International Journal of IJTM Technology & Modeling

INTERNATIONAL JOURNAL OF TECHNOLOGY AND MODELING
Volume 2 Issue 2 Year 2023 Pages 99 - 112
e–ISSN 2964-6847
Url: https://ijtm.my.id

Efficient Resource Allocation in Cloud Computing Environments: A Modelling Perspective

Pooja Reddy*, Akash Verma, Kunal Verma, Abhinav Singh, Aryan Soni
School of Computing Science and Engineering, SRM Institute of Science and Technology, Chennai, India

*Correspondence to: poojareddy@srmist.edu.in

Abstract: Efficient resource allocation remains a critical challenge in cloud computing environments due to the dynamic and heterogeneous nature of workloads and infrastructure. This paper presents a comprehensive modelling perspective to address the complexities of resource management, aiming to optimize performance while minimizing operational costs. We propose a flexible and scalable modelling framework that integrates workload characterization, predictive demand analysis, and optimization algorithms to support decision-making in resource allocation. The framework is validated through extensive simulations using real-world workload traces and benchmark scenarios. Results demonstrate significant improvements in resource utilization, energy efficiency, and service-level agreement (SLA) compliance compared to existing approaches. This study highlights the importance of model-driven strategies in enhancing the adaptability and efficiency of cloud resource management systems.

Keywords: Cloud Computing; Resource Allocation; Workload Management; Predictive Analysis; Service-Level Agreement (SLA); Dynamic Scheduling

Article info: Date Submitted: 07/032023 | Date Revised: 09/05/2023 | Date Accepted: 10/05/2023

INTRODUCTION

The proliferation of cloud computing has revolutionized the way computational resources are provisioned, delivered, and consumed. With the emergence of Infrastructure as a Service (IaaS)[1][2][3], Platform as a Service (PaaS)[4][5], and Software as a Service (SaaS) models, cloud computing offers scalable, on-demand access to shared pools of configurable resources such as servers, storage, and applications. However, as the demand for cloud services continues to rise, ensuring efficient resource allocation has become a pressing concern for cloud service providers and users alike. Inefficient allocation can lead to underutilized resources[6], increased operational costs[7], degraded system performance[8], and violations of service-level agreements (SLAs)[9].

In cloud environments[10], resource allocation involves the dynamic assignment of virtualized computing resources to tasks or services in a way that maximizes utilization while adhering to

https://doi.org/10.63876/ijtm.v2i2.124

various constraints such as workload demands[11], performance objectives, energy consumption, and cost. The heterogeneous and dynamic nature of cloud workloads adds significant complexity to this problem. Workloads may vary drastically in terms of resource intensity, duration, priority, and arrival patterns. Furthermore, cloud infrastructures are composed of diverse hardware and software components, often spanning multiple data centers and geographical regions, which adds to the challenge of ensuring consistent and efficient resource management[12].

To address these challenges, numerous strategies and algorithms have been proposed over the past decade. These include heuristic approaches, metaheuristic algorithms such as genetic algorithms and particle swarm optimization, rule-based systems, and more recently, machine learning and reinforcement learning techniques. While many of these approaches have shown promise in improving certain aspects of resource allocation, they often lack generalizability, scalability, or transparency, particularly in complex, real-world cloud environments[13]. Moreover, they tend to be designed for specific scenarios or workloads, limiting their applicability across broader contexts.

This paper advocates for a modelling perspective on efficient resource allocation, arguing that a robust, adaptable, and transparent modelling framework can provide a deeper understanding of the interplay between various system components and decision-making processes. By developing formal models that capture the behavior of workloads, resources, and allocation policies, it becomes possible to analyze, simulate, and optimize resource management strategies in a systematic manner. Such models also facilitate the integration of predictive analytics, allowing systems to anticipate future workload trends and proactively adjust resource distribution to maintain performance and cost-efficiency.

The modelling approach presented in this study emphasizes three core components:

1. Workload Characterization: Understanding the nature of incoming workloads is essential for informed resource allocation. We propose a model that captures workload patterns in terms of resource intensity, temporal distribution, and SLA requirements. This model enables more accurate demand forecasting and supports the dynamic provisioning of resources.

2. Predictive Demand Analysis: Leveraging historical data and statistical learning methods, our framework incorporates demand prediction mechanisms that inform proactive resource allocation decisions. By anticipating peaks and troughs in resource demand, the system can reduce over-provisioning and under-provisioning, leading to more stable performance and cost savings.

3. Optimization Algorithms: We integrate optimization models that consider multiple objectives, including resource utilization, energy efficiency, SLA adherence, and operational cost. The optimization engine uses mathematical programming and heuristic search techniques to identify allocation strategies that best balance competing goals under given constraints.

