International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

MogoDB: A NoSQL Database with Amazing Advantages and Features

Jyothi J[a]

[a]*CSE department, PDIT, Hospet, India*

DOI:

A B S T R A C T

The extremely compressed data stores on non-relational database management systems, commonly referred to as NoSQL data stores, which facilitate the management of data for internet user programs, do not now provide this feature. Some of the words that come to mind are data security, encryption methods, MongoDB, and NoSQL. Without negatively compromising the database's performance in terms of speed or memory utilization, the appropriate encryption methods utilized to various essential data fields provide data protection. This is possible because the right encryption methods are being used. Databases are often used to manage different types of data since document-oriented and unstructured data are typically stored there. These databases include Cassandra, MongoDB, CouchDB, Redis, Hyper-table, and others. They must provide strong security and safeguard users' private data when it is in use or at rest because they are open source. There is a need for a single solution that may enhance data transmission security by offering an improved encryption technique that expedites operations and consumes less memory when managing databases.Today's IT growth is attributed to the development of more data-intensive applications. Due to stringent limitations on data structure, data relations, and other factors, relational databases do not allow us to work with huge data sets or maintain high volume databases. Unstructured data from many industries, including a variety of forms, must be processed and stored in databases. Thus, NoSQL types offer solutions to a number of problems associated with huge data bases. Largely because of benefits like flexibility and horizontal scalability, NOSQL (Not Only SQL) is in demand. This essay explores many NoSQL database types, such as MongoDB, CouchDB, HBase, Cassandra, etc.

Keywords:NoSQL database, Cassandra, Column Database, Big Data, Big Data Analytics, RDBMS, NoSQL, MongoDB, CouchDB, HBase

1. Introduction

In earlier technologies, relational databases were frequently utilized by numerous applications to gather data. The relational database had a basic structure that looked like a table with columns and records. The data in those tables can be modified, reviewed using well-defined, organized queries, and preserved in accordance with requirements [1]. Due to the advancement of technology, it is now necessary to have a huge database, support for several formats, ease of use, and flexibility. New technology is required to satisfy all of the aforementioned criteria as well as evolving needs. NoSQL[2] is a technology built on non-relational data that satisfies all of the aforementioned requirements. In information systems today, NoSQL databases have distinct popularity. A database designer must choose a NoSQL database when he or she is unable to determine whether to utilize a relational or non-relational database with the vast volume of data they have acquired in a variety of formats[2]. The purpose of this paper is to outline the key elements and traits of MongoDB technology, particularly for NoSQL as a replacement for relational databases. The structure of this essay is as follows: A basic introduction to both NoSQL databases and conventional relational databases is provided in the introduction. The part where some accomplished results from various researchers are mentioned continues in the following section. Some relevant works on this subject are also mentioned in this section. A thorough explanation of the many kinds of NoSQL databases, MongoDB's explanation, and the results of various queries are in the third section. The fourth segment concludes with a conclusion [3].

* *Corresponding author.* Tel.: +0-000-000-0000 ; fax: +0-000-000-0000.
E-mail address: shahida@pdit.ac.in

2. Review of Literature

Understanding SQL and NoSQL databases:

The relational data model (like SQL databases) offers specific mechanisms to handle data consistency within the system, to manage concurrency control making sure that accurate results are generated for concurrent operations, and finally to guarantee transaction ACID properties defined as follows [4]: (1) Atomic: Everything in a transaction must succeed or the entire transaction is rolled back. (2) Consistent: An inconsistent state cannot be present when a transaction leaves the database. (3) Isolated: Transactions are kept apart from one another. (4) Long-lasting: Completed transactions survive server restarts. By employing SQL and a unique attribute known as ACID, every user can perform one or more database table transaction operations [4]. Eric Brewer's CAP theorem defines such configuration as a requirement in a scale-up approach. There is a difference between ACID and BASE. Non-relational data models (like as NoSQL databases) are following BASE (Basic Availability, Soft-state, Eventual consistency property). In general, it is harder to develop software in the fault-tolerant BASE world compared to the particular ACID world. (2) Availability: The promise that each request will receive feedback on whether it was successful or unsuccessful. (3) System operation is unaffected by arbitrary segmentation as a result of recurring network outages [4]. NoSQL stands for "Not Only Sql," however this does not imply that this phrase opposes SQL; on the contrary, it may be adequate when compared to relational databases, which can collect data in the form of tables and store data that is structured as well [5]. Relational and non-relational databases can coexist in NoSQL systems. They are appropriate for situations where a lot of data is involved. In this instance, the data is semi-structured, unstructured, or structured. There are various NoSQL database types. These include, among others, Cassandra, Mongo DB, Couch DB, and HBase. They all have different structures and ways of being used, depending on the situation. Such databases have been further categorized into different categories based on these elements. NoSQL databases cannot be utilized interchangeably, hence in this study, their actual applications are classified into the following categories [19][20]. Like relational databases, they are incompatible. You cannot use a different category of NoSQL when one is required in the situation. Each database type has its own unique specialized characteristics in that specific context, allowing the designer to pick the appropriate database type to deploy based on requirement. Key-Valued Stores, Column Family Stores, Document Databases, and Graph Databases are the four types of NoSQL databases.

