Electronics, Communications, and Computing Summit
Vol. 1, No. 1, Oct - Dec 2023, pp. 20-28
DOI: https://doi.org/10.17051/ECC/01.01.03

Adaptive Resource Allocation in Cloud Data Centers: A Machine Learning-Driven Framework for Energy Efficiency and SLA Compliance

Prerna Dusi[1], Dr.Gaurav Tamrakar[2]

[1]Assistant Professor, Department of Information Technology, Kalinga University, Raipur, India,
Email: ku.PrernaDusi@kalingauniversity.ac.in
[2]Assistant Professor, Department of Mechanical, Kalinga University, Raipur, India,
Email: ku.gauravtamrakar@kalingauniversity.ac.in

Article Info

Article history:

Received : 13.10.2023
Revised : 18.11.2023
Accepted : 12.12.2023

Keywords:

Adaptive Resource Allocation,
Cloud Computing,
Machine Learning,
SLA Compliance,
Energy Efficiency,
Reinforcement Learning,
LSTM,
CloudSim

ABSTRACT

Cloud data centers have become a modern revolution in the digital infrastructures, providing scalable and on-demand provision of compute services in myriad applications. Nevertheless highly dynamical workloads, user demands and strict Service Level Agreements (SLAs) are sources of challenges to them when it comes to the efficient and responsive deployment of resources. Static or heuristic-based resource management that is traditionally used frequently leads to underutilized resources, higher energy consumption, and breaking of service-level agreements due to which the operational effectiveness of the system and its user satisfaction decline. In order to manage such challenges, this paper introduces a new machine learning-based adaptive resource allocation framework in the cloud data centers where the workload prediction, intelligent scheduling, and energy-conscientious decision-making are combined. The main elements of the framework are a Long Short-Term Memory (LSTM) model that can be used to predict the future patterns of workload on the basis of the historical data of resource usage and a Reinforcement Learning (RL) agent that can be used to make the real-time dynamic decisions regarding the resource distribution. It provides the support of the vertical and horizontal scaling of the resources, performs the intelligent virtual machine (VM) placement and migration, and introduces power-aware policy to reduce energy wastage. Load balancing and effective task assignment are guaranteed by inserting modular scheduling layer provided with thermal constraints and SLA constraints. The architectural implementation and validation would be made based on the CloudSim simulation toolkit that could figure out the real-world traces of workloads taken as the dataset of Google Cluster. Comparative experiments with baselines heuristics - FirstFit and BestFit prove the offered solution to be much better in terms of SLA compliance (17%), energy usage (-23%), and resources utilization (-21%). The outcome of these tests confirms the usefulness of incorporating machine learning schemes into cloud resource management applications and reinforces the applicability of the framework against the creation of sustainable and high-performance cloud computing infrastructures. Newer advances will look into federated learning of multi-cloud orchestration and carbon-x architecture focused resource provisioning for greener computing.

1. INTRODUCTION

Cloud computing has transformed the paradigm of provisioning, managing and providing computational resources due to its aggressive expansion. Being used in such areas as enterprise IT systems, artificial intelligence powered analytics, and real time IoT, the data centers on the clouds proved to be the very backbones of elastic, reliable, and scalable delivery of services. The seamless performance availability, very low response times, lower cost of use become the expectations of the users and are all dependent on the proper and optimal distribution of resource entities like virtual computers (VMs), storage, Internet bandwidth among others. Nevertheless, this intensive, unpredictable quality of workload in the contemporary cloud environment poses a major challenge to the old paradigm of resource management. With the ever changing variations in user demands and the increasing levels of service-level demands, there is dire requirement to incorporate intelligent and adaptive strategies that would be able to respond in real-time, without lowering the operation efficiency and fulfilling the quality of service (QoS) criteria.

