International Journal of Mathematics and
Computer Science, **19**(2024), no. 3, 881–892

# Oracle Database Performance Improvement: Using Trustworthy Automatic Database Diagnostic Monitor Technology

**Ghusoon Idan Arb**

College of Computer Science and Information Technology
University of Sumer
Thi qar, 64005, Iraq

email: eng.computer38@gmail.com

(Received November 6, 2023, Revised February 9, 2024,
Accepted February 10, 2024, Published February 12, 2024)

**Abstract**

The Oracle database is widely acknowledged for its high level of trustworthiness. This paper elucidates the manner in which Oracle DB provides an automated methodology for the identification and optimization of performance issues. The capacity for self-tuning is an essential element in the development of a dependable and trustworthy self-directed database. This paper offers a comprehensive examination of Oracle's self-tuning capabilities, focusing on the Automatic Database Diagnostic Monitor (ADDM) as a prominent automatic tuning solution. The discussion encompasses an overview of these capabilities and a thorough presentation of ADDM's features and functionalities. The ADDM (Automatic Database Diagnostic Monitor) is a tool that systematically evaluates and scrutinizes the data obtained from the Automatic Workload Repository (AWR) in order to detect and assess possible performance concerns within the Oracle Database. To assess the efficacy of ADDM, we formulated a program to replicate substantial server workloads and effectively implemented ADDM suggestions to address the identified issues, thereby ensuring a trustworthy resolution.

**Key words and phrases:** Oracle Database Performance, Automatic Database Diagnostic Monitor, Automatic Workload Repository, Server Bottleneck.
**AMS (MOS) Subject Classifications**: 68Uxx.
**ISSN** 1814-0432, 2024, http://ijmcs.future-in-tech.net

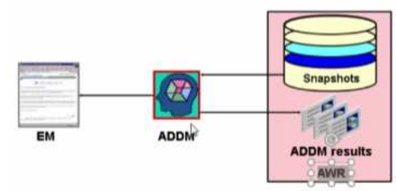


Figure 1: EM tool.

# 1 Introduction

ADDM is completed out of the container once consistently, every time an automatic workload repository snapshot is created. Consequences of these examinations are recorded for the DBA to message previous performance tasks. Programmed execution determination is exceptionally challenging since present day DB frameworks have difficult cooperation through their sub-segments and can work with an assortment of uses [1], [2]. This outcome in an especially enormous rundown of possibility performance approaches such a programmed examination could recognize (see Figure 1). ADDM was structured with the accompanying goals:

1. Should force a comprehensive perspective on the DB and comprehend the collaborations between different DB parts.

2. Should be fit for recognizing side effects from the genuine underlying driver of execution bottlenecks.

3. Should give components to analyze performance action on their first event.

4. Should effortlessly stay aware of changing advancements [3].

Automatic workload repository architecture for information gathering, examination, and arrangement proposals. Automatic workload repository baseline is a group of automatic workload repository snapshots for execution examination. Metric is a piece of progress in an aggregate static. Measurement is information assortment giving DB and article subtleties. It is worked in store of execution data; its depiction of DB measurements taken at regular intervals and held for a month, AWR institution for all self-administration capacities (see Figures 2 and 3). AWR creates depictions of execution information once consistently, and holds the insights in the outstanding burden vault for 7 days. It is conceivable for changing the default metric for both

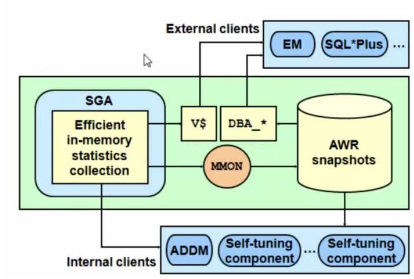


Figure 2: Automatic workload repository infrastructure.

