Resource Management in Cloud Computing Using Deep Reinforcement Learning: A Survey

Yuxin Feng(✉) and Feiyang Liu

AVIC, Xi'an 710068, China
609369689@qq.com

Abstract. Next generation aircrafts will not only require high-performance and intelligent computing capabilities, but also a fast design-developing-integration-update time cycle. Cloud computing technology provides a platform with large amounts of hardware resources and software services, making applications development and deployment much more convenient. Thus, airborne cloud computing becomes an important design methodology for next generation avionics system. However, the dynamic and uncertain cloud environment makes efficient resource management very complicated. Due to the characteristics of dynamic autonomous decision-making, deep reinforcement learning has become a promising resource scheduling algorithm. This paper firstly analyzes the requirements of airborne cloud computing systems, then studies the basic theory of cloud resources management and scheduling strategies. The resource management algorithms based on deep reinforcement learning (DRL), some common DRL models, experimental platforms, and evaluation parameters are introduced in details. Finally, some critical problems and challenges in the design of DRL-based resource management algorithm are summarized. This paper can provide some technique supports for the airborne cloud computing system.

Keywords: Airborne computing system · Cloud computing · Resource management · Deep reinforcement learning

1 Introduction

Avionics system is a common information processing platform in an aircraft. It can provide shared computation resources among different software applications in the form of central processing module. With the fast development of sensor systems, such as high-definition radio-frequency radar, optical radar, and distributed aperture system (DAS), next generation aircrafts must have high-performance and intelligent computing capabilities to processing the surged information. Meanwhile, the design time and full lifecycle cost of a new aircraft increase significantly recent years, next generation aircrafts should also improve the utilization of resources and speed up the design-developing-integration-update time cycle. Cloud computing is a widely used technology in commercial computing systems, it can provide an efficient platform with large amounts of hardware resources and software services, making applications development and deployment much more

Chinese Society of Aeronautics and Astronautics: CASTYSF 2022, LNEE 972, pp. 635–643, 2023.
https://doi.org/10.1007/978-981-19-7652-0_56

convenient. Thus, airborne cloud computing becomes an important design methodology for next generation avionics systems [1].

Cloud computing system provides a shared pool of configurable computing resources (*e.g.* networks, servers, storage, applications, services) that can be rapidly provisioned and released with the minimal management effort or interaction. Resource management is one of the key technologies in cloud computing. However, the dynamic and uncertain cloud environment makes efficient re-source management very complicated. With the continuous development of machine learning, researchers gradually use various algorithms to deal with the resource allocation. Reinforcement learning, is a special machine learning method that can be trained without a priori knowledge, and its policies are dynamically adaptive, which is very suitable in resource management. This paper studies the basic theory of cloud resources management and scheduling strategies, and analyzes the resource management algorithms based on deep reinforcement learning (DRL). It can provide some technique supports for the design of airborne cloud computing system.

2 Resource Management

2.1 Overview of Could Computing and Its Scheduling

In the cloud computing system, storage and computing resources are centrally placed in a public cloud resource pool, enabling customers to access them in a convenient, pay-as-you-go manner over the network. Cloud resource pools are the core of cloud services. Virtualization of cloud services is a major way to implement cloud computing to build resource pools [2]. Resource management requires the optimal utilization of virtual resources and efficient management of runtime resources to provide application-based services that meet the needs in a cost-effective manner.

Resource scheduling in cloud computing is divided into three layers: scheduling resources for applications, scheduling virtual resources (e.g. virtual machines) to physical resources, physical resource scheduling and real applications. Moreover, at each layer there can be several different objectives for optimization. In the case of resource management, for example, at the virtual resource layer, load balancing can be optimized, resource utilization such as CPU and memory shares can be improved, in addition to cost efficiency or energy savings. Therefore, the cloud resource management problem can be abstracted as a combinatorial optimization problem. A certain path or rule is used to obtain a solution that satisfies the user's requirements.