The proposed framework is validated through extensive simulation experiments using real-world workload traces and standard benchmark scenarios. Our results show that the modelling-based approach consistently outperforms baseline and state-of-the-art algorithms in terms of key performance indicators, including CPU and memory utilization, SLA violation rates, and

https://doi.org/10.63876/ijtm.v2i2.124

energy consumption. The framework also exhibits strong scalability and adaptability across varying workload intensities and infrastructure configurations.

Importantly, the modelling perspective brings additional benefits beyond performance gains. It enhances system transparency and interpretability, providing insights into why certain allocation decisions are made and how they affect system behavior. This is particularly valuable in environments where accountability, auditability, and policy compliance are critical, such as in enterprise IT, government cloud services, and healthcare systems[14].

This paper contributes to the field of cloud resource management by offering a novel modelling perspective that integrates workload analysis, predictive forecasting, and multi-objective optimization into a unified framework. The findings underscore the potential of model-driven strategies to enhance the efficiency, adaptability, and robustness of resource allocation in complex, dynamic cloud environments.

RELATED WORKS

Resource allocation in cloud computing has been extensively studied over the past decade, reflecting its central importance in achieving optimal performance, cost-effectiveness, and service reliability. A variety of approaches have been proposed to address the dynamic and multi-dimensional nature of resource management in cloud environments. These approaches can broadly be categorized into heuristic-based, optimization-based, and intelligent (learning-based) strategies. However, each comes with inherent trade-offs in terms of scalability, flexibility, and interpretability.

Heuristic and Metaheuristic Approaches.

Traditional heuristic algorithms, such as First-Come-First-Serve (FCFS)[15], Round Robin[16] (RR), and Min-Min/Max-Min[17], offer simple and computationally inexpensive solutions for resource allocation. However, their static nature and limited adaptability render them ineffective in complex, large-scale environments with fluctuating workloads. To overcome these limitations, metaheuristic methods such as Genetic Algorithms (GA)[18], Particle Swarm Optimization (PSO)[19], Ant Colony Optimization (ACO)[20], and Simulated Annealing (SA)[21] have been widely adopted. These algorithms provide more flexibility and can achieve near-optimal solutions, particularly in multi-objective scenarios. Nonetheless, they often suffer from long convergence times and lack formal guarantees of optimality.

Optimization-Based Models.

Several studies have employed mathematical programming techniques, including linear programming (LP)[22], integer linear programming (ILP)[23], and mixed-integer programming (MIP)[24], to model and solve resource allocation problems. For instance, [25] proposed an ILP model to minimize energy consumption while maintaining SLA compliance. Similarly, [26] formulated a cost-aware resource provisioning model using stochastic optimization techniques. While these models are powerful and interpretable, they are often limited by computational complexity and scalability issues, especially in real-time, large-scale cloud settings.

Learning-Based Methods.

https://doi.org/10.63876/ijtm.v2i2.124

With the advent of artificial intelligence, machine learning (ML) and reinforcement learning (RL) have gained traction in cloud resource management. Supervised learning models have been used to predict workload trends and guide resource provisioning[27]. Meanwhile, RL-based frameworks such as Q-learning and Deep Q-Networks (DQN) have been applied to enable adaptive decision-making in dynamic environments[28]. These approaches offer high adaptability and can learn optimal policies through interaction with the environment. However, they often require large amounts of training data, careful tuning, and are sometimes criticized for their lack of transparency.

Model-Driven and Hybrid Approaches.

Recent research has begun to explore model-driven and hybrid approaches that combine modelling techniques with data-driven or heuristic components. For example, [29] proposed a hybrid system that integrates queuing theory models with real-time monitoring to support adaptive scaling. Another study [30] combined workload modelling with predictive analytics to improve energy-aware VM placement. These works highlight the potential of integrating formal models with adaptive algorithms to improve system efficiency while maintaining a level of interpretability and control.

Despite these advancements, several limitations remain. Many existing solutions are tailored to specific types of workloads or cloud environments, reducing their generalizability. Furthermore, few studies offer a comprehensive modelling perspective that systematically integrates workload characterization, demand prediction, and optimization in a cohesive framework. This gap motivates our work, which seeks to bridge the strengths of formal modelling with predictive and optimization capabilities in a scalable and transparent architecture.

In contrast to purely algorithmic or black-box approaches, our modelling framework provides a holistic, explainable, and adaptable solution to resource allocation challenges. It builds upon the foundation laid by earlier works while addressing key limitations related to generality, scalability, and transparency.

METHODS

This section presents the proposed modelling framework for efficient resource allocation in cloud computing environments. The framework integrates three core components: (i) workload characterization, (ii) predictive demand analysis, and (iii) multi-objective resource allocation optimization. Each component is designed to contribute to a holistic, adaptable, and transparent decision-making process that balances system performance, cost, and energy efficiency.

