ISSN 2278-3091
Volume 14, No.4, July - August 2025
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse011442025.pdf
https://doi.org/10.30534/ijatcse/2025/011442025

Database Performance Monitoring and Analysis Tools

Anatolii Bobunov
[1]Bachelor's Degree, Moscow University for Industry and Finance «Synergy», Moscow, Russia
dev.bobunov@rambler.ru

Received Date: June 24, 2025 Accepted Date: July 26, 2025 Published Date: August 06, 2025

ABSTRACT

This article examines database performance monitoring and analysis tools, including their classification, data collection methods, and analysis techniques. It analyzes built-in solutions provided by DBMS developers as well as third-party platforms that offer greater versatility and advanced functionalities. The importance of such tools is emphasized in the context of system stability, efficiency of resources, and overall performance. The future technology trends of AI integration, automation, and multi-cloud support are also addressed in the paper. Special attention is given to the integration monitoring of other IT systems and process automation so that the requirements of modern business scenarios could be met.

Key words : monitoring, performance analysis, databases, monitoring tools, automation, optimization.

1. INTRODUCTION

In the context of the rapid growth of data volumes and increasing application complexity, the need for reliable tools to assess database performance is becoming increasingly significant. The quality of such monitoring directly touches not only stability but also possible problems that may manifest as performance degradation and can lead to failure. Databases today, whether on-premise or cloud, need some modern and innovative methods with specialized toolkits that promise very high performance while reducing their time to market or recovery in the event of an outage.

This paper aims to perform an analysis of the performance monitoring tools and methods of analysis that exist for databases, classify them, and point out the advantages and limitations of these. The investigation covered the built-in tools of database management system (DBMS) and third-party products, their function of detection and solution of performance issues, and their ability for integration into other systems.

2. CLASSIFICATION OF DATABASE PERFORMANCE MONITORING TOOLS

Specialized tools are utilized to perform effective database performance monitoring. These tools can be divided into groups according to origin, functionality, methods of data collection, and scope of application [1].

Built-in monitoring tools are developed by DBMS providers and are designed to work exclusively within a specific platform. These solutions are deeply integrated with the DBMS core, providing access to internal metrics and database processes. They have the advantage of being native: they are more tightly coupled with the DBMS, data granularity is fine, and there is no need for extra setup. These tools can be somewhat limiting when it comes to functionality and scalability, especially in multi-platform setups.

Third-party monitoring tools are developed independently and include universal solutions applicable to different DBMS platforms. They are designed around maximum functionality: data visualization, trend analysis, and integration with DevOps systems. Third-party solutions, as a general rule, are flexible and can operate on different platforms, what is particularly relevant in the case of heterogeneous IT infrastructures. These tools may take considerable time for deployment and configuration.

Monitoring tools can differ in their data collection approaches: the choice of a specific method depends on the monitoring objectives, available resources, and required data granularity (table 1).

Table 1: Data collection approaches in database monitoring tools [2, 3]

Data collection approach	Description	Examples and details
Agent-based collection	Uses software agents installed on the database server to collect performance metrics.	New Relic, DataDog – provide high-level detail but increase server load.
Agentless collection	Data is collected via built-in APIs or system logs. This approach	Built-in API solutions, PostgreSQL logs

	reduces server load but may be less detailed.	– reduce server load but provide less detail.
Real-time monitoring	Continuous observation of database processes, allowing immediate issue detection and resolution.	Datadog, Prometheus – provide real-time metrics, useful for high-load systems.
Historical data analysis	Stores and processes data over a defined period for trend analysis and forecasting.	Datadog, Zabbix – allow trend analysis, useful for strategic planning.

It provides an excellent classification of the benefits and shortcomings of each approach and helps to choose the best fit for a particular system or organization. Current trends are toward using hybrid and cloud-based monitoring tools due to their flexibility and ease in handling distributed infrastructures.

3. PERFORMANCE ANALYSIS METHODS

Database performance analysis is a crucial component in ensuring the stability and efficiency of information systems. Properly selected analysis methods not only help identify existing issues but also predict potential bottlenecks in the system, preventing performance degradation (table 2).

Table 2: Metrics for database performance analysis

Metric	Description
CPU utilization	Indicates the level of computational resource usage by the server running the database.
Response time	Measures the speed of query execution, including transaction processing and data transfer time.
Memory usage	Characterizes the use of RAM for temporary data storage and caching.
Throughput	Defines the volume of data processed by the database per unit of time.
Transaction latency	Describes delays in performing operations related to data writing and reading.

