MongoDB

NANDITA SATHE[1,*]

[1] *School of Informatics and Computing, Bloomington, IN 47408, U.S.A.*
[*] *Corresponding author: nsathe@iu.edu*

paper-1, April 12, 2017

MongoDB is a NoSQL database. Instead of using tables and rows as in relational databases, MongoDB is built on an architecture of collections and documents. Documents comprise sets of key-value pairs and are the basic unit of data in MongoDB. Collections contain sets of documents and function as the equivalent of relational database tables. MongoDB database is used when data size is expected to be huge and schema is not stable.

Keywords: I524, MongoDB, NoSQL

https://github.com/cloudmesh/sp17-i524/raw/master/paper1/S17-IO-3017/report.pdf

1. INTRODUCTION

MongoDB is a non-RDBMS key-value data store. The data to be stored does not necessarily have to follow a fixed schema. In relational database data is stored primarily in tables. In NoSQL database like MongoDB data is stored in collections. A collection acts as a container for a 'Document'. A Document is equivalent to a row of a table. The data in MongoDB Document is stored in JSON array like data structure. The database also supports large volume of data storage and offers very high data insert speed due to its schema-less design [1].

2. INFRASTRUCTURE AND PERFORMANCE

Following sections discuss key deployment considerations.

2.1. Working set size

The set of data and indexes that are accessed most frequently during normal operations is called the working set. Working set resides in memory rather than on disk to ensure low latency database operations. Ideally working set should fit into RAM. Before MongoDB deployment it is necessary to assess the working set size based on the size of data and kind of operations that will be performed on data, so that adequate RAM size can be determined. Page faults occur in case working set size exceeds the available RAM. In this case either the RAM size should be increased or sharding should be done (see Section 2.4).

2.2. Storage and Disk I/O

If working set size is far larger than any available memory, then selecting the proper disk type for deployment is important. Local disks should be used as far as possible as the network storage can cause high latency and poor performance.

2.3. CPU Selection

MongoDB performance is typically not CPU-bound. As MongoDB rarely encounters workloads and is able to leverage large numbers of cores, it is preferable to have servers with faster clock speeds than numerous cores with slower clock speeds [2].

2.4. Sharding

MongoDB partitions data across servers using a technique called Sharding. Balancing of data across shards is automatic, and shards can be added and removed without taking the database offline [3]. Sharding allows the database to scale out on hardware deployed on-premises or in the cloud, enabling almost unlimited growth with higher throughput and lower latency than relational databases [4]. Figure 1 shows scaling of database.

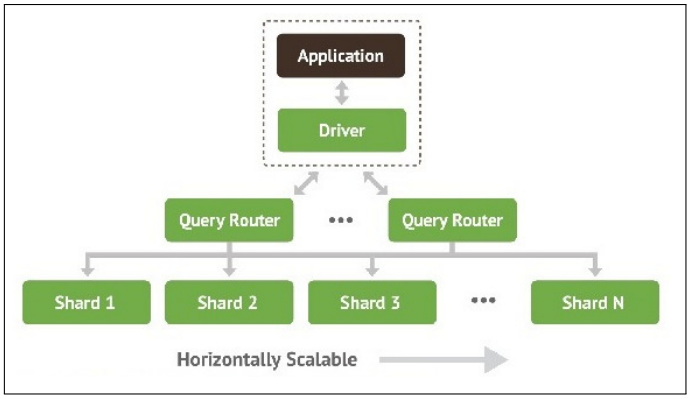

Fig. 1. MongoDB Auto-sharding [2].

Users should consider deploying a sharded MongoDB cluster in the following situations [2]:

RAM Limitation: If the working set size is expected to exceed the capacity of the maximum amount of RAM in the system.

Disk I/O Limitation: If the system has a large amount of write activity, and the operating system cannot write data fast enough to meet demand.

Storage Limitation: If the data set approaches or exceeds the storage capacity of a single node in the system.

2.5. Replication

MongoDB maintains multiple copies of data called replica sets using native replication. A replica set consists of multiple replicas. At any given time, one member acts as the primary replica set member and the other members act as secondary replica set members. Reads and writes are issued to a primary copy of the data. If the primary member fails for any reason (e.g., hardware failure, network partition), one of the secondary members is automatically elected to primary and begins to process all writes [5].

