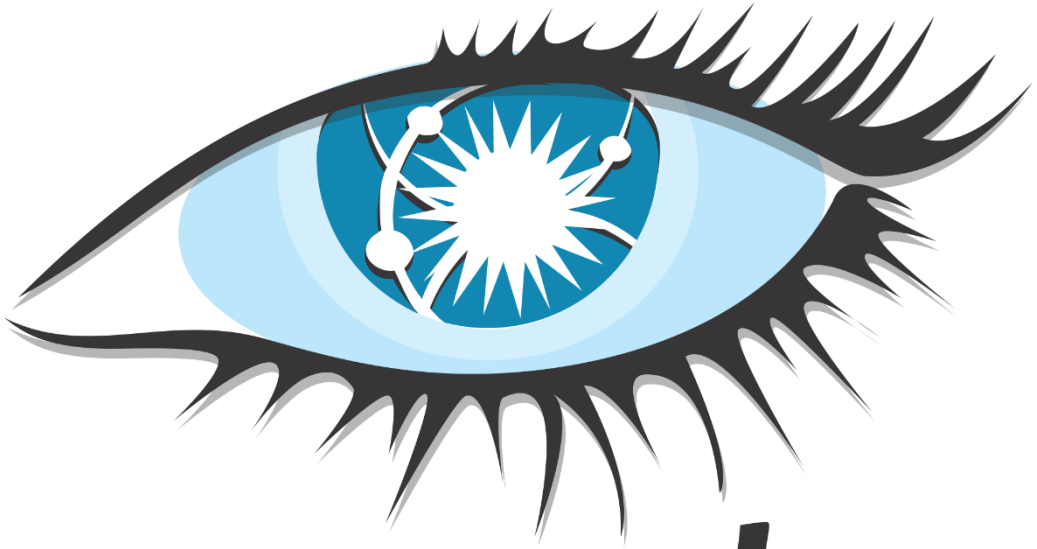


A.A.KABIRI ZAMANI

Overview

- History
- What is Cassandra?
- Properties
- Structure
- Durable Writes
- Creating Keyspaces, Tables, etc
- Who’s using Cassandra?

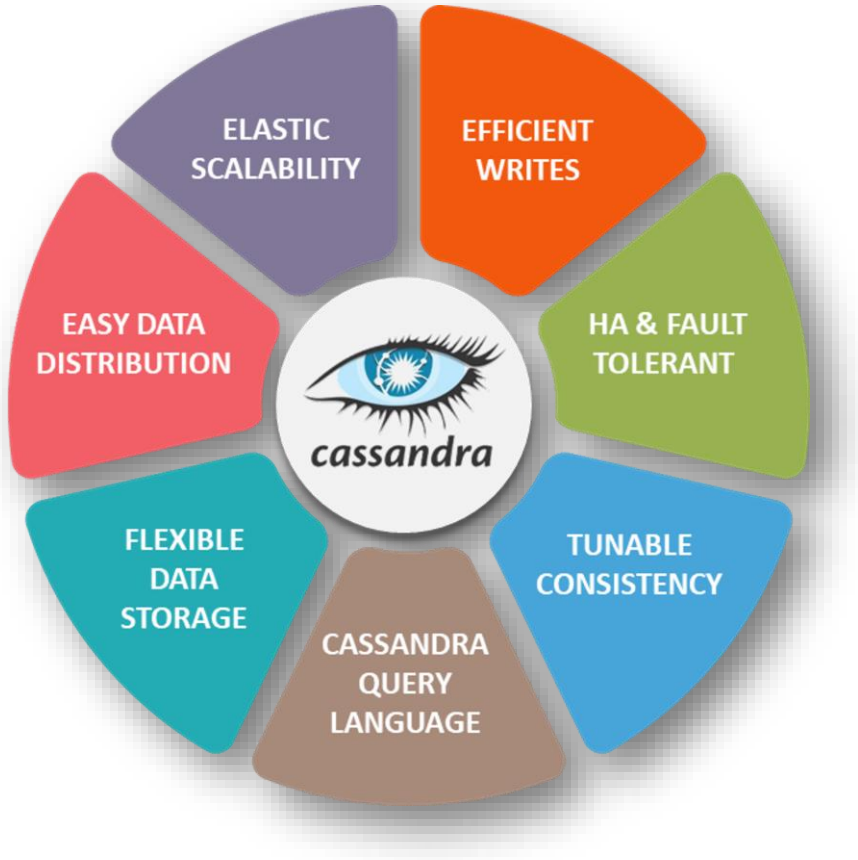
History

- Initially developed to power the Facebook inbox search feature by:
  - Avinash Lakshman: One of the authors of Amazon's Dynamo
  - Prashant Malik: The investor!
- Released in July 2008 as an open-source project

What is Cassandra?

