

A Review on Various Aspects of MongoDB Databases

Anjali Chauhan

M.tech Scholar, CSE Department,
Rawal Institute of Engineering and Technology,
Faridabad, Haryana, India

Abstract—MongoDB is most popular among the NoSQL databases. For building data warehouses, it is a great tool especially because of its ability to fully utilize so called “sharding-nothing cluster architecture.” It is an open source database, which makes it ideal for building high performance data warehouses. In this paper, a review of various aspects of MongoDB is discussed and some key issues are framed. In future research can be done on any of these issues. So this paper opens some areas for research in MongoDB databases.

Keywords — No-SQL, MongoDB, Database, RDBMS, Non-relational databases

I. INTRODUCTION

MongoDB is an open-source document database that provides high performance, high availability, and automatic scaling. A record in MongoDB is a document, which is a data structure composed of field and value pairs. MongoDB documents are similar to JSON objects. The values of fields may include other documents, arrays and arrays of documents. The advantages of using documents are:

- Documents (i.e. objects) correspond to native data types in many programming languages.
- Embedded documents and arrays reduce need for expensive joins.
- Dynamic schema supports fluent polymorphism.

A. Key Features of MongoDB

- High Performance – MongoDB provides high performance data persistence. In particular, it supports for embedded data models reduces I/O activity on database system, indexes support faster queries and can include keys from embedded documents and arrays.
- Rich Query Language - MongoDB supports a rich query language to support read and write operations (CRUD) as well as Data aggregation, Text Search.
- High Availability – MongoDB’s replication facility, called replica set, provides automatic failover and data redundancy. A replica set is a group of MongoDB servers that maintain the same data set, providing redundancy and increasing data availability.
- Horizontal Scalability – MongoDB provides horizontal scalability as part of its core functionality. Sharding distributes data across a cluster of machines.

B. Overview of MongoDB

MongoDB is an open-source document database and leading NoSQL database. MongoDB is written in C++. MongoDB is a cross-platform, document oriented database

that provides high performance, high availability and easy scalability. MongoDB works on concept of collection and document.

- Database - Database is a physical container for collections. Each database gets its own set of files on the file system.
- Collection - Collection is a group of MongoDB documents. It is the equivalent of a RDBMS table. Collections do not enforce a schema. Documents within a collection can have different fields. Typically, all documents in a collection are of similar or related purpose.
- Document - A document is a set of key- value pairs. Documents have dynamic schema. Dynamic schema means that documents in the same collection do not need to have the same set of fields or structure, and common fields in a collection’s documents may hold different types of data.

```
{
  _id:ObjectId(7df78ad8902c)
  Title:'MongoDB Overview'
  Description:' MongoDB is no sql database
  Comments: [
    { user:'user1',
      Message:'my first comment',
      dateCreated: new Date(2011,1,20,2,15),
    },
    { user:'user2',
      Message: 'my second comments'
      dateCreated:    new    Date    (2011,1,25,7,45),
    }
  ]
}
```

- Sample Document - Above example shows the document structure of a blog site, which is simply a comma separated key value pair.

C. Advantages of MongoDB

Any relational database has a typical schema design that shows number of tables and the relationship between these tables. While in MongoDB, there is no concept of relationship. Advantages of MongoDB over RDBMS can be described as:

- MongoDB is a document database in which one collection holds different documents. Number of fields, content and size of the document can differ from one document to another.
- Structure of a single object is clear.
- No complex joins.

- Deep query ability. MongoDB supports dynamic queries on documents using a document-based query language that's nearly as powerful as SQL.
- Ease of scale out. MongoDB is easy to scale.
- Conversion/mapping of application objects to database objects not needed.
- Uses internal memory for storing the (windowed) working set, enabling faster access of data.

D. Uses of MongoDB

MongoDB has document oriented storage; data is stored in the form of JSON style documents. It can be indexed on any attributes. We can also explain where to use MongoDB:

- Big Data
- Content Management and Delivery
- Mobile and Social Infrastructure
- User Data Management
- Data Hub

E. Importance of MongoDB

MongoDB is a document-oriented database. This is “as opposed to” other types of DBs: Relational, Graph, Key/Value, Queue, FTS, Map/Reduce, etc. The leads to lessons like: data organization relative to query patterns, indexing options, handling polymorphic objects in code, performing manual joins on the client.

MongoDB is also a DB that highlights the use of multiple servers in two ways: Replica Sets and Sharding

Replica Sets:

- Redundancy and failover
- Zero downtime for upgrades and maintenance
- Master-slave replication
- Strong consistency
- Delayed consistency
- Geospatial features

Sharding:

- Distributes a single logical database system across a cluster of machines
- Uses range-based partitioning to distribute documents based on a specific shard key
- Automatically balances the data associated with each shard
- Can be turned on and off per collection(table)

This leads to lessons around things like “write safety”, handling master fail-over, shard keys and shard balancing. MongoDB also provides a simple framework for performing map/reduce or aggregation operation across multiple computers. This leads to lessons around projection of objects and basic aggregation primitives.

II. COMPARATIVE STUDY

As shown in Table 1, in MongoDB, some MySQL terms, such as table or row, get another name, namely collection, respectively BSON document. In other words, we can say that MongoDB contains collections, collections contain documents and a document contains multiple fields.

