Journal of Advanced Engineering Trends
ISSN : 2682 - 2091

Vol. 44, No. 1, January 2025

Enhancing Quality Factors in Data Warehousing Through Study and Improvement

Ola M. Hashem[*,1], Kamel Rahouma[1], Nagwa S. Abd El Hameed[1]

[1]*Electrical Engineering Dep., Faculty of Engineering, Minia University, Minia, Egypt*
** Corresponding author(s) E-mail: eng.olamohamed89@gmail.com*

ARTICLE INFO

Article history:
Received: 15 September 2024
Accepted: 9 December 2024
Online: 25 February 2025

Keywords:
Data Warehouse (DW)
Optimization
Indexing

ABSTRACT

Managing Data Warehouses (DWs) presents increasing challenges, particularly in handling diverse query loads that affect overall performance. This paper aims to enhance the performance of overloaded data warehouses by improving key factors such as storage efficiency, retrieval speed, and query execution time. To optimize these areas, strategies such as indexing, data compression, materialized views, and partitioning are applied. Star Schema designs are employed in Data Modeling Optimization to improve performance in read-heavy environments. Columnstore indexes are used to expedite data retrieval, while Row-level and Page-level compression reduce storage requirements and improve I/O performance. Materialized Views precompute complex queries to eliminate processing overhead, and partitioning increases retrieval efficiency by dividing large tables based on key attributes. Using the Olympic Games Data Warehouse as a case study, the implementation of these techniques in Microsoft SQL Server 2022 demonstrates substantial improvements in storage performance, scalability, and query efficiency. These enhancements contribute to maximizing the utility of the data warehouse, improving data accessibility, reducing operational costs, and supporting continuous performance improvements through ongoing monitoring and optimization.

1. Introduction

The concept of a Data Warehouse (DW) was first introduced by Bill Inmon in 1990, where he described it as a "subject-oriented, integrated, time-variant, and non-volatile collection of data" aimed at supporting management's decision-making processes [1]. Unlike traditional relational databases that focus on transaction processing, data warehouses are specifically designed for querying and analysis. They typically store historical data derived from various transactional sources, but can also include information from other origins. By separating analytical workloads from transactional ones, DW enable organizations to consolidate and analyze data from multiple sources effectively [2]. A comprehensive DW environment is not limited to just a relational database; it also encompasses an extraction, transformation, and loading (ETL) solution, an online analytical processing (OLAP) engine, and various client analysis tools [3]. These components work together to manage the data collection process and deliver insights to business users.

The success and efficiency of a DW depend on several critical factors, including storage management, data retrieval time, and query execution speed. This research focuses on identifying and enhancing these key performance factors to improve overall data warehouse effectiveness. The study reviews and categorizes the various strategies that can be implemented to optimize these aspects [4]. Enhancing DW quality involves a multi-faceted approach, including improvements in data modeling, indexing, compression, and optimization techniques. In this research, we applied these strategies using the Olympic Games DW as a case study. Given the growing complexity and scale of the Olympic Games, effective data management and analysis are essential for optimizing performance and enabling strategic decision-making [5].

To improve the quality factors of DWs, we focused on several key strategies. Data modeling was optimized for performance and storage efficiency through the use of Star Schema designs, which are well-suited for read-heavy environments. Proper indexing was implemented to enhance query performance and storage efficiency by allowing faster data retrieval, particularly for frequently accessed data [6]. Data compression techniques, including Row-level and Page-level compression, were employed to reduce storage space and improve input/output performance, thereby speeding up data retrieval. Additionally, materialized views were utilized to precompute complex queries, reducing the overhead associated with query execution. Partitioning techniques were also applied to divide large tables into smaller, more manageable segments, improving query performance and scalability [7].

The structure of this study is organized as follows: Section 2 reviews related work on enhancing quality factors in DWs. Section 3 discusses the data and methods used in the study. Section 4 presents the results of our tests and Discussion, and Section 5 concludes Conclusions and Future Research.

2. Related work

Enhancing the quality factors in data warehousing has been a key focus of research over the years, with efforts spanning various phases of data warehousing, including planning, implementation, and maintenance. Ballou and Tayi (1999) introduced a conceptual framework that underscores the importance of assessing the current level of data quality and understanding the specific quality requirements of decision-making processes [5,8]. Their work highlights the necessity for a systematic approach to data quality, especially in complex environments where users have varying needs. The growing recognition of the critical role of data quality in effective decision-making and operational efficiency within DW environments is evident from these early contributions [9] .