Figure 2 shows replica sets maintained by MongoDB.

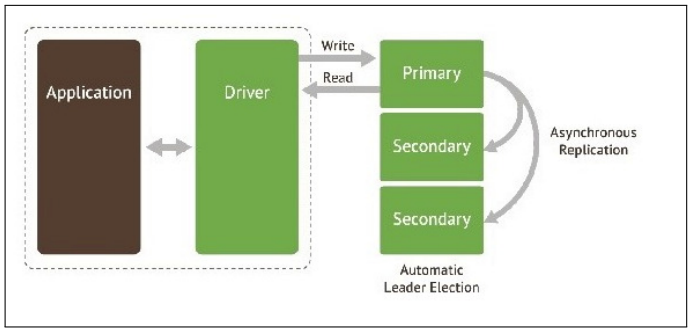

Fig. 2. MongoDB replica sets [2].

2.6. Indexing

Indexes are a crucial mechanism for optimizing system performance and scalability. MongoDB includes support for many types of secondary indexes that can be declared on any field in the document, including fields within arrays:

One can define compound, unique, array, partial, TTL, geospatial, sparse, hash and text indexes to optimize for multiple query patterns, multi-structured data types and constraints.

Index intersection enables MongoDB to use more than one index to optimize an ad-hoc query at run-time [4].

3. FEATURES

These are main features of MongoDB.[6]

- General purpose database, almost as fast as the key:value NoSQL type.
- Scalability: (from a standalone server to distributed architectures of huge clusters). This allows us to shard our database transparently across all our shards. This increases the performance of our data processing.
- Aggregation: batch data processing and aggregate calculations using native MongoDB operations.
- Load Balancing: automatic data movement across different shards for load balancing. The balancer decides when to migrate the data and the destination Shard, so they are evenly distributed among all servers in the cluster. Each shard stores the data for a selected range of our collection according to a partition key.
- Native Replication: syncing data across all the servers at the replica set.
- Security: authentication, authorization, etc.
- Advanced users management.
- Automatic failover: automatic election of a new primary when it has gone down.
- Zero downtime upgrades.
- There are no bottlenecks processing large volumes of data.
- MongoDB uses JSON objects to store and transmit information.
- We can do queries and geospatial operations in 2D and 3D.
- We can utilize Map-Reduce for information processing using JavaScript functions at the server side.
- JavaScript execution: Ability to store JavaScript functions on the server for queries and aggregation functions
- MongoDB Management Service. (MMS) is a powerful web tool that allows us tracking our databases and our machines and also backing up our data.
- Monitoring: MMS tracks the database and hardware metrics for managing MongoDB deployment. Performance is visualized in a rich web console to help optimize your deployment. Discover issues via custom alerts before MongoDB instance will be affected.
- Backup: Continuous backup with point-in-time recovery of replica sets and sharded clusters. Multiple copies of every backup are archived across data centers (geographically distributed and fault-tolerant)

4. MONGODB FOR BIGDATA ANALYTICS

At the top level there are 3 Vs that define BigData - Volume, Variety and Velocity. MongoDB supports storage of high volume of data which is complex in nature. Its dynamic schema provides a major advantage for businesses that need to ingest, store, and process rapidly evolving data streams from new sources. The term velocity refers to high and volatile inbound data, faster query at low latency. MongoDB is designed to support insertion of high volume of data in less time and faster response to query [7].

For data analysis one can utilise built-in aggregation functionality provided by MongoDB. In case of more complex data analytics one can leverage Hadoop and Apache Spark framework. To be able to connect efficiently to MongoDB Hadoop and Spark connectors are available and can be configured easily. In this scenario data is pulled from MongoDB and complex processing is performed in Hadoop/Spark, output of the processing can then be written back to MongoDB for later querying and ad-hoc analysis.

5. MONGODB USE CASES

Some of the common MongoDB use cases involve operational intelligence, Internet of Things, content management and real-time analytics. Following are some of the MongoDB customers [8].

BOSCH: The Bosch IoT Suite is a cloud-enabled software package for developing applications in the Internet of Things (IoT). Bosch has built its Internet of Things suite on MongoDB. MongoDB has been used to manage massive volume and unstructured nature of IoT data and perform real-time analytics.

