cct | College Dublin
Computing • IT • Business

Big Data Storage and Processing

MSc in Data Analytics

CCT College Dublin

Apache Cassandra

Week 7

Lecturer: Dr. Muhammad Iqbal*

Email: miqbal@cct.ie

Agenda

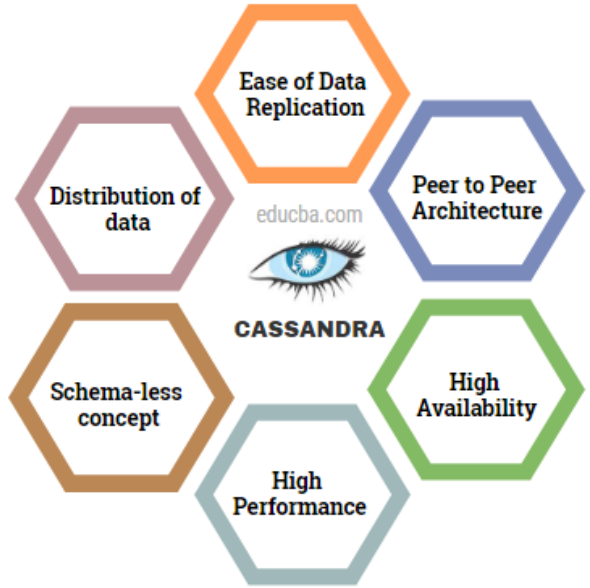
cct College Dublin
Computing • IT • Business

- Introduction to Cassandra
- CAP Theorem
- Characteristics of Cassandra: Distributed and Decentralized, Elastic Scalability, High Availability and Fault Tolerance, Tuneable Consistency, Row-Oriented and High Performance
- Is Cassandra “SCHEMA-FREE”?
- Cassandra Architecture
- Data Replication in Cassandra
- Components of Cassandra
- Cassandra Data Model
- Cassandra **CQLSH**
- Write and Read Operations

Introduction to Cassandra

cct | College Dublin
Computing • IT • Business

- The **Apache Cassandra** data storage system differs greatly from a relational database management system in many ways. In January 2009, Apache Cassandra was first incubated as a project.
- Apache Cassandra is an **open source, distributed, decentralized, elastically scalable, highly available, fault-tolerant, tuneably consistent, column-oriented database** that bases its distribution design on Amazon's Dynamo and its data model on Google's Bigtable.
- Created at Facebook and it is now used at some of the most popular sites on the Web.
- In addition to performing blazingly fast writes, it can store hundreds of terabytes of data, and it is decentralized and symmetrical, so the failure point is eliminated. It provides ***schema-free data models*** and is highly available.
- Cassandra's interface allows it to be accessed from a variety of languages, including **C#, Scala, Python, and Ruby**.



cassandra

What is Cassandra?

cct College Dublin
Computing • IT • Business

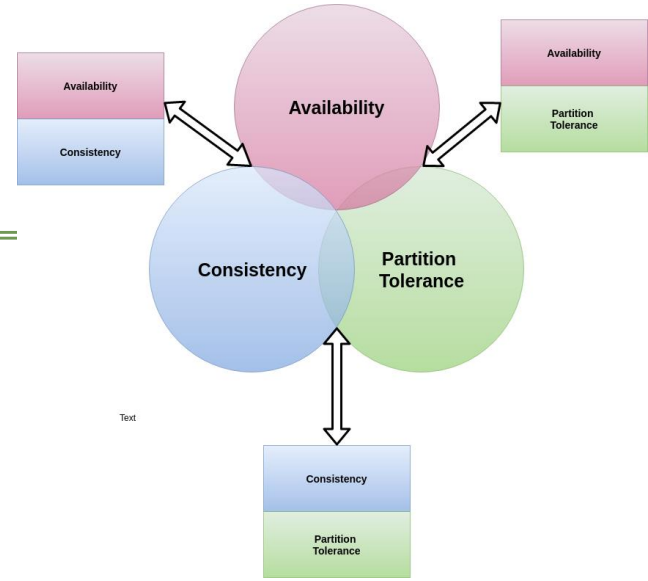
- **Cassandra** is a fully **distributed**, **masterless database**, offering superior scalability, and **fault tolerance** as compared to traditional single-master databases.
- Many of the world's leading technology firms, such as Apple, Netflix, and Instagram, are using **Cassandra** and are open-sourcing, including new features.
- It is being actively developed and has one of largest open-source communities with close to 200 contributors and over 20,000 commits.
- Some of the companies that use Cassandra in their production clusters
  - IBM, Adobe, HP, eBay, Ericsson, Symantec
  - Twitter, Spotify
  - PBS Kids
  - Netflix: uses Cassandra to keep track of your current position in the video you're watching.

https://www.jcount.com/7-reasons-netflix-uses-cassandra-databases/

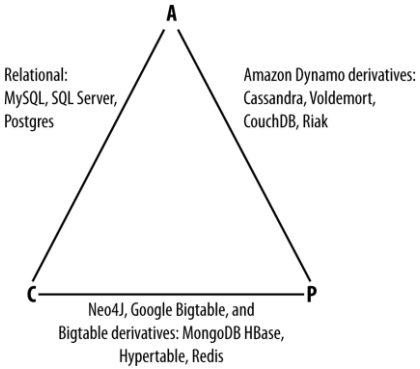
CAP Theorem

Revision

- **Consistency:** All database clients will read the same value for the same query, even given concurrent updates.
- **Availability:** All database clients will always be able to read and write data.
- **Partition tolerance:** The database can be split into multiple machines; it can continue functioning in the face of network segmentation breaks.
- **CA:**
  - To support **consistency** and **availability** means that you are likely using two-phase commit for distributed transactions.
  - It means that the system will block when a network partition occurs, so it may be that your system is limited to a single data center cluster in an attempt to mitigate this.
- **CP:**
  - To support **consistency** and **partition tolerance**, you may try to advance your architecture by setting up data shards in order to scale.
  - Your data will be consistent, but you still run the risk of some data becoming unavailable if nodes fail.
- **AP:**
  - To support **availability** and **partition tolerance**, your system may return inaccurate data, but the system will always be available, even in the face of network partitioning.
  - **DNS** is the most popular example of a system that is massively scalable, highly available, and partition tolerant.



