HIGH SCHOOL EDITION
Journal of Student Research
Volume 13 Issue 3 (2024)

Using Reinforcement Learning Algorithms to Dynamically Allocate Computing Resources in Cloud Environments

Nirav Jaiswal[1] and Hao-Lun Hsu[#]

[1]Foothill High School, USA
[#]Advisor

ABSTRACT

Finding the method to most efficiently allocate resources in cloud computing environments has long been a challenge facing the cloud computing field, necessitating unique strategies to optimize performance while minimizing costs. The dynamic nature of cloud computing environments paired with users' desire for cost-effective solutions requires intelligent and adaptive resource allocation methods. In this paper, we investigate the possibility of utilizing reinforcement learning (RL) algorithms to effectively handle this resource allocation problem. The findings provide insight into the possibility of using RL algorithms for cloud resource management and provide a base for further research and exploration in the area.

Introduction

Cloud computing has recently risen to the forefront of modern technologies by providing innovative access to computing resources. Cloud computing environments provide unmatched scalability, flexibility, and cost-effectiveness, making them important tools for a wide array of applications. As popular cloud computing providers aim to provide high performance at competitive costs, dynamic allocation of resources has emerged as a leading challenge.

The task of efficiently allocating resources in cloud environments consists of efficiently and effectively spreading compute resources across numerous workloads to achieve the optimal performance for a minimized cost. Traditional fixed-resource models prove ineffective due to dynamic and unpredictable workloads hitting systems: often creating situations with too many resources, or not enough. Furthermore, the cloud's pay-as-you-go pricing model necessitates cost-conscious resource utilization to avoid unnecessary expenses.

The application of machine learning to deal with the complexities of dynamically allocating resources in these cloud environments have shown promise. Amongst these methods, reinforcement learning (RL) shows great potential because of its ability to have agents make sequential decisions, adapt their resource allocation strategies based on real-time feedback, and learn optimal or near-optimal policies through trial and error.

This paper investigates the application of different RL algorithms, specifically focusing on Deep Q-Network (DQN) and Proximal Policy Optimization (PPO), to try and tackle efficiently and effectively dynamically allocating resources in cloud environments.

The results of this research showcase the potential of DQN and PPO in effectively allocating computing resources in dynamic cloud environments. DQN's capacity to handle continuous action spaces and PPO's robustness in complex scenarios position them as valuable solutions for adaptive resource allocation. The findings contribute to a deeper understanding of RL algorithms' efficacy in cloud resource management, paving the way for more efficient cloud services and adaptive policies.

Literature Review

Reinforcement Learning

Reinforcement Learning (RL) stands as a robust framework for addressing sequential decision-making challenges, making it particularly well-suited for optimizing the allocation of computing resources in dynamic cloud environments.

MDP Formulation

At the core of RL lies the formulation of problems as Markov Decision Processes (MDPs). This framework encapsulates states, actions, rewards, and transition probabilities, allowing RL agents to learn policies that maximize cumulative rewards over time. In the context of cloud resource allocation, the MDP formulation provides a structured way to model the interactions between agents and the cloud environment.

Reinforcement Learning Algorithms

The integration of Reinforcement Learning (RL) algorithms has significantly reshaped the landscape of sequential decision-making problems, offering solutions that adapt dynamically to various domains. Within the RL paradigm, Proximal Policy Optimization (PPO) and Deep Q-Network (DQN) emerge as standout algorithms, each possessing unique attributes and capabilities.

Proximal Policy Optimization (PPO): Proximal Policy Optimization (PPO) stands as a robust policy-based RL algorithm recognized for its direct optimization of actions, independent of value function reliance. The key to PPO's success is its utilization of a surrogate objective function, which guides policy updates through a controlled "proximal" constraint. This methodology engenders stability and efficiency in learning dynamics, particularly advantageous in cases involving continuous action spaces.

PPO's strengths encompass its applicability across tasks requiring precise and intricate adjustments within the action space. Its adaptability to diverse scenarios empowers systems to swiftly recalibrate strategies, resulting in improved performance while concurrently minimizing operational costs.

Deep Q-Network (DQN): Complementing policy-based approaches, the Deep Q-Network (DQN) presents a compelling value-based RL algorithm that marries Q-learning with deep neural networks. Through learning action-value functions from experience, DQN effectively handles discrete action spaces, enabling optimal decision-making.

DQN's merit lies in its suitability for scenarios where discrete choices define the decision landscape. While the dynamic nature of environments necessitates prudent exploration-exploitation balance, DQN's ability to generalize across diverse state spaces promises effective and informed resource allocation strategies.

