International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
DOI: https://doi.org/10.17762/ijritcc.v11i10.8464
***Article Received:** 21 July 2023 **Revised:** 13 September 2023 **Accepted:** 28 September 2023*

Deep Reinforcement Learning Framework with Q Learning For Optimal Scheduling in Cloud Computing

Nihar Ranjan Sabat[1], Rashmi Ranjan Sahoo[2]
[1]Faculty of Engineering
Biju Patnaik University of Technology(BPUT)
Rourkela-769015,India
e-mail: n.ranjan9@gmail.com
[2]Department of Computer Science and Engineering
Parala Maharaja Engineering College(PMEC)
Berhampur-761003,India
e-mail: rashmiranjan.cse@pmec.ac.in

Abstract—Cloud computing is an emerging technology that is increasingly being appreciated for its diverse uses, encompassing data processing, The Internet of Things (IoT) and the storing of data. The continuous growth in the number of cloud users and the widespread use of IoT devices have resulted in a significant increase in the volume of data being generated by these users and the integration of IoT devices with cloud platforms. The process of managing data stored in the cloud has become more challenging to complete. There are numerous significant challenges that must be overcome in the process of migrating all data to cloud-hosted data centers. High bandwidth consumption, longer wait times, greater costs, and greater energy consumption are only some of the difficulties that must be overcome. Cloud computing, as a result, is able to allot resources in line with the specific actions made by users, which is a result of the conclusion that was mentioned earlier. This phenomenon can be attributed to the provision of a superior Quality of Service (QoS) to clients or users, with an optimal response time. Additionally, adherence to the established Service Level Agreement further contributes to this outcome. Due to this circumstance, it is of utmost need to effectively use the computational resources at hand, hence requiring the formulation of an optimal approach for task scheduling. The goal of this proposed study is to find ways to allocate and schedule cloud-based virtual machines (VMs) and tasks in such a way as to reduce completion times and associated costs. This study presents a new method of scheduling that makes use of Q-Learning to optimize the utilization of resources.The algorithm's primary goals include optimizing its objective function, building the ideal network, and utilizing experience replay techniques.

Keywords- Cloud Computing;Datacenter;Load Balancing;Scheduling;Virtual Machines.

I. INTRODUCTION

Cloud computing is standard for huge data sets [1]. Networks, servers, and resources make up cloud data centers. Global acceptance of cloud computing has increased due to its capacity to provide organizations with extensive storage and resources. Proper management, legislation, and security can successfully deploy these resources. Virtualization, scalability, autonomous system provisioning, and broad network access are cloud computing capabilities [2]. In recent years, cloud computing has advanced. It has improved computational skills across areas to meet client needs. Adding resources solves the problem easily. Many cloud computing applications need quick computer capacity growth to meet client requests. Resources can be added to solve the problem easily. This effort is impracticable due to its high costs. Literature suggests many solutions. To deploy resources efficiently, task scheduling can be refined [4]. To maximize resource utilization, load balancing is another option [1]. One suggestion is to run offline and online operations simultaneously to maximize resources [5].

Numerous heuristic techniques have been proposed to overcome the concerns. Examples are the first feet [6], [7], sample packing strategies [8], [9], and others. The ant colony algorithm [10] and genetic algorithm [9] are among many complex meta-heuristics. The effectiveness of heuristic techniques depends on resource demand patterns and manual checks and changes. If the environment changes, adjusting is needed.

Machine learning, especially reinforcement learning and deep neural networks, is advancing rapidly, offering new ways to solve the challenge. DeepRM uses image-encoded deep reinforcement learning for task planning [11]. Peng et al. [13] use Q-learning for task scheduling, while GoSu [12] uses graph

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
DOI: https://doi.org/10.17762/ijritcc.v11i10.8464
***Article Received:** 21 July 2023 **Revised:** 13 September 2023 **Accepted:** 28 September 2023*

convolution network (GCN).Active involvement with the environment is necessary to gather enough data for reinforcement learning training. Environmental changes or unexpected shocks could cause the trained model to fail due to its limited flexibility. This causes the system to perform poorly on a small sample and require significant retraining to get a policy that works under current conditions. Method is time-consuming. This study uses deep reinforcement learning techniques, particularly Q-learning, to optimize scheduling in CloudSim 4, a cloud computing simulation framework. In many cloud systems, this research optimizes resource allocation policies and performance. The research includes CloudSim 4 configuration, resource allocation policy design, and state and action space creation, deep neural network implementation for Q-values, Q-learning agent training, and algorithm integration into CloudSim 4 for realistic evaluation. This study introduces a novel approach by utilizing a ranking table generated through the reinforcement learning algorithm Q-learning to facilitate job scheduling.