These performance metrics give the administrator an overview of database efficiency and allow them to identify and manage performance problems effectively without creating system stability issues. Through ongoing monitoring of CPU utilization, response time, memory consumption, throughput, and transaction latency, organizations can maximize resource usage, enhance query performance, and avoid potential bottlenecks.

Apart from this, this allows for historical examination of these measures in terms of forecasting trends to allow enterprises to expand their infrastructure in advance to cater to the rising demands. With databases continuing to handle volumes of data in increasing order of magnitude, leveraging these metrics in concert with automated monitoring and artificial intelligence (AI) driven analytics will hold the key to optimal performance and reliability.

It should also be noted that database performance monitoring and analysis tools are widely used in software testing processes [4]. They provide both regression and load testing along with real-time monitoring of such critical parameters as the response time of queries, CPU and memory usage, and open transactions. This allows for instant identification of performance hot spots, slow query profiling, and examination of the impact of changes in data structure or DBMS parameters on total performance. Therefore, the tools represent a crucial part of quality assurance (QA) practice and DevOps process, ensuring system reliability through all stages in the software life cycle.

4. APPLICATION OF MONITORING TOOLS IN DIFFERENT DBMS

Contemporary DBMS offer integrated monitoring utilities tailored for their architectures and operational features. These tools aid in gathering performance metrics, analyzing system efficiency, and diagnosing issues, rendering them essential for managing intricate information systems.

For instance, Oracle Database utilizes Automatic Workload Repository (AWR) and Automatic Database Diagnostic Monitor for performance monitoring and analysis [5]. AWR automatically collects and stores key performance metrics, creating system snapshots every 60 minutes (by default). These snapshots capture data on CPU utilization, memory usage, active session count, transaction duration, and executed queries (figure 1).

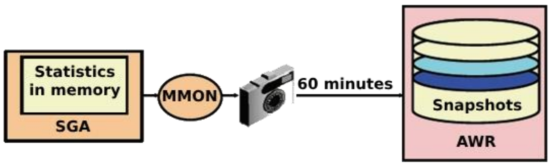

Figure 1: AWR scheme

The data is kept in the repository for a certain amount of time so that current and historical trends can be analyzed. AWR also generates reports pinpointing problems-such as slow queries, disk subsystem bottlenecks, or overloaded processors. The Automatic Database Diagnostic Monitor (ADDM), which operates based on AWR data, automatically performs database performance diagnostics and generates recommendations for optimization (figure 2).

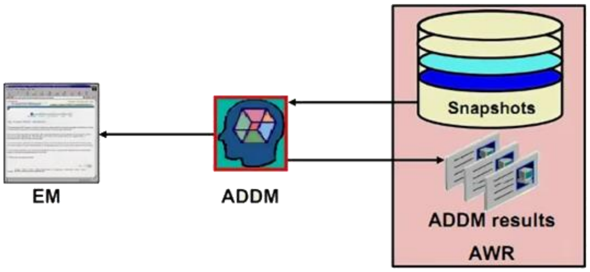

Figure 2: ADDM scheme

The collected metrics are analyzed by ADDM to identify the root causes of performance issues, such as lock contention, improper configuration of indexes, poor input/output performance, or overload of resources. Further, ADDM gives practical recommendations for optimization based on this analysis. Together, AWR and ADDM offer a powerful combination ranging from monitoring to diagnostics, therefore becoming an inseparable tool in managing high-load database systems.

Microsoft SQL Server has some inbuilt tools for the same, namely SQL Server Profiler and Extended Events, to enable administrators to monitor database performance, analyze query behavior, and identify system issues [6]. SQL Server Profiler is a traditional tracing tool used to record the events of a database, including SQL query execution, table modifications, locks, and transaction delays. It gives detailed query analysis by tracing execution time, the number of rows processed, and index usage. Because of this, it is particularly handy in troubleshooting slow queries and performance optimization. However, prolonged use of SQL Server Profiler can create a significant system load, limiting its applicability in high-performance environments.

Extended Events is a more advanced, flexible diagnostic tool intended to supersede Profiler. It makes possible the selecting and collection of only the most relevant events by the administrator, minimizing system overhead. Working on the principle of asynchronous logging, the extended events are more efficient than Profiler in monitoring query execution, user activities, server configuration changes, and performance-related problems. It works with SQL Server Management Studio for visual data representation.

