Replication Strategies

Determines how many copies of data are stored

SimpleStrategy

Best for dev setups and single node clusters

NetworkTopologyStrategy

Indicating each datacenter and how many copies in each datacenter to store

Requires an appropriate snitch be configured, such as a GossipingPropertyFileSnitch

Dynamically determines which nodes to copy data to based on datacenters and RACs

## Tunable consistency

Nodes in a cluster may become unresponsible for any of several reasons

Node may be down, or slow

Consistency

How many acknowledgements of an action from replicas is required before the coordinator responds with a positive response

Consistency guarantees can be tuned on a per statement level as opposed to at a keyspace or other level

### Write consistency

Guarantees that are provided that write attempts to the database are stored redundantly such that the data survives any failure of a cluster node

Write Consistency levels

1 means only one replica must respond before the coordinator returns positively

Quorum indicates a majority of nodes in the cluster must ack

All

Any even if write goes to just the coordinator, it’s considered a success

In the event of node failure during a write, various strategies are employed by the cluster to ensure the data eventually is replicated to the node.

Hinted handoff

If a node fails during write, the coordinator takes note and repeatedly re-attempts the write until it succeeds.

If the coordinator fails before that can succeed, a different strategy is employed by the cluster.

### Read consistency

Indicates how data is consumed from the cluster such as to retrieve the most up to date information

Tunable consistency allows the caller to specify on an action level how important it is to receive the most up-to-date information.

In a read, the coordinator consults with a specified number of nodes to retrieve data.

The coordinator pull data from one node, and a digest from other nodes.

Read consistency values

one - the value obtained from a single node is good enough

two-...

quorum - data from one node and a majority of digests will be used

all - data from one node and digests from all nodes holding copies of the data will be consulted

In a multi-datacenter setup, there are additional levels of consistency you can achieve

Each\_quorum - operation must succeed in a majority of nodes in each DC

Local\_quorum - operation must succeed in a majority of nodes in only the local DC where the coordinator exists

Local\_one - operation must succeed only on the coordinator node of the local DC

Allows you to have more control over when you want to cross over the datacenter boundary

### Read Repair

What if a hinted handoff failed earlier and one of the nodes has old data?

db will read data from one node and digests from other nodes

digest of the out-of-date node will differ from other nodes

node coordinator will then request data (not digest) from the nodes

once it determines which data is newest, it will then write out the correct data to the out-of-date node. (Read repair)

Read repair may either occur as a part of a query as part of the queries Read Consistency guarantees, or in a background process.

Read Repair has overhead. DB can be tuned to migrate some of this to background processing as opposed to as part of queries themselves (TODO: Is that correct?) Configurable on a per-table basis

Only affects data being read.

### Consistency

How do you achieve strong consistency?

By default, Cassandra supports eventual consistency

To achieve:

(Write consistency + Read consistency) > Replication Factor

Examples

Write Read RepFactor Test?

1 Q 3 N The Quorum may only include nodes not written to.

1 A 3 Y

1 1 3 N

Q Q 3 Y

Data is expected to be stored redundantly across different nodes in a cluster

Gives you increased reliability and performance improvements

Storing data redundantly protects you from node failures and supports targetted reads from nodes that are geographically close

## Reads and Writes

### Coordinator node

When a client connects to Cassandra to perform an action, it connects to a coordinator node

The coordinator node acts as an agent and determines which node(s) should actually process the given command

The coordinator then communicates with those nodes (replica nodes)

Across interactions with cassandra for different clients, different nodes may act as the coordinator node.

So though any node in a cluster may perform any function, in the context of an interaction, each node takes on a specific role.

In some cases, coordinator nodes may also act as replica nodes

### Replica node

Terms

Keyspace

Highest abstraction of data storage

Similar to a mysql tablespace

Contains tables

configured with a replication strategy indicating how many copies of any particular piece of data are stored

A table is stored in partitions

Partitions are allocated based on partition key

All data in a partition is stored together

Clients interact most directly with the partition when reading / writing from / to cassandra

Data in a partition may be stored in one or more rows

## 

Digest - a hash representation of data to be retrieved. Used as part of read-consistency mechanism to facilitate providing client with up to date data.