A. Motivation

In order to handle a wide range of workloads, cloud computing platforms must efficiently allocate and schedule resources like virtual machines, containers, and storage. When compared to other scheduling algorithms, Deep Reinforcement Learning (DRL) is more cloud-friendly. In the cloud, workloads change throughout the day. By optimizing resources in response to changing workloads, DRL helps businesses meet and reduce waste. The power needs of cloud data centers are considerable. Energy efficiency can be increased in a number of ways through the use of optimal scheduling, including the consolidation of workloads, the shutdown of unused resources, and the foundation of decisions on actual need. It is important to balance response time, resource utilization, and cost when scheduling in the cloud. By modelling dependencies between objectives, DRL frameworks can optimize problems with multiple goals. It has the ability to learn from previous experiences and adapt accordingly. This enables it to outperform traditional heuristics at discovering optimal scheduling policies by learning from past results. Cloud computing often results in resource failures. A DRL framework can handle service interruptions gracefully due to its capacity for real-time resource reallocation. Thousands of concurrent resources and workloads are no problem for DRL. Because of this, it is an excellent option for massive cloud data centres.Machine learning models help DRL allocate and schedule resources proactively by predicting traffic and workload patterns.DRL can adjust its prices to reflect the current market. Businesses and service providers in the cloud stand to benefit from delivering faster, more reliable services at lower costs. In this case, DRL based scheduling can be helpful.

B. Contribution

The contribution of this work is to explore the issue of virtual machine (VM) allocation and task scheduling inside a cloud environment, aiming to minimize the time required to complete jobs and the corresponding costs. This research introduces a unique scheduling strategy that employs Q-Learning to enhance the efficacy of resource utilization summarized below:

- Optimization of objective function
- Construction of the desired network
- Utilization of experience replay approaches

C. Paper Organization

This paper is structured into five distinct sections, where the second section provides a thorough analysis of the current literature pertaining to the diverse algorithms employed in task scheduling. The third section of the paper presents a comprehensive analysis of the methodology and theoretical frameworks employed in the study. The fourth section of this study presents a comprehensive analysis of the findings obtained through the application of Q-learning to the task scheduling problem in cloud computing. The research concluded in section five, wherein a concise summary of the noteworthy findings was provided.

II. LITERATURE SURVEY

The task scheduling issue draws researchers. Two work scheduling strategies exist. Heuristic and meta-heuristic algorithms are traditional. Also, DRL-based strategies. Traditional methods are being improved. Pradhan et al. [14] proposed a round-robin resource allocation method to reduce wait time and accommodate client needs. Geography and energy-conscious load-balancing by the DGLB reduce data center energy consumption [15]. Networks balance power and workloads. Smart grid infrastructure uses energy storage to integrate renewable energy. Researchers use incentives.Webservice composition using linear programming was proposed by Ghobaei-Arani et al. [16]. A distributed environment uses LPWSC to choose the best cloud service for individual requests. For better service.

The control MAPE (monitor, analyze, plan, execute) loop paradigm was proposed by Ghobaei-Arani et al. [17] for autonomic resource provisioning. An innovative dimensionality reduction method by Megh [18] maps a cumulatively expansive action-state space to a polynomial-dimensional space. We simulate the Markov decision process of performance-efficiency resource management and energy. This shrinks problem representation. It dedimensionalizes space. Kumar et al. developed PSO-COGENT [19], inspired by particle swarm.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
DOI: https://doi.org/10.17762/ijritcc.v11i10.8464
***Article Received:** 21 July 2023 **Revised:** 13 September 2023 **Accepted:** 28 September 2023*

The algorithm presented in this study improves time efficiency, execution costs, and cloud data center energy consumption. APSOVI addresses PSO prematurity and divergence. This method improves search control with an average ideal nonlinear velocity. Jin et al. [20] used genetic algorithms to optimize reservation selection and increase dispatch probability. This biological evolution-inspired method maximized delivery success. Cloud-based applications improve energy efficiency and response times.

Medara et al. [21] developed WWO, a meta-heuristic, to balance cloud energy consumption and performance. This method disables inactive hosts. In multidimensional optimization problems, the WWO algorithm efficiently finds near-optimal solutions. These methods depend on mathematical models or expert knowledge, which is a drawback. Cloud computing often uses DRL-based methods.

QEEC [22] schedules using a two-phase separation process. First-generation queuing models use M/M/S. After that, a Q-learning scheduler assigns virtual machine jobs. This method speeds task response and server CPU use. RLTS [23] schedules tasks quickly with a deep Q network. Rewards depend on makespan when an action is completed at state s and transitioned to s'. A job execution time-based reward function improved QoS, response time, and costs in deep Q-learning networks by Cheng et al. [24], [25], and [26]. Deep Q learning was used by Wei et al. [27] to create a scheduler that could assign tasks without prior knowledge. Huang et al. [28] used deep Q learning to add adversarial imitation learning to cloud scheduling. The agent is given a good scheduling policy by imitation learning. Edge-mobile computing deep reinforcement learning multiagent MADRL [29] has lower calculated latency and better channel access using actor-critic. In DeepRM_Plus [30], Guo et al. encapsulated the resource management model with CNN. To reduce power consumption, Liu et al. [31] added a model-free RL power manager and workload predictor-based LSTM to the local tier. Xu et al. [32] also used a Deep Neural Network (DNN) to transform the resource allocation problem into a convex optimization problem and estimate the action-value function.

Models predict big energy savings. Due to data requirements, Deep Reinforcement Learning (DRL) training takes time. To adapt to new conditions, retrain the policy. Training may ignore the previous policy, lowering sample task performance.

