MongoDB – a comparison with NoSQL databases

Hema Krishnan, Research Scholar, CUSAT
M.Sudheep Elayidom, Associate Professor, School of Engineering, CUSAT
T.Santhanakrishnan, Scientist E, NPOL

Abstract— Web based applications and their data management needs have changed dynamically in the past few years. Variety of features and strict data consistency is provided by the relational databases. Due to massive cost of storing and manipulating data in classical relational database systems, NoSQL databases have been developed. NoSQL databases provide more scalability and heterogeneity when compared to RDBMS. MongoDB, a NoSQL database provides high scalabilty, performance and availability. MongoDB is a document based NoSQL database designed for Internet and web based applications. Data model of MongoDB is easy to build on due to its inherent support for unstructured data. MongoDB doesnot require costly and time consuming migrations when application requirements change. MongoDB's documents are encoded in a JSON like format called BSON. This paper describes the advantages of MongoDB when compared to other NoSQL databases and its applications in sentiment analysis.

Index Terms— MongoDB, NoSQL, Document, Sharding, Replication, Indexing.

1 INTRODUCTION

During the last decade, there was an enormous growth in the database sizes. This made monolithic database systems struggle to keep up with today's requirements. Products based on the well known RDBMS traits are available today. But applications those require data and or functional partitioning, either because of the sheer size of the data or for the purpose of load balancing have to rely on custom built relations or utilize alternative database systems. Since distributed partitioning implementation has become a real challenge with the high demand for multi-machine databases, a whole set of NoSQL (not only SQL) databases [1] have emerged to fill the gap of RDBMS.

Common features of NoSQL products are the divergence from the relational data model, simplification of transactional model and transaction processing and most importantly the shift to the imperative programming model from the declarative style SQL language. Relational databases are often being replaced by other viable alternatives, such as NoSQL [2] databases, for reasons of scalability and heterogeneity. MongoDB, a NoSQL database, is an agile database built for scalability, performance and high availability. It can be deployed in single server environment [3] and also on complex multi-site architectures. The application of MongoDB in twitter sentiment analysis is also a noticing one.

In this paper we compare MongoDB with other NoSQL databases based on different factors. Some of its applications in sentiment analysis are also listed.

2. DATA MODELS

Data models [8] are designed to be quite flexible in order to support the storage needs arising from applications dealing with highly heterogeneous data. Also, the wide-spread use of dynamically typed scripting languages has made less strictly structured background storage system favourable. While a highly generic data model looks reasonable from the aspect of the client, efficient server-side processing makes certain restrictions on the data model necessary. As a result, many NoSQL systems offer semi-structured models and list-like data types. A primary objective of NoSQL database systems is to evenly distribute data among shards.

2.2 No SQL Data Models

- Key-values Stores
  The main idea here is using a hash table where there is a unique key and a pointer to a particular item of data. The Key/value model is the simplest and easiest to implement. But it is inefficient when you are only interested in querying or updating part of a value, **amo**ng other disadvantages.
  Examples: Redis, Voldemort, Oracle BDB, Amazon SimpleDB, Riak

- Column Family Stores
  These were created to store and process very large amounts of data distributed over many machines. There are still keys but they point to multiple columns. The columns are arranged by column family.
  Examples: Cassandra, HBase

- Document Databases
  These were inspired by Lotus Notes and are similar to key-value stores. The model is basically versioned documents that are collections of other key-value collections. The semi-structured documents are stored in formats like JSON. Document databases are essentially

the next level of Key/value, allowing nested values associated with each key. Document databases support querying more efficiently.
Examples: CouchDB, MongoDb

- Graph Databases
Instead of tables of rows and columns and the rigid structure of SQL, a flexible graph model [5] is used which, again, can scale across multiple machines. NoSQL databases do not provide a high-level declarative query language like SQL to avoid overtime in processing. Rather, querying these databases is data-model specific. Many of the NoSQL platforms allow for RESTful interfaces to the data, while other offers query APIs.
Examples: Neo4J, InfoGrid

3 MONGODB

MongoDB [8] is a database management system designed for the Internet and web-based applications. It is a document based NoSQL database. The data model and persistence strategies are built for high read and write throughput and the ability to scale easily with automatic failover. MongoDB's document data model makes it easy to build on, since it has inherent support for unstructured data and does not require costly and time consuming migrations when application requirements change.