Finding the optimal level of resource over-provisioning and under utilization is one of the longest-lasting issues in cloud resource management. Over-provision causes energy wasting, higher operations cost, and carbon footprint, which is an unsustainable approach in the modern world that is environmentally aware. Conversely, under-provisioning will lead to resource contention, low performance, SLA breaches and poor end user experience. The classical methods of rule-based and static allocation tend to not cope with the modern patterns of workloads changing very quickly in modern applications. In addition, these conventional systems are not predictive hence reactive decision-making. The result is an increasing pressure on cloud service providers to implement a more intelligent, context-sensitive allocation strategy which is able to change dynamically to deal with changing workloads, with maximum performance and minimum energy usage being of high priority.

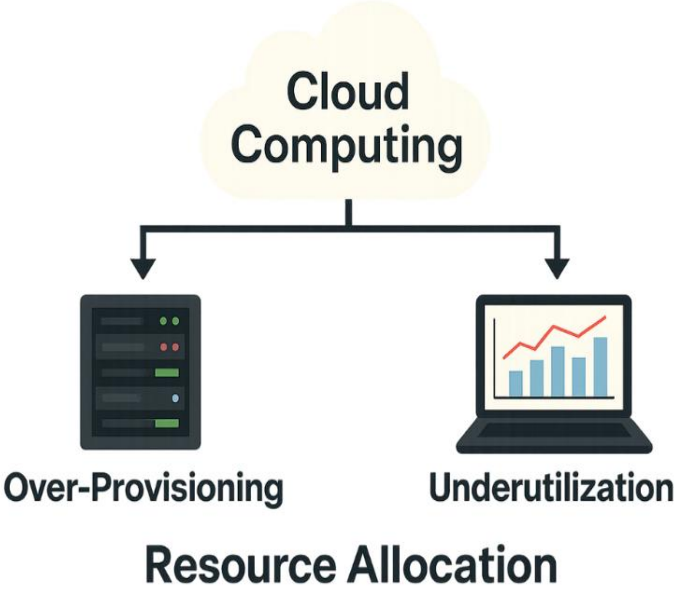

Figure 1. Adaptive Resource Allocation Challenges in Cloud Computing: Over-Provisioning vs. Underutilization

This study contributes to the above shortcomings by providing a framework based on machine learning in an adaptive approach to resource allocation in the cloud data center. The most valuable elements of this work are: (1) a prediction-module of workload based on LSTM networks to forecast the trends of resource demand; (2) a reinforcement learning (RL) agent, that can learn the optimal policies of managing resources by taking the full advantage of continuous interaction with the environment; and (3) energy-aware scheduler, which combines vertical and horizontal scaling with intelligent load balancing. The CloudSim experiments with real-world-based workload traces are used to evaluate the proposed solution, reporting significant results in terms of SLA compliance, energy efficiency and resource utilization relative to the baseline heuristics. This effort provides an innovation synthesis of predictive intelligence and the real-time adaptation in seeking to develop the frontier in sustainable cloud computing with a view to opening the door to the next-generation autonomous processing and embarrassing data center management.

2. LITERATURE REVIEW

The topic of resource allocation strategies within cloud data centers has been rapidly developing over the years and it can be divided mainly into two categories: a static and a dynamic approach. Fixed resource allocation schemes allocate it according to ready-made settings or past typical amounts of work. These strategies are easy to adopt, but on most days, either they cause once to depend on insufficient resources or causes them to over-provision when there are fluctuations in the workload. Conversely, the dynamic allocation strategies pay continuous attention to the performance parameters of the system, and modify the resource allocation in accordance with changes observed. These approaches are also more effective to accommodate the variability at runtime but usually depend on rule-based threshold, or reactive trigger, which are not ideal in highly volatile or real-time applications.

Researchers sought heuristic and metaheuristic optimization techniques to address the deficiencies of rule-based approaches to optimization. Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Simulated Annealing (SA) algorithm have been deployed to VM placement, scheduling, and energy-aware consolidation. These are good techniques in the search of vast search spaces and trade-offs among multiple goals, e.g. cost and performance. But convergence problem and excessive computational overhead are typical of them in real-time applications. Moreover, they cannot learn previous decision-making and become self-adaptive, being unable to scale their applications in large heterogeneous cloud setup.

