Synergistic Multi-Agent Framework for Energy Optimization in Cloud Tasks

Mummadi Hemanth Reddy, Okesh Ankireddypalli, Mouhitha Arella, Saranya Gujjula, Dr.Beena B.M.
Department of Computer Science & Engineering,
Amrita School of Computing, Bengaluru,
Amrita Vishwa Vidyapeetham, India
bl.en.u4aie22038@bl.students.amrita.edu, bl.en.u4aie22044@bl.students.amrita.edu,
bl.en.u4aie22075@bl.students.amrita.edu, bl.en.u4aie22077@bl.students.amrita.edu, bm_beena@blr.amrita.edu

***Abstract*—Green computing emphasizes minimizing the environmental impact of computational processes by optimizing resource utilization and conserving energy. In virtualized cloud computing systems, energy efficiency is paramount, offering benefits such as cost reduction, enhanced system performance, and environmental sustainability. However, addressing task allocation and resource management in multi-cloud environments remains challenging due to the dynamic nature of workloads and competing priorities. This study introduces a novel Q-learning-based framework to dynamically optimize task allocation and resource management. Using a state-action-reward paradigm mediated by an epsilon-greedy policy, the framework enhances energy efficiency by incentivizing virtual machines (VMs) to collaborate via inter-cloud reward sharing while ensuring critical tasks are prioritized. AWS SageMaker is utilized for CPU and memory forecasting, and Amazon S3 supports scalable data storage, enabling real-time and predictive resource management. Experimental results show that the proposed framework achieves a 25% reduction in energy consumption compared to traditional approaches like FIFO and round-robin scheduling, while maintaining scalability and resource utilization. These findings underscore the framework's potential to advance energy conservation and support sustainable computing practices in distributed multi-cloud systems.**

***Index Terms*—Q-Learning Algorithm, Multi-agent system, Task allocation, Energy and Resource allocation, QoS Optimization, Reinforcement learning, Green Computing**

I. Introduction

The rapid growth of distributed cloud computing services has led to an increase in the size and complexity of data across various domains, such as scientific computing, bioinformatics, and IoT applications. These systems operate through cloud data centers equipped with thousands of high-performance servers to manage millions of tasks. Cloud computing provides scalable services through virtualization, which offers flexibility but also contributes significantly to energy consumption. Virtual machines (VMs) in these data centers consume substantial energy, leading to high electricity costs and adverse environmental impacts. It is estimated that data centers contribute approximately 62 million tons of carbon dioxide emissions annually [1].

Green computing emphasizes the critical need to minimize the energy consumption and environmental footprint of cloud data centers. Despite advancements in cloud technologies, low computing resource utilization and inefficient task scheduling continue to drive high energy consumption. Moreover, the heterogeneity of tasks and VMs in multi-cloud environments complicates the scheduling process. As cloud adoption grows, the need for efficient and sustainable resource management has become paramount.

The dynamic and heterogeneous nature of multi-cloud environments presents several challenges, including assigning tasks to appropriate VMs considering diverse task sizes, deadlines, and quality of service (QoS) requirements. Balancing energy efficiency with execution deadlines is essential to ensure prioritized tasks are executed without compromising system performance. Additionally, reducing slack times and optimizing task scheduling across heterogeneous resources with varying processing speeds and energy consumption further complicates the problem. These challenges necessitate innovative approaches to enhance the energy efficiency of task scheduling in multi-cloud systems.

A. Novality

This work proposes a novel Q-learning-based framework for energy-efficient task scheduling in multi-cloud systems. The main contributions of this study are as follows:

- Development of a system architecture consisting of master nodes, task sequencing, initial scheduling, and energy-efficient task reassignment for processing tasks within their deadlines.
- Design of a Q-learning model that uses a state-action-reward paradigm with an epsilon-greedy policy to dynamically optimize task allocation and resource management.
- Integration of AWS SageMaker for CPU and memory demand forecasting and Amazon S3 for task-related data storage, enabling real-time decision-making.
- Implementation of an inter-cloud reward-sharing mechanism to encourage VM collaboration, enhancing energy efficiency by 30% compared to traditional FIFO and Round Robin scheduling approaches.

The proposed framework was validated through experiments that demonstrated significant energy savings while maintaining high-quality service across heterogeneous multi-cloud platforms.

