

Data Warehouse Workload Management: A Review of Prediction, Optimization, and Resource Allocation Techniques

Dr. Abid Hussain

Professor

School of Computer Application & Technology & Dean, Research

Career Point University

Kota

abid.hussain@cpur.edu.in

Corresponding Author*Dr. Abid Hussain (abid.hussain@cpur.edu.in)

Received: 30 July 2026 | Revised: 10 August 2026 | Accepted: 14 August 2026

Abstract—Data warehouse workload management has become increasingly important due to the growing scale, complexity, and variability of analytical workloads. Different query patterns, varying arrival rates and execution times, huge amounts of data and different resource usage creates challenges to maintaining performance and efficient resource usage. This survey is an overview of the recent technologies used to deal with the workload of data warehouses, focusing on workload prediction, workload optimization and resource allocation. Machine learning, deep learning, and advanced temporal models-based approaches to workload prediction are discussed for workload pattern identification and workload forecasting. The review also covers query optimization, workload scheduling, task prioritization, fragmentation, and workload prediction techniques for optimization. Also studied are resource allocation strategies such as dynamic scaling, workload-aware resource management strategies and proactive allocation of computing resources in cloud-based data warehouse. Survey focuses on the main approaches and their characteristics, including their objectives, techniques, benefits, limitations, and applicability to dynamic analytical environments. Besides, prediction accuracy, workload variation, scalability, adaptability, computational overhead and cost-efficient resource utilization are discussed. Finally, new research opportunities are identified for more intelligent, adaptive, and integrated workload management frameworks, which involve prediction, optimization, and resource allocation to provide efficient data warehouse operation.

Keywords—Data Warehouse, Workload Management, Workload Prediction, Query Optimization, Resource Allocation, Resource Scheduling.

I. INTRODUCTION

These growing size and complexity of analytical applications have generated a variety of workloads such as ad hoc queries, concurrent queries, batch operations, and resource-heavy analytical workloads. These workloads have

fluctuations in query arrival frequencies, execution times, data volume and resource usage, which causes significant problems in maintaining system performance and efficient resource usage [1].

Basic workload management methods include workload monitoring and policies, pre-defined scheduling policies, query optimization, and a static allocation of resources. However, these approaches may become ineffective when workload characteristics change dynamically or exhibit bursty and unpredictable behavior. For this reason, workload prediction has become a significant tool to estimate future workload based on the past query and execution data. Machine learning and deep learning methods can detect the temporal patterns and workload characteristics for the support of proactive management [2][3].

The predicted workload information can also be used for query optimization, scheduling, task prioritization, and dynamic resource allocation; this can further optimize system utilization. Elastic resources add to the opportunities for workload-aware scaling and adaptive resource management in cloud-based data warehouses [4]. However, accurate prediction, efficient optimization, scalability, adaptability and cost-efficient resource allocation still pose challenges.

This survey examines the techniques for workload management on data warehouse systems focusing on workload prediction, workload optimization, and resource allocation, along with their methods, advantages, drawbacks, and lines of future research.

A. Structure of the paper

The paper is organized as follows. Section II discusses workload prediction techniques. Section III reviews workload optimization approaches. Section IV presents resource allocation and adaptive workload management. Section V concludes the paper and outlines future directions.

II. DATA WAREHOUSE WORKLOAD PREDICTION TECHNIQUES

Data warehouse workload prediction techniques estimate resource requirements from historical workload patterns to support efficient resource allocation. Machine-learning approaches can predict query or job resource consumption, while time-series models forecast warehouse-level utilization [5]. However, in capacity-billed and session-isolated data warehouses, limited pre-execution information, delayed telemetry, and session-level resource assignment can restrict the effectiveness of conventional prediction methods. The reviewed study therefore emphasizes historical job-identity profiling, using past resource consumption to estimate a safe resource envelope for recurring ETL workloads.

A. Workload Characteristics and Prediction Requirements

Data warehouse workloads are characterized by varying query structures, data volumes, resource demands, and execution behavior [6]. These characteristics must be captured to enable accurate workload prediction. Workload patterns may also change over time and can exhibit bursty or highly variable resource demands, making historical workload information important for forecasting as illustrated in Fig. 1.

