Session 18:

- **Introduction to Cassendra**

  Apache Cassandra is an open-source, distributed database that stores data for applications that need fast read and write performance. It's a NoSQL database with a hybrid design between a tabular and key-value store. Cassandra is popular with developers and large companies.

  Cassandra is made up of a cluster of peer-to-peer network nodes, where each node is equally important. The partitioning system stores, retrieves, and chooses where to store copies of data. For example, a node can own data for that range, and one piece of data can be replicated to multiple replica nodes. This ensures reliability and fault tolerance.

  Cassandra can be used to store user profile information for online video games, device metadata for internet of things (IoT) applications, or records for events.

  Cassandra was created at Facebook and later released as an open-source project in July 2008.

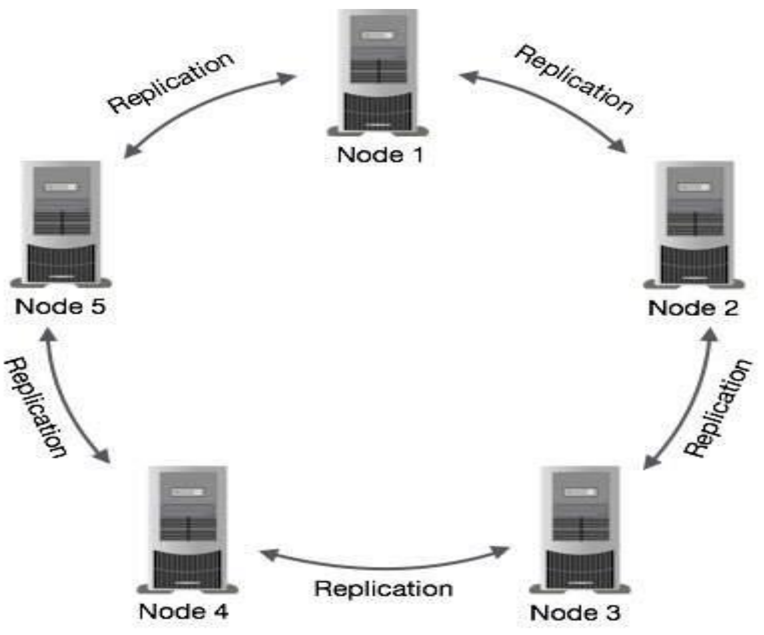
Features of Cassandra

- **Elastic scalability** − Cassandra is highly scalable; it allows to add more hardware to accommodate more customers and more data as per requirement.
- **Always on architecture** − Cassandra has no single point of failure and it is continuously available for business-critical applications that cannot afford a failure.
- **Fast linear-scale performance** − Cassandra is linearly scalable, i.e., it increases your throughput as you increase the number of nodes in the cluster. Therefore, it maintains a quick response time.
- **Flexible data storage** − Cassandra accommodates all possible data formats including: structured, semi-structured, and unstructured. It can dynamically accommodate changes to your data structures according to your need.
- **Easy data distribution** − Cassandra provides the flexibility to distribute data where you need by replicating data across multiple data centers.
- **Transaction support** − Cassandra supports properties like Atomicity, Consistency, Isolation, and Durability (ACID).

- **Fast writes** − Cassandra was designed to run on cheap commodity hardware. It performs blazingly fast writes and can store hundreds of terabytes of data, without sacrificing the read efficiency.

Data Replication in Cassandra

In Cassandra, one or more of the nodes in a cluster act as replicas for a given piece of data. If it is detected that some of the nodes responded with an out-of-date value, Cassandra will return the most recent value to the client.



Components of Cassandra: The key components of Cassandra are as follows

- **Node** − It is the place where data is stored.
- **Data center** − It is a collection of related nodes.
- **Cluster** − A cluster is a component that contains one or more data centres.
- **Commit log** − The commit log is a crash-recovery mechanism in Cassandra. 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.

- **Bloom filter** – These are nothing but quick, nondeterministic, algorithms for testing whether an element is a member of a set. It is a special kind of cache. Bloom filters are accessed after every query.