In parallel with data quality improvements, significant research has been conducted to optimize data warehouse (DW) performance. Bischoff and Alexander (1997) laid a foundational groundwork by implementing techniques such as query optimization, caching, and parallel processing. These strategies aimed to reduce response times and enhance overall system responsiveness, which are crucial for improving user experience in data-intensive environments [10]. One notable advancement in flexible query performance is the development of a fuzzy indexing technique for relational database management systems (DBMS). This approach, evaluated across six different indexing techniques and two datasets, has shown promise in improving query flexibility and performance in complex database environments, thereby enhancing the overall efficiency of DW systems [11,12].

Further contributions to DW optimization include the introduction of a modular approach for forecasting query execution times in large data volumes. This analytical model, designed to predict execution times for database access, I/O operations, and various SQL operators, plays a crucial role in optimizing resource allocation and improving DW performance under heavy workloads [13]. Shin et al although data warehouses are acknowledged as a strategic resource for decision-makers, there has been a notable lack of academic research on data warehousing practices and the factors critical to its success [14].

Optimizing data warehouse (DW) systems has been a significant focus of research, particularly in predicting resource demands for large-scale data processing tasks. A structured method for estimating disk usage and CPU costs has been developed to better understand the resource requirements for complex queries, which is essential for designing and implementing more efficient DW systems. Prior studies have investigated techniques for predicting execution times and resource usage in database systems. For example, Bielecki and Śmiałek (2022) explored methods to estimate execution times, stressing the need for accurate resource demand forecasting before execution [15]. Their research contributes to better resource management in data-intensive environments, aligning with the broader goal of improving DW performance. Additionally, workload forecasting has been studied within the context of autonomous database management systems (DBMS). A method presented at SIGMOD 2018 proposed clustering queries based on their arrival patterns to predict future workload changes. This proactive approach to workload management helps optimize system performance under fluctuating conditions by anticipating resource demands [16]. Furthermore, shared disk architectures have been explored as a way to enhance data access in distributed systems. These architectures allow multiple processors to access a common set of disks, increasing data availability and enabling concurrent access. However, challenges such as I/O contention and bottlenecks remain, which have led researchers to explore various optimization strategies to mitigate these issues.

In the context of managing resource usage during highly concurrent workloads, DBSeer was introduced as a framework aimed at optimizing the performance of Online Transaction Processing (OLTP) systems. By effectively managing resources, DBSeer demonstrated significant improvements in system performance, particularly during peak load periods [17]. These studies collectively highlight the diverse approaches that have been explored to enhance DW performance and data quality. Building on this foundation, our research aims to further optimize key quality factors in DW systems, particularly focusing on data modeling, indexing, and compression techniques for query optimization. By integrating these strategies, we seek to improve the overall effectiveness and efficiency of DWs, ultimately supporting better decision-making and operational success [18,19].

3. Methodology

This methodology provides a structured, data-driven approach to improving the performance and quality of the Olympic DW. It focuses on utilizing traditional database optimization techniques to enhance system efficiency, ensuring the warehouse remains effective in handling large volumes of data while meeting performance expectations.

3.1. Data

The Olympic Data Warehouse represents a pivotal advancement in the management and analysis of extensive sports-related data, particularly focusing on the Olympic Games. This initiative aims to establish a comprehensive and centralized repository that consolidates historical results, athlete profiles, and real-time competition data. It significantly improves the accessibility and usability of Olympic data for diverse stakeholders, including athletes, international federations, and organizing committees. Initially launched by the Association for Summer Olympic International Federations (ASOIF) and now maintained by the International Olympic Committee (IOC) and the Olympic Channel, the project standardizes data across multiple sources, ensuring consistency and accuracy. Utilizing advanced technologies such as data lakes, analytics platforms, and cloud computing, the Olympic DW efficiently manages vast datasets from over 200 participating nations and multiple sporting disciplines. These detailed datasets, spanning past and present Olympic events, are vital for performance analysis, strategic decision-making, and long-term planning. Moreover, athletes benefit from direct access to their performance metrics, which allows them to optimize their training and competition strategies. The data warehouse enhances transparency and operational

efficiency within the Olympic ecosystem, ensuring that data is readily available for analysis, reporting, and future research. By offering a robust platform for data-driven insights, the Olympic DW plays a key role in the legacy and continuous evolution of the Olympic movement. The authors get Olympic dataset from Kaggle (Public Data Platform) which hosts several cleaned and processed Olympic datasets, such as 120 years of Olympic history: athletes and results that we use it as a case study [20].

