Multi Objective Reinforcement Learning for Cloud Resource Allocation

Hiba Qasim[1], Huda Abdulaali[2] and Soukaena Hasan[2]
[1]*Department of Computer Science, College of Science, Al-Mustansiriyah University, 10052 Baghdad, Iraq*
[2]*Department of Computer Science, College of Computer Science, University of Technology, 10066 Baghdad, Iraq*
hibaqasimjaber@uomustansiriyah.edu.iq, huda_it@uomustansiriyah.edu.iq, Soukaena.h.hashem@uotechnology.edu.iq

Keywords: Cloud Computing, Resource Allocation, Reinforcement Learning Q-learning, Makespan, Load Balancing and Cost Optimization.

Abstract: Cloud computing has become an essential technology for providing scalable, on-demand services. Cloud computing is a way to use computing resources (like servers, storage, databases, software, and networking) over the internet on demand, instead of buying and managing everything yourself. Cloud computing means you use computers (servers) that belong to a company and are located somewhere else (in a data center). However, appropriate resource allocation remains a significant difficulty. Traditional techniques frequently fail to handle the dynamic and complicated nature of multi-cloud settings, resulting in poor performance and higher costs. This paper presents a new technique based on reinforcement learning and using Q- learning to optimize resources allocation. To optimize resource allocation by achieving three primary goals: reduced makespan, reduced operating costs, and promising load balance among virtual machines. Evaluations were conducted using the Cloud SIM simulator and real-world traces from a dataset. The findings show that the method significantly improve resource usage, enhances load balancing by up to 12%, and increases make span stability as virtual machines grow. It is noteworthy that cost remains independent of virtual machine scalability, demonstrating the model's effectiveness in resource management without incurring extra costs.

1 INTRODUCTION

Cloud computing, a technology that may offer a range of services in response to customer requests via the Internet [1]. The business filed makes extensive use of development cloud computing, which provides a wide range of services in marketing, technology, and other disciplines [2]. One important part of a cloud architecture is a broker, which manages end-user requests and uses a scheduling algorithm to route them to the most pertinent virtualized resources [3], [4]. Allocating resources is one of the hardest parts of cloud computing [5]. Resource allocation in cloud- computing refers to the process of distributing and managing the resources that are present on a computer system [6]. Now a days, to transfer and exchange information using the communication mobile devices or computers are used to send/receive images, audio, and videos. So, it should be handled carefully [7]. Virtualization is the primary foundation of cloud computing; virtual resources, such as virtual computers, are frequently leased to clients [8]. Task-scheduling is employed for the purpose of allocating user's requests to the best available resources [9]. The traditional techniques may lack the suppleness needed to address the numerous and variable needs of current applications, resulting in poor performance and increasing operational expenses [10]. reinforcement learning (RL) is type of machine learning [11]. It has been demonstrated that one of the most effective ways to allocate resources for technology and communication is through reinforcement learning (RL) [12]. The primary contributions of this study are summarized below:

1) Model provides three objective processes; makespan, Cost and Load Balance.
2) Model using Q-learning, decision-making processes with highest cumulative rewards.
3) NASA dataset was used to evaluate the effectiveness of the proposed method.

2 RELATED WORK

The challenge of resource allocation in cloud computing has received significant scientific attention, with several studies helping to solve a problem. This part, will cover the relevant work on cloud resource- allocation

In this study, technique is proposed for efficient resource allocation for high-quality cloud software services at low cost. a model combines a Q-learning algorithm with many machines learning. The PCRA technique can manage resource allocation processes with an accuracy of 93.7% [13]. This research suggested a dynamic Q-learning system, hence lowering the usage of energy, makespan duration, and resource allocation efficiency. The suggested method achieves a 20% improvement over previous experiments [14].

This article proposes a unique resource- allocation approach depend on reinforcement-learning and intelligent (multi-agent) systems (IMARM). It improves cloud resource allocation performance by combining the Q-learning approach with multi-agent characteristics [15].

