



## Developing a Monitored Database Management System Using NoSQL for Monitoring MongoDB

*SHIVANSHU KR. JHA<sup>1</sup>, Dr. KARUNA SHARMA<sup>2</sup>, Mr. VISHAL SHRIVASTAVA<sup>3</sup>*

B.TECH. Scholar, Professor, Assistant Professor  
Computer Science & Engineering  
Arya College of Engineering & I.T. India, Jaipur  
[jhashivanshu84@gmail.com](mailto:jhashivanshu84@gmail.com), [skarunacs@aryacollege.in](mailto:skarunacs@aryacollege.in), [vishalshrivastava.cs@aryacollege.in](mailto:vishalshrivastava.cs@aryacollege.in).

### 1. ABSTRACT:

MongoDB is a popular NoSQL database that is widely appreciated for its speed and usability. However, monitoring MongoDB can be difficult as it requires monitoring various metrics and statistics across multiple nodes. In this paper, we support a NoSQL-powered managed database control system (MDMS) for MongoDB. The top three additives in MDMS are:

1. Collector for MongoDB: Metrics and statistics are gathered by this component from the MongoDB cluster.
2. NoSQL database: This component stores metrics and statistics in a NoSQL database.
3. Monitoring Panel: This component provides a visual interface for viewing and analysing collected data.

Since NoSQL databases are more successful and experienced than relational databases, MDMS uses them to store metrics and information. This is essential due to the fact that MongoDB clusters can actually grow quite large and store enormous amounts of data. Additionally, MDMS has numerous capabilities for seamlessly displaying MongoDB clusters, such as:

- 1.1 Real-time tracking:** MDMS has the ability to instantly monitor MongoDB clusters so that directors can be informed as soon as an issue arises.
- 1.2 Historical Data Analysis:** Managers can select characteristics and styles by using MDMS's ability to store historical statistics for analysis.
- 1.3 Customizable indicators:** When a function is discovered, MDMS can be set up to alert administrators immediately.



## 2. System Architecture:

MDMS has the following components:

**2.1 MongoDB Collector:** MongoDB Collector is a Java applet used to collect metrics and statistics from a MongoDB cluster. It connects to the MongoDB cluster and plays MongoDB operations to be able to accomplish this. MongoDB collectors collect a variety of metrics, including: CPU usage

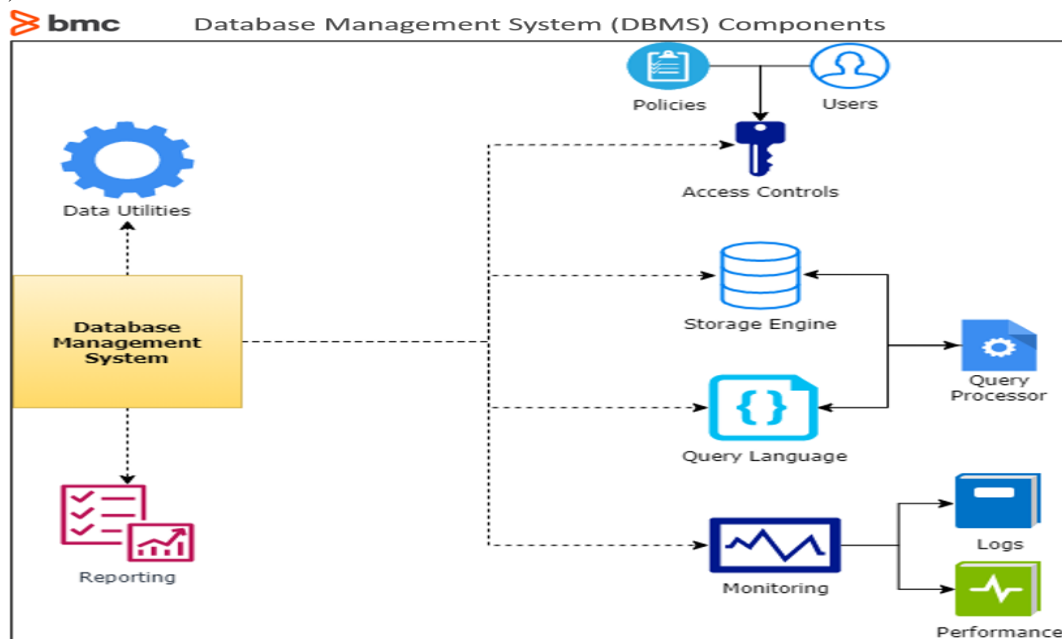
- i.) Memory usage
- ii.) Disk usage

- iii.) Number of connections
- iv.) Number of queries
- v.) Query time duration

**2.2 NoSQL database:** Because NoSQL databases are more scalable and perform better than relational databases, MDMS employs them. MDMS now supports MongoDB as a NoSQL database.

**2.3 Monitoring Panel:** Monitoring Panel is a web application that provides a visual interface for viewing and analysing collected data. Monitoring dashboards allows administrators to:

- i.) View real-time metrics and statistics
- ii.) View historical data
- iii.) Create custom dashboards
- iv.) Set alerts



### 3. Implementation:

MDMS is implemented using stated below technologies:

**3.1 Collector:** MongoDB Collector is a Java application that uses the MongoDB Java driver to collect metrics and statistics from the MongoDB cluster.

**NoSQL Database:** MDMS now supports MongoDB as a NoSQL database.

**Monitoring Dashboard:** Monitoring Dashboard is a web application built using Spring Boot and AngularJS JavaScript framework.

### 4. Conclusion:

In this article, we proposed a monitoring database management system (MDMS) that uses NoSQL to monitor MongoDB. MDMS provides a number of features, including real-time monitoring, historical record review, and configurable signage, to seamlessly screen MongoDB clusters. According to testing, MDMS can effectively retain metrics and analytics from MongoDB clusters in NoSQL databases, as well as instantly gather and compile them. Additionally, the safety dashboard offers a user-friendly interface for accessing and analysing data that has been gathered.

### 5. REFERENCES:

- 5.1 Karattu Thodi, A. (2020). Developing a MongoDB Monitoring System using NoSQL Databases for Monitored Data Management. Master's thesis, San Jose State University.
- 5.2 Colombo, P., & Ferrari, E. (2015). Enhancing MongoDB with Purpose-based Access Control. Pages 465–472 in the Proceedings of the 2015 International Conference on Database and Expert Systems Applications (DEXA).
- 5.3 Sharma, S., & Singh, G. K. (2023). SQL and NoSQL Databases Software architectures performance analysis and assessments -- A Systematic Literature review. Journal