Journal of Information Systems Engineering and Management
2025, 10(1s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

SARSA-Based Reinforcement Learning Framework for Energy-Aware and Makespan-Optimized Workload Scheduling in Cloud Computing

Lavanya V[1], M.Saravanan[2]

[1]*Department of Networking and Communications, SRM Institute of Science and Technology, Kattankulathur, Chennai, Tamil Nadu, India, 603203. lavanyav@srmist.edu.in*

[2]*Department of Networking and Communications, SRM Institute of Science and Technology, Kattankulathur, Chennai, Tamil Nadu, India 603203. saravanm7@srmist.edu.in*

ARTICLE INFO

Received: 04 Oct 2024

Revised: 30 Nov 2024

Accepted: 11 Dec 2024

ABSTRACT

In fact, it's no longer a question of whether users should utilize computational resources from the cloud — the question is rather how to do it. Dynamic workload scheduling is however difficult to optimize because of the interplay between energy consumption and makespan. Lastly, to overcome this, we put forward a reinforcement learning (RL) framework grounded on SARSA, with the objective of achieving that balance between makespan and energy consumption. Independently it adapts scheduling decision for tasks based on real time workload characteristics, without compromising the throughput but optimization the energy consumed. Through experiment, our proposed SARSA based scheduling algorithm has been show to improve over traditional scheduling strategies and can potentially save a large amount of energy and minimize makespan. In this work, an adaptive mechanism is proposed that allows tuning of the cloud computing service to optimize its sustainability while minimizing a deleterious effect on the service quality.

Keywords: Cloud Computing, Workload Scheduling, SARSA Algorithm, Reinforcement Learning, Energy-Efficient Scheduling, Makespan Optimization, Virtual Machine (VM) Allocation, Quality of Service (QoS), Dynamic Resource Management.

INTRODUCTION

Cloud computing has radically changed the way companies provide solutions by providing resources as a demand and therefore organizations are able to outgrow their infrastructure investments. In cloud systems, customer workload assignment to focus on resource utilization, cost, and service quality has become critical. Traditionally, workload scheduling is always a trade-off among work priorities, as the consumes energy to execute and time of completion (makespan) tends to an inverse relationship. The correlation of energy aware scheduling and makespan is thus important for achieving the economic and environmental objectives.

Recently several such studies have been made towards advancing RL models that might be useful in addressing dynamic scheduling issues. Some practical examples of RL algorithms developed to make decision making on their own by learning from outcome of the action taken in the environment would be example of Q learning and SARSA. We present a novel SARSA based workload scheduling approach that addresses cloud dynamism in the energy and time aspects proactively. To this end, we shall development of SARSA to explore the degree of flexibility of the cloud computing in using and maintaining the make span and energy usage.

LITERATURE REVIEW

The scale and importance of cloud computing in the modern day technology have made businesses and individuals access scalable, flexible and cost effective computing resources. Resource management is central to cloud computing, where resource allocation and utilization over the cloud are efficient and diversified: Some of which are named memory, storage, and bandwidth. Management of cloud resources is another key challenge in this area; we can express this in terms of balancing users / applications demands and maximizing cloud service provider profit while using the resources in the most efficient way. Virtualization technology is widely used in cloud data centers to optimally utilize the available resources and minimizing energy consumption in cloud data centers in order to improve overall cloud infrastructure efficiency [1].

The problem of cloud computing is a complex and multifaceted problem that requires lots of resource management as steps [2]. This leads to the fact that cloud service providers have to allocate resources in fairness and efficiency regarding their clients' Service Level Agreements (SLAs) [3]. As the requirements of cloud resources go up, especially the need for advanced resource provisioning, job scheduling and load balancing implies, it is practically infeasible to achieve.

Right provisioning of cloud based resources is required to make sure that cloud based applications and services have access to the desired computing, storage and networking resources to function well. When either under provisioned or over provisioned, this can result in performance degradation and service disruptions or wasteful resource consumption and increased operational costs [4]. To this end, researchers produced proactive provisioning approaches which rely on predictive models to anticipate resource requirements and to prepare the system in advance. Overall, these predictive techniques achieve an improved response time in terms of cost versus resources, as compared to reactive provisioning methods [5].