2.2 Cloud Resource Management Optimization Objectives

The cloud resource management scheduling problem is actually an optimization problem. The optimization objective, then, is closely related to the performance of the cloud computing resource scheduling system. Researchers often use SLA and QoS parameters as optimization parameters in resource scheduling, which can ensure that the requirements of SLA agreements are met, avoid violating the terms of the agreements, and efficiently integrate and allocate effective resources in the data center. From the cloud service provider's perspective, load balancing, resource utilization, and energy efficiency are the most important goals in task scheduling. Specifically, the optimization objectives of cloud resource management can be divided into four areas: energy optimization, time optimization, load balancing optimization, and other optimizations. Specifically, they are described as follows.

(1) Energy consumption optimization. It can be summarized into two categories, total energy consumption and energy utilization (energy efficiency). Energy consumption minimization is a common optimization objective in the literature.

(2) Time-related optimization. The time metrics for scheduling in cloud computing can be summarized into four types, including maximum completion time (execution time), delay time, response time, and waiting time. The optimization goals for cloud resource management are usually minimization of completion time, minimization of delay time (or delayed service), and minimization of response time.

(3) Load balancing optimization. Depending on the load object, the load balancing metrics are divided into two types: task volume load balancing and task number load balancing. The degree of load balancing can also be divided into two types of cumulative load balancing and real-time load balancing according to the time period. Maximizing load balancing is the usual optimization goal. In the literature, there are various functions used to calculate the degree of load balancing, such as variance or standard deviation, average success rate, coefficient of variance, load value and imbalance.

(4) Other optimization. It mainly includes resource utilization maximization, resource utilization minimization, etc.

2.3 Resource Management Strategies

Existing resource management methods can be divided into two main categories, traditional methods and intelligent methods. Traditional methods focus on adapting and extending traditional scheduling methods that rely almost entirely on manual computation, such as First In First Out (FIFO), shortest job first (SJF), First Come First Serve (FCFS), Round Robin (RR), Minimum-Minimum (Max-Max), and Maximum-Minimum (Max-Min). In [3], Xu et al. proposed a multi-workflow multi-QoS constrained scheduling policy (MQMW) for cloud computing. In [4], Li et al. developed a system cost function for jobs and a non-preemptive priority M/G/1 queuing model to help the policy and algorithm obtain the approximate optimal service value for each job. These scheduling algorithms are effective for a wide range of resource management problems. However, they can only support a limited number of parameters for optimization. In a cloud environment, many parameters need to be optimized simultaneously, intelligent algorithms should find the optimal solution.

Intelligent algorithms, on the other hand, are based on mimicking natural body algorithms, mainly ant colony algorithms, simulated annealing methods, particle swarm optimization, and genetic algorithms, which are capable of optimizing multiple parameters simultaneously. Reference [5] aims to minimize three conflicting objectives, i.e., completion time, resource utilization, and execution cost. For this purpose, a multi-objective optimization problem is proposed and then a composite discrete artificial bee colony technique based on epsilon fuzzy dominance is used to derive the Pareto optimal solution; a method based on multiple swarm GA is introduced in [6]. The method uses multiple swarm genetic algorithms to solve the load balancing problem, and the method achieves good results in terms of improving completion time, cost and load balancing.

However, similar to traditional methods, intelligent scheduling methods can lead to long optimization time and inefficient scheduling. In recent years, with the booming field of deep learning, cloud resource management through neural networks has become a major research hotspot. These methods use models such as neural networks to design scheduling strategies to avoid imbalanced resource allocation.

3 DRL Algorithm for Scheduling in Cloud Computing

Resource scheduling problems are NP-complete or NP-hard problems. Most of the deep learning-based resource scheduling algorithms are based on supervised learning. These works use the idea of recent workload to predict the current/future workload. Their main goal is to minimize the number of servers or virtual machines to save energy and reduce costs. After supervised learning, reinforcement learning is the second most prevalent approach to exploration. Most existing algorithms require hand-crafted rules that are difficult to adapt to complex and dynamic systems. In recent years, researchers have found that reinforcement learning (RL) ideas are very suitable for resource management. Reinforcement learning is a special machine learning method that can be trained without prior knowledge and is well suited for applications in cloud environments where labeled data is difficult to obtain (Fig. 1).