CAP theorem indicates that you can realize only two of these properties at once

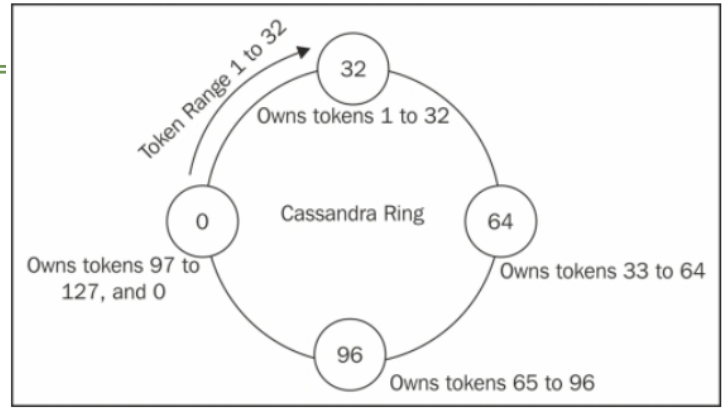


Different databases appear on the CAP continuum

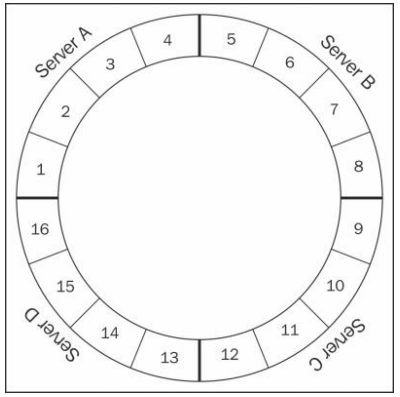
Characteristics of Cassandra

Distributed and Decentralized

- Cassandra is distributed, which means that it is capable of running on multiple machines while appearing to users as a unified whole.
- For optimizing performance across multiple data center racks, and for a single Cassandra cluster running across geographically dispersed data centers, we can confidently write data to anywhere in the cluster and Cassandra can get it.
- Once we start to scale many other data stores, such as MySQL, Bigtable, some nodes need to be set up as masters in order to organize other nodes, which are set up as slaves.
- The feature of decentralization basically means there is no master-slave paradigm or identical nodes. Cassandra has each and every separate node that has the capacity to present itself to an end-user as a replica, complete or partial, of the database.
- **The fact that Cassandra is decentralized means that there is no single point of failure.**



Token ownership and distribution in a balanced Cassandra ring



Server Symmetry means that all nodes in a Cassandra cluster function exactly in a similar way.

Characteristics of Cassandra

Distributed and Decentralized

cct | College Dublin
Computing • IT • Business

- Cassandra's high availability is largely due to its decentralized design.
- In the world of RDBMS, master/slave replication is easy to grasp.
- The two main benefits of decentralization are that it is easier to use than master/slave and that it helps you to prevent outages.
- This means that scaling is simple and that setting up 50 nodes is not all that different from setting up one.
- In a master/slave setup, ***the master can become a single point of failure (SPOF) which can be avoided by considering multiple masters in the Cassandra environment.***
- Cassandra is distributed and decentralized, there is no single point of failure, which allows high availability.
- All of the replicas in Cassandra are identical, therefore failures of a node won't affect service.

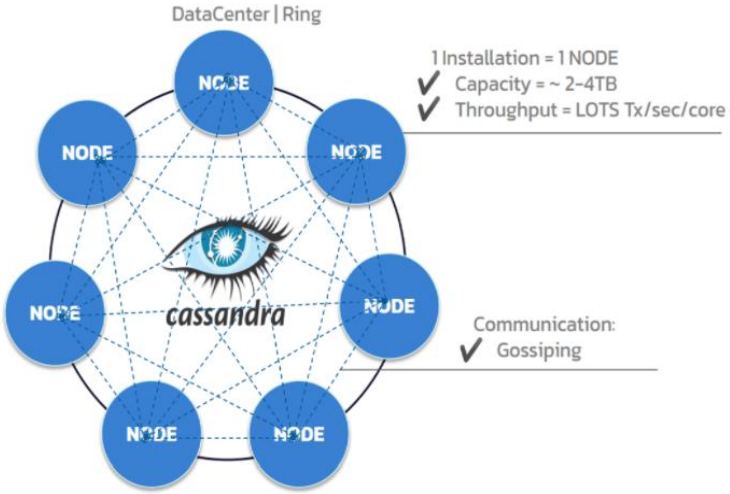
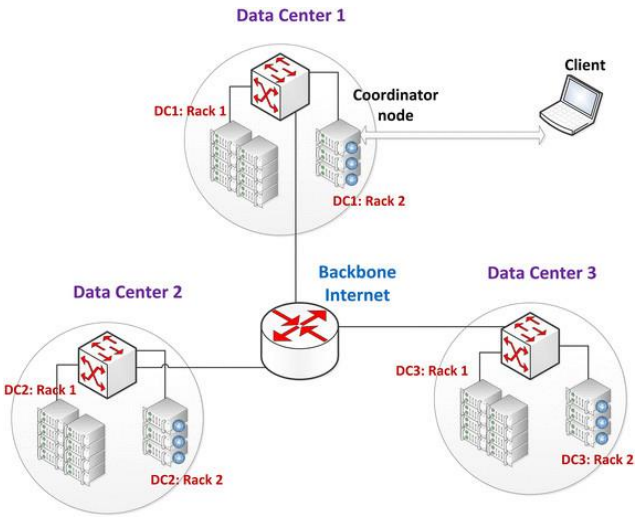