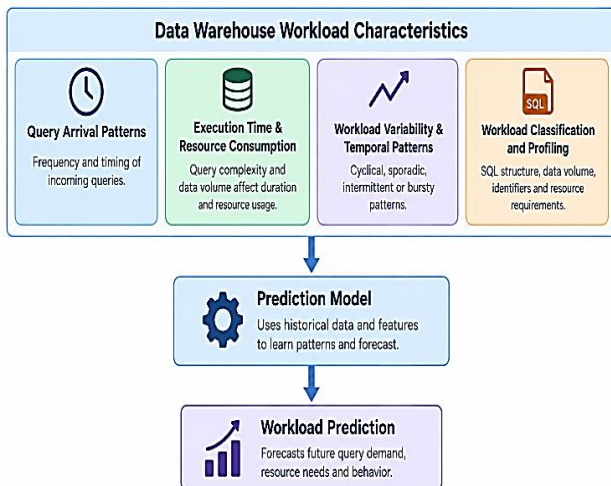


Fig. 1. Data Warehouse Workload Characteristics

Workload characteristics is as follows:

- **Query Arrival Patterns:** Frequency and timing of incoming queries influence future workload demand [7].
- **Execution Time and Resource Consumption:** Query complexity and processed data volume affect execution duration and resource usage.
- **Workload Variability and Temporal Patterns:** Workloads can be cyclical, sporadic, intermittent, or bursty, requiring models to capture temporal changes.
- **Workload Classification and Profiling:** Queries can be profiled using SQL structure, data volume, workload identifiers, and resource requirements to improve prediction.

Overall, effective prediction requires historical data, appropriate data granularity, relevant workload features, and models capable of adapting to workload variability and burstiness.

B. Traditional Workload Prediction Methods

Traditional workload prediction methods mainly use statistical time-series models to forecast future workload from historical observations. Common approaches include Autoregressive (AR), Moving Average (MA), Autoregressive Moving Average (ARMA), and Autoregressive Integrated Moving Average (ARIMA) models. These methods model temporal dependencies and are valued for their simple structure and interpretability. ARIMA-based approaches have also been used for proactive resource allocation, while exponential smoothing has been applied to improve workload scheduling and elastic scaling.

The Key methods are as follows:

- **AR:** Predicts future workload using previous workload observations.
- **MA:** Uses previous forecasting errors to estimate future workload.
- **ARMA:** Combines AR and MA components for time-series prediction.
- **ARIMA:** Extends ARMA by handling non-stationary series through differencing.
- **Exponential Smoothing:** Gives greater importance to recent observations for workload forecasting.

C. Machine Learning-Based Workload Prediction

ML methods improve workload prediction by learning nonlinear relationships between workload features and future resource demand. Unlike traditional statistical models, these approaches can capture more complex workload behavior. Common methods include Support Vector Regression (SVR), Decision Trees, and Random Forests [8]. SVR can model nonlinear relationships through kernel functions, while Random Forest combines multiple decision trees and aggregates their predictions. These models can be trained using historical workload features and adjusted according to changing workload conditions.

Key approaches:

- **Supervised Learning:** Uses historical workload data with known target values to predict future workload or resource demand.
- **Workload Classification:** Machine-learning models can classify workload behavior into different patterns when categorical workload states are defined.
- **Decision Trees and Ensemble Methods:** Random Forest combines multiple decision trees to improve prediction stability and capture nonlinear patterns.
- **Support Vector Machines and Regression:** SVR predicts continuous workload values and can handle nonlinear relationships using kernel functions.

D. Deep Learning-Based Workload Prediction

The ability of DL techniques to learn complex nonlinear patterns and temporal relationships from historical workload data makes them a popular method for cloud workload prediction [9]. RNNs, specifically LSTM and GRU, are useful for sequential workload forecasting and can learn the relationship between past and future workload use for resources. LSTM is able to store important information for a long period of time in memory cells and through gating mechanisms, whereas GRU is a simpler, more computationally efficient approach. CNNs can capture local temporal patterns and hybrid CNN-LSTM models can

integrate local feature extraction with long-term sequence learning. These methods help to accurately predict CPU, memory, and other resource needs, which in turn allows for preemptive resource provisioning, autoscaling, scheduling, and capacity planning.

E. Intelligent and Adaptive Workload Prediction

Intelligent workload prediction requires identifying workload patterns that change in length, amplitude, and shape over time. Traditional methods are not very effective when the workload is non-stationary, or the workload is very large and needs a lot of manpower to participate in work [10]. The reviewed approach provides a workload generic workload pattern and periodicity detection technique that can incorporate a variety of workload types of time-series data, such as CPU utilization and throughput. It adopts a digital-print extraction and a sliding-window method for pattern comparison to detect short-term periodic patterns and long-term periodic patterns. This allows workload behavior change to be detected, and may help with provisioning resources and identifying abnormal behavior. The proposed method is intended to be an offline method, but the generic pattern-detection mechanism has potential to be adapted to an online cloud method with suitable modifications.