The proposed approach improves resource allocation, VM throughput, and load balancing by taking into account makespan, cost, and resource usage. This approach combines a Q-learning algorithm with an Artificial-Bee-Colony Algorithm (ABC) [16].

This article suggests a hybrid resource provisioning method that blends autonomous computing and reinforcement learning (Q-learning) for cloud applications. the proposed effort aims to lower the total cost of providing cloud resources. Violations of service level agreements have decreased, response times have decreased, and resource utilization has increased as a result of the suggested approach [17].

The current research utilizes reinforcement learning techniques to enable cloud computing environment providers to optimally allocate resources under unpredictable and vary workloads. This framework employs a Q-learning algorithm based on fuzzy logic, which can continuously increase the capacity of virtual machines (VMS) in a cloud computing data center to meet service level agreements [18].

This article applies Deep Reinforcement Learning (DRL), to optimize resource allocation across many cloud platforms. The proposed system aims to manage a variety of workloads while reducing latency and increasing cost effectiveness [19].

The Actor-Critic-Based Computer-Intensive Workload-Allocation Scheme ((AC)-(CIWAS)) is presented in this work. AC-CIWAS continually captures the server's dynamic properties and assesses how various workloads influence energy consumption in order to accomplish task allocation. With QoS assurance, the suggested approach (AC-CIWAS) can reduce the assigned task's load by about 20% [20]. The present research develops two resource allocation strategies: cost-effective with private cloud (CERAI) and cost-effective with public cloud (CERAU) that employ deep deterministic policy gradients and P-DQN. The results of the investigation suggest that CERAI and CERAU can effectively reduce operating costs in a variety of challenging circumstances [21].

3 PROPOSED WORK

Q-learning is the method suggested in this propose system. It belongs to the Reinforcement Learning (RL) technique. A comprehensive method is employed to ensure that cloud systems learn and generalize in a range of situations and that resources are allocated appropriately. Figure 1 illustrates the Q-learning algorithm's operation.

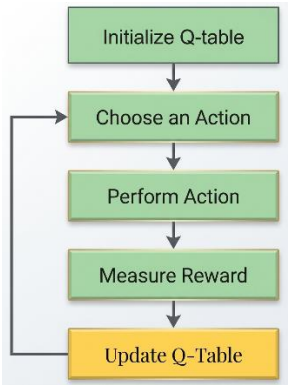

Figure 1: Q-learning algorithm process.

There are actions specific to each state in the cloud system environment, S= [S1, S2, ……. Sn]. At the time (t) in each state St ∈ S, the A= [a1, a2, a3,……am] the Q-learning's agent selects an action at ∈A to proceed to the following state st+1 ∈ S via the environment rewards the transition process with rt+1.

The basic purpose of determining the optimal policy in cloud computing is to accomplish tasks by taking the best course of action that enhances the Q-value of all states. (Q-value) can be determined using (1):

$$Qt + 1(st,at) = Q(st + at) + \alpha[rt + \Upsilon max\ a't\,[Qt\,(\delta\,(st + at)\ ,a't\,] - Qt\,(stat)\ldots$$

$Qt + 1(st,at)$ ….. new Q-value,
$Q(st + at)$ …… current Q-value. (1)

$max\ a't\ [Qt(\delta\ (st + at), a't\ ]$ ... helps agent predict the best future action.

The learning rate is defined as α, the discount factor and influence of the preceding action on the next state are defined by ϒ (0 < ϒ < 1), and the punishments or rewards given for acts in the current state st are denoted by rt [16]. One potential cloud computing operation is balancing load, which is explained by Algorithm 1 and allocates resources among virtual machines.

