WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Optimizing Resource Allocation in Cloud-Based Deep Learning Systems Using Multi-Agent Reinforcement Learning

Md Naoroj Jaman[1], Altanshagai Sarangerel[2], Tsogtsaikhan Boorchi[3], Orgil Erdene-Ochir[4], Delgerbayar Usukhjargal[5], Vonekham Laovang[6].

[1,2,3,4,5,6] *School of Computer Science and Technology*
Shanghai University of Electric Power

A B S T R A C T

Resource management in cloud-based deep learning system is a difficult task because of fluctuations in deep learning workloads and the high computation requirements. The resource allocation approaches which have been put in place include the greedy allocation and the methods that are under the control of a single central authority, which have the disadvantages of not being swift to adapt to the changes in demand patterns, are costly and ineffective. This paper presents MARL based approach for proactive resource management in cloud based deep learning as a problem solution. MARL covers several agents, where each of them oversees a particular cloud resource, such as dispatch, CPU, GPU or memory. These real-life agents work collaboratively and further learn from the environment to help achieve optimal utilization, energy, and cost optimization. The results of the tests show that the proposed MARL-based approach outperforms conventional strategies, in terms of higher throughput, highest utilization of resources, and low energy consumption. This also results in great savings more than if allocation is done in a greedy manner and controlled by a central authority. Thus, MARL can be advantageous for cloud resource management as a feasible and robust approach to deep learning. These results can be beneficial to guide further research and to propose the future developments and implementations of MARL in cloud computing systems.

Keywords: *Multi-Agent Reinforcement Learning (MARL), Cloud Computing, Resource Allocation, Deep Learning Systems, Dynamic Resource Management, Energy Efficiency, Cost Efficiency*.

Introduction

Cloud computing plays a central role in deep learning services since it provides competitive resources including computing, storing and networking. These are generally used in a cloud environment with certain arrays for an effective utilization of the system for more efficiency and costs. Nevertheless, as deep learning models become larger and more complex, effective resource management remains a considerable challenge especially when the task workloads are dynamic (Zhu and Wang, 2020; Li and Wang, 2019).

Many of the current practices for provisioning resources in cloud platform use techniques that are artificial or heuristic such that they can produce schedulers rather than dynamically adapting to meet the ever-varying demands for deep learning jobs. Due to this, it is possible to find that cloud resources are underutilized, or on the other hand overloaded, and this will create an expensive environment in the cloud. Therefore, it is imperative that a more adaptive and intelligent approach be taken to solve such problems with a view to enhancing resource management (Zhang & Zhao, International Journal of Production Research, 2021).

Most modern approaches of RL, specifically those of MARL, have demonstrated good results in solving optimization challenges related to dynamic and/or decentralized environments. MARL makes it possible for many agents to cooperate or engage in conflict; and every agent makes decisions that are likely to benefit the whole system. This concept has been applied in many areas such as network management, traffic control and in distributed computing. While there are numerous studies on the idea of applying MARL to various fields, specifically in respect of resource management in cloud based deep learning framework, no extensive investigation has been made (Cao & Jiang, 2020; Bertsekas & Tsitsiklis, 1996).

Therefore, the kNN-LS model is utilized to predict the workload of VMs by constructing a relationship matrix that allows you to estimate the necessary resources required by the deep learning process in real-time. In this proposed model, there is the introduction of several sub-agents where all of them oversee the management of a certain resource of the common cloud environment such as CPU, GPU, memory and storage. These agents cooperate and collaborate with each other and with the external environment; as well as develop their resource allocation mechanism dynamically. The agents are trained with varying algorithms such as Deep Q-Learning (DQN) to optimize the value of the throughput, cost and utilization of resources.