III. MATERIALS AND METHODS

The Java-based cloud simulation framework CloudSim is used by researchers to evaluate the proposed approach. Cloud infrastructure is created and customized using the framework. In this work, Java was used to implement deep reinforcement learning in a virtual cloud. The user tasks came from PAMAP2. Four 500 MB RAM, 1000 MB storage Virtual Machines were used during our simulations.

This research is systematic. The simulation environment will accurately replicate cloud architecture and facilitate resource allocation algorithms using CloudSim 4 APIs. State and action spaces will be created using cloud infrastructure and allocation policies. Tensor Flow improves information acquisition by estimating Q-value. CloudSim 4 and deep learning library APIs will define and train Q-learning. CloudSim 4's Q-learning algorithm will validate the schedule optimization method.

A. Centralized task allocation

Depending on size, a cloud data center can have a few thousand to several hundred thousand service nodes. To distribute jobs effectively, a centralized dispatcher is needed due to customer needs and cloud data centers' decentralization. It is believed that cloud data centers use a global request queue to buffer and manage user requests. Task dispatchers efficiently monitor and process user requests and route them to physical servicing nodes. The following section will justify a cloud data center centralized task dispatcher deployment scientifically.

B. Q-learning-based task scheduling

Every physical server-like service node has a local request queue. This queue holds node-specific user requests. Requests are dynamically reorganized by SLA service levels. Q-learning scheduling assigns virtual machines jobs to reduce CPU use. It improves system performance. In the proposed study, Reinforcement Learning builds the Q table. Sorting in descending order and the Q-Learning algorithm help schedule rewards. Proposed research flowchart is shown in Figure 1.

C. Reinforcement Learning

Figure 2 illustrates the evolutionary trend of reinforcement learning. The Markov decision process (MDP) is a commonly employed methodology in the domain of reinforcement learning (RL) for tackling the fundamental challenge of establishing sequential behavior [33]. The incorporation of the value function into the agent's framework is a crucial characteristic that has a strong correlation with the Bellman equation.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
DOI: https://doi.org/10.17762/ijritcc.v11i10.8464
Article Received: *21 July 2023* ***Revised:*** *13 September 2023* ***Accepted:*** *28 September 2023*

