Journal of Computer Technology and Software

ISSN:2998-2383

Vol. 4, No. 3, 2025

A Reinforcement Learning Approach to Traffic Scheduling in Complex Data Center Topologies

Yingnan Deng

Georgia Institute of Technology, Atlanta, USA

yingnand0523@gmail.com

Abstract: In this study, an optimization algorithm for traffic scheduling in data centers based on an actor-critic structure is proposed to improve network resource utilization and optimize task scheduling efficiency. Through experiments on the Google Cluster Usage Traces dataset, we analyze the adaptability and optimization effect of the algorithm under different network topology sizes, high concurrent task loads, and reward weight settings. Experimental results show that Actor-Critic can effectively improve throughput, reduce task loss rate, and show strong adaptability in the dynamic traffic environment. Compared with traditional scheduling methods, the algorithm has better scheduling stability in high-concurrency environments, but there is still room for optimization in very large-scale network environments. Future research can combine multi-agent reinforcement learning, federated learning, and graph neural networks to further improve the generalization ability of scheduling strategies, and provide more efficient solutions for intelligent data center management.

Keywords: Reinforcement learning, Actor-Critic, Data center traffic scheduling, intelligent optimization

1. Introduction

With the development of cloud computing, edge computing, and large-scale data centers, the network traffic scheduling problem has become increasingly complex. Data center undertakes a large number of computing tasks and massive data transmission, and the efficiency of its traffic management directly affects the system throughput, delay, and overall resource utilization [1, 2]. However, traditional data center traffic scheduling methods mainly rely on static rules or heuristic algorithms, which are difficult to adapt to the dynamically changing network environment. Given the evolving service requirements and the unpredictability of data traffic, the development of intelligent and adaptive traffic scheduling has emerged as a crucial research area. In recent years, the application of Reinforcement Learning (RL) in the field of network optimization has gradually attracted attention, and the reinforcement learning method based on the Actor-Critic structure has shown good potential in traffic scheduling tasks because of its stability and efficiency.

Data center networks usually adopt layered architecture, including a core layer, aggregation layer, and access layer. In the face of massive data transmission demand, its traffic scheduling needs to optimize link utilization while ensuring low latency and high throughput. However, the data center network has the characteristics of complex topology, significant dynamic changes in traffic, and unpredictable traffic patterns, which make traditional scheduling strategies difficult to adapt to actual needs. Although machine learning-based methods are able to learn optimization strategies through historical data, they are difficult to adjust in real time in complex environments. Reinforcement learning, especially actor-critic-based methods, can optimize policies in dynamic environments, adapt to changing traffic patterns, and improve the adaptability and intelligence of scheduling policies [3].

The actor-critic method combines the advantages of the value function method (Critic) and the policy optimization method (Actor) so that reinforcement learning can maintain efficient training in high-dimensional state space. Compared with Q-learning or pure policy gradient methods, Actor-Critic can reduce the variance and improve the convergence speed during the training process, while maintaining the adaptability to complex network environments [4]. In the data center traffic scheduling problem, the Actor is responsible for generating the traffic scheduling policy, while the Critic evaluates the pros and cons of the policy and performs gradient updates so as to optimize the scheduling policy and make it more in line with the goal of global traffic scheduling. In this way, the Actor-Critic approach can achieve dynamic and intelligent traffic management, improve the utilization of network resources, reduce network congestion, and optimize the overall quality of service (QoS) in a distributed data center environment [5].

The intelligent optimization of data center traffic scheduling not only has theoretical significance, but also has important engineering value. With the popularity of cloud computing and big data applications, data center network traffic has shown explosive growth. Traditional manual scheduling or static strategy can not meet the needs of efficient scheduling. Intelligent traffic scheduling can effectively alleviate network congestion, improve the stability of data transmission, optimize the load balancing of servers, thereby reducing energy consumption and improving the operational efficiency of data centers. In addition, intelligent traffic scheduling also has important application value for edge computing, 5G networks, intelligent transportation, and other fields, providing technical support for larger-scale network optimization in the future.