III. DATA WAREHOUSE WORKLOAD OPTIMIZATION TECHNIQUES

A data warehouse is designed to accommodate a variety of data types: handling the variety of workloads is difficult, such as batch processing, incremental data loads, reports, and ad hoc queries. By integrating together query execution-time prediction, workload scheduling, admission control and resource management, automated workload optimization can enhance performance. The method discussed in this paper is based on the prediction of execution-time ranges based on historical query and system-load data using PQR (Predicting Query Run-time) models [11]. These are the strategies that the BI Batch Manager employs for workload control to prevent system overload and thrashing: Memory-based admission control, Priority Gradient Multiprogramming (PGM) and Largest Memory Priority (LMP) scheduling. Experimental tests conducted on enterprise data warehouses demonstrated a high throughput when this approach was used and that the approach was robust to workload resource estimation errors.

A. Query Workload Optimization

One of the goals of query workload optimization is to boost the speed and cost-effectiveness of multiple queries running in parallel in a cloud data warehouse. The reviewed approach considers the problem of workload optimization as a Mult objective problem, where the response time of the queries and monetary cost are minimized at the same time. It takes advantage of cloud elasticity by dynamically allocating appropriate VM configurations, network bandwidth, alternative query execution plans based on workload needs [12]. Three methods of optimization are compared: Simple Heuristic Algorithm (SHA), Multiobjective Branch-and-Bound (MOBB), and Multiobjective Genetic Algorithm (MOGA). The results indicate that MOBB can obtain optimal Pareto solutions, whereas MOGA can get near optimal solutions with much less optimization time. Therefore, resource allocation and alternative query plans can be used together to achieve flexible performance and cloud resource usage costs.

B. Query Execution and Scheduling Optimization

Query execution and scheduling optimization: It optimizes data warehouse performance by deciding when, where and in what order to run queries or subqueries. Workload characteristics and available CPU, memory, disk, and network resources can be utilized in scheduling strategies to balance system load and minimize response time. The order of queries is determined by the criteria of execution cost, resource requirements, priority and waiting time. Workload-aware execution dynamically tunes the scheduling and resource allocation based on the workload and system contention. The parallel query processing enables partitioning of large queries into independent tasks which can be processed concurrently on multiple processing nodes, thereby optimizing resource usage and query processing time as illustrated in Fig. 2.

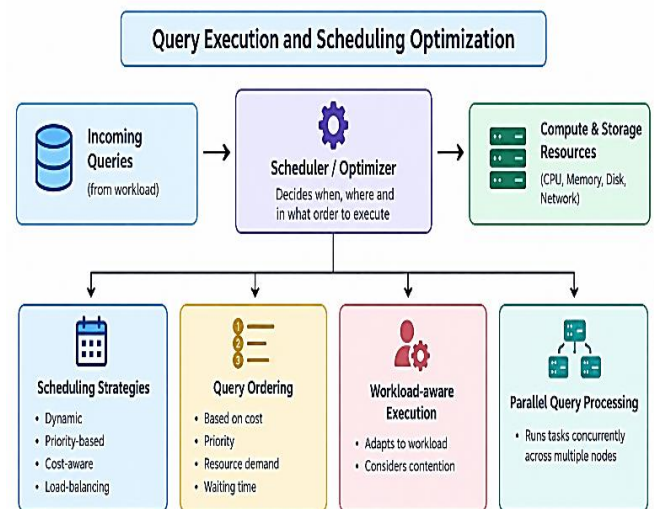


Fig. 2. Query Execution and Scheduling Optimization

Key aspects:

- **Scheduling strategies:** Dynamic, priority, cost-aware and load-balancing.
- **Query ordering:** Sequence in which queries are executed, depending on query costs, priorities, resources needed, or waiting time.
- **Workload-aware execution:** Makes decisions about execution based on workload intensity and resource contention.
- **Parallel query processing:** Runs independent query tasks on different nodes at the same time to speed up response.

C. Machine Learning-Based Workload Optimization

Machine learning based workload optimization leverages historical query execution data to forecast query performance and assist in making improved optimization decisions. The learning model can be used to predict query execution time based on query-plan and operator-level features, which can aid in predictive query optimization and in making more effective workload-aware decisions [13]. Plan-level models are capable of higher accuracy for stable workloads, whereas operator-level and hybrid models are more adaptable to changing workloads. The following techniques can be used to determine proper execution plans and dynamically update optimization decisions as workloads change.

D. Reinforcement Learning (RL) for Workload Optimization

The essence of RL is to learn the decisions made during query execution, interacting with the DBMS environment to optimize workload. The RL QOptimizer employs Q-learning and Deep Q-Networks (DQN) to learn efficient join orders from the actual query execution time as the reward [14]. This approach allows flexible decisions on how to execute a query in ways that minimize query latency, and adaptive query optimization based on feedback from queries that have been executed. The policy learned can create efficient execution plans with almost constant planning time and can generalize itself to previously unseen query states, which makes RL an appropriate approach for adaptive and policy-based workload optimization.