Figure 2. Apache Cassandra structure.



Characteristics of Cassandra

Elastic Scalability

College Dublin
Computing • IT • Business

- **Scalability** is an architectural feature of a system that can continue serving a greater number of requests with little degradation in performance. Apache Cassandra is highly scalable software.

- **Elastic scalability** refers to a special property of horizontal scalability. It means that your cluster can scale up and scale back down.

https://docs.datastax.com/en/cassandra-oss/3.x/cassandra/cassandraAbout.html

- When you scale down, your cluster's processing power is reduced. For business purposes, you might do this to accommodate seasonal demands in the travel or retail industries.

- It allows an organization to expand its hardware to accommodate an industry's growing data demands.

- With Cassandra, capacity may be readily increased by simply bringing new nodes online. This is known as linear scalability.

Characteristics of Cassandra

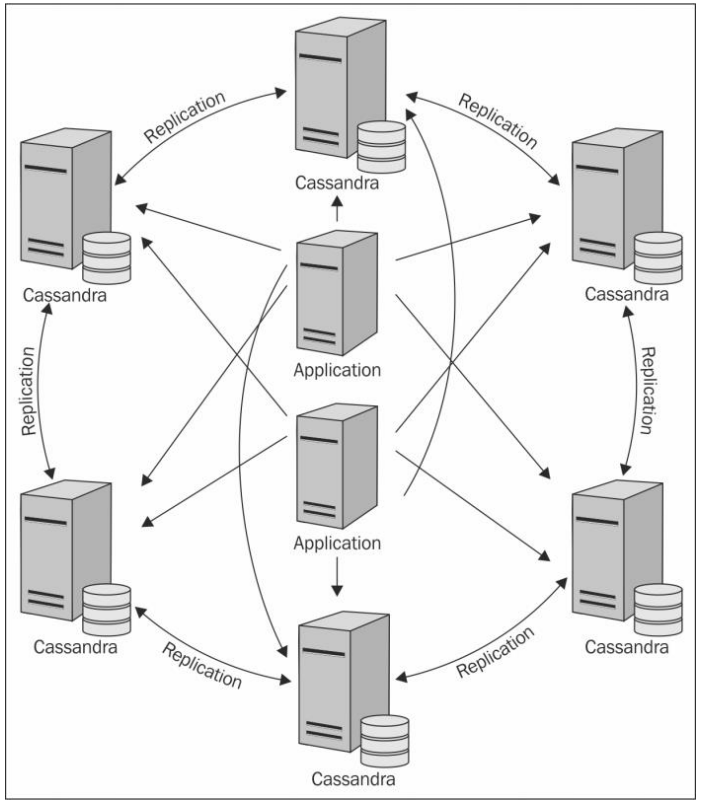
High Availability and Fault Tolerance

- A system's availability is determined by its capacity to handle requests including to handle different types of computer failures, including corrupting software, network outages, and hardware component failures.

- Cassandra is really accessible. Without any downtime, you may replace broken nodes in the cluster.

- We can replicate data across different data centres to increase local performance and guard against downtime in the event that a single data center is damaged by a disaster like a fire or flood.

- One of Cassandra's finest advantages is the automatic replication of the data across several nodes. Cassandra fault tolerance activates the product's ability to duplicate. Furthermore, because they quickly replace the lost nodes, many data centres support replication.

