INTERNATIONAL JOURNAL OF TECHNO-ENGINEERING
IJTE

ISSN: 2057-5688

Techniques for Reinforcement Learning-Based Dynamic Resource Allocation in Cloud-Based Systems

SATHEESH MASURAM
Research Scholar
Department of Computer Science and Engineering
J.S.UNIVERSITY, Shikohabad, UP

Dr. SARATH BABU DUGGIRALA
Supervisor , Associate Professor
Department of Computer Science and Engineering
J.S.UNIVERSITY, Shikohabad, UP

Dr.Vittapu Mani Sarma
Co-Supervisor, Malla Reddy Engineering College and
Management Sciences ,Medchal, Telangana

Abstract: When it comes to meeting the demanding requirements of today's applications, more and more people are turning to cloud computing as a scalable and effective answer. Efficiently distributing resources to suit the diverse requirements of users and apps is a major difficulty in this setting. Given its capacity for dynamic adaptability to evolving circumstances, reinforcement learning (RL) presents a viable strategy for addressing this issue. The purpose of this study is to provide a framework for efficient resource allocation in cloud-based architectures using RL methods. Our method makes use of agents that can sense their surroundings in the cloud and adapt their resource allocation strategies accordingly. As part of their efforts to optimise overall resource allocation, these agents take into account the diverse needs and priorities of various applications. According to our results, conventional approaches to resource allocation fall short when compared to our suggested RL framework in terms of efficiency and reaction time. Furthermore, it is resilient to sudden shifts in the workload and does a good job of adjusting to new circumstances. We believe our method might greatly improve cloud-based architectures' resource utilisation, cost-effectiveness, and user experience. As an added bonus, our system is very adaptable and can be simply modified to manage intricate resource allocation issues with many objectives. This makes it a highly flexible method for managing resource allocation in the cloud.

Keywords: High Demands, Dynamic Adaptation, Allocate Resources, Robust, Reduce Costs

1. Introduction

One important aspect of cloud-based architectures is dynamic resource allocation, which basically means that the system may dynamically add or remove resources based on the demand and workload at any given time [1]. Applications are guaranteed to be highly available and operate at a high level, and resources may be used efficiently and effectively. Multiple users share resources including storage, network bandwidth, and virtual machines in a cloud-based architecture [2]. A dynamic system that may distribute resources as required is preferable than a fixed allocation that might lead to underutilisation or oversubscription [3] since resource demand can fluctuate substantially based on workload and use patterns. In Dynamic Resource Allocation, a number of parts work together. The Resource Pool is the first one; it's a shared pool of resources that programs may use whenever they need them.

Servers and storage devices, both real and virtual, may be part

of this pool [4]. Second, there's the Resource Manager, whose job it is to oversee the pool's resources and distribute them accordingly.
A number of apps submit requests for resources, and it distributes those resources according to laws and procedures that have already been established [5].

The term "dynamic resource allocation" describes the practice of allocating storage, memory, processing power, and other computer resources to various applications and services in response to their actual needs. Since resources in cloud-based architectures are shared across several users and apps, dynamic resource allocation is essential for making the most of such resources. The effective deployment and operation of cloud-based systems, however, is not without its share of obstacles [7]. Managing resource contention is a key challenge in dynamic resource allocation. Instances of contention may arise when more resources are required to satisfy demand, as a consequence of competing applications for resources [8].

Improper management of this might result in a decrease in performance or even system breakdowns [9].

Consequently, in order to tackle this difficulty, cloud providers need establish strong regulations and algorithms for allocating resources.

Resolving the challenge of reliably projecting future resource consumption is another potential development area [10]. Due to the ever-changing nature of resource consumption, pinpointing exactly how much is required to run a service or application may be a real challenge.

