

# MongoDB

Ayswarya Babu<sup>1</sup>

<sup>1</sup>Computer Science  
Vimala College  
Thrissur, India

Bhavana R Warriar<sup>2</sup>

<sup>2</sup>Computer Science  
Vimala College  
Thrissur, India

**Abstract:-** MongoDB is the latest and the fastest growing document oriented database which is widely used in Big data applications. Big data is a widely discussed terminology now, has a large scope in dealing with large volume of exponentially increasing data of all the types-structured, semi structured, unstructured. MongoDB being considered as an effective tool has a prominent place in dealing with the issues and challenges of Big data. This paper discusses about MongoDB's features, strengths and limitations.

**Keywords—** RDBMS, Map-Reduce, GridFS, NoSQL, SQLGROUP BY, SQL

## I. INTRODUCTION

MongoDB (from humongous) is a cross-platform document oriented database. Mongo DB has rapidly grown to become a popular database for web applications and is a perfect fit for Node. Released under a combination of the GNU Affero General Public License and the Apache License, MongoDB is free and open-source software. First developed by the software company 10gen (now MongoDB Inc.) in October 2007 as a component of a planned platform as a service product, the company shifted to an open source development model in 2009, with 10gen offering commercial support and other services. Since then, MongoDB has been adopted as backend software by a number of major websites and services, including eBay, FourSquare, Viacom, Craigslist, Source Forge and The New York Times. As of July 2015, MongoDB is the fourth most popular type of database management system, and the most popular for document stores.

## II. WHAT IS MONGODB???

MongoDB is a document oriented database that provides high performance, high availability, and easy scalability and an open source. It is a document database as documents (objects) map nicely to programming language data types. Embedded documents and arrays reduce the need for joins. Dynamic schema makes polymorphism easier. Embedding makes reading and writing fast. Indexes can include keys from embedded documents and arrays. Optional streaming writes. This makes it a high performing database. MongoDB is highly available, i.e.; Replicated servers with automatic master failover makes this happen. As automatic sharding distributes collection data across machines, eventually-consistent reads can be distributed over replicated servers. Thus MongoDB is highly scalable [1]. With MongoDB Management Service

(MMS), it supports a complete backup solution and full deployment monitoring, which ensures the capability of MongoDB in providing advanced operations.

And it is also a rich document based queries for easy readability. Its full index support for high performance and replication and failover for high availability, auto sharding for easy scalability and map / reduce for aggregation. SQL was invented in the 70's to store data. As everyone works with objects now-a-days, we need databases to persist our objects. There comes the role of MongoDB. MongoDB's initial release was in 2009. Before the invention of NoSQL, almost all databases were structural. This means that developers had to define the structure of the database before using it. Later, the NoSQL concept and all the technologies related to it were invented to rescue programmers. Instead of relational databases they use NoSQL. Now there are many different opinions about the benefits of relational over non relational databases. The major reasons are more flexible data model and a dynamic schema, scalability and better efficiency and performance. MongoDB is the leading NoSQL database, with stunning implementation, and it has a vibrant community.

## III. KEY FEATURES

### A. Document-oriented

Instead of taking a business subject and breaking it up into multiple relational structures, MongoDB can store the business subject in the minimal number of documents. For example, instead of storing title and author information in two distinct relational structures, title, author, and other title-related information can all be stored in a single document called Book, which is much more intuitive and usually easier to work with.

### B. Ad hoc queries

MongoDB supports search by field, range queries, regular expression searches [2]. Queries can return specific fields of documents and also include user-defined JavaScript functions.

### C. Indexing

Any field in a MongoDB document can be indexed (indices in MongoDB are conceptually similar to those in RDBMS)[3]. Secondary indices are also available.

#### D. Replication

MongoDB provides high availability with replica sets. A replica set consists of two or more copies of the data [1]. Each replica set member may act in the role of primary or secondary replica at any time. The primary replica performs all writes and reads by default. Secondary replicas maintain a copy of the data on the primary using built-in replication. When a primary replica fails, the replica set automatically conducts an election process to determine which secondary should become the primary. Secondary replicas can also perform read operations, but the data is eventually consistent by default.

#### E. Load balancing

MongoDB scales horizontally using sharding. The user chooses a shard key, which determines how the data in a collection will be distributed. The data is split into ranges (based on the shard key) and distributed across multiple shards. (A shard is a master with one or more slaves.) MongoDB can run over multiple servers, balancing the load and/or duplicating data to keep the system up and running in case of hardware failure. Automatic configuration is easy to deploy, and new machines can be added to a running database.

#### F. File storage

MongoDB can be used as a file system, taking advantage of load balancing and data replication features over multiple machines for storing files [1].

