

Wide-column Stores

Introduction	1
How it relates to a key-value store	2
Case study: Apache Cassandra	3
Data model	3
Data distribution	6
*References	8

Introduction

A **Columnar** data store will store each column separately on disk.

Wide-column stores such as Bigtable and Apache Cassandra are **not column stores**.

Wide-column stores support **column families** that are stored separately. Each column family typically contains **multiple columns** that are used together. Within a given column family, all data is stored in a **row-by-row** fashion, such that the columns for a given row are stored together.

Row A	Column 1	Column 2	Column 3
	Value	Value	Value
Row B	Column 1	Column 2	Column 3
	Value	Value	Value

* Each row may have different columns. It is schemaless.

How it relates to a key-value store

We look at wide-column stores as a 2-dimensional key-value store, where the first key is used as a row identifier and the second is used as a column identifier.

Key-value store	Wide-column store
key = rowID, value = data	key = (rowID, columnID), value = data

Key-value store:

Key	Value
-----	-------

Wide-column store:

Key	Col1	Col2	Col3	Col4	Col5
	Val1	Val2	Val3	Val4	Val5

Case study: Apache Cassandra

Data model

A data model represents **entities** and their **relationships**. Cassandra's **performance** is highly dependent on the way the data model is designed.

Cassandra's data model consists of four building blocks:

name
value

Column

super column name		
name	...	name
value		value

Super Column

row key	name	...	name
	value		value

Column Family

row key	super column name			...	super column name		
	name	...	name		name	...	name
	value		value		value		value

Super Column Family

Columns and Super Columns are both a tuple of a **name** and a **value**. The difference is that a Column's value is a "**string**" but a Super Column's value is a **map of Columns**.

The Column Family and Super Column family simply gives a **row id** to the first two models in the picture for quicker retrieval of information.

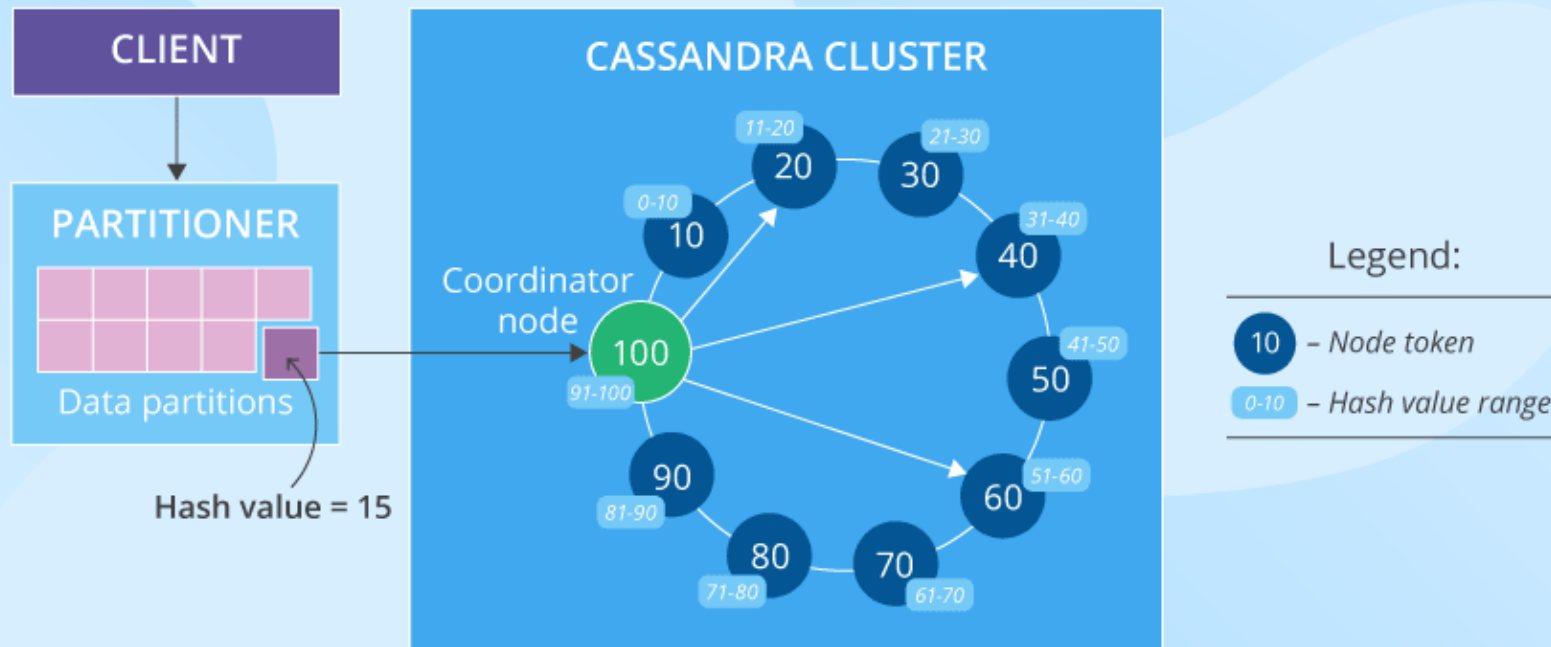
Note that data is always **sorted**: (Super) Columns are always sorted by their **name**. There are built-in data types for (Super) Columns' names. The comparison method can be customized.

