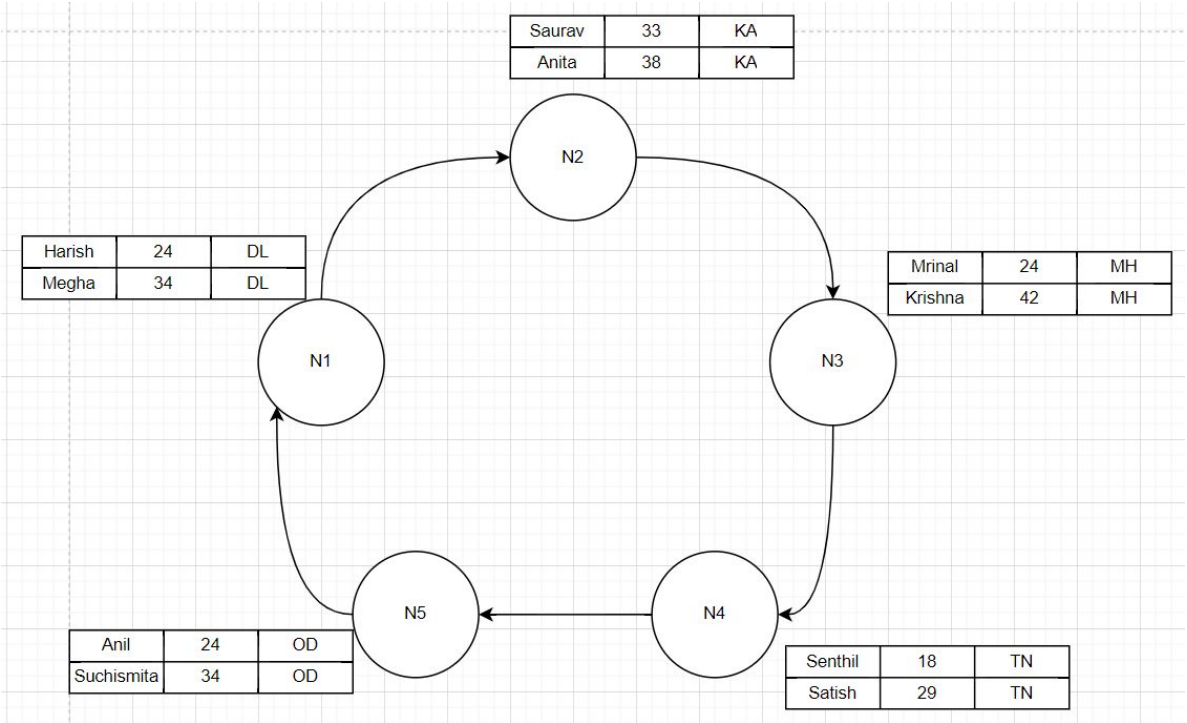
- Introduction to Cassandra
- Distributed Architecture
- How Data gets distributed
- Cassandra vs SQL databases
- Cassandra vs other NoSQL
- Cassandra Use Cases
- Installing Cassandra

Introduction to Cassandra

Apache Cassandra® is a distributed NoSQL database used by the vast majority of Fortune 100 companies. By helping companies like Apple, Facebook, and Netflix process large volumes of fast-moving data in a reliable, scalable way, Cassandra has become essential for the mission-critical features we rely on today.

Understanding Distributed Nature of Cassandra

```
CREATE TABLE user (
        email text,
        age int,
        state text,
        PRIMARY KEY ((state), email)
);
```