Algorithm 1: Q-Update (S):

1) Assign inputs to the discount factor ϒ, reward r, and the learning rate α;
2) Set initial Q-values;
3) for i= 1to n where (n) is the state's number;
4) for j=11to p where (p) is the actions number in each state;
5) Q (si,aj) = 0;
6) end -for;
7) end -for;
8) Execute an action ai from A = [a1, a2, a3,……am] and implement to the next state (st+1);
9) Compute the learning rate value;
10) Calculate reward value;
11) Update (Q t+1) by eq. (1);
12) Repeat this procedure for each new state until convergence is achieved.

The proposed model has M PMs as P ={ PM1,PM2,..,PMM}. Every PM uses L VMs as VM ={ VM1,VM2,….,VML}. The cloud environment includes S different user tasks, represented by T = { T1,T2,…,TS}. The resources of the host computer, including the CPU, RAM, and hard disk space, are used by AVM. For execute the user request, the available virtual machine's resources are needed. The mapped tasks will execute simultaneously to complete the assignments task when they are allocated to one virtual machine (VM) or several VMs, as illustrated in Figure 2. When a user sent many requests to the propose system, the ask is transferred to the cloud-broker.

Users submit requests via the internet, which are kept in VMs. This procedure is dependent on the scheduling policy's performance (data broker). Hypervisors allow for the operation of several virtual machines (VMs) on the same hardware layer. After the tasks are sent to the broker, which in turn distributes the tasks to the virtual machines in a way that ensures efficient load distribution, maximizing the use of resources at the lowest cost, and providing the fastest response time. After the tasks are executed, they are sent back to the broker to execute the user's request. Figure 2 shown the proposed system architected.

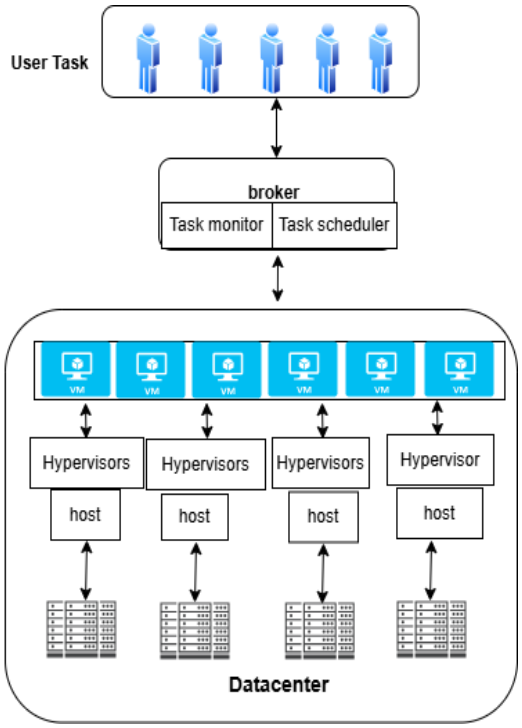

Figure 2: The proposed system- diagram.

4 EVALUATION AND ISCUSSION OF EXPERIMENTS

This section explains the experiment's findings and parameter settings.

4.1 Cloud-Simulation-Environment

An experiment evaluating the performance of the suggested method (Q-Learning) is presented in this section. The CloudSim 3.0.3 was used to construct and develop the simulation in this work. The PC used to simulate this experiment had an Intel Core i7-8565U CPU running at 1.8 GHz and 8 GB of RAM. The experiment's simulation environment has been created as indicated in Table 1.

Table 1: The simulation environment features.

Type	Parameter	Value
Host	host no.	8
VM	No. of vm VmScheduler	100-400 Time Shared
Datacenter	No. of datacenter	4
Cloudlet	No. of Cloudlet	100-600

4.2 Datasets

This work using NASA dataset that consist of 42,264 jobs are thoroughly discussed, including user groups, resource use, and start and submit times.

4.3 Experimental Results

In this section presents a comparative analysis of reinforcement learning (Q-Learning) algorithm across different number of task and VM by highlighting key performance metrics encompassing multi-objective such as (makespan, load balance and cost).