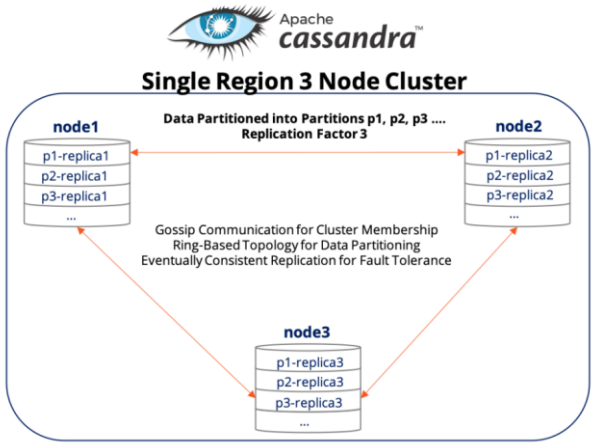


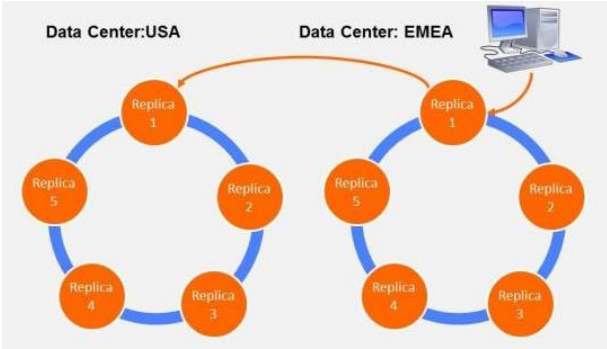
Characteristics of Cassandra

Tuneable Consistency

cct | College Dublin
Computing • IT • Business

- Consistency means that a read always returns most recently written value.
- Imagine that on an e-commerce site, two users are attempting to add the same last item to their shopping baskets. The last item in stock should be added to your cart and the other customer should be informed immediately of no longer availability of the item.
- This is guaranteed to happen when the state of a write is consistent among all nodes that have that data.
- Scaling data stores means making certain trade-offs **between data consistency, node availability, and partition tolerance**.
- Cassandra is termed as "**tuneably consistent**," which means it allows you to easily decide the level of consistency you require, in balance with the level of availability.



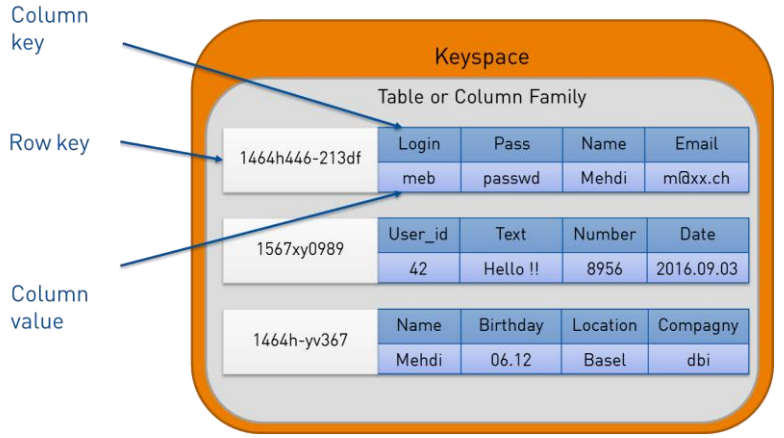


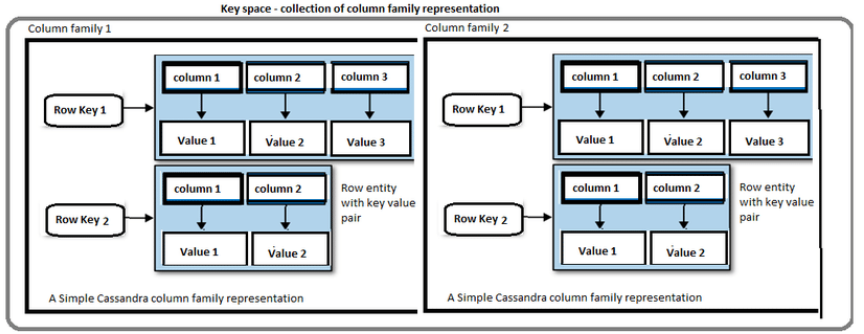
Characteristics of Cassandra

Row-Oriented

cct | College Dublin
Computing • IT • Business

- Cassandra's data model can be described as a partitioned row store, in which data is stored in sparse multidimensional hashtables.
- "Sparse" means that for any given row you can have one or more columns, but each row doesn't need to have all the same columns as other rows like it (as in a relational model).
- "Partitioned" means that each row has a unique key which makes its data accessible, and the keys are used to distribute the rows across multiple data stores.
- A **column-oriented database** is one in which the data is stored by columns, as opposed to relational databases, which store data in rows.



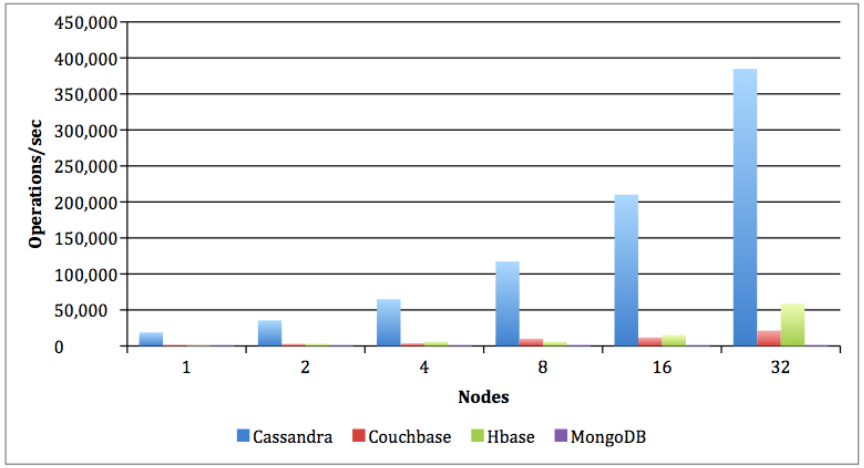


Characteristics of Cassandra

High Performance

cct College Dublin
Computing • IT • Business

- Cassandra was designed from the ground up to take full advantage of multiprocessor/ multi-core machines, and to run across many dozens of these machines housed in multiple data centres.
- Cassandra scales smoothly and consistently to many terabytes.
- Cassandra has been shown to perform exceptionally well under heavy load.
- **It consistently can show very fast throughput for writes per second on basic commodity computers, whether physical hardware or virtual machines.**
- You can preserve all of Cassandra's advantageous characteristics without compromising performance as you can add more servers.



