Journal of Theoretical and Applied Information Technology
30th September 2024. Vol.102. No. 18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

TASK SCHEDULING FOR VIRTUAL MACHINE MIGRATION IN CLOUD COMPUTING USING ADAPTIVE REINFORCEMENT LEARNING

MOHAMMED ZIAUR RAHMAN[1] , ANANDARAJ SHANTHI PICHANDI[2]

[1]Research Scholar and Assistant Professor, School of Computer Science and Engineering, Presidency University, Bangalore, India

[2]Professor and HoD, School of Computer Science and Engineering, Presidency University, Bangalore, India

E-mail: [1]ziyarmn@gmail.com, [2]anandsofttech@gmail.com

ID 55269 Submission	Editorial Screening	Conditional Acceptance	Final Revision Acceptance
05-08-24	12-08-2024	10-09-2024	21-09-2024

ABSTRACT

The number of cloud users and their respective workloads is continuously increasing due to the inherent benefits of Cloud Computing (CC). Due to the rapid increase in the use of cloud services, the energy consumption of cloud data centres is dramatically increasing. An overloaded or underloaded Virtual Machine (VM) leads to enhanced response time and energy consumption due to suboptimal resource utilisation. Adaptive Reinforcement Learning (ARL) approach is proposed in this research based on Task Scheduling (TS) for VM migration in a CC environment to overcome this limitation. The proposed ARL approach signifies the advancement in cloud resource management by incorporating Ant Colony Optimization (ACO) for dynamic VM migration and TS optimization which offers efficiency and adaptability in cloud environments. The ACO is selected for task scheduling because of its distributed optimisation adaptability, capability to enhance resource utilization, and the VM migration performances. The ARL approach performance is evaluated with metrics of throughput, migration time, response time, load, energy consumption and resource availability. The ARL achieves a throughput of 2.33, migration time of 12.64ms, response time of 128.57ms, and energy consumption of 0.45W which is superior when compared to the Selection strategy based on correlation and utilization (SS-CAU).

Keywords: *Adaptive Reinforcement Learning, Ant Colony Optimization, Cloud Computing, Task Scheduling, Virtual Machine Migration*

1. INTRODUCTION

Cloud Computing (CC) technology offers on-demand access to shared computing resources over the Internet. When cloud data centers are overwhelmed by end-users, they effectively handle VMs to maintain cost-effectiveness and ensure Quality of Service (QoS) which is crucial for service providers [1, 2]. The virtual technology allows users to create numerous VMs on a physical server to enhance software deployments and application capabilities [3]. The three service delivery models in CC are Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) [4]. The SaaS enables cloud users to access a set of software and applications, PaaS offers a platform for running and developing applications, and IaaS is a cloud environment backbone that offers the entire infrastructure from cloud environments [5-7]. The cloud-based applications are generally considered by Service-Oriented Architecture (SOA) to enable flexible communication among distinct services in clusters [8]. The VMs optimize the utilization of cloud resources of processing power and storage space. The cloud system has uneven distribution of resources while VMs face challenges in accessing the required resources [9, 10].

The high-scale data center in virtualization technology offers a chance to strongly combine data center in VMs [11]. The dynamic VM technique utilizes real-time VM migration to optimize numerous possible VMs on the host. It reduces the number of less-used hosts, thereby minimizing power consumption and enhancing resource usage of hosts in data centers [12-14]. To achieve optimal and stable resource utilization, tasks are distributed simultaneously between VMs using Task Scheduling (TS) algorithms [15]. Therefore, it is crucial to define distribution to make sure not every task is allocated

to a single VMs, leading to the imbalance or unavailability of other VMs [16]. In order to avoid this, the schedule requires to include aspects of cost, makespan and resources. Thus, variable capacities applications running on VM migrations leads to the hindering of QoS, as a result, enhancing some aspects of failure, time-out and response time [17-19]. Additionally, it enhances VM migration and energy consumption, and therefore, the VM consolidation having lesser QoS and enhancing the energy consumption [20]. The VM is overloaded or underloaded due to the maximum resource utilization which increases the response time and energy consumption. To address this issue, this research proposes an ARL based TS for VM migration in a CC environment. The following are the key contributions of this research:

- The proposed ARL approach signifies the advancement in cloud resource management through incorporating ACO for dynamic VM migration and TS optimization which offers efficiency and adaptability in cloud environments.
- The ACO is selected for task scheduling because of its distributed optimisation adaptability and capability to enhance resource utilization and VM migration performances.