Because of this, you risk under-provisioning, which means subpar performance, or over-provisioning, which means resource waste and higher expenses. Here are the key points of the study's:

- Maximised productivity: Computing, storage, and network resources may be dynamically and scalably allocated in cloud-based systems. As a result, resources may be used efficiently, with no resources being wasted and total expenses being minimised.
- The capacity to scale up resources in response to spikes in workload demand is a key feature of cloud-based systems that allows for seamless scalability. This keeps services up and running at all times and allows them to adapt to changes in demand with little to no downtime.
- Efficiency in cost management: By adjusting resource allocation in real-time according to workload demand, cloud-based architectures may maximise efficiency and save expenses. If a company's workload fluctuates, this feature may help them prevent over-provisioning and pay for resources only when they're really used. This leads to substantial savings for businesses.

2. Literature Review

In a server-less context, AI-based resource allocation may be used to continually monitor and modify resource allocation using reinforcement learning, a kind of machine learning, as explained by Schuler, L., et al. [11]. This allows for effective auto-scaling, which optimises performance and cost by dynamically adding or removing resources in response to real-time demand. This method has been described by Wei, Y., et al. [12]. It makes use of reinforcement learning to adapt the resources of a SaaS provider in a cloud environment that is always changing.

With the use of historical data and user interactions, it is able to generate smart forecasts and optimise the allocation of resources, guaranteeing that customer service is delivered efficiently and affordably.

The deep reinforcement learning-based dynamic computational offloading method for cloud robotics has been described by Penmetcha, M., et,al.[13].

This approach uses machine learning algorithms to optimise the task distribution between the robot and the cloud, thus reducing the computational burden on the robot. The total performance and usefulness of cloud robotics systems are enhanced by enabling efficient and flexible decision-making in real-time.

For the purpose of allocating resources in industrial IoT systems, Rosenberger, J., et al. [14] have addressed a deep reinforcement learning multi-agent system. In order to maximise efficiency in the distribution of resources inside industrial IoT systems, this intricate decision-making system employs machine learning techniques. It improves performance and efficiency by using deep reinforcement learning and a network of agents to adapt and optimise resource allocation in real-time.

Cloud resource scheduling, defined as the act of assigning

INTERNATIONAL JOURNAL OF TECHNO-ENGINEERING
IJTE

ISSN: 2057-5688

virtual resources (such as computing power and storage) in a cloud context, has been covered by Guo, W., et al. [15]. To maximise efficiency and save costs, this process may be optimised with the help of deep reinforcement learning and imitation learning, two state-of-the-art machine learning algorithms that use data and experience to intelligently allocate resources. In their discussion of deep reinforcement learning-based SLAs-aware online task scheduling, Ran, L., et al.[16] sought to optimise cloud-based work allocation and resource utilisation. To guarantee effective and dependable task allocation, deep reinforcement learning can adapt the scheduling approach in real-time according to incoming tasks and their associated SLA requirements. Machine learning approach known as deep reinforcement learning (DRL) has been covered by Mangalampalli, S., et al. [17]. DRL allows agents to learn the best rules for making decisions by experimenting. This method is used by DRL-based task-scheduling algorithms in cloud computing to automatically and effectively schedule jobs and distribute computer resources. An approach to optimising the distribution of computing resources in edge computing networks has been presented by Khani, M., et al.[18]: deep reinforcement learning-based resource allocation in multi-access edge computing. It learns and decides how to distribute resources effectively across activities via a feedback loop, which improves performance and energy efficiency over time. A method for dynamically allocating micro services in cloud computing settings that draws on fuzzy logic and reinforcement learning concepts is the one described by Joseph, C. T., et al.[19] called fuzzy reinforcement learning-based micro service allocation.

To optimise efficiency and resource utilisation, it leverages user input and historical data to intelligently decide how to distribute micro services among available resources. One aspect of cloud computing that has been covered by Jyoti, A., et al.[20] is dynamic resource provisioning. This involves dynamically assigning or scaling computer resources according to the demand or load that is present. This is accomplished by using service broker rules and load balancing strategies, which aid in resource management and guarantee cloud service performance at its best.

3. Proposed model

To maximise the use, allocation, and management of resources in a constantly changing and developing environment, the Reinforcement Learning Strategies for Dynamic Resources model is suggested. It makes judgements and takes actions that optimise performance and resource efficiency using reinforcement learning, an AI that learns by trial and error. There are three parts to the model: the setting, the agent, and the incentive and punishment structure.

