International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 01 | Jan - 2025 SJIF Rating: 8.448 ISSN: 2582-3930

Optimized Cloud-Edge Task Allocation and Load Balancing Strategy

Saharsha B P
Dept.of.CS&E PESITM
Shimogga,India
saharsha160@gmail.com

Sumanth S V
Dept.of.CS&E PESITM
Shimogga,India
sumanthsv19@gmail.com

Sumkha Gowda *Dept.of.CS&E PESITM*
Shimogga,India
Sumukagowda177@gmail.com

Supreeth E *Dept.of.CS&E PESITM*
Shimogga,India
supreeth59@gmail.com

Abstract- The rapid evolution of distributed systems has necessitated innovative approaches for managing task deployment and achieving load balancing in cloudedge environments. This paper addresses the prevalent issue of data overload at cloud computing centers and edge nodes by introducing a Joint Cloud-Edge Task Distribution (JCETD) model. Utilizing pruning algorithms and deep reinforcement learning, the proposed approach ensures efficient resource utilization and optimized task allocation. Key contributions include minimizing computational complexity and enhancing system performance through strategic pruning and adaptive learning techniques.

Keywords- Distributed systems, cloud-edge environments, task deployment, load balancing, data congestion, cloud computing centers, edge nodes, JCETD model, pruning algorithms, deep reinforcement learning, resource optimization, task allocation, computational overhead, system efficiency, adaptive learning methods.

I. INTRODUCTION

The convergence of cloud and edge computing has become a groundbreaking approach in distributed computing, uniting centralized and decentralized data processing. This cloud-edge collaboration combines the extensive computational capabilities of cloud centers with the close proximity of edge nodes to users, facilitating processing frequently hinder system real-time responsiveness and efficient resource management. Nevertheless, issues like resource overload and irregular task performance.

If a user's job request is beyond the existing physical resources, or these are fully allocated and not released on time, the distribution process will be disrupted and end in failure. If the resource requirements of a task hit the critical threshold of any host, then its execution is appreciably slowed. Such delaying is impeding the timely delivery of the next tasks, thereby leading to inconsistencies in the "cloud-edge" data center. When such inconsistencies occur, the system cannot provide real-time responsiveness to the end users, denying the benefits of cloud-edge integration. For this purpose, tasks have to be managed by a cloud-edge joint framework.

System reward refers to the feedback value that the environment provides after a certain action is taken in a certain system state, which reflects the degree of positive or negative consequences of that action in that state. In this paper, we define the system reward according to the extent of load balancing achieved after each deployment action. The lower the degree of load balancing, the fairer the load distribution is within the "cloud-edge" datacenter. The system reward, namely, reward, can be defined as follows.

Improving the deployment efficiency of the integrated "cloud-edge" data center also simplifies the deep reinforcement learning algorithm by reducing its state space. At the start of modeling the system environment with this algorithm, we define the system's

state space. As the agent observes and interacts with the environment, the system's state reflects the available resources in the cloud-edge setup.

The good energy efficiency and work balance distribution model. The key to achieve the above goal is to provide the necessary work with the right resources. By putting the required tasks at the user level, the average response time of the tasks can be reduced, and the calculated results can be returned to the users in a shorter time, thus the quality of services provided to users is improved. At the system level, not only the cloud edge integration is improved, but also the long-term stability of the cloud edge integration is guaranteed. However, at present, the research on the implementation of cloud management strategy has not been completed, there is no guarantee that the activities will always be allocated to quality capital. The resource management and distribution strategy based on pruning algorithm [2] and deep learning is proposed. First, put the existing physical equipment between the air centers and the center of the joint edge. With the concept of pruning algorithm, the physical hosts without an exit path are pruned to be brought to the non-host process, which is the initial state of the deep learning algorithm. Then, simulate the distribution process based on the deep learning algorithm process. Through the continuous exploration of the environment, the efficient distribution and balance of "cloud" cooperation has finally been achieved. The key to achieve the above goal is to distribute the requested work to the physical body of "cloud" computing for computing, so that the center can voluntarily "connect" the "cloud" data and achieve better counting ability. This provides users with the best services to achieve cloud-based integration of the same product. The main contributions of this paper are as follows:

• Based on the joint "cloud-edge" architecture, the deep reinforcement learning algorithm is applied to achieve the efficient deployment and long-term load balancing of the "cloud-edge" system.

• The idea of pruning algorithm is seamlessly connected to the process of the deep reinforcement learning, which not only prunes the unreasonable physical host, but also reduces the state space of deep reinforcement learning algorithm.

•The DDPG algorithm of deep reinforcement learning can be used to continuously and efficiently deploy tasks in the continuous action space.

The rest of this paper is organized as follows. **Section 2** provides a brief overview of related work on task deployment methods in edge and cloud computing environments. **Section 3** introduces and formalizes the problem. **Section 4** outlines the design process for deployment in the integrated "cloud-edge" environment, followed by a detailed description of the algorithm's design and implementation. **Section 5** presents the experimental results, which demonstrate the effectiveness of the proposed algorithm. The paper concludes in **Section 6**.