This research paper is organized as follows. The literature works are examined in section 2 and the framework of the proposed methodology is explained in section 3. The results of the proposed methods are shown in Section 4, while the conclusion of this research is given in Section 5.

2. LITERATURE REVIEW

Some of the existing researches related to task scheduling for VM migration in CC are analyzed in this section. The process, advantages and limitations are explained below for all the existing researches.

Sharma et al. [21] presented energy-efficient multi-dimensional VM allocation and migration in cloud data center. This approach introduced a Branch-and-Price-based algorithm for VM allocation and Multi-Dimensional Virtual Machine Migration (MDVMM). The VM allocation algorithm optimized energy consumption by selecting Physical Machines (PM), while the MDVMM minimized the energy consumption, evading the Service Level Agreement (SLA) through VM migrations. The MDVMM offered better energy savings and improved the resource utilization. However, it did not estimate the resource availability as it mainly focused on optimizing VMs based on energy efficiency which also changed the load imbalance issue among resources.

Khan and Santhosh [22] presented a hybrid optimization algorithm that included Cuckoo search and Particle Swarm Optimization (PSO) for VM Migration in cloud computing. VM migration techniques were deployed in cloud computing to ensure an efficient service delivery and minimum energy consumption. The algorithm optimized VM placements by minimizing unnecessary migrations, aiming at minimizing energy consumption, computation time, and migration costs, while maximizing resource utilization. However, hybrid optimization was restricted by dynamic cloud workloads, potentially leading to suboptimal VM placements and resource allocation.

Ma et al. [23] presented an SS-CAU to optimize energy consumption in cloud data centres through virtualization migration and consolidation. Additionally, adaptive dynamic threshold methods were introduced for migration timing determination and correlation-based strategies to select VMs for migration. These techniques enhanced sustainability and cost-efficiency in cloud computing environments. Yet, the developed model only focused on VM consolidation migration and did not address other aspects of the cloud computing systems.

Gupta and Namasudra [24] suggested a technique that included Host Selection Migration Time (HSMT), VM Reallocation Migration Time (VMRMT), and VM Reallocation Bandwidth Usage (VMRBU) for accelerating migration in cloud computing. This approach was divided into two components of master server which handled user requests and the slave server which chose the optimal target server. These techniques enhanced the performance of cloud computing by minimizing the migration time. The VMRMT faced limitations in handling large-scale migrations which in turn led to high response time.

Kruekaew and Kimpan [25] implemented an independent TS approach in CC. This approach utilized multi-objective task scheduling using Artificial Bee Colony (ABC) with Q-learning which was the RL approach. The developed model aided the algorithm in working faster than the MOABCQ approach. This method optimized scheduling and resource utilization, maximized VM throughput, and created load balancing. However, the MOABCQ did not consider energy consumption which led to an increased response time.

Sudheer Mangalampalli et al. [26] suggested a Deep Reinforcement Learning Based Task Scheduling in Cloud Computing (DRLBTSA). Initially, random produced workload was applied for simulation. The Deep Q-learning was applied which is related to reinforcement learning and incorporated into scheduling approach. The DRLBTSA was utilized which consider decisions dynamically based on existing and upcoming tasks. The self-adaptive nature based on its earlier states enhance the performance. However, it required significant computational resources for training which leads to enhanced energy consumption.

Based on the above analysis, the limitations of the models in handling large-scale migrations result in high execution costs. Moreover, they also estimated resource availability which changed the load imbalance issue among resources. It only focused on VM consolidation migration and did not address other aspects of cloud computing systems such as energy consumption which in turn leading to enhanced execution time. To overcome these problems, the ARL is proposed based on TS for VM migration in CC. The ARL approach indicates the advancement in cloud resource management by incorporating ACO for VM migration and TS optimization that offers efficiency and adaptability in cloud environments. The ACO is selected for TS due to its distributed optimization adaptability, capability which enhances the resource utilization and VM migration performances.

3. PROPOSED METHODOLOGY

The ARL approach is proposed in this research based on TS for VM migration in the CC environment. The proposed ARL framework signifies the advancement in cloud resource management through incorporating ACO for dynamic VM migration and TS optimization which offers efficiency and adaptability in cloud environments. When a user submits a request to the system, the task is transferred to the data broker. The proposed ARL effectively submits the tasks to its suitable VM through the user. The user creates their request through the internet and these requests are kept in VMs. Figure 1 presents the framework of the proposed methodology.