There are several varieties of NoSQL databases, including: 1) Key-Value databases, which are among the most well-liked ones. With these databases, a key-value pair can be used to perform operations on the database. Key-value stores provide exceptional performance relative to other NoSQL databases and are simple to scale because they rely mostly on primary-key access. CouchDB: Couchbase is one of the most popular key-value databases. It is a product of the Apache Software Foundation, was inspired by Lotus Notes, and is also an open-source NoSQL database with a primary focus on usability [1][2][19][20][21][22][23]. It operates as a single noded database in the same way as other databases [6]. Although it typically starts with a single node instance, clustering is a uniform upgradeable option. It has the capacity to execute a single database across several servers or virtual machines (VMs). Comparatively speaking, a cluster-based CouchDB offers higher capacity and availability than a single-node CouchDB. It is written in the general-purpose language Erlang, and CouchDB is lockless, thus writing to the database doesn't require locking it. This database's documents support the HTTP protocol, JSON, and the addition of non-JSON files as attachments. Therefore, CouchDB is interoperable with any program or application that supports the JSON format [6]. 2) Document databases: The primary formats in these types of databases are documents. It stores and restores XML, JSON, BSON documents. The database stores all information associated with a certain object as a single instance. As a result, document repositories are frequently utilized in web application programming. They offer self-describing, hierarchical tree structures with maps, collections, and scalar values frequently included as embedded documents, which are self-describing documents [5][6]. A simpler transition from relational databases is achievable thanks to the robust and rich query language and structures found in document databases like MongoDB. MongoDB and CouchDB are two well-liked document databases. 3) Column family stores: The data in these databases will be displayed and saved as rows of column families. Numerous columns in these rows are connected to each row's key. the grouping of comparable data that can be accessed at once. A container of rows in an RDBMS table can be compared to each column family. The key in RDBMS systems is used to identify each row (record). Here, each row is made up of many columns (fields). One of the well-known column-family databases is Cassandra, and HBase and Cassendra are also significant column value databases [9][10][15]. Data is logically structured into tables, rows, and columns in HBase, a distributed column-oriented data store built on top of HDFS. HBase files are internally stored in HDFS. Cassendra: Apache Cassandra is an open source, distributed and decentralized/distributed storage system (database), to manage very large amounts of structured data spread out across the world. Without a single point of failure, it offers highly available service [8]. Cassandra [9] has the following characteristics: • It is a column-oriented database. • It is scalable, fault-tolerant, and highly consistent. • It was made for Facebook and afterwards made available as open source. • The data model is based on Google Bigtable. • Amazon Dynamo is the foundation of the distributed design. 4) Graph databases: Graph databases make it easier to store things and the links or connections that connect them. Nodes, which have properties attached to them, are another term for entities. A node is nothing but occurrence of an object in the program. The connections between the nodes—known as edges—show the relationships and can have various features. Nodes are arranged depending on these relationships, and edges indicate the directionality. With this configuration, an intriguing analytics approach that focuses on keywords and topic mining and information retrieval from document-based NoSQL enables knowledge discovery [9]. The most well-known open-source graph database, Neo4j, was created utilizing Java technology. It is very scalable and has no schema (NoSQL).