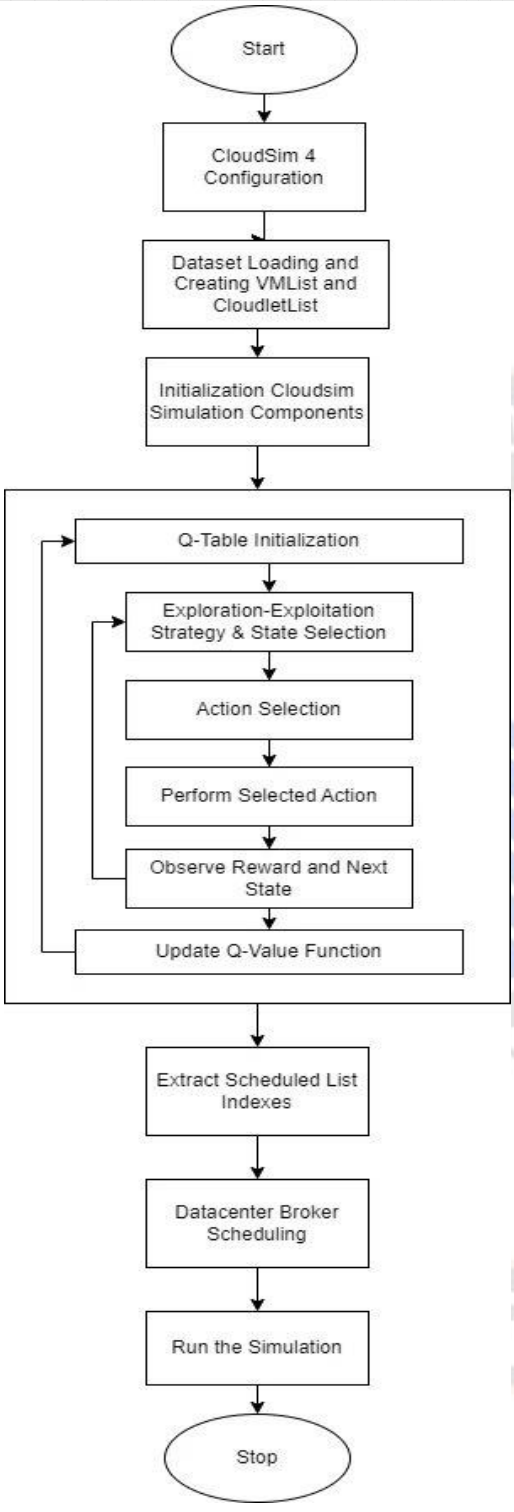

Figure 1. Flowchart of the proposed work

D. Pseudo Code of Proposed Research

Step 1:
CloudSim 4 Configuration
Set up the cloud infrastructure, including the data center, hosts, and virtual machines
Define initial values for simulation parameters such as host memory, storage, and bandwidth
Step 2:
Dataset Loading, Creating VMList and CloudletList
Load the PAMAP2 dataset
Create an empty VMList
Create an empty CloudletList
for each data entry in the PAMAP2 dataset
  Create a new VM based on the data entry
  Add the VM to the VMList
  Create a new Cloudlet based on the data entry
  Add the Cloudlet to the CloudletList
end for
Step 3:
Initialization
Initialize Q-table with random values
Set learning rate (α) and discount factor (γ)
Set maximum episodes and steps per episode.
Step 4: Q-Learning Algorithm
for episode = 1 to maximum number of episodes
  Reset the environment
  Set initial state (s)
  for step = 1 to maximum number of steps per episode
Choose action (a) utilizing an exploration-exploitation strategy (e.g., epsilon-greedy)
    Perform action (a) and analyze the reward (r) and the next state (s')
    Update the Q-value function utilizing the Q-Learning update equation:
Q(s, a) = Q(s, a) +
        α * (r + γ * max_a'(Q(s', a')) - Q(s, a))
Update the current state (s) to the next state (s')
if the episode is finished (reached a terminal state)
      break
  end for
end for
Step 5: Extract Scheduled List Indexes
Set up an array for cloudlet reward scores
for each cloudlet in the CloudletList
  Calculate the reward score based on the Q-table:
  reward_score = Q(state, action)
  Store the reward score in the array
end for
Sort rewards descending and extract indexes.
Step 6: Datacenter Broker Scheduling
Create the Datacenter Broker
Extract indexes to assign cloudlets to virtual machines.
Step 7: Run the Simulation
Run CloudSim 4 with scheduled cloudlets.

The Markov decision process (MDP) is a fundamental concept in the field of reinforcement learning, which addresses the problem of sequential decision-making. The formulation of the Bellman equation is facilitated by utilizing the value function and Markov Decision Process (MDP) within the context of reinforcement learning. The following part will discuss the utilization of Q-learning in addressing the task of solving the Bellman equation. In order to optimize the efficacy

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
DOI: https://doi.org/10.17762/ijritcc.v11i10.8464
***Article Received:** 21 July 2023 **Revised:** 13 September 2023 **Accepted:** 28 September 2023*

of reinforcement learning, it is advisable to employ a proficient approach that effectively addresses the Bellman equation [34].

E. Markov Decision Process

The formal characterization of the problem of sequential action choice is represented by a mathematical construct known as a Markov Decision Process (MDP). The usage of a random number generator is necessary for determining compensation and selecting the transition state after completing an activity, due to the inherent unpredictability of the environment. Policies are a set of defined principles that dictate the selection of actions to be undertaken within a particular state or context. The formal notation utilized for characterizing reinforcement learning approaches is the Markov Decision Process (MDP) [35].

1) State

The state of observable agent states is represented by a set S. The phrase "state" means "observation of your situation" [36].

2) Action

An action is a group of possible actions A in a specific state S. Most of the time, an agent's activities are the same throughout all states. A single set of A is defined as a result [37].

3) State Transition Probability Matrix

An agent's move from a unit state S to another S' when acting A is represented mathematically by the STA (state transition probability). The subsequent MDP compensation and states are purely dependent on the present state and activities. As a result, [38] provides the possibility that the following condition will be compensated by the following compensation as well as its quantity. The likelihood is:

$$P_{ss'}^{a} = P[S_{t+1} = s' \mid S_t = s, A_t = a] \quad (1)$$

Where (1) $P_{ss'}^{a}$ is the probability comprised in the matrix P of shifting to state s', when action a is done in state s, where t signifies the time.

4) Reward

The reward is knowledge that is provided to the agent in the environment in order to ensure that agent can learn. The agent receives the following reward when the action is a and the state is s at time t:

$$R_s a' = E[R_{t+1} \mid S_t = s, A_t = a] \quad (2)$$

Where (2) $R_s a'$ is the reward function's definition. The variable t marks the time when action a takes place and the system transitions from state a to state s', while the variable E represents the reward's expected value. The agent can express the compensation value as an expected value since, depending on the environment, it may result in various rewards even when the same task is completed under identical conditions. When the agent is in state S, the environment interacts with it and does action A to inform it of the reward it wishes to deliver and the next state S' that it will enter. The environment alerts the agent at time t+1. Consequently, Rt + 1 denotes the compensation to be acquired by the agent.

5) Discount Factor

The discount factor (DF) was conceptualized as a purposeful strategy to reduce the adverse consequences arising from compensating actions. Upon the agent's successful completion of their specific obligations within each jurisdiction, they

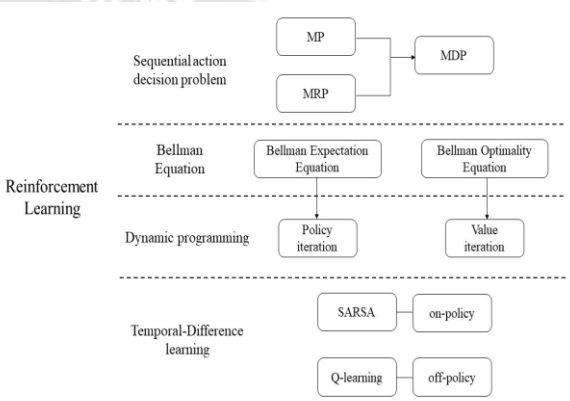

Figure2. Illustration of the flowchart of reinforcement learning.

become eligible to receive pay. The concept of depreciation is founded on the acknowledgement that the worth of a specific benefit diminishes progressively as time elapses. The remuneration earned by an agent experiences a steady reduction over a certain period of time as a result of depreciation, which is quantified by a numerical value ranging from 0 to 1 [39].