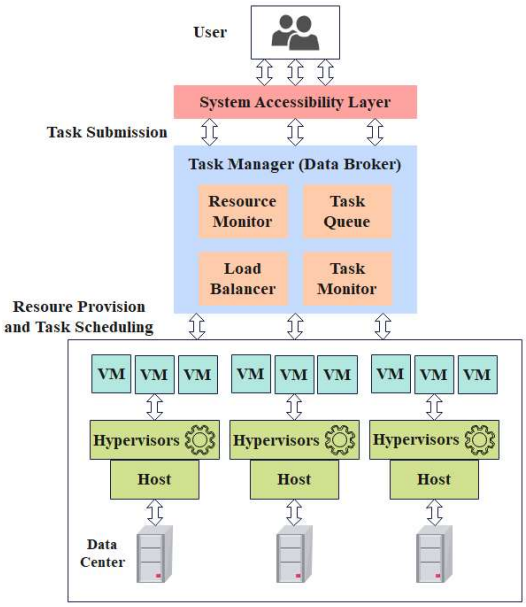

Figure 1: The framework of the proposed method

3.1 System Model

In the cloud platform, each PM has different quantities of VMs and a group of everywhere and n is PMs. The set of VMs in a datacenter is signified as $VM = \{VM_1, VM_2, \ldots, VM_m\}$ where, m is a few VMs. The tasks are scheduled for every VM in each PM to achieve load balancing in the cloud network. The tasks are scheduled to a particular PM to create overload in the unbalanced cloud network.

3.2 Task Scheduling using Adaptive Reinforcement Learning

Reinforcement Learning (RL) utilizes agents to make decisions according to the given inputs to the system. Normally, every Machine Learning (ML) model contains various stages. Hence, the agent takes certain actions according to the input that produces rewards in terms of bad or good. These rewards are helpful for decisions to be considered by the algorithm in the flowing states. The RL is according to this reward and agents try to take actions in the upcoming state according to the rewards produced in the present state. This adaptive nature according to its past states is the main advantage of the RL approach. RL is used for TS and in that, the agent learns from incoming user request sequences in which the agent is trained using previous the user's tasks or requests within the cloud. Primarily, user request is prioritized and given to the agent which carries out decision scheduling based on cloud environment situations. The decision is an output of the user requests or executed tasks that is an agent reward. If the reward score is bad and the

agent enhances its decision through updating the model constraints. If reward values are good, then they are kept in the present state and used for the following process whether it is made by agent in the next state. In RL, the Q-learning is a powerful approach, and it does not require any data from the present scheme. It makes decisions according to previous actions kept as Q-function as a pair through dual conditions designed as $q(S, A)$ in the equation (1).

$$q(S^t, A^t) \leftarrow q(S^t, A^t) + \sigma \times [re^t + \mathcal{F} \times maximum^a q(S^{t+1}, A) - q(S^t, A^t)] \quad (1)$$

Where, σ is learning rate in the range of $[0, 1]$, re^t is a reward for taking actions, $\mathcal{F}$ is a discount factor in the range of $[0, 1]$. In traditional Q-learning, every q score are kept in the q-table' however, scheduling in CC is a problem in these Q-learning models. It is complex for optimal and adaptive decisions in traditional models as the actions and states are higher for the TS approach. Thus, Ant Colony Optimization (ACO) is integrated with RL which helps the schedule problems in CC. The major aim of applying the scheduling model is to create a scheduler where no previous data is provided to the agent and algorithms are required to decide if actual data is provided as input. It contains various states such as action space, state space and reward function which are described in the below sections.