Aadhar: India's Unique Identification project, Aadhar, maintains biometrics database. Aadhar is in the process of capturing demographic and biometric data of over 1.2 billion residents. Aadhar has used MongoDB as one of its databases to store this huge amount of data.

MetLife: MetLife uses MongoDB for its customer service application called 'The Wall'. It provides a consolidated view of MetLife customers, including policy details and transactions.

eBay: eBay has a number of projects running on MongoDB for search suggestions, metadata storage, cloud management and merchandizing categorization.

6. LICENSING

MongoDB database server and tools are available under GNU AGPL v3.0.

7. EDUCATIONAL MATERIAL

To get started on learning MongoDB, following resources can prove helpful.

- Helpful tutorials for beginner developers - https://www.tutorialspoint.com/mongodb/
- MongoDB's documents. You will find reference material for everything here - https://docs.mongodb.com/v3.2/#getting-started
- Fixed schedule course on MongoDB - https://university.mongodb.com/
- MongoDB course specifically for Data Scientists - https://www.udacity.com/course/data-wrangling-with-mongodb–ud032

8. COMPARISON

There are various alternatives for MongoDB which are good in their own arena, here we are going to compare 3 most widely used databases Cassandra, Hbase and MongoDB. Table 1 shows comparison amongst MongoDB, HBase and Cassandra [9].

The comparison shows that MondoDB supports the maximum operating systems. Secondary indexes makes it easy to query in MongoDB. MongoDB is a step ahead in case of ease of use, as it is easy to program in JSON.

Table 1. MongoDB vs. Cassandra Vs. HBase [9].

Name	Cassandra	HBase	MongoDB
Supported server operating systems	BSD, Linux, OS X, Windows	Linux, Unix, Windows	Linux, OS X, Solaris, Windows
Datatype support	yes	no	yes
Secondary indexes	restricted	no	yes
APIs and other access methods	CQL and an API based on Apache Thrift	Java API, RESTful HTTP API, Thrift	proprietary protocol using JSON
Server-side scripts	no	yes	JavaScript
Triggers	yes	yes	no
Partitioning methods	Sharding	Sharding	Sharding
Replication methods	selectable replication factor	selectable replication factor	Master-slave replication
MapReduce	Yes	Yes	Yes
Concurrency	Yes	Yes	Yes
In-memory capabilities	No	No	Yes
Privileges	Access rights for users can be defined per object	Access Control Lists (ACL)	Access rights for users and roles

REFERENCES

[1] MongoDB, Inc, "Mongodb: Bringing online big data to business intelligence & analytics," PDF on Web, Jun 2016. [Online]. Available: http://s3.amazonaws.com/info-mongodb-com/MongoDB_BI_Analytics.pdf

[2] Mat Keep, "Preparing for your first mongodb deployment: Capacity planning and monitoring," Web Page, Oct 2013. [Online]. Available: https://www.infoq.com/articles/mongodb-deployment-monitoring

[3] MongoDB, Inc, “Capacity planning and hardware provisioning for mongodb in ten minutes,” Web Page. [Online]. Available: https://www.mongodb.com/blog/post/capacity-planning-and-hardware-provisioning-mongodb-ten-minutes
[4] ——, “Mongodb architecture,” Web Page. [Online]. Available: https://www.mongodb.com/mongodb-architecture
[5] IBM Business Partner Network, “Mongodb architecture explained,” Web Page, Jun 2016. [Online]. Available: https://www.ibmbpnetwork.com/linux-blog/mongodb-architecture
[6] Cesar Trigo Esteban, “Mongodb: Characteristics and future,” Web Page, Aug 2014. [Online]. Available: http://www.mongodbspain.com/en/2014/08/17/mongodb-characteristics-future/
[7] MongoDB, Inc, “Real-time analytics,” Web Page. [Online]. Available: https://www.mongodb.com/use-cases/real-time-analytics
[8] Sudha Gopinath, “Real world use cases of mongodb,” Web Page, Jan 2014. [Online]. Available: https://www.edureka.co/blog/real-world-use-cases-of-mongodb
[9] solid IT, “System properties comparison cassandra vs. hbase vs. mongodb,” Web Page. [Online]. Available: http://db-engines.com/en/system/Cassandra%3BHBase%3BMongoDB