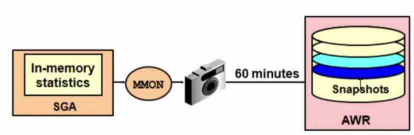


Figure 3: Automatic workload repository snapshot.

the snapshot interim and the maintenance time frame. The information in the snapshot interim is then broke down by ADDM. AWR looks at the distinction between previews to figure out which SQL announcement to catch, in view of the impact on the framework load. This diminishes the number in EM program, the performance synopsis page shows metric qualities for disk input /output use, memory CPU use and the main 10 procedures requested by both memory and CPU use.
To decide whether the host framework has enough assets accessible to run the DB, first set up suitable desires for the measure of resources which the framework ought to utilize. At that point, decide if adequate assets are accessible and take note of when the framework is utilizing such a large number of resources. Start by deciding the measure of disk, memory, and CPU [4], [5].

# 2 The Method

Performance of various entities of the DBs are measured using many metrics. For example, getting impartial feedback from clients, or getting a full set of OS, DB, and application statistics from the system when the performance is at its best and its worst. ADDM performs investigation of AWR every hour or at some other client-characterized time for AWR action. The essential target of ADDM is to screen the instance activity, inform the admin of any issues

and record likely reformist activities. ADDM examines the side-effects and expects to distinguish their motivation so as to limit the time database use in preparing client demands. By presenting this shared factor, it improves tuning by binding together the affected measure empowering correlations between a few regions of benefit [3]. An example yield of ADDM by EM is introduced in Figure 1. It permits penetrating down the significant data so as to assemble more data and comprehension of emerging issues. ADDM additionally proposes a lot of activities that intend to fix found issues. Those activities might be elective ways prompting the mitigation of experienced execution issues and must not be actualized completely (as in SQL Access Advisor). Among other things, ADDM will break down the accompanying regions of database activity:
I. Input/output issues.
II. Memory failure.
III. Top burden SQL.
IV. Central Processing Unit bottlenecks.
ADDM goes about as a core guide, and it is important to remember other guides in this manner planning tuning exercises. Its subsequent records are held for a month which makes checking common performance levels a simpler undertaking. Its concentrated endeavors expect to minimize the time that the database spends handling client demands. Different goals incorporate [4]:

- Adjusting to changing advancements and new executions.
- Giving comprehensive framework and including associations between parts.
- Isolating features from causes.
- Featuring performance issues as right on time as their first event.

Reactive DB Tuning such as AWR and ADDM record investigation can increase comprehension of the purposes behind client's objections, correlation of good and awful execution times and other authentic information, as well as ASH record examination for brief performance issues. AWR is the core checking instrument in the database. It makes a screenshot on essential performance measurements consistently as delta amount and stores them sequentially for ADDM's examination. Establishing baselines by choosing reports of good execution for certain times, directs the framework toward

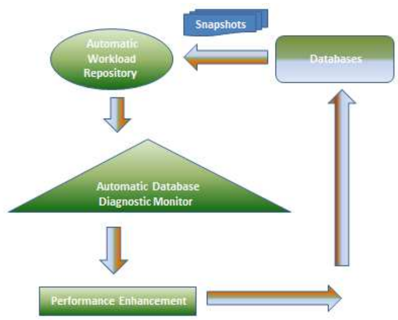


Figure 4: ADDM improves performance.

the expected performance levels. Update limits permit greater adaptability and adjustment to the evolving outstanding burden. Creating the AWR announces a merited piece of self-tuning and self-checking database engineering. Preview recording happens like clockwork, and in spite of the fact that it is conceivable to change meantime and maintenance periods, ASH report were actualized to give extra contribution to ADDM at an alternate degree of granularity. Active Session Hashing accumulates insights with avoidance of inactive session and records them in a piece of SGA, which is restricted by the physical memory accessible. This may cause inadequate information, yet it gives an increasingly gritty contribution to ADDM's examination, which thus may highlight places that ought to be additionally explored. Despite the fact that SQL Trace is excluded from the analytic gathering appraisal, contrasting its usefulness and ASH reports is valuable in understanding the best setting for usage of the two devices (see Figure 4 in [2]).