II. RELATED WORKS

Task distribution and load balancing are key areas of focus in distributed computing research. Existing approaches can be categorized into clustering, reinforcement learning, and load balancing algorithms. Collaborative computation frameworks typically tackle resource constraints by distributing tasks across multiple nodes. For instance, Sahni et al. proposed a model that integrates task planning and coordination to minimize execution time. Schafer et al. introduced a hybrid scheduling system for edge and cloud environments, optimizing resource allocation across both. Li et al. developed a two-stage scheduling strategy that prioritizes edge computing and transfers tasks to the cloud when needed.

Despite these advancements, many methods fall short in addressing system-wide balance and stability. Recent studies have turned to reinforcement learning techniques to overcome these challenges, incorporating dynamic decision-making with task allocation strategies to ensure efficient resource usage and optimal performance. Techniques like clustering, reinforcement learning, and load balancing algorithms help distribute tasks across multiple nodes, preventing overload and reducing execution time. Collaborative computation frameworks such as Apache Spark and TensorFlow utilize these strategies to divide large tasks into smaller subtasks, distributing them across clusters to enhance scalability and speed up computations.

Job scheduling and resource allocation across devices can be optimized using deep learning, which combines incremental learning techniques to solve complex problems. Wang et al. [11] used the DQN model for job scheduling to minimize completion time and costs, while Dong et al. [12] introduced RLTS, a task scheduling algorithm based on deep learning to reduce processing time for cloud servers. Xiong et al. [13] proposed a resource allocation

strategy for IoT edge applications using a Markov decision process and deep learning, though their approach focuses more on system value than stability. Cheng et al. [14] presented a deep reinforcement learning-based resource allocation and task scheduling method that learns from changing environments to reduce service costs for cloud service providers. Tham et al. [15] addressed load balancing and optimization at the edge, solving an optimization problem using gradient descent to reduce application completion time. Jyoti et al. [16] developed a dynamic resource allocation method that assigns tasks to virtual machines based on priority alarms, improving performance and reducing response time. Lee et al. [17] proposed an edge distribution function to solve load balancing issues between edge nodes, showing that task assignment to the least loaded nodes reduces processing time. Ghasemi et al. [18] introduced an RL-based multi-objective virtual machine transfer algorithm for resource recovery and balancing in data centers. Lastly, Tong et al. [19] developed DQTS, a dynamic scheduling set of rules that uses deep Q-learning for better scalability and efficient load balancing compared to other algorithms.

Building on the research mentioned above, this paper successfully integrates the joint "cloud-edge" model with deep reinforcement learning, continuously exploring the environment under this approach. The paper sets up task sets in both the cloud computing center and the edge computing center, ensuring tasks are processed efficiently while minimizing overall response time. Additionally, the integration enhances the computing capacity and load balancing of the joint "cloud-edge" data center.

.

III. PROPOSED PROBLEM AND ITS FORMALIZATION

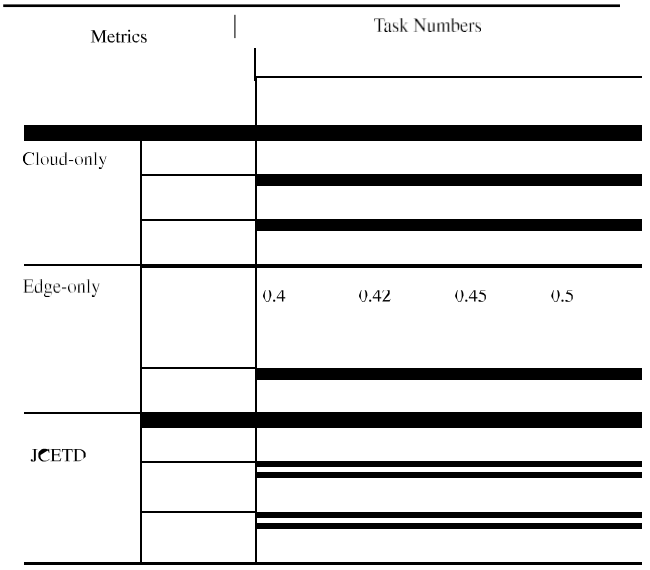
A. PROBLEM STATEMENT

The system handles many resource-intensive tasks that need to be processed within the cloud computing environment. Tasks are randomly deployed across hosts in either the cloud or edge. When the resources requested by users exceed the remaining capacity of the physical host, it can result in a decline in the data center's computing and service capabilities. This may prevent the system from delivering real-time results to users and disrupt the current load balancing state. Furthermore, if computing resources are not promptly released, tasks placed within the data center could lead to data loss and an inability to provide effective services to users. When dealing with large-scale computational tasks, different deployment modes and resource allocation strategies affect efficiency and load balancing. In situations where the data center has limited computing resources, the optimal deployment strategy is the joint cloud-edge model.

B. FORMALIZATION OF THE PROBLEM

In the cloud-edge computing model, the task deployment problem can be defined as follows: within a specific time period, the system collects **n** task requests, which are independent of each other and do not have dependencies. Both the edge computing center and the cloud computing center need to be used for deployment. Several tasks must be processed within these centers.