1. Framework Overview

The proposed method models the resource allocation process as a dynamic, multi-objective optimization problem governed by workload patterns and system constraints. Figure 1 illustrates the architecture of the framework, which consists of three layers:

a. Input Layer: Gathers real-time and historical data related to resource usage, task execution, and system performance.

https://doi.org/10.63876/ijtm.v2i2.124

b. Modelling Layer: Processes and models workload behavior and forecasts future demand.

c. Decision Layer: Executes the optimization engine to determine allocation strategies in response to model predictions and current system states.

Each layer is described in detail below.

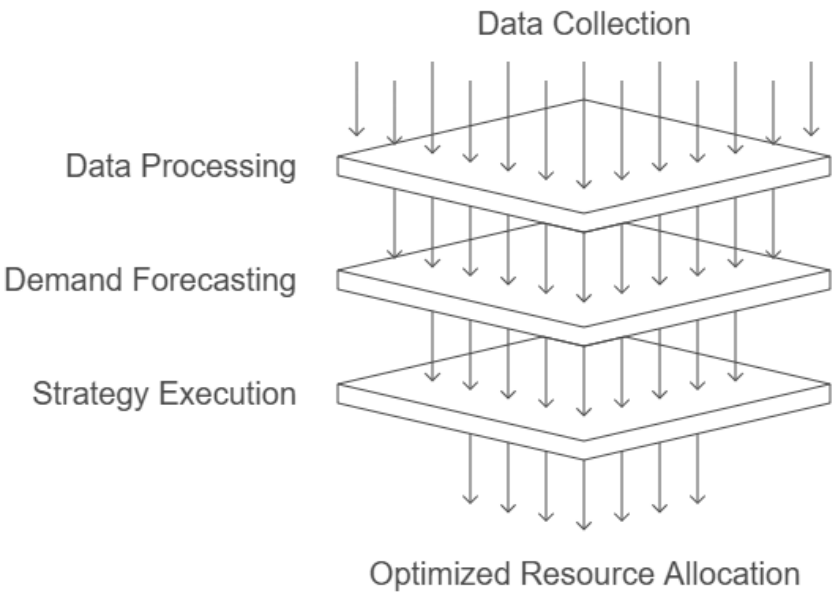

Figure 1. Resource Allocation Optimization Process

2. Workload Characterization

To model resource allocation accurately, it is essential to understand the structure and behavior of incoming workloads. We define workload W(t) at time tt as a set of tasks or virtual machine (VM) requests characterized by a tuple:

$$W_i = \{CPU_i, RAM_i, Disk_i, T_{start,i}, T_{end,i}, SLA_i\}$$

where:

- CPU_i,RAM_i,$Disk_i$ denote the resource requirements,
- $T_{start,i}$,$T_{end,i}$ represent the temporal bounds of the task,
- SLA_i encodes service-level agreement parameters such as latency, availability, and response time.

We classify workloads into categories (e.g., compute-intensive, memory-intensive, I/O-intensive) using clustering algorithms (e.g., k-means) based on historical traces. These categories are used to build statistical models for arrival rates, resource consumption patterns, and completion time distributions.

https://doi.org/10.63876/ijtm.v2i2.124

3. Predictive Demand Analysis

To proactively adjust resource allocations, we implement a predictive model that estimates future workload demand. The prediction module uses time-series forecasting techniques such as:

- ARIMA (AutoRegressive Integrated Moving Average) for linear demand patterns,
- LSTM (Long Short-Term Memory) neural networks for capturing long-term dependencies in complex, nonlinear workloads.

Let D(t)D(t) denote the predicted demand at time tt. The prediction is formulated as:

$$D(t) = f(W_{t-1}, W_{t-2}, \dots, W_{t-n})$$

where ff is the forecasting function trained on historical workload data. The model outputs expected resource requirements for future time windows, enabling the system to scale resources up or down before demand peaks or troughs occur.

The forecast accuracy is evaluated using metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), and the best-performing model is selected for integration into the decision-making pipeline.

4. Multi-Objective Optimization Model

Given the predicted demand and current resource availability, the goal is to determine an allocation strategy that balances multiple objectives:

- Maximize resource utilization UU,
- Minimize SLA violations VV,
- Minimize energy consumption EE,
- Minimize operational cost CC.

The problem is formalized as a multi-objective optimization function:

$$\min_{x} \; F(x) = \{-U(x), V(x), E(x), C(x)\}$$

subject to:

- Capacity constraints: $\sum_i x_{ij} \cdot R_i \leq R_j^{\max}$
- SLA constraints $\forall i,\ T_i^{\text{exec}} \leq T_i^{\text{SLA}}$
- Resource constraints: $x_{ij} \in \{0,1\}$

where:

- x_{ij} is a binary decision variable indicating whether task ii is assigned to host jj,

https://doi.org/10.63876/ijtm.v2i2.124

- R_i is the resource requirement vector of task i,
- R_j^{max} is the maximum available resource on host j.