3. Discussion

MongoDB

Developed by MongoDB, Inc., MongoDB is an open-source document database. MongoDB stores data in JSON-like documents that may vary in structure. The MongoDB query language groups together related data for quick query access. Because MongoDB employs dynamic schemas, records can be created without first specifying the structure, such as the fields or the kinds of information they will hold. Simply adding new fields or removing existing ones will alter the record structure [7][14][15][16][17][18]. You may easily define hierarchical relationships, store arrays, and other more complicated structures with this data model. There is no requirement that all documents in a collection have the same set of fields, and denormalization of data is common. B. Fig. 2. Features of MongoDB. MongoDB features 1. MongoDB offers excellent performance. When compared to relational databases, the majority of MongoDB operations are quicker. 2. In the event of a failure, MongoDB's auto replication (return) capability enables speedy data recovery. 3. Because of sharing, MongoDB supports horizontal scaling. Sharding is the process of partitioning data and distributing it over several machines while maintaining the data's original arrangement. Vertical scaling vs. horizontal scaling: Horizontal scaling refers to adding new machines to handle the data, whereas vertical scaling refers to adding more resources to the present machine. Implementing vertical scaling is challenging, while doing it for horizontal scaling is simple. Examples of databases with horizontal scaling include MongoDB, Cassandra, etc. Horizontal scaling enables MongoDB to maintain and balance the load. 5. High Availability: Auto Replication boosts the MongoDB database's availability. 6. Indexing: Indexes are used to recognize and quickly locate data in order to avoid having to look through every single document in C. The JSON (Java Script Object Notation) format for data storage in MongoDB is based on a subset of the Java Script programming language. This data transfer format is lightweight. JSON supports arrays, hashes, and all the basic data types, including numbers, texts, and Boolean values [3]. In a traditional database, JSON records are used by document databases like MongoDB. Here is a document that stores records in the same way that tables and rows do. This is an example of JSON D. Screenshots: Execution of Queries in MongoDB The execution of the queries in MongoDB to create databases, insert documents, search documents, remove documents, and perform other operations

4. Summary

Schemaless databases are a feature of NoSQL systems. NoSQL databases have the ability to add fields without changing the structure thanks to this feature. NoSQL databases are highly helpful in coping with the enormous development in structured, semi-structured, and unstructured data. In particular, MongoDB's features are highlighted in this paper. With screen photos of a few MongoDB queries being executed, the data representation in JSON format in MongoDB is described. The query performance of MongoDB can be compared with that of other NoSQL databases as part of this study's future focus. It may be possible to implement a new system that combines both types of technologies, allowing relational and NoSQL databases to be configured jointly within the same database management and to work together without hindrance. In addition, they may be able to combine the advantages of each technology solution to produce a system that is more potent and effective.A compliance monitor has been created to put the suggested security measure into action. Between a MongoDB user and a MongoDB server, it serves as a middleman and controls access by monitoring the flow of messages that are exchanged back and forth between the two parties. We also wish to broaden the available approach to support a number of NoSQL data stores. One of the key concerns that must be considered while exchanging data between users is the strengthening of data security. Although the use of unstructured data has increased across many industries, there is still a significant demand for access criteria such authentication, access control, data encryption, secure configuration, and auditing. The proper encryption methods used to various crucial data fields provide data protection without adversely affecting the database's performance in terms of speed or memory usage. Due to the use of appropriate encryption techniques, this is achievable. Since document-oriented and unstructured data are generally kept in databases, it is frequently used to handle various sorts of data. Examples of such databases include MongoDB, Cassandra, CouchDB, Redis, Hyper-table, and others. Since they are open source, there is a critical necessity to offer good security and protect the user's private information when it is in transit or at rest. There is a need for a single solution that can improve the security of data transfer by providing an improved encryption method that speeds up processes and uses less memory when maintaining databases.

REFERENCES

[1] Abramova, V., & Bernardino, J. (2013, July). NoSQL databases: MongoDB vs Cassandra. In Proceedings of the international C* conference on computer science and software engineering (pp. 14-22).

[2] Ali, W., Shafique, M. U., Majeed, M. A., & Raza, A. (2019). Comparison between SQL and NoSQL Databases and Their Relationship with Big Data Analytics. Asian Journal of Research in Computer Science, 4(2), 1-10

[3] Becker, M. Y., & Sewell, P. (2004, June). Cassandra: Flexible trust management, applied to electronic health records. In Proceedings. 17th IEEE Computer Security Foundations Workshop, 2004. (pp. 139-154). IEEE.

[4] Berg, K. L., Seymour, T., & Goel, R. (2013). History of databases. International Journal of Management & Information Systems (IJMIS), 17(1), 29-36.

[5] Bjeladinovic, S., Marjanovic, Z., & Babarogic, S. (2020). A proposal of architecture for integration and uniform use of hybrid SQL/NoSQL database components. Journal of Systems and Software, 168, 110633.