This tool is highly effective at determining complex issues such as long-running locks, parallel transactions, and system-wide failures. It also grants the capability of explaining stored procedures execution, CPU usage, and memory consumption, therefore it is also a great solution for deep performance analysis.

PostgreSQL has pg_stat_statements and pgAdmin as monitoring utilities [7]. Pg_stat_statements gathers statistics about executed queries, including the number of executions, total time taken, and resources used by the queries, thus helping to identify the most resource-intensive operations. On the other hand, pgAdmin is a graphical interface to monitor and manage PostgreSQL, with load analysis and system configuration capabilities.

MySQL has special tools for performance monitoring and analysis, such as Performance Schema and MySQL Enterprise Monitor, which allow the administrator to get detailed information on the state of the database system, find bottlenecks, and further optimize its performance [8]. Performance Schema a built-in monitoring mechanism-collects low-level data about internal database processes without high system overhead (figure 3).

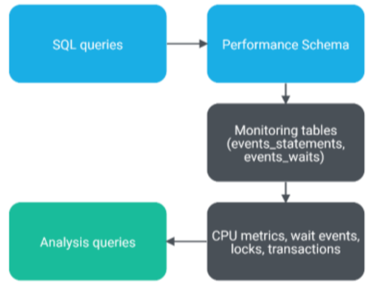

Figure 3: Performance Schema

Another of the major benefits of Performance Schema is its flexibility: it enables administrators to enable and disable the gathering of certain metrics, hence controlling the granularity of the data. This tool also performs query performance analysis, such as average execution time and index usage frequency, which becomes important for query optimization, especially in the case of slow ones.

MySQL Enterprise Monitor offers a more comprehensive monitoring solution, incorporating data visualization, automated analysis, and alerts for potential issues (figure 4).

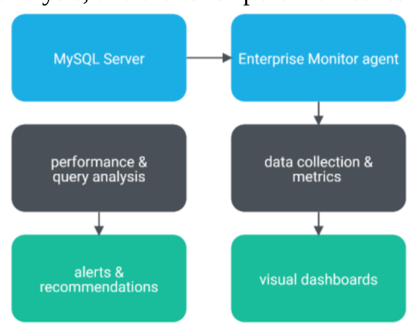

Figure 4: MySQL Enterprise Monitor scheme

This solution offers a basic web-based interface by which administrators can view real-time data on system load, connection activity, and performance metrics. Unlike Performance Schema, which requires working with SQL queries in order to analyze data, MySQL Enterprise Monitor works on one's behalf to create graphs and reports, rendering it much easier to diagnose and conduct trend analysis in determining performance.

In addition to native tools, third-party tools play a critical role in managing database performance management in particularly complicated systems. Cross-platform support, in-depth analytics, and customizability as their particular design objectives, tools like Zabbix, Prometheus, Datadog, and New Relic all possess straightforward compatibility with other database platforms and IT infrastructure. Zabbix provides end-to-end full-stack monitoring for databases, servers, and networks through centralized dashboards for real-time system health monitoring. Prometheus provides optimized time-series data gathering and, along with Grafana, provides decent database performance visualization. Datadog and New Relic utilize AI and machine learning in order to identify anomalies, foresee issues, and root out performance bottlenecks. All of these tools are particularly handy in hybrid and multi-cloud environments, with the promise of providing seamless database performance. They are adaptable, automated, and built into DevOps, making them vital for active monitoring, rapid problem discovery, and continuous optimization.

5. PRACTICAL ASPECTS OF IMPLEMENTING MONITORING SYSTEMS

Implementation of database monitoring systems is complex and complicating work that ensures stability, performance, and predictability of information systems. Effective deployment necessitates attention to several crucial factors, beginning with the choice and setup of a tool to its incorporation with the current infrastructure and automation.

One of the first actions is selecting the suitable monitoring tool, dependent on system needs, the type of DBMS being utilized, and available resources. After selection, it has to be installed and configured; installation will differ based on the environment it will be deployed in. In case of on-premises DBMS, it will be about performance parameters tuning, enabling the necessary modules, and integrating with visualization tools. For instance, MySQL Performance Schema requires enabling metric collection, while in SQL Server Extended Events, special sessions need to be created that will monitor the activity. Monitoring set up in cloud environments involves connecting to cloud services, creating dashboards, and setting up automated alerts. As an example, AWS CloudWatch provides a possibility to configure metrics and set up notification triggers for critical thresholds.