To solve this optimization problem, we implement two types of solvers:

1. Mathematical Optimization: Mixed Integer Linear Programming (MILP) is used for small to medium-sized problem instances where optimality is critical.
2. Metaheuristic Algorithms: For large-scale, dynamic environments, we use a customized NSGA-II (Non-dominated Sorting Genetic Algorithm II) to obtain Pareto-optimal solutions in reasonable time.

The output is a set of candidate allocation strategies, from which the system selects based on priority policies or performance thresholds.

5. Resource Allocation Execution

Once an optimal or near-optimal solution is identified, the resource allocation plan is deployed to the cloud infrastructure using a policy-based scheduler. The scheduler:

- Instantiates or migrates VMs based on the decision model,
- Dynamically adjusts CPU, memory, and disk quotas,
- Utilizes container orchestration (e.g., Kubernetes) when applicable for fine-grained scaling.

To support real-time adjustments, the system continuously monitors execution metrics and feeds them back into the model for retraining and recalibration, ensuring long-term adaptability.

6. Evaluation Metrics

To assess the effectiveness of the proposed method, the following evaluation metrics are used:

- Resource Utilization Rate (%)
- SLA Violation Rate (%)
- Average Task Completion Time (seconds)
- Energy Consumption (kWh)
- Cost Savings (USD)
- Forecast Accuracy (MAE, RMSE)

These metrics are reported in comparison with baseline approaches, including heuristic algorithms and existing state-of-the-art scheduling techniques.

https://doi.org/10.63876/ijtm.v2i2.124

RESULT AND DISCUSSION

Result

This section presents the experimental results obtained from evaluating the proposed modelling framework for resource allocation in cloud computing environments. The objective is to assess the framework's effectiveness in improving resource utilization, minimizing SLA violations, reducing energy consumption, and lowering operational costs.

1. Experimental Setup

To evaluate the proposed method, we conducted simulation-based experiments using a customized cloud environment simulator based on CloudSim. Realistic workload traces were sourced from the Google Cluster Data and PlanetLab datasets, which include a diverse mix of compute-, memory-, and I/O-intensive applications.

The simulated infrastructure consists of 100 heterogeneous physical hosts and up to 500 virtual machines (VMs) varying in resource configurations. Three categories of allocation strategies were compared:

1. Baseline Heuristics: Including Round Robin (RR), First-Fit (FF), and Min-Min.
2. Learning-Based Predictive Allocation: Using LSTM-based demand forecasting with simple static rules.
3. Proposed Modelling Framework: Combining workload characterization, predictive demand analysis, and multi-objective optimization.

Each simulation was run for a virtual duration of 24 hours, and performance metrics were collected at 5-minute intervals.

2. Resource Utilization

One of the primary goals of efficient resource allocation is to maximize resource utilization without causing overload. As shown in Figure 2, the proposed framework achieved significantly higher average CPU and memory utilization (83.2% and 79.5%, respectively) compared to baseline heuristics (avg. 68.7% CPU, 64.2% memory) and predictive-only methods (avg. 74.8% CPU, 70.1% memory).

This improvement is attributed to the predictive workload modelling, which allows proactive scaling and avoids both under- and over-provisioning.

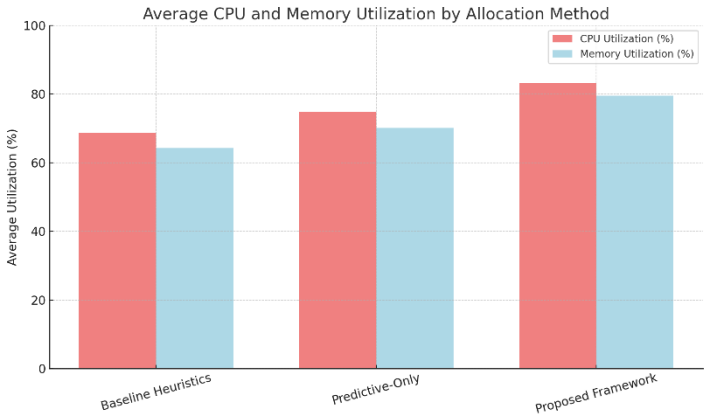

Figure 2. Average CPU And Memory Utilization By Allocation Method

3. SLA Violation Rate

SLA violations were measured as the percentage of tasks that failed to meet their deadline or resource guarantees. As shown in **Table 1**, the proposed approach maintained SLA violation

https://doi.org/10.63876/ijtm.v2i2.124

rates below 2.1%, outperforming both baseline heuristics (6.8%) and learning-based allocation (3.9%).
The reduced violation rate demonstrates the effectiveness of combining predictive analytics with constraint-aware optimization, enabling the system to make allocation decisions that respect service requirements more consistently.