In this subsection we define the reward function, value function and Q function from this MDP for the reinforcement learning.

$$L = \sum_{b=1}^{V} \gamma^{b-1} l_b \qquad (1)$$

We then define the value function of each state V π (s), denoting the expected total reward for an agent starting from state's with the policy π. Among all policy π'q, there existing an optimal policy π ∗ that makes T q to be maximum.

$$\pi^* = \arg\max_{\pi} T^{\pi}(q)$$

Within the time constraint period T, the transmission rate of the data packet with a size of B is determined.

The environment includes all the resources and variables affecting their availability and performance. The agent is the decision-making entity that interacts with the environment and learns from its actions.

The rewards and penalties system provides feedback to the agent based on its decisions and actions, encouraging it to learn and improve over time.

$$F\{\sum_{v=1}^{V} \sum_{n=1}^{N} \rho y[n] D_y{}^c[n,v] \geq \frac{I}{\Delta_V}$$

The resource allocation problem under consideration can be defined as follows: for all k ∈ K and m ∈ the transmission power of the V2V link can be adjusted continuously as a variable.

The local channel information that an agent can observe comprises its own channel gain$ey[n]$, the interference channel $ey, y\,'[n]$] from other V2V link transmitters, the interference channelchannel $en, y[n]$ from all V2V senders, for all m ∈ M.

INTERNATIONAL JOURNAL OF TECHNO-ENGINEERING
IJTE

ISSN: 2057-5688

$$Q_v^y = \{ey[n], ey, y'[n], eyI[n], e =_n .y[y]\} \quad (4)$$

Constant regulation of the T2T link's transmission power is at the heart of the Ivor's resource allocation architecture, which is the subject of this article.

Because it starts off ignorant of its surroundings, the agent must engage in exploratory and interactive tasks in order to learn about it. The agent learns, by trial and error, which behaviours get them rewards and which get them punishments. An agent's ability to foresee which behaviours will provide the greatest rewards improves in tandem with its level of environmental interaction.

3.1. Construction

Machine learning's Reinforcement Learning (RL) method lets agents figure out how to make the best decisions by seeing and interacting with their surroundings. Agents in dynamic resource management might benefit from RL techniques when deciding how to best allocate and use resources in an environment where conditions are constantly changing. Building the suggested model is shown in Fig. 1.

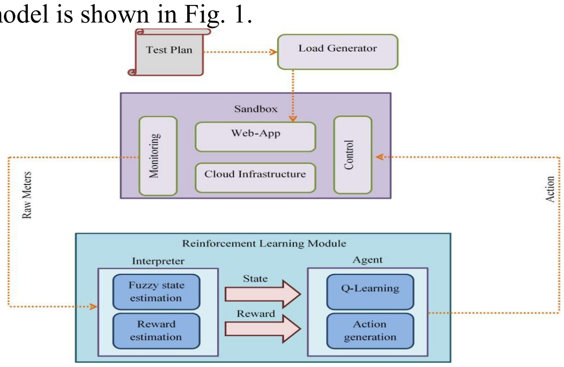


Fig 1 construction of the proposed model

Numerous technical aspects, such as the agent, environment, incentives, and learning algorithms, are involved in the development of RL methods for dynamic resource management. To begin, an accurate model of the environment is required to capture the dynamics of resource availability and the effects of resource allocation on the natural world.

Then, the reward is set to a constant β, which is greater than the maximum V2V link transmission rate.

$$R_y(v) = \{\sum_{n=1}^{N} \rho y[n] D_y^c[n,v], I_y \geq 0 \quad (5)$$

The reward function in RL is crucial for achieving optimal performance in high dimensional and complex environments.

The reward function in this paper is designed to balance the trade-off between the total capacity of the V2I link and the V2V link load's probability of successful transmission.

$$\lambda \sum_{n} D_n^d[n,v] + (1-\lambda) \sum_{y} R_y(v) \quad (6)$$

Neural networks are used in the Deep Deterministic Policy Gradient algorithm, a method that fits the value function.