Nodes	Cassandra	Couchbase	HBase	MongoDB
1	18,925.59	1,554.14	973.85	1,278.81
2	35,539.69	2,985.28	3,430.59	1,441.32
4	64,911.39	3,755.28	6,451.95	1,801.06
8	117,237.91	10,138.80	6,262.63	2,195.92
16	210,237.90	11,761.31	15,268.93	1,230.96
32	384,682.44	21,375.02	58,463.15	2,335.14

https://www.isaacbigdata.com/performance-cassandra/

IS CASSANDRA "SCHEMA-FREE"?

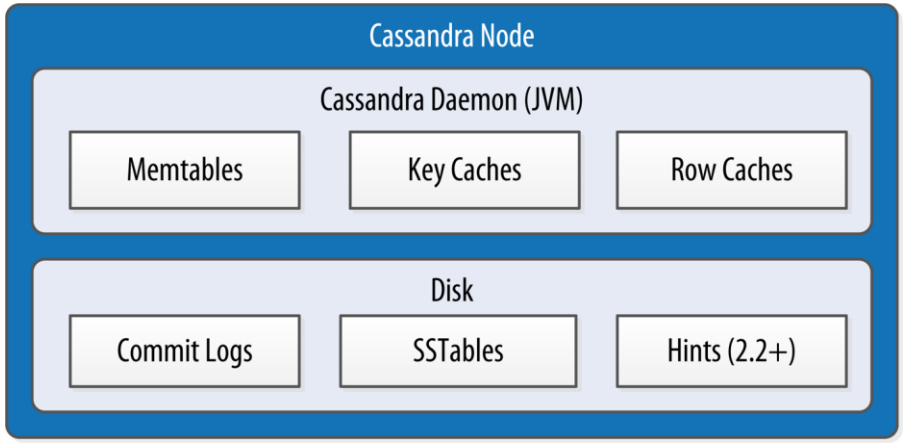
cct | College Dublin
Computing • IT • Business

- In its early versions, Cassandra was faithful to the original Bigtable whitepaper in supporting a "schema-free" data model in which new columns can be defined dynamically.
- **The major drawback of schema-free databases is the difficulty in determining the meaning and format of data, which limits the ability to perform complex queries.**
- Due to these drawbacks, many startups found it difficult to adopt the system, especially when they grew into larger enterprises involving multiple developers and administrators.
- The solution for those users was the introduction of the **Cassandra Query Language (CQL)**, which provides a way to define schema via a syntax similar to the **Structured Query Language (SQL)**.
- Initially, **CQL** was provided as another interface to Cassandra alongside the schema-free interface based on the Apache Thrift project. **During this transitional phase, the term "Schema-optional" was used to describe that data models could be defined by schema using CQL.**

Components of Cassandra

cct College Dublin
Computing • IT • Business

- **Node:** A Cassandra node is a place where data is stored.
- **Data center:** Data center is a collection of related nodes.
- **Cluster:** A cluster is a component which contains one or more data centres.
- **Commit log:** In Cassandra, the commit log is a crash-recovery mechanism. Every write operation is written to the commit log.
- **Mem-table:** A mem-table is a memory-resident data structure. After commit log, the data will be written to the mem-table. Sometimes, for a single-column family, there will be multiple mem-tables.
- **SSTable:** It is a disk file to which the data is flushed from the mem-table when its contents reach a threshold value.



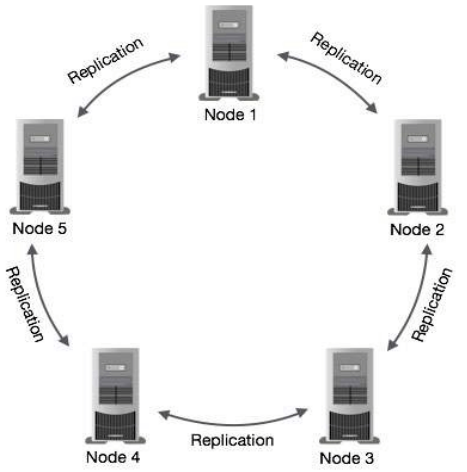
Internal data structures and files of a Cassandra node

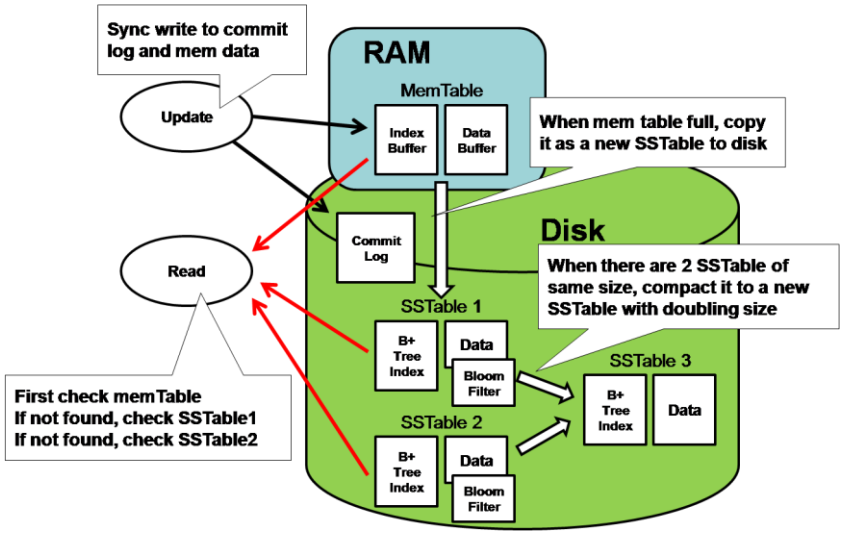
- Cassandra stores data both in memory and on disk to provide both high performance and durability.
- We focus on Cassandra's storage engine and its use of constructs called **memtables**, **SSTables**, and **commit logs** to support the writing and reading of data from tables.