6) Policy

Once an autonomous agent reaches a specific state, it determines its subsequent course of action by referring to the prescribed policy.

$$\pi(a \mid s) = P[A_t = a \mid S_t = s] \quad (3)$$

Where (3) π is the policy probability at time t that the agent selects a in state. In a nutshell, reinforcement learning aims to acquire enhanced policies in comparison to the existing ones, with the objective of generating an optimal policy [40].

F. Q-LEARNING

In contrast to prior methodologies that have demonstrated little capacity in distinguishing between behavior and learning, Q-learning has implemented an off-policy approach to decouple

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
DOI: https://doi.org/10.17762/ijritcc.v11i10.8464
***Article Received:** 21 July 2023 **Revised:** 13 September 2023 **Accepted:** 28 September 2023*

the learning policy from the acting policy. As a result, the pertinent information was disregarded during the update of the Q-function for the present state, leading to an unfavorable conclusion described by a suboptimal judgment. This effect persists even when the chosen course of action for the future state is of moderate quality [41]. However, the integration of off-policy in Q-learning effectively addresses the aforementioned challenge. The equation for the Q-value is expressed as follows:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[R + \gamma \max Q(s',a') - Q(s,a)] \qquad (4)$$

The symbol α denotes the learning rate, a numerical value that is bounded between 0 and 1. The variable R represents a reward and indicates the rate at which the benefit decreases with time.

The process of Q-learning entails the iterative update of the Q-value for each state by employing the equation given before. The Q-table consists of pre-established rewards before the commencement of the Q-learning procedure. If an agent in the initial state choose an action according to a policy, it proceeds to the next state by utilizing (1).The procedure indicated above is repeatedly employed several times until the cumulative Q-value surpasses a pre-established threshold. At now, the Q-table is employed for the aim of resolving a specific problem. The Q-learning algorithm is a computational technique that combines Monte Carlo methods with dynamic programming approaches in order to effectively solve the Bellman problem.

Q-learning is widely regarded as the fundamental foundation for a wide array of reinforcement learning algorithms. The main factor contributing to this phenomenon is often ascribed to its straightforward implementation and its significant effectiveness in situations involving a single actor, setting it apart from other approaches. On the other hand, Q-learning demonstrates a limitation on the adjustment of a value, allowing it to take place solely once per action. The existence of recent state legislation is a significant obstacle in effectively resolving intricate matters within the context of several state operations. Furthermore, the necessity to assign a significant quantity of memory for storage arises from the specified dimensions of the Q-table utilized for rewards. In the framework of a multi-agent paradigm, the necessity for multiple agents to engage in collaboration necessitates the utilization of a substantial state-action memory, which might give rise to a range of challenges. The limited efficacy demonstrated by basic Q-learning algorithms becomes apparent when utilized within a multi-agent context, since they exhibit a deficiency in achieving desirable learning outcomes.

IV. RESULTS AND DISCUSSION

The Q-learning technique is widely recognized and utilized in the domain of reinforcement learning due to its capacity to efficiently acquire optimal policies in complex and demanding scenarios. The system operates in a manner that is independent of a pre-established model, allowing the agent to acquire knowledge exclusively through interactions with the environment, without any prior information. By means of an iterative approach, Q-values are continuously updated. These Q-values represent estimations of the total rewards that are linked to the selection of specific activities within particular states. Q-learning enables the agent to make intelligent judgments on resource allocation in cloud computing environments. The Q-learning algorithm is employed for the processing of the dataset, leading to the production of a ranking table based on the prescribed methodology. The Q-learning technique is employed to ascertain the reward associated with a specific action and state.

Figure 3 illustrates the graphical representation of the correlation between action and reward inside the Q learning procedure throughout the task rating phase. Tasks that generate favorable outcomes are given due consideration, while tasks that result in negative incentives are disregarded. The employment of the Q learning algorithm facilitates the execution of this strategy.

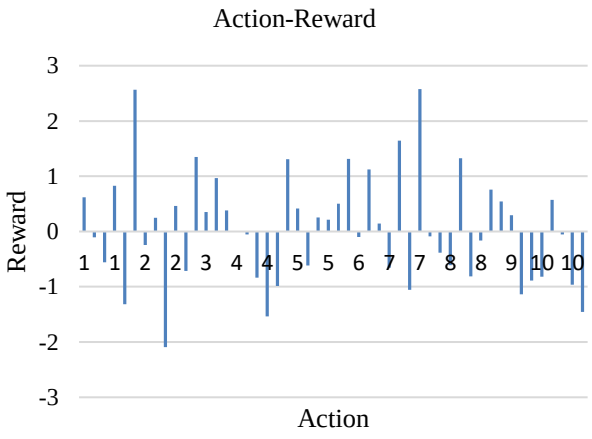

Figure 3: Action-Reward Relationship

Figure 4 depicts the graphical depiction of the correlation existing between the state and the corresponding reward. The value function represents the total reward accumulated by the Q-learning algorithm for every state. Figure 5 exhibits a graphic representation that illustrates the association between action and state. Every action is linked to a condition that demonstrates a hierarchical relationship.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
DOI: https://doi.org/10.17762/ijritcc.v11i10.8464
Article Received: *21 July 2023* ***Revised:*** *13 September 2023* ***Accepted:*** *28 September 2023*