1) Makespan. This measures the amount of time needed to finish tasks. Table 2 compares performance for different task counts (100–600) from the dataset and employing 100 VM.

Table 2: Makespan of various task counts (100-600).

Task	Makespan
100	43.90
200	79.35
400	79.67
600	80.74

When the task size increased from 100 to 200, the commitment increased from 43.90 to 79.35 (an increase of nearly 80%). However, when the task size increased again (200 to 400), only a small fraction was committed (79.35 to 79.67), and even when the task size increased to 600, only 80.74, while vm size stable at 100vm. Figure 3 show represent of make span of task size increased from 100 to 600.

We notice an increase in the load from (5.59 to 11.68) when use (100-200 task), load increased 109.8% approximately, while load rise from (11.68 to 15.11) when use (400 task) load increased approximately to 29%. Finaly, the load becomes heavy in 600 tasks around 23.65. Figure 4 shows the different load balance rate from different no. of task.

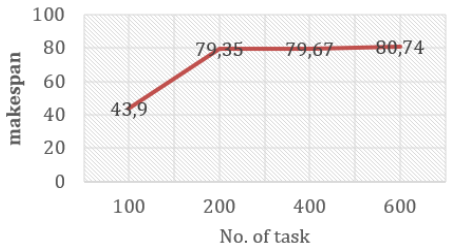

Figure 3: Show represent of makespan for different cloudlet.

Table 3 shows an analysis of the makespan with varied VM counts (100 to 400) and implementing 200 tasks.

Table 3: Makespan for varying VM count (100 to 400).

No. of vm	Makespan
100	79.35
200	79.55
300	79.55
400	79.55

The makespan increases little from 79.35 to 79.55 when the number of virtual machines (VMs) rises from 100 to 200. Even the total number of (VMs) is increased was made for the 300 and 400, the makespan stays constant at about 79.55 after 200.

2) Load balance. Is one of the key approaches used to guarantee an equal allocation of workload and effective resource use. Table 4 demonstrates a comparison between performance for multiple task counts (100-600) with 100 VM.

Table 4: Load balance with various task counts (100-600).

No. of Task	Load balance
100	5.56
200	11.68
400	15.11
600	23.65

Table 5 shown Comparison of the load balance by varying VM count (100 to 400) and using 200 tasks.

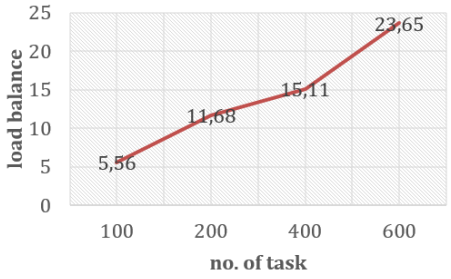

Figure 4: Different load balance rate.

Table 5: Load balance by varying VM.

No. of vm	Load balance
100	11.68
200	8.80
300	7.32
400	6.40

This table shows the load variation for several different virtual machines (100-400vm), unlike the previous1 table, which varied the workload, load balance improves when vm increase 400 vm show best value at 6.40 (12) % improvement compare to 300vm). This indicates that scaling resources effectively enhances the system's ability to distribute workload evenly, leading to more efficient operation.

3) Cost. Cost words refer to the cost of the request that will be handled by the task. It can be computed using CPU cost, memory consumption cost, bandwidth cost, and storage utilization cost. Table 6 displays the results of 100 vm and varying number of assignments task.

Table 6: Result of multi no. of task.

No. of Task	Cost
100	624.70
200	1825.18
400	3832.30
600	7607.64

The table shows the cost for multiple task size (100-600) while the virtual machines are fixed at 100. The increase accelerates with increasing tasks, From 100 to 200 tasks, the cost increases from 624.70 to 1825.18. From 200 to 400 tasks, the cost increases from 1825.18 to 3832.30. From 400 to 600 tasks, the cost increases from 3832.30 to 7607.64. indicating that the system or data requires more resources as tasks increase. Effective resource management and optimization are essential to maintain cost efficiency at larger scales. Figure 5 below shows various cost rate with different task size.