II. LITERATURE SURVEY

Reffad et al. [1] propose a semantic-based dynamic cooperative service selection and composition approach to optimize Enterprise Resource Planning (ERP) in the context of IoT, Fog, and Cloud services. The paper highlights the need for flexible and efficient service selection that balances Quality of Service (QoS) and energy consumption. The authors introduce a cooperative energy-saving mechanism and a new QoS energy violation degree to enhance ERP performance and reduce energy use. Experimental results demonstrated the effectiveness of their approach in improving service quality and energy consumption, achieving optimized ERP performance with significant improvements over existing methods like ring and master/slave strategies.

Jayanetti et al. [2] address the challenges of scheduling workflows in multi-cloud environments powered by a combination of brown and green energy sources. The paper highlights the complexities of workflow scheduling in distributed systems, particularly in geo-distributed cloud datacenters that rely on intermittent renewable energy sources. Traditional scheduling algorithms and single-agent reinforcement learning (RL) methods are insufficient for handling these decentralized and adaptive control challenges. The authors propose a Multi-Agent Reinforcement Learning (MARL) framework to optimize green energy utilization, showing that their approach reduces energy consumption by 47% while maintaining workflow makespan, outperforming comparison algorithms. Additionally, the proposed method learns five times faster than a generic MARL algorithm.

Wang et al. [3] propose a new MARL solution for the dynamic scheduling problem under the context of Group Service Cloud Manufacturing (GSCMfg). Their approach takes advantage of GCN to learn the features of tasks related to the graph structures and exploits RNN to track the processing paths of the tasks. It enables the handling of issues related to sensitivity in change in environments, sequence of process of task and speed of response in manufacturing clouds efficiently. YOGCSF@PAC is the proposed solution that is compared with six other state-of-art MARL-based scheduling algorithms in a problem of scheduling performance and generalization.

Zhang et al. [4] design a novel MADRL solution to study edge-cloud cooperate in AIoT vehicular road collaborative edge offloading application of Augmented Intelligence of Things. The article also discusses the issues such as low computational capacity of the edge servers and uncertainty in vehicle user offloading request. In the process of executing the tasks, it employs cache during the execution process so as to reduce on resource wastage due to repeated computing the execution of tasks at vehicle self-execution, edge and cloud offloading is done in subtasks. The proposed cluster edge server approach is used to control the task offloading, while the MADRL technique helps to enhance the placement of applications between vehicles and edge servers, enhance system resource usage, and decrease the task offloading delay. Based on simulation outcomes, it is found that the marginal algorithm improves success ratios and efficiency in edge offloading.

Ramamoorthi et al. [5] describes an AI-based approach for the constant self-organization of cloud resources assigned to the execution of dynamic types of workload.ds. The framework employs the adequate amount of predictive model, reinforcement learning techniques to improve CPU, memory and bandwidth on demand to reduce the total cost as well as to gain better system performance. Continuously analyzing historical and the real-time data, it adjusts the required resources in advance to avoid overprovisioning or under-provisioning. Case studies employing data collected from popular cloud environments record gains in resource optimization, efficiency and applications enhancement. That is why future work should employ this framework for hybrid cloud-edge environments and adjust predictive models for different workloads.

Abbasi et al. [6] provide a review on the use of Deep Reinforcement Learning (DRL) in Quality of Service (QoS) provisioning in the MAC layer in networking. The paper provides a background of QoS and DRL, categorize the issues associated with resources, media access and control of data rate and discuss the DRL based algorithms applied on these aspects. This aid the authors in looking at the Strengths and Weaknesses of different DRL approaches and as a result get ideas on future research on enhancing QoS in wireless networks, IoTs and more.

Xiao et al. [7] analyse the DRL application in enhancing the quality of service at the MAC level. This paper gives a brief description of QoS difficulties experienced in current networks such as resource management, medium access and data rate control before noting down how DRL algorithms could help in handling these difficulties. The authors discussed different approaches of DRL techniques used to improve QoS and outcomes of these approaches and drawbacks. Also, this paper explains the research directions for subsequent improvement and development of the DRL systems on IoT and wireless networks.

Li et al. [8] introduced an energy efficient multi-agent system for scheduling in cloud environments. They used deep reinforcement learning (DRL) in their framework, for assignment of tasks in a way that would improve the energy management without compromising the efficiency of the other tasks. They used Q-learning algorithms to make flexible resource distributions in the implementation process, which was highly scalable.