3.2. Data Modeling

The Olympic Games produce vast data across numerous areas, including athlete information, competition outcomes, event schedules, and historical records. Collected from more than 200 participating nations, this data encompasses a broad spectrum of sports disciplines, making it extensive in both scope and variety. Beyond just competition results, Olympic data contains detailed athlete biographies, statistics from both current and previous games, as well as logistical data like venue locations and event timings. Given the complexity and sheer scale of this information, careful organization is essential to meet the needs of various stakeholders such as athletes, international sports federations, media outlets, and organizing committees. Efficient management and analysis of this data are vital for tracking performance, supporting media coverage, maintaining historical archives, and facilitating strategic planning for future Olympic events.

Converting Olympic Data into a dedicated DW is essential for efficiently managing and analyzing the complex information generated during the Olympic Games. Traditional data storage systems lack the capacity to handle the large volumes and diverse types of data with the required efficiency. A DW overcomes this limitation by integrating multiple data sources into a single, well-organized repository, making data more accessible and easier to manage. This structure facilitates advanced querying, detailed analysis, and the extraction of valuable insights. It ensures consistency across datasets, accelerates data retrieval, and enhances performance analysis. Centralizing the data not only improves decision-making for event organizers, coaches, and athletes but also increases transparency and operational efficiency in handling Olympic data. [21].

In this paper, we convert Olympic data into an Olympic Data Warehouse to address the challenges of overloaded data systems. Key quality factors, such as storage efficiency, retrieval speed, and query execution time, are optimized using strategies like indexing, data compression, materialized views, and partitioning. These techniques aim to improve the overall performance and usability of the DW.

- *Data Warehouse Schemas*

DW schemas are essential for structuring data to optimize storage, retrieval, and analysis. The choice of schema can significantly impact the efficiency of querying and data management. Three prominent types of schemas are the Star Schema, Snowflake Schema, and Galaxy Schema, each offering unique advantages and use cases.

Star Schema: The Star Schema is characterized by its simple, intuitive design where a central fact table is connected to multiple dimension tables. This schema facilitates straightforward and efficient querying by minimizing the number of joins between tables. It is particularly suited for environments where rapid data access and ease of understanding are critical, making it ideal for reporting and business analysis [22,23].

Snowflake Schema: The Snowflake Schema builds on the Star Schema by further normalizing dimension tables into multiple related sub-tables. This approach reduces data redundancy and improves data integrity but may lead to more complex queries due to the increased number of joins. It is advantageous in scenarios where data consistency and storage efficiency are prioritized [24].

Galaxy Schema: The Galaxy Schema, also known as the Fact Constellation Schema, involves multiple fact tables sharing common dimension tables. This schema supports complex, multi-dimensional analysis by linking various business processes through shared dimensions. It is beneficial for organizations with diverse analytical needs, providing a comprehensive view of data while maintaining consistency and reducing redundancy across fact tables [25].

Finally, the Star Schema is characterized by its straightforward, user-friendly design, featuring a central fact table connected to multiple dimension tables, forming a star-like pattern. In our study, we employ the Star Schema due to its suitability for our data and query patterns. This schema is particularly effective in read-heavy environments where performance and simplicity are paramount. In this configuration, the Fact Edition serves as the central fact table, while the surrounding Dimension tables—Dim Athlete, Dim Date, Dim Country, and Dim Result—provide contextual information. The Star Schema's simplicity in design allows for efficient querying and data retrieval, minimizing the number of joins needed and thus improving overall query performance. This makes it an ideal choice for scenarios requiring fast access to large volumes of data, as it aligns well with our analytical needs and the nature of our data. Illustration of the Star Schema of the Olympic games dataset (As shown in “Figure 1”), with the central Fact Edition table connected to Dimension tables including Dim Athlete, Dim Date, Dim Country, and Dim Result [25].