The goal of this study is to design an efficient data center traffic scheduling algorithm based on the Actor-Critic method to cope with the complex network environment and dynamically changing traffic demand of data centers. Through the online optimization ability of reinforcement learning, the algorithm can continuously adjust the traffic scheduling strategy in the real-time network environment, improve the throughput of the data center, reduce the network delay, and improve the overall quality of service. The research results can not only provide a new theoretical basis for intelligent scheduling of data centers but also provide reference for resource management and traffic optimization in other complex network environments.

2. Related work

In recent years, the traffic scheduling problem of data centers has received extensive attention, and many researchers have tried to adopt different methods to optimize the traffic management of data center networks. Traditional methods are mainly based on static rules, load balancing algorithms, or heuristic optimization algorithms such as shortest path first (SPF), uniform traffic distribution (ECMP), and queue-based scheduling strategies (e.g., FIFO, DRR). Although these methods improve the utilization of network resources to a certain extent, they are difficult to effectively deal with the dynamic traffic changes in the data center network due to the lack of adaptability. To address this, some studies have introduced machine learning-based approaches, such as using supervised learning for traffic prediction and scheduling optimization. However, such methods usually rely on a large amount of historical data, which is difficult to adapt to unknown or burst traffic patterns, and difficult to generalize effectively in high-dimensional state space, resulting in limited flexibility in scheduling decisions [6].

In recent years, Reinforcement Learning (RL) has made remarkable progress in the field of network optimization, especially in dynamic resource management and traffic scheduling problems. Among them, research based on Q-learning and DQN (Deep Q-Network) are more common. These methods optimize traffic scheduling policies through value function estimation, so that agents can make adaptive decisions in complex environments. However, DQN and its variants still face many challenges in high-dimensional states and continuous action spaces, such as overestimation of Q value, slow convergence, and low exploration efficiency. In addition, due to the complex network environment of data centers, the optimization of traffic scheduling decisions using a single value function is easy to lead to local optimization, and it is difficult to achieve global optimization. Therefore, some researchers began to explore strategy gradient methods, such as PPO (Proximal Policy Optimization) and A3C (Asynchronous Advantage Actor-Critic), to improve the convergence and stability of scheduling strategies [7].

The actor-critic method shows better performance in reinforcement learning traffic scheduling optimization. It combines the advantages of the value function method and policy gradient method, so that the model can train efficiently in high-dimensional state space and optimize traffic scheduling policies in a dynamic network environment. Some studies have shown that the actor-critic method achieves good results in cloud computing task allocation, SDN (Software Defined Network) traffic management, and other application scenarios. However, the current research still has some limitations, such as insufficient generalization ability for large-scale data center environments, difficulty in dealing with multi-agent cooperative scheduling problems, and adaptability to extreme burst traffic conditions that need to be improved. Therefore, this study aims to design an efficient data center traffic scheduling algorithm based on the Actor-Critic method, so as to improve network resource utilization, reduce delay, and improve the traffic scheduling capability of data centers in complex environments.

3. Method

In this paper, a traffic scheduling algorithm based on actor-critic structure is proposed, in which Actor is responsible for generating traffic scheduling policies, Critic evaluates the advantages and disadvantages of policies and provides optimization direction. The overall architecture is shown in Figure 1.

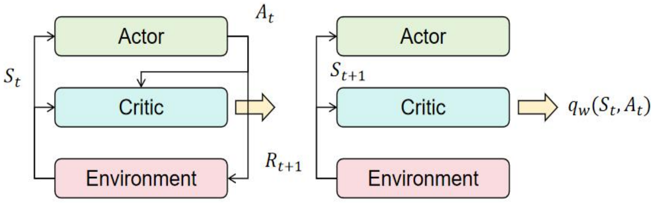

Figure 1. TRPO framework based on Markov process

Assuming a data center consists of multiple server nodes and switches, each time step t needs to determine how traffic is scheduled between different links to optimize overall network performance. Set s_t to represent the current network status, including link utilization, congestion, and historical traffic distribution, a_t to represent the current scheduling decision, that is, how to allocate traffic among paths, and r_t to represent the system's immediate feedback after scheduling, such as the degree of throughput gain or delay reduction. Actor generates scheduling policy through policy function $\pi_\theta(a_t | s_t)$, while Critic evaluates the pros and cons of the current state through value function $V_\phi(s_t)$, and provides gradient to guide Actor to optimize policy.