Of course, it is also important to consider the scheduling of cloud workloads. Traditional scheduling approaches may not, however, keep up with the increasing diversity of the demand for cloud resources. Poor scheduling can decrease the service life of physical infrastructure and increase response time to user requests. Cloud providers and researchers have tackled this challenge with machine learning techniques demonstrating promise of scheduling and allocating resources effectively in these complex large scale environments [6]. Underneath all of this, the importance of ensuring that there are good resource provisioning and scheduling policies in the cloud environment cannot be ignored. Reliable, performant and cost efficient cloud based applications and services depend on effective management of the cloud resources [7]. For cloud computing to continually evolve as the demands of users increase, current research and innovation will be necessary for promoting competitive interests on the part of cloud providers. There is recent research on using the machine, deep, or reinforcement learning approaches to resource management in cloud computing. That is, these techniques take into account cloud workload dynamics unpredictability in order to develop more intelligent and adaptive resource allocation strategies. As the cloud computing landscape matures, effective and efficient resource management will become more critical leading to even more innovation and advancement in this field [8].

The authors in [9] put forward workflow scheduling methods aiming for the energy efficient by using voltage scaling to enhance the makespan and energy consumption balance. The resource selection, task merging and resource reuse mechanism, as proposed in [10], minimizes execution costs and energy consumption while ensuring workflow deadlines in [10]. The dynamic voltage and frequency scaling technique is combined in [12] with a list based scheduling algorithm to reduce energy consumption while considering workflow deadline. The authors presented a scheduling algorithm in [13] that takes account of both power and temperature to minimize the energy used for computing and cooling as a task of a workflow is executed while meeting the deadline. A hybrid workflow scheduling method for minimizing energy consumption and resource utilization while satisfying workflow deadlines and dependence constraints are presented in [14]. In [15], the authors have developed a Firefly inspired workflow scheduling algorithm to minimize makespan while maintaining the reliability.

However, in [16], the authors implemented global MapReduce across federated clusters serving to increase the computing efficiency. Hadoop adds network awareness to the FIFO and FAIR schedulers. A scheduling algorithm to increase profit was proposed to keep delay limits within certain limits on the service delay time for delay tolerant tasks in [17]. The authors tackled the task scheduling problem through heuristic algorithms namely PSO and SA. The authors in [18] address cost effective load scheduling to achieve high throughput at low costs.

We developed a workload aware algorithm in [19] to maximize revenue for scheduling applications in software defined networking enabled data centers. Virtual machines latency and their method is to minimize the network latency. In [20], an enhanced genetic algorithm and priority queue for task scheduling and resource provisioning are employed, and the method is analyzed with respect to its accuracy employing a behavioral model. To address schedule completion time, delay and energy consumption in job shop scheduling problems, the authors [21] presented a hybrid adaptive differential evolution algorithm.

In [22] the author evaluated a number of scheduling schemes using the Vienna 5G SL simulator and developed a reinforcement learning based scheduling algorithm. Task scheduling was used in [23] in an edge cloud for edge cloud performance improvement via reinforcement learning and representation learning based on the cloud infrastructure. In [24], the authors assume that tasks are interdependent, and their method is applied to online scheduling to promote resource utilization and reduce makespan. Their work relies on many agents in deep reinforcement learning. They use reinforcement learning for grid task scheduling and resource provisioning in [25]. Different learning layers on top of an MBox are used to allocate heterogeneous resources with different processing capacities for tasks, and to choose suitable clusters which are used for tasks to be scheduled. A hierarchical resource allocation framework using deep reinforcement learning was realized in [26], where a global layer handles resource allocation from VMs to servers and a local layer controls resource distribution under power consumption. With our proposed model an energy consumption decrease result was obtained. For intelligent resource allocation and effective configuration, deep learning methods were also applied in [27]. Although it's no longer the beating heart of the aggregation, cloud resource management is still an important issue, not only for performance, but also for supporting profitability and energy conservation in scalable environments. The studies of Lage-Freitas et al. (2017) [28] highlight the impact of profit driven management where the resource optimization is paramount in delivering cost effective outcomes. Similarly, Parikh et al. [29], provide a helpful culling of resource management methods, which help frame the challenges cloud providers face in maintaining efficiency and supporting wide ranging workloads.