The integration of PPO and DQN into various domains makes for adaptable, data-driven decision-making capabilities. These RL algorithms allow systems to efficiently optimize resource allocation strategies by addressing complex challenges with calculated actions.

In summary, the incorporation of Reinforcement Learning algorithms, exemplified by PPO and DQN, marks a significant advance in addressing sequential decision-making challenges. These algorithms offer insights into intelligent, adaptive strategies that navigate complex decision landscapes. Subsequent sections delve into the exploration and evaluation of these algorithms, uncovering their efficacy in various contexts.

Cloud Computing

Cloud computing has transformed the landscape of computing resource management, with its scalability and flexibility enabling efficient allocation of resources to meet varying workloads.

Auto-Scaling Groups (ASGs)

Auto-Scaling Groups (ASGs) are a commonly employed approach in cloud resource management. ASGs dynamically adjust resource allocation based on predefined thresholds, aiming to maintain performance within specified limits. However, ASGs often rely on heuristics and predefined rules, limiting their adaptability and optimization capabilities.

Cloud Computing with RL: Differences from ASGs and Advantages

In contrast, the integration of Reinforcement Learning (RL) techniques into cloud computing environments offers distinct advantages. Unlike ASGs, RL algorithms enable dynamic learning from real-time feedback, enabling adaptive resource allocation in response to fluctuating workloads. RL also offers the potential to consider a broader range of metrics, such as response times, application-specific performance, and cost functions, providing a more comprehensive and fine-grained resource allocation strategy.

By leveraging RL, this paper aims to extend beyond the constraints of ASGs, harnessing RL's capabilities to optimize performance metrics while minimizing costs. The inherent adaptability and learning capacity of RL agents present a promising avenue for achieving intelligent and efficient cloud resource management.

In summary, this literature review has explored the core concepts of reinforcement learning, particularly its MDP formulation, and its suitability for addressing sequential decision-making in cloud resource allocation. It has also highlighted the limitations of traditional ASGs and explained the potential advantages of integrating RL techniques into cloud computing environments. Building on these insights, this paper explores the application of RL algorithms for dynamic resource allocation, examining their effectiveness in optimizing performance and cost-efficiency in the context of cloud computing.

Methodology

Problem Formulation

The situation of dynamically allocating computing resources in cloud environments to minimize cost can be formulated as a Markov Decision Process (MDP). In this formulation, the goal is to optimize performance metrics, so the costs can be kept to a minimum, through the allocation of computing resources.

State Definition

The state in the MDP represents the current state of the cloud environment, this includes the performance metric, cost metric, and amount of allocated resources at that moment.

Action Definition

The action space in the MDP consists of a discrete decision to increase or decrease the computing resources by 1, this would be the equivalent of horizontally scaling the environment.

Reward Definition

The reward function in the MDP is defined based on the cost-to-performance ratio, capturing the trade-off between increasing performance metrics and lowering costs.

Reinforcement Learning Algorithms

To address the resource allocation problem in cloud environments, we considered two state-of-the-art reinforcement learning (RL) algorithms: Deep Q-Network (DQN) and Proximal Policy Optimization (PPO).

Deep Q-Network (DQN)

DQN is an off-policy algorithm that learns from the current policy and several past policies to make discrete actions. DQN's ability to learn from past policies in its replay buffer paired makes it a strong choice for this scenario. However, its overestimation bias may prove to be a problem with the algorithm.

Proximal Policy Optimization (PPO):

PPO is an on-policy policy optimization algorithm that learns solely from the current policy. PPO would be an effective choice for this scenario because its learning from only the current sample leads to better stability. However, its computationally intensive nature might be a cause for concern.

Experiment

The Problem

The specific cloud computing problem we are aiming to solve with this experiment is the optimization of cost to performance ratio in a cloud computing environment through the dynamic allocation of computing resources. The goal is to develop and evaluate reinforcement learning algorithms that can effectively allocate resources in a cloud environment based on the defined cost-to-performance trade-off.

Gym Environment Construction

To evaluate the performance of the reinforcement learning algorithms, we built a custom environment using the OpenAI Gymnasium framework that reflects the setting described in Method Section 1.