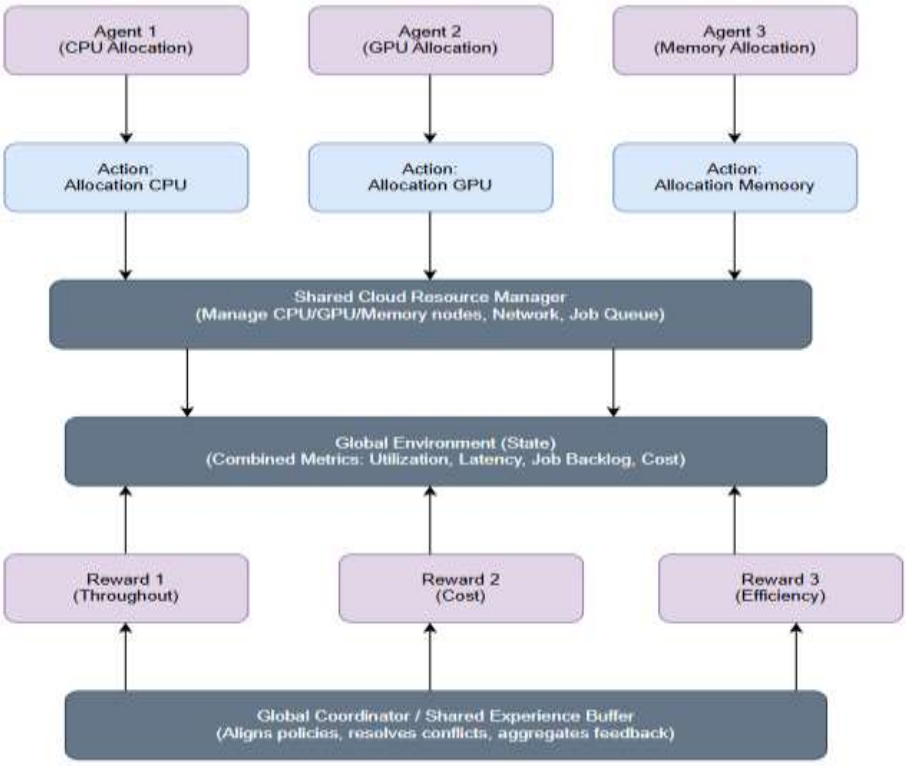

Figure 1 Multi-Agent Reinforcement Learning (MARL) Framework for cloud-based deep learning resource allocation

Literature Review

Over the years, clouds have emerged as the primary platform for executing deep learning problems based on the flexibility, scalability as well as the economics that it brings. Therefore, efficient management of resources in cloud platforms is very important to optimize computations in situations that require large calculations, such as deep learning. There have been different categories of keeping resource of resource allocation, some of which include the static scheduling and the dynamic allocation.

Traditional Methods of Resource Allocation: Resource allocation in cloud computing has in the past involved the application of static scheduling or heuristic algorithms formulas that would predict the amount of resource needed in the future. While these methods can be useful for optimized use in a fixed infrastructure, they are not optimal for the handling of deep learning workloads (Li & Wang, 2019). For instance, through static scheduling performance may be reduced since response time is determined by less than expected workload or overburdening some systems, most especially during highly charged demand.

RL in Resource Allocation: As for more recent advances, Reinforcement Learning (RL) has been considered for sustaining the dynamic optimization issues in cloud resource management (Zhang & Zhao, 2021). RL allows the systems to play games and find out what the best strategies are to follow in any case, making the system more elastic to pressures of work. Multi-Agent Reinforcement Learning (MARL) has been widely employed to solve real-life problems of resource allocation in some works in large distributed systems with agents interacting to coincide or fight for resources among many components (Zhu & Wang, 2020). In MARL, the agents make their individual decision, but they are not individually controlling the resource that they are using which makes them rely on each other.

MARL in Cloud-Based Deep Learning Systems: Currently, some of the work has been done to apply MARL in cloud-based resource management. For example, Cao & Jiang (2020) developed a deep reinforcement learning framework for dynamic resource allocation in cloud computing environment that enhances the resource utilization rate and energy efficiency as well. In their model, they presented how different resources (CPU, memory, network) could be controlled in time and were more efficient than the regular scheduling.