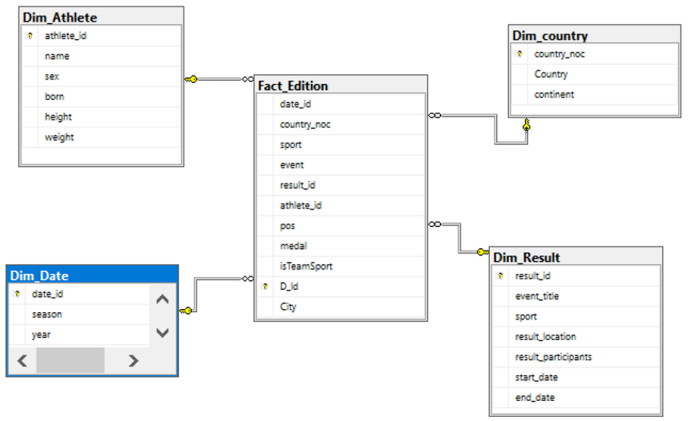

Figure 1: the star schema of the Olympic games dataset

3.3. Improve Practices for Managing Olympic Data Warehouses

To optimize the management and performance of the Olympic DW, several key practices are implemented. These practices address critical aspects such as indexing, data compression, data partitioning, and materialized views, each playing a significant role in enhancing the efficiency and effectiveness of data handling and retrieval. The following subsections detail these strategies and their contributions to improving the overall functionality of the DW.

Indexing

Proper indexing is crucial for optimizing the performance and efficiency of the Olympic Data Warehouse. Effective indexing strategies play a significant role in enhancing query performance, optimizing storage usage, and mitigating issues related to slow query execution. To achieve this, We employ CLUSTERED INDEX on columns of "FACT_EDITION" table that are frequently queried. These columns like **(**date_id) column which It is used a lot in filter conditions, such as (WHERE d. year = 2016), **(**athlete_id) column which used JOIN clause to join with (Dim_Athlete) table, and **(**result_id) column to join with (Dim_result) table. Clustered indexes are instrumental in determining the physical order of data within a table, which reduces the amount of data that needs to be scanned and accelerates retrieval times. To maintain peak performance, it is essential to regularly monitor and rebuild indexes to prevent fragmentation, which can impair performance over time. By periodically reorganizing and updating indexes, we ensure that the data remains well-structured and accessible, thereby supporting more efficient query processing and improving overall system performance [26]**.**

Data Compression

Data compression techniques play a crucial role in optimizing performance and reducing storage needs within a data warehouse environment. To enhance both the efficiency and effectiveness of the Olympic DW, various compression methods are employed. Specifically, Row-level compression, which suitable for reducing space in transactional systems or detailed data tables in Olympic DW by minimizing storage space for individual rows by removing Unused Space and storing only the actual data length. And Optimizing storage by using fewer bytes for NULL and zero values. and Page-level compression which works at the data page level and applies advanced techniques like: Dictionary Compression which Identifies repeated values across rows in a page and replaces them with references to a dictionary. And Column Prefixing ,which stores common prefixes of column values once and uses references. Page-level compression is ideal for large, repetitive datasets like fact tables or aggregated data in data warehouses, providing greater compression than row-level techniques. Row-level compression technique is applied to the "FACT_EDITION" table to reduce the space of fixed columns such as medal, pos, and isTeamSport and "Dim_Date" table to compress NULL values or zeros also it is applied on "Dim_Result " to save space for fixed-length columns such as event names. While Page-level compression is applied to the "FACT_EDITION" table to compress repetitive data such as sport, country_noc, and city and "Dim_Athlete" table to compress athlete names that have similar prefixes or repeated values such as gender. Also, it is applied on" Dim_Country" table to compress frequent country codes and names. These methods effectively reduce the volume of data that must be read from or written to storage, thereby accelerating query responses and decreasing overall storage costs. Additionally, Columnar Storage Compression is utilized to address the needs of analytical workloads. This technique organizes data by columns rather than rows, which not only improves storage space efficiency but also significantly boosts query performance. However, it is important to acknowledge that this method may slightly increase data loading times. To maximize the benefits of compression, a Column store Index is created on the "FACT_EDITION" table. This index further leverages the advantages of columnar compression by enhancing data retrieval processes and overall system efficiency. By integrating these compression strategies, the Olympic DW achieves superior performance and cost-efficiency, making it well-suited to handle the complex and voluminous data associated with the Games [27].