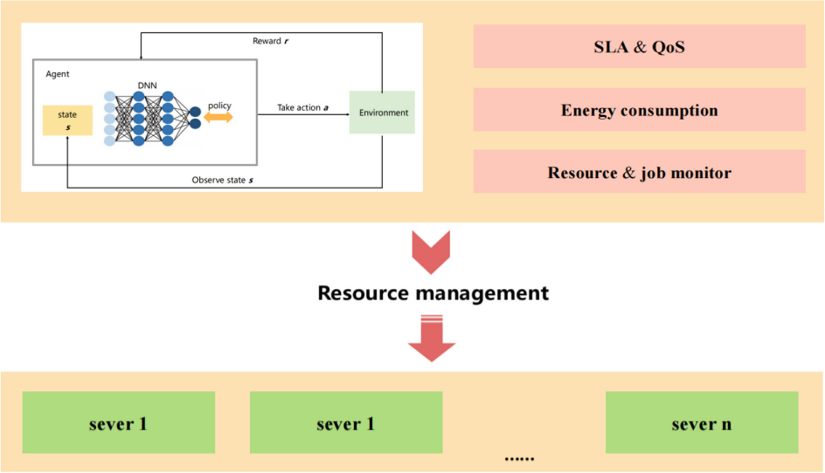

Fig. 1. Framework of resource management using deep reinforcement learning.

DeepRM [7] is the first DRL-based resource scheduler with a policy gradient reinforcement learning algorithm designed to schedule hierarchical jobs to heterogeneous resources and applied to a SoC chip scheduling simulator. However, resource management aims at minimizing the completion time of individual jobs, which can only correspond to a simple scenario. Later DeepRM_Plus [8], based on DeepRM, instead uses a network structure with six layers of convolutional neural networks to describe a decision mapper, which is based on the great success of DNNs for image processing. In DeepRM_Plus, data center clusters, waiting queues and to-do queues constitute the state of the environment, which uses hash codes as input state and reduces the size of the state space. The follow-up work of DeepRM Decima [9] in order to solve the scheduling problem of multiple Directed Acyclic Graph (DAG) tasks when running on multiple Executor. It uses a scalable GNN to represent the scheduling policy and can handle DAG-like tasks of arbitrarily large shape and size. However, the limitation is that it mainly provides performance for clustered applications on similar resource scheduling targets. DeepSoCS [10] extends the Decima architecture to schedule SoC jobs in DS3 by applying heuristic algorithms to map tasks to available resources. However, it has limited performance at faster job injection rates.

In recent years, DRL has shown superior performance in the current research on cloud computing resource management. QEEC [11] is a Q-learning based task scheduling framework for energy-efficient cloud computing, it uses Q-value tables to represent decision makers for actions. According to the operating state and task characteristics of the cluster environment, [3] uses deep Q learning to solve the CPU-GPU heterogeneous computing scheduling problem. RLPAS [12] introduced an algorithm based on the SARSA model, which was used for resource scheduling in cloud environments. RLPAS uses the idea of parallel multi-agent algorithm to learn from the environment and allocate resources to different tasks, thereby improving resource utilization, shortening response time and improving throughput.

At the core of RL is a policy mapper for cloud scheduling. In cloud scheduling using RL, the policy mapper is usually represented by a deep neural network or Q-table. Table 1 summarizes some of the applications of RL in cloud computing resource scheduling, including the resources scheduled, the task types, the RL model tuning used, and so on. It can be seen that the cloud resource scheduling environment is complex, heterogeneous and dynamic. In the use of resources, virtual machines, CPU, memory and other resources are used. DQN and Q-learning become the first choice of most researchers in algorithm selection.

4 Experiment and Evaluation

Experimentation and evaluation are important aspects to verify the effectiveness of algorithms. This section summarizes the direction of the selection of experimental datasets, experimental platforms and evaluation metrics for resource management algorithms based on reinforcement learning by categorizing and summarizing.

4.1 Experimental Datasets

Common datasets in the papers include GoogleCluster, Alibaba Cluster Data, Google-TaskEvents and Azure's scheduling dataset. The GoogleCluster dataset contains data on resource requirements and availability of tasks and virtual machines. The GoogleCluster dataset contains data about the resource requirements and availability of tasks and

Table 1. Summary of resource scheduling algorithms based on deep reinforcement learning.