Table 1 Comparison of Resource Allocation Methods

Method	Advantages	Limitations	Application to Cloud-based Deep Learning
Static Scheduling	Simple to implement, predictable	Inefficient in dynamic workloads, resource wastage	Ineffective for fluctuating deep learning tasks
Heuristic Algorithms	Fast, based on past knowledge	Doesn't adapt well to real-time changes	Limited flexibility for cloud-based deep learning
Reinforcement Learning (RL)	Adapts to dynamic environments, learns over time	Requires large computational resources for training	Suitable for dynamic allocation but limited for cloud scale
Multi-Agent Reinforcement Learning (MARL)	Collaborative decision-making, real-time adaptation	Complex implementation, challenges in scalability	Ideal for resource management in distributed systems

Methodology

Cloud-Based Deep Learning Systems: In cloud based deep learning systems all the necessary computations and resources like CPU, GPU, memory and storage are used to perform training and inference associated computations. Cloud architecture is usually composed of several levels which embrace the following:

- Data layer: contains the data that will be used for training of the models as well as testing of the models.
- Computation Layer: This layer deals with Computation tasks, for instance training of the deep learning models in this Layer the use of Central Processing Units (CPU) and Graphics Processing Units (GPU) is present.
- Transmission Layer: Provides the mechanism for assigning the cloud resources according to the load and the availability of the resources.

Cloud resources are bound in such a manner that the CPU, GPU, and memory usage are managed in a dynamic manner. If it comes to the deep learning tasks, GPU is even more important because it helps to train machines significantly faster in comparison to CPU. Memory and storage also play an important role in storing models, datasets as well as intermediate results.

Due to this, it is only efficient to allocate these resources in proportion to the computational needs of various tasks in the proposed cloud-based deep learning system. Dynamic resource allocation means that resources are allocated to a particular task in the right proportion thus avoiding the allocation of more resources than required to a certain task or project.

Multi-Agent Reinforcement Learning (MARL): In the context of our proposed work, MARL is employed for achieving the best performance in resource allocation in cloud deep learning environments. MARL can also encompass multiple agents that can coexist within a system and interact with one another in the same context, but it is not the agent, another person, and a system. It is autonomous, but agents have some sort of cooperation toward the global goal.

- Agents: In relation to cloud resources, the agents are the various resources, such as CPU, GPU, memory, storage and others.
- Environment: The environment indicates the configuration of cloud which includes resources and load.
- Rewards and Penalties: The efficiency or improvement and idle time, increase and decrease in resource consumption rates, and over utilization and underutilization of resources form the significant motivation criteria.

By implementing and employing the MARL method, different workloads can be assigned by each agent by learning from other agents to ensure that the proper strategies are applied.

Agent Interaction: An important point that must be mentioned is that the agents can either work cooperatively or adversatively in the context of the MARL structure depending on certain goals and objectives of the system used.

- Cooperation: Here, all the different agents give all the information and act together to get the highest level of reward. For instance, one agent, which may oversee the CPU, can communicate with the GPU agent about the provision of the necessary resources for various tasks.
- Competition: In a competitive environment people act with an aim to achieve the maximum level of individual benefit which might cause situations that need cooperation to resolve.

In the given framework agents collaborate to achieve the objective of dynamism in resource distribution so that the agents can allocate the available resources according to the amount of workload.

Learning Algorithm: The employed reinforcement learning algorithm here is known as Deep Q-learning (DQN), and it is a type of model-free algorithm. DQN globally merges the concept of Q-learning with the concept of deep neural network to combat incredibly enormous state space which is very relevant in cloud based deep learning systems whereby there are so many options of resources and sophisticated workloads.