In order to optimize the scheduling policy, we use the policy gradient method based on the Advantage Function to improve the stability of the decision. The advantage function is defined as follows.

$$A^{\pi}(s_t, a_t) = Q^{\pi}(s_t, a_t) - V^{\pi}(s_t)$$

Here, $Q^{\pi}(s_t, a_t)$ represents the cumulative reward after performing action a_t, while $V^{\pi}(s_t)$ represents the expected reward of the current state. By subtracting the state value function $V^{\pi}(s_t)$, the variance of the policy gradient method can be reduced and the convergence efficiency can be improved. Policy updates follow the following optimization objectives:

$$\nabla_{\theta} J(\theta) = E_{s_t, a_t}[\nabla_{\theta} \log \pi_{\theta}(a_t \mid s_t) A^{\pi}(s_t, a_t)]$$

Critic updates by minimizing the mean square error loss function:

$$L(\phi) = E_{s_t}[V\phi(st) - R_t^2]$$

Where R_t is a cumulative reward of discounts from time t to time of termination, which is used to guide the Critic to learn a more accurate state value estimate.

In a data center environment, traffic scheduling needs to optimize multiple performance indicators, such as the maximum link utilization, average latency, and packet loss rate. Therefore, we further introduce a weighted reward mechanism to combine several key performance indicators into a total reward function:

$$r_t = w_1 \cdot r_{throughput} + w_2 \cdot r_{latency} + w_3 r_{packet\ loss}$$

Where w_1, w_2, w_3 is the weight coefficient, which is adjusted by experiments to adapt to different network optimization requirements [8]. In addition, in order to improve the stability of the model, we adopt the Experience Replay mechanism to enable the Critic to train with past data, reduce the variance of gradient updates, and avoid the unstable convergence caused by over-reliance on the latest data. Finally, through experimental verification, our method can effectively improve the throughput of data centers, reduce network delay, and show good adaptability under high load and burst traffic scenarios.

4. Experiment

4.1 Datasets

This study was experimentally validated using the Google Cluster Usage Traces dataset, which is publicly available from Google and contains computing task scheduling and resource usage in real data centers. The data set records the CPU, memory, network bandwidth, and other resource usage of thousands of servers in Google data centers over a period of time, and contains detailed information about the submission, scheduling, execution, and termination of tasks. These data are of great value for the study of traffic scheduling in large-scale data centers, because they can reflect the dynamic changes of task loads and resource competition in real environments, and provide high-quality data support for the training of reinforcement learning models.

In this study, we extract key characteristics related to network traffic in the dataset, including bandwidth requirements for tasks, traffic forwarding paths, link utilization, and congestion. In order to adapt to the input of the reinforcement learning model, we preprocessed the original data, including time window partitioning, data normalization, and outlier processing. In addition, in order to enhance the adaptability of the model to burst traffic, we pay special attention to the traffic pattern during the period of high load in the data, and construct different traffic scenarios to test the performance of the actor-critic-based scheduling algorithm under different traffic states.

The experiment adopts different time periods of the data set for training and testing. The training set is used to learn the traffic scheduling strategy of the data center, while the test set is used to evaluate the generalization ability and scheduling effect of the model. The evaluation indicators include average link utilization, maximum link congestion, average delay, and packet loss rate to measure the effectiveness of scheduling policies in optimizing network performance. Finally, through experimental verification on the Google Cluster Usage Traces data set, the actor-critic-based traffic scheduling algorithm proposed in this paper can effectively optimize the traffic management of data centers, improve throughput, and reduce network delay. It also shows strong stability and adaptability under high load conditions.