E. Resource-Aware Workload Optimization

Resource-aware workload optimization takes into account both the execution plan of the query and the computing resources to optimize workload performance and minimize execution cost. Jindal et al. proposed Resource and Query Optimization (RAQO), where query plans should be selected based on resource configurations as the number of containers, container size, memory, or cluster availability can significantly impact query performance. They found that, if they are not taken into account, resource conditions can lead to slower execution and monetary costs. RAQO thus combines query planning and resource planning, and takes the current cluster state into account in the selection of an execution plan [15]. The methodology contains rule-based resource-aware optimization based on decision trees and cost-based optimization based on regression models, hill climbing resource planning and resource plan caching. It allows workloads to adjust resource allocation based on data characteristics, performance needs, and resource availability, maximizing the efficiency of workload execution in shared cloud environments.

F. Cloud and Distributed Data Warehouse Optimizations

Cloud-based distributed data warehouses require optimization of query response time, monetary cost, and resource utilization. Cloud virtualization allows for flexible provisioning of CPU, memory, I/O and network resources based on demand. Alternative query plans and virtual-machine configurations can thus be considered together in the context of optimization, for efficient execution. The goal of the optimization problem is to find Pareto optimal solutions that offer a reasonably balanced combination of performance and cost. Various query plans can require varying amounts of resources; for instance, some can work well with increased network bandwidth, whereas others may need more memory or processing. These alternatives can be examined using multiple-objective optimization methods like branch-and-bound and genetic algorithms to find the cost-effective deployment configuration. Experimental results indicate that selecting resources based on workload characteristics can yield better cost-performance tradeoffs than fixed resources or uniformly provisioned resources.

IV. RESOURCE ALLOCATION AND ADAPTIVE WORKLOAD MANAGEMENT

Effective resource allocation is essential for managing dynamic cloud workloads while maintaining QoS, SLA compliance, and efficient resource utilization. The discussed approach is based on a model that employs a queuing theory

to dynamically match the workload requirements to computing resources. It takes into account the unpredictable workload variations, heterogeneous resources and the real-time resource demands to prevent over-provisioning and under-provisioning [16]. VM allocation is dynamic in nature; depending on workload conditions it can be adjusted to provide better response time, utilization, scalability, and efficiency. The model also takes into account energy use and operational expense in resource management. It was tested with JMT and CloudSim simulations under normal and fault-tolerant conditions and was verified on AWS Cloud, which yielded more or less consistent performance of the dynamic allocation strategy and managed the response times well. So, adaptive resource allocation offers a useful approach to meeting varying loads, while achieving performance, cost, energy and service quality.

A. Resource Allocation in Data Warehouses

Cloud data warehouse resource allocation refers to the process of dynamically managing the virtual machines' CPU, memory, network bandwidth, storage, and I/O resources based on workload demands [17]. Dökeroğlu et al. presented a Mult objective approach that takes into account virtual resource allocation, alternative query plans and materialized views together to minimize query response time and monetary cost. Different VM configurations can be selected according to workload requirements, exploiting cloud resource elasticity. The method also takes into account storage, processing and data transferring expenses and delivers Pareto optimal solutions that offer various performance-cost trade-offs as illustrated in Fig. 3.

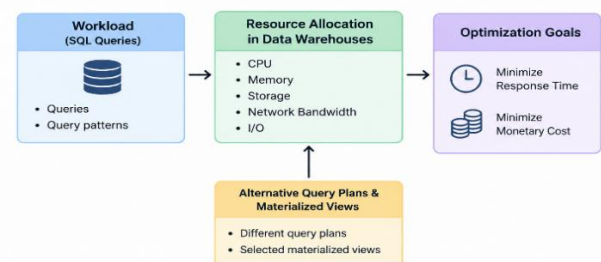


Fig. 3. Resource allocation and Mult objective optimization in cloud data warehouses.

B. Static and Dynamic Resource Allocation

Static resource allocation is assigning resources to cloud applications in advance of execution [18]. A static configuration is hard to define and can lead to over or under provisioning, however, in multi-tier applications with very dynamic workloads. Dynamic resource allocation eliminates this problem by continually monitoring the workload and response times, identifying bottlenecks, and adding more resources to the overloaded tiers. The system can also determine when resources are no longer needed and scale the resource back to minimize using resources and reduce costs. Therefore, dynamic allocation offers greater flexibility, adherence to SLA and resource efficiency compared to static allocation in evolving cloud workloads.