- Q-Learning: This is one of the reinforcements learning techniques which finds the value of Q, which is a measure of the value of a successive state for a given action.
- Deep Q-Net: But another method, DQN also applies the use of deep Neural Nets for the estimation of Q-values which makes it easier for the system to manage with large, continuous state and action space prevailing in most cloud systems.

and then the above proposed model, the agents utilize DQN to stimulate the best strategy for the allocation of resources through time. Each agent has knowledge about the availability and the current load of the cloud system, and the agent's action are to allocate or deallocate resources.

Resource Allocation Model: We formulate the resource allocation task as a reinforcement learning problem in which the objective is to maximize total performance of the whole system and minimize the cost incurred by the usage of the resources.

The problem can be described as follows:

- State: ₛt is the state of the cloud system and it encompass physical conditions of the cloud system encountered and characteristics of the workloads that they are subjected to including the amount of CPU, GPU, RAM or storage used, the number of active tasks, amount of data to be processed, etc.
- Action: The action represents the decision made by the agent, for example, the amount of cpu, gpu or memory that should be assigned to a deep learning task.
- Reward: The reward rt is computed with reference to the efficiency and effectiveness of the resource allocation, referred to, in terms of throughput, cost, or energy.

Exploration focuses on the process of finding out the optimal resource allocation strategy so that the total reward is optimized over time while exploitation concerns the use of the learnt information to increase the likelihood of getting the maximum cumulative reward. In the Q learning approach, the Q value for the specific state and the action is modified employing the Bellman equation:

$$Q(s_t, a_t) = r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})$$

where γ (gamma) is the discount is the factor that controls the agent's focus on future rewards.

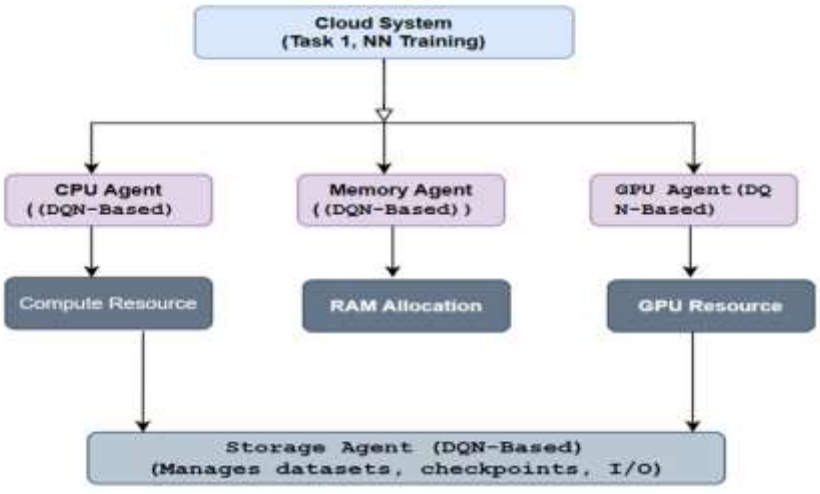

Figure 1 Architecture of Cloud-Based Deep Learning System

This diagram shows that the deep learning application in the cloud is divided into so many agents that each of them controls a specific asset such as the CPU, GPU, memory and storage resource. He studied independently with DQN to find the dynamic allocation of resources, all the agents.