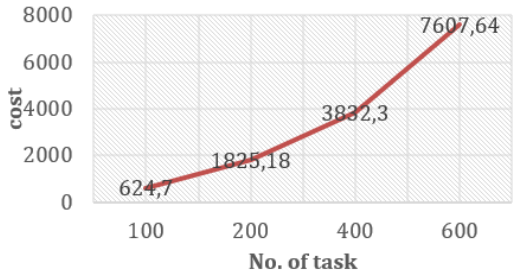

Figure 5: Cost of different cloudlet size.

Therefore, we will increase the number of virtual machines, and note the difference in the following table using 200 tasks as illustrate in Table 7.

Table 7: Cost for different vms size (100-400).

No. of vm	Cost
100	1825.18
200	1825.18
300	1825.18
400	1825.18

The following table shows that the cost remains unchanged regardless of the increase in the number of virtual machines (100-500). Increasing the number of machines does not affect the total cost, which is useful for scalability without increasing costs.

5 CONCLUSIONS

In this paper, we proposed a reinforcement learning-based resource allocation technique in cloud computing (Q- learning). The proposed study aims to tackle cloud-related difficulties such as cost reduction, load balancing among virtual machines, and makespan improvement. The result indicates a significant improvement in load distribution as the number of virtual machines increases, with load balance values decreasing from 11.68 (at 100 VMs) to 6.40 (at 400 VMs). The 12% improvement observed at 400 VMs compared to 300 VMs demonstrates that scaling resources enhances workload distribution, leading to more stable and efficient system performance.

The results show that increasing the number of VMs does not necessarily increase operational costs proportionally, implying the model's ability to manage resources efficiently without incurring additional expenses. The approach's adaptability ensures that resource allocation remains optimal even as the environment's complexity and workload volume grow. In general, involving reinforcement learning into the resource allocation process for cloud computing is a promising technique to controlling cloud computing complexity, which leads to higher service quality (QoS) and user-satisfaction. Future work will focus on integrating reinforcement learning with meta-heuristic algorithms.

ACKNOWLEDGEMENT

The authors of this paper are thankful to Mustansiriyah- University https://uomustansiriyah.edu.iq/) in Baghdad, Iraq, for providing support with the current work.

REFERENCES

[1] G. Zhou, W. Tian, R. Buyya, R. Xue, and L. Song, "Deep reinforcement learning-based methods for resource scheduling in cloud computing: A review and future directions," arXiv preprint arXiv:2105.04086, 2021, doi: 10.48550/arXiv.2105.04086.

[2] S. K. Pradhan, S. K. Bisoy, S. Kautish, M. B. Jasser, and A. W. Mohamed, "Intelligent decision-making of load balancing using deep reinforcement learning and parallel PSO in cloud environment," IEEE Access, vol. 10, pp. 76939-76952, 2022.

[3] D. R. Abdulrazzaq, N. M. Shati, and H. K. Hoomod, "Bi-objective task scheduling based on heuristic initialization of the jellyfish search algorithm in cloud computing," in Proc. 3rd Int. Sci. Conf. Eng. Sci. (ISCES), 2023, pp. 25-30.

[4] P. S. Prabha and A. Rengarajan, "Adaptive cloud resource allocation using attention-driven deep reinforcement learning," Eng. Technol. Appl. Sci. Res., vol. 15, no. 2, 2025.

[5] H. Hallawi, "Development a framework for validating multi-capacity cloud resource allocation based on SimGrid," in AIP Conf. Proc., vol. 2144, no. 1, p. 050012, 2019.

