MongoDB

ITC 6000 - Database Management Systems
Final Project

College of Professional Studies

Northeastern University - Vancouver

REPRESENTATIVES

Smit Parmar

Abhilash Dikshit

Shamim Sherafati

Nastaran Zamanian

Abstract

Software developers have started to take NoSQL data storage solutions into consideration in light of the high needs of Big Data. The performance of a NoSQL database in terms of speed of data access and processing, particularly response times to the most crucial CRUD activities, is one of the key factors to consider when choosing a NoSQL database for an application (CREATE, READ, UPDATE, DELETE). The behaviour of the most popular document-based NoSQL database, MongoDB, was examined in this study in terms of the difficulty and efficiency of CRUD operations, particularly query operations. The primary goal of the study is to conduct a comparative examination of the effects that each unique database has on the efficiency of the application when processing CRUD queries. The document-based MongoDB database, which is designed to simulate and streamline the activity of service providers who use a lot of data, was analysed and studied for this project. We shall begin with the introduction of NoSQL and MongoDB and then go into a detailed study about MongoDB.

Introduction

Since its release, MongoDB has steadily and securely increased in popularity, making it the most widely used type of NoSQL database. It is a cross-platform, open-source NoSQL database that maintains documents in the JSON format and is document-based (built in C++). Its flexible structure, which can change often during development, and improvements to each version enable automatic scalability with excellent performance and availability. Since MySQL only started offering a non-relational database option in 2018 with version 8.0, which has some parallels to MongoDB's model approach but also some distinctions, the document-based MySQL is still not that well-known.

How Does MongoDB work?

The MongoDB environment essentially gives users access to a server that they can launch and use to host several databases using MongoDB. The data is kept in collections and documents because of its NoSQL database. As a result, as illustrated below, the database, collection, and documents are related to one another.

- Similar to how a MySQL database has tables, a MongoDB database has collections. Multiple databases and collections could be made by users.
- The information we wish to save in the MongoDB database is contained in documents that are present within the collections. Multiple papers can be found in a single collection.
- The fields that are key-value pairs in the documents are used to generate them, just as columns in a relational database. Any BSON data type, including double, string, boolean, and others, may be used as the value for a field.
- The information is kept in MongoDB as BSON documents. Binary representation of JSON documents is known as BSON. The JSON data is transformed into BSON, a binary format that is more effectively stored and queried by the MongoDB server.

- In addition, MongoDB enables users to store nested data, which enables users to establish complicated relationships between data and store them in the same document, making dealing with and retrieving data much more efficient than with SQL. To obtain the data from tables 1 and 2 in SQL, users must create intricate joins. The BSON document can be up to 16MB in size.

Key Features

1. High Performance and Speed - Due to its features like scalability, indexing, replication, etc., the performance of MongoDB becomes very high as also data persistence, as compared to any other databases.
2. Sharding - MongoDB's sharding feature offers significantly more horizontal scalability. Because of horizontal scaling, each shard in each cluster basically serves as a separate database by housing a subset of the relevant dataset. A single, complete database created by combining the data from the scattered shards is considerably more suited to meeting the demands of a popular, expanding application with minimal downtime.
3. Replication - By replicating the data and distributing it across several servers, MongoDB makes sure that it is always accessible, even in the event of a server failure.
4. Aggregation - Performing actions on the grouped data in MongoDB also enables you to obtain a single result or a computed result. It offers three different aggregations, including a map-reduce function, an aggregation pipeline, and special-purpose aggregation techniques.
5. Scalability - Sharding in MongoDB enables horizontal scaling. Sharding is the process of dividing data among several servers. A substantial amount of data is divided into parts using the Shard Key, and these data chunks are then evenly dispersed across Shards that span numerous Physical Servers.

Data Modelling in MongoDB

MongoDB provides two types of data models: — Embedded data model and Normalised data model. Based on the requirement, you can use either of the models while preparing your document.

1) Embedded Data Model - In this model, you can have (embed) all the related data in a single document, it is also known as de-normalized data model.
2) Normalised Data Model - In this model, you can refer to the sub documents in the original document, using references. For example, you can rewrite the above document in the normalised model as:

MongoDB Query Language

Relational Databases are queried using Structured Query Language (SQL), MongoDB can be queried using MongoDB Query Language (MQL). MQL supports CRUD Operations.