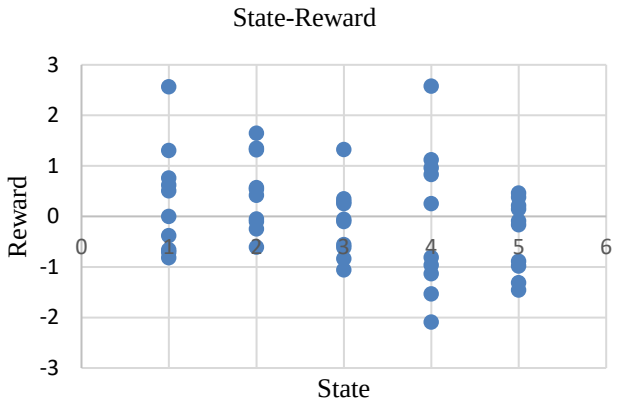

Figure 4: State-Reward Correlation

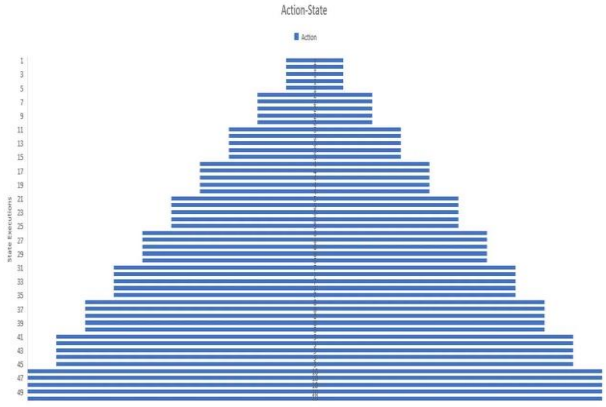

Figure 5 Illustration of Action-State Reward

The allocation of jobs to Virtual Machines is determined by the rating produced by the Q-learning algorithm. Figure 6 depicts the aggregate expense linked to each virtual machine (VM) in relation to the execution of the designated tasks. The expenses associated with task processing can be ascribed to several elements, including the cost of CPU utilization, the cost of memory usage, and the cost of bandwidth consumption. The data collected indicates that there are recurring swings in the overall cost.

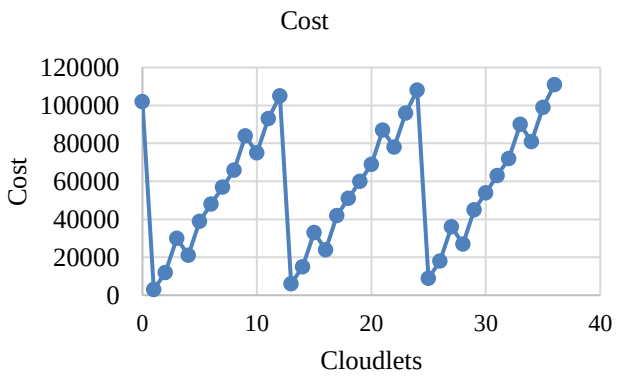

Figure 6: Illustration of Total cost

Figure 7 illustrates the temporal span of the virtual machine's operation while performing discrete activities. The temporal span that encompasses the entirety of the computing process, commencing at the initial commencement time and concluding at the final termination time, while also accounting for the time necessary for scheduling. The mean duration of execution for each task on every virtual machine is 3 seconds.

Figure 7: Portrayal of Execution time

Figure 8 illustrates a graphic that visually depicts the makespan of each task. The concept of makespan refers to the calculation of the maximum time required to accomplish a specific collection of jobs. The determination is made by considering the temporal demands associated with the completion of the final assignment. In the domain of work scheduling, a frequently pursued objective in the field of optimization is the minimization of the makespan.

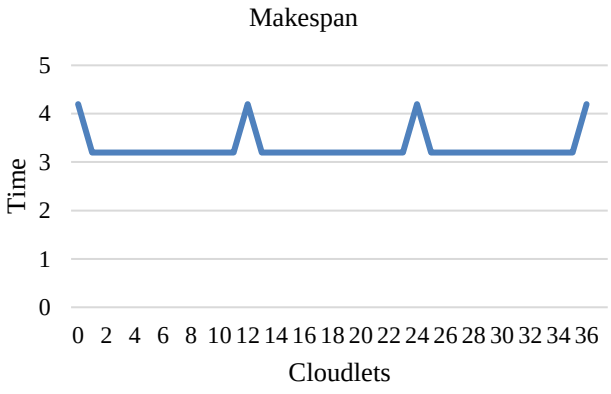

Figure 8: Representation of Makespan

The mean duration for each activity is 3 seconds. The term "Finish time" denotes the exact point in time when the last task of the final virtual machine (VM) is completed, as illustrated in Figure 9.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
DOI: https://doi.org/10.17762/ijritcc.v11i10.8464
***Article Received:** 21 July 2023 **Revised:** 13 September 2023 **Accepted:** 28 September 2023*

Figure 9: Illustration of Finish time

The determination of the termination time is dependent on the categorization of tasks into groups of cloudlets. A thorough examination was performed on a collective of four sets of cloudlets.

