Proceedings of the 6th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/64/20241359

Application of machine learning optimization in cloud computing resource scheduling and management

Yifan Zhang[1,*], Bo Liu[2,7], Yulu Gong[3,8], Jiaxin Huang[4,9], Jingyu Xu[5,10], Weixiang Wan[6,11]

[1]Executive Master of Business Administration, Amazon Connect Technology Services (Beijing), Xi'an, Shaanxi, China
[2]Software Engineering, Zhejiang University, HangZhou China
[3]Computer & Information Technology, Northern Arizona University, Flagstaff, AZ, USA
[4]Information Studies, Trine University, Phoenix, USA
[5]Computer Information Technology, Northern Arizona University, Flagstaff, AZ, USA
[6]Electronics & Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China

*Corresponding author: yifan.ibm@gmail.com
[7]lubyliu45@gmail.com
[8]yg486@nau.edu
[9]jiaxinhuang1013@gmail.com
[10]jyxu01@outlook.com
[11]danielwanwx@gmail.com

Abstract. In recent years, cloud computing has been widely used. Cloud computing refers to the centralized computing resources, users through the access to the centralized resources to complete the calculation, the cloud computing center will return the results of the program processing to the user. Cloud computing is not only for individual users, but also for enterprise users. By purchasing a cloud server, users do not have to buy a large number of computers, saving computing costs. According to a report by China Economic News Network, the scale of cloud computing in China has reached 209.1 billion yuan.Rational allocation of resources plays a crucial role in cloud computing. In the resource allocation of cloud computing, the cloud computing center has limited cloud resources, and users arrive in sequence. Each user requests the cloud computing center to use a certain number of cloud resources at a specific time.

Keywords: Cloud computing, Resource scheduling, Machine learning optimization, Artificial intelligence.

1. Introduction

In recent years, cloud computing has been widely used. Cloud computing refers to the centralized computing resources, users through the access to the centralized resources to complete the calculation, the cloud computing center will return the results of the program processing to the user. Cloud

DOI: 10.54254/2755-2721/64/20241359

computing is not only for individual users, but also for enterprise users. By purchasing a cloud server, users do not have to buy a large number of computers, saving computing costs. Cloud service providers can dynamically schedule computing resources according to users' access requirements to maximize computing resource utilization efficiency. According to a report by China Economic News Network, the scale of cloud computing in China has reached 209.1 billion yuan. At present, the more mature cloud service providers in China are Ali Cloud, Baidu Cloud, Huawei Cloud and so on.

Rational allocation of resources plays a crucial role in cloud computing. In the resource allocation of cloud computing, the cloud computing center has limited cloud resources, and users arrive in sequence. Each user requests the cloud computing center to use a certain number of cloud resources at a specific time. The resource allocation of cloud computing needs to consider various needs of users. Different users have different requirements on cloud resources. This study mainly considers assigning users to servers that are close to improve the service quality of users. However, if users are assigned to servers that are close to them, some servers may be congested and the waiting time will be long. Therefore, this study considers a cloud resource allocation method based on deep reinforcement learning. Deep reinforcement learning can determine the dynamic allocation of resources according to the current state of the system, thus maximizing the utilization efficiency of cloud resources and reducing user waiting time.

2. Related work

2.1. Resource scheduling by deep reinforcement learning

In their research, Mao et al. (2016) used deep learning methods to explore resource allocation in cloud computing environments.provides important reference and inspiration for the application of deep learning in the field of cloud computing resource management, and provides new ideas and methods for solving resource allocation problems.

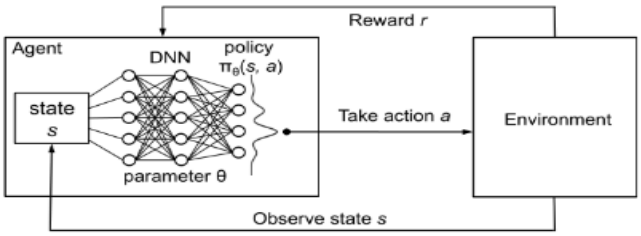

Figure 1. Reinforcement learning methods for policy networks (Figure: Mao et al. (2016))

