Template for the Research Paper of ADB

Fname1 Lname1, Fname2 Lname2, Fname3 Lname3

School of Computer Science & Information Systems

Northwest Missouri State University

Abstract

This article describes how you should write up your research paper.

You must write your article with the same format as this document.

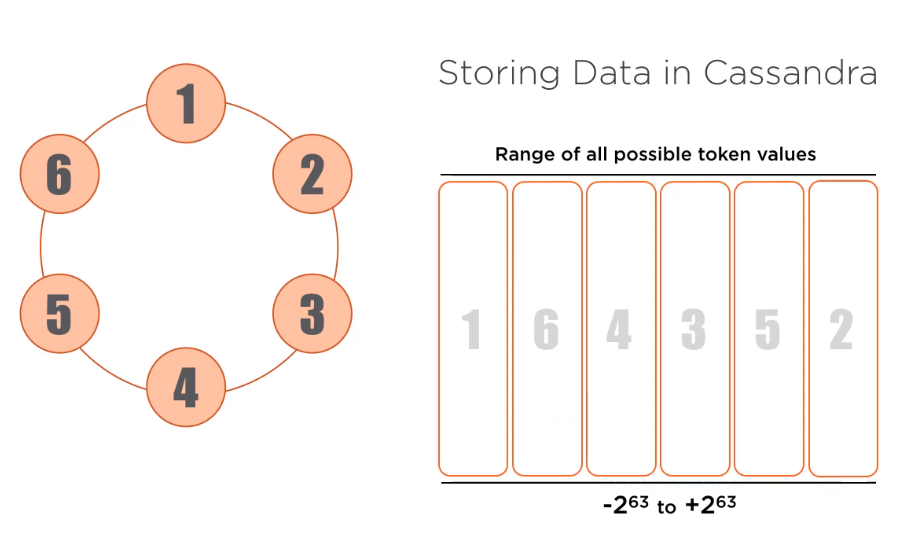
Please read this document carefully.

1. Introduction

Apache Cassandra is a distributed open source database that can be referred to as a "NoSQL database" or a "wide column store". Cassandra has been used by some of the world’s most largest companies to leverage the service that expands the globe.

1. What is Cassandra?
2. We shall start with the brief history of Cassandra, Apache Cassandra is a NoSQL, or “non-relational,” database. Cassandra a top level Apache project was born at Facebook, back in 2008. It was designed as the store for a Facebook feature Inbox Search. Such a feature requires a storage system that can handle a tremendous number of rights as well as geographical replication to reduce search latencies for end users.
3. Like most modern distributed CSQL systems Cassandra was founded on the principal outline in 2 related papers on the theory: Googles Cloud Big table and Amazon Dynamo. Cassandra integrate the distributed nature of Dynamo and data model of Googles big table. Facebook open sourced Cassandra in 2008 and in 2009 it became an apache incubator project in 2010.
4. Netflix recently migrated from Oracle to Cassandra running in Amazon public cloud, Apple had also exhibit to benefited from Cassandra running more than 75000 nodes and storing tons of Terra bytes of data.

Cassandra Cluster



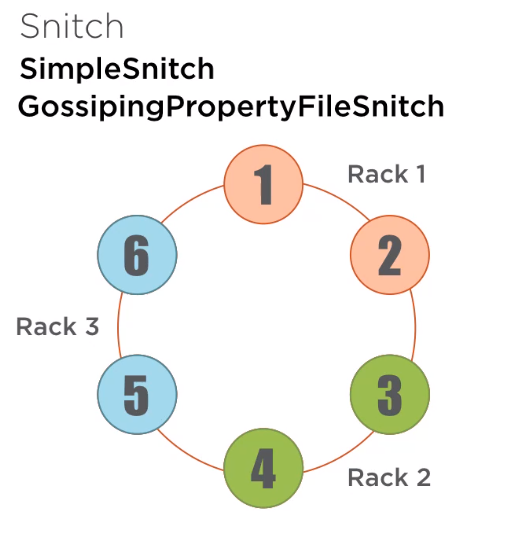
This diagram is probably the most common representation of a Cassandra installation. Each dot on the circle is a Node, which represent a running instance of Cassandra. This diagram helps strive on the point that a Cassandra cluster is a true master-less peer to peer system with no single point of failure. All nodes can perform all Cassandra function.

1. How Cassandra operates under the hood

All data stored in Cassandra is associated with a token, there are astronomical number of possible token values, falls in this range -2 63 to + 2 63. As new node are added each node taking over a contiguous range of token values and storing the data associated with it. 2nd node, 3rd node, and so on till 6th node.

1. Structure Data in Cassandra : Snitches

A snitches is what Cassandra uses to gain an understanding of an environment physical or virtual, in which the cluster is been run. It is used efficiently to round requests and is consulted when storing multiples copies of the data. The default name is “SimpleSnitch” is suitable for development in single data environments. A much more interesting snitch is the "GossipingPropertyFileSnitch" Gossip is the protocol Cassandra nodes use to talk to one another and keep everyone up to date on the state of the cluster.



1. Sample data
2. Consistency levels and Replication Strategies

We looked at how a single copy of data is distributed across the cluster. Here we will discuss Replication Strategies to store multiple copies of data in a cluster. Cassandra support for Tunable Consistency, while reading and writing data to the cluster.

1. Replication strategies

A virtual node shows data been written to a specific V node owned by a node in the cluster. In a Cassandra it is expect to store multiple copies of the data on different nodes throughout the cluster.

This gives increased reliability as well as performance. Not only it can easily tolerate a node becoming unavailable but in certain circumference we may choose to read a specific copy of the data from a node for example: In a data center geographically closer to the system making the query.