Cassandra Architecture

cct | College Dublin
Computing • IT • Business

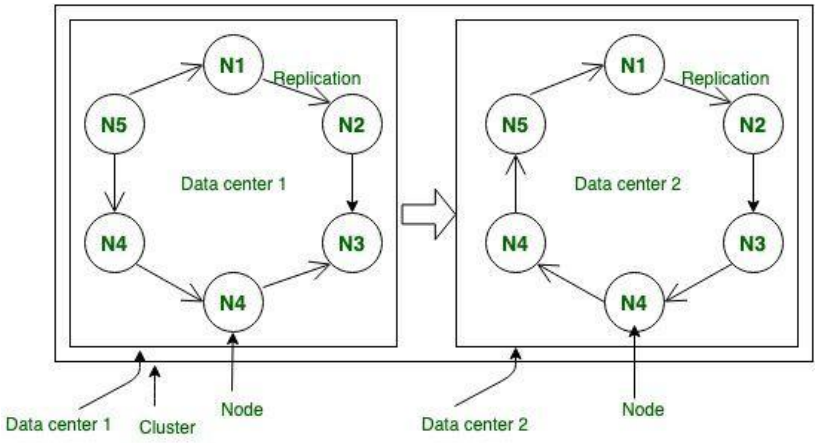
- Cassandra was designed to handle big data workloads across multiple nodes without a single point of failure.
- It has a peer-to-peer distributed system across its nodes, and the data is distributed among all the nodes in a cluster.
- In Cassandra, each node is independent and at the same time interconnected to other nodes. All the nodes in a cluster play the same role.
- Every node in a cluster can accept read and write requests, regardless of where the data is located in the cluster.
- In the case of failure of one node, Read/ Write requests can be served from other nodes in the network.

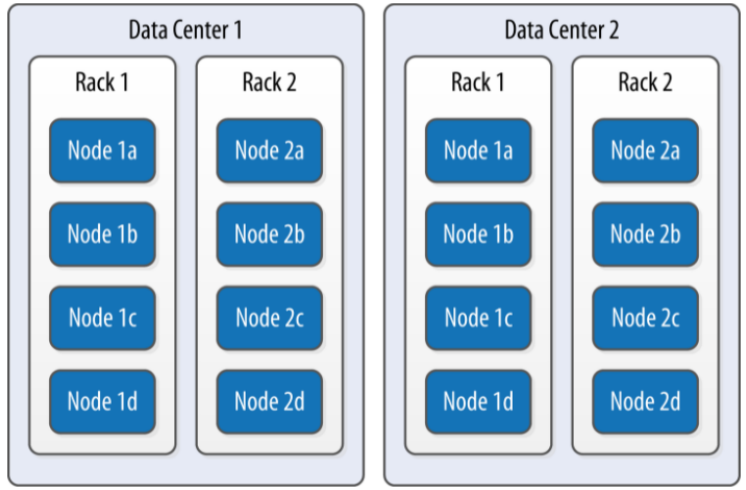




Data Replication in Cassandra

- Nodes in a Cassandra cluster serve as replicas for a certain piece of data.
- Cassandra will give the client the most recent value if some of the nodes respond with an out-of-date value.
- Cassandra updates the outdated values in the background after returning the most recent value.
- The illustrations make it obvious how Cassandra uses data replication among cluster nodes to prevent a single point of failure.



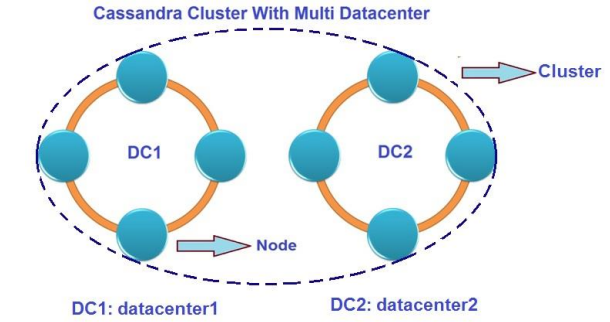


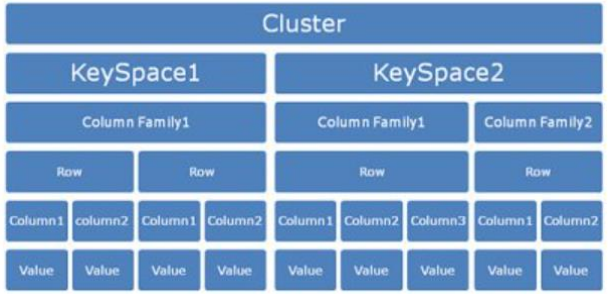
Topology of a sample cluster with data centers, racks, and nodes

Cassandra Data Model

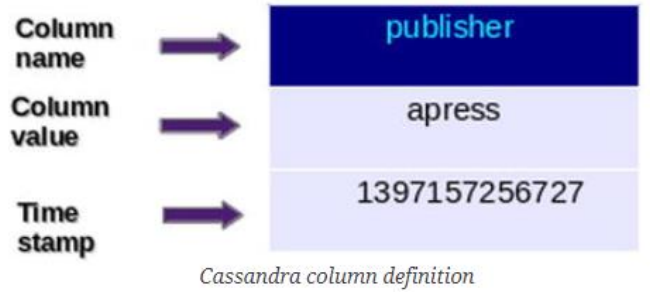
cct College Dublin
Computing • IT • Business

- **Cluster:** Cassandra database is distributed over several machines that are operated together. Cassandra arranges the nodes in a cluster, in a ring format, and assigns data to them.
- **Keyspace:** Keyspace is the outermost container for data in Cassandra.
- **Column families:** Column families are placed under keyspace. A keyspace is a container for a list of one or more column families while a column family is a container of a collection of rows. Each row contains ordered columns. Column families represent the structure of your data.
- **Replication factor:** It specifies the number of machines in the cluster that will receive copies of the same data.