To improve monitoring effectiveness, it ought to be connected with logging systems and DevOps platforms for cohesive data gathering, thereby facilitating automated problem identification. Instruments such as Grafana facilitate the development of real-time dashboards, whereas Zabbix and Datadog enable automated alerts to be generated when a decline in performance is identified.

Another important feature is monitoring automation, which enables prompt reaction in case of a potential failure and reduces human involvement. This involves setting threshold values for critical metrics; for instance, automatic notification in case of CPU usage exceeding 80%. In addition, predictive analytics mechanisms will analyze past trends to predict issues such as increased execution time of queries or increased load on the system.

Once monitoring is implemented, continuous analysis and adaptation are necessary. Regular review of reports allows for the identification of long-term trends, such as increased query response times or a growing number of connections. Based on this data, database optimization can be done, including indexing slow queries, reallocation of resources, or modification of configuration settings.

Thus, the successful implementation of database monitoring systems involves an overall approach: tool selection, proper configuration, integration with IT systems, and automation. This would lead to a definite improvement in performance and reliability, less downtime, and lower administration costs for the databases.

6. PROSPECTS FOR THE DEVELOPMENT OF MONITORING TOOLS

Database monitoring systems continue to evolve, adapting to growing data volumes, advancements in cloud technologies, and increasing demand for automation [9]. The key directions in their evolution include the integration of AI, process automation, and multi-cloud environment support.

Moving down, machine learning for predictive analytics, further down the line, will allow us to see overloads and failures far in advance and provide an automatic solution. Systems like Datadog, AWS CloudWatch, and New Relic apply AI algorithms today to detect anomalies and predict problems. In years to come, databases will self-optimize queries, workloads dynamically redistribute their workloads, and bottlenecks get avoided with minimal administrator's intervention.

Another is observability, which is a growing demand and involves monitoring, query tracing, and logging with real-time analytics and stream processing technologies like Apache Kafka and Spark Streaming. All these will help to analyze workload fluctuations and transaction delays instantly, keeping the system stable. Therefore, database monitoring of the future would be about automation, cloud integration, and AI-driven solutions for the creation of more autonomous and intelligent systems.

7. CONCLUSION

Database performance monitoring and analysis tools are an essential attribute of contemporary IT infrastructure, which ensures stability, efficiency, and predictability of a system. From the built-in tool to the third-party platform, it can be adapted for a wide range of solutions for different tasks and environments, whether for on-premises or cloud infrastructures.

Modern monitoring technologies keep on developing, offering increasingly intelligent, automated, and integrated approaches. Successful implementation will also reduce administration costs while contributing to the development of more resilient and flexible information systems that are better able to meet the challenges presented by continuous growth in data volume and more and more complex business processes.

REFERENCES

1. R. Altaher. **Transparency Levels in Distributed Database Management System DDBMS.** 2024.
2. A. Dudak. **Virtualization and rendering of large data lists,** *Cold Science,* no. 9, pp. 17-25, 2024.
3. K. Zekhnini, B. A. Chaouni, A. Cherrafi. **A multi-agent based big data analytics system for viable supplier selection,** *Journal of Intelligent Manufacturing*, Vol. 35, no. 8, pp. 3753-3773, 2024.
4. A. Bobunov. **Approaches to monitoring and logging in automated testing of financial applications,** *Sciences of Europe,* no. 142, pp. 62-66, 2024.
5. G. I. Arb. **Oracle Database Performance Improvement: Using Trustworthy Automatic Database Diagnostic Monitor Technology,** *Computer Science*, Vol. 19, no. 3, pp. 881-892, 2024.
6. T. V. Kumar. **A Comparison of SQL and NO-SQL Database Management Systems for Unstructured Data**. 2024.
7. A. R. Ravshanovich. **Database structure: postgresql database,** *Psixologiya va sotsiologiya ilmiy jurnali,* Vol. 2, no. 7, pp. 50-55, 2024.
8. Y. V. Kumar, A. K. Samayam, N. K. Miryala. **MySQL Enterprise Monitor, Mastering MySQL Administration: High Availability, Security, Performance, and Efficiency,** *Berkeley, CA: Apress*, pp. 627-659, 2024.
9. R. Garifullin. **Development and implementation of high-speed frontend architectures for complex enterprise systems,** *Cold Science,* no. 12, pp. 56-63, 2024.