Action space: In the action space, n VMs are considered with the entire incoming requirements primarily given into the task manager. After estimating the priority of every task, the scheduler uses ACO that makes decisions and transmits tasks to the execution queue through the task priorities. Then, decision-making and task allocation to the execution queue based on their priorities is carried out. The action space is defined in equation (2).

$$A = [VM_1, VM_2, VM_3, \dots, VM_n] \quad (2)$$

State space: It contains the state of tasks at a particular time and the state of VM at the same period when the task reaches. Considering that the task t reaches at time T denoted by T_t, the task state is defined in equation (3).

$$S_{Tt} = S_t \cup S_{Tt}^{VM} \quad (3)$$

Where, S_t is a state of task t at time T and S_{Tt}^{VM} is the state of VM if a task t comes under a VM at time T as in equation (4).

$$S_{Tt} = \left[t_k^l, t^{prio}, et_{t_k}, ft^{t_k}, m^k, e_{VM_n}^{con}, SLAV\right] \quad (4)$$

Where, t_k^l, t^{prio}, et_{t_k}, ft^{t_k} and m^k are a length, priority, execution time, finish time and makespan of k tasks. The $e_{VM_n}^{con}$ is the energy consumption of n VMs and $SLAV$ is an SLA Violation.

Reward Function: The reward function is used to discover optimum mapping among cloud resources, and varied tasks through the assistance of TS to enhance substantial QoS constraints of SLAV, time and energy consumption. Hence, it should be minimized as shown in the equation. (5),

$$re = min\left(m^k, e_{VM_n}^{con}, SLAV\right) \quad (5)$$

3.2.1 ACO Algorithm

The ACO is a type of heuristic algorithm that is comparable to swarm algorithms which try to discover the best optimal solutions in less response time. The ACO stimulates the real behaviour of ants when it forages and ants try to discover adjacent food sources and leave the nest randomly. In ACO, each ant is initialized randomly, and the optimal solution is searched. At each iteration, every ant moves to complete service configuration and creates its solution according to QoS. To simulate the ACO algorithm, every ant carries out its function, as given in equation (6).

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{s \subset S_j} [\tau_{is}(t)]^\alpha [\eta_{is}(t)]^\beta} & if\ j \in S_j \\ 0 & otherwise \end{cases} \quad (6)$$

Where, τ_{ij} is a heuristic data value on path (i, j), S_j is a service configuration list in following service set, α and β are coefficient parameters applied to define the significance of local heuristics and pheromone. The pheromone is proven efficient through evaporation possibilities $P_{ij}^k(t)$, while the ant transfers from one node to another through equation (7).

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\tau_0 \quad (7)$$

Where, τ_0 is a primary pheromone value and ρ is an evaporation rate. At every iteration end, the ants estimate the solution quality they create. However, the ACO algorithm utilizes greedy selection to obtain optimal solutions. Then, pheromone paths are updated through the ant, which is called global pheromone updating, formulated in equations (8) and (9).

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\Delta\tau_{ij} \quad (8)$$

$$\Delta\tau_{ij} = \begin{cases} \frac{1}{L(t)} & if\ path(i,j) \in thebest\ path \\ \tau_{ij} & otherwise \end{cases} \quad (9)$$

Where, $L(t)$ is an optimum set of service configurations. The ant behaviours are strongly influenced by the values of numerous parameters

$(\alpha, \beta, \rho, \tau_0)$ which efficiently balance exploration and exploitation, facilitating the ACO to evade premature issues. This output is attained through effectively tuning the parameters of ACO.

4. EXPERIMENTAL RESULT

The result obtained from the proposed ARL is discussed in this section with the evaluation metrics of throughput, migration time, response time, load, energy consumption and resource availability. The simulation is performed in a Python environment with version 3.8 and system configurations like 16GB RAM, windows 10 OS and i7 processor. The proposed ARL framework signifies the advancement in cloud resource management through incorporating ACO for dynamic VM migration and TS optimization which offers efficiency and adaptability in cloud environments. Table 1 presents the simulation parameters for the ARL approach.

Table 1: Simulation Parameters

Parameter	Value/Range
Scheduling interval	0.01 - 1 seconds
Worker node speed	200 - 400 instructions/second
Worker node memory size	2 - 8 GB
Worker node swapping cost	2 - 10 instructions/GB
Worker quantum	0.01 - 0.5 seconds
Worker node startup time	120 - 600 seconds
Worker node scheduler notification time	1-5 instructions
Simulation time	10000ms

4.1 Performance Analysis

The results of the proposed ARL are discussed in this subsection in terms of evaluation metrics of throughput, migration time, response time, load, energy consumption and resource availability. The state-of-art methods such as Random, Round Robin, Longest Processing Time (LPT), Improved Round Robin (IRR) and First-Come, First-Serve (FCFS) are estimated and compared with the proposed ARL. The proposed ARL framework signifies the advancement in cloud resource management through incorporating ACO for dynamic VM migration and TS optimization which offers efficiency and adaptability in cloud environments. Table 2 displays the performance of the proposed ARL approach.