## 2.1 Trustworthy concepts and roles

Trustworthiness is a crucial factor to consider when assessing the dependability, integrity, and credibility of information, systems, or organizations. Trustworthiness can be defined as the capacity of technology users to place confidence in a system or entity to operate as intended and promote their own interests. Trustworthiness encompasses a set of indispensable attributes [7-12] (see Figure 5): 1. Security: In order to protect data and infrastructure from unauthorized access, breaches, and malicious activities, reliable systems prioritize the implementation of security measures. The aforementioned measures encompass robust access limitations, encryption protocols, authentication methodologies, and regular security enhancements.
2. Trustworthy systems ensure the preservation of user privacy through the

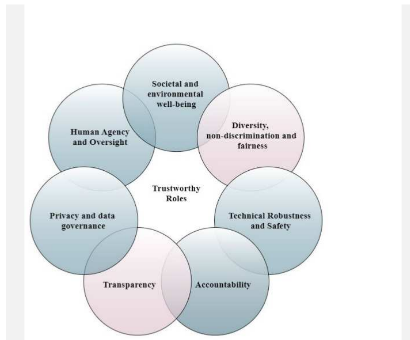


Figure 5: Trustworthiness roles.

implementation of appropriate measures for data protection. The organization obtains user consent as necessary and maintains clear guidelines pertaining to the acquisition, retention, and utilization of data. Additionally, they offer customers the means to exert control over their data and ensure transparency regarding the handling of user data.
3. Reliability refers to the consistent operation of systems without any unexpected faults or malfunctions. In the event of system outages or disturbances, the organization has implemented robust backup and disaster recovery protocols to ensure the continuous availability of data.
4. Transparency is a fundamental characteristic exhibited by reputable organizations, as they demonstrate a commitment to openness and honesty in their operational practices, commercial transactions, and management of data. The terms of service, privacy policies, and data use regulations are presented in a lucid and accessible manner. Any potential conflicts of interest that may have an impact on their services are also disclosed.
5. Accountability: Trustworthy systems demonstrate a willingness to accept responsibility for their actions and take ownership of any shortcomings or violations. The organization exhibits prompt responsiveness to issues, offers effective resolutions, and implements preventive measures to mitigate the likelihood of future incidents. Individuals have the ability to employ these strategies in order to express their apprehensions and offer their opinions.
6. Ethical Conduct: Respected enterprises demonstrate ethical behavior through adherence to legal regulations and professional standards. The individuals refrain from engaging in unethical business practices, deceptive advertising, or unauthorized exploitation of consumer data. The company prioritizes the well-being of its customers and diligently strives to ensure that its services are characterized by equity and fairness.

7. In order to enhance credibility, it is imperative to employ a combination of technological measures, organizational protocols, and ethical deliberations. Various forms of verification, such as independent audits, certifications, and adherence to established guidelines and best practices, can be considered as indicators of reliability.
8. Ultimately, the establishment of credibility is imperative in order for consumers to place their trust in technology, systems, or organizations when it comes to safeguarding their sensitive data, conducting transactions, and engaging in online interactions. The incorporation of user interactions is known to have a positive impact on the overall user experience, thereby contributing to the success and longevity of technological solutions.

# 3 Challenge, Motivations and Limitations

This part discusses, the challenges, motivations and limitations of this study (see Figure 6).

## 3.1 Motivations

The motivation for doing this research arises from the considerable importance of database performance in ensuring the effectiveness of applications reliant on Oracle databases. The use of Oracle databases is widespread in the administration of substantial workloads, so enhancing their performance is a subject of significant significance. The objective of this research is to examine the use of Oracle's Automatic Database Diagnostic Monitor (ADDM) technology in a specific environment. The primary aim is to autonomously identify and address issues related to performance, hence enhancing the overall trustworthiness and dependability of the database.

## 3.2 Challenges

The complexity of contemporary database systems is an important challenge within the industry. These systems have a complex nature and are distinguished by the existence of several interrelated components. The complex nature of this complexity is a significant challenge in accurately determining the exact sources of performance bottlenecks, since problems might arise from the interaction of several components. The determination of the fundamental factor responsible for performance bottlenecks is a multifaceted endeavor. In order to provide exact recommendations for improvement, it is crucial for automated systems like ADDM to efficiently distinguish between symptoms and the underlying causes. The reliability of recommendations is based upon the accuracy and reliability of data collection. Ensuring the precision of performance metrics is of paramount importance in effectively identifying and resolving issues.