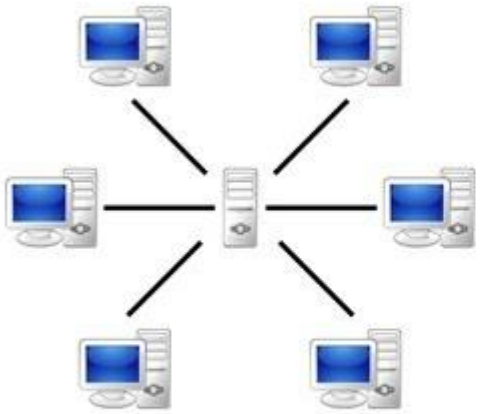
- Apache Cassandra is a
  - Free
  - Distributed
  - High performance
  - Extremely scalable
  - Fault tolerant
  - Post-relational

Database.

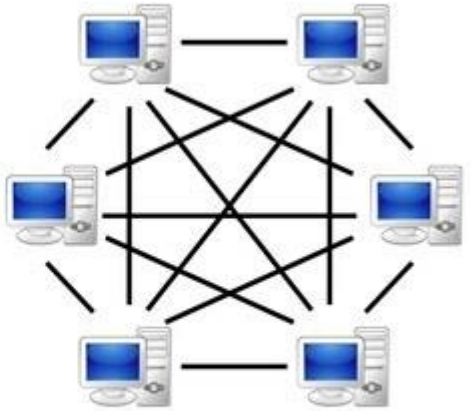


Properties

- Peer-to-peer distributed system
  - Rather than Master-Slave Structure
  - Using Gossip protocol
- Fault tolerant
  - No single point of failure
- Scalable
  - Capable of handling proliferating requests only by adding linear number of nodes



Server-based



P2P-network

Properties

- Tunable consistency
  - Following CAP theorem
  - Availability and partition tolerance are generally considered to be more important than consistency
- MapReduce support

Structure

- Cassandra's CQL interface return data in tabular format and it might give the illusion that we can query it just like any RDBMS, but that's not the case
- All Cassandra data is persisted in SSTables
- Cassandra versus Relational model is as follow:

Relational Model	Cassandra Model
Database	Keyspace
Table	Column Family
Primary Key	Row Key
Column Name	Column Name/Key
Column Value	Column Value

Structure

- The Cassandra data model consists of Keyspaces (analogous to databases), column families (analogous to tables in the relational model), keys and columns.
- Creating a Keyspace:
  - CREATE KEYSPACE ks1 WITH replication={'class':'SimpleStrategy','replication_factor':1};
  - select * from system_schema.keyspaces where keyspace_name = 'ks1';

Strategy Name	Description
Simple Strategy	Specifies a simple replication factor for the cluster.
Network Topology Strategy	Using this option, you can set the replication factor for each data-center independently.
Old Network Topology Strategy	This is a legacy replication strategy.

Durable Writes

- Durable Write: When a write request is received, the node first writes a copy of the data to an on-disk append-only structure called Commitlog. Then, it writes the data to an in-memory structure called Memtable. When the Memtable is full or reaches a certain size, it is flushed to an on-disk immutable structure called SSTable. Setting durable writes to true ensures data is written to the Commitlog. In case the node restarts, the Memtables are gone since they reside in memory. However, Memtables can be reconstructed by replaying the Commitlog, as the disk structure won't get wiped out even with node restarts.

Creating a Table

- CREATE TABLE user_tracking (user_id text, action_category text, action_id text, action_detail text, PRIMARY KEY(user_id, action_category, action_id));

- select * from system_schema.tables where keyspace_name = 'ks1';

Add a Record

- insert into ks1.user_tracking(user_id, action_category, action_id, action_detail) VALUES ('user1', 'auth', 'a1', 'Logged in from home page');

- SELECT * FROM user_tracking ;

Glossary

- **Gossip protocol:** A gossip protocol is a procedure or process of computer peer-to-peer communication that is based on the way that epidemic spread. Some distributed systems use peer-to-peer gossip to ensure that data is routed to all members of an ad-hoc network. Some ad-hoc networks have no central registry and the only way to spread common data is to rely to each member to pass it along to their neighbors.
- **No single point of failure:** Every node in the cluster has the same role. There is no single point of failure. Data is distributed across the cluster (so each node contains different data), but there is no master as every node can service any request. In the case of server's confusion, there is not any node which can play the role of the server.
- **CAP Theorem:** states that it is impossible for a distributed data store to simultaneously provide more than two out of the following three guarantees:
  - Consistency: Every read receives the most recent write or an error
  - Availability: Every request receives a (non-error) response – without the guarantee that it contains the most recent write
  - Partition tolerance: The system continues to operate despite an arbitrary number of messages being dropped (or delayed) by the network between nodes

Glossary

- **MapReduce:** It is a programming paradigm that enables massive scalability across hundreds or thousands of servers in a Hadoop cluster.
- **Apache Hadoop:** The Apache™ Hadoop® project develops open-source software for reliable, scalable, distributed computing.
- **CQL:** Stands for Cassandra Query Language.
- **SSTables:** Sorted Strings Table is a file of key/value string pairs, sorted by keys.