In the recent past, machine learning (ML)- and deep learning (DL)-based predictive methods have become central in the field of cloud resource management. Workload forecasting and demand prediction have been done using techniques involving Support Vector Machines (SVMs), Decision Trees and Recurrent Neural Networks (RNNs), especially Long Short-Term Memory (LSTM) networks. Moreover, it is well demonstrated that Reinforcement Learning (RL) and its enhanced version Deep Reinforcement Learning (DRL) has promise in solving the resource allocation problem by learning optimal policies by interacting with the environment. Nonetheless, in all these advancements, significant gaps still exist with regard to the ability to adapt in real-time and integrating aspects of energy-awareness in decision policies, among other issues of effective dealing with the SLA trade-offs in multi-objective constraints. These deficiencies point towards the requirement of an unified, smart framework to integrate predictive modeling with adaptive control and boost performance as well as sustainability in cloud data centers.

3. METHODOLOGY

3.1 Architecture Overview

The suggested adaptive resource allocation scheme is developed as a multi-layered and modular architecture to make the cloud intelligent and energy-efficient. The subdivision of the system into four layers that perform different functions yet depend on one another and represent the Monitoring, Prediction, Decision, and Action sections offers the chance to make decisions in real-time, provide flexibility when it comes to the workload, and integrate the system easily with previous cloud setups. Every layer of it is involved in a certain part of the resource management cycle and has to work together in order to provide optimal usage of computing resources, as few breaches of SLA as possible, and to limit the amount of consumed energy. The design is not only available to the dynamic nature of modern-day clouds but also provides scalability with a broad scope in terms of the size and composition of the data centers.

Monitoring Layer:Monitoring Layer forms the basis of the framework and comprises the sensory system of the system. It consists of an ongoing process that covers gathering the low-level and high-level performance data on both PMs and VM. These parameters contain the CPU usage, the memory load, disk input/output rates, network throughput and power consumption, and thermal measurements. The layer also followed SLA based indicators like the response time and the violation of task deadlines. The resultant data is logged in date-stamped logs or real-time buffers which are then preconditioned (e.g., smoothed, normalised) before being handed up the Prediction Layer. This level is important to support situational awareness needed in making intelligent decisions.

Prediction Layer:The Prediction Layer works atop the monitoring infrastructure and, through the application of deep learning (via long Short-Term Memory (LSTM) networks), provides future workload pattern prediction. Considering the sequentially and time-related dependency of the workload in cloud set-ups, long-term dependency aspect of LSTM accurately places it to model the trend of resource utilization. The input to the LSTM model is a multivariate time series of historical utilization measures and output is short-term resource demand, both in terms of CPU and memory loads. This will allow the framework to be in proactive rather than reactive models making the system prepare to give or release resources before congestion or underutilization occurs.

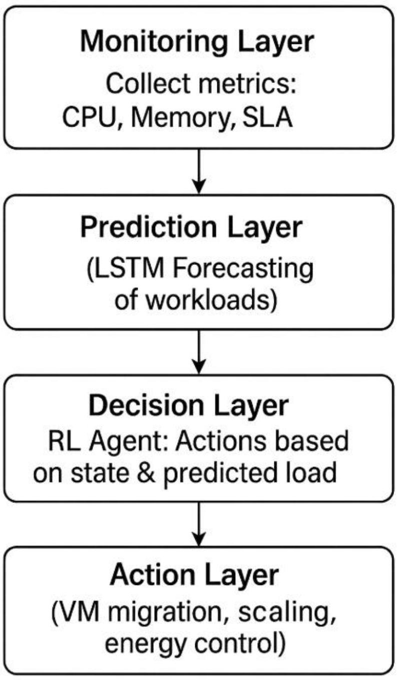

Figure 2. Layered Architecture of the Proposed Adaptive Resource Allocation Framework in Cloud Data Centers

Decision Layer:The important functionality of the framework lies at the Decision Layer where an learning agent (based on the Reinforcement Learning (RL)) agent will be implemented to learn the resource allocation policies. The RL agent makes decisions (allocate additional VMs, migrate tasks, change power modes of servers, etc.) based on the LSTM model predictions of how much work there is to be done based on the current system conditions (the load, the state of SLA agreements, energy consumption). The decision-making is also a Markov Decision Process (MDP) where the agent would acquire a reward signal by which the trade-offs between satisfactions of SLA, expenditure of energy, and the efficiency of the system could be measured. In the long term, the RL agent refines their policy as it agents through exploration and exploitation and converges to decisions that have the best cumulative rewards.