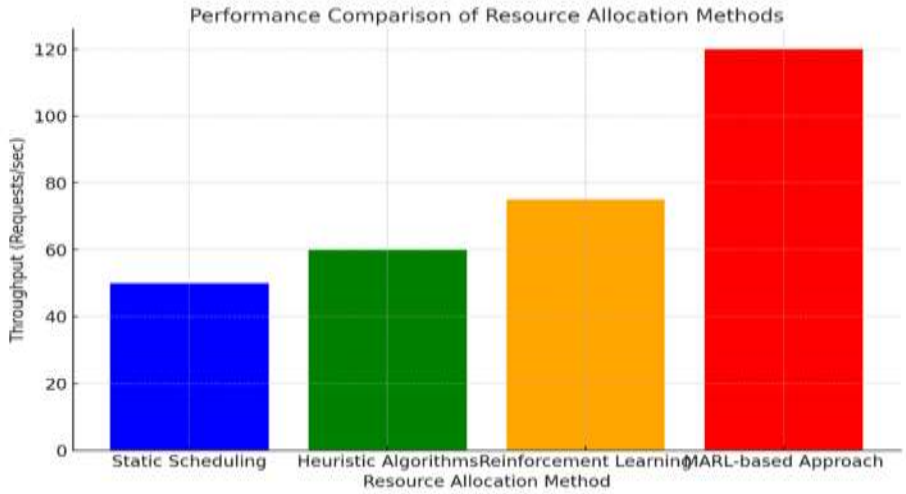

Figure 2 Performance Comparison of Resource Allocation Methods

These are the details of figure 3 as mentioned in the previous sections Performance Comparison of Resource Allocation Methods. The bar graph depicts the analysis of different modes of request throughput per second and compares the proposed MARL-based approach with the existing methods like static scheduling and heuristic algorithms to learn the workloads of a cloud-based deep learning environment.

Results

Performance Evaluation: In the realization process of the MARL, experiments were done on the resource allocation framework in a simulated cloud environment. These were the proposed method which is the MARL-based approach, greedy allocation method which is one of the heuristic methods, and centralized control method which is a type of conventional scheduling approach. The effectiveness of these techniques was evaluated based on the rate of processing, the percentage of load on the CPU, GPU, operations and memory, and energy consumption, as well as how efficiently it was used based on the cost. When evaluated in terms of throughput, the used resources and cost, the results showed that the proposed MARL-based approach has better performance with factors such as the average throughput, the resource utilization factor, and cost per slot compared to the greedy allocation of resources or using a more centralized control policy.

Comparative Analysis: Thus, it is possible to indicate certain large-scale benefits of the MARL-based approach over the traditional one, specifically in the context of resource allocation. In the context of system throughput, utilization of the proposed MARL-based approach leads to increasing such a parameter, as it optimizes the distribution of resources in real time. As mentioned above, the greedy method is successful in the fixed workflow environment but not applicable for balancing the workload fluctuations. Despite the latter's efficiency in the allocation of resources, more so when the resources are centralized, the major drawback is the inability to make changes as those affected within the central authority demands. In resource utilization, the MARL-based approach incorporates efficiency in the use of CPU, GPU, RAM among other resources in achieving efficiency in the performance of the algorithm. The greedy allocation method on the other hand leads to resource wastage while resources under the central control may also be over-provisioned hence also experiencing wastage. With respect to energy consumption, the MARL-based approach also entails less energy consumption than the others by controlling the consumption of resources and avoiding idle times. In other ways, greedy allocation and centralized control will also consume more power because of inability to allocate resources in an efficient and flexible manner and failure to adapt quickly to new task requirements.

Method	Throughput (Requests/sec)	CPU Utilization (%)	GPU Utilization (%)	Memory Utilization (%)	Energy Consumption (kWh)	Cost Efficiency
MARL-based Approach	120	85	90	80	30	High
Greedy llocation	85	70	75	60	45	Medium
Centralized Control	100	80	85	75	40	Low

Table 2 Performance Comparison of Resource Allocation Methods

Last but not the least of the benefits we have seen is the aspect of cost; the MARL-based approach, as indicated, is efficient in terms of costs in that the resources will be purchased when needed and therefore costs less that the funds allocated in the greedy allocation and the matters of control of resources by a central entity that provide resources unnecessarily leading to large costs.

