A Survey on Load Balancing Approach in MongoDB

[1]Hemangni Mehta, [2]Prof.Jignasa Patel
[1]Department of Information Technology, [2]Department of Information Technology,
[1]Shri S'ad Vidya Mandal Institute of Technology, [2]Shri S'ad Vidya Mandal Institute of Technology,
[1]Bharuch, Gujarat, India, [2]Bharuch, Gujarat, India

***Abstract :** In current emphasis on "Big Data", NoSQL databases are very popular for data storage. NoSQL performs better than SQL databases. When data and number of requests increases, Structure database cannot handle huge data and requests efficiently. One of the solution to overcome these issues is to shift data centres on NoSQL unstructured databases. Here, compare of Relational Database with Non-Relational Database. Then after Explain that how to perform Load balancing technique according to sharding data in MongoDB. The time required for insertion in different databases as well as searching for a different number of threads in the database with a different number of entries. This work also studies the importance of the Sharding and Configuration of the cluster for MongoDB.*

***IndexTerms** - Big data, Mongo DB, SQL and relational database, Non-relational Database, Auto Sharding, Load Balancing.*

I. INTRODUCTION

Traditional information systems for storage are supported the relative model. These are wide referred to as SQL databases named when the language they were queried by. Within the last few years, however, non-relational databases have dramatically up in quality. These databases ar usually referred to as NoSQL databases, clearly marking them Different from the normal SQL databases. Most of those ar supported storing easy key-value pairs on the premise that simplicity ends up in speed .With the rise in accessibility of web and also the availableness of low-cost storage, immense amounts of structured, semi structured, and unstructured information are captured and keep for a spread of applications. Such information is often observed as huge information. Process such huge quantity of information needs speed, versatile schemas, and distributed databases. NoSQL databases became the well-liked currency for operative huge information they claim to satisfy these needs. This conjointly ends up in a surge within the range of NoSQL information offerings. There ar many industrial and ASCII text file implementations of NoSQL databases huge Table and HBase.

II. CHARACTERISTICS: COMPARISION OF NOSQL DATABASE

The use of NoSQL databases and its quality has exaggerated as great deal of information ar being processed. These databases have heap additional benefits compared to relative databases, particularly for big volumes of information. This analysis provides benchmarks and models for 3 of the foremost common NoSQL databases Cassandra, MongoDB and HBase. The databases were tested on the yahoo cloud platform exploitation differing kinds of virtual machines to review the result of various configurations and illustrate the performance behaviour of the databases. The results showed that the models ar ready to capture a lot of of the most performance characteristics of the studied databases.

III. FEATURES OF MONGODB

MongoDB can support the following features:

- **Ad hoc queries -** MongoDB is a document-oriented database with no transactions and joins. So it is easier to write queries.
- **Indexing -** Any attributes can be indexed in the database. Secondary indices are also available.
- **Replication -** MongoDB supports master-slave replication, which ensures redundancy, backup, and automatic failover.
- **Load balancing -** MongoDB scales horizontally using Sharding, which distributes a single logical database system across a cluster of machines.
- **File storage -** Any files can be stored in MongoDB with the function Grid FS. So MongoDB could be used as a file system.
- **Aggregation -** MongoDB supports Map Reduce, which enables users to obtain the result for which SQL GROUP BY

IV. COMPARISION OF RELATIONAL DATABASE WITH MONGODB

Relational information bases ar nice for imposing data integrity. They're the tool of alternative for on-line dealings process. Applications like information entry systems or on-line ordering applications. RDBMS needs that information be normalized so it will offer quality results and forestall orphan records and duplicates. It uses primary and secondary keys and indexes to permit queries to quickly retrieve information. However all of the great intentions that the RDBMS has for guaranteeing information integrity go with a price. Normalizing information needs additional tables, which needs additional table joins, so requiring additional keys and indexes. As databases begin to grow into the terabytes, performance starts to considerably fall off. Often, hardware is thrown at the matter, which may be dear each from a capital end point associated from an in progress maintenance and support point of view. One amongst the popular Document-oriented databases is MongoDB. It's a part of the NoSQL family of information systems. Rather than storing information in tables as is completed in a very "classical" electronic database, MongoDB stores structured information as JSON like documents with dynamic creating the mixing of information in bound kinds of applications easier and quicker.