[6] Chandra, D. G. (2015). BASE analysis of NoSQL database. Future Generation Computer Systems, 52, 13-21.

[7] Chen, J. K., & Lee, W. Z. (2019). An introduction of NoSQL databases based on their categories and application industries. Algorithms, 12(5), 106.

[8] Cuzzocrea, A., & Shahriar, H. (2017, December). Data masking techniques for NoSQL database security: A systematic review. In 2017 IEEE International Conference on Big Data (Big Data) (pp. 4467-4473). IEEE.

[9] de Oliveira, V. F., Pessoa, M. A. D. O., Junqueira, F., & Miyagi, P. E. (2021). SQL and NoSQL Databases in the Context of Industry 4.0. Machines, 10(1), 20.

[10] Deka, G. C. (2013). A survey of cloud database systems. It Professional, 16(2), 50-57. IEEE.

[11] Di Martino, S., Fiadone, L., Peron, A., Riccabone, A., & Vitale, V. N. (2019, June). Industrial Internet of Things: Persistence for Time Series with NoSQL Databases. In 2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE) (pp. 340-345). IEEE.

[12] dos Santos Ferreira, G., Calil, A., & dos Santos Mello, R. (2013, December). On providing DDL support for a relational layer over a document NoSQL database. In Proceedings of International Conference on Information Integration and Web-based Applications & Services (pp. 125-132).

[13] Gessert, F., Wingerath, W., Friedrich, S., & Ritter, N. (2017). NoSQL database systems: a survey and decision guidance. Computer Science-Research and Development, 32(3), 353-365.

[14] Guimaraes, V., Hondo, F., Almeida, R., Vera, H., Holanda, M., Araujo, A., ... & Lifschitz, S. (2015, November). A study of genomic data provenance in NoSQL document-oriented database systems. In 2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) (pp. 1525-1531). IEEE.

[15] Rodriguez, K. M., Reddy, R. S., Barreiros, A. Q., & Zehtab, M. (2012, June). Optimizing Program Operations: Creating a Web-Based Application to Assign and Monitor Patient Outcomes, Educator Productivity and Service Reimbursement. In DIABETES (Vol. 61, pp. A631-A631). 1701 N BEAUREGARD ST, ALEXANDRIA, VA 22311-1717 USA: AMER DIABETES ASSOC.

[16] Kwon, D., Reddy, R., & Reis, I. M. (2021). ABCMETAapp: R shiny application for simulation-based estimation of mean and standard deviation for meta-analysis via approximate Bayesian computation. Research synthesis methods, 12(6), 842–848. https://doi.org/10.1002/jrsm.1505

[17] Reddy, H. B. S., Reddy, R. R. S., Jonnalagadda, R., Singh, P., & Gogineni, A. (2022). Usability Evaluation of an Unpopular Restaurant Recommender Web Application Zomato. Asian Journal of Research in Computer Science, 13(4), 12-33.

[18] Reddy, H. B. S., Reddy, R. R. S., Jonnalagadda, R., Singh, P., & Gogineni, A. (2022). Analysis of the Unexplored Security Issues Common to All Types of NoSQL Databases. Asian Journal of Research in Computer Science, 14(1), 1-12.

[19] Singh, P., Williams, K., Jonnalagadda, R., Gogineni, A., &; Reddy, R. R. (2022). International students: What’s missing and what matters. Open Journal of Social Sciences, 10(02),

[20] Jonnalagadda, R., Singh, P., Gogineni, A., Reddy, R. R., & Reddy, H. B. (2022). Developing, implementing and evaluating training for online graduate teaching assistants based on Addie Model. Asian Journal of Education and Social Studies, 1-10.

[21] Sarmiento, J. M., Gogineni, A., Bernstein, J. N., Lee, C., Lineen, E. B., Pust, G. D., & Byers, P. M. (2020).Alcohol/illicit substance use in fatal motorcycle crashes. Journal of surgical research, 256, 243-250.

[22] Brown, M. E., Rizzuto, T., & Singh, P. (2019). Strategic compatibility, collaboration and collective impact for community change. Leadership & Organization Development Journal.

[23] Sprague-Jones, J., Singh, P., Rousseau, M., Counts, J., & Firman, C. (2020). The Protective Factors Survey: Establishing validity and reliability of a self-report measure of protective factors against child maltreatment. Children and Youth Services Review, 111, 104868