RESEARCH ARTICLE OPEN ACCESS

Action Based Load Balancing Scheduling Algorithm in Cloud Environment

Arabinda Pradhan[1], Satyaranjan Mishra[2], Prasanta Kumar Bal[3], Sukant Kishoro Bisoy[4], Chinmay Parida[5]
[1]*Associate Professor, Dept. of MCA, Raajdhani Engineering College, Bhubaneswar, Odisha,*
[2]*Associate Professor, Dept. of MCA, Raajdhani Engineering College, Bhubaneswar, Odisha,* [3]*Professor, Dept. of CSE, GITA, Bhubaneswar, Odisha,*
[4]*Professor, Dept. of CSE, C. V. Raman Global University, Bhubaneswar, Odisha,*
[5]*Assistant Professor, Dept. of MCA, Raajdhani Engineering College, Bhubaneswar, Odisha,*

ABSTRACT
Cloud computing offers its users a variety of services for a minimal cost and in a shorter period of time. As a result, a lot of people submit requests, but the cloud datacenter does not have enough servers. This problem demonstrates a cloud datacenter load balancing difficulty. This work proposes an Action based Load Balancing Scheduling Algorithm (ALBSA) to address this problem. This ALBSA is compared to various meta-heuristic task scheduling algorithms on the Python platform in order to demonstrate its superior performance. The results demonstrate that our suggested scheduling technique reduces the makespan time by 4.09% to 14.22% when the number of tasks is progressively increased and by 9.23% to 14.99% when the number of virtual machines (VMs) is increased. Additionally, when the number of iterations grows, the accuracy increases from 5.82% to 7.14%.
Keywords: Cloud computing, load balancing, server

Date of Submission: 20-05-2025 Date of acceptance: 30-05-2025

I. INTRODUCTION

Users can request a range of services via cloud computing, a web-based technology. Using virtualization techniques, a number of servers in the cloud datacenter that offers these services are divided into several virtual machines (VMs) (Pradhan et al., 2019; Pradhan et al., 2025). Because cloud computing is dynamic, there are more incoming demands than servers or virtual machines can accommodate. Thus, a load balancing issue occurs. Numerous scheduling algorithms, including heuristic optimization technology, existing reinforcement learning (RL), and the Deep Q-Network algorithm, have been presented to address this issue; however, they have yet to demonstrate great efficiency (Pradhan et al., 2022a, Pradhan et al., 2022b).The outcomes of the RL and DQN algorithms are inaccurate in the case of a dynamic environment since no preset action structure is defined. To solve the problem with the existing DQN methodology, we proposed an effective scheduling method in this study termed the Action based Load Balancing Scheduling Algorithm (ALBSA). In this method, we employ a smart step to reduce the makespan time for each activity. The basic cloud computing concept is shown in Fig. 1. It contains two layers such as user layer and datacenter layer. User layer contains task which is present in task queue. Datacenter layer contains number of servers and VMs. The scheduling algorithm is used to find the suitable VM from datacenter that handles the incoming task to optimize the result.

The main contribution of this study is summarized as: 1) using an effective task scheduling algorithm based on the DQN technique, an agent can select the best course of action to maximize the reward function, such as minimizing the makespan time. 2) The recommended approach interacts with the environment to yield the optimal result, managing the dynamic and continuous nature of state space and demonstrating high learning efficiency. The outcome of the experiment demonstrates the accuracy and reward rate of our suggested method.

The remainder of the paper is organized as follows. Existing work is described in Section 2. The ALBSA technique is described in Section 3. The Results and Discussions are explained in Section 4. The paper is summarized in Section 5.