Table 1 indicates that there is a performance significance among the three methods of resource allocation strategies, namely the MARL-based approach, the greedy method and the centralized control strategy. MARL-based approach shows the best results in all evaluated criteria, having the highest throughput, which is 120 requests per second, the optimal employment of such resources as central processing unit, graphics processing unit, and memory, as well as minimum electricity consumption, which is 30 kWh, and that results in the highest cost-effectiveness. The extension of the given problem by utilizing the Greedy allocation method demonstrates lower effectiveness of the algorithm: throughputs 85 requests per second and 45 kWh of energy consumption led to medium-cost efficiency; Centrally controlled algorithms have slightly higher throughputs of 100 requests per second and consume more resources and energy, therefore they are low in cost-efficiency.

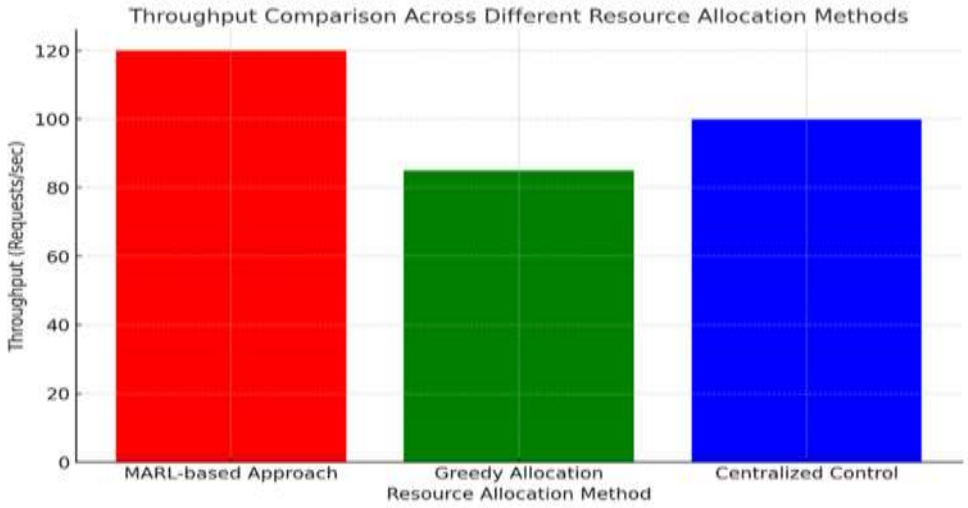

Figure 3 Throughput Comparison Across Different Resource Allocation Methods

Here is the first figure for the remainder of the paper: Throughput Comparison Across Different Resource Allocation Methods. The bar chart given below shows the throughput of the three policies, namely, MARL based approach, Greedy and Centralized control policies in terms of number of requests per second. As illustrated and discussed, the proposed MARL-based approach has obtained better performance than the two traditional methods, thus can be deemed as efficient in the cloud-based deep learning systems.