The first step is to catalogue all of the environmental factors that are important, such as the available resources, how they are used, and how much more of them are needed. Second, it's crucial that the agent may learn how to best manage its resources via interaction with its surroundings.

In accordance with policy Qi(), the Critic network uses the state-action value function to assess the quality of the action chosen by the Actor-network. Ski t stands for agent k's input state, while γ is the discount factor for the immediate reward. The Lyv

$$S_y(Q^y, J^y) = E[L^y + \Upsilon S(Q^{y}, J^{y'})] \quad (7)$$

The DDPG algorithm aims to obtain an optimal policy π∗ y and learn the corresponding state-action value function until it reaches convergence.

The agent's goal is to maximize a reward function, which reflects the desired outcome of resource utilization, such as maximizing efficiency or minimizing cost.

The Actor and Critic networks update their evaluation network parameters based on the input mini-batch samples. The Critic

INTERNATIONAL JOURNAL OF TECHNO-ENGINEERING
IJTE
ISSN: 2057-5688

network's loss function can be expressed with 3.

$$(\theta_y^{S}) = G[(L_v^{y} + \gamma S_v'(Q_v^{y'}, J_v^{y} | \theta_y^{S'}) - S_y(S_v^{y}, J_v^{y} | \theta_y^{S}))^2]$$

The action-state value function of the target network is denoted as Q′ k (). If L sq. k) is continuously differentiable, then sq. k can be updated.

The agent can take action by using various resource allocation and utilization strategies.

3.2. Operating principle

One subfield of machine learning, known as reinforcement learning (RL), enables agents to acquire knowledge and decision-making capabilities by observing and responding to their surroundings. The application of RL to the field of dynamic resource management allows for the optimisation of real-time resource allocation and utilisation. The agent, the environment, and the reward system are the three key components that make up the operational principle of reinforcement learning systems for dynamic resource management. Figure 2 illustrates how the suggested paradigm works.

Fig 2 operating principle of the proposed model

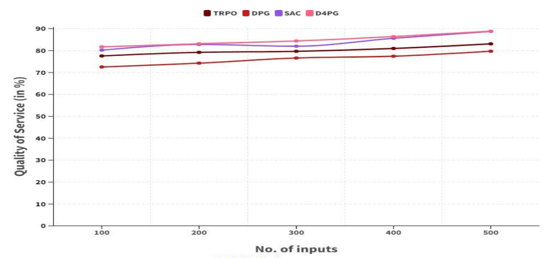
An agent is something that can make decisions and engages in environmental interactions. It responds to its surroundings by acting in accordance with its present condition and taking in rewards or punishments accordingly.

Whether it's a computer network or a manufacturing system, the environment is a representation of the optimised system or process.

Although beyond the scope of this article, the controller may initiate task reallocation and service relocation in response to a substantial decline in Qi or the introduction of new requirements from Iota users.

$$\chi_o(v) = \arg\max_{x} HH_o(v), x \in X'. \quad (9)$$

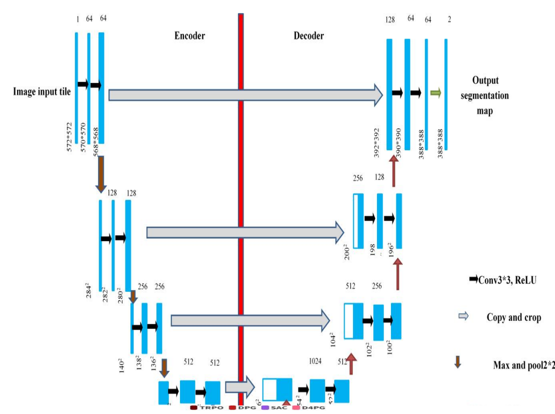
We employ a widely-used edge computing model whereby the computing latency depends on the computing capacity requirement and the allocated computing capacity.