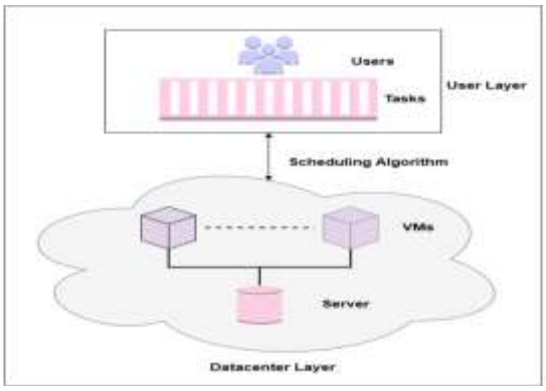

Fig.1. Scheduling technique in cloud computing model

II. EXISTING WORK

The three primary components of task scheduling methodology—heuristic, meta-heuristic, and hybrid approaches—are an efficient means of addressing the load balancing issue (Pradhan et al., 2022c). After the virtual machines (VMs) have been assigned to the datacenter, task scheduling is responsible for allocating them evenly and making sure that none of them are overloaded or underloaded. It is also responsible for optimizing several load balancing measures, such as makespan time, execution time, reaction time, resource use, energy consumption, etc. (Pradhan et al., 2022d). However, we just consider makespan time in this paper.Tennakoon et al. (2023) proposed a double Q-learning approach (DQA) to distribute the workload across the virtual machines in order to enhance QoS. It is also used to optimize the distribution of resources to boost customer satisfaction. Based on an adaptive-dynamic synchronous parallel distributed computing model (A-DSP), the adaptive fast reassignment (AdaptFR) technique was created by Li et al. (2020) in order to balance the load and enhance system performance. In order to improve response time and maintain load balance, (Ran et al., 2019) developed the deep deterministic policy gradients (DDPG) task scheduling technique, which is based on deep reinforcement learning (DRL). A task scheduling method based on deep reinforcement learning algorithms was introduced in order to optimize makespan and resource usage (Che et al., 2020). A deep reinforcement learning architecture (RLTS)-based task scheduling algorithm was proposed in (Dong et al., 2020) to manage the high dimensional environment and complexity by reducing the task execution time. To reduce the makespan time and increase the incentive, a DRL environment for work shop scheduling was proposed (Tassel et al., 2021). The authors of (Cheng et al. 2018) proposed a novel Deep Reinforcement Learning (DRL)-based Resource Provisioning (RP) and Task Scheduling (TS) system named DRL-Cloud to manage the large number of user requests received every day and lower energy expenditures. The A2C approach, which is utilized to decrease makespan time, was proposed by Pradhan et al. in 2022e.

III. ALBSA APPROACH

The load balancing algorithm's fundamental parameters, including makespan time, are displayed in this section. Our suggested technique, which is based on the Deep Q Network (DQN) algorithm, aims to minimize this parameter by taking the proper actions in the cloud environment and attempting to achieve an optimal outcome that is near the desired value.

3.1 Objective Function

The reduction of makespan time is our primary goal. When more activities are completed successfully at the datacenter, makespan time (MT) may decrease. MT depends on how long each task takes to complete, as shown in Eq. (1). Where TE is the total execution time and AL is the allocation time.

$MT=TE+AL$ … (1)

3.2 Reward Function

The basic work of this section is that an agent should be able to efficiently manage a large number of incoming tasks, assigning each task to a VM with a high transfer rate for each i number of iterations. It enables an agent to achieve the maximum reward in the environment and has a direct impact on makespan time minimization.

To reduce the makespan time, we need to increase the task transfer rate (TR), which reduces the allocation time. The virtual machine load (VML) in proportion to allocation time determines the task transfer rate, which is shown in Eq. (2). Using Eq. (3), the selected action at iteration (a_i) can then be calculated.

The agent is motivated to select a state with a larger reward function in order to enhance system performance. The agent advances to the states with the largest reward function after learning. The agent first learns by switching between random states. In our approach, the task transfer rate is increased by minimizing makespan time, which defines the reward function. The reward is represented by Eq. (4).

$$\mathrm{TR}=\frac{VML}{AL} \quad (2)$$

$$\mathrm{a_i}=\max(\mathrm{TR}) \quad (3)$$

$$\mathrm{r_i}=\min\{\mathrm{MT}\}(4)$$

3.3 Proposed Algorithm

In this algorithm the state of VM of each iteration is representing as s_i, random weights of both prediction Q-network and target Q-network is representing as Θ and Θ respectively, where $\Theta' = \Theta$.

```
Initialize capacity, reply memory, prediction and target Q-network. 1. **Start**
2. For each iteration��do
3. Initialize the stateand load of VM
For each task in task-queue do
if probability ∊ then
choose a random action
Otherwise
Selectthe action based on Eq. (3)
End if
Applying the selected action, calculate reward
Move to the new state
Store transition in reply memory
Set the target function
Perform a gradient descent to define error value
Every iteration, update target Q-network
```

```
End For
4. End For
5. Return reward
6. Stop
```

IV. RESULT AND DISCUSSION

This section first describes the simulation environment and then assesses the performance of our proposed ALBSA algorithm and compares it to an existing DQN approach.

Our tests are conducted in Google Colab using TensorFlow 1.4.0 and Python 3.9. In our simulation, each task is independent and continuous. Ten thousand tasks with durations ranging from 500 to 1000 are chosen. These tasks are performed by ten virtual computers. In our proposed model, we used multi-layered neural networks, each layer including ten neurons. The replay memory has been set to C = 1000. The learning rate "λ" is set to 0.1, the discount factor "γ" is set to 0.9, and the ∊ is lowered from 0.9 to 0.02 for each learning iteration. Based on the data provided, we execute our simulation to evaluate the performance of our proposed algorithm.