As shown in Figure 1, the input of the neural network is the current state of the system, and the output is the action. When the system adopts the action, the environment will return the corresponding reward. The goal of system optimization is to minimize the user's waiting time, that is, to maximize it:

$$\nabla_\theta E_{\pi\theta}[\sum_{t=0}^{\infty}\gamma^t r_t] = E_{\pi\theta}\left[\nabla_\theta \log \pi_\theta\ (s,a) Q^{\pi\theta}(s,a)\right] \quad (1)$$

$$\nabla_\theta E_{\pi\theta}[\sum_{t=0}^{\infty}\gamma^t r_t] \quad (2)$$

y∈(0,1), where "is the discount factor. The method of strategic gradient descent is mainly to gradient the total reward :-1 where $Q^{\pi\ \theta}$ (s,a) is the expected cumulative discounted reward for taking action a in state s. The Monte Carlo method is used to calculate the cumulative discount return v, and the parameters of the neural network are updated by the following method.

$$\theta \leftarrow \theta + \alpha \sum_t \nabla_\theta \log \pi_\theta\ (s_t, a_t) v_t \quad (3)$$

DOI: 10.54254/2755-2721/64/20241359

2.2. Resource scheduling considering time-varying characteristics

The study of Mondal et al. (2021) focuses on the differences in resource utilization over different time periods when users use cloud computing resources for a long period of time, especially in cloud computing tasks such as neural network training that require a long running time. The framework of the study is as follows:

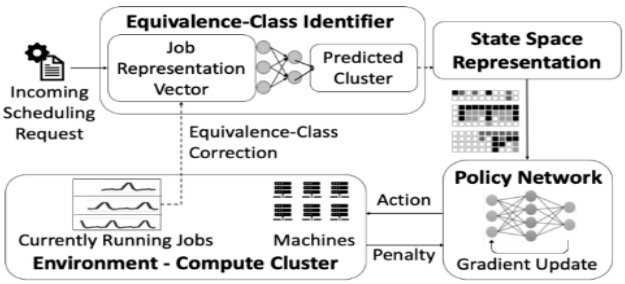

Figure 2. Deep reinforcement learning method considering time-varying features

In this framework, users' historical resource usage data is first collected and analyzed to understand patterns and trends in resource utilization over different time periods. Additionally, the inclusion of dynamic time warping facilitates the identification of users exhibiting similar temporal utilization patterns through the application of K-means clustering. In the part of deep reinforcement learning, the idea of this study is roughly similar to that of Study 1. Study 2 puts forward a number of definitions of reward worthy of reference. It mainly includes the following parts:

$$P_C = -\sum_d \sum_{m \in M} K_c * C_r(m,d) \tag{4}$$

2.2.1. Competition

The competition describes the usage of server resources by different resources. The proportion of encouragement (punishment). Cr(m.d) is mainly calculated by the following formula, representing the inner product of the resources used by different services.

$$C_r(m,d) = \sum_{W_i \in m_W} \sum_{W_J \in m_W, j>i} \langle R(W_i,d), R(W_j,d) \rangle \tag{5}$$

2.2.2. Machine utilization rate

Machine utilization is mainly evaluated for the proportion of machines currently in use. Adding this reward can make reinforcement learning allocate resources using machines rather than non-machine resources in resource allocation.

$$P_U = -\sum_d \sum_{m \in M_u} |U_m(t,d)|^{K_u} \tag{6}$$

Where U (t; d) represents the unused resource of resource d by the used machine m at time t; M. Represents the collection of machines currently in use.

2.2.3. Excessive use of penalties

$$P_O = -\sum_d \sum_{m \in M} K_o * \mathbb{I}_{m,d}[First\ overshoot\ for\ the\ TVW] \tag{7}$$

Overuse penalties are a measure of penalties given when resource usage is higher than machine usage, where |md is an indicator of overuse.

2.2.4. Use time penalties

The use of time penalties is mainly calculated by the number of requests waiting in the queue, mainly by the following formula:

DOI: 10.54254/2755-2721/64/20241359

$$P_W = -K_\omega * |Q_t| \tag{8}$$

Where $|Q_t|$ represents the number of requests waiting in the queue.

3. Experiment and methodology

3.1. Experimental environment

The CloudSim3.0.2 cloud simulation platform of the Grid Laboratory of the University of Melbourne was used in the experiment to test the performance of the author's algorithm. The experiment involved simulation comparison and result analysis with the basic ant colony algorithm (ACO)[10] and simulated annealing algorithm (SA). Firstly, in the cloud simulation platform, the MyAll-ocationTest class is created to perform the initial configuration of the cloud environment. Subsequently, cloudsim objects are created to add cloud computing tasks. Moreover, GAACO, ACO, and SA algorithms are implemented in the DatacenterBroker class. The relevant parameters of the genetic ant colony algorithm are presented in Table 1.