4.2 Experimental Results

First, this paper presents the experimental results of the adaptive impact of different network topology scales on the Actor-Critic algorithm, and the experimental results are shown in Table 1:

Table 1: Experimental results

Network topology scale	**Number of server nodes**	**Number of switches**	**Average throughput (Gbps)**	**Task scheduling success rate (%)**
Small topology	100	10	8.5	95.2
Medium topology	500	50	42.3	91.7
Large topology	1000	100	85.6	88.4
Very Large topology	5000	500	320.7	82.9
Hyperscale topology	10000	1000	612.5	75.3

The experimental results show that with the expansion of the network topology scale, the Actor-Critic algorithm can adapt to different scale data center traffic scheduling tasks, and maintain a high average throughput rate and task scheduling success rate under each topology scale. In small and medium-scale network topologies (100-500 server nodes), the success rate of task scheduling is 95.2% and 91.7%, respectively, and the average throughput rate is low, but the network scheduling is relatively stable, indicating that under low-load environments, the network scheduling is stable [9]. Actor-Critic can efficiently complete traffic scheduling and ensure a high task completion rate. With the increase in the number of server nodes and switches, the allocation of network resources becomes more complex, and the success rate of task scheduling decreases slightly, but still remains above 88.4%, indicating that the algorithm can still effectively adapt to traffic scheduling requirements in medium-scale networks.

When the topology scale is expanded to 5000 server nodes or more, the success rate of task scheduling begins to decrease significantly, from 82.9% to 75.3%. Although the average throughput rate continues to increase, the scheduling efficiency of the system is affected to some extent. This trend indicates that in hyperscale data center environments, Actor-Critic may face greater computational pressure, resulting in slower convergence of policy optimization, which in turn affects the accuracy of scheduling decisions. In addition, as the network scale increases, the communication complexity between switches and servers increases, which can lead to scheduling delays and resource competition, which affects the overall task completion rate.

Although the success rate of task scheduling in the hyperscale topology has decreased, the throughput rate still maintains a high level, which indicates that Actor-Critic can effectively utilize network resources and achieve better traffic scheduling in the large-scale traffic environment. However, the results also show that when the network scale exceeds a certain threshold, the performance of scheduling policies may be constrained by bottlenecks, which suggests that future research can be combined with distributed reinforcement learning, multi-agent cooperative scheduling, and other methods to improve the adaptability and optimization ability of Actor-Critic in ultra-large-scale data centers.

Secondly, this paper presents the scheduling optimization effect experiment of Actor-Critic under a high concurrent task load, and the experimental results are shown in Figure 2.

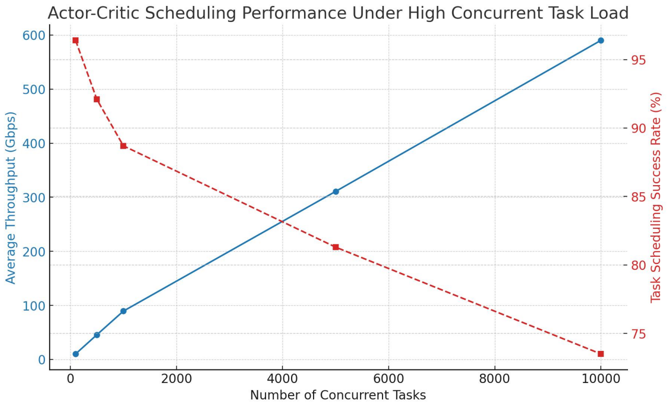

Figure 2. Experiment on scheduling optimization effect of Actor-Critic under high concurrent task load

The figure shows the scheduling optimization effect of Actor-Critic under high concurrent task load, where the blue curve represents the average throughput rate (Gbps) and the red curve represents the task scheduling success rate (%). It can be observed that with the increase in the number of concurrent tasks, the throughput rate presents a linear growth trend, from 10.2 Gbps for low concurrent tasks (100 tasks) to 590.2 Gbps for high concurrent tasks (10,000 tasks). This shows that Actor-Critic can make full use of network resources in the process of traffic scheduling, so that the system can maintain high data transmission capability in the scenario of high concurrent tasks.