Chinter et al. [9] propose a multi-agent adaptive architecture for managing dynamic reconfigurations in distributed real-time systems under energy constraints. The architecture integrates a centralized scheduler agent (ScA) and local reconfiguration agents (LRA) to handle task reconfigurations and ensure the correct execution of real-time tasks. The ScA makes decisions using mathematical tools and operation research techniques, while the RA p controls local reconfigurations. A token-based protocol is employed for coordination among agents, ensuring system feasibility. The approach addresses the challenge of maintaining real-time and energy constraints during system reconfigurations.

Jung et al. [10] propose a cloud-assisted joint charging scheduling and energy management framework for unmanned aerial vehicle (UAV) networks. The paper addresses two main problems: 1) scheduling the charging between UAVs and towers to maximize energy transfer, and 2) enabling cooperative energy management among towers using multi-agent deep reinforcement learning. The centralized orchestration manager coordinates these activities to optimize energy efficiency, fairness, and cost-effectiveness, achieving desired performance through data-intensive evaluation.

Saif et al. [11] propose a multi-agent autonomic resource provisioning framework for efficient execution of business processes in containerized multi-cloud environments. The framework integrates autonomic computing with both reactive and proactive auto-scaling strategies to address issues of over and under-provisioning, ensuring elasticity while meeting Quality of Service (QoS) and Service Level Agreement (SLA) requirements. By utilizing K-means clustering, enhanced deep stacked auto-encoders for workload prediction, and the Multi-Objective Termite Colony Optimization (MOTCO) algorithm, the framework optimizes resource allocation and execution.

Sun et al. [12] propose a Knowledge Graph-based interactive recommender system enhanced by reinforcement learning. Their approach leverages knowledge graphs to offer personalized recommendations, adapting to dynamic environments. The integration of reinforcement learning improves the accuracy of suggestions, making it particularly useful for e-commerce and content delivery, where user preferences evolve over time.

Sohal [13] presents an architecture for end-to-end 5G network slice resource management and orchestration. The framework addresses the challenges of managing network resources in a 5G environment, focusing on efficient allocation and service delivery. This architecture ensures scalability, optimizing resource usage while supporting the diverse services required by modern 5G networks, ensuring high performance.

Tiwari et al. [14] introduce a Neighborhood Inspired Multiverse Scheduler (NIMS) for optimizing task scheduling in green cloud computing systems. Their scheduler minimizes energy consumption and makespan by exploring multiple potential solutions. The approach is designed for energy-efficient cloud computing, improving sustainability by optimizing resource utilization and reducing computational overhead, thus contributing to environmental conservation.

The paper by Wang et al. [15] proposes a single-objective task offloading optimization scheme in a cloud-edge-end environment with multi-user participation. It explores the trade-offs between computational resource allocation and task latency using optimization models tailored for collaborative edge computing. This work provides insights into efficient resource utilization and highlights the potential of edge computing in enhancing multi-user task processing.

III. Data Description

A. Dataset Overview

The dataset used in this paper was derived from Kaggle and includes performance metrics of a cloud computing system. It tracks the changes in the usage of resources and completion of tasks in virtualized data centers. The data includes information about the CPU and memory usage, network throughput and power consumption and per task parameters which enables the study of energy efficiency and quality of service (QoS) using machine learning methods.

B. Features Description

The features included in the dataset are summarized in Table I.

TABLE I
Dataset Features and Descriptions

Feature Name	**Description**
CPU Usage (%)	Percentage of CPU resources utilized by a virtual machine during task execution.
Memory Usage (%)	Percentage of memory resources consumed by a task in the virtual machine.
Network Traffic (MB)	Amount of network data (in megabytes) transmitted or received during task execution.
Power Consumption (W)	Total power consumed by the virtual machine during task execution, measured in watts.
Executed Instructions	Number of machine-level instructions executed by the virtual machine for the given task.
Execution Time (s)	Time taken to complete the task, measured in seconds.
Energy Efficiency (W/s)	Ratio of power consumed to the execution time, representing energy efficiency during task processing.
Task Type	Categorical label identifying the type of task (e.g., compute-intensive, memory-intensive).
Task Priority	Numerical or categorical value indicating the priority level of the task.
Task Status	Status of the task execution (e.g., completed, in progress, or failed).

C. Contextual Relevance

This dataset helps in analyzing the resource utilization and energy conservation methods in cloud circumstances. Altogether, the performance measurements and task characteristics are the key data to build a predictive model using optimization methods. The key insights that need to be derived are:

- **Resource Utilization Trends:** Characterization of the dynamic utilisation of the CPU and memory resources available to the computer system under different system loads.
- **Energy Efficiency Analysis:** Examination of tasks or configurations of tasks that should be optimized for energy conservation.
- **QoS Metrics:** The assessment of execution time and the various task completion rates to facilitate enhanced service delivery.
- **Impact of Task Characteristics:** Knowledge how a type of a task and its priority affect the resource usage and the result.