Table 1. Genetic ant colony algorithm parameter table

Parameter Symbol	Meaning	Value
evolution Num	Evolution generations	100
population	Population size	10
m	Number of ants	31
Pc	Crossover probability	0.35
Pm	Maximum mutation probability	0.08
A max	Maximum pheromone factor	1.00
.max	Maximum expected pheromone factor	2.00
Y max	Maximum pheromone evaporation coefficient	0.10
Q	Maximum pheromone intensity	50.00

The experimental design and result analysis in this paper focus on comparing the performance of the genetic ant colony algorithm across four key aspects: average time cost, average cost, algorithm service quality, and system resource load rate.

3.2. Experimental parameter

Initially, the task size is set to 10, with cloud computing resources consisting of 10 VMs. Each VM has a storage size of 10 GB, memory size of 256 MB, one CPU, and a bandwidth of 1,000 MB. The unit time bandwidth cost and unit time instruction cost are 0.01 yuan/s each. Tasks are experimented with in increments of 10. The quality of service for the algorithm is represented by multiQoS. The resource load rate is defined as follows:

$$resource\ load\ rate = \frac{use_i}{useAvg \times n} \tag{9}$$

The average time cost for each algorithm is calculated as the number of tasks increases by 10. The results are depicted in Figure 3. It's observed that the time cost of GAACO is superior to that of ACO, albeit longer than SA. Moreover, as the number of tasks increases, the time gap widens, with GAACO reducing time by 50.9% compared to ACO and showing a 3% difference compared to SA. It can be seen that the difference between algorithms is not large, and the average cost is only about 1%.

DOI: 10.54254/2755-2721/64/20241359

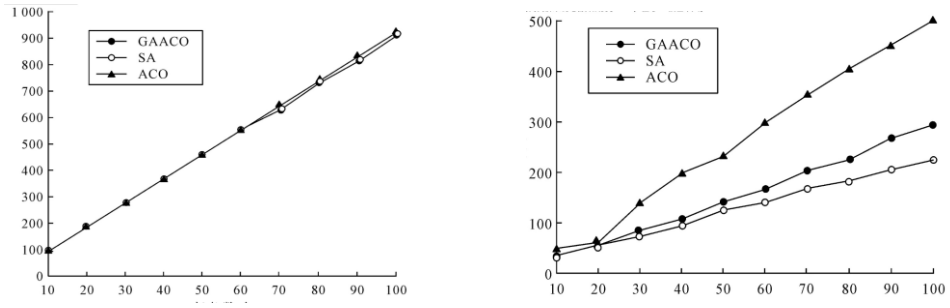

Figure 3. Average time cost of each algorithm and Cost of each algorithm

The experimental results of service quality of each algorithm are shown in Figure 4. It can be seen that the service quality of the author's algorithm and SA increases slowly with the increase of the number of tasks, while ACO presents a linear and sharp rise. Service quality is a comprehensive index of cost, time and reliability, and it can be seen that the comprehensive performance of GAACO is better than that of ACO and SA. They reduced by 14.4% and 76.8%, respectively.

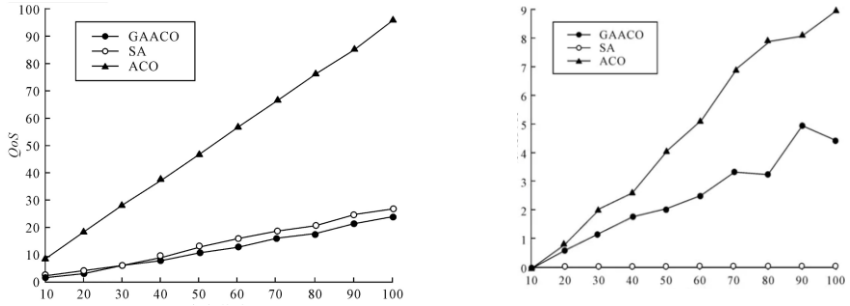

Figure 4. Service quality of each algorithm and system load of each algorithm

The experimental results regarding the algorithm's system load are presented in Figure 4. It's evident that the system load of ACO has consistently remained high, whereas GAACO exhibits a higher load compared to SA but is notably superior to ACO. Specifically, GAACO achieves a 50.2% reduction in average system load compared to ACO. Furthermore, in conjunction with Figures 2 and 3, it's observed that SA tends towards an evenly distributed task assignment to virtual machines, resulting in a system load of 0 as per equation (9).