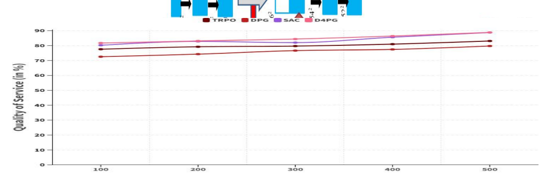


$$c_o^{f}(v) = \frac{y_o(v) T_o^{v}(v)}{F_o^{d}(v)}, \quad (10)$$

It is dynamic and constantly changing, creating a complex and uncertain environment for the agent to learn from. The reward system provides feedback to the agent based on its actions. The agent's goal is to maximize its total reward over time, so it learns to take actions that lead to high rewards and avoid actions that lead to penalties.

4. Result and Discussion





The Trust Region Policy Optimisation (TRPO), Deterministic Policy Gradient (DPG), and Soft Actor-Critic (SAC) models have been contrasted with the proposed Distributed Distributional Deterministic

ISSN: 2057-5688

Policy Gradients (D4PG) model.

4.1. Convergence Rate: This setting determines how quickly the cloud-based architecture's reinforcement learning algorithm can pick up new information and adjust its use of resources.

4.2. An improved system's overall performance is shown by a greater convergence rate, which means the algorithm can efficiently and rapidly adapt resource allocation to changes in real-time. Figure 3 displays the convergence rate comparison.

Fig.3 Comparison of Convergence Rate

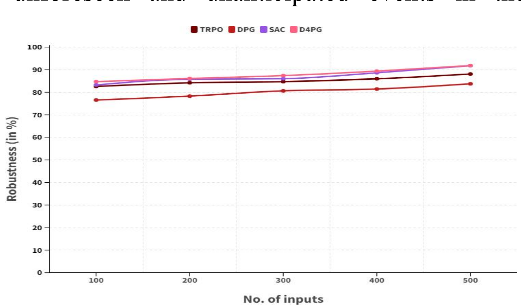
4.3. Resource Utilization: Efficiency in using and assigning cloud-based architectural resources is measured by the resource utilisation parameter, which is part of the reinforcement learning technique. With a greater rate of resource utilisation, available resources are used more efficiently, leading to better system performance and cost savings. A comparison of resource use is shown in Fig. 4.

Fig.4 Comparison of Resource Utilization

4.4. Quality of Service (QoS): Users of the cloud-based architecture may gauge the reinforcement learning strategy's capacity to sustain and enhance service quality using this metric. Considerations like availability, dependability, and reaction speed fall within this category. The comparison of service quality is shown in Fig. 5.

Fig.5 Comparison of Quality of Service

4.5. Robustness: How well the reinforcement learning method deals with rapid changes in workload or resource failures, or other unforeseen and unanticipated events in the



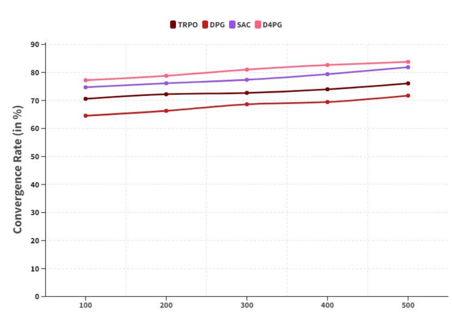
cloud-based architecture, is measured by the resilience parameter. Figure 6 illustrates the Contrast in Robustness

Fig.6 Comparison of Robustness

Conclusion:

This research delves into the possibilities of RL methods for optimising cloud-based architectures' dynamic resource allocation. The suggested RL architecture proved capable of effectively managing cloud resources and adapting to different workload demands. The RL agents outperformed the conventional techniques of allocation in terms of resource utilisation, reaction times, and overall performance by learning from real-time feedback. This adaptive method not only makes cloud systems more resilient and scalable, but it also provides a technique to optimise costs by allocating resources based on exact demand levels.The experimental findings demonstrated that the framework can adapt to changing workloads, which is significant for a dynamic cloud environment. Cloud computing optimisation research may take use of the framework's extensibility to tackle increasingly difficult multi-objective resource allocation challenges.

When applied to resource allocation, reinforcement learning techniques show promise as a remedy for the problems encountered by contemporary cloud infrastructures; these techniques also provide significant enhancements to performance, scalability, and efficiency.



ISSN: 2057-5688

It is recommended that more study be conducted to enhance the model, resolve any possible security issues, and investigate its potential uses in other cloud settings.