V. CONCLUSION

The present work explores the issue of virtual machine (VM) allocation and task scheduling inside a cloud environment, aiming to minimize the time required to complete jobs and the corresponding costs. This research introduces a unique scheduling strategy that employs Q-Learning to enhance the efficacy of resource utilization. The algorithm's main objectives encompass the optimization of its objective function, the construction of the desired network, and the utilization of experience replay approaches. In future endeavors, there is a stated objective to integrate more sophisticated approaches for cloud modelling, particularly by incorporating edge or fog computing alongside an increased number of schedulers. This strategy enables the mitigation of workload on cloud data centers and expedites the scheduling processes.

ACKNOWLEDGMENT

The authors would like to express their gratitude to Dr. Raghunath Sahu, Principal of College of IT & Management Education (CIME), Bhubaneswar, and the entire CIME family for their help and support throughout the research work.

REFERENCES

[1] A. Ullah, N. M. Nawi, and S. Ouhame, "Recent advancement in VM task allocation system for cloud computing: review from 2015 to2021," Artificial Intelligence Review, vol. 55, pp. 2529–2573, 2022.

[2] A. J. Ferrer, J. M. Marquès,and J. Jorba,"Towards the Decentralised Cloud: Survey on Approaches and Challenges for Mobile, Ad hoc, and Edge Computing,"ACM Computing Surveys, Vol. 51, pp. 1–36,2019.

[3] R. Nazir et al., "Cloud Computing Applications: A Review," EAI Endorsed Transactions on Cloud Systems , Vol. 6, pp. e5-e5,2020.

[4] V. V., J. B. G.,and S. Rajagopal,"Review on Mapping of Tasks to Resources in Cloud Computing,"International Journal of Cloud Applications and Computing (IJCAC), vol. 12(1), pp.1-17, 2022.

[5] B. Zheng, L. Pan,and S. Liu,"Market-oriented online bi-objective service scheduling for pleasingly parallel jobs with variable resources in cloud environments,"Journal of Systems and Software, vol. 176, June 2021.

[6] W. Song,Z. Xiao,Q. Chen,and H. Luo,"Adaptive Resource Provisioning for the Cloud Using Online Bin Packing,"IEEE Transactions on Computers, vol. 63(11), pp.2647-2660,July 2013.

[7] R. Grandl, G. Ananthanarayanan, S. Kandula, S. Rao,and A. Akella,"Multi-resource packing for cluster schedulers,"ACM SIGCOMM Computer Communication Review, vol. 44(4), pp. 455–466,August 2014.

[8] A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker,and I. Stoica,"Dominant resource fairness: Fair allocation of multiple resource types,"8th USENIX symposium on networked systems design and implementation (NSDI 11), 2011.

[9] Y. Xie,Y.Sheng,M. Qiu,and F. Gui,"An adaptive decoding biased random key genetic algorithm for cloud workflow scheduling,"Engineering Applications of Artificial Intelligence, vol. 112, p.104879, June 2022.

[10] M. S. Ajmal,Z. Iqbal,F. Z. Khan,M. Ahmad,I. Ahmad, and B. B. Gupta," Hybrid ant genetic algorithm for efficient task scheduling in cloud data centers,"Computers and Electrical Engineering, vol. 95, p.107419,October 2021.

[11] H. Mao,M. Alizadeh,I. Menache,and S. Kandula,"Resource Management with Deep Reinforcement Learning,"HotNets'16, pp. 50–56,November 2016.

[12] H. Lee,S. Cho,Y. Jang,J. Lee,and H. Woo,"A Global DAG Task Scheduler Using Deep Reinforcement Learning and Graph Convolution Network,"IEEE Access, 9, pp. 158548-158561,November 2021.

[13] Z. Peng, D. Cui, J. Zuo, Q. Li, B. Xu and W. Lin,"Random task scheduling scheme based on reinforcement learning in cloud computing,"Cluster Computing, vol. 18, pp.1595–1607, September 2015.

[14] P. Pradhan , P. K. Behera,and B.N.B. Ray,"Modified Round Robin Algorithm for Resource Allocation in Cloud Computing,"Procedia Computer Science,vol. 85, pp. 878-890,June 2016.

[15] T. Chen, A. G. Marques,and G. B. Giannakis,"DGLB: Distributed Stochastic Geographical Load Balancing over Cloud Networks,"IEEE Transactions on Parallel and Distributed Systems,vol.28(7),pp.1866-1880,December 2016.

[16] M. G. Arani and A. Souri,"LP-WSC: a linear programming approach for web service composition in geographically distributed cloud environments,"The Journal of Supercomputing, vol. 75, pp.2603–2628, October 2018.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
DOI: https://doi.org/10.17762/ijritcc.v11i10.8464
***Article Received:** 21 July 2023 **Revised:** 13 September 2023 **Accepted:** 28 September 2023*

[17] M. G. Arani, S. Jabbehdari and M. A. Pourmina,"An autonomic approach for resource provisioning of cloud services," Cluster Computing, vol. 19, pp.1017–1036, May 2016.