Data partitioning

Data partitioning is a valuable strategy for efficiently managing extensive datasets by segmenting them into smaller, more manageable sections. In the context of the Olympic Data Warehouse, Range Partitioning is employed on the "FACT_EDITION" table. This technique involves horizontally partitioning the data based on specific ranges, such as dates or event categories. By doing so, it minimizes the volume of data that needs to be scanned during queries, thereby enhancing query performance and optimizing data management. The partitioning approach facilitates more precise and efficient data retrieval, which is particularly beneficial for handling large and complex datasets. Through this method, the warehouse can handle extensive data more effectively, improve overall performance, and streamline data management processes [28].

Materialized Views

Materialized views are a crucial technique for enhancing query performance within the Olympic DW. By precomputing and storing the results of complex queries, materialized views significantly reduce the time required for data retrieval, as they eliminate the need to recalculate results with each query execution. This optimization not only speeds up the query process but also reduces the computational load, making resource utilization more efficient. In the context of the Olympic Data Warehouse, We create a materialized view "v_athlete_results" by merging and joining some of the tables of Olympic DW such as " Fact_edition" table which contains detailed data about events, medals, and athletes, "Dim_Athlete" table which contains athlete data, such as names and IDs and "Dim_Result" table which contains details of events, such as event names and type of sport. In order to convert a view to a Materialized View, a Clustered

index "idx_v_athlete_result "must be applied to it to ensure that the data is unique and ordered, which improves query performance. This view stores pre-computed data, which alleviates the computational burden during query processing and contributes to the overall efficiency of the system [29].

These strategies, including indexing, data compression, data partitioning, and materialized views, are designed to address common challenges in data warehousing, particularly in complex environments like the Olympic DW. By carefully monitoring and continually adjusting these practices, the warehouse can maintain optimal performance and effectively manage its evolving data management needs. This approach ensures that the system remains responsive and efficient, capable of handling the extensive and diverse datasets associated with the Olympic Games.

4. Results and Discussion

4.1. Results

This section details the outcomes of our study aimed at enhancing the storage efficiency and performance of the Olympic DW. We implemented a series of optimization techniques and assessed their effectiveness through comprehensive performance metrics and data quality analysis. Our results indicate notable improvements in various aspects of data management. First, by using Eq. (1) the data volume experienced a reduction of 23.86%, decreasing from 64,208 KB to 48,888 KB. This reduction demonstrates that the applied modifications significantly enhanced data compression, thereby lowering space consumption and optimizing storage resources.

$$\textit{Optimization Ratio of Data Volume} = 100\left(\frac{DB-DA}{DA}\right), \quad (1)$$

where DB and DA are Data volume optimization before and Data volume optimization after, respectively.

Conversely, the index volume saw a substantial increase of 206.64% by using Eq. (2), growing from 1,944 KB to 5,960 KB. While this may seem counterintuitive, the increase in index volume reflects the creation of additional indexes, which were crucial for enhancing query performance. Despite the added space required for these indexes, the improvement in query execution efficiency justifies this trade-off.

$$\textit{Optimization ratio of index volume} = 100\left(\frac{IB-IA}{IB}\right) \quad (2)$$

where IB and IA indicate to the index volume before optimization and the index volume after optimization, respectively.

In terms of reserved space, we observed a reduction of 15.55% by using Eq. (3), with the reserved volume decreasing from 68,912 KB to 58,184 KB. This decrease indicates that the space allocated for data and indexes has been optimized, contributing positively to resource management and overall storage efficiency.

$$\textit{Optimization ratio of data reserved volume} = 100\left(\frac{RB-RA}{RB}\right) \quad (3)$$

where RB and RA are data reserved volume before optimization and data reserved volume after optimization, respectively.

Furthermore, the execution time for queries was significantly reduced, from 1.16 seconds to 0.70 seconds, marking an average improvement of 39.66% by using Eq. (4). This reduction underscores the effectiveness of the optimization techniques in accelerating query processing, thus enhancing the responsiveness of the data warehouse. Summarizing the results of the optimization study (As shown in "Table 1") are illustrated:

$$\textit{Optimization ratio of Execution time} = 100\left(\frac{TB-TA}{TB}\right) \quad (4)$$

where TB and TA are Execution time before optimization and Execution time after optimization, respectively.