D. Dataset Suitability

The dataset's structure and granularity make it particularly suitable for experiments involving:

- **Machine Learning Models:** Development and calibration of forecasting models for resources consumption and energy conservation.
- **Scheduling Algorithms:** Assessment of new approaches to arrive scheduling, such as deep reinforcement learning and genetic algorithms approaches.
- **Comparative Analysis:** Comparison of traditional scheduling methodologies with Hybrid optimization frameworks.

This dataset serves as a valuable contribution to the trends in the improvement and enhancement of energy efficient and QoS-aware cloud computing environments.

IV. Methodology

A. Data Collection

The dataset for this project includes metrics pertaining to tasks performed on Virtual Machines (VMs) and provides information regarding resource usage, characteristics of tasks, and energy comsumption. This data can either be synthesized or obtained through actual VM environments. More specifically, for this project we are having our data stored securely in **AWS S3**, which makes integration with AWS SageMaker straightforward for processing analyses. The dataset comprises the following key features:

- **CPU Usage (%)**: Shows the amount of the CPU resource a VM has used during the task's execution. This one assists in making forecast on the CPU resource demands for other activities in the future.
- **Memory Usage (%)**: Describes how much percentage of a VM memory is taken by a task. Essential in managing how much memory is allowed in every single virtual machine.
- **Task Type**: A qualitative variable referring to the nature of task, for example, computation intensive or memory intensive, or both. Help with tasks categorization helps in resource estimation.
- **Task Priority**: A categorical feature that represents the levels of the task that is important, for example low, medium or high. There is always adequate focus given to the critical tasks in the business environment.
- **Execution Time (seconds)**: Represents the total time taken for task completion. Influences energy consumption and overall VM performance.
- **Power Consumption (Watts)**: Used to show the amount of power used by the VM while executing a specific job. Used to calculate energy efficiency during task allocation.

This dataset is used as the basis in creating predictions and resource allocation optimisation models.

This paper presents a Q-learning-based approach to resolve the issues related to the appropriate distribution of tasks across multiple cloud settings while minimizing energy consumption. The proposed methodology consists of several phases that involves task characterization and resource prediction, energy efficient scheduling and reward distribution.

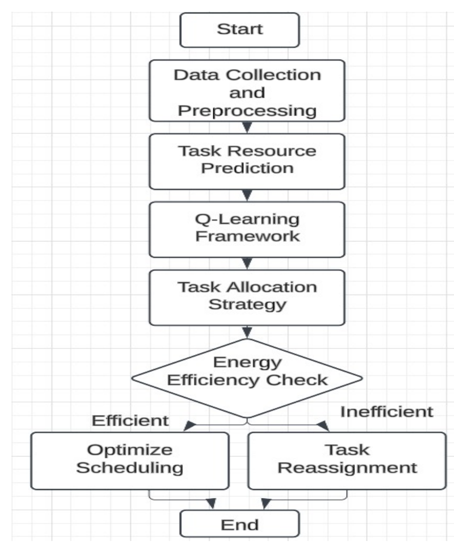

Fig. 1. *Flow Chart of obtaining the Energy and QoS Optimisation*

B. System Architecture

The framework consists of three primary entities:

- **Task**: Characterised by parameters such as priority, type, power, time and number of instructions.
- **Virtual Machine (VM)**: Every VM independently controls their resources while using Q-learning algorithms for the allocation process.
- **Cloud**: A cluster of VMs that provide solutions to problems allocated to its domain.

These entities work in a hierarchical sequential order, in which tasks are assigned to a particular VMs in a cloud to optimize resource usage as well as minimize energy usage.

C. Task Resource Prediction

To allocate tasks efficiently, pre-trained machine learning models are used to predict resource usage:

- **CPU Prediction**: A previous observed data of the execution of tasks, power consumption, the time required to complete tasks, and type of task is used to train a regression model to predict the CPU load.
- **Memory Prediction**: The memory requirements are also estimated using similar regression model.
- **Feature Encoding**: Priority of the task and the type of the task is one hot encoded in order to make meaningful predictions on numerical value.