MongoDB's documents are encoded in a JSON-like format, called BSON. BSON is a natural fit for modern object oriented programming methodologies and is lightweight, fast and traversable. MongoDB uses BSON as network transfer format for documents. BSON at first seems BLOB-like, but there exists an important difference: the MongoDB database understands BSON internals. This means that MongoDB can reach inside BSON objects, even nested ones, using dot notation. This allows MongoDB to build indexes and match objects against query expressions on both top-level and nested BSON keys.

MongoDB also supports rich queries and full indexes. This distinguishes it from other document databases in which a separate server layer is used to handle complex queries. Its other features include automatic sharding, replication, and easy storage. The increasing popularity of MongoDB and the large amounts of user-related sensitive information stored in these databases raise the concern for the confidentiality and privacy of the data and the security provided by these systems. When MongoDB was initially designed, security was not a primary concern of its designers.

3.1 Components of MongoDB

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 [4] gets its own set of files on the file system. A single MongoDB server typically has multiple databases.

- Collection
Collection is a group of MongoDB documents. It is the equivalent of an RDBMS table. A collection exists within a single database. 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.

Table 1 shows the 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 provided by mongodb itself)
Database Server and Client	
Mysqld/Oracle	mongod
mysql/sqlplus	mongo

4 FEATURES OF MONGODB

- Document Oriented Storage :
Data is stored in the form of JSON style documents
Index on any attribute
- Replication & High Availability [5]
MongoDB achieves replication by the use of replica set. A replica set is a group of mongod instances that host the same data set. In a replica one node is primary node that receives all write operations. All other instances, secondaries, apply operations from the primary so that they have the same data set. Replica set can have only one primary node.Replica set is a group of two or more nodes (generally minimum 3 nodes are required). In a replica set one node is primary node and remaining nodes are secondary. All data replicates from primary to secondary node. At the time of automatic failover or maintenance, election establishes for primary and a new primary node is elected. After the recovery of failed node, it again join the replica set and works as a secondary node

- Auto-Sharding [6]
  Sharding is the process of storing data records across multiple machines and it is MongoDB's approach to meeting the demands of data growth. As the size of the data increases, a single machine may not be sufficient to store the data nor provide an acceptable read and write throughput. Sharding solves the problem with horizontal scaling. With sharding, we add more machines to support data growth and the demands of read and write operations.

- Rich Queries
- Fast In-Place Updates

5 CONCLUSION

NoSQL systems offer much less functionality than traditional relation database management systems [7], especially in transaction isolation and scan operations. But they can be successfully used when complex database logic is not, but large-scale, distributed operation is an objective.

MongoDB is an effective document oriented database which can be used for tweet analysis and other applications. JSON format of data stored in MongoDB helps in analysing the data easily for further processing.

In future we can analyse more number of features of MongoDB and compare with NoSQL.

REFERENCES

[1] Laszlo Dobos, Balazs Pinczel etal. Sneddon, "A Comparative evaluation of NoSQL systems". Annales Univ. Sci. Budapest., Sect. Comp. 42 (2014) 173-198.
[2] Peng Wang, Yan Qi, Hua-min Yang, "Analysis and study on the performance of query based on NOSQL "in Computer Modelling & New Technologies 2014 18(9) 153-159.
[3]Prabhakaran Murugesan, Indrakshi Ray "Audit Log Management in MongoDB", in 2014 IEEE 10th World Congress on Services.
[4] D.R.Merlin Shalini & Mr.S.Dhamodharan, "Performance and Scaling Comparison Study of RDBMS and NoSQL (MongoDB", in COMPUSOFT, An international journal of advanced computer technology, 3 (11), November-2014 (Volume-III, Issue-XI).
[5] Anju Abraham, "A Dynamic Query Form System for Mongodb", in SSRG International Journal of Computer Science and Engineering (SSRG-IJCSE) – volume1 issue9 Nov 2014.
[6] Sanobar Khan, Prof.Vanita Mane, "SQL Suport over MongoDB using Metadata", in International Journal of Scientific and Research Publications, Volume 3, Issue 10, October 2013
[7] Yong-Lak Choi, Woo-Seong Jeon , and Seok-Hwan Yoon, "Improving Database System Performance by Applying NoSQL" in J Inf Process Syst, Vol.10, No.3, pp.355~364, September 2014
[8] http://www.tutorialspoint.com/mongodb/