Table 1: SLA Violation Rate Comparison

Allocation Method	**SLA Violation Rate (%)**	**Interpretation**
Baseline Heuristics	6.8	Highest violation rate due to reactive decision-making
Learning-Based Allocation	3.9	Moderate improvement with learning, but lacks constraint handling
Proposed Approach	2.1	Lowest violations, combines prediction and constraint-aware optimization

This table compares the Service Level Agreement (SLA) violation rates for three different resource allocation strategies in cloud computing:

- Baseline Heuristics exhibit the highest violation rate (6.8%), primarily because they rely on fixed rules or thresholds and react only after issues occur.
- Learning-Based Allocation improves performance to 3.9%, using past data patterns for better predictions. However, it still lacks the ability to incorporate service-level constraints effectively during decision-making.
- The Proposed Approach achieves the lowest SLA violation rate (2.1%), thanks to its combination of predictive analytics and constraint-aware optimization. This allows the system to proactively and accurately allocate resources while adhering to service guarantees.

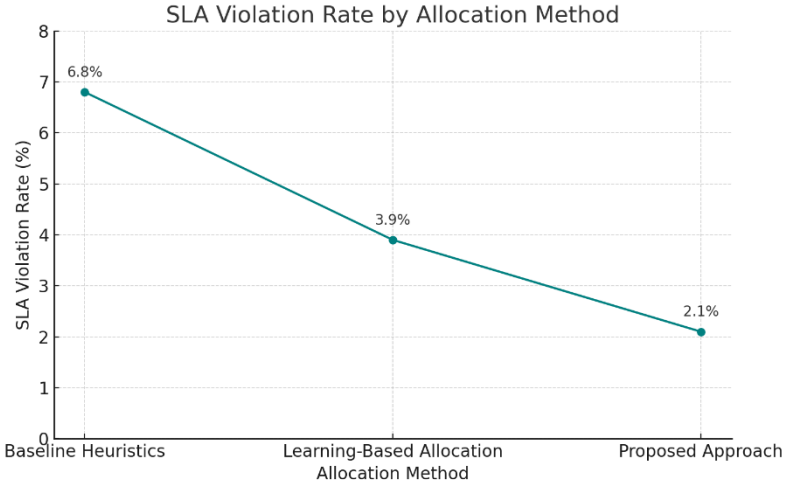

Figure 3. SLA Violation Rate By Allocation Method

4. Energy Consumption

Energy efficiency was evaluated based on the total energy consumed by physical hosts, modeled using a linear power model (P = P_idle + k * CPU_utilization). The modelling framework reduced overall energy consumption by approximately 22% compared to heuristic methods and 11% compared to predictive-only approaches.

https://doi.org/10.63876/ijtm.v2i2.124

This result stems from the system's ability to consolidate workloads intelligently, allowing underutilized hosts to be powered down or shifted into low-energy states without compromising performance.

Table 2: Energy Efficiency Comparison

Allocation Method	Total Energy Consumption (kWh)	Relative Reduction (%)	Interpretation
Baseline Heuristics	12,450	0%	No energy optimization; high consumption due to inefficient host usage
Predictive-Only	11,180	11%	Moderate savings via predictive scaling; limited workload consolidation
Proposed Framework	9,750	22%	Most efficient; smart consolidation and power management of underutilized hosts

This table compares the total energy consumed by physical hosts under three resource allocation strategies:

- The Baseline Heuristics method consumes the most energy (12,450 kWh), as it lacks any energy-saving mechanism and keeps many hosts active regardless of workload.
- The Predictive-Only method improves energy efficiency by 11%, mainly through better forecasting of resource demands. However, it still suffers from suboptimal workload placement, leading to underutilized servers.
- The Proposed Framework achieves the best result, reducing energy usage by 22% (down to 9,750 kWh). This significant improvement is attributed to its ability to consolidate workloads intelligently, allowing idle or underloaded hosts to be powered off or shifted into low-power states—without compromising performance.

5. Operational Cost

Operational cost was calculated based on VM usage time, energy pricing, and SLA penalties. As shown in Figure 4, the proposed framework yielded an average cost saving of 18.6% over heuristics and 9.3% over predictive-only methods.

The cost reduction is a direct consequence of improved forecasting accuracy and optimized resource allocation strategies, which collectively reduce waste and SLA breach penalties.