In the classical RDBMS model, the data is organized in the form of relations and is represented in a table consisting of rows and columns. Relational databases employ the usage of a parameter known as key. There are several types of keys

available albeit primary key is one of the most important key of the table; it is used to identify each row of the table uniquely. There are four main operations used to access the database they are known as CRUD namely, Create, Read, Update and Delete associated with the data. These operations use the Structured Query Language –SQL. ACID properties are one of the most significant and important attributes of a SQL database. This is the key difference between SQL and NoSQL database systems. The NewSQL approach on the other hand, conserves and supports the properties of relational model, at the same time incorporating the features of NoSQL model.

TABLE I. MYSQL VS MONGODB TERMS

MySQL	MongoDB
Database	Database
Table	Collection
Index	Index
Row	BSON document
Column	BSON field
Join	Embedded documents and linking
Primary key	Primary key
Group by	Aggregation

Unlike MySQL, where the database is presented graphically in the form of a table, in MongoDB, a database has the following graphic structure:

```
{
  "_id": "d4acaf3a76e4378b853eb15fde21672",
  "username": "andra",
  "email": andra@gmail.com,
}
{
  "_id": "d4rvgf3a76e4378b853eb15fde21672",
  "username": "iona",
  "email": iona@gmail.com,
}
```

The example above shows a database for users, each user having an id that is unique and automatically generated, a username and an email address.

The application will have 3 classes of users, namely the administrators, the moderators and the regular users. Each user has the right to create a private forum/subforum. Within a subforum, the moderators have the right to edit/delete the subforum and they can also moderate other users’ discussions, while regular users are only allowed to post discussions and leave comments. If a relational database has been used, the columns for forums and subforums should have appeared at all forum users, although normal users will never have the right to create, modify or delete them, unless of course, they are the administrators of that particular forum. Using MongoDB, these fields regarding the forum and subforum will appear only to users who have that right (moderators and administrators), thus significantly reducing storage space, which is much higher using MySQL.

As in relational databases, MongoDB also has one-to-many relationships, but in this case the concept of foreign key is not used; instead, the concept of annotations is used. Thus,

in this case, regarding a forum, the connection between the forum and its subforums is as follows: in the forum document, the subforums are referenced using the annotation.

MongoDB provided lower execution times than MySQL in all four basic operations (Insert, Select (query), Update, Delete), which is essential when an application should provide support to thousands of users simultaneously.

III. ISSUES WITH MONGODB

MongoDB is a popular option for database storage. It's easy to learn and faster than competing RDBMs, but still there are some potential pitfalls in it. It's denormalised, meaning data is stored in a nested document structure rather than relational tables. This makes for faster lookups as Mongo doesn't rely on expensive join operations seen with MySQL and other database engines. Despite such strengths, there are several problems with MongoDB that one should consider using it as a database engine.

- **Problems with Reliability** – MongoDB writes are asynchronous by default. The main advantage of this is you don't have to wait for confirmations for every insert or update operation before the next one starts. This makes updates faster but less reliable. Even if some of the updates are unsuccessful, the write operation will still partially succeed. With this engine, it's all or nothing. While this may be slower, it's a more consistent and reliable way to perform writes operations. When things only partially work you can end up with data inconsistencies and buggy data.
- **Problems with Schema-less Design** – Since MongoDB is denormalised, it doesn't adhere to a relational schema. Everything is stored in nested JSON objects called documents. While this allows for greater flexibility with your data models, it forces more schema based design decisions on the app logic than the db. Without the schema in place, the rules and regulations of your data models are dictated by your app logic rather than the db itself.

IV. CONCLUSION AND FUTURE SCOPE

MongoDB is currently the most popular Document-oriented DB, but it is hardly the most robust or performant implication. MongoDB is a relative newcomer in the database arena, and is the most popular among the NoSQL databases. It is a great tool for building data warehouses, especially because of its ability to fully utilize so called "shared-nothing cluster architecture." It is an open-source database, which makes it ideal for building high performance data warehouses. It is also well documented, well supported, and easy to install, integrate into PHP, and test. Also, because it is so new, updated versions are released practically every day, so one has to approach the project for which MongoDB is considered with a sense of adventure. The next generation NonSQL (NoSQL) databases are mostly non-relational, distributed and horizontally scalable and are able to satisfy most of the needs of the present day applications. The main characteristics of these databases are schema-free, no join, nonrelational, easy replication support, simple API and eventually consistent.

The result of this study open new avenues for future research of performance of data access when there are hotspots in data because it supposes all the data will be accessed in same patterns. A future scope of this work would be the implementation of the third model in MongoDB, directly or indirectly.

REFERENCES

- [1] Benymol Jose, Sajimon Abraham, "Exploring the Merits of NoSQL: A Study Based on MongoDB", 978-1-5090-6590-5/17/\$31.00 ©2017 IEEE
- [2] YunhuaGu, ShuShen, Jin Wang, Jeong-UkKim, "Application of NoSQL Database MongoDB", 978-1-4799-8745-0/15/\$31.00©2015 IEEE
- [3] Cornelia Gyrödi, Robert Gyrödi, George Pecherle, AndradaOlah, "A Comparative Study: MongoDB vs. MySQL", 978-1-4799-7650-8/15/\$31.00 ©2015 IEEE
- [4] <https://www.stackchief.com/search/mongodb>
- [5] <https://docs.mongodb.com/manual/introduction/>
- [6] <https://www.stackchief.com/blog/Problems%20with%20MongoDB>
- [7] <https://www.thegeekstuff.com/2014/01/sql-vs-nosql-db/>