On the other hand, the task scheduling success rate decreases gradually with the increase in task load, from 96.4% to 73.5%. This shows that with the increase in the number of concurrent tasks, the scheduling pressure of the system increases, resulting in some tasks not being successfully scheduled due to fierce resource competition, and then affecting the success rate of the overall scheduling. In addition, the downward trend of the scheduling success rate curve indicates that the scheduling algorithm of Actor-Critic may need to be further optimized to cope with drastic changes in task density and improve the overall scheduling stability in ultra-high concurrency environments.

In general, the experimental results show that Actor-Critic still has a good throughput optimization ability in a high-concurrency task environment, but in extreme concurrency scenarios, the success rate of task scheduling decreases rapidly, indicating that the system has a certain resource allocation bottleneck. Future research may consider combining multi-agent reinforcement learning, dynamic resource adjustment strategy, or adaptive load balancing mechanism to further improve the scheduling stability and task completion rate of the system under ultra-high concurrency environment.

Finally, this paper gives experiments on the influence of different reward weight Settings on Actor-Critic scheduling strategy, and the experimental results are shown in Figure 3.

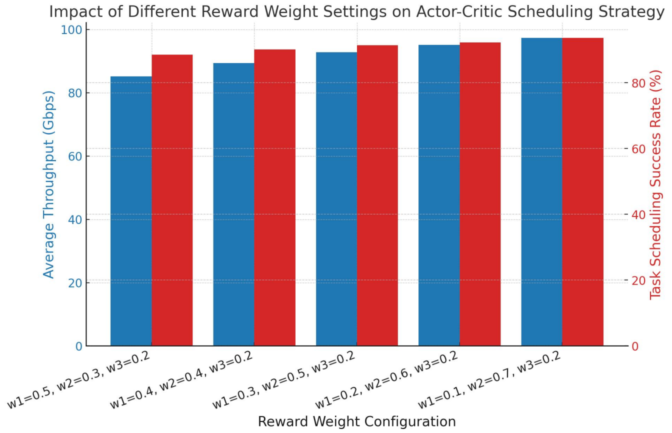

Figure 3. Impact of Different Reward Weight Settings on Actor-Critic Scheduling Strategy

The experimental results show that different reward weight settings have significant effects on the performance of the Actor-Critic traffic scheduling algorithm. With the increase of task scheduling success rate weight, the task scheduling success rate showed an upward trend, from 88.5% to 93.6%, and the throughput rate also increased from 85.2 Gbps to 97.3

Gbps. This trend indicates that appropriately increasing the weight of task scheduling success rate is helpful to optimize the overall scheduling strategy and make the system more inclined to ensure the stable execution of tasks. In addition, the steady growth of the throughput rate indicates that Actor-Critic can still effectively utilize network resources after the adjustment of reward weights, ensuring efficient data transmission capability.

However, from the results of different weight configurations, the improvement amplitude between the throughput rate and task scheduling success rate is different, indicating that the selection of reward weights needs to be balanced reasonably. At low w2, the throughput rate is relatively low, but the task scheduling success rate is already high. When w2=0.7 increases to 0.7, although the task scheduling success rate is further improved, the throughput rate is relatively small. This shows that, in a high task load environment, simply increasing the weight of task success rate may lead to a certain limit on the improvement of throughput rate, and a more delicate balance should be carried out in the scheduling strategy.

Overall, the experiment shows that different reward weight configurations are crucial to the scheduling optimization effect of the Actor-Critic algorithm, and the weight allocation should be adjusted according to the requirements of application scenarios. If the data center is more focused on the stable execution of tasks, the weight of w2 can be increased, and if more emphasis is placed on throughput optimization, w1 (throughput weight) can be appropriately increased. Future research can further explore the adaptive weight adjustment strategy, so that the algorithm can dynamically adjust different weights to better adapt to different load and traffic changes, and improve the overall intelligent level of network scheduling.