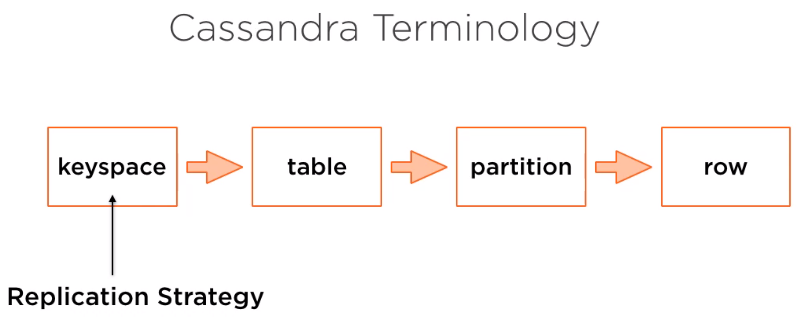
1. Cassandra Terminology

Keyspaces: At the highest level data in Cassander in organized into keyspaces. The closest analogy to this in the relational world would be oracle or mysql Tablespace.

Tables: With in a Cassandra keyspaces there are one or more table. A table here is a pretty close match conceptually to it is relational counter-part.

Partitions: All data written to Cassandra is associated with a partition key. This partition key determines where the data is located in the cluster, and all data in a partition is stored together.

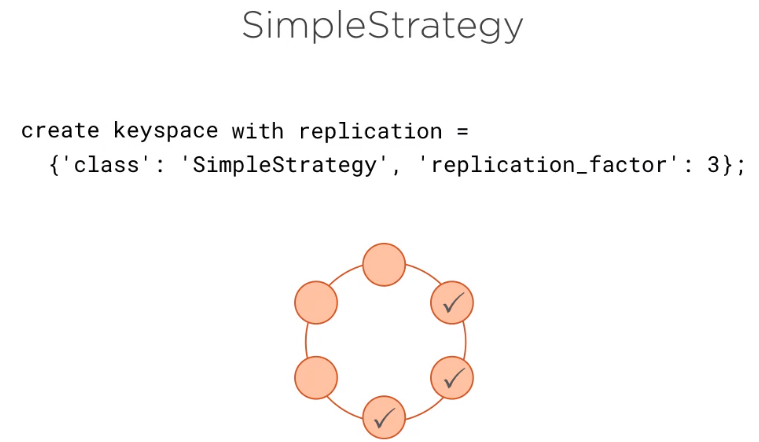
The partition is the primary interaction point when reading or writing data to Cassandra. Finally data within a partition may be represented as one or more rows.



The specifics of a replication strategy are to determine at the key space levels, if the partition key is used to determine the location of the first copy of the data written to a Cassandra cluster, the keyspace settings are used to determine the number of copies of the data and where they are stored throughout the cluster.

The two strategies for configuring this replication:

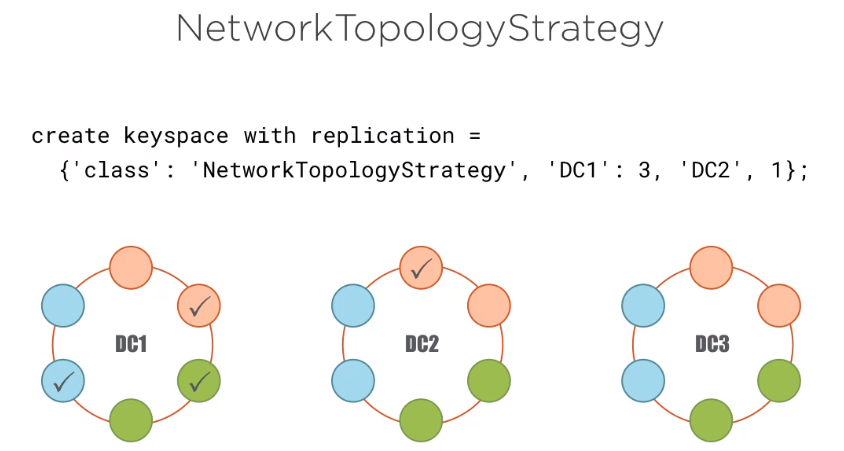
**Simple Strategy**: As the name says it's the best used in development environments or single data center clusters. Example a SQL to create a keyspace with a simple strategy replication, we are asking Cassandra to store 3 copies of all the partitions in all the tables written to the cluster in this keyspace.



We can see a pattern here with Snitches and replication strategies working hand in hand. And a clustered configured with a simple Snitch. This replication strategy is already available, we can ask Cassandra to store multiple copies of data and it will do its best to store them on different nodes.

More interesting is network topology strategy, which is configured here.

**Network Topology Strategy**: We are not simply specifying how many copies of data to store, but instead enumerating each data center and specifying how many copies of data are stored in each. In this case, we are storing 4 copies of data for each partition in each table in keyspace. But more than that we are telling Cassandra we tell to store 3 copies in Data Center 1 and 1 in Data center 2.



This replication strategy is ideal for production environments especially in multi data center clusters. It requires an understanding the clusters topology which requires the cluster to be configured with an appropriate Snitch such as “GossipingPropertyFileSnitch”or one specific to cloud provider.

Note: It not only puts the replicas of the data in multiple data centers as requested but also uses the knowledge of Racks to smartly distribute the replicas within each data center as well.

Tunable consistency

1. Sample
2. Sample 2
3. Making the most of Cassandra

basics of how Cassandra works

Consistency levels and replication strategies

Cassandra Query Language

More recent additions

Materialized views

Secondary indexes

Batches

Transactions

User defined functions & aggregates