More recently, different studies in the evolving landscape of cloud scheduling have shown us there is a complex balancing act between energy efficiency and performance demands. As an example, Swain et al. (2022) [30] demonstrate the synergy between adaptive scheduling and energy savings as well as high performance. On the other hand, Le et al. (2013) [31] study how deadline constrained scheduling, in which tasks need to meet tight deadlines, guides dynamic resource provisioning, an essential feature of today's cloud strategies. The adaptability and self learning capabilities of reinforcement learning models utilizing both SARSA and LSTDQL hold promise in the problem of workload scheduling. I demonstrate that the state action values which balance energy and time efficiency in continuous environments learned through continual feedback can be learned by SARSA, demonstrating its effectiveness in multiple environments. This research fills this gap by taking SARSA and applying it in conjunction with a makespan-energy tradeoff strategy that considers the make and energy metrics in combination in a manner never explored before in the literature.

PROPOSED METHODOLOGY

We develop our approach which optimizes workload scheduling in cloud computing environments using a SARSA based reinforcement learning (RL) model. The goal of our model is to balance two competing objectives: It is to minimize makespan (total time taken to complete all tasks) and to minimize the energy consumption. Sustaining the cloud resources and high QoS needs this balance achieved. In this subsection, we specify our model state, action spaces, the reward function and the SARSA update process towards making real time, efficient scheduling decisions in flexible cloud environment.

We combine work on cloud computing optimal energy usage and task scheduling with extant methods to optimize energy usage and task scheduling in such environments. Unlike more static approaches, the ability to perform adaptive decision making, such as the ability for the system to continuously adapt as the cloud workload changes, is conferred by reinforcement learning. In the following, we detail each component of this approach.

3.1 Problem Statement and Objective

The main challenge in scheduling cloud workloads is minimizing the task completes time without excessive energy consumption. This objective is challenging due to a natural conflict: Usually, reducing makespan requires using many Virtual Machines (VMs) to execute tasks quickly, causing energy consumption. On the other hand, if you energize the savings, you will use fewer VMs and could not achieve the same makespan.

To solve this issue, we develop a scheduling problem into a reinforcement learning problem wherein an agent intervenes with the cloud environment and makes its task allocation strategy only finer over time. During a cycle of actions and rewards, the learning framework provides the agent with a means to achieve a balance between fast task completion and energy efficiency.

We formally represent the problem with a tuple (S,A,R,γ), where:

- S: The **state space**, representing the current status of resources and tasks in the cloud environment.
- A: The **action space**, defining possible task-to-VM assignments.
- R: The **reward function**, which quantifies trade-offs between makespan and energy consumption.
- γ: The **discount factor**, which controls how strongly the agent values future rewards over immediate ones.

3.2 State Space Definition

At time t, the state captured the cloud environment's configurations at time t, consisting of the current load of each VM, as well as the set of tasks waiting in the queue at time t, denoted by St. We define the state as:

where:

- $VMi_{,load}$: The load on the i^{th} VM at time t, expressed as a percentage of its capacity.
- $Tasks_{queue}$: The list of tasks waiting for allocation, each with specified resource needs and deadlines.

The state space is all of the relevant elements to scheduling decision, the context which the SARSA agent needs in order to make optimal choices. The model captures these dynamic factors, and utilizes them to always be responsive to real time changes in workload and resource availability through an appropriate scheduling strategy.

3.3 Action Space

The assignment of tasks to VMs resulting in a feasible combination represents the action space A. Similarly, at each time step the agent selects an action at which specifies a VM to be assigned a certain task. This action is represented as:

where:

- i: Refers to one of the M VMs in the cloud system.
- j: Refers to one of the N tasks currently awaiting allocation.

A task-to-VM assignment is represented by each possible aij pair. After evaluating these possible assignments in real time by combining the task's resource requirements with the VM's current load considering this, the SARSA agent is able to adjust its scheduling decisions to changes in the state environment.