3.3. Experimental conclusion

After conducting thorough research on cloud computing task scheduling, the proposed algorithm has been rigorously compared with both the basic ant colony algorithm and the simulated annealing algorithm across four critical aspects. The comprehensive analysis of the results reveals that the proposed algorithm consistently outperforms the other two algorithms. Notably, the proposed algorithm demonstrates superior capabilities in balancing various factors including time cost, monetary cost, reliability, and system load. This balanced optimization ensures that the algorithm can effectively meet the multidimensional quality of service (QoS) requirements of users.

4. Conclusion

In conclusion, this study presents a comprehensive approach to address the challenges of resource scheduling and management in cloud computing environments. By leveraging machine learning optimization techniques, particularly deep reinforcement learning, the proposed algorithm demonstrates significant improvements in system performance and efficiency. Through extensive experimentation and analysis, it is evident that the proposed algorithm outperforms traditional methods such as ant colony

DOI: 10.54254/2755-2721/64/20241359

optimization and simulated annealing in terms of time cost, cost effectiveness, service quality, and system resource load. Additionally, deep learning algorithms can continuously improve over time through experience, leading to enhanced performance and scalability in cloud scheduling tasks. Overall, the integration of deep learning with cloud computing scheduling holds great promise for addressing the increasingly complex and dynamic nature of modern cloud environments. AI-driven insights derived from vast amounts of cloud data will enable businesses to make more informed decisions and gain competitive advantages. As AI technologies continue to evolve, their integration with cloud computing will usher in a new era of innovation and transformation across industries, paving the way for smarter, more efficient, and more resilient digital ecosystems.

References

[1] X. Mo and J. Xu, "Energy-efficient federated edge learning with joint communication and computation design," Journal of Communications and Information Networks, vol. 6, no. 2,pp. 110–124, 2021.

[2] Q. Zeng, Y. Du, K. Huang, and K. K. Leung, "Energy-efficient resource management for federated edge learning with cpu-gpu heterogeneous computing," IEEE Transactions on Wireless Communications, vol. 20, no. 12, pp. 7947–7962, 2021.

[3] Hussain H, Malik S U R, Hameed A, et al. A Survey on Resource Allocation in High Performance Distributed Computing Systems[J]. Parallel Computing (S0167-8191), 2013, 39(11): 709-736.

[4] Bellendorf J, Mann Z Á. Classification of Optimization Problems in Fog Computing[J]. Future Generation Computer Systems (S0167-739X), 2020, 107(1): 158-176.

[5] Brogi A, Forti S, Guerrero C, et al. How to Place Your Apps in the Fog: State of the Art and Open Challenges[J]. Software: Practice and Experience (S0167-739X), 2019, 1(1): 1-8.

[6] Wu C, Li W, Wang L, et al. Hybrid Evolutionary Scheduling for Energy-efficient Fog-enhanced Internet of Things[J]. IEEE Transactions on Cloud Computing (S2168-7161), 2018, 1(1): 1-1.

[7] Atzori L, Iera A, Morabito G. The Internet of Things: A Survey[J]. Computer Networks (S1389-1286), 2010, 54(15): 2787-2805.

[8] Bonomi F, Milito R, Natarajan P, et al. Fog Computing: A Platform for Internet of Things and Analytics. N. Bessis, C. Dobre. Big Data and Internet of Things: A roadmap for smart environments[M]. Cham: Springer, 2014, 546: 169-186.

[9] Xiaoxi Zhang, Jianyu Wang, Li-Feng Lee, Tom Yang, Akansha Kalra,Gauri Joshi, Carlee Joe-Wong, "Machine Learning on Volatile Instances:Convergence, Runtime, and Cost Trade-offs", IEEE/ACM Transactions on Networking, 30(1):215—228, 2022

[10] Yichen Ruan, Xiaoxi Zhang, Shu-Che Liang,Carlee Joe-Wong, "Towards Flexible Device Participation in Federated Learning for Non-IID Data", International Conference on Artificial Intelligence and Statistics(AISTATS), 2021.

[11] Yichen Ruan, Xiaoxi Zhang, Carlee Joe-Wong, "How Valuable Is Your Data? Optimizing Device Recruitment in Federated Learning", submitted to ToN,prelimianary results are published in WiOpt 2021.