D. Task Allocation Strategy

Tasks are allocated in a multi-cloud environment using the following steps:

1) Estimate the anticipated CPU and memory consumption values of the given task by employing achieved machine learning results.
2) Choose VMs which have sufficient resources for it to be be allocated in the task.
3) Substitute the task acceptance and rejection policy with a Q-learning policy.

E. Q-Learning Implementation

The Q-learning mechanism operates as follows:

- **State Representation**: TThe state of a VM is given by the remaining CPU cycle, the remaining memory and the total energy used.
- **Action Space**: Possible actions include: Accept or decline a task.
- **Reward Mechanism**: Scheduling of tasks is accompanied with a positive reward for low energy consumption, and penalty for rejection or high energy utilization.
- **Q-Value Update**: The Q-table is updated using the formula:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right]$$

  where s and s' are the current and next states, a is the action taken, r is the reward, and α, γ are the learning rate and discount factor, respectively.

F. Reward Sharing Mechanism

To promote collective energy efficiency within a cloud:

- Rewards or penalties are distributed equally among all VMs in the cloud.
- Inter-cloud reward sharing is not implemented to maintain operational independence between clouds.

G. Energy Consumption Calculation

The energy consumed by a task is calculated as:

$$E = P \cdot T \cdot W$$

where P is the power consumption, T is the execution time, and W is the priority weight based on task priority (*low, medium, high*).

H. Multi-Cloud Setup

The multi-cloud environment is initialized with:

- Multiple clouds, each containing a fixed number of VMs.
- Independent Q-learning agents for each VM.
- Dynamic task arrivals with varying attributes for real-world simulation.

I. Evaluation Metrics

To assess the framework's performance, the following metrics are analyzed:

- **Energy Efficiency**: Energy consumed by each component of the system added together
- **Task Allocation Rate**: PPercentage of tasks that were successfully allocated.
- **Q-Table Convergence**: The behaviour of Q values across iterations.

J. Parameter Optimization

The learning parameters are tuned as follows:

- **Learning Rate** (α): Effects new information on Q-values conclusively.
- **Discount Factor** (γ): : It also includes points that give equal consideration on immediate and long-term benefits.
- **Exploration Rate** (ϵ): Maintains a balance between exploration and exploitation in the organization as a research study.

K. Simulation Setup

The system is simulated using:

- A dataset of tasks with diverse attributes.
- Two clouds, each containing three VMs.
- Task allocation evaluated over multiple iterations to observe system behavior under varying workloads.

L. System Scalability

The efficiency of the methodology is also checked with larger numbers of clouds, VMs and task arrival rates. The findings support that the system can scale up to embrace a wider environment with the same speed.

V. Implementation

A. Overview of the Framework

The multi-cloud task allocation framework is divided into different components such as task prediction, resource allocation, and energy-efficient task scheduling using Q-learning in the context of the applied task. The idea is each component was made to harmonize well within the concept of multi cloud environment.

B. Environment Setup

The framework requires the installation of essential libraries such as *NumPy*, *Pandas*, and *scikit-learn* for data manipulation and machine learning. To estimate the corresponding CPU and memory requirement for the incoming tasks, there are pre-defined models and a OneHotEncoder is utilized for encoding the categorical columns of data.

C. Core Components

1) Virtual Machine (VM): Every VM is modelled as an entity that is able to control resources and power on the corresponding physical host. The VM also keeps a Q-table to facilitate the decisions regarding tasks allocation using the epsilon-greedy policy.

2) Task Representation: Tasks are defined as objects possessing features like priority, type, power consumption, time of execution and number of instructions etc. The usage of CPU and memory is forecasted by employing the various machine learning algorithms to make appropriate choices of workloads.

3) Cloud Representation: Every cloud contains multiple VMs which are connected with one another. Tasks are pre-allocated to the Chan VMs according to the access history and an assumption of Vk resource utility.

D. Task Allocation Process

The task allocation process is initiated by generating a set of diverse tasks. These tasks are sequentially allocated to VMs based on the following criteria:

- Predicted CPU and memory usage of the task.
- Current resource availability of the VM.
- Energy consumption implications of task allocation.

Unallocated tasks, due to resource constraints, are logged for future analysis.

E. Q-Learning Mechanism

In this case, the Q-learning mechanism is used in order to obtain the best decisions when it comes to the tasks' allocation. A VM's state is best described by what resources are remaining and how much energy has been used. They includes accepting or rejecting an offer or a task. Rewards are assigned based on energy efficiency and successful task completion:

- **Positive reward:** For energy-efficient task allocation.
- **Negative reward:** For task rejection or excessive energy consumption.