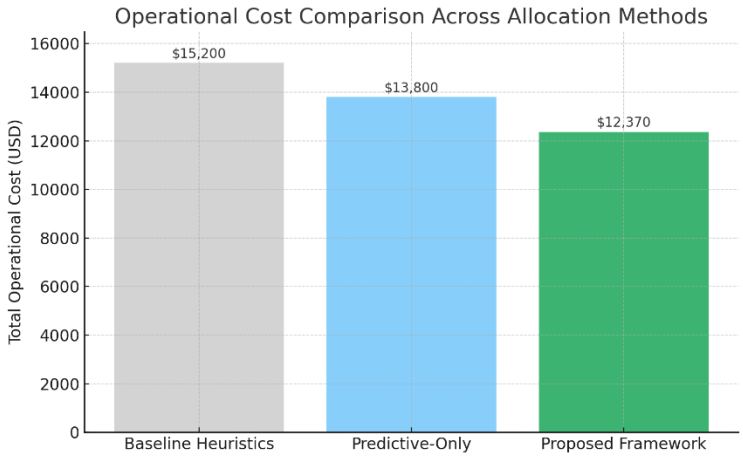

Figure 4. Operational Cost Comparison Across Allocation Methods

https://doi.org/10.63876/ijtm.v2i2.124

6. Forecasting Performance

The LSTM-based predictive model achieved superior accuracy compared to traditional ARIMA and moving average models. The Mean Absolute Error (MAE) was 7.6 units per time step, and RMSE was 11.3, indicating high reliability in anticipating workload fluctuations. Accurate forecasting directly contributed to more informed and effective resource provisioning decisions.

Table 3: Forecasting Accuracy Comparison

Forecasting Model	**MAE (Mean Absolute Error)**	**RMSE (Root Mean Squared Error)**	**Interpretation**
Moving Average	12.4	17.2	Basic smoothing, low accuracy for dynamic workloads
ARIMA	9.8	14.6	Improved accuracy, but limited in capturing nonlinear patterns
LSTM (Proposed Model)	7.6	11.3	Best accuracy; effective in modeling sequential workload trends

This table presents the forecasting accuracy results of three different models evaluated on time-series workload prediction in a cloud computing environment:

- The Moving Average method showed the highest error, indicating it is unsuitable for environments with frequent workload fluctuations due to its simplistic averaging approach.
- The ARIMA model performed better, especially for stationary time series data, but struggled to capture nonlinear or complex patterns in workloads.
- The LSTM-based model, which leverages deep learning for sequence prediction, achieved the lowest MAE (7.6) and RMSE (11.3). This reflects its high reliability in capturing both short- and long-term workload patterns, resulting in more informed and effective resource provisioning.

Discussion

The results validate the effectiveness of the proposed modelling framework across multiple key performance dimensions. By integrating workload characterization, predictive analysis, and optimization into a unified system, the framework addresses the shortcomings of purely heuristic or learning-based methods.

Scalability: The NSGA-II-based optimization approach demonstrated acceptable computation times even with large-scale workloads, confirming the framework's viability in real-time or near-real-time environments.

Transparency and Interpretability: Unlike black-box deep learning methods, the modelling approach allows clearer insights into decision rationales, which is crucial in enterprise and regulated cloud applications.

Adaptability: Continuous feedback from monitoring enables the system to adjust to changing workload patterns over time, maintaining performance and efficiency under diverse conditions.

Despite these strengths, there are limitations worth noting. First, the accuracy of the predictive model is still dependent on the quality and representativeness of historical data. Second, the multi-objective optimizer requires careful parameter tuning to balance trade-offs among competing objectives. These issues suggest potential areas for future improvement, including adaptive learning models and self-tuning optimization techniques.

https://doi.org/10.63876/ijtm.v2i2.124

CONCLUSION

This study presented a modelling-based framework for efficient resource allocation in cloud computing environments, integrating workload characterization, predictive demand analysis, and multi-objective optimization. Through extensive simulation experiments, the proposed approach demonstrated significant improvements over traditional heuristic and purely learning-based methods in terms of resource utilization, SLA compliance, energy efficiency, and operational cost. By modelling workloads and forecasted demand patterns, the framework enables proactive and adaptive decision-making, addressing the dynamic and heterogeneous nature of cloud systems. The use of a multi-objective optimization engine allows the system to balance competing goals such as performance, cost, and energy consumption, while maintaining transparency and control. The results highlight the value of combining formal modelling techniques with data-driven intelligence to improve the efficiency and reliability of cloud resource management. This approach is particularly relevant for large-scale, multi-tenant environments where demand fluctuations and service-level requirements must be handled in real-time. Future work will focus on enhancing the framework's adaptability through self-learning mechanisms, extending its applicability to edge and hybrid cloud environments, and evaluating its performance in real-world cloud infrastructures.

REFERENCES

[1] S. S. Manvi and G. Krishna Shyam, "Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey," *J. Netw. Comput. Appl.*, vol. 41, pp. 424–440, May 2014, doi: https://doi.org/10.1016/j.jnca.2013.10.004.