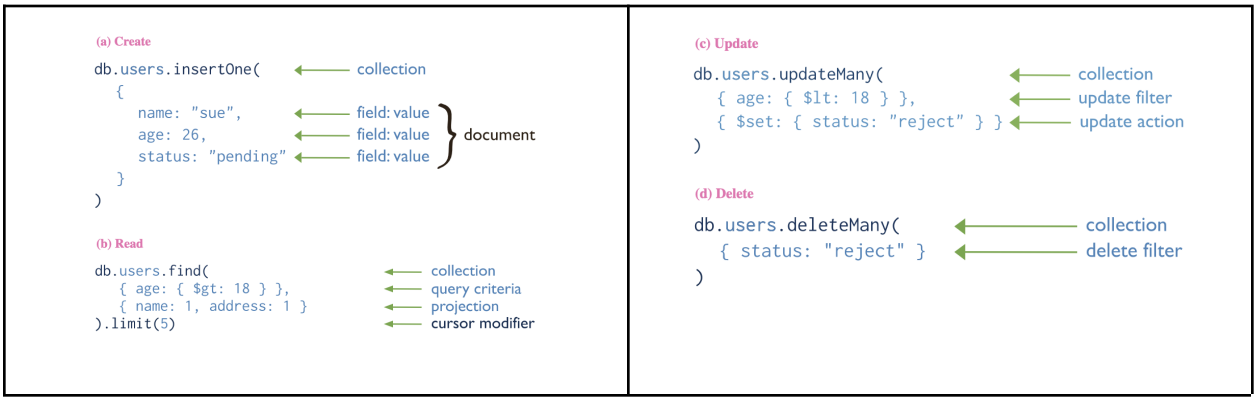
- Results can be sorted, grouped, filtered, and counted via aggregation pipelines.
- Special operations such as text search and geospatial queries are possible.

- Multi-document transactions are supported.

CRUD Commands in MongoDB

Database operations are conveniently referred to as *CRUD (Create, Read, Update, Delete)*. MongoDB supports CRUD with the following methods of Collection class:

- **Create**: Add documents to a collection. Methods include insertOne()and insertMany(). If the collection doesn't exist, it will be created. Unique _id field is automatically added to each document if not specified in the method calls.
- **Read**: Retrieve documents from a collection. Main method is find(), called with query criteria (how to match documents) and projection (what fields to retrieve).
- **Update**: Modify existing documents of a collection. Methods include updateOne(), updateMany() and replaceOne(). Method calls include query criteria and what to update.
- **Delete**: Remove documents from a collection. Methods include deleteOne() (first matching document is deleted) and deleteMany(), called with query criteria.



Advantages and Disadvantages of MongoDB

Advantages

1. **High speed/performance** - Higher speed than a traditional relational database.
2. **Support ad-hoc queries -** MongoDB supports ad-hoc queries and has worked to optimise the way this type of query is handled. It allows for real-time ad-hoc query updates, for instance.
3. **It's flexible -** MongoDB is a NoSQL database system, it's suitable for both structured and unstructured data.

Disadvantages

1. **Joins are awkward -** MongoDB doesn't support joins
2. **High reliance on good indexes -** If your indexing is incorrectly implemented, it negatively impacts the speed and performance of MongoDB. This means that you will need to spend time fixing and analysing the indexes you use.
3. **Duplicate data is a common problem -** Duplicate data can prove an issue. It can make it hard to manage your data efficiently.

Cloud Services associated with MongoDB

The Best MongoDB Storage and Hosting Services are as follows: MongoDB Atlas, Kamatera, Scale Grid, Clever Cloud, IBM Cloud, ObjectRocket, Scaling

Conclusion

In this project, we developed an understanding of NoSQL Databases and how they work and when to use them instead of SQL Databases. We also put emphasis on one of the leading NoSQL databases named MongoDB and contributed towards an in-depth study of the following NoSQL database. We started with the question of Why and When to use MongoDB and then also understood about the Data Modelling and Query Languages in MongoDB. Lastly, we portrayed the CRUD operations in MongoDB and what are their advantages and disadvantages.

Bibliography

[1] Taylor, D. (2022). *What is MongoDB? Introduction, Architecture, Features & Example*. https://www.guru99.com/what-is-mongodb.html

[2] M, A. (2021, October 16). *MongoDB Query Language*. Devopedia. https://devopedia.org/mongodb-query-language

[3] BasuMallick, C. (2022, August 17). *What Is MongoDB? Working, Architecture, Features, and Use Cases*. Devopedia. https://www.spiceworks.com/tech/cloud/articles/what-is-mongodb/

[4] Mullins, C. S., Vaughan, J., & Beal, B. (2021, April 8). *NoSQL (Not Only SQL database)*. Data Management. https://www.techtarget.com/searchdatamanagement/definition/NoSQL-Not-Only-SQL