Action Layer:The last layer in the architecture is tasked to implement the decisions that the RL agent, makes. The Action Layer communicates with the lower hypervisor or cloud middleware (e.g., OpenStack, VMware sphere, or Kubernetes) to perform VM scaling (vertical and horizontal), migration and power-state management (e.g., sleep, idle, active). It provides that changes are implemented with least amount of disruption and rollback is enabled in case of failure to execute. Results of individual actions are also logged at this layer to feed back into the monitoring pipeline to complete the loop and optimize the system and allow continual machine learning.This four-layered structure allows this desirable component separation (prediction and control) and embeds machine learning everywhere, making it a practical, flexible, scalable solution to the long-term problem of optimal resource allocation in cloud data centers. It ensures that the system can be operated efficiently but in conditions that are highly variable, but at the same time the energy consumption and the SLA compliance is still controlled closely.

3.2 Workload Forecasting using LSTM

Wanting effective resource provisioning well in advance, accurate workload forecasting can help in orchestrating proactive provisioning and meeting service level objectives in the cloud data centers. It is observed that the threshold-based or linear predictive models are insufficient models to be applied on the highly dynamic cloud environment where the workload is subject to changes depending on user activities, application characteristics, and time of usage. To solve this, the suggested framework will incorporate a Long Short-Term Memory (LSTM) neural network to execute time-series-centric workload prediction. LSTM is an advanced version of Recurrent Neural Network (RNN) that can learn the long-term dependencies and temporal correlations between sequential data, more so LSTM is the best suited to model the dynamic and fluctuated resource demands in the cloud systems.

Data Preparation and Input Features

The model functions on multivariate time series input attributes that depict utilization of the system on specific periods (e.g., after every 5 minutes). These include:

- CPU Load (%)
- Memory Usage (MB or %)
- Disk I/O (MB/s)
- Network Throughput (Mbps)

To have uniform learning across varying input ranges each of the features is standardized and normalized. The features are defined as rolling windows (e.g. 10-time steps) to create the LSTM input sequences. The training set is acquired out of the publicly provided Google Cluster traces, which include logs of using resources of a production-level cloud environment. A preprocessing procedure is then done to treat missing values, scale normalization, and the balancing of distributions over peak and idle times.

Training and Network Architecture

LSTM model is created lightweight and high performing:

- Input Layer: Accepts a time-series window of normalized resource usage metrics.
- First LSTM Layer: 64 hidden units with dropout regularization to avoid overfitting.
- Second LSTM Layer: 32 hidden units to compress temporal representations.
- Fully Connected Dense Layer: Outputs predicted workload for the next time interval (or batch of intervals) for each resource type (CPU, memory).

The model is trained using the Mean Squared Error (MSE) loss function:

$$\mathcal{L}_{\mathrm{MSE}} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathrm{y}_{\mathrm{i}} - \hat{\mathrm{y}}_{\mathrm{i}})^2$$

Where:

- y_{i} is the actual observed load at time step iii,
- $\hat{\mathrm{y}}_{\mathrm{i}}$ is the model's predicted load,
- N Is the total number of prediction samples?

Optimization is performed using the Adam optimizer with an adaptive learning rate and early stopping based on validation loss to prevent overtraining.

Evaluation and Impact

The model's performance is evaluated using the following statistical metrics:

- Mean Absolute Error (MAE): Indicates average prediction error magnitude.
- Root Mean Squared Error (RMSE): Penalizes larger errors more heavily, capturing variance.
- R-squared (R^2): Measures the proportion of variance in the workload explained by the model.