Algorithm	Resource	Task type	DRL Model
W-FCFS [13]	VMs	Burst workload management	Q-learning
TSMRL [14]	VMs	Scheduling customers requests in dynamic cloud environments	NFQ-based
CDDQLS [15]	VMs	Task Scheduling in cloud computing environments	DQN
DeepRM2 [16]	CPU, Memory	Resource scheduling	DQN
SchedRL [17]	VMs, CPU, Mermory	VM scheduling in a Non Uniform Memory Access architecture (NUMA)	DQN
DRL-LSTM [18]	VMs	Automated task scheduling	Q-learning

virtual machines. Google Cluster consists of many machines connected by high-speed networks. The dataset includes 670,000 logging traces, which record about 40 million task events over 30 days for over 12,000 machines. Google traces contains production workload scheduling requests for 29 days. Alibaba Cluster Data contains production traces for 4k machines over 8 days. Both contain the CPU/memory numbers used by each workload at a granularity of 5 min, as well as scheduling details such as priority, class, and raw resource requests.

4.2 Experimental Platform

The right simulation platform has a very important impact on the effective verification of algorithms. The more commonly used cloud computing platform in the papers is CloudSim, in addition to PlanetLab, RapidMiner Studio, Pegasus Toolkit, and Xen hypervisor, which are also partially used.

CloudSim is an open-source framework, which is used to simulate cloud computing infrastructure and services. It is developed by the University of Melbourne, and is written entirely in Java. It is used for modelling and simulating a cloud computing environment. It can facilitate several cloud related processes such as data center definition, resource definition and scheduling method simulation. It also has several predefined examples and scenarios that can be modified based on algorithms. The PlanetLab project allows researchers to experiment with new services in real-world conditions and at large scale.

4.3 Experimental Metrics

Typically, energy consumption, time cost and load balancing are the main bases for evaluating the performance of scheduling algorithms in cloud computing. Resource scheduling based on intelligent algorithms can be measured in terms of cost, time, utilization, energy awareness, completion time, scalability, workload management, and profit. Specifically, [19] uses a combination of utilization metrics such as CPU utilization, average CPU utilization, RAM utilization, and average RAM utilization to evaluate the system performance, and [13] evaluates the scheduling strategy by calculating the standard deviation of the average resource utilization. in addition to this, [20] evaluates the effectiveness of the strategy by convergence speed. [14] evaluates the average load by load performance of the system. The SLA specifies the QoS content such as response time, reliability, and security of cloud services, so it can be said that SLA is a very important criterion for resource scheduling.

5 Research Challenges

The dynamic and uncertain characteristics of the cloud environment make the resource scheduling very complicated. Due to its characteristics of active learning and adaptive strategy, deep reinforcement learning is more and more applied in the design of cloud resource scheduling and management algorithms. However, the resource scheduling algorithm should also consider the following issues.

SLA violation. When designing the algorithm, the workload and energy efficiency balance should be fully considered to meet the requirements of SLA and QoS as much as possible.

Virtual machine management. Virtualization brings challenges to resource management, so the migration and placement of virtual machines and the creation of instances have a significant impact on algorithm performance.

Scalability. Resource can be scaled to manage demand extensively. Algorithms to meet different resource requirements as much as possible.

Fault tolerance. When one server fails, the algorithm should be able to shift the workload to another server to ensure the user's mission needs.

Complexity. The resource scheduling algorithm needs to optimize multiple objectives at the same time, so the execution time and computing power should be reduced as much as possible while meeting the needs of customers.

Energy consumption. Energy-efficient algorithms not only save costs, but also help data center communication systems reduce carbon emissions and protect the environment.

6 Conclusion

Next generation aircrafts will use airborne cloud computing system as an efficient information processing platform with large amounts of hardware resources and software services, making applications development and deployment much more convenient. Due to the characteristics of dynamic autonomous decision-making, deep reinforcement learning has become a promising resource scheduling algorithm to improve the utilization of resources, which can greatly improve the computing performance and shorten the design-developing-integration-update time cycle. This paper analyzes the requirements of airborne cloud computing systems, and summarizes some critical problems and challenges in the design of DRL-based resource management algorithm. It will provide some technique supports for the airborne cloud computing system.