CloudEnvironment Gym

The CloudEnvironment Gym is designed to simulate a cloud environment and encapsulates the problem formulation as an MDP. It includes the following components and methods:

- Observation Space: The observation space consists of performance metrics, cost metrics, and resources currently allocated. It is a three-dimensional box, as described in the Method section.
- Action Space: The action space is discrete and allows for two actions: increasing or decreasing the allocated computing resources by a set amount.
- Step Function: The step function updates the environment based on the chosen action, recalculates the performance and cost metrics, and determines the reward. It follows the dynamics described in the Method section.
- Reset Function: The reset function initializes the environment to the initial state, resetting the performance metrics, cost metrics, and available resources to their starting values.

Scenarios and Workloads

To evaluate the performance of the reinforcement learning algorithms under various scenarios and workloads, varying resource demands and traffic patterns were used. These patterns were designed to simulate real-world situations and represented fluctuating workloads requiring adaptive resource allocation.

Experimental Setup

Algorithm Selection

For the experiment, we selected the Deep Q-Network (DQN) and Proximal Policy Optimization (PPO) algorithms, as described in the Method section. These algorithms were chosen for their suitability for discrete action spaces and their effectiveness in learning resource allocation policies in cloud environments.

Hyperparameter Settings

We carefully tuned the hyperparameters of the DQN and PPO algorithms to optimize their performance. Tuning these hyperparameters led to faster and more stable learning, and ultimately better overall performance of the reinforcement learning agent.

Training and Evaluation Procedure

The training and evaluation procedure involved the following steps:

- Initialization: The DQN and PPO agents were initialized with random or pre-trained weights, depending on the experiment's requirements.
- Training: The agents interacted with the CloudEnvironment Gym by selecting actions based on their learned policies and exploring the environment. They updated their policies using the chosen RL algorithms, choosing actions that increase or decrease the performance metric to optimize the cost metric. The training process continued until the chosen criteria were met.
- Evaluation: After training, the performance of the RL agents was evaluated by running multiple episodes in the CloudEnvironment Gym. The agents' resource allocation decisions, as well as the resulting performance metrics and costs, were used for analysis.

Results

Reward Function Outputs

The evaluation of Proximal Policy Optimization (PPO) and Deep Q-Network (DQN) algorithms involved 500 simulation runs for each algorithm. The resulting reward function outputs are summarized below in Table 1, presenting the mean value and a range to provide insight into the algorithms' performance.

Table 1. Mean Reward Function Outputs and Ranges for PPO and DQN Algorithms

Algorithm	Average Reward (Mean ± 1 Std)± Std
PPO	7.987 ± 2.038
DQN	8.890 ± 1.671

The collected reward function outputs will undergo further statistical analysis to extract meaningful insights into the performance of both algorithms. Mean, standard deviation, and other relevant statistical measures will be computed to provide a comprehensive understanding of their behavior.

In summary, the preliminary results in Table 1 showcase the mean reward values and corresponding ranges for PPO and DQN algorithms. Further statistical analysis will contribute to a more detailed interpretation of their performance in dynamic resource allocation within cloud environments.

Analysis

Algorithm Performance

From the final reward function outputs, it is apparent that both PPO and DQN algorithms demonstrate promising performance in the context of dynamically allocating computing resources. The mean reward values of 7.987 for PPO and 8.890 for DQN highlight the efficacy of both algorithms in optimizing the trade-off between performance enhancement and cost minimization.

Additionally, the ranges of reward values, represented as the mean ± standard deviation, provide insights into the variability of algorithm performance across the multiple simulation runs. The relatively small standard deviations (2.038 for PPO and 1.671 for DQN) suggest that the algorithms consistently produce favorable results with relatively low variance.

Comparisons

Comparing the mean reward values, DQN demonstrates a slightly higher mean reward than PPO. However, the significance of this difference requires further investigation, potentially through statistical significance tests. Notably, both algorithms showcase performance that is consistent with their design principles—PPO's policy-based approach and DQN's value-based methodology.

The outcomes of this analysis have several implications for cloud resource allocation. Both PPO and DQN algorithms exhibit the potential to offer intelligent and adaptable solutions for optimizing resource allocation decisions. The algorithms' ability to consistently yield rewards close to their respective means reinforces their robustness and applicability in addressing dynamic cloud computing environments.

Future Direction

While this analysis provides valuable insights into the capabilities of PPO and DQN algorithms, further investigation could explore additional dimensions. The inclusion of statistical significance tests, the examination of individual simulation runs, and the exploration of hyperparameter settings can deepen our understanding of algorithm behaviors and identify opportunities for optimization.