- **Comparison between Cassendra and MongoDB**

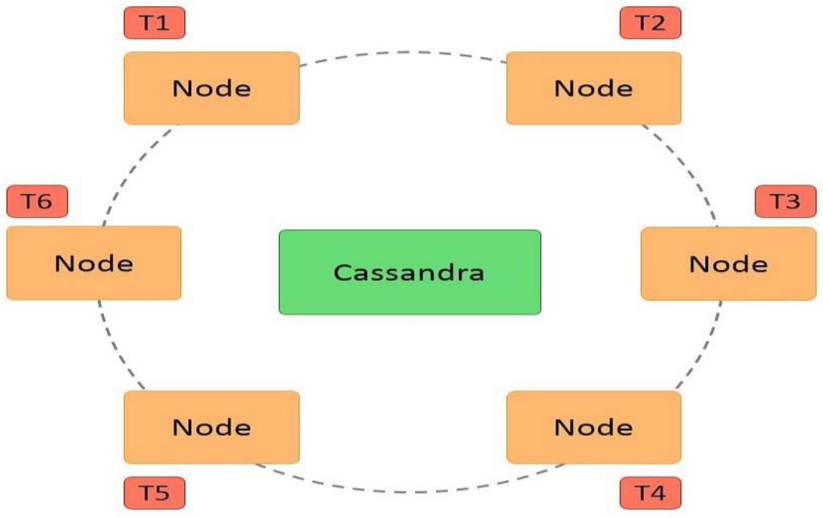
	Cassandra	MongoDB
Data Model	Cassandra stores data in a wide column tabular store. Each row contains a unique identifier as its primary key and a following set of columns. The primary key serves as a partition hash key as data is distributed across the cluster.	MongoDB stores documents in an optimized BSON format. Documents are grouped in databases and collections and returned as JSON documents with support for a large number of data types.
Indexing	Cassandra offers standard built-in indexing as well as basic secondary indexes to index additional columns to allow queries filtering. Maintaining a large secondary index can potentially cause scalability issues as the cluster grows.	MongoDB supports many index types for various use cases. Secondary indexes on any field are available and supported in different types: Compound, Text, Geo, TTL, Partial, Wildcard and Compound Wildcard indexes.
Query Language	Cassandra uses a proprietary language called CQL, which is similar to SQL. Applications can use different drivers provided mostly by third parties and a shell.	MongoDB has a rich query language called MQL. It supports a wide variety of modern native drivers as well as a shell. The data can be edited, deleted, inserted, and queried in many shapes and forms.
Transactions	Cassandra does not support transactions.	MongoDB supports fully ACID compliant transactions.

Concurrency	Cassandra isolates on a row-level. This means that a specific row for a specific partition can be operated simultaneously by one client.	MongoDB allows multiple database users to concurrently access the same data by managing a well-defined concurrency control. MongoDB uses document-level locking, so writes to a single document occur either in full or not at all, and clients always see consistent data. Together with those mechanisms, MongoDB supports different read and write concerns for distributed clusters and retryable reads and writes.
High Availability and Scalability	Cassandra replicates its keyspace across the cluster nodes based on the keyspace replication factor. Since it's a multi-master ring, each node is holding a range of the partition key per table, while also hosting parts of other nodes' replicas. Each node coordinates reads to the correct node to retrieve or operate the data while it also needs to repair data that got out of consistency across the nodes. The only way to replace the default hash partitioning is by changing the partitioner component for the entire cluster.	MongoDB was built from the ground up to support distribution of data using replication and sharding mechanisms. Replica sets host an identical copy of the data and elect a primary which receives all the writes, while other nodes are secondaries replicating all the data. Sharding allows you to easily scale your collections across multiple replica sets. With geo-zone sharding, you can also easily manage data sovereignty requirements. The ability to define specific shard keys and reshard collections with zero downtime when a shard key is no longer optimal gives your application a huge advantage when managing massively distributed datasets at scale. The shard key advisor commands will help you refine your shard keys.
Security	Apache Cassandra supports the following basic security	MongoDB supports enterprise-grade security mechanisms to secure your