Table 2: Performance of proposed ARL

Performance Metrics	Methods					
	Random	Round-robin	LPT	IRR	FCFS	ARL
Throughput	1.29	1.31	2.27	1.07	0.12	2.33
Migration Time (ms)	20.32	19.54	17.76	23.43	34.54	12.64
Response Time(ms)	233.27	229.40	132.17	280.23	476.65	128.57
Load	0.007	0.006	0.005	0.011	0.013	0.022
Energy Consumption (W)	0.48	0.47	0.46	0.49	0.51	0.45
Resource availability	0.8334	0.8459	0.8732	0.8321	0.5443	0.9928

In the above table 2, the ARL approach's performance is analyzed with throughput, migration time, response time, load, energy consumption, and resource availability. The state-of-art methods such as Random, Round Robin, LPT, IRR and FCFS are considered and evaluated in which the proposed ARL achieves better performance. The proposed ARL framework signifies the advancement in cloud resource management through incorporating ACO for dynamic VM migration and TS optimization which offers efficiency and adaptability in cloud environments. The proposed ARL achieves a throughput of 2.33, migration time of 12.64ms, response time of 128.57ms, load of 0.0022, energy consumption of 0.45W, and resource availability of 0.9928, rendering a superior performance than state-of-art methods.

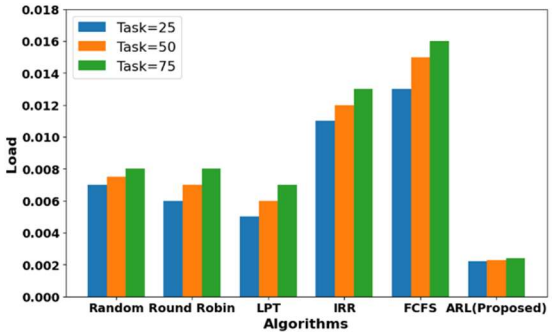

Figure 2: ARL performance in terms of Load

From Figure 2, the performance of ARL is shown in terms of load with various number of tasks of 25, 50 and 75. The state-of-art methods such as Random, Round Robin, LPT, IRR and FCFS are considered

and evaluated in which the proposed ARL achieves a superior performance. The proposed ARL achieves less Load of 0.0022, 0.0023 and 0.0024 for tasks 25, 50 and 75 which proves to be better than the state-of-art methods.

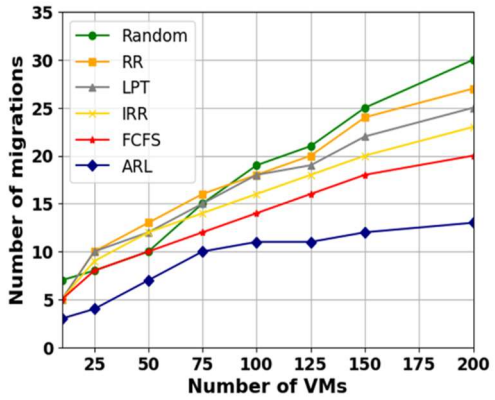

Figure 3: ARL performance in terms of number of migrations

In figure 3, the performance of ARL is shown in terms of the number of migrations with multiple number of VMs of 25, 50, 75, 100, 125, 150, 175 and 200. The state-of-art methods such as Random, Round Robin, LPT, IRR and FCFS are considered and evaluated in which the proposed ARL exhibits a better performance. The proposed ARL achieves a smaller number of migrations: 3, 4, 7, 10, 11, 11, 12 and 13 for respectively 25, 50, 75, 100, 125, 150, 175 and 200 number of VMs, offering a commendable outcomes than the state-of-art methods.