References

[1] Thein, T., Myo, M. M., Parvin, S., & Gawanmeh, A. (2020). Reinforcement learning based methodology for energy- efficient resource allocation in cloud data centers. Journal of King Saud University-Computer and Information Sciences, 32(10), 1127-1139.

[2] Chen, X., Zhu, F., Chen, Z., Min, G., Zheng, X., & Rong, C. (2020). Resource allocation for cloud-based software services using prediction-enabled feedback control with reinforcement learning. IEEE Transactions on Cloud Computing, 10(2), 1117-1129.

[3] Belgacem, A., Mahmoudi, S., & Kihl, M. (2022). Intelligent multi-agent reinforcement learning model for resources allocation in cloud computing. Journal of King Saud University-Computer and Information Sciences, 34(6), 2391-2404.

[4] Yang, Z., Nguyen, P., Jin, H., & Nahrstedt, K. (2019, July). MIRAS: Model-based reinforcement learning for microservice resource allocation over scientific workflows. In 2019 IEEE 39th international conference on distributed computing systems (ICDCS) (pp. 122-132). IEEE.

[5] Chen, Y., Sun, Y., Wang, C., & Taleb, T. (2022). Dynamic task allocation and service migration in edge-cloud iot system based on deep reinforcement learning. IEEE Internet of Things Journal, 9(18), 16742-16757.

[6] Desai, B., & Patil, K. (2023). Reinforcement Learning-Based Load Balancing with Large Language Models and Edge Intelligence for Dynamic Cloud Environments. Journal of Innovative Technologies, 6(1), 1-13.

[7] Nascimento, A., Olimpio, V., Silva, V., Paes, A., & de Oliveira, D. (2019, May). A reinforcement learning scheduling strategy for parallel cloud-based workflows. In 2019 IEEE international parallel and distributed processing symposium workshops (IPDPSW) (pp. 817-824). IEEE.

[8] Nascimento, A., Olimpio, V., Silva, V., Paes, A., & de Oliveira, D. (2019, May). A reinforcement learning scheduling strategy for parallel cloud-based workflows. In 2019 IEEE international parallel and distributed processing symposium workshops (IPDPSW) (pp. 817-824). IEEE.

[9] Xu, Z., Tang, J., Yin, C., Wang, Y., Xue, G., Wang, J., & Gursoy, M. C. (2020). ReCARL: resource allocation in cloud RANs with deep reinforcement learning. IEEE Transactions on Mobile Computing, 21(7), 2533-2545.

[10] Asghari, A., Sohrabi, M. K., & Yaghmaee, F. (2020). A cloud resource management framework for multiple online scientific workflows using cooperative reinforcement learning agents. Computer Networks, 179, 107340.

[11] Schuler, L., Jamil, S., & Kühl, N. (2021, May). AI-based resource allocation: Reinforcement learning for adaptive auto-scaling in serverless environments. In 2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid) (pp. 804-811). IEEE.

[12] Wei, Y., Kudenko, D., Liu, S., Pan, L., Wu, L., & Meng, X. (2019). A reinforcement learning based auto-scaling approach for SaaS providers in dynamic cloud environment. Mathematical Problems in Engineering, 2019(1), 5080647.

[13] Penmetcha, M., & Min, B. C. (2021). A deep reinforcement learning-based dynamic computational offloading method for cloud robotics. IEEE Access, 9, 60265-60279.

[14] Rosenberger, J., Urlaub, M., Rauterberg, F., Lutz, T., Selig, A., Bühren, M., & Schramm, D. (2022). Deep reinforcement learning multi-agent system for resource allocation in industrial internet of things. Sensors, 22(11), 4099.

[15] Guo, W., Tian, W., Ye, Y., Xu, L., & Wu, K. (2020). Cloud resource scheduling with deep reinforcement learning and imitation learning. IEEE Internet of Things Journal, 8(5), 3576-3586.

[16] Ran, L., Shi, X., & Shang, M. (2019, August). SLAs-aware online task scheduling based on deep reinforcement learning method in cloud environment. In 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS) (pp. 1518-1525). IEEE.