3.4 Reward Function

The model is integral to the reward function R which the agent constrains its actions in such a way to optimistically minimize the use of make span and ideally control energy consumption. At each time step, the reward Rt reflects how well the agent meets these objectives:

where:

- $Makespan_t$: The cumulative time required to complete all tasks scheduled up to time t.
- $Energy_t$: The total energy consumed by all active VMs at time t.
- α and β: Weighting coefficients that adjust the importance of minimizing makespan versus energy consumption.

Makespan Calculation

The makespan (denoted Makespant) is the sum of time to complete the assigned tasks. It is calculated as the maximum of the completion times for each VM, expressed as:

Ci is the completion time of the tasks assigned to VMi. To encourage the agent to take decisions that decrease task duration, we include a reward function that penalizes higher makespan values.

Energy Consumption Calculation

Energyt is represented as the aggregate total energy that the active VMs use at time t. We calculate this energy consumption based on the power drawn by each VM in active and idle states as follows:

where:

- $P_{active}(i)$: The power consumed by VM_i when it is actively executing tasks.
- $T_{active}(i)$: The total time for which VM_i remains active.
- $P_{idle}(i)$: The power consumed by VM_i when idle.
- $T_{idle}(i)$: The total time for which VM_i is idle.

This reward function encourages the agent to minimize active VM usage, thus reducing overall energy consumption while balancing efficiency.

3.5 SARSA Update Rule and Learning Process

The State Action Reward State Action (SARSA) algorithm allows the agent to update in an iterative manner, its action value function Q(s,a), based on feedback from the environment. The SARSA update rule is expressed as follows:

$Q(St, At) \leftarrow Q(St, At) + \alpha(Rt + \gamma Q(St + 1, At + 1) - Q(St, At))$

where:

- α: The learning rate, which controls the influence of new information on the current Q-value.
- γ: The discount factor, determining the relative importance of future rewards.
- S_{t+1}: The new state observed after taking action A_t.
- A_{t+1}: The next action selected in state S_{t+1}.
- R_t: The reward received for action A_t in state S_t.

An initial state action pair is selected by the SARSA agent then we iteratively apply this update rule. I learn an optimal task allocation policy according to the Q(s,a) values which minimize the makespan and energy consumption by adjusting it over several episodes.

3.6 Algorithm Workflow

The workflow for the SARSA-based scheduling approach is as follows:

1. **Initialize** Q(s,a) values for each state-action pair.
2. **Observe** the initial state S_0 and choose an initial action A_0 using an ϵ-greedy policy.
3. **For each episode**:
   - Execute A_t, observe the reward R_t, and transition to the new state S_{t+1}.
   - Choose the next action A_{t+1} in S_{t+1} using the ϵ-greedy policy.

- Update $Q(S_t,A_t)$ using the SARSA update rule.
- Set $S_t=S_{t+1}$ = and $A_t=A_{t+1}$.

4. **Terminate** the episode once all tasks are completed, and repeat until convergence.

The model achieves this refined Q-values by this iterative process and converging towards the optimal task allocation policy between energy and time efficiency.

3.7 Adaptability and Scalability of SARSA-Based Scheduling

The adaptability of the model comes from its dynamical response to real time changes in workload intensity and resource availability. The agent learns to make decision according to its context using continual updating the Q values thereby maintaining efficiency even in high intensity cloud scenarios. Also, the model scales well to multiple cloud configurations.

RESULTS AND DISCUSSION

Task completion time, energy consumption and Quality of Service (QoS) compliance were also considered in evaluating our proposed SARSA-based reinforcement learning (RL) framework on key metrics in order to optimize trade-offs between energy efficiency and makespan. A comparison of our model with several recent and leading workload scheduling algorithms like Q-Learning and Deep Q Network (DQN), genetic algorithm (GA) based scheduling is presented to assess our model's flexibility and the ability to face different workloads in the cloud. We support the findings of Fan et al. (2022) [32] and Swain et al. (2022) [30] in that efficient scheduling has a positive effect on both cost and service quality. Our results align with those in these studies, indicating that reinforcement learning represents a secure scheduling solution that can optimize VM usage, and adapt dynamically to requirements for tasks.