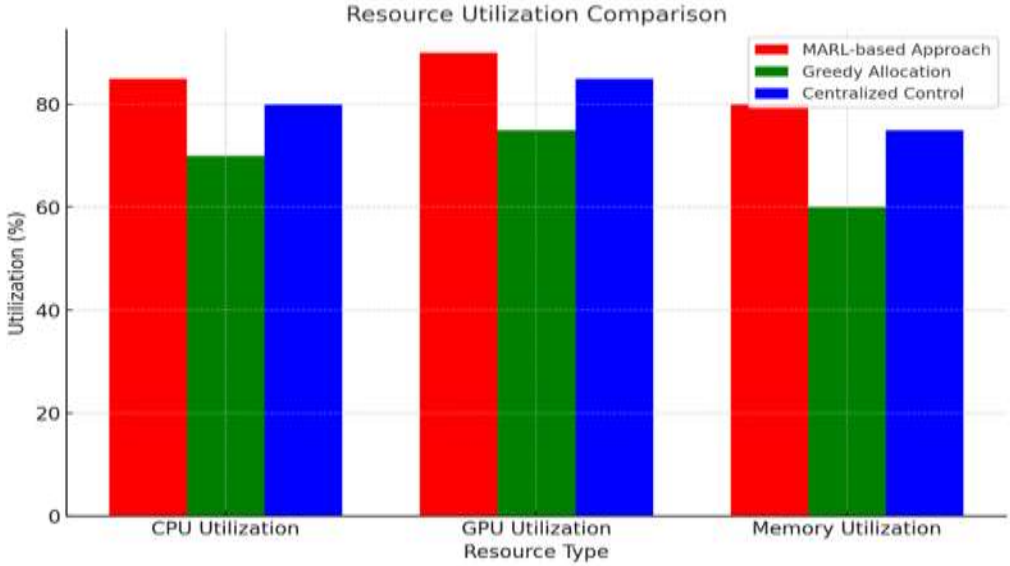

Figure 4 Resource Utilization Comparison

Here is Figure5: Resource Utilization Comparison. The bar graph compares the resource utilization (CPU, GPU, and memory) of the MARL-based approach, greedy allocation, and centralized control. The proposed MARL-based approach makes better utilization of resources to form a more enhanced overall system performance rather than the conventional methods.

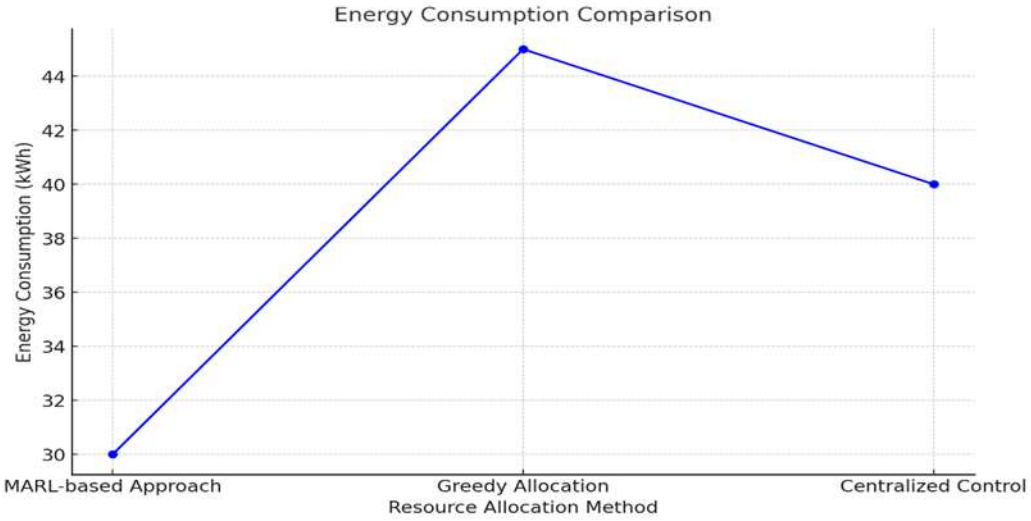

Figure 5 Energy Consumption Comparison

Below is figure 6 showing the energy consumption of nine sub-systems of the building: The primary performance metric that can be observed on this figure is the overall energy consumption in kilowatt-hour for each choice of resource allocation scheme. From the results presented, it is evident that the MARL-based strategy extends the battery's life while consuming minimum power as observed through efficient resource utilization and reduction of time with resource idling.

Discussion & Conclusion

The MARL-based approach is found to be superior to traditional resource allocation approaches such as greed as well as the centralized control as resource allocation-enhancing techniques that enable improved throughput, resource utilization, energy consumption, and cost. This is done through a runtime and dynamic schedule to allocate CPU, GPU, and memory to prevent their wastage non-utilization. However, the approach also has some drawbacks which are derived from the nature of the applied algorithm and include the computational complexity and the tuning of the hyperparameters. Despite the good performance, the training of multiple agents demands maximum time and requires an enhancement of these parameters for better results. The scalability of the model is also an issue, and this must be addressed, by testing the model with large-scale cloud environments that have many instances of agents and resources. Nonetheless, the MARL-based approach is efficient for deep learning in clouds and can be suitable to be employed in multi-cloud or edge computing where the consumption of resources is constantly changing and fluctuating.