V. DESIGN OF WORKLOAD-DRIVEN APPROACH

Style OF WORKLOAD-DRIVEN APPROACH There are 3 elements in MongoDB: Routers, the client interface of MongoDB are answerable of accretive information manipulation requests, locating target information by querying Config Server and dispatching the requests to the fragment Servers hosting target data; fragment Servers, providing distributed storage for information, host chunks of information and support dynamic migration of chunks; Config Servers, hosting information of MongoDB, maintain 2 mappings, the one between chunks and their hosted shards, and also the one between id ranges and chunks. The design of our approach is shown in Fig during this design, the 3 broken rectangles severally represent the clusters of fragment Servers, Routers and Config Servers. Solely the elements tagged with italic font are new extended ones we have a tendency to design to modify workload-driven information balance and also the others tagged with traditional font ar original ones of MongoDB. 2 new elements, Monitor and analyzer run on every fragment Server as add-ons. Monitor is to blame for assortment of period standing of fragment Server. Analyzer is accountable to spot the standing of fragment server supported collected information and send sporadically analysis result to Balancer. For every sort of computing resource, associate edge of utilization is preconfigured. If the monitored utilization is on the far side the edge, analyzer can determine the native hotspots of information analyzing log file. Balancer has four components: Collector receives the local analysis results and also the server's load standing sent by all Analyzers on fragment servers and stores them into the Log Storage of Balancer. GA sporadically performs international analysis on Log Storage to spot whole standing of fragment Servers. DM generates the work equalization resolution by the rule of equally dispersive heat of information and minimizing the number of information to be migrated, the particular work equalization is accomplished by invoking Original API of MongoDB. DM sends the invocations to Routers whereas the latter forwards them to fragment Servers. Then, the load are balanced between original fragment Servers and target ones.

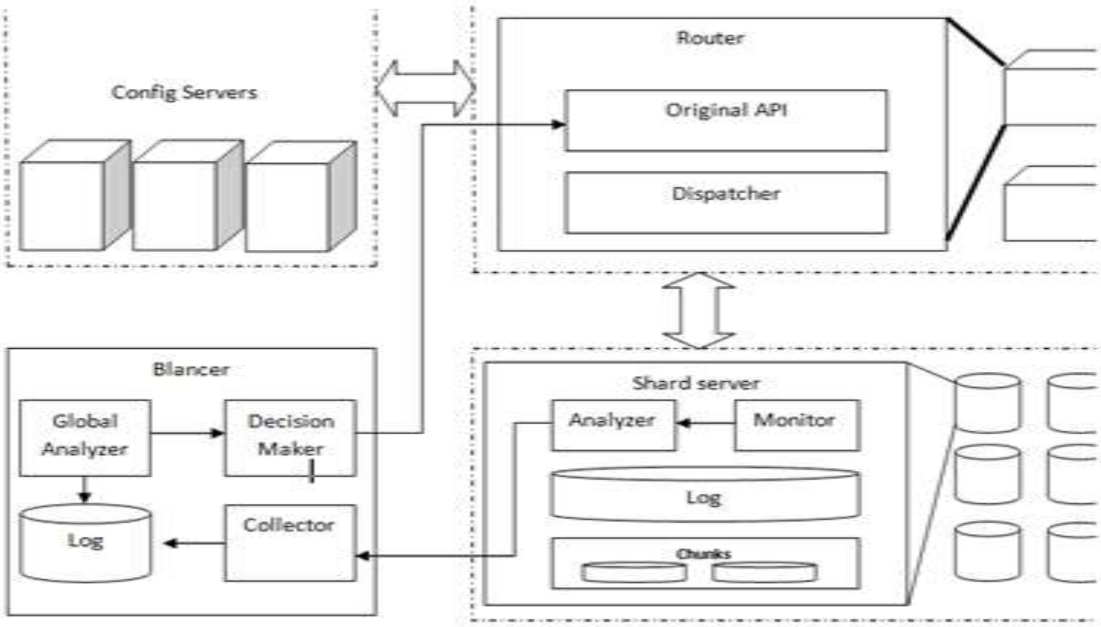

Figure 1. The architecture of workload-driven approach to dynamic data balancing in MongoDB [10]

VI. Related Study

In the earlier works on Big Data Sonali Srivastava Surbhi Agrawal, Aman Srivastava, Adesh Kumar Pandey [1] has proposed a Methodological Approach given about big data concept, its dimensions, its architecture comparison between the earlier concept and the latest, the storage possible

Yishan Li and Sathiamoorthy Manoharan define the A performance comparison of SQL and NoSQL Databases[2] compare read, write, delete, and instantiate operations on key-value stores implemented by NoSQL and SQL databases.