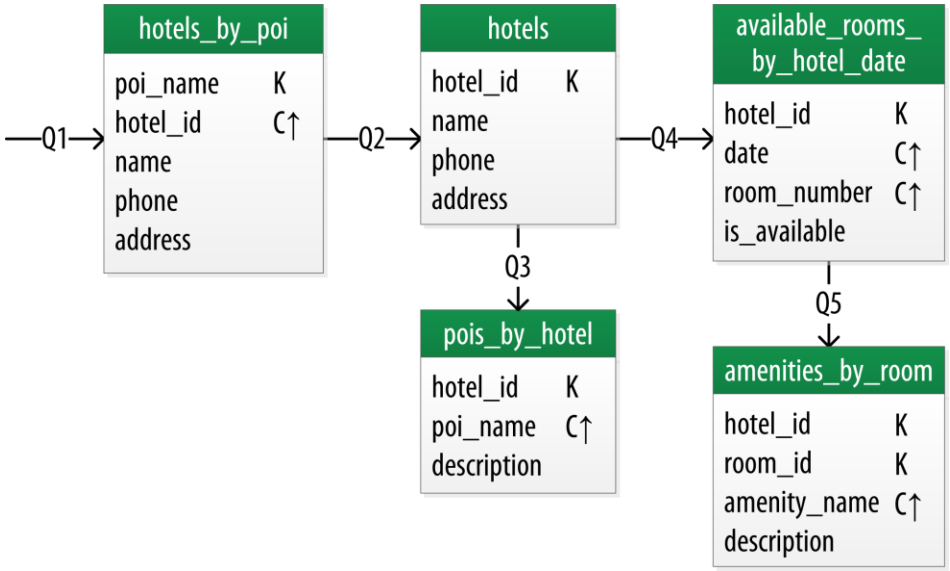


Cassandra Data Models Rules



Cassandra column definition

- **Cassandra** doesn't support JOINS, GROUP BY, OR clause, aggregation etc. You have to make sure the data is stored in a way that it can be accessed at any time.
- **Cassandra** is optimized for high **write** performances so you should maximize your writes for better read performance and data availability. There is a trade-off between data write and data read. So, optimize your data **read** performance by maximizing the number of data writes.
- Maximize data duplication because **Cassandra** is a distributed database and data duplication provides instant availability without a single point of failure.

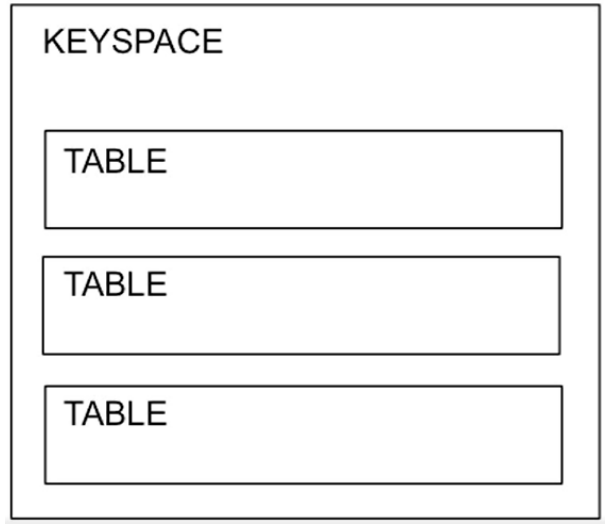


- **Q1.** Find hotels near a given point of interest.
- **Q2.** Find information about a given hotel, such as its name and location.
- **Q3.** Find points of interest near a given hotel.
- **Q4.** Find an available room in a given date range.
- **Q5.** Find the rate and amenities for a room.

Cassandra CQLSH

Internal Type	CQL Name	Description
BytesType	blob	Arbitrary hexadecimal bytes (no validation)
AsciiType	ascii	US-ASCII character string
UTF8Type	text, varchar	UTF-8 encoded string
IntegerType	varint	Arbitrary-precision integer
LongType	int, bigint	8-byte long
UUIDType	uuid	Type 1 or type 4 UUID
DateType	timestamp	Date plus time, encoded as 8 bytes since epoch
BooleanType	boolean	true or false
FloatType	float	4-byte floating point
DoubleType	double	8-byte floating point
DecimalType	decimal	Variable-precision decimal
CounterColumnType	counter	Distributed counter value (8-byte long)

- Cassandra **CQLSH** stands for Cassandra CQL shell.
- **CQLSH** specifies how to use Cassandra commands.
- After installation, Cassandra provides a prompt Cassandra query language shell (**cqlsh**). It facilitates users to communicate with it.
- Cassandra commands are executed on **CQLSH**. It looks like this:



A keyspace can be defined through the `CREATE KEYSPACE` command

```
CREATE KEYSPACE vehicle_tracker
WITH REPLICATION = { 'class' : 'NetworkTopologyStrategy',
'dc1' : 3, 'dc2' : 2};
```