The objective of our proposed approach is to reduce the makespan time. The simulation's results are displayed in Figures 2 through 4. Figures 2 and 3 compare the makespan time of our proposed ALBSA approach with existing DQN method. 100 virtual machines (VMs) are assigned 1000–10,000 jobs in Figure 2. 10 to 100 VMs are handling 10,000 jobs in Figure 3. Tables 1 and 2 display the datasets for Figs. 2 and 3, respectively. According to the experiment, our suggested approach exhibits 4.09% to 14.22% declines when the number of tasks is progressively increased, and 9.23% to 14.99% decreases when the number of virtual machines (VMs) is increased.The accuracy behavior of our suggested method in comparison to an existing DQN method is displayed in Fig. 4. The results clearly demonstrate that as the number of iterations grows, our suggested strategy improves accuracy by 5.82% to 7.14%. Table 3 displays the datasets for Fig. 4.

Table 1: Makespan time of 1000 to 10000 task count

Task No.	ALBSA	DQN
1000	2587.22	2695.34
2000	2634.47	2841.65
3000	2772.42	2924.78
4000	2881.66	3087.4
5000	2935.94	3165.35
6000	3113.07	3347.42
7000	3455.24	3831.57
8000	3746.97	4282.11
9000	4057.36	4561.74
10000	4409.85	5085.33

Table 2:Makespan time of 10 to 100 VMs

VM Count	ALBSA	DQN
10	3477.54	3814.25
20	3394.12	3708.78
30	3235.24	3564.88
40	3175.47	3477.46
50	3011.84	3374.62
60	2877.63	3245.36

70	2704.51	3106.81
80	2532.04	2985.33
90	2378.94	2754.24
100	2214.57	2573.55

Table 3: Accuracy of different scheduling approach

Iteration	ALBSA	DQN
10	0.53	0.50
20	0.55	0.49
30	0.64	0.58
40	0.77	0.72
50	0.83	0.79
60	0.83	0.80
70	0.838	0.79
80	0.845	0.8
90	0.86	0.81
100	0.87	0.81