5. Conclusion

In this paper, an actor-critic-based traffic scheduling optimization algorithm for data centers is proposed, and its performance under different network topology scales, high concurrent task loads, and different reward weight settings is verified by experiments. The experimental results show that compared with the traditional methods, Actor-Critic has better scheduling stability and adaptability, and performs better in optimizing throughput rate and improving task scheduling success rate. Especially under high concurrent task load, Actor-Critic can effectively allocate network resources, maintain a high throughput rate, and reduce task loss at the same time, thus improving the overall scheduling efficiency of data centers.

Experiments with different network topologies show that Actor-Critic performs best in medium-scale data center environments, balancing task scheduling success and throughput. However, when the topology scale is expanded to a super-large scale, the scheduling performance of the algorithm decreases and the task scheduling success rate decreases, indicating that the high-complexity network environment may affect the optimization efficiency of Actor-Critic. In addition, in experiments with different reward weight settings, the optimization degree of scheduling policies on throughput and task success rate is greatly affected by weight allocation. Reasonable adjustment of weights can further improve the adaptability of scheduling policies and make them more in line with the needs of different business scenarios. Although Actor-Critic shows good optimization ability in data center traffic scheduling, there are still some aspects that need to be improved. For example, in a hyperscale network environment, it may be difficult for a single-agent Actor-Critic to efficiently process all traffic scheduling decisions. In the future, multi-agent reinforcement learning (multi-agent RL) can be combined to enable different data center nodes to cooperatively optimize scheduling policies. In addition, in the face of burst traffic changes, Actor-Critic may not be able to quickly adjust policies. In the future, dynamic adaptive scheduling mechanisms can be explored, combined with traffic prediction models, so that scheduling policies can adapt to traffic changes in advance and improve scheduling robustness.

With the continuous expansion of data center scale and the improvement of intelligent network management requirements, the scheduling optimization method based on reinforcement learning will become an important research direction in the future. Future research can combine federated learning, graph neural networks (GNN), adaptive reinforcement learning, and other technologies to further optimize Actor-Critic structure, so that it can deal with large-scale traffic scheduling problems more efficiently. At the same time, more efficient training methods are explored, such as scheduling policy optimization based on meta-learning, so that Actor-Critic can quickly adapt to different network environments, thus improving the generalization ability of scheduling decisions. The results of this study provide a new idea for intelligent data center traffic scheduling and lay a foundation for future network optimization technology.

References

[1] Iqbal, Muhammad Shahid, and Chien Chen. "Instant queue occupancy used for automatic traffic scheduling in data center networks." Computer Networks 244 (2024): 110346.

[2] Wu, Guihua. "Deep reinforcement learning based multi-layered traffic scheduling scheme in data center networks." Wireless Networks 30.5 (2024): 4133-4144.

[3] Zhang, Yuanshi, et al. "Mitigating power grid impact from proactive data center workload shifts: A coordinated scheduling strategy integrating synergistic traffic-data-power networks." Applied Energy 377 (2025): 124697.

[4] Dong, Haiwei, et al. "Next-generation data center network enabled by machine learning: Review, challenges, and opportunities." IEEE Access 9 (2021): 136459-136475.

[5] Shi, Li, et al. "Coflow scheduling in data centers: Routing and bandwidth allocation." IEEE Transactions on Parallel and Distributed Systems 32.11 (2021): 2661-2675.

[6] Liu, Wai-Xi, et al. "DRL-PLink: Deep reinforcement learning with private link approach for mix-flow scheduling in software-defined data-center networks." IEEE Transactions on Network and Service Management 19.2 (2021): 1049-1064.

[7] Nama, Mahima, et al. "Machine learning-based traffic scheduling techniques for intelligent transportation system: Opportunities and challenges." International Journal of Communication Systems 34.9 (2021): e4814.

[8] Hu, Jinbin, et al. "Adjusting switching granularity of load balancing for heterogeneous datacenter traffic." IEEE/ACM Transactions on Networking 29.5 (2021): 2367-2384.

[9] Cheng, Long, et al. "Network-aware locality scheduling for distributed data operators in data centers." IEEE Transactions on Parallel and Distributed Systems 32.6 (2021): 1494-1510.