C. AI-Based Resource Allocation

Resource Allocation with AI-Driven Optimization: This involves dynamically allocating and optimizing cloud

resources using machine learning techniques to meet evolving workload requirements. The reinforcement learning is widely used due to its adaptability and deep learning and hybrid approaches aid multi-objective optimization [19]. The main methods are aimed at minimizing latency, optimizing the use of resources, lowering the operational cost and energy efficiency. AI-powered methods can continuously adjust resource allocation and autoscaling decisions to ensure performance and SLA compliance. Research also demonstrates that AI-driven approaches can achieve better resource allocation results than traditional approaches, but these are typically evaluated in simulation environments, not in practical deployments.

D. Resource Scheduling and Workload Isolation

Resource scheduling assigns computing resources like CPU and I/O capacity based on the needs and priorities of transactional and analytical workloads. Good allocation can minimize contention for resources and prevent any one workload from heavily impacting the performance of another workload [20]. Explored resource isolation with CPU allocation, I/OPS limits and I/O weights for different transaction and analytical consumers. They found that by allocating resources with workload awareness, isolation can be enhanced and performance interference can be reduced. The throughput frontier can also be leveraged to measure workload interference, and a composite measure can be created that integrates throughput, data freshness, and workload priorities to measure overall performance.

E. Elastic Resource Management in Cloud Data Warehouses

Cloud data warehouses can dynamically allocate resources as needed, based on workload, resource availability and performance goals, with elastic resource management. Elastic allocation techniques are resilient to workload variations by scaling up or down CPU, memory, storage, and network resources based on demand [21]. This reviewed resource management research emphasizes the need of adaptive allocation, workload migration, resource utilization, scalability and latency-aware scheduling in distributed environments. In the case of a cloud data warehouse, this can be done to any cloud data warehouses as well, where the number of queries and the types of workloads vary and resources need to be dynamically provisioned and reallocated. There are also many different goals that can be considered in resource management, such as resource usage, energy consumption, latency, and system performance. Thus, elastic resource management can support performance in case of changes in workload and prevent over-provisioning the resources and their resulting expenses.

V. LITERATURE REVIEW

The reviewed studies explore AI-driven data warehouse workload management through workload prediction, optimization, scheduling, fragmentation, and resource allocation. Machine learning, deep learning, foundation models, and predictive scaling improve efficiency, while challenges remain in adaptability, interpretability, scalability, and real-time management.

Mate and Mate (2026) The paper proposes a predictive optimization framework in which metadata and historical execution information are collected, processed, and supplied to machine learning models for workload forecasting.

Predictions are subsequently used to support dynamic scheduling, resource allocation, task prioritization, and pipeline configuration. A conceptual comparison of conventional and ML-enabled ETL approaches demonstrates the potential benefits of predictive optimization in improving resource utilization, throughput, latency, scalability, and operational efficiency [22].

Xie and Chen (2026) this paper proposes the MST-iTransformer model, which integrates multi-scale temporal encoding, sparse attention, and adaptive feature selection mechanisms. Specifically, a multi-scale temporal encoding module is developed to capture and fuse temporal dependencies across multiple periodic scales. Furthermore, an adaptive feature selection module is introduced to dynamically assign importance weights to resource features, enhancing informative variables while suppressing redundant ones. Meanwhile, a sparse attention mechanism is incorporated to reduce computational overhead while maintaining forecasting accuracy [23].

Rodríguez-Mazahua et al. (2026) This study presents a scoping review conducted using Arksey and O'Malley's methodological framework and reported according to the PRISMA guidelines, covering 58 primary studies on horizontal fragmentation published from 2015 to 2025. analysis focuses on five aspects: (1) determining the main techniques used in horizontal fragmentation works for analytical repositories, (2) the classification of these studies, (3) the performance metrics considered when evaluating the horizontal fragmentation scheme, (4) the information type indexed by the repositories, and (5) the technologies most used by the approaches. Findings suggest that horizontal fragmentation is a good opportunity to improve the performance of analytical workloads in most cases [24].

Bang et al. (2025) This paper evaluates their practical potential for WLM through a focused case study on time series similarity. Present concrete use cases, characterize a real-world query arrival dataset from Microsoft Fabric Warehouse, and compare the foundational model MOMENT against conventional similarity methods. Findings reveal that while foundational models offer computational efficiency, they produce overly generalized similarities with limited interpretability compared to hand-engineered features. Identify key challenges and research directions needed to make foundational models practical for workload management [25].