In our study, we ran 15 queries in two stages: queries to measure basic performance before applying any improvements and queries to apply the improvements, measure their effectiveness and test their effectiveness, with a variety of query types to cover all aspects of performance. We evaluate and compute this enhancement by using four types of queries, such as:

1. *Simple Queries* which show direct data about athletes who won medals.
2. *Complex Queries* which include JOIN operations and aggregations such as calculating the number of medals for each athlete.
3. *Temporal Queries* which focus on specific time periods using specific dates.
4. *Storage Queries* which measure storage size before and after improvements.

The data that the queries deal with covers athlete names, athlete_id, city names and sport types, events, event_title, medals, and dates.

4.2. Discussion

The findings from this study clearly show that implementing a combination of optimization techniques has a substantial impact on improving the performance of the Olympic DW. One of the most significant outcomes is the marked reduction in both data volume and total reserved storage space. By applying techniques such as data compression, partitioning, and indexing, the data warehouse became more efficient in its storage management, freeing up valuable resources and ensuring that the system operates at a higher capacity.

An important observation from the results is the increase in index volume, which, while initially appearing as a trade-off, proved beneficial. The larger index volume enhanced the organization and accessibility of the data, ultimately contributing to faster query responses. The increase in index volume is thus a positive effect of the optimization techniques, as it directly supports the overall improvement in data retrieval processes.

Most notably, the optimization techniques resulted in a significant reduction in query execution time, which is critical for DW performance (As shown in “Figure 2”). Faster query execution leads to better user experience, more efficient data analysis, and quicker decision-making processes, especially in data-heavy environments like the Olympic DW. The reduction in

execution time is a strong indicator of the effectiveness of these strategies in improving the overall system performance.

Table 1: The results of the optimization study

Metric	Before Optimization	After Optimization	Difference	Impact
Data Volume	**64208 Kbyte**	**48888 Kbyte**	**-23.86%**	**Data volume reduced, indicating better compression and space saving.**
Index Volume	**1944 Kbyte**	**5960 Kbyte**	**+206.64%**	**Index volume increased due to additional indexes, improving query performance despite the larger index volume**
Reserved Volume	**68912 Kbyte**	**58184 Kbyte**	**-15.55%**	**Reserved volume reduced, leading to more efficient resource management.**
Execution Time	**1.16 sec**	**0.7 sec**	**-39.66%**	**Query execution time decreased, resulting in significantly faster query performance.**

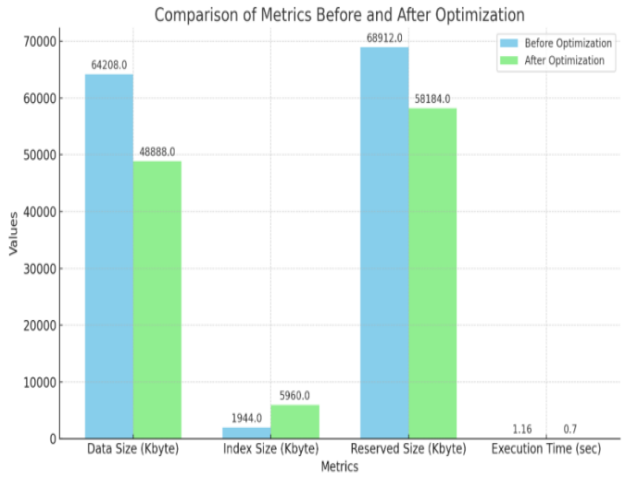

Figure 2: System optimization in the Olympic Data Warehouse

It is important to note, however, that these results are based on a hypothetical DW and a specific dataset. In practice, the actual results may vary depending on factors such as the size of the dataset, the nature of the DW, and the particular optimization methods employed. Different implementations may experience varying levels of success based on their unique characteristics and the challenges they face.

Future research should continue to refine these optimization techniques, with attention to specific warehouse environments and real-world datasets. As data continues to grow in both size and complexity, maintaining optimal performance will require ongoing adjustments and advancements in data management strategies. The techniques applied here offer a strong foundation for future developments in DW optimization.