Rajith Kumar & R. Roseline Mary work on Comparative Performance Analysis of various NoSQL Databases: MongoDB, Cassandra and HBase on Yahoo Cloud Server[3] In which final result shows the performance of databases at different workload levels and the result can be compared to find out the best among these three databases.

Gansen Zhao, Weichai Huang, Shunlin Liang, Yong Tang did study on[4] Modeling MongoDB with Relational Model.

Cornelia Gy rödi, Robert Gy rödi, George Pecherle Andrada Olah difine A Comparative Study: MongoDB vs. MySQL [5] mainly focus our presentation on one implementation of the NoSQL database technology, namely MongoDB, and make a comparison with another implementation of relational databases, namely MySQL.

Yunhua Gu1, Shu Shen, Jin Wang, Jeong-Uk Kim did study on Application of NoSQL Database MongoDB [6].

Priyanka Dhaka, Rahul Johari study on Big Data Application: Study and Archival of Mental Health Data, using MongoDB [7].

Satyadhya ChickerurAnoop,Goudar, Ankita Kinnerkar did study on Comparison of Relational Database with Document-Oriented Database (MongoDB) for Big Data Applications[8]It is to illustrate how a problem being solved using MySQL will perform when MongoDB is used on a Big data dataset. The results are encouraging and clearly showcase the comparisons made.

Gokul Prabagaren [9] define Systematic Approach for validating Java-MongoDB Schema.

Shan Lin,Haopeng Chen, Fei Hu analyse on A Workload-Driven Approach to Dynamic Data Balancing in MongoDB[10].

Maeva Antoine, Laurent Pellegrin, Fabrice Huet and Françoise Baude create [11] A generic API for load balancing in distributed systems for big data management.

Yimeng Liu, Yizhi Wang,Yi Jin define [12] Research on The Improvement of MongoDB Auto-Sharding in Cloud Environment.

VII. CONCLUSION

This paper analyses the principle of the MongoDB Auto-Sharding. For the problem of uneven distribution of data among shards, we introduce an improved balancing algorithm. The concurrent writing and reading performance of the Auto-Sharding cluster is significantly improved by using of sharding algoritham.

REFERENCES

[1] Srivastava, S. (2016). BIG DATA - AN EMERGING AND INNOVATIVE TECHNOLOGY : SURVEY. https://doi.org/10.1109/CICT.2016.43

[2] Li, Y., & Manoharan, S. (2015). A performance comparison of SQL and NoSQL databases A performance comparison of SQL and NoSQL databases, (November). https://doi.org/10.1109/PACRIM.2013.6625441

[3] S, R. K., & Mary, R. R. (2017). Comparative Performance Analysis of various NoSQL Databases : MongoDB , Cassandra and HBase on Yahoo Cloud Server, (4), 265–269.

[4] Zhao, G., Huang, W., Liang, S., & Tang, Y. (2013). Modeling MongoDB with Relational Model. https://doi.org/10.1109/EIDWT.2013.25

[5] Gyęrödi, C., Gyęrödi, R., Pecherle, G., & Olah, A. (2015). A Comparative Study : MongoDB vs . MySQL, 0–5.

[6] Principles, A. D. (2015). Application of NoSQL Database MongoDB, 158–159.

[7] Dhaka, P., & Johari, R. (2016). Big Data Application : Study and Archival of Mental Health Data , using MongoDB, 3228–3232.

[8] Chickerur, S. (2015). Comparison of Relational Database with Document-Oriented Database (MongoDB) for Big Data Applications, 41–47. https://doi.org/10.1109/ASEA.2015.19

[9] Nadu, T., & Prabagaren, G. (2014). Schema, (978).

[10] Lin, S., Chen, H., & Hu, F. (2015). A Workload-Driven Approach to Dynamic Data Balancing in MongoDB, 786–791. https://doi.org/10.1109/SmartCity.2015.163

[11] Antoine, M., Pellegrino, L., Huet, F., & Baude, F. (2016). A generic API for load balancing in distributed systems for big data management, (August 2015), 2440–2456. https://doi.org/10.1002/cpe

[12] Liu, Y., Wang, Y., & Jin, Y. (2012). Research on The Improvement of MongoDB Auto- Sharding in Cloud Environment, (Iccse).