Bansal and Kumar (2023) This paper proposes a deep learning-based approach that uses a combination of neural networks to analyze historical workload data and forecast future workloads. The model is evaluated using real existing datasets which are verified and collected by the real-time problems and the output of the proposed algorithm accurately predicts future workloads and improves the user satisfaction level and system efficiency [26].

Prepalli (2018) This paper investigates predictive workload optimization techniques for cloud-native data warehouses, with particular emphasis on Snowflake's multi-cluster shared-data architecture, which enables independent scaling of compute without data movement. Building on foundational research in column-oriented database systems, cloud resource autoscaling, and workload forecasting published prior to 2018, the study proposes a predictive optimization framework that integrates historical workload

analysis, query-pattern classification, and proactive compute scaling decisions [27].

The Table I summarizes recent research on data warehouse workload management, focusing on workload prediction, optimization, resource allocation, and adaptive

management. Existing approaches use machine learning, deep learning, foundation models, fragmentation, and predictive scaling, while highlighting challenges in adaptability, interpretability, scalability, and real-time management.

TABLE I. SUMMARY OF THE STUDY ON DATA WAREHOUSE WORKLOAD MANAGEMENT

Reference	Research Focus	Approach	Key Findings	Challenges and Limitations	Future Directions
Mate and Mate (2026)	Predictive workload optimization for ETL and data warehouse workflows	Collects metadata and historical execution information and applies ML-based workload forecasting. Predictions support dynamic scheduling, resource allocation, task prioritization, and pipeline configuration.	Predictive optimization can improve resource utilization, throughput, latency, scalability, and operational efficiency compared with conventional ETL approaches.	Reliance on historical execution and metadata; conceptual comparison may not fully capture real-world deployment complexity.	Develop more adaptive predictive models, real-time workload management, and integrated scheduling and resource-allocation mechanisms.
Xie and Chen (2026)	Deep-learning-based workload/resource forecasting	Proposes MST-iTransformer, combining multi-scale temporal encoding, sparse attention, and adaptive feature selection.	Captures temporal dependencies at multiple scales while reducing computational overhead and emphasizing informative resource features.	Forecasting complexity and dependence on appropriate resource features; maintaining prediction efficiency while modeling complex temporal patterns remains challenging.	Develop more efficient transformer-based forecasting, adaptive feature selection, and real-time workload/resource prediction.
Rodríguez-Mazahua et al. (2026)	Horizontal fragmentation techniques for analytical repositories	Conducts a PRISMA-based scoping review of 58 studies published from 2015–2025, analyzing techniques, classifications, performance measures, indexed information, and technologies.	Horizontal fragmentation is identified as a promising technique for improving analytical workload performance in many cases.	Fragmentation effectiveness depends on workload characteristics, repository structure, evaluation criteria, and technology choices.	Develop workload-aware fragmentation strategies and investigate adaptive fragmentation for changing analytical workloads.
Bang et al. (2025)	Foundation models for workload management (WLM)	Uses a real-world query-arrival dataset from Microsoft Fabric Warehouse and compares the MOMENT foundation model with conventional time-series similarity methods.	Foundation models provide computational efficiency but may produce generalized similarities with lower interpretability than hand-engineered features.	Limited interpretability and overly generalized similarity representations can restrict practical WLM applications.	Improve foundation-model adaptation to database workloads, interpretability, workload-specific representations, and practical WLM deployment.
Bansal and Kumar (2023)	Deep-learning-based workload prediction	Uses a combination of neural networks to learn from historical workload data and forecast future workloads using real datasets.	The proposed approach demonstrates the ability to predict future workloads and potentially improve system efficiency and user satisfaction.	Prediction depends on historical workload patterns and the quality/representativeness of workload data; the specific model generalizability may require further validation.	Explore more advanced deep-learning architectures, adaptive prediction, larger workload datasets, and integration with proactive resource management.
Parepalli (2018)	Predictive workload optimization for cloud-native data warehouses	Proposes a framework combining historical workload analysis, query-pattern classification, workload forecasting, and proactive compute scaling for Snowflake's multi-cluster architecture.	Predictive workload analysis can support proactive compute scaling and workload optimization in cloud-native data warehouses.	Cloud workload behavior is dynamic; prediction errors can affect scaling decisions, and the approach is closely associated with a particular cloud data warehouse architecture.	Develop cloud-independent predictive optimization, adaptive scaling, real-time workload prediction, and integration with broader resource-management strategies.