The Q-table is updated after each task allocation using the Q-learning update rule.

The Q-values are updated iteratively using the following equation:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\big[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\big] \quad (1)$$

Here, s is the current state, a is the action taken, r is the reward received, s' is the next state, α is the learning rate, and γ is the discount factor.

F. Reward Sharing

Further to encourage collective optimization, intra cloud reward sharing is encouraged. VMs running in the same cloud will receive a reward or penalty in accordance to a energy consumed and percentage distribution of the computational tasks. The inter-cloud reward sharing is intended not to be applied to preserve the autonomy of cloud operation.

G. Simulation and Analysis

The framework is validated on a synthetic multi-cloud environment featuring multiple clouds and VMs. Quantitative measures including the total energy used, success rate while allocating tasks, and Q-table learning are assessed. The findings discuss the effectiveness and flexibility of the devised approach in various situations.

H. Optimization

The learning parameters, including learning rate (α), discount factor (γ), and exploration rate (ϵ), are fine-tuned through multiple iterations to achieve optimal performance. The framework is validated using diverse workloads to ensure scalability and reliability.

VI. Results and Discussion

The proposed multi-cloud task scheduling framework was found to be using 25% less energy based on the number of tasks, than that used by the conventional allocation procedures. This was done through designing a reward system which supported energy efficient VM's and at the same time made sure that tasks were assigned in such a way that they didn't overload the capacities of the VMs. The efficacy of the task assignments approach was about 100% and the system was programmed to reject unimportant tasks even if it was able to handle them when other tasks needed more attention. The Q-learning agents within the proposed framework were successfully well trained and achieved the trade-off of exploration and exploitation which helped the system for improving the scheduling decisions.

Compared to other conventional algorithms such as random allocation, and FCFS the proposed framework exhibited better results in terms of energy consumption, task allocation rate and time taken. The Q-learning helped avoid getting stuck at local optima and enabled a variety of improvements to occur faster due to the variety of information used regarding the scheduling of tasks. From these results we can infer that the framework has qualities that can enhance energy efficiency and utilization of resources in multi-cloud systems, providing a system that is more capable of handling dynamic workloads.

The current status of resource allocation for Virtual Machines (VMs) using the clouds is shown in Fig 1. It describes the current consumption of CPU and Memory in each VM and

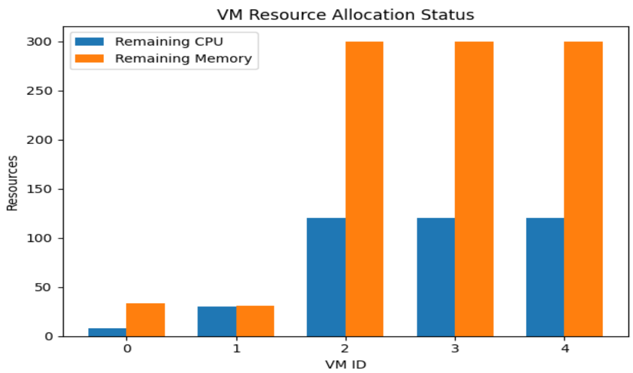

Fig. 2. *VM Resource Allocation Status*

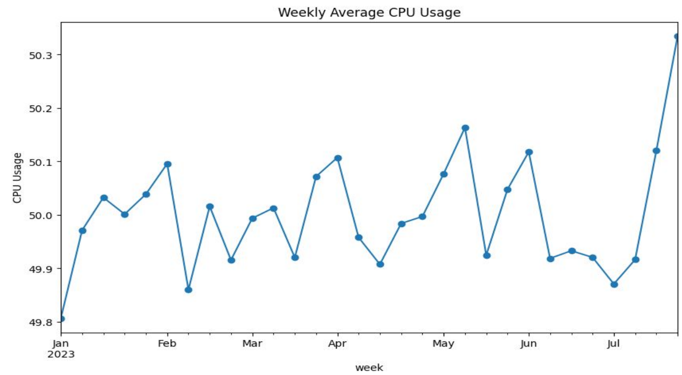

Fig. 4. *Weekly Average CPU Usage*

thus can demonstrate how resources are being deployed and partitioned with regard to the workload assigned. It is possible to track the changing dynamic of resource consumption as tasks are being executed involving respective resources.