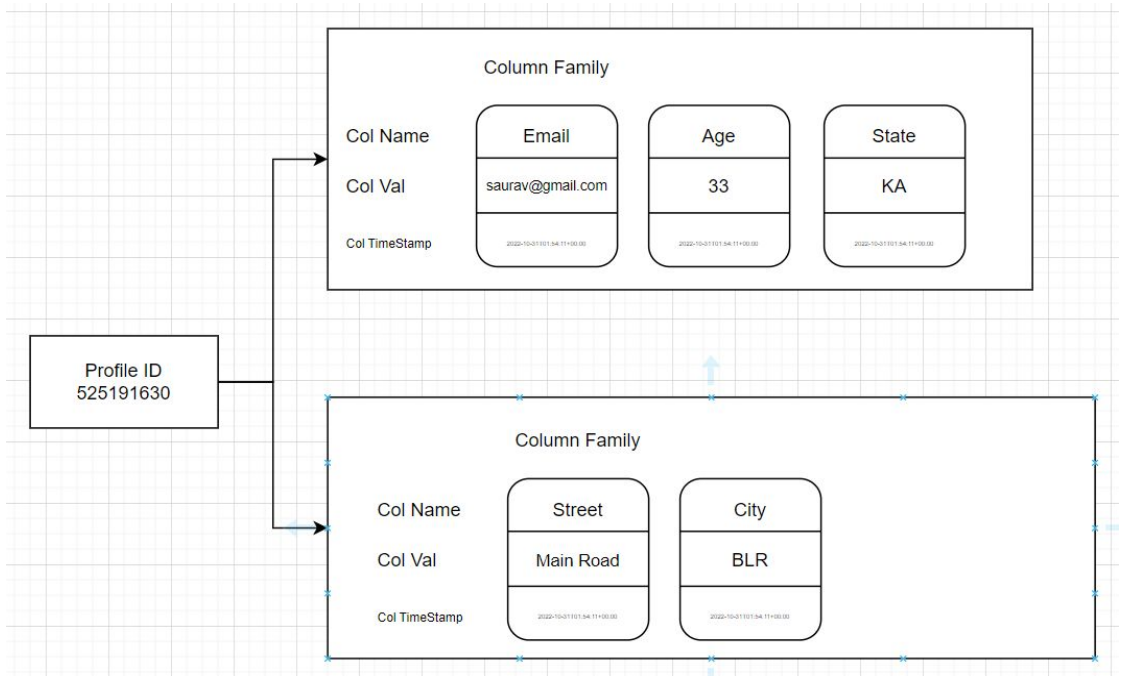
Cassandra vs SQL

Relational database management systems (RDBMS) dominated the market for decades. Then, with the rise of Big Tech like Apple, Facebook, and Instagram, the global datasphere skyrocketed 15-fold in the last decade. And, RDBMS simply weren't ready to handle the new data volume or the new performance requirements.

NoSQL was not only invented to cope with massive volumes of data, but also to tackle the challenge of both **velocity** (speed requirements) and **variety** (all the different types of data and data relations in the market).

Cassandra	SQL
The primary database model for Cassandra is Wide column store or Column Family	The primary database model for PostgreSQL is Relational DBMS.
It supports Secondary indexing but in a restricted way, i.e., only equality queries, not always the best performing solution.	It supports Secondary indexing.
It does not supports Server-side scripting.	It has user defined functions for Server-side scripts.
It supports selectable replication factor method.	Native replication is not supported
In Cassandra, partitioning or Sharding can be done	Native partitioning or sharding is not supported
It offers an API for user-defined Map/Reduce methods.	It does not offers an API for user-defined Map/Reduce method.
Doesn't have the concept of foreign key	SQL provides the concept of Referential Integrity and have Foreign keys.

Data Model



Cassandra vs Other NoSQL

Other than Wide Column databases like Cassandra, we've also seen the rise of other types of NoSQL databases, such as:

Time-series databases (e.g. **Prometheus**)

Document databases (e.g. **MongoDB**)

Graph databases (e.g. **DataStax Graph**)

Ledger databases (e.g. **Amazon QLDB**)

Key/value databases (e.g. **Amazon DynamoDB**)

Apache Cassandra Use Case

E-commerce and inventory management

- Resilient with zero downtime: Distributed with multi-region replication, Cassandra ensures zero downtime. Even the loss of an entire region won't bring it down.
- Highly responsive: Cassandra's peer-to-peer architecture also allows data to reside in regions around the world and closer to any particular customer—allowing the system to be highly responsive and fast.
- Analyzes its catalog and inventory in real time.

Apache Cassandra Use Case

Personalization, recommendations, and customer experience

- Fast response times.
- Extremely low latency, even as your customer base expands.
- Handles all types of data, structured and unstructured, from a variety of sources.
- Built to scale while staying cost-effective.
- Ability to store, manage, query, and modify extremely large datasets, while delivering personalized experiences to millions of customers.
- High read/write capacity.

Apache Cassandra Use Case

Internet of things and edge computing

- Cassandra can ingest concurrent data from any node in the cluster, since all have read/write capacity.
- Ability to handle a large volume of high-velocity, time-series data.
- High availability.
- Supports continuous, real-time analysis.

Installing Apache Cassandra in Windows

Follow the below document to setup Cassandra in your VM

https://github.com/saurav-samantray/flask-microservices-training/blob/main/slides/Setup%20Apache%20Cassandra%20on%20Windows.pdf

Q and A