Figure 3 illustrates the comparative CPU utilization before and after optimization. The graph clearly depicts a substantial decrease in CPU usage following optimization. Prior to optimization, CPU utilization was measured at 140%, as shown by the red bar, indicating a high level of resource consumption. In contrast, the green bar represents CPU utilization after optimization, which has significantly decreased to 60%. This notable reduction underscores the effectiveness of the optimization efforts in improving system performance. The y-axis of the chart is scaled from 0% to 150% to provide a clear view of the change, with exact utilization percentages annotated above each bar. This visual representation highlights the impact of the optimization process on enhancing efficiency and reducing resource overhead.

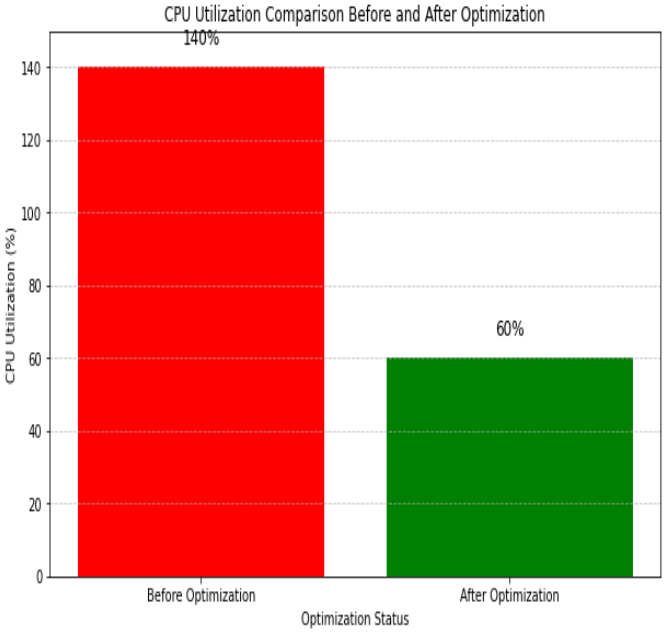

Figure3: Comparative CPU utilization before and after optimization.

5. Conclusions and Future Research

A combination of techniques such as columnar storage, data compression, partitioning, indexing, and materialized views can lead to substantial performance gains, improved storage efficiency, enhanced data quality, and greater scalability. Implementing these methods in the Olympic DW results in faster query execution, better data accessibility, and reduced operational costs. However, continuous monitoring and analysis are critical to

ensure the chosen strategies remain optimal as data volumes increase and workloads evolve.

The integration of machine learning (ML) techniques, such as the Random Forest Regression Algorithm, into query performance optimization presents a powerful and adaptive solution for enhancing database efficiency. ML enables predictive modeling, automated optimizations, and real-time monitoring, which not only improve query execution times and resource utilization but also lower costs and enhance scalability. This makes ML an essential tool for organizations looking to optimize their data systems in fast-paced, data-intensive environments.

Looking ahead, future research should focus on emerging trends in DW optimization. This includes exploring the use of artificial intelligence and machine learning to automate data quality processes and boost analytical capabilities. Additionally, studies that assess the long-term impacts of data warehouse initiatives on organizational performance and competitive advantage will provide further insight into the strategic value of DW, ensuring that these systems continue to evolve in ways that support informed decision-making and operational success.

Conflict of Interest

The authors declare no conflict of interest.