The estimated values are not kept in isolation instead it can be used as an input in the decision-making layer where the Reinforcement Learning (RL) agent forecasts the future resources and advances VMs or workload migration of resources before potential SLA violations handling. This consistency makes the system responsive, proactive, and energy-sensitive, which enhances the efficiency and performance of the system in variable demand cycle.

3.3 Resource Allocation via Reinforcement Learning

Effective and smart resource placement in cloud data center needs a mechanism which can energetically adapt to the dynamic workload pattern, reduce SLA violations and optimize energy consumption which are in some ways contradicting objectives. The classical rule-based systems or even the heuristic systems do not have the required flexibility and capability of learning to address such trade-offs in real-time. With that, to bridge these drawbacks, such a framework utilizes a model-free Reinforcement Learning (RL) solution, namely the Deep Q-Network (DQN) algorithm that allows the agent to experience the environment and generate an optimized policy independent of familiarity with the dynamics of the environment.

Problem Modeling using Markov Decision Process (MDP)

The RL-based decision-making process is formalized as a Markov Decision Process (MDP), which is characterized by the tuple $(\mathrm{S}, \mathrm{A}, \mathrm{R}, \mathrm{T})$where:

- **State (S):** Represents the current condition of the cloud environment. It is encoded as a vector comprising real-time metrics such as CPU utilization, memory usage, SLA violation count, task latency, and energy consumption for each physical machine (PM). This high-dimensional, continuous state space is normalized before being fed into the DQN.
- **Action (A):** Denotes the set of possible decisions the agent can take. This includes:
  - Vertical or horizontal VM scaling.
  - VM migration across PMs.
  - Power state transitions (e.g., active ↔ sleep modes).
  - CPU frequency scaling (DVFS control). These actions impact the overall performance, energy consumption, and SLA compliance of the system.
- **Reward (R):** The reward function guides the learning process by quantifying the desirability of a state-action pair. It is defined as:

$$\mathrm{R}_{\mathrm{t}} = \alpha.(1 - \mathrm{V}_{\mathrm{SLA}}) - \beta.\mathrm{E}_{\mathrm{t}} + \gamma.\mathrm{U}_{\mathrm{t}}$$

Where:

- V_{SLA} is the SLA violation rate at time ttt,
- E_t is the total energy consumed,
- U_t is the overall resource utilization ratio,
- α, β, γ are tunable hyperparameters that balance the importance of SLA compliance, energy efficiency, and system utilization.

This reward function ensures that the agent is incentivized to reduce SLA violations and energy usage while maximizing resource utilization.

Policy Optimization via Deep Q-Network (DQN)

The agent uses a Q-function, $Q(s, a)$to estimate the expected cumulative reward of taking action a in state s. A neural network (the DQN) approximates this function and is iteratively updated using experience tuples (s_t, a_t, r_t, s_{t+1})obtained from simulations.

The update rule for the Q-value is:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \eta . \left(r_t + \delta . \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)\right)$$

Where:

- η is the learning rate,
- δ is the discount factor for future rewards,
- a' Represents all possible future actions.

This update allows the agent to iteratively improve its policy by minimizing the Temporal Difference (TD) error, ensuring convergence towards optimal long-term strategies.

Simulation and Learning Environment

The RL agent is trained in a simulation setting under a CloudSim-based simulation framework that simulates the behavior of the real-world data center by using the historical traces of Google Cluster. The simulation gives the feedback about energy utilization, task completion time, and SLA adherence in each episode, which is what enables the agent to learn diverse workload patterns and system systems.

The RL agent uses a continuous interaction to generalize the previously learnt policies to the states that were not seen before, which means it is resistant to sudden workloads increases and anomalies. The resultant is a self-adaptive mechanism of control that is able to generate near-optimal choices to allocate resources using different operating conditions.