The issue of privacy and security becomes relevant throughout the process of gathering performance data, since it has the potential to include sensitive or confidential information. The challenge is in achieving a harmonious equilibrium between the need to get complete performance insights and the obligation to maintain the confidentiality and protection of data. The prompt identification of performance issues is crucial in order to minimize the possible impact on users of an application. The issue of guaranteeing timely diagnosis and suggestions is especially evident in intricate and ever-changing database configurations.

## 3.3 Limitations

1. The focus of this research is primarily on Oracle databases, which restricts its relevance and application to other database management systems.
2. The study heavily relies on the use of Oracle's Automatic Database Diagnostic Monitor technology. One possible shortcoming of this technique is its inability to include alternative or supplementary diagnostic tools provided by other vendors.
3. Performance Implications: The use of diagnostic tools, such as ADDM, might introduce a measure of performance overhead on the database system, potentially affecting its responsiveness during the diagnostic procedure.
4. The role of human expertise in the adoption of ADDM's automated suggestions is crucial to ensure that they are aligned with specific application needs and validated.

The study may not thoroughly examine the performance difficulties that emerge due to the dynamic nature of workloads and changing system settings.

In brief, the main aim of this research is to enhance the effectiveness of Oracle databases by using the capabilities of the Automatic Database Diagnostic Monitor (ADDM) technology. However, complicated information systems, data collection accuracy, worries about security, and the promptness of diagnosis provide further challenges. The limitations of the study include its limited scope, which only focuses on Oracle databases, its dependence on diagnostic tools for data gathering, and the necessary reliance on

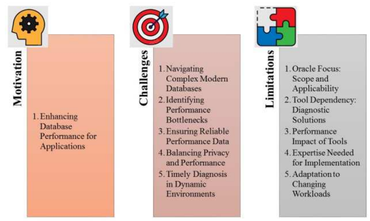


Figure 6: Challenge, motivations and limitations.

human expertise throughout the implementation phase.

# 4 Results and Discussion

For testing the oracle database performance, we write a hundred thousand records in the table and segregate each record. This causes the server to suffer from a high load, even writer processes will suffer from an overload. We will write a code to try entering the server in a high load situation, then we use ADDM recommendation to solve this problem. For the code in Figure 4, the first loop has a hundred thousand iterations and the second loop has ten thousand iterations until it becomes bottleneck on the server. When it combines this to a set of records in the new table called (test_adv), it remains implemented a hundred thousand times for the filling the new records in the table. These operations are performed via transaction [13]-[14]. This is unproblematic for the server because one loop takes a split of a second, however database writer process would be slow, and needs check point processes to do the work. In this case, it becomes a load on the server and memory. Because memory works quickly while waiting for the server to work, memory waits until the database writer process is implemented. Throughout this process, some problems may arise with the archiving process and log processes which consist of three files; each file has an area of fifty Megabytes, while these hundred thousand records need more than fifty Megabytes. So, the server enters the process until the 50-megabyte is filled and it is confirmed. The server works block by block then writes it in the hard drive, thus it continues until the last block and waits for its writing on the hard drive from database writer processes. We conclude that when executing this command, a load burdens the server. Although the insert sentence is a normal sentence, repeating it a thousand times causes a decrease in server performance. As such, we must be configuring the database and the server in order to be able to absorb the prob-


Figure 7: Killing session causing low performance.

lem of overloading with large sizes or other similar problems. We mentioned previously that ADDM technology could uncover the bottleneck and gives a recommendation for solving the issue. Practically, we go to the EM and at the performance tab, we note that this padding sentence exhausted large volumes of resources allocation and that the transaction number exceeded the 8,000 operations per second. To solve this problem, we use ADDM technology. Note that we take a snapshot and analyse the current performance. For experiment results in the (test_adv) table, the ADDM suggests solutions to improve the server performance as below:
1) Host Configuration: Meaning the server needs a configuration that gives advice which is that we have to go to the memory and increase its space manually. Another tip is having enlarged the System Global Area space.
2) The other problem is that there is in the select sentence and a tip is to enable SQL Tuning Advisor.
3) Redo log file, increase their size from 50 MB to 2 GB.
Now to work with these recommendations, we stop this terrible insert sentence and note that there is session id = 131. It takes 78% of its server work, therefore we must stop this session to improve the server performance; see the code in Figure 7.