In Cassandra, data modeling is **query-driven** (while in relational databases, data modeling is table-centric). That means each query is backed by a table, so data is duplicated across multiple tables.

“dogmas” of Cassandra’s data modeling:

- Disk space is cheap.
- Writes are cheap.
- Network communication is expensive.

PARTITIONING IN CASSANDRA

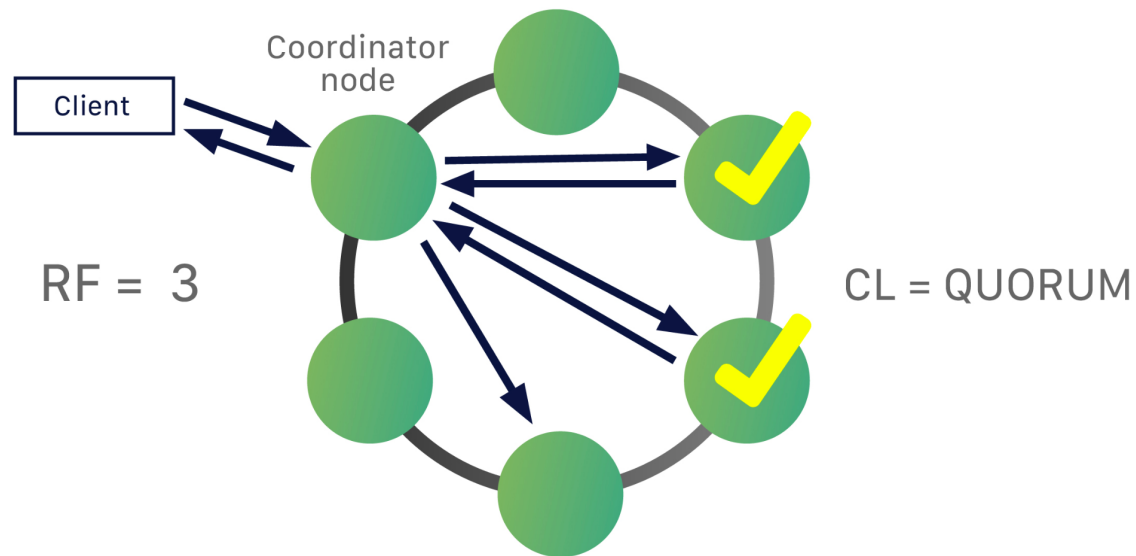


A partitioner is a **hash function** for computing the token (it's hash) of a **partition key**. Each **row** of data is uniquely identified by a partition key and distributed across the cluster by the value of the token.

Nodes communicate with one another through a protocol called **gossip**, which is a process of computer peer-to-peer communication. Cassandra also has a **masterless** architecture – any node in the database can provide the exact same functionality as any other node

When data comes in, the **coordinator** node takes on the job of assigning to a given partition. The coordinator node isn't a single location; the system would be fragile if it were. It's simply **the node that gets the request** at that particular moment. Any node can act as the coordinator.

Cassandra is by default an **AP** (Available Partition-tolerant) database, hence it is “always on”. But you can indeed **configure** the **consistency** on a per-query basis. In this context, the consistency level represents the minimum number of Cassandra nodes that must **acknowledge** a read or write operation to the coordinator before the operation is considered successful.



*References

https://en.wikipedia.org/wiki/Wide-column_store

<https://cassandra.apache.org/doc/latest/>

<https://www.scylladb.com/glossary/wide-column-database/>

<https://dandkim.com/wide-column-databases/>

<https://stackoverflow.com/questions/62010368/what-exactly-is-a-wide-column-store>

<https://arin.me/post/40054651676/wtf-is-a-supercolumn-cassandra-data-model>

<https://www.scnsoft.com/blog/cassandra-performance>