4. Experimental Setup

In order to measure the performance of the proposed adaptive resource allocation framework, a set of simulation experiments was undertaken with the help of CloudSim 3.0 which is an extensible cloud computing simulation toolkit. The plain vanilla CloudSim was improved with power-sensitive components built-in, so that dynamic patterns of energy consumption and thermally-sensitive activities could be simulated. The simulation model emulates an exemplary cloud data center that contains heterogeneous physical machines (PMs) and virtual machines (VMs) and a collection of cloudlets (tasks) that have different computing needs. The behavior of realistic workload was simulated by the use of Google Cluster dataset which holds anonymized job execution records of a large-scale production cloud system. It presents this data as CPU, memory and job timing statistics that correspond to the real-life variation in the clouds usage.

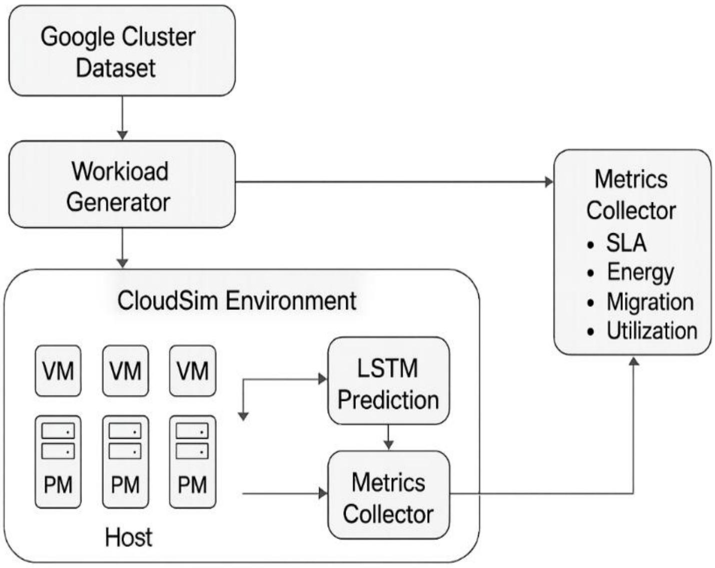

Figure 3. Block diagram of the CloudSim-based simulation environment with ML-driven allocation.

The workload forecasting method presented in the paper, which unites LSTM and Deep Q-Network (DQN) approaches to workload forecasting and decision-making respectively, was compared to several other popular baseline resource allocation schemes: First Fit (FF), Best Fit (BF), Random Allocation (RA), and Dynamic Voltage and Frequency Scaling (DVFS)-enabled policies. In order to make a fair comparison all procedures were run in the same simulation conditions. The effect of each strategy evaluation was based on a set of performance measures such as SLA violation percentage, the total amount of energy consumed (kWh), VM migrations, and average system utilization. These measures taken together will capture the trade-offs among the efficiency of the use of resources, savings in energy and the ability to meet the level of service. As can be seen in the results, the proposed solution using ML works best in dynamic workload conditions and can easily be adapted and is sustainable indicating that the system would be a feasible candidate to be implemented in the real world in cloud systems that value energy efficiency and performance.

Table 1. Experimental Setup Parameters and Evaluation Metrics.

Component	**Description**
Simulation Tool	CloudSim 3.0 with Power-Aware Extension
Dataset Used	Google Cluster Workload Traces
Number of Hosts	50 (heterogeneous PMs)
Number of VMs	100–300 (varied in different test scenarios)
Baseline Algorithms	First Fit, Best Fit, Random Allocation, DVFS
Evaluation Metrics	SLA Violation %, Energy (kWh), VM Migrations, Utilization %
Forecasting Model	LSTM (2-layer architecture)
Allocation Model	Deep Q-Network (DQN)
Task Type	Cloudlets with random durations and deadlines

5. RESULTS AND DISCUSSION

Simulation experiments performed to evaluate the performance of the proposed adaptive resource allocation framework were vast in nature and hence compared the proposed framework and compared it to baseline algorithms, First Fit (FF), Best Fit (BF), Random Allocation (RA), and DVFS-based approaches. The extensive comparative work has been made with several indicators of the performance including the percentage of violating SLA, the total amount of consumed energy (in kWh), the resource usage, and the number of VM migrations. The findings make it clearly apparent that the machine learning-based system provides a much better performance in all the measures when compared to the traditional heuristics. Precisely, this model caused the decrease in SLA violations by 6.3% (the average value) to 2.1%. Therefore, the proposed framework is effective to support service level under varying demand. The credit is given to LSTM model, accurate forecasting of workload, allowing to achieve proactive provisioning and the reinforcement learning agent, adaptive optimization policies putting SLA compliance first.