VI. CONCLUSION AND FUTURE WORK

Data warehouse workload management is one of the most critical aspects for keeping analytical processing efficient across a variety of workloads and conditions. This survey considered the current solutions in the area of workload prediction, workload optimization and resource allocation. Machine learning and deep learning methods offer the possibility of predicting workload behavior based on the information about the previous executions, optimization methods can help in the optimization of query processing,

scheduling, prioritization, and analytical workload performance. Resource Allocation techniques also help in dynamic scaling and workload-aware management, especially in the cloud-based Data Warehouse environment. The literature reviewed shows that a combination of prediction with optimization and resource allocation can offer a more proactive and adaptive workload management than traditional static approaches. But there are difficulties in workload variability that must be represented with high accuracy, prediction errors that need to be minimised, bursty workloads to manage, understanding the model, and

computational and operational costs to manage. Further studies are needed to address the problem of integrated frameworks to perform workload prediction, optimization and resource allocation in real time. More adaptive models need to be developed to adapt to the changing workload patterns in a minimal time. Other areas of research that could be explored in the future include explainable AI, foundation models, reinforcement learning, multi-tenant workload isolation, and cost-aware resource allocation. Additionally, realistic testing on a variety of data warehouse workloads and cloud environments is essential to demonstrate the scalability, resilience, and applicability of intelligent workload management methods.

- **Funding:** This research received no external funding.
- **Institutional Review Board Statement:** Not applicable.
- **Informed Consent Statement:** Not applicable.
- **Data Availability Statement:** The data supporting the findings of this study are available from the corresponding author upon reasonable request.
- **Conflicts of Interest:** The author declares no conflict of interest.

How to cite this article:

A. Hussain, "Data Warehouse Workload Management: A Review of Prediction, Optimization, and Resource Allocation Techniques," *Journal of Global Research in Multidisciplinary Studies (JGRMS)*, vol. 2, no. 8, pp. 18–24, Aug. 2026. Doi, XXXX