TABLE 1. Comparison of Cloud-only, Edge-only, and JCETD on throughput, Makespan, and ART.



C. COMPARISON WITH UNILATERAL COMPUTING

In the fifth set of experimental scenarios, this paper compares the proposed joint "cloud-edge" computing model with edge computing and cloud computing, evaluating metrics such as Makespan, throughput, and ART to highlight the advantages of the joint "cloud-edge" framework from different perspectives. As shown in Table 1, as the number of tasks increases, the performance metrics of the joint "cloud-edge" system consistently outperform those of both cloud and edge computing. This improvement is attributed to the joint deployment strategy, which is guided by deep reinforcement learning. The system continually searches for the most suitable physical host for tasks, whether in the cloud or at the edge. As a result, this

"cloud-edge" integration, using deep reinforcement learning, ensures both high-quality user experiences and effective load balancing within the system. In contrast, single edge computing may struggle to meet the task demands of a large number of users, leading to a decline in service quality and system performance. For single cloud computing, increased user access and data, coupled with long communication delays, can cause load imbalance at the cloud center and delays in returning results. The experimental results demonstrate that the performance of joint "cloud-edge" computing surpasses both cloud and edge computing systems.

IV. CONCLUSION AND FUTURE WORK

This paper proposes a resource management and task deployment strategy called JCETD, based on pruning algorithms and deep reinforcement learning, within the joint "cloud-edge" computing framework. The main concepts, implementation process, and evaluation of this strategy are discussed. Initially, the resource management process applies a pruning algorithm to filter the set of "cloud-edge" hosts, using the attribute values of physical hosts to obtain a non-dominated host set. This set has the potential to efficiently deploy tasks and achieve load balancing within the "cloud-edge" data center to some extent. Subsequently, deep reinforcement learning is integrated into the joint "cloud-edge" computing model, with the host set obtained during the resource preprocessing stage serving as the system state for the learning process. Through continuous exploration and interaction with the environment, the system successfully achieves efficient computing and load balancing within the "cloud-edge" architecture. To evaluate the JCETD algorithm, several simulation experiments were conducted to measure performance metrics, confirming its effectiveness from both the user and system perspectives. The experimental results demonstrate that JCETD not only enables efficient task deployment and computation but also promotes a more balanced overall load within the "cloud-edge" data center.

This study introduces the JCETD model as an innovative solution for task deployment and load balancing in cloud-edge environments. By combining pruning algorithms with deep reinforcement learning, the approach ensures efficient resource management and long-term system stability. Future research will concentrate on refining parameter tuning and exploring additional optimization techniques to further improve the model's performance..

V. REFERENCES

[1] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, ‘‘Edge computing: Vision and challenges,’’ *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[2] Y. Chen, J. Huang, C. Lin, and X. Shen, ‘‘Multi-objective service compo- sition with QoS dependencies,’’ *IEEE Trans. Cloud Comput.*, vol. 7, no. 2, pp. 537–552, Apr. 2019.

[3] X. Xu, Q. Liu, Y. Luo, K. Peng, X. Zhang, S. Meng, and L. Qi, ‘‘A com- putation offloading method over big data for IoT-enabled cloud-edge com- puting,’’ *Future Gener. Comput. Syst.*, vol. 95, pp. 522–533, Jun. 2019.

[4] H. Tang, C. Li, J. Bai, J. Tang, and Y. Luo, ‘‘Dynamic resource alloca- tion strategy for latency-critical and computation-intensive applications in cloud–edge environment,’’ *Comput. Commun.*, vol. 134, pp. 70–82, Jan. 2019.

[5] Y. Huang, Y. Zhu, X. Fan, X. Ma, F. Wang, J. Liu, Z. Wang, and Y. Cui, ‘‘Task scheduling with optimized transmission time in collaborative cloud- edge learning,’’ in *Proc. 27th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2018, pp. 1–9.

[6] Y. Sahni, J. Cao, and L. Yang, ‘‘Data-aware task allocation for achieving low latency in collaborative edge computing,’’ *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3512–3524, Apr. 2019.

[7] D. Schafer, J. Edinger, J. Eckrich, M. Breitbach, and C. Becker, ‘‘Hybrid task scheduling for mobile devices in edge and cloud environments,’’ in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2018, pp. 669–674.

[8] C. Li, C. Wang, and Y. Luo, ‘‘An efficient scheduling optimization strategy for improving consistency maintenance in edge cloud environment,’’ *J. Supercomput.*, vol. 76, pp. 6941–6968, Sep. 2020.

[9] Y. Xie, Y. Zhu, Y. Wang, Y. Cheng, R. Xu, A. S. Sani, D. Yuan, and Y. Yang, ‘‘A novel directional and non-local-convergent particle swarm optimization based workflow scheduling in cloud–edge environment,’’ *Future Gener. Comput. Syst.*, vol. 97, pp. 361–378, Aug. 2019.

[10] A. I. Orhean, F. Pop, and I. Raicu, ‘‘New scheduling approach using reinforcement learning for heterogeneous distributed systems,’’ *J. Parallel Distrib. Comput.*, vol. 117, pp. 292–302, Ju