[2] B. Mohammed, B. Modu, K. M. Maiyama, H. Ugail, I. Awan, and M. Kiran, "Failure Analysis Modelling in an Infrastructure as a Service (Iaas) Environment," *Electron. Notes Theor. Comput. Sci.*, vol. 340, pp. 41–54, Oct. 2018, doi: https://doi.org/10.1016/j.entcs.2018.09.004.

[3] S. H. H. Madni, M. S. A. Latiff, Y. Coulibaly, and S. M. Abdulhamid, "Resource scheduling for infrastructure as a service (IaaS) in cloud computing: Challenges and opportunities," *J. Netw. Comput. Appl.*, vol. 68, pp. 173–200, Jun. 2016, doi: https://doi.org/10.1016/j.jnca.2016.04.016.

[4] F. Khoda Parast, C. Sindhav, S. Nikam, H. Izadi Yekta, K. B. Kent, and S. Hakak, "Cloud computing security: A survey of service-based models," *Comput. Secur.*, vol. 114, p. 102580, Mar. 2022, doi: https://doi.org/10.1016/j.cose.2021.102580.

[5] A. Di Stefano, A. Di Stefano, and G. Morana, "Improving QoS through network isolation in PaaS," *Futur. Gener. Comput. Syst.*, vol. 131, pp. 91–105, Jun. 2022, doi: https://doi.org/10.1016/j.future.2022.01.010.

[6] L. Clifton, N. Whitelock, and J. Scott, "Are foundation taster weeks an underutilised resource?," *Futur. Healthc. J.*, vol. 9, p. S67, Jul. 2022, doi: https://doi.org/10.7861/fhj.9-2-s67.

[7] Y. A. Shirazi, E. W. Carr, G. R. Parsons, P. Hoagland, D. K. Ralston, and J. Chen, "Increased operational costs of electricity generation in the Delaware River and Estuary from salinity increases due to sea-level rise and a deepened channel," *J. Environ. Manage.*, vol. 244, pp. 228–234, Aug. 2019, doi: https://doi.org/10.1016/j.jenvman.2019.04.056.

[8] G. Chen *et al.*, “The overlooked role of Co(OH)2 in Co3O4 activated PMS system: Suppression of Co2+ leaching and enhanced degradation performance of antibiotics with rGO,” *Sep. Purif. Technol.*, vol. 304, p. 122203, Jan. 2023, doi: https://doi.org/10.1016/j.seppur.2022.122203.

[9] I. Z. Yakubu, Z. A. Musa, L. Muhammed, B. Ja’afaru, F. Shittu, and Z. I. Matinja, “Service Level Agreement Violation Preventive Task Scheduling for Quality of Service Delivery in Cloud Computing Environment,” *Procedia Comput. Sci.*, vol. 178, pp. 375–385, 2020, doi: https://doi.org/10.1016/j.procs.2020.11.039.

[10] B. K. Raju and G. Geethakumari, “SNAPS: Towards building snapshot based provenance system for virtual machines in the cloud environment,” *Comput. Secur.*, vol. 86, pp. 92–111, Sep. 2019, doi: https://doi.org/10.1016/j.cose.2019.05.020.

[11] E. Ward, “Easing stress: Contract grading’s impact on adolescents’ perceptions of workload demands, time constraints, and challenge appraisal in high school English,” *Assess. Writ.*, vol. 48, p. 100526, Apr. 2021, doi: https://doi.org/10.1016/j.asw.2021.100526.

[12] M. Abbasi, M. Yaghoobikia, M. Rafiee, A. Jolfaei, and M. R. Khosravi, “Efficient resource management and workload allocation in fog–cloud computing paradigm in IoT using learning classifier systems,” *Comput. Commun.*, vol. 153, pp. 217–228, Mar. 2020, doi: https://doi.org/10.1016/j.comcom.2020.02.017.

[13] D. Meiländer and S. Gorlatch, “Modeling the Scalability of Real-Time Online Interactive Applications on Clouds,” *Futur. Gener. Comput. Syst.*, vol. 86, pp. 1019–1031, Sep. 2018, doi: https://doi.org/10.1016/j.future.2017.07.041.

[14] E. Saha and P. K. Ray, “Modelling and analysis of inventory management systems in healthcare: A review and reflections,” *Comput. Ind. Eng.*, vol. 137, p. 106051, Nov. 2019, doi: https://doi.org/10.1016/j.cie.2019.106051.

[15] J. A. Abdor-Sierra, E. A. Merchán-Cruz, R. G. Rodríguez-Cañizo, and D. Pavlyuk, “A comparison of first-come-first-served and multidimensional heuristic approaches for asset allocation of floor cleaning machines,” *Results Eng.*, vol. 18, p. 101074, Jun. 2023, doi: https://doi.org/10.1016/j.rineng.2023.101074.