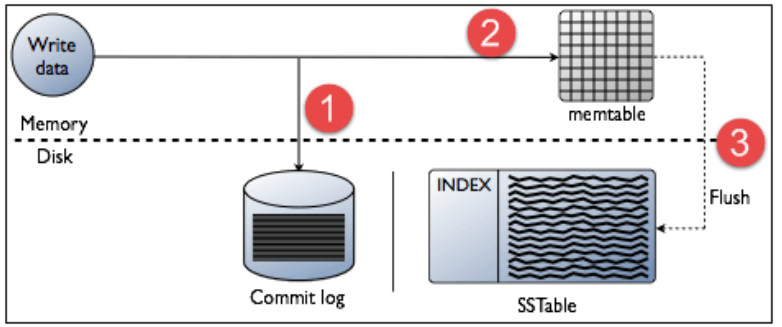
```
CREATE KEYSPACE vehicle_tracker
WITH REPLICATION = { 'class' : 'SimpleStrategy',
'replication_factor' : 1};
```

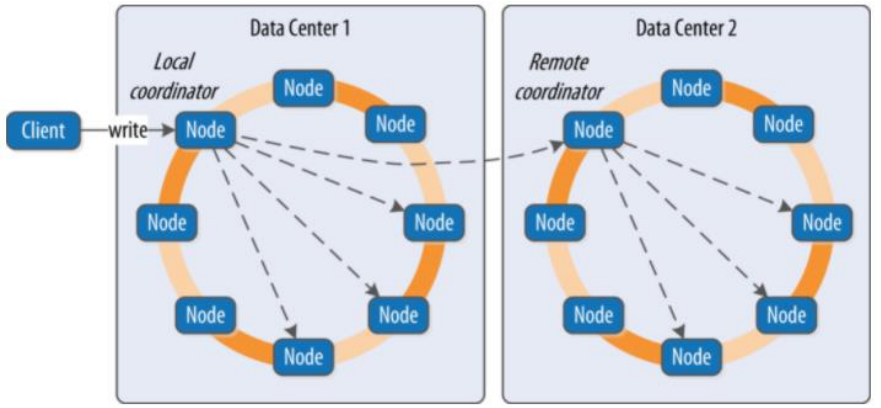
Cassandra

Write Operations

College Dublin
Computing • IT • Business

- Every write activity of nodes is captured by the **commit logs** written in the nodes.
- Later the data will be captured and stored in the **mem-table**.
- Whenever the **mem-table** is full, the data will be written into the **SStable** data file.
- All writes are automatically partitioned and replicated throughout the cluster.
- Cassandra periodically consolidates the **SSTables**, discarding unnecessary data.





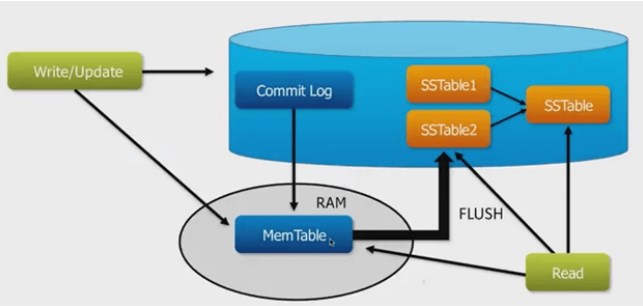
Interactions between nodes on the write path

Cassandra

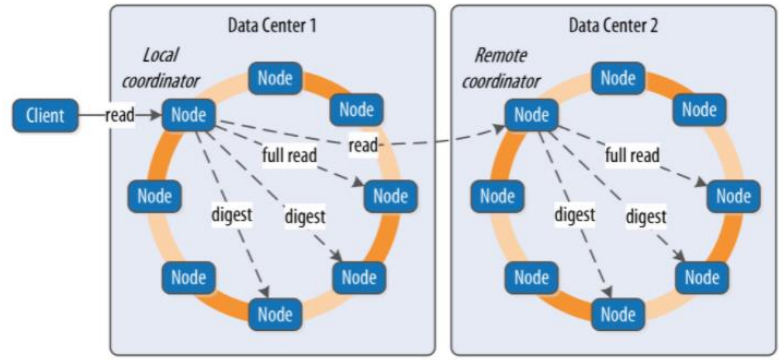
Read Operations

College Dublin
Computing • IT • Business

- In Read operations, Cassandra gets values from the **mem-table** and checks the **bloom filter** to find the appropriate **SSTable** which contains the required data. There are three types of read request that is sent to replicas by coordinators.
  - Direct request
  - Digest request
  - Read repair request



- The coordinator sends direct request to one of the replicas. After that, the coordinator sends the digest request to the number of replicas specified by the consistency level and checks if the returned data is an updated data.
- The coordinator then issues digest requests to all of the surviving copies. A background read repair request will update the data if any node provides an out-of-date value. This process is called read repair mechanism.



Interactions between nodes on the read path

Resources/ References

- Cassandra: The Definitive Guide, 2nd Edition, Jeff Carpenter, Eben Hewitt, O'Reilly Media, Inc., July 2016.
- Learning Apache Cassandra - Second Edition, Sandeep Yarabarla, Packt Publishing, April 2017.
- Data modeling with Cassandra, Jeff Carpenter, Eben Hewitt, O'Reilly Media, Inc., December 2016.
- https://www.javatpoint.com/cassandra-create-table
- Some images are used from Google search repository (https://www.google.ie/search) to enhance the level of learning.