[18] D. Basu, X. Wang, Y. Hong, H. Chen,and S. Bressan,"Learn-as-you-go with Megh: Efficient Live Migration of Virtual Machines,"IEEE Transactions on Parallel and Distributed Systems,volume. 30(8), pp.1786-1801,August 2019.

M. Kumar and S.C. Sharma,"PSO-COGENT: Cost and energy efficient scheduling in cloud environment with deadline constraint,"Sustainable Computing: Informatics and Systems, vol. 19, pp. 147-164,September 2018.

[20] H. Z. Jin, L. Yang,and O. Hao,"Scheduling strategy based on genetic algorithm for Cloud computer energy optimization,"2015 IEEE International Conference on Communication Problem-Solving (ICCP),pp. 516-519,October 2015.

[21] R. Medara, R. S. Singh,and Amit,"Energy-aware workflow task scheduling in clouds with virtual machine consolidation using discrete water wave optimization,"Simulation Modelling Practice and Theory,vol. 110,p.102323, July 2021.

[22] D. Ding, X. Fan, Y.Zhao, K. Kang, Q. Yin, and J. Zeng,"Q-learning based dynamic task scheduling for energy-efficient cloud computing," Future Generation Computer Systems, vol. 108, pp.361-371, 2020.

[23] T. Dong, F. Xue, C. Xiao,and J. Li,"Task scheduling based on deep reinforcement learning in a cloud manufacturing environment,"Concurrency and Computation: Practice and Experience, vol. 32(11), p.e5654,January 2020.

[24] F. Cheng et al.,"Cost-aware job scheduling for cloud instances using deep reinforcement learning," Cluster Computing, vol. 25, pp.619–631, 2022.

[25] L. Cheng et al.,"Cost-aware real-time job scheduling for hybrid cloud using deep reinforcement learning,"Neural Computing and Applications, vol. 34, pp.18579–18593, June 2022.

[26] J. Yan et al.,"Energy-aware systems for real-time job scheduling in cloud data centers: A deep reinforcement learning approach,"Computers and Electrical Engineering,vol. 99,p.107688, April 2022.

[27] Y. Wei, L. Pan, S. Liu, L. Wu,and X. Meng,"DRL-Scheduling: An Intelligent QoS-Aware Job Scheduling Framework for Applications in Clouds,"IEEE Access,vol. 6,pp.55112-55125,September 2018.

[28] Y. Huang et al.,"Deep Adversarial Imitation Reinforcement Learning for QoS-Aware Cloud Job Scheduling,"IEEE Systems Journal, vol. 16(3),pp. 4232-4242,November 2021.

[29] Z. Cao, P. Zhou, R. Li, S. Huang,and D. Wu,"Multiagent Deep Reinforcement Learning for Joint Multichannel Access and Task Offloading of Mobile-Edge Computing in Industry 4.0,"IEEE Internet of Things Journal,vol. 7(7), pp. 6201-6213,March 2020.

[30] W. Guo, W. Tian, Y. Ye, L. Xu,and K. Wu,"Cloud Resource Scheduling With Deep Reinforcement Learning and Imitation Learning,"IEEE Internet of Things Journal,vol.8(5),pp.3576-3586,September 2020.

[31] N. Liu et al.,"A Hierarchical Framework of Cloud Resource Allocation and Power Management Using Deep Reinforcement Learning,"IEEE 37th international conference on distributed computing systems (ICDCS),pp. 372-382,2017.

[32] W. Wang et al.," Virtual machine placement and workload assignment for mobile edge computing ," IEEE 6th International Conference on Cloud Networking (CloudNet),pp. 1-6,2017.

[33] C. C. White III,and D. J. White,"Markov decision processes,"European Journal of Operational Research,vol. 39(1), pp.1-16,1989.

[34] R. Sutton,"Generalization in reinforcement learning: Successful examples using sparse coarse coding," Advances in neural information processing systems,1995.

[35] E. E.Dar, Y. Mansour,and P Bartlett,"Learning Rates for Q-learning," Journal of machine learning Research, vol. 5, pp.1-25, 2003.

[36] A.R. Cassandra,"Exact and approximate algorithms for partially observable Markov decision processes," Brown University ProQuest Dissertations Publishing,1998.

[37] M. L. Littman,"Value-function reinforcement learning in Markov games," Cognitive Systems Research,vol. 2(1), pp.55-66,April 2001.

[38] R. Sutton, D. McAllester, S. Singh,and Y Mansour,"Policy gradient methods for reinforcement learning with function approximation,"Advances in neural information processing systems, 1999.

[39] T. G. Dietterich,"Hierarchical Reinforcement Learning with the MAXQ Value Function Decomposition," Journal of artificial intelligence research, vol. 13, pp.227-303,November 2000.

[40] M. Irodova and R. H. Sloan,"Reinforcement Learning and Function Approximation,"The Florida AI Research Society Conference,pp. 455-460,2005.

[41] L. Shoufeng,L. Ximin,and D. Shiqiang,"Q-Learning for Adaptive Traffic Signal Control Based on Delay Minimization Strategy,"IEEE International Conference on Networking, Sensing and Control,pp. 687-691,April 2008.