This function, called GridFS, is included with MongoDB drivers and available with no difficulty for development languages (see "Language Support" for a list of supported languages). MongoDB exposes functions for file manipulation and content to developers. GridFS is used, for example, in plugins for NGINX and lighttpd. Instead of storing a file in a single document, GridFS divides a file into parts, or chunks, and stores each of those chunks as a separate document. In a multi-machine MongoDB system, files can be distributed and copied multiple times between machines transparently, thus effectively creating a load-balanced and fault-tolerant system.

#### G. Aggregation

Map reduce can be used for batch processing of data and aggregation operations. The aggregation framework enables users to obtain the kind of results for which the SQLGROUP BY clause is used.

#### H. Server-side JavaScript execution

JavaScript can be used in queries, aggregation functions (such as Map Reduce), and sent directly to the database to be executed.

#### I. Capped collections

MongoDB supports fixed-size collections called capped collections. This type of collection maintains insertion order and, once the specified size has been reached, behaves like a circular queue.

And another remarkable feature of MongoDB is schema less data i.e.; developers are able to store any data model or

change the schema during or after inserting data. MongoDB has official drivers for a variety of popular programming languages and development environments. There are also a large number of unofficial or community-supported drivers for other programming languages and frameworks.

#### IV. COMPARISON BETWEEN RDBMS AND MONGODB

RDBMS	MONGODB
Database	Database
Table	Collection
Tuple/Row	Document
Column	Field
Table Join	Embedded documents
Primary Key	Primary Key (Default key_id)[5]

#### V. ADVANTAGES OF MONGODB OVER RDBMS

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. So MongoDB is schema less. Its structure of a single object is clear. It has no complex joins. MongoDB supports dynamic queries on documents using a document-based query language that's nearly as powerful as SQL. MongoDB is easy to scale[4]. In MongoDB, conversion / mapping of application objects to the database objects is not needed. It uses internal memory for storing the (windowed) working set, enabling faster access of data. Whereas RDBMS doesn't have the features such as scalability, flexibility and performance and it need a DBA. If you don't have a DBA and normalization of data is not needed, you should consider MongoDB. We use MongoDB in the case of big data, content management and delivery, Mobile and Social Infrastructure, User Data Management and Data Hub etc.

#### VI. LIMITATIONS

If something crashes while it's updating 'table-contents' – you lose all of your data. Repair takes a lot of time, but usually ends up in 50%-90% data loss if you aren't lucky. So, only way to be fully secure is to have 2 replicas in different data centers.

Indexes take up a lot of RAM. They are B-tree indexes and if you have many, you can run out of system resources really fast. Data size in MongoDB is typically higher as each document has field names stored in it. Less flexibility with more complex querying (e.g. no joins) makes MongoDB not supportable for highly transactional applications; but certain atomic operations are supported, at a single document level. At the moment Map/Reduce (e.g. to do aggregations/data analysis) is OK, but not blisteringly fast. So if that's required, something like Hadoop may need to be added into the mix.

## VII. CONCLUSION

MongoDB has the best features of storing key/values of document databases and relational databases in one. Before MongoDB; ORACLE, PostgreSQL, MySQL were used to deal with voluminous data in a systematic manner. In MongoDB maximum document size is only 16 MB and document nesting level can go only up to 100[6]. Still the MongoDB is a highly sought after. In most cases the performance of MongoDB is good enough but if you want to process lot of data with complex queries, it's better to use another database. However the data insertion, updating and deletion performance is very good. This makes MongoDB a good database for projects with simple data access. One good sample scenario is logging; which only uses simple inserts and since MongoDB is schema less the logged information can easily be extended. In MongoDB there is no concept of relationship and fast In-place Updates available in MongoDB gives it a professional support. So MongoDB is one of the emerging databases in this world. MongoDB is a great NoSQL database that can be configured quickly and used in any of internet based applications dealing with huge amount of data. For node applications, we can start up MongoDB quickly so that we can get to the fun part, building applications! So we can conclude that it will run well in the future combining the delegating model and functionality at which MongoDB is good at.

## REFERENCES

- [1] Afshin Mehrabani, "MongoDB High Availability", Packt Publishing Ltd.
- [2] <https://www.mongodb.org/>
- [3] Kristina Chodorow, Michael Dirolf, "MongoDB The Definitive Guide", O'Reilly Media Inc, First Edition.
- [4] <http://blog.iprofs.nl/2011/11/25/is-mongodb-a-good-alternative-to-rdbms-databases-like-oracle-and-mysql/>
- [5] <https://www.mongodb.com/mongodb-and-mysql-compared>
- [6] <http://tech.tulentsev.com/2014/02/limitations-of-mongodb/>