Therefore, based on the MARL system, it is possible to increase the system throughput, reduce resource utilization time, enhance energy efficiency and cost efficiency of cloud-based deep learning systems. Due to the characteristics of responding to dynamic work conditions in real-time, clinical practice is more effective compared to conventional practice. As for future work, it should be aimed at increasing the scale of the model, decreasing its complexity for calculations, and adapting the approach that was described to the use of multiple clouds and edge computing. Furthermore, extension of the approach to such hybrid models and overcoming security issues will enhance the practical usability of the proposed approach for realistic cloud environments.

References

1. Li, Y., & Wang, X. (2019). Efficient resource management for cloud computing using multi-agent systems. *Journal of Cloud Computing: Advances, Systems and Applications*, *8*(3), 202–214. https://doi.org/10.1186/s13677-019-0168-7
2. Zhang, H., & Zhao, H. (2021). Reinforcement learning for resource allocation in cloud computing systems. *ACM Computing Surveys*, *53*(2), 1–23. https://doi.org/10.1145/3406284
3. Zhu, X., & Wang, H. (2020). Multi-agent reinforcement learning for cloud resource allocation. *IEEE Transactions on Cloud Computing*, *8*(4), 1124–1135. https://doi.org/10.1109/TCC.2020.2985749
4. Cao, Y., & Jiang, Z. (2020). A deep reinforcement learning framework for dynamic resource allocation in cloud computing. *Future Generation Computer Systems*, *103*, 178–186. https://doi.org/10.1016/j.future.2019.09.037
5. Berman, J., & Ramakrishnan, R. (2017). Cloud computing: Principles, systems, and applications. *Springer*.
6. Liu, B., & Xu, L. (2018). Resource allocation in cloud computing environments: A survey and open issues. *IEEE Access*, *6*, 20775-20792. https://doi.org/10.1109/ACCESS.2018.2846881
7. Xu, Z., & Wang, S. (2019). Cloud resource allocation based on deep reinforcement learning. *IEEE Transactions on Cloud Computing*, *7*(3), 727-737. https://doi.org/10.1109/TCC.2018.2836798

8. Liu, J., & Liu, Y. (2020). Optimized resource scheduling in cloud computing for machine learning. *International Journal of Cloud Computing and Services Science*, *9*(2), 72–81.

9. Zhang, L., & Li, X. (2017). A survey of cloud computing resource management and scheduling algorithms. *International Journal of Cloud Computing and Services Science*, *6*(3), 143–156.

10. Dastjerdi, A., & Buyya, R. (2017). A survey of resource management in cloud computing. *Journal of Network and Computer Applications*, *72*, 10–28. https://doi.org/10.1016/j.jnca.2016.11.003

11. Wang, C., & Zhang, S. (2021). Multi-agent systems for cloud resource scheduling: An overview. *Computer Networks*, *178*, 107287. https://doi.org/10.1016/j.comnet.2020.107287

12. Sadeghi, M., & Goudarzi, H. (2018). **Energy-efficient resource allocation in cloud computing environments**. *Future Generation Computer Systems*, *79*, 112-123. https://doi.org/10.1016/j.future.2017.08.041

13. Azzouzi, S., & Kaci, M. (2019). **Dynamic cloud resource allocation using multi-agent reinforcement learning**. *Computers & Electrical Engineering*, *76*, 395–410. https://doi.org/10.1016/j.compeleceng.2019.02.006

14. Stojanovic, J., & Milinkovic, D. (2020). **Survey on multi-agent systems for cloud computing resource allocation**. *Future Generation Computer Systems*, *106*, 147–162. https://doi.org/10.1016/j.future.2019.11.042

15. Wei, S., & Yu, H. (2018). **Cloud resource management using reinforcement learning: A survey**. *Journal of Computer Science and Technology*, *33*(3), 436–456. https://doi.org/10.1007/s11390-018-1833-2