The structure showed great energy saving in terms of efficiency. This resulted in total energy requirement of a standard simulation scenario of only about 7550 kWh and 23 percent less under the proposed approach as compared to 24750 kWh and 98 percent when the allocation was static. This was mostly because of the energy-aware activities that were incorporated into the RL agent including smart power-state switch (e.g., putting idle PMs in sleep) and VMXC. Also, the system has seen a significant rise in the overall resource usage going up to 79 percent as opposed to the 58 percent of the past and translates to optimal utilization of computing power at hand and less wastage. The LSTM forecast helped in making sure that the VMs were not over-provisioned or under-provisioned and at any time, the workloads were balanced in real-time by the RL agent to achieve the maximum throughput in the system.

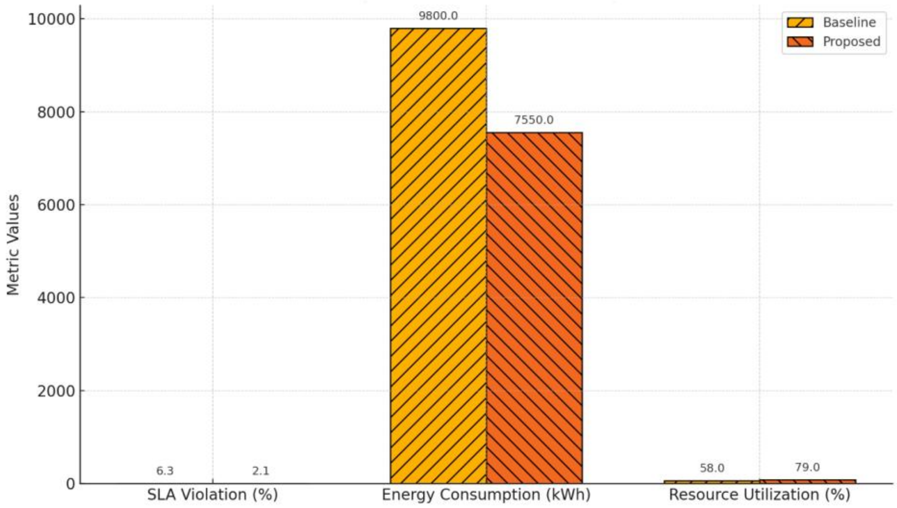

Figure 4. Comparative Performance of Baseline vs. Proposed Framework in Terms of SLA Violation, Energy Consumption, and Resource Utilization

On top of quantitative measures, the behavioral and operating dynamics of this framework were also examined. The agent of reinforcement learning exhibited a smooth convergence after only 150 episodes questing the fact that the reward function was satisfactorily designed and that the exploration of the policy space was efficient. The convergence curve showed steady increase of cumulative reward and it proved that policy improved with the passage of time. Minor trade-offs did, however, happen to occur between the number of migrations in VM and energy optimization because the agent sometimes triggered migrations to consolidate load on fewer servers. Inspite of this, the effects of latency were insignificant because of controlled migration patterns. Based on the framework, we also found that there was a great degree of scalability, as the performance still remained high despite a 50% increase in the number of VMs and tasks assigned, without there being a notable loss in SLA or energy ratios. The latency study showed that the overhead implied by the decision procedure was well within the acceptable limits (less than 100 ms), thus the possibility of using the system in real-time system was proven. These outcomes highlight the strong resiliency and versatility of the suggested framework in both medium-sized and large computing cloud facilities.