[6] S. A. Jabber, S. H. Hashem, and S. H. Jafer, "Task scheduling and resource allocation in cloud computing: A review and analysis," in Proc. 3rd Int. Conf. Emerg. Smart Technol. Appl. (eSmarTA), 2023, pp. 1-8.

[7] H. Mao, M. Alizadeh, S. Shakkottai, and R. Katz, "Resource management with deep reinforcement learning," in Proc. 15th ACM Workshop, 2016, doi: 10.1145/3005745.3005750.

[8] S. M. Rozehkhani, F. Mahan, and W. Pedrycz, "Efficient cloud data center: An adaptive framework for dynamic virtual machine consolidation," J. Netw. Comput. Appl., vol. 226, p. 103885, 2024.

[9] L. Abualigah, A. M. Hussein, M. H. Almomani, R. A. Zitar, H. Migdady, A. I. Alzahrani, and A. Alwadain, "Improved synergistic swarm optimization algorithm to optimize task scheduling problems in cloud computing," Sustain. Comput. Inform. Syst., vol. 43, p. 101012, 2024.

[10] R. Khan, "Dynamic load balancing in cloud computing: Optimized RL-based clustering with multi-objective optimized task scheduling," Processes, vol. 12, no. 3, p. 519, 2024.

[11] R. Buyya and A. V. Dastjerdi, Cloud Computing: Principles and Paradigms, John Wiley & Sons, 2024.

[12] Z. Xu, Y. Wang, J. Tang, J. Wang, and M. C. Gursoy, "A deep reinforcement learning based framework for power-efficient resource allocation in cloud RANs," in Proc. IEEE Int. Conf. Commun. (ICC), 2017, pp. 1-6.

[13] X. Chen, F. Zhu, Z. Chen, G. Min, X. Zheng, and C. Rong, "Resource allocation for cloud-based software services using prediction-enabled feedback control with reinforcement learning," IEEE Trans. Cloud Comput., vol. 10, no. 2, pp. 1117-1129, 2020.

[14] Muthusamy and R. K. Dhanaraj, "Dynamic Q-learning-based optimized load balancing technique in cloud," Mob. Inf. Syst., vol. 2023, p. 7250267, 2023.

[15] S. Belgacem, S. Mahmoudi, and M. Kihl, "Intelligent multi-agent reinforcement learning model for resources allocation in cloud computing," J. King Saud Univ. Comput. Inf. Sci., vol. 34, no. 6, pp. 2391-2404, 2022.

[16] Kruekaew and W. Kimpan, "Multi-objective task scheduling optimization for load balancing in cloud computing environment using hybrid artificial bee colony algorithm with reinforcement learning," IEEE Access, vol. 10, pp. 17803-17818, 2022.

[17] R. Panwar and M. Supriya, "RLPRAF: Reinforcement learning-based proactive resource allocation framework for resource provisioning in cloud environment," IEEE Access, vol. 12, pp. 95986-96007, 2024.

[18] T. Thein, M. M. Myo, S. Parvin, and A. Gawanmeh, "Reinforcement learning based methodology for energy-efficient resource allocation in cloud data centers," J. King Saud Univ. Comput. Inf. Sci., vol. 32, no. 10, pp. 1127-1139, 2020.

[19] F. Varghese, Dynamic Resource Allocation in Multi-Cloud Environments Using Reinforcement Learning, Ph.D. dissertation, National College of Ireland, Dublin, Ireland, 2025.

[20] M. Arvindhan and D. R. Kumar, "Adaptive resource allocation in cloud data centers using actor-critical deep reinforcement learning for optimized load balancing," Int. J. Recent Innov. Trends Comput. Commun., vol. 11, no. 5s, pp. 310-318, 2023.

[21] J. Xu, Z. Xu, and B. Shi, "Deep reinforcement learning based resource allocation strategy in cloud-edge computing system," Front. Bioeng. Biotechnol., vol. 10, p. 908056, 2022.