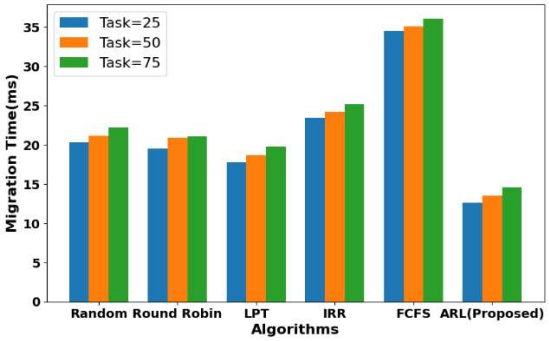

Figure 4: ARL performance in terms of migration time

In Figure 4, the performance of ARL is shown in terms of migration time (ms) with 25, 50 and 75 number of tasks. The state-of-art methods such as Random, Round Robin, LPT, IRR and FCFS are considered and evaluated in which the proposed ARL attains a higher performance. The proposed ARL obtains less migration time of 12.64ms, 13.54ms and 14.56ms for tasks 25, 50 and 75, respectively, which displays improved outcomes than the state-of-art methods.

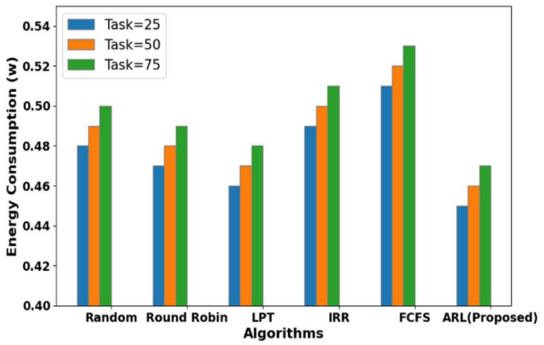

Figure 5: ARL performance in terms of energy consumption

In Figure 5, the performance of ARL is shown in terms of energy consumption (W) with various number of tasks: 25, 50 and 75. The state-of-art methods such as Random, Round Robin, LPT, IRR and FCFS are considered and evaluated in which the proposed ARL achieves better performance. The proposed ARL demands lesser energy of 0.45W, 0.46W and 0.47W for 25, 50 and 75 tasks, respectively, thereby offering better than the state-of-art methods.

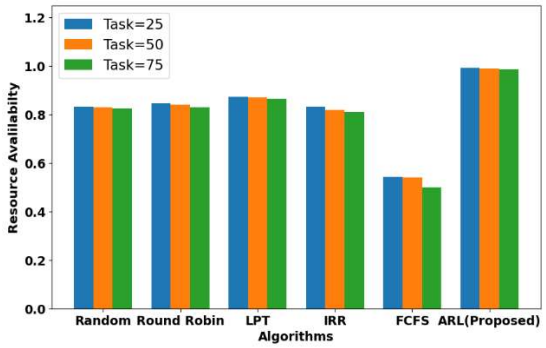

Figure 6: ARL performance in terms of resource availability

In the above figure 6, the performance of ARL is shown in terms of resource availability with 25, 50 and 75 number of tasks. The state-of-art methods such as Random, Round Robin, LPT, IRR and FCFS are considered and evaluated in which the proposed ARL achieves a better performance. The proposed ARL possessed high resource availability of 0.990, 0.987 and 0.987 for 25, 50 and 75 tasks, exhibiting a better performance than the state-of-art methods.

4.2 Comparative Analysis

The results of the proposed ARL are discussed in this subsection in terms of the evaluation metrics of throughput, migration time, response time, load,

energy consumption and resource availability. The existing methods such as Hybrid optimization [22] and SS-CAU [23] are taken and compared with the proposed ARL approach. The proposed ARL framework signifies the advancement in cloud resource management through incorporating ACO for dynamic VM migration and TS optimization which offers efficiency and adaptability in the cloud environments. The proposed ARL achieves a throughput of 2.33, migration time of 12.64ms, response time of 128.57ms, load of 0.0022, energy consumption of 0.45W, and resource availability of 0.9928. Table 3 displays the performance of the comparative analysis.

Table 3: Comparative analysis

Methods	Throughput	Energy Consumption (W)	Resource availability
Hybrid optimization [22]	NA	0.4703	0.9914
SS-CAU [23]	0.0068	1.5833	NA
Proposed ARL	2.33	0.45	0.9928