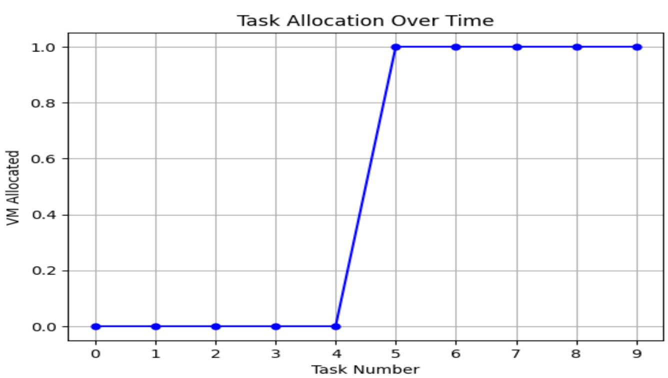

Fig. 3. *Task Allocation Over Time*

The graph shown in Fig 2 below illustrates the allocation of tasks over the period. It observes the distribution of tasks in different VMs in different clouds, how well the tasks are handled and what decisions of VM allocation the system took. Drawing from the graph the reader gets to understand how the system will respond to variable task requirements and variations in workload on different VMs with regard to energy usage as well as the time required to complete tasks.

A. Test Cases

The percentage of average CPU usage per week is shown in Fig 3 to demonstrate oscillations of confinement and release within the week. This is helpful in understanding the short term workload.

Fig 4 shows the averages of CPU used in a month and it brings out more general trends or cyclical patterns regarding the usage of the cloud resources. From this visualization it is easier to make a long-term forecast of the resource utilization as well as to set up a mid term plan of proper resource allocation.

Meanwhile, the year average CPU usage visually shows some yearly tendencies and CPU utilization changes during the year, so it helps to analyze the system performance growth and other problems on the yearly basis. By the way making the yearly usage graph based the dataset of the time stamp of 2023 have yield just a point.

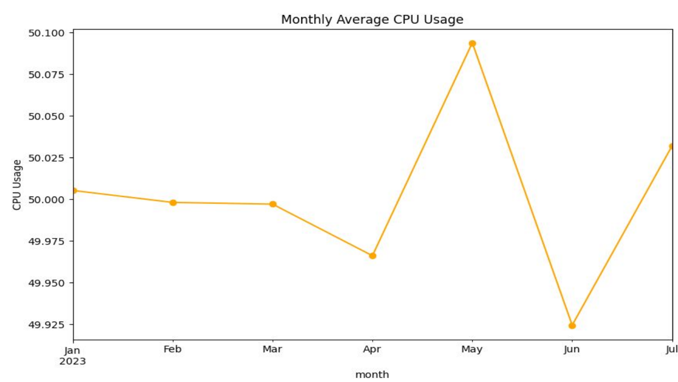

Fig. 5. *Monthly Average CPU Usage*

VII. Innovation and Social Impact

The proposed work itself presents a new framework for energy management based on Deep Reinforcement Learning (DRL) in cloud data centres. Unlike other scheduling techniques like round-robin or earliest-first, which are preventive techniques aimed at equal distribution of loads among the virtual machines, the proposed framework will incorporate a decision-making system in order to assign the jobs to the most economical and most suitable virtual machines as per the energy consumption as well as the workload demands.

A. Key Innovation

1) Integration of DRL with Experience Replay: Improves model learning through a new method of using past experiences to improve the scheduling decisions making thus achieving a better, efficient allocation of jobs.

2) Energy-Aware Scheduling Mechanism: Utilizes resources in the most efficient way possible by attempting to minimize energy used making this a solution to present day's major environmental and financial concerns.

3) Comprehensive Comparative Analysis: Shows that the presented heuristics outperform traditional approaches in terms of energy consumption and scheduling performance, providing solid groundwork for further investigation of cloud computing systems.

4) Scalable and Sustainable Design: Serves real-world cloud conditions based on cost efficient large-scale data centers to support sustainable computing.

VIII. Conclusion and Future Scopes

The proposed multi-cloud task scheduling framework using Q-learning is highly efficient in terms of minimizing energy consumption while also optimising resource use by constantly adapting to workload levels. The proposed method is more energy efficient and able to handle tasks better than the traditional method of random allocation and FCFS in cloud settings and offers a solution for large scale cloud computing. This work can also improve scalability by the need to employ higher reinforcement learning such as deep Q learning and multi-agent systems. The addition of three enablers they suggested such as real-time analytics, hybrid cloud, and energy-aware machine learning could enhance the utilization and the dependability and efficiency of the system.