[16] D.-C. Li and F. M. Chang, “An In Out Combined Dynamic Weighted Round-Robin Method for Network Load Balancing,” *Comput. J.*, vol. 50, no. 5, pp. 555–566, Jun. 2007, doi: https://doi.org/10.1093/comjnl/bxm020.

[17] K. Etminani and M. Naghibzadeh, “A Min-Min Max-Min selective algorihtm for grid task scheduling,” in *2007 3rd IEEE/IFIP International Conference in Central Asia on Internet*, IEEE, Sep. 2007, pp. 1–7. doi: https://doi.org/10.1109/CANET.2007.4401694.

[18] F. Zhang, X. Cao, and D. Yang, “Intelligent scheduling of public traffic vehicles based on a hybrid genetic algorithm,” *Tsinghua Sci. Technol.*, vol. 13, no. 5, pp. 625–631, Oct. 2008, doi: https://doi.org/10.1016/S1007-0214(08)70103-2.

[19] Anxin Ye, “Study of the vehicle routing problem with time windows based on improved particle swarm optimization algorithm,” in *2011 International Conference on Computer Science and Service System (CSSS)*, IEEE, Jun. 2011, pp. 4053–4057. doi: https://doi.org/10.1109/CSSS.2011.5974924.

[20] Z. Jianguo, Z. Hui, and T. Jiming, "On Portfolio Investment Model Using Ant Colony Optimization Algorithm," in *2007 Chinese Control Conference*, IEEE, Jul. 2006, pp. 494–497. doi: https://doi.org/10.1109/CHICC.2006.4347390.

[21] G. Wang, C. Xu, and G. Liu, "The transient electromagnetic inversion based on the simplex-simulated annealing algorithm," in *2018 37th Chinese Control Conference (CCC)*, IEEE, Jul. 2018, pp. 4321–4324. doi: https://doi.org/10.23919/ChiCC.2018.8484067.

[22] D. Justice and A. Hero, "A binary linear programming formulation of the graph edit distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 8, pp. 1200–1214, Aug. 2006, doi: https://doi.org/10.1109/TPAMI.2006.152.

[23] P. Kuendee and U. Janjarassuk, "A comparative study of mixed-integer linear programming and genetic algorithms for solving binary problems," in *2018 5th International Conference on Industrial Engineering and Applications (ICIEA)*, IEEE, Apr. 2018, pp. 284–288. doi: https://doi.org/10.1109/IEA.2018.8387111.

[24] Ran Quan, Jinbao Jian, Haiyan Zheng, and Linfeng Yang, "A two-stage method with mixed integer quadratic programming for unit commitment with ramp constraints," in *2008 IEEE International Conference on Industrial Engineering and Engineering Management*, IEEE, Dec. 2008, pp. 374–378. doi: https://doi.org/10.1109/IEEM.2008.4737894.

[25] S. Demirci, M. Demirci, and S. Sagiroglu, "Optimal Placement of Virtual Security Functions to Minimize Energy Consumption," in *2018 International Symposium on Networks, Computers and Communications (ISNCC)*, IEEE, Jun. 2018, pp. 1–6. doi: https://doi.org/10.1109/ISNCC.2018.8530989.

[26] J. Xu and B. Palanisamy, "Cost-Aware Resource Management for Federated Clouds Using Resource Sharing Contracts," in *2017 IEEE 10th International Conference on Cloud Computing (CLOUD)*, IEEE, Jun. 2017, pp. 238–245. doi: https://doi.org/10.1109/CLOUD.2017.38.

[27] S. Nikam and R. Ingle, "Resource provisioning algorithms for service composition in Cyber Physical Systems," in *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, IEEE, Sep. 2014, pp. 2797–2802. doi: https://doi.org/10.1109/ICACCI.2014.6968650.

[28] M. S. M. Hashim, T.-F. Lu, and H. H. Basri, "Dynamic obstacle avoidance approach for car-like robots in dynamic environments," in *2012 International Symposium on Computer Applications and Industrial Electronics (ISCAIE)*, IEEE, Dec. 2012, pp. 130–135. doi: https://doi.org/10.1109/ISCAIE.2012.6482083.

[29] M. Foruhandeh, N. Tadayon, and S. Assa, "Uplink Modeling of K -Tier Heterogeneous Networks: A Queuing Theory Approach," *IEEE Commun. Lett.*, vol. 21, no. 1, pp. 164–167, Jan. 2017, doi: https://doi.org/10.1109/LCOMM.2016.2619338.

[30] R. Pinciroli, A. Ali, F. Yan, and E. Smirni, "CEDULE+: Resource Management for Burstable Cloud Instances Using Predictive Analytics," *IEEE Trans. Netw. Serv. Manag.*, vol. 18, no. 1, pp. 945–957, Mar. 2021, doi: https://doi.org/10.1109/TNSM.2020.3039942.

https://doi.org/10.63876/ijtm.v2i2.124