4.3 Discussion

The limitations of the existing research and the advantages of the proposed ARL are explained in this section. Hybrid optimization [22] is restricted by dynamic cloud workloads, potentially leading to suboptimal VM placements and resource allocation. The SS-CAU [23] only focuses on VM consolidation migration and does not address other aspects of CC systems. DRLBTSA [26] required significant computational resources for training which leads to enhanced energy consumption. The proposed ARL framework signifies the advancement in cloud resource management through incorporating ACO for dynamic VM migration and TS optimization which offers efficiency and adaptability in cloud environments. The ACO is selected for task scheduling because of its distributed optimisation adaptability and capability that enhances resource utilization and VM migration performances. The proposed ARL achieves a throughput of 2.33, migration time of 12.64ms, response time of 128.57ms, load of 0.0022, energy consumption of 0.45W, and resource availability of 0.9928. The ARL achieves better results in terms of less energy consumption, high throughput and resource availability by learning and adjusting policies continuously based on its environmental feedback. It optimizes resource allocation dynamically through selecting actions which reduces energy consumption while enhancing throughput. By adapting to change learning and conditions from previous feedbacks, the model balance tradeoff and achieve better result.

5. CONCLUSION

The ARL approach is proposed in this research based on TS for VM migration in the CC environment. The proposed ARL approach signifies the advancement in cloud resource management through incorporating ACO for dynamic VM migration and TS optimization which offers efficiency and adaptability in cloud environments. The ACO is selected for task scheduling because of its distributed optimisation adaptability and capability that enhances efficient resource utilization and the VM migration performances. The ARL approach performance is evaluated in terms of metrics of throughput, migration time, response time, load, energy consumption and resource availability. The ARL optimizes resource allocation through learning which action leads to energy-efficient operations and enhancing throughput while reducing resource usage. It constantly improves these policies by trial and error which ensures the resource availability is maintained without conceding performance ultimately leading much effective and system operation. The ARL achieves a throughput of 2.33, migration time of 12.64 ms, response time of 128.57 ms, and energy consumption of 0.45W. In the future, various optimization algorithms can be applied for further effectively scheduling tasks in VM migration.

REFERENCES:

[1] C.H. Tran, T.K. Bui, and T.V. Pham, “Virtual machine migration policy for multi-tier application in cloud computing based on Q-learning algorithm”, *Computing*, Vol. 104, No. 6, 2022, pp. 1285-1306.

[2] T. Alyas, T.M. Ghazal, B.S. Alfurhood, M. Ahmad, O.A. Thawabeh, K. Alissa, and Q. Abbas, “Performance Framework for Virtual Machine Migration in Cloud Computing”, *Computers, Materials & Continua*, Vol. 74, Noo. 3, 2023, pp. 6289-6305.

[3] R. Shaw, E. Howley, and E. Barrett, “Applying reinforcement learning towards automating energy efficient virtual machine consolidation in cloud data centers”, *Information Systems*, Vol. 107, 2022, p. 101722.

[4] S. Supreeth, K. Patil, S.D. Patil, S. Rohith, Y. Vishwanath, and K.S.V. Prasad, “An Efficient Policy-Based Scheduling and Allocation of Virtual Machines in Cloud Computing

Environment", *Journal of Electrical and Computer Engineering*, 2022, p. 5889948.

[5] M.H. Sayadnavard, A.T. Haghighat, and A.M. Rahmani, "A multi-objective approach for energy-efficient and reliable dynamic VM consolidation in cloud data centers", *Engineering science and technology, an International Journal*, Vol. 26, 2022, p. 100995.

[6] D. Saxena, and A.K. Singh, "OFP-TM: an online VM failure prediction and tolerance model towards high availability of cloud computing environments", *The Journal of Supercomputing*, Vol. 78, No. 6, 2022, pp. 8003-8024.

[7] A. Kaur, S. Kumar, D. Gupta, Y. Hamid, M. Hamdi, A. Ksibi, H. Elmannai, and S. Saini, "Algorithmic Approach to Virtual Machine Migration in Cloud Computing with Updated SESA Algorithm", *Sensors*, Vol. 23, 2023, p. 6117.

[8] Y. Xu, and K. Abnoosian, "A new metaheuristic-based method for solving the virtual machines migration problem in the green cloud computing", *Concurrency and Computation: Practice and Experience*, Vol. 34, No. 3, 2022, p. e6579.

[9] H. Zhao, N. Feng, J. Li, G. Zhang, J. Wang, Q. Wang, and B. Wan, "VM performance-aware virtual machine migration method based on ant colony optimization in cloud environment", *Journal of Parallel and Distributed Computing*, Vol. 176, 2023, pp. 17-27.

[10] A. Bhardwaj, and C.R. Krishna, "A container-based technique to improve virtual machine migration in cloud computing", *IETE Journal of Research*, Vol. 68, No. 1, 2022, pp. 401-416.