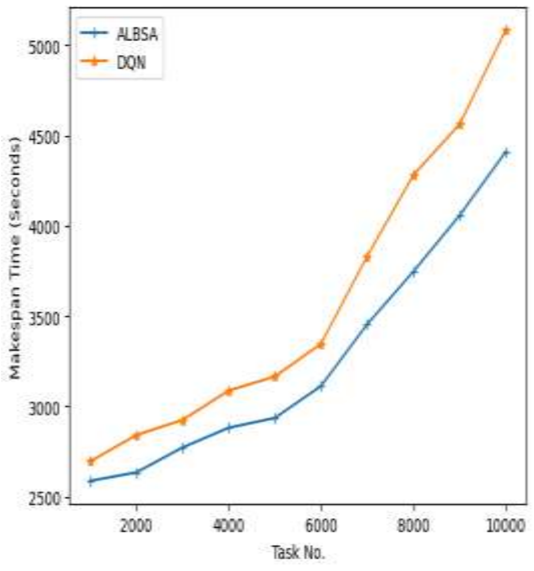

Fig.2. Makespan time under different task counts

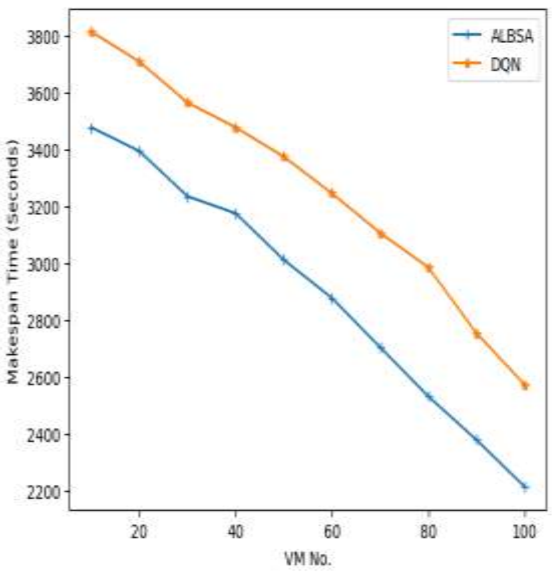

Fig.3. Makespan time under different VMs

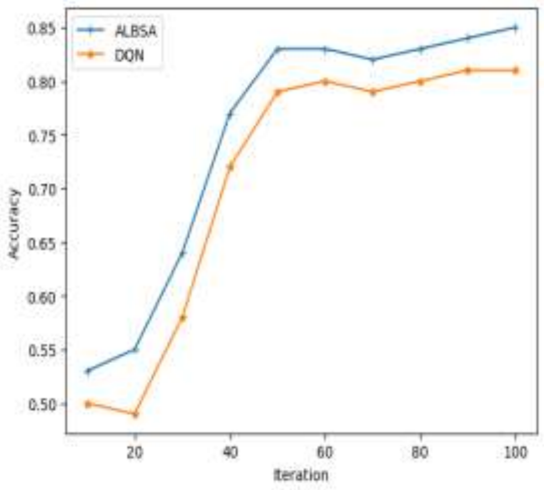

Fig.3. Accuracy of different scheduling algorithm

V. CONCLUSION AND FUTURE WORK

One of the most important issues in cloud computing is load balancing. To address this issue, we proposed the effective action based approach ALBSA. With the help of the DQN technique, we carry out an efficient activity that can reduce makespan time. Our simulation results show that we perform better than an existing DQN method. The results demonstrate that our suggested scheduling technique reduces the makespan time by 4.09% to 14.22% when the number of tasks is progressively increased and by 9.23% to 14.99% when the number of virtual machines (VMs) is increased. Based on the simulation findings, we achieve an accuracy rate of between 5.82% to 7.14% at the end of iteration 100. It illustrates the superiority of our proposed scheduling strategy over the existing DQN method. To improve resource scheduling in the future, we will develop a model based on the DQN algorithm. Using this paradigm, we allocate the incoming job to the relevant virtual machine.

REFERENCES

[1]. Che, H., Bai, Z., Zuo, R. and Li, H. (2020). A Deep Reinforcement Learning Approach to the Optimization of Data Center Task Scheduling. Hindawi Complexity, Wiley, Vol 2020, Article ID 3046769, pp 1-12.

[2]. Cheng, M., Li, J. and Nazarian, S. (2018). DRL-cloud: Deep reinforcement learning-based resource provisioning and task scheduling for cloud service providers. Proceedings of the Asia and South Pacific Design Automation Conference, ASP DAC, 129–134.

[3]. Dong, T., Xue, F., Xiao, C. and Li, J. (2020). Task scheduling based on deep reinforcement learning in a cloud manufacturing environment. Concurrency and Computation: Practice and Experience. Wiley, pp 1-12.

[4]. Li, M., Zhang, J., Wan, J. et al. Distributed machine learning load balancing strategy in cloud computing services. Wireless Netw 26, 5517–5533 (2020).

[5]. Pradhan, A., Bisoy, S. K. and Mallick, P. K. (2019). Load balancing in cloud computing: Survey. Innovation in Electrical Power Engineering, Communication, and Computing Technology: Proceedings of IEPCCT 2019. pp 99-111.

[6]. Pradhan, A., Bisoy, S. K., Kautish, S., Jasser, M. B. and Mohamed, A. W. (2022a). Intelligent Decision Making of Load Balancing Using Deep Reinforcement Learning and Parallel PSO in Cloud Environment. IEEE Access, vol. 10, 76939-76952.

[7]. Pradhan, A. and Bisoy, S. K. (2022b). Intelligent Action Performed Load Balancing Decision Made in Cloud Datacenter Based on Improved DQN Algorithm. 2022 International Conference on Emerging Smart Computing and Informatics (ESCI). pp 1-6.

[8]. Pradhan, A., Bisoy, S. K. and Das, A. (2022c). A survey on PSO based meta-heuristic scheduling mechanism in cloud computing environment. Journal of King Saud University –Computer and Information Sciences, Elsevier. Vol 34, Issue 8, pp 4888-4901.

[9]. Pradhan, A. and Bisoy, S. K. (2022d). A novel load balancing technique for cloud computing platform based on PSO. Journal of King Saud University – Computer and Information Sciences, Elsevier, Vol. 34, no. 7, pp. 3988 – 3995.

[10]. Pradhan, A., Bisoy, S. K. and Sain, M. (2022e). Action-Based Load Balancing Technique in Cloud Network Using Actor-Critic-Swarm Optimization. Wireless Communications and Mobile Computing. Vol. 2022, Issue 1, pp 6456242.

[11]. Pradhan, A., Das, A. and Bisoy, S. K. (2025). Modified parallel PSO algorithm in cloud computing for performance improvement. Cluster Computing. Vol 28, Issue 2, pp 131.

[12]. Ran, L., Shi, X. and Shang, M. (2019). SLAs-aware Online Task Scheduling based on Deep Reinforcement Learning Method in Cloud Environment. 21st International Conference on High Performance Computing and Communications; 17th International Conference on Smart City; 5th International Conference on Data Science and Systems, IEEE, pp 1518-1525.

[13]. Tassel, P., Gebser, M. and Schekotihin, K. (2021). A Reinforcement Learning Environment for Job-Shop Scheduling. arXiv:2104.03760 [cs. LG].

[14]. Tennakoon, D., Chowdhury, M. and Luan, T. H. (2023). Cloud based load balancing using double Q-learning for improved Quality of Service. Wireless Netw 29, 1043–1050.