	methods: • TLS/SSL support for client connections • User authentication • User authorizations with roles	MongoDB deployments. Most of them are on by default in MongoDB Atlas cloud offering: • Authentication and authorization using built-in SCRAM or certificates • TLS/SSL, x509, Queryable Encryption and Client-Side Field Level Encryption • Server-Side storage engine encryption • LDAP and Kerberos integrations Additionally, MongoDB cloud offerings have strong security compliance certifications. Read more on our trust center.
Mobile Support	There is no specific Apache Cassandra version or tools for mobile development. There is also no specific mobile oriented driver or SDK.	Atlas for the Edge streamlines data management between the database and edge devices. Within this solution, you will find two key components for mobile development: Atlas Device SDKs and Atlas Device Sync Atlas Device SDKs, available for most popular languages, frameworks and platforms, offer a lightweight reactive on-device object-store for mobile/edge/IoT devices. Atlas Device Sync offers offline-first synchronization between MongoDB Atlas clusters and the on-device database with automatic conflict resolution and strong eventual consistency. Changesets may arrive any time that connectivity allows.

Cloud Offerings	Provided via third-party services on different clouds and platforms. Compatibility tests to the Apache Cassandra version need to be verified with each vendor.	MongoDB Atlas, the *database-as-a-service* platform, offers clusters in all three major cloud providers, starting from a free tier to a fully blown production cross-region and cross-cloud cluster. Together with Atlas, you get the advantages of using Atlas App Services application services, Charts, and Data Federation for querying your data lake alongside Atlas clusters, as well as Atlas Search for optimized full text search.
Documentation & University	Apache Cassandra offers documentation on their main website, including a dedicated community. There are no official university or online courses on the Cassandra website.	Detailed documentation with examples and full tutorials including a full community and developer hub websites.
Native Data Visualization Tooling	There are no native data visualization tools provided by the Cassandra project.	MongoDB Charts provides a quick, simple, and powerful way to perform data visualization with MongoDB Atlas data. Additionally, MongoDB Compass provides a GUI client for MongoDB. MongoDB also provides a connector to integrate with popular third-party business intelligence tools.

- **Architecture of Cassandra**
  - Cassandra is designed such that it has no master or slave nodes.
  - It has a ring-type architecture, that is, its nodes are logically distributed like a ring.
  - Data is automatically distributed across all the nodes.
  - Similar to HDFS, data is replicated across the nodes for redundancy.
  - Data is kept in memory and lazily written to the disk.
  - Hash values of the keys are used to distribute the data among nodes in the cluster.



Cassandra architecture

Difference between Cassandra and MongoDB:

S.NO.	**Cassandra**	**MongoDB**
1.	Developed by Apache Software foundation and released on July 2008.	Developed by MongoDB Inc. and initially released on 11 February 2009.
2.	Cassandra is written only in Java language.	MongoDB is written in C++, Go, JavaScript, Python languages.
3.	Writing scalability in Cassandra is very high and efficient.	Writing scalability is limited in MongoDB
4.	Read performance is highly efficient in Cassandra as it takes O(1) time.	Read performance is not that fast in MongoDB when compared to Cassandra.

S.NO.	Cassandra	MongoDB
5.	Cassandra has only cursory support for secondary indexes i.e secondary indexing is restricted.	MongoDB does supports the concept of secondary indexes.
6.	Cassandra only supports JSON data format.	MongoDB supports both JSON and BSON data formats.
7.	The replication method that Cassandra supports is Selectable Replication Factor.	The replication method that MongoDB supports is Master Slave Replication
8.	Cassandra does not provides ACID transactions but can be tuned to support ACID properties.	MongoDB provides Multi-document ACID transactions with snapshot isolation.
9.	Server operating systems for Cassandra are BSD, Linux, OS X, Windows.	Server operating systems for MongoDB are Solaris, Linux, OS X, Windows.
10.	Famous companies like Hulu, Instagram, Intuit, Netflix, Reddit, etc uses Cassandra.	Famous companies like Adobe, Amadeus, Lyft, ViaVarejo, Craftbase, etc uses MongoDB.