[11] R. Boopathi, and E.S. Samundeeswari, "An Optimized VM Migration to Improve the Hybrid Scheduling in Cloud Computing", *International Journal of Intelligent Engineering & Systems*, Vol. 17, No. 1, 2024, pp. 483-492.

[12] N.I. Henry, C. Anbuananth, and S. Kalarani, "Hybrid meta-heuristic algorithm for optimal virtual machine placement and migration in cloud computing", *Concurrency and Computation: Practice and Experience*, Vol. 34, No. 28, 2022, p. e7353.

[13] B. Liang, X. Dong, Y. Wang, and X. Zhang, "A high-applicability heterogeneous cloud data centers resource management algorithm based on trusted virtual machine migration", *Expert Systems with Applications*, Vol. 197, 2022, p. 116762.

[14] A. Kushchazli, A. Safargalieva, L. Kochetkova, and A. Gorshenin, "Queuing Model with Customer Class Movement across Server Groups for Analyzing Virtual Machine Migration in Cloud Computing", *Mathematics*, Vol. 12, 2024, p. 468.

[15] K. Tuli, and M. Malhotra, "Optimal Meta-Heuristic Elastic Scheduling (OMES) for VM selection and migration in cloud computing", *Multimedia Tools and Applications*, Vol. 83, No. 12, 2024, pp. 34601-34627.

[16] J. Wang, H. Gu, J. Yu, Y. Song, X. He, and Y. Song, "Research on virtual machine consolidation strategy based on combined prediction and energy-aware in cloud computing platform", *Journal of Cloud Computing*, Vol. 11, No. 1, 2022, p. 50.

[17] T. Alyas, I. Javed, A. Namoun, A. Tufail, S. Alshmrany, and N. Tabassum, "Live migration of virtual machines using a mamdani fuzzy inference system", *Computers, Materials & Continua*, Vol. 71, No. 2, pp. 3019-3033, 2022.

[18] S. Talwani, J. Singla, G. Mathur, N. Malik, N.Z. Jhanjhi, M. Masud, and S. Aljahdali, "Machine-Learning-Based Approach for Virtual Machine Allocation and Migration", *Electronics*, Vol. 11, 2022, p. 3249.

[19] R. Mangalagowri, and R. Venkataraman, "Ensure secured data transmission during virtual machine migration over cloud computing environment", *International Journal of System Assurance Engineering and Management*, 2023.

[20] C. Saravanakumar, R. Priscilla, B. Prabha, A. Kavitha, M. Prakash, and C. Arun, "An Efficient On-Demand Virtual Machine Migration in Cloud Using Common Deployment Model", *Computer Systems Science & Engineering*, Vol. 4, No. 1, pp. 245-256, 2022.

[21] N.K. Sharma, S. Bojjagani, Y.C.A.P. Reddy, M. Vivekanandan, J. Srinivasan, and A.K. Maurya, "A Novel Energy Efficient Multi-Dimensional Virtual Machines Allocation and Migration at the Cloud Data Center", *IEEE Access*, Vol. 11, 2023, pp. 107480-107495.

[22] M.S.A. Khan, and R. Santhosh, "Hybrid optimization algorithm for vm migration in cloud computing", *Computers and Electrical Engineering*, Vol. 102, 2022, p. 108152.

[23] Z. Ma, D. Ma, M. Lv, and Y. Liu, "Virtual machine migration techniques for optimizing energy consumption in cloud data centers", *IEEE Access*, Vol. 11, 2023, pp. 86739-86753.

[24] A. Gupta, and S. Namasudra, "A novel technique for accelerating live migration in cloud computing", *Automated Software Engineering*, Vol. 29, 2022, p. 34.

[25] B. Kruekaew, and W. Kimpan, "Multi-objective task scheduling optimization for load balancing in cloud computing environment using hybrid artificial bee colony algorithm with reinforcement learning", *IEEE Access*, Vol. 10, 2022. pp. 17803-17818.

[26] S. Mangalampalli, G.R. Karri, M. Kumar, O.I. Khalaf, C.A.T. Romero, and G.A. Sahib, "DRLBTSA: Deep reinforcement learning based task-scheduling algorithm in cloud computing," *Multimedia Tools and Applications*, Vol. 83, No. 3, 2024, pp.8359-8387.