References

1. Inmon, W. H. *Building the Data Warehouse*. John Wiley & Sons, 2005.
2. Kempe S. A short history of data warehousing. DATAVERSITY. DATAVERSITY Education. 2012.
3. Ahmad, I., Azhar, S., & Lukauskis, P. Development of a decision support system using data warehousing to assist builders/developers in site selection. Autom. Constr. 13(4), 525-542 (2004).
4. Vaisman, A., & Zimányi, E. Data warehouse systems. Data-Centric Syst. Appl., 9 (2014).
5. Ballou, D. P., & Tayi, G. K. Enhancing data quality in data warehouse environments. Commun. ACM, 42(1), 73-78 (1999).
6. Bogojevic, P. Project management in data warehouse implementations: a literature review. IEEE Access, 8, 225902-2234 (2020).
7. Jung, R., & Winter, R. Justification of data warehousing projects. In Data warehousing and web engineering (pp. 219-228). IGI Global (2002).
8. Zellal, N., & Zaouia, A. An examination of factors influencing the quality of data in a data warehouse. In IJCSNS International Journal of Computer Science and Network Security (pp. 161-169). 2017.
9. Singh, R., & Singh, K. A descriptive classification of causes of data quality problems in data warehousing. In International Journal of Computer Science Issues (IJCSI) (pp. 41). 2010.
10. Bischoff, Joyce, and Ted Alexander. Data warehouse: Practical advice from the experts. Upper Saddle River, NJ: Prentice Hall, 1997
11. Gour, V., Sarangdevot, S. S., & Tanwar, G. S. Performance tuning mechanisms for data warehouse: query cache. In International Journal of Computer Applications (pp. 975-8887). 2010.
12. Jarke, M., Lenzerini, M., Vassiliou, Y., & Vassiliadis, P. Fundamentals of data warehouses. In Springer Science & Business Media. 2013.
13. Singhal, R., & Nambiar, M. Predicting SQL query execution time for large data volume. In Proceedings of the 20th International Database Engineering & Applications Symposium (pp. 378-385). 2016.
14. Shin, B. An exploratory investigation of system success factors in data warehousing. In Journal of the Association for Information Systems (pp. 6). 2003.
15. Bielecki, J., & Śmiałek, M. Estimation of execution time for computing tasks. Cluster Computing, 26(6), 3943-3956. 2023.
16. Kolajo, T., Daramola, O., & Adebiyi, A. Big data stream analysis: a systematic literature review. Journal of Big Data, 6(1), 47. 2019.
17. Mozafari, B., Curino, C., Jindal, A., & Madden, S. Performance and resource modeling in highly-concurrent OLTP workloads. In Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data (pp. 301-312). 2013.
18. Ma, L., Van Aken, D., Hefny, A., Mezerhane, G., Pavlo, A., & Gordon, G. J. Query-based workload forecasting for self-driving database management systems. In Proceedings of the 2018 International Conference on Management of Data (pp. 631-645). 2018.
19. Huang, K., Li, X., Yuan, M., & Zhang, J. Robust auto-scaling with probabilistic workload forecasting for cloud databases. In Proceedings of the IEEE 40th International Conference on Data Engineering (ICDE). 2024.
20. https://www.kaggle.com/datasets/heesoo37/120-years-of-olympic-history-athletes-and-results, last accessed in 12:48 PM -11 Sept.2024.
21. Hang H, Tang X, Sun J, Bao L, Lo D, Wang H. Robust Auto-Scaling with Probabilistic Workload Forecasting for Cloud Databases. In2024 IEEE 40th International Conference on Data Engineering (ICDE) 2024 May 13 (pp. 4016-4029). IEEE.
22. O'Neil, P., O'Neil, E., Chen, X., & Revilak, S. The star schema benchmark and augmented fact table indexing. In Performance Evaluation and Benchmarking: First TPC Technology Conference, TPCTC 2009, Lyon, France, August 24-28, 2009, Revised Selected Papers 1 (pp. 237-252). Springer Berlin Heidelberg. 2009.
23. Sanchez, J. A review of the star schema benchmark. arXiv preprint arXiv:1606.00295. 2016.
24. Iqbal, M. Z., Mustafa, G., Sarwar, N., Wajid, S. H., Nasir, J., & Siddique, S. A review of star schema and snowflake schema. In Intelligent Technologies and Applications: Second International Conference, INTAP 2019, Bahawalpur, Pakistan, November 6–8, 2019, Revised Selected Papers 2 (pp. 129-140). Springer Singapore. 2020.
25. Başaran, B. P. A comparison of data warehouse design models. Atılım Üniversitesi. 2005.
26. Jarke, M., Lenzerini, M., Vassiliou, Y., & Vassiliadis, P. Fundamentals of data warehouses. Springer Science & Business Media. 2013.
27. He, Y., Lee, R., Huai, Y., Shao, Z., Jain, N., Zhang, X., & Xu, Z. RCFile: A fast and space-efficient data placement structure in MapReduce-based warehouse systems. In 2011 IEEE 27th International Conference on Data Engineering (pp. 1199-1208). IEEE. 2011.
28. Haneen, A. A., Noraziah, A., Gupta, R., & Fakherldin, M. A. Review on data partitioning strategies in big data environment. Advanced Science Letters, 23(11), 11101-11104. 2017.
29. Gosain, A., Sabharwal, S., & Gupta, R. Architecture-based materialized view evolution: a review. Procedia Computer Science, 48, 256-262. 2015.