REFERENCES

- [1] Q. Li, J. Kawash, and R. Alhajj, "Database workload forecasting for hybrid transactional/analytical processing scenario," *J. Big Data*, vol. 13, no. 1, p. 93, 2026, doi: 10.1186/s40537-026-01424-6.
- [2] B. Raza, Y. J. Kumar, A. K. Malik, A. Anjum, and M. Faheem, "Performance prediction and adaptation for database management system workload using Case-Based Reasoning approach," *Inf. Syst.*, vol. 76, pp. 46–58, Jul. 2018, doi: 10.1016/j.is.2018.04.005.
- [3] D. Intelligence, "Database Workload Prediction Method Based on iForest-BiLSTM-Attention," 2022.
- [4] Z. Wu *et al.*, "Stage: Query Execution Time Prediction in Amazon Redshift," in *Companion of the 2024 International Conference on Management of Data*, New York, NY, USA: ACM, Jun. 2024, pp. 280–294, doi: 10.1145/3626246.3653391.
- [5] Z. Ke, "Right-Sizing Cloud Data Warehouses Under Asymmetric Loss via One-Sided Conformal Prediction," *IEEE Access*, vol. 14, pp. 94748–94763, 2026, doi: 10.1109/ACCESS.2026.3704918.
- [6] M. Kaviani, P. Asghari, and M. Sabeti, "A comprehensive survey and taxonomy of workload forecasting techniques in cloud computing," *Comput. Networks*, vol. 286, p. 112454, Aug. 2026, doi: 10.1016/j.comnet.2026.112454.
- [7] P. K. Pathak, "Pre-Execution Query Slot-Time Prediction in Cloud Data Warehouses: A Feature-Scoped Machine Learning Approach," 2026, doi: 10.48550/arXiv.2604.20145.
- [8] M. Nashaat, W. Moussa, R. Rizk, and W. Saber, "Dynamic machine learning approach for workload prediction in cloud environments," *Sci. Rep.*, vol. 16, no. 1, p. 10983, 2026, doi: 10.1038/s41598-026-40777-z.
- [9] E. Alshboul, H. Suleiman, and M. Alshboul, "An LSTM-Based Resource Prediction Model in Google Cloud Data Center," *Appl. Comput. Intell. Soft Comput.*, vol. 2026, no. 1, pp. 01–22, Jan. 2026, doi: 10.1155/acs/1160180.
- [10] C. St-Onge, N. Kara, O. A. Wahab, C. Edstrom, and Y. Lemieux, "Detection of time series patterns and periodicity of cloud computing workloads," *Futur. Gener. Comput. Syst.*, vol. 109, pp. 249–261, Aug. 2020, doi: 10.1016/j.future.2020.03.059.
- [11] A. Mehta, C. Gupta, S. Wang, and U. Dayal, "Automated Workload Management for Enterprise Data Warehouses," *IEEE Data Eng. Bull.*, vol. 31, pp. 11–19, 2008.
- [12] T. Dokeroglu, S. A. Sert, and M. S. Cinar, "Evolutionary Multiobjective Query Workload Optimization of Cloud Data Warehouses," *Sci. World J.*, vol. 2014, pp. 1–16, 2014, doi: 10.1155/2014/435254.
- [13] M. Akdere, U. Çetintemel, M. Riondato, E. Upfal, and S. B. Zdonik, "Learning-based Query Performance Modeling and Prediction," in *2012 IEEE 28th International Conference on Data Engineering*, IEEE, Apr. 2012, pp. 390–401, doi: 10.1109/ICDE.2012.64.
- [14] M. Ramadan, A. El-Kilany, H. M. O. Mokhtar, and I. Sobh, "RL_QOptimizer: A Reinforcement Learning Based Query Optimizer," *IEEE Access*, vol. 10, pp. 70502–70515, 2022, doi: 10.1109/ACCESS.2022.3187102.
- [15] A. Jindal, L. Viswanathan, and K. Karanasos, "Query and Resource Optimizations: A Case for Breaking the Wall in Big Data Systems," 2019, doi: 10.48550/arXiv.1906.06590.
- [16] O. Ghandour, S. El Kafhali, and M. Hanini, "Adaptive workload management in cloud computing for service level agreements compliance and resource optimization," *Comput. Electr. Eng.*, vol. 120, p. 109712, Dec. 2024, doi: 10.1016/j.compeleceng.2024.109712.
- [17] T. Dökeroğlu, S. A. Sert, M. S. Çinar, and A. Coşar, "Designing Cloud Data Warehouses using Multiobjective Evolutionary Algorithms," in *Proceedings of the 6th International Conference on Agents and Artificial Intelligence*, Esco, Angers, Loire Valley, France: SCITEPRESS - Science and Technology Publications, 2014, pp. 571–576, doi: 10.5220/0004906805710576.
- [18] W. Iqbal, M. N. Dailey, D. Carrera, and P. Janeczek, "Adaptive resource provisioning for read intensive multi-tier applications in the cloud," *Futur. Gener. Comput. Syst.*, vol. 27, no. 6, pp. 871–879, Jun. 2011, doi: 10.1016/j.future.2010.10.016.
- [19] S. Alam *et al.*, "AI-driven resource allocation in cloud computing: a systematic review revealing critical sustainability and evaluation gaps," *Computing*, vol. 108, pp. 01–26, 2026, doi: 10.1007/s00607-026-01655-8.
- [20] W. Wu *et al.*, "HTAP benchmark in financial scenarios," *Secur. Saf.*, vol. 4, p. 2025010, Jul. 2025, doi: 10.1051/sands/2025010.
- [21] N. Kaftantzis, D. G. Kogias, and C. Z. Patrikakis, "Exploring the Impact of Resource Management Strategies on Simulated Edge Cloud Performance: An Experimental Study," *Network*, vol. 4, no. 4, pp. 498–522, 2024, doi: 10.3390/network4040025.
- [22] J. Mate and Z. Mate, "Machine Learning-Based Workload Prediction for Self-Optimizing ETL Pipelines in Cloud Data Engineering," 2026.
- [23] X. Xie and J. Chen, "Cloud Resource Workload Forecasting Method Based on the MST-iTransformer Model," *Futur. Internet*, vol. 18, no. 9, 2026, doi: 10.3390/fi18090448.
- [24] N. Rodríguez-Mazahua, L. Rodríguez-Mazahua, G. Alor-Hernández, J. Cervantes, and F. Castro-Medina, "A Decade of Horizontal Fragmentation Methods in OLAP from Data Warehouse to Data Lakehouse: A Scoping Review," *Futur. Internet*, vol. 18, no. 3, 2026, doi: 10.3390/fi18030176.
- [25] T. Bang, I. Roumpos, S. Matushevych, G. Christofidi, Y. Tian, and T. D. Doudali, "Advancing Workload Management with Foundational Models: Challenges in Time Series Similarity and Interpretability," in *Proceedings of the 1st Workshop Connecting Academia and Industry on Modern Integrated Database and AI Systems*, in MIDAS '25, New York, NY, USA: Association for Computing Machinery, 2025, pp. 16–25, doi: 10.1145/3737412.3743491.
- [26] S. Bansal and M. Kumar, "Deep Learning-based Workload Prediction in Cloud Computing to Enhance the Performance," in *2023 Third International Conference on Secure Cyber Computing and Communication (ICSCCC)*, 2023, pp. 635–640, doi: 10.1109/ICSCCC58608.2023.10176790.
- [27] S. Parepalli, "Predictive Workload Optimization in Cloud Data Warehouses: Forecast-Driven Scaling for Elastic and Cost-Efficient Analytics," 2018, doi: 10.5281/zenodo.18084288.