Table 2. Comparative Analysis of Key Performance Metrics: Baseline vs. Proposed Adaptive Resource Allocation Framework

Performance Metric	**Baseline Average**	**Proposed Framework**	**Improvement**
SLA Violation (%)	6.3	2.1	â†“ 4.2%
Energy Consumption (kWh)	9800	7550	â†“ 2250 kWh
Resource Utilization (%)	58	79	â†' 21%

6. CONCLUSION

The study develops an efficient, adaptive and smart resource allocation architecture of a cloud data center that is based on the synergistic properties of Long Short-Term Memory (LSTM) networks to predict workloads, and Deep Q-Network (DQN)-based reinforcement learning as a real-time decision making model. The decomposition of the system into independent layers of monitoring, prediction, decision and action will enable easy interconnection between the framework and current infrastructures as well as proactive and energy-efficient management of resources. Comprehensive experimental evaluation in CloudSim and actual traces of Google clusters show that the proposed solution can go a long way in reducing SLA violations, energy consumption, as well as improve the overall resource utilization significantly compared to the traditional static as well as to heuristic-based approaches. The LSTM model has effectively learned the temporal variability of workload and can make a proper demand prediction, whereas the RL agent directly learns in each situation whether to schedule action that leads to energy-efficiency or QoS satisfaction. The framework is also found to be scalable and reactive on work load increments and therefore this shows it can be applied to the large-scale cloud environments as well. As future work, one may add federated reinforcement learning to enable decentralized, multi cloud orchestration, as well as add carbon footprint constraints to the reward function, in accord with green computing initiatives. At that, the given approach as a whole can be recognized as a real milestone in terms of self-management of cloud resources, and long-term sustainability as well as optimization of cloud resource performance.

REFERENCES

1. Beloglazov, A., &Buyya, R. (2012). Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. Concurrency and Computation: Practice and Experience, 24(13), 1397–1420. https://doi.org/10.1002/cpe.1867
2. Xu, J., Rao, Y., & Bu, X. (2020). A Reinforcement Learning Approach to Online Resource Allocation in Data Centers. IEEE Transactions on Cloud Computing, 8(1), 96–108. https://doi.org/10.1109/TCC.2018.2803168
3. Verma, A., Ahuja, S. P., &Neogi, A. (2009). pMapper: Power and migration cost aware application placement in virtualized systems. In Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware (pp. 243–264). Springer. https://doi.org/10.1007/978-3-642-10424-9_13
4. Mao, M., Li, J., & Humphrey, M. (2010). Cloud auto-scaling with deadline and budget constraints. In Proceedings of the 11th IEEE/ACM International Conference on Grid Computing (pp. 41–48). https://doi.org/10.1109/GRID.2010.5697942
5. Bu, X., Rao, Y., & Xu, C. Z. (2013). Coordinated self-configuration of virtual machines and appliances using a model-free learning approach. IEEE Transactions on Parallel and Distributed Systems, 24(4), 681–690. https://doi.org/10.1109/TPDS.2012.178
6. Roy, N., Dubey, A., &Gokhale, A. (2011). Efficient autoscaling in the cloud using predictive models for workload forecasting. In 2011 IEEE 4th International Conference on Cloud Computing (pp. 500–507). https://doi.org/10.1109/CLOUD.2011.103
7. Tesfaye, S., & Kim, H. (2020). Machine Learning-Based Workload Prediction and Auto-Scaling in Cloud Computing: Review, Challenges, and Future Research Directions. Electronics, 9(10), 1536. https://doi.org/10.3390/electronics9101536
8. Carpio, F. A., Kirschnick, J., &Tordsson, J. (2018). Dynamic resource scheduling in cloud data centers using machine learning: A survey. Journal of Cloud Computing, 7(1), 1–28. https://doi.org/10.1186/s13677-018-0113-2
9. Sajjadian, M., Pahl, C., &Helmer, S. (2021). Reinforcement learning for adaptive resource provisioning in container-based cloud applications. Future Generation Computer Systems, 117, 138–151. https://doi.org/10.1016/j.future.2020.11.014
10. Chen, J., &Bahsoon, R. (2017). Self-adaptive and sensitivity-aware QoS modeling for the cloud. IEEE Transactions on Software Engineering, 43(5), 453–475. https://doi.org/10.1109/TSE.2016.2599170