## 5 Conclusions

In summary, this study has presented a thorough examination of performance monitoring within the context of database systems, with a particular focus on highlighting the significance of ensuring accuracy and trustworthiness throughout the monitoring procedure. The implementation of streamlined technologies, such as AWR and ADDM, within Oracle Database exemplifies the company's dedication to providing a reliable and dependable solution. The experimental findings provide clear evidence that optimizing the database has the potential to enhance planning performance and decrease response time for end users. Additionally, it fosters trust by minimizing resource utilization for comparable actions. The automated functionalities of ADDM in Oracle Database not only streamline the process of identifying performance metrics, but also offer trustworthy tuning recommendations with the goal of optimizing the overall throughput of the database. ADDM ensures a reliable and trustworthy approach to improving database performance by providing actionable suggestions to address bottlenecks.

## References

[1] K. Kamaraj, B. Lanitha, S. Karthic, P. N. S. Prakash, R. Mahaveerakannan, A Hybridized Artificial Neural Network for Automated Software Test Oracle. Computer Systems Science and Engineering, **45,** no. 2, (2023), 1837–1850.

[2] K. Dias, M. Ramacher, U. Shaft, V. Venkataramani, G. Wood, Automatic Performance Diagnosis and Tuning in Oracle, Conference on Innovative Data Systems Research, (2005), 84–94.

[3] B. Dageville, K. Dias, Oracle's Self-Tuning Architecture and Solutions, IEEE Data(Base) Engineering Bulletin, **29,** (2006), 24–31.

[4] K. M. Dobies, Evaluation of Sql Performance Tuning Features in Oracle Database Software, (2013).

[5] A. Nanda, S. Feuerstein, Oracle PL/SQL for DBAs-security, scheduling, performance and more: covers through Oracle database 10g, release 2, (2005).

[6] H. Al-Breiki, M. H. Rehman, K. Salah, D. Svetinovic, Trustworthy Blockchain Oracles: Review, Comparison, and Open Research Challenges, **8**, (2020), 85675–85685.

[7] B. Li, Q. Pan, B. Liu, S. Di, J. Liu, J. Pei, J. Yi, B. Zhou, Trustworthy AI, From Principles to Practices, arXiv (Cornell University), (2021).

[8] A. S. Albahri, A. M. Duhaim, M. A. Fadhel, A. Alnoor, N. S. Baqer, L. Alzubaidi, O. S. Albahri, A. H. Alamoodi, J. Bai, A. Salhi, J. Santamaría, C. Ouyang, A. Gupta, Y. Gu, M. A. Deveci, Systematic Review of Trustworthy and Explainable Artificial Intelligence in Healthcare: Assessment of Quality, Bias Risk, and Data Fusion, Information Fusion, **96,** (2023), 156–191.

[9] D. Kaur, S. Uslu, K. J. Rittichier, A. Durresi, Trustworthy Artificial Intelligence: A Review, ACM Computing Surveys, **55,** no. 2, (2022), 1–38.

[10] S. Thiebes, S. Lins, Sunyaev, A. Trustworthy Artificial Intelligence. Electronic Markets, **31,** no. 2, (2020), 447–464.

[11] G. Caldarelli, Understanding the Blockchain Oracle Problem: A Call for Action, Information, **11,**, no. 11, (2020), 509.

[12] G. Caldarelli, C. Rossignoli, A. Zardini, Overcoming the Blockchain Oracle Problem in the Traceability of Non-Fungible Products, Sustainability, **12,** no. 6, (2020), 2391.

[13] M. A. Mansor, S. Sathasivam, Optimal Performance Evaluation Metrics For Satisfiability Logic Representation In Discrete Hopfield Neural Network, International Journal of Mathematics and Computer Science, **16,** no. 3, (2021), 963–976.

[14] R. B. Madhumala, T. Harshvardhan, C. Devarajaverma, A Reliable Frame Work for Virtual Machine Selection in Cloud Datacenter Using Particle Swarm Optimization, International Journal of Mathematics and Computer Science,**16,** no. 2, (2021), 677–685.