References

1. Jianchun, X., Zhonghua, W., Yahui, L.: The distributed computing framework research for avionics cloud[C]. In: IEEE International Conference on Networking and Network Applications, pp. 390–393 (2018)
2. Qin, Y., Wang, H., Yi, S., et al.: Virtual machine placement based on multi-objective reinforcement learning[J]. Appl. Intell. **50**(8), 2370–2383 (2020)
3. Dasjupta, P., Leblanc, R.J., Apple, W.F.: The clouds distributed operating system: Functional description, implementation details and related work[C]. In: The 8th International Conference on Distributed, pp. 2–3 (1988)
4. Li, L.: An optimistic differentiated service job scheduling system for cloud computing service users and providers[C] 2009. In: Third International Conference on Multimedia and Ubiquitous Engineering, pp. 295–299 (2009)
5. Gomathi, B., Krishnasamy, K., Bslaji, B.S.: Epsilon-fuzzy dominance sort-based composite discrete artificial bee colony optimisation for multi-objective cloud task scheduling problem[J]. Int. J. Bus. Intell. Data Min. **13**(1–3), 247–266 (2018)
6. Wang, B., Li, J.: Load balancing task scheduling based on multi-population genetic algorithm in cloud computing[C]. In: 2016 35th Chinese Control Conference (CCC), pp. 5261–5266 (2016)
7. Mao, H., Alizadeh, M., Menache, I. et al.: Resource management with deep reinforcement learning[C]. In: Proceedings of the 15th ACM Workshop on Hot Topics in Networks, pp. 50–56 (2016)
8. Zhang, Y., Yao, J., Guan, H.: Intelligent cloud resource management with deep reinforcement learning[J]. IEEE Cloud Comput. **4**(6), 60–69 (2017)
9. Mao, H., Schwarzkopf, M., Venkatakrishnan, S.B., et al.: Learning scheduling algorithms for data processing clusters[G]. In: Proceedings of the ACM Special Interest Group on Data Communication, pp. 270–288 (2019)
10. Sung, T.T., Ha, J., Kim, J. et al.: Deepsocs: a neural scheduler for heterogeneous system-on-chip (soc) resource scheduling[J]. Electronics **9**(6), 936 (2020)
11. Ding, D., Fan, X., Zhao, Y., et al.: Q-learning based dynamic task scheduling for energy-efficient cloud computing[J]. Futur. Gener. Comput. Syst. **108**, 361–371 (2020)
12. Bibal Benifa, J., Dejey, D.: Rlpas: Reinforcement learning-based proactive auto-scaler for resource provisioning in cloud environment[J]. Mob. Netw. Appl. **24**(4), 1348–1363 (2019)
13. Balasubramanian, V., Aloqaily, M., Tunde-onadele, O., et al.: Reinforcing cloud environments via index policy for bursty workloads[C]. In: NOMS 2020–2020 IEEE/IFIP Network Operations and Management Symposium, pp. 1–7 (2020)
14. Balla, H.A., Sheng, C.G., Jing, W.: Reliability-aware: task scheduling in cloud computing using multi-agent reinforcement learning algorithm and neural fitted Q.[J]. Int. Arab J. Inf. Technol. **18**(1), 36–47 (2021)
15. Swarup, S., Shakshuki, E.M., Yasar, A.: Task scheduling in cloud using deep reinforcement learning[J]. Procedia Comput. Sci. **184**, 42–51 (2021)
16. Ye, Y., Ren, X., Wang, J. et al.: A new approach for resource scheduling with deep reinforcement learning[J] (2018). arXiv:1806.08122
17. Sheng, J., Hu, Y., Zhow, W. et al.: Learning to schedule multi-NUMA virtual machines via reinforcement learning[J]. Pattern Recognit. **121**, 108254 (2022)
18. Rjoub, G., Bentahar, J., Abdel Wahab, O., et al.: Deep and reinforcement learning for automated task scheduling in large-scale cloud computing systems[J]. Concurr. Comput.: Pract. Exp. **33**(23), e5919 (2021)

19. Yu, Z., Machado, P., Zahid, A., et al.: Energy and performance trade-off optimization in heterogeneous computing via reinforcement learning[J]. Electronics **9**(11), 1812 (2020)
20. Chou, Q., Fan, W., Zhang, J.: A reinforcement learning model for virtual machines consolidation in cloud data center [C]. In: 2021 6th international conference on automation, control and robotics engineering (CACRE), pp. 16–21 (2021)