In conclusion, the final reward function outputs demonstrate the effectiveness of Proximal Policy Optimization and Deep Q-Network algorithms in dynamically allocating computing resources within cloud environments. The insights gained from this analysis underscore the potential of these algorithms for enhancing resource allocation decisions in real-world cloud computing scenarios.

Conclusion

This research paper delved into the dynamic resource allocation challenge within cloud computing environments, exploring the potential of Reinforcement Learning (RL) algorithms to optimize performance metrics while minimizing costs. Through the design of a custom Gym environment and rigorous evaluation of Proximal Policy Optimization (PPO) and Deep Q-Network (DQN) algorithms, we have shed light on the efficacy of RL-based solutions in addressing this complex problem.

Key Contributions

The primary contributions of this study lie in the empirical exploration and analysis of PPO and DQN algorithms for cloud resource allocation. By conducting 500 simulation runs for each algorithm and meticulously analyzing their reward function outputs, we obtained meaningful insights into their performance dynamics. The presented reward function outputs provided a quantitative representation of how well these algorithms strike a balance between enhancing performance and minimizing costs.

Real World Application

The findings of this research paper bear practical significance for cloud computing practitioners and system architects. The demonstrated potential of RL algorithms in optimizing cloud resource allocation aligns well with the dynamic and variable nature of real-world cloud environments. The outcomes can guide the development of adaptive and intelligent resource allocation strategies that improve performance while minimizing operational costs.

Concluding Remarks

In conclusion, this research highlights the transformative potential of Reinforcement Learning algorithms, exemplified by PPO and DQN, in addressing the intricate resource allocation challenges within cloud computing environments. The empirical evaluation and analysis presented in this paper contribute valuable insights to the ongoing discourse on effective cloud resource management. As cloud computing continues to evolve, these insights can drive advancements in optimal resource allocation strategies that shape the future of efficient and cost-effective cloud-based systems.

Limitations

While our research has shown promising results regarding the application of reinforcement learning (RL) algorithms for dynamic resource allocation in cloud environments, certain limitations stopped further work on the model and its real-world application.

Firstly, resource constraints prevented us from deploying and testing the RL model in a live cloud computing environment. As a result, all testing was conducted exclusively within our custom Gym environment. Although the Gym environment allowed us to conduct initial experimentation and gain insights, it lacks the complexities of real-world cloud systems.

Moreover, the inability to test the model with real-world costs and usage patterns may limit proper application of the model without further testing and adaptation. Instead of using these real-world patterns, we resorted to using predetermined cost multipliers and randomly generated usage patterns for testing purposes;

these artificial setups may not fully capture the complexities and variabilities of actual cloud workloads, possibly impacting the model's use in certain scenarios.

Additionally, our research focused on two popular RL algorithms, DQN and PPO, out of a vast array. While DQN and PPO displayed promising performance in our experimental setup, other RL algorithms might offer unique strengths or be better suited to specific challenges. A more comprehensive study considering a broader spectrum of RL techniques could provide deeper insights and potentially reveal superior approaches.

Furthermore, hyperparameter tuning, while performed diligently in our experiments, can be an iterative process with significant computational demands. Due to computational constraints, we may not have fully explored the hyperparameters of both DQN and PPO. More exhaustive hyperparameter searches could potentially lead to improved performance and better fine-tuning of the models.

Despite these limitations, our research contributes meaningful insights into the effectiveness of RL algorithms for dynamic resource allocation in cloud environments. Future work can build upon these findings by addressing the limitations we faced to explore additional RL algorithms, conduct experiments in real cloud computing environments, and perform more computationally intensive hyperparameter tuning.

Acknowledgments

I would like to thank Mr. Hao-Lun Hsu for providing me with invaluable help in learning about both the research process and the RL field.

References

Schulman, John, et al. "Proximal Policy Optimization Algorithms." 2017. *arXiv*, https://doi.org/10.48550/arXiv.1707.06347.

Fan, Jianqing, et al. "A Theoretical Analysis of Deep Q-Learning." 2020. *arXiv*, https://doi.org/10.48550/arXiv.1901.00137.

Garí, Yisel, et al. "Reinforcement Learning-based Application Autoscaling in the Cloud: A Survey." 2020. *arXiv*, https://doi.org/10.48550/arXiv.2001.09957.

Allen, Michael, et al. "Developing an OpenAI Gym-compatible Framework and Simulation Environment for Testing Deep Reinforcement Learning Agents Solving the Ambulance Location Problem." 2021. *arXiv*, https://doi.org/10.48550/arXiv.2101.04434.