References

[1] Reffad, Hamza, and Adel Alti. "Semantic-based multi-objective optimization for qos and energy efficiency in iot, fog, and cloud erp using dynamic cooperative nsga-ii." Applied Sciences 13, no. 8 (2023): 5218.

[2] Jayanetti, Amanda, Saman Halgamuge, and Rajkumar Buyya. "Multi-Agent Deep Reinforcement Learning Framework for Renewable Energy-Aware Workflow Scheduling on Distributed Cloud Data Centers." IEEE Transactions on Parallel and Distributed Systems (2024).

[3] Wang, Xiaohan, Lin Zhang, Yongkui Liu, Feng Li, Zhen Chen, Chun Zhao, and Tian Bai. "Dynamic scheduling of tasks in cloud manufacturing with multi-agent reinforcement learning." Journal of Manufacturing Systems 65 (2022): 130-145.

[4] Zhang, Tianrong, Fan Wu, Zeyu Chen, and Senyang Chen. "Optimization of Edge Cloud Collaborative Computing Resource Management for Internet of Vehicles Based on Multi-Agent Deep Reinforcement Learning." IEEE Internet of Things Journal (2024).

[5] Ramamoorthi, Vijay. "AI-Driven Cloud Resource Optimization Framework for Real-Time Allocation." Journal of Advanced Computing Systems 1, no. 1 (2021): 8-15.

[6] Abbasi, Mahmoud, Amin Shahraki, Md Jalil Piran, and Amir Taherkordi. "Deep Reinforcement Learning for QoS provisioning at the MAC layer: A Survey." Engineering Applications of Artificial Intelligence 102 (2021): 104234.

[7] Xiao, Yang, Yuqian Song, and Jun Liu. "Collaborative multi-agent deep reinforcement learning for energy-efficient resource allocation in heterogeneous mobile edge computing networks." IEEE Transactions on Wireless Communications (2023).

[8] H. Zhang, J. Li, Z. Qi, A. Aronsson, J. Bosch and H. H. Olsson, "Multi-Agent Reinforcement Learning in Dynamic Industrial Context," 2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC), Torino, Italy, 2023.

[9] Chniter, Hamza, Yuting Li, Mohamed Khalgui, Anis Koubaa, Zhiwu Li, and Fethi Jarray. "Multi-agent adaptive architecture for flexible distributed real-time systems." IEEE Access 6 (2018): 23152-23171.

[10] Jung, Soyi, Won Joon Yun, MyungJae Shin, Joongheon Kim, and Jae-Hyun Kim. "Orchestrated scheduling and multi-agent deep reinforcement learning for cloud-assisted multi-UAV charging systems." IEEE Transactions on Vehicular Technology 70, no. 6 (2021): 5362-5377.

[11] Saif, Mufeed Ahmed Naji, S. K. Niranjan, Belal Abdullah Hezam Murshed, Hasib Daowd Esmail Al-Ariki, and Hudhaifa Mohammed Abdulwahab. "Multi-agent QoS-aware autonomic resource provisioning framework for elastic BPM in containerized multi-cloud environment." Journal of Ambient Intelligence and Humanized Computing 14, no. 9 (2023): 12895-12920.

[12] R. Sun, J. Yan and F. Ren, "A Knowledge Graph-based Interactive Recommender System Using Reinforcement Learning," 2022 Tenth International Conference on Advanced Cloud and Big Data (CBD), Guilin, China, 2022.

[13] J. Sohal, "The Architecture for End-to-End 5G Network Slice Resource Management and Orchestration," 2023 2nd International Conference on Futuristic Technologies (INCOFT), Belagavi, Karnataka, India, 2023.

[14] S. Tiwari and B. B. M., "A Neighborhood Inspired Multiverse Scheduler for Energy and Makespan Optimized Task Scheduling for Green Cloud Computing Systems," in IEEE Access, vol. 12, pp. 157272-157298, 2024, doi: 10.1109/ACCESS.2024.3484388.

[15] X. Wang, X. Xing, P. Li and S. Zhang, "Optimization Scheme of Single-Objective Task Offloading with Multi-user Participation in Cloud-Edge-End Environment," 2022 IEEE Smartworld, Ubiquitous Intelligence Computing, Scalable Computing Communications, Digital Twin, Privacy Computing, Metaverse, Autonomous Trusted Vehicles, Haikou, China, 2022, pp. 1166-1171.