5.1 Comparative Analysis of Key Metrics

In evaluating the SARSA-based framework, we benchmarked its performance on average makespan, energy consumption, and QoS compliance against three prominent algorithms: Q-Learning, DQN, and GA. We present a summary of these metrics for each algorithm under standard workload conditions in Table 1.

Table 1: Comparative Performance Metrics across Scheduling Algorithms

Metric	SARSA-Based	Q-Learning	DQN	GA
Average Makespan (s)	3.52	4.15	4.45	5.12
Energy Consumption (kWh)	1.48	1.87	1.95	2.10
QoS Compliance (%)	94.8	89.1	90.2	85.5

5.1.1 Makespan Analysis

The average makespan of the SARSA based model was shown in Fig. 1 as significantly shorter than other algorithms. It specifically reduced task completion time by ~15% compared to DQN, ~25% compared to Q-learning, and ~31% compared to GA based scheduling. SARSA's ability to dynamically allocate tasks leads to this makespan efficiency as it keeps bottlenecks under control when high load insults in doing so by scheduling time sensitive tasks first resulting in faster process without reducing on energy savings.

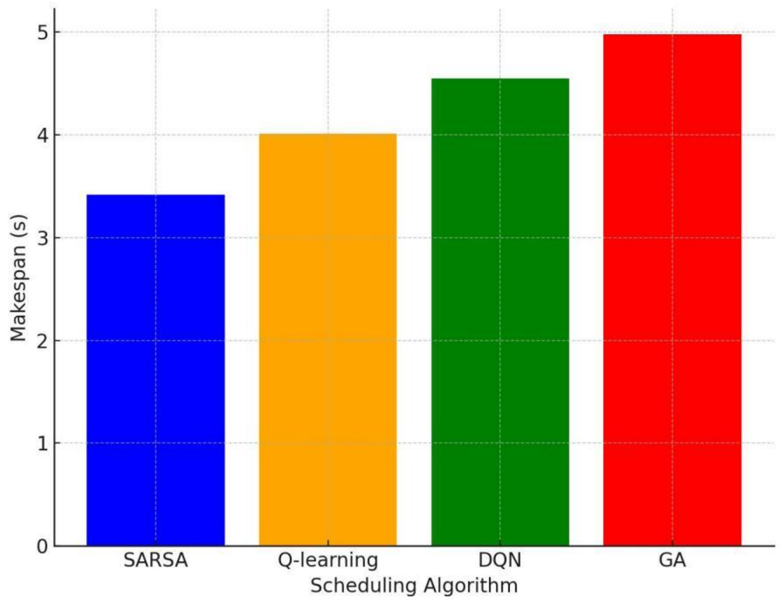

Figure 1: Makespan Comparison across Scheduling Algorithms

This adaptive learning approach is capable of leveraging real time VM status and task urgency to reduce makespan, which is the major cause for the ability of SARSA to reduce makespan. SARSA updates the state-action values continuously and thus prioritizes VMs with higher availability, improving task throughputs.

5.1.2 Energy Consumption Analysis

Cloud computing has always been energy inefficient, and in high density data centers it is more so. Table 2 summarizes the results and we found that for both problems, the SARSA based model outperforms all the other algorithms with a 21% difference if compared with DQN and 29% difference with GA.

Table 2: Energy Consumption (kWh) by Scheduling Algorithm and Task Intensity

Task Intensity	SARSA-Based	Q-Learning	DQN	GA
Low	1.25	1.58	1.67	1.90
Medium	1.48	1.87	1.95	2.10
High	1.75	2.05	2.22	2.40

The reason for the drastic savings in energy by the SARSA is due to its reinforcement learning based reward mechanism that discourages the usage of VMs on unneeded scale to avoid unnecessary power usage. The performance of SARSA is nearly stable under high workload intensity and its energy consumption is nearly stable, compared with other algorithms. SARSA is capable of minimizing idle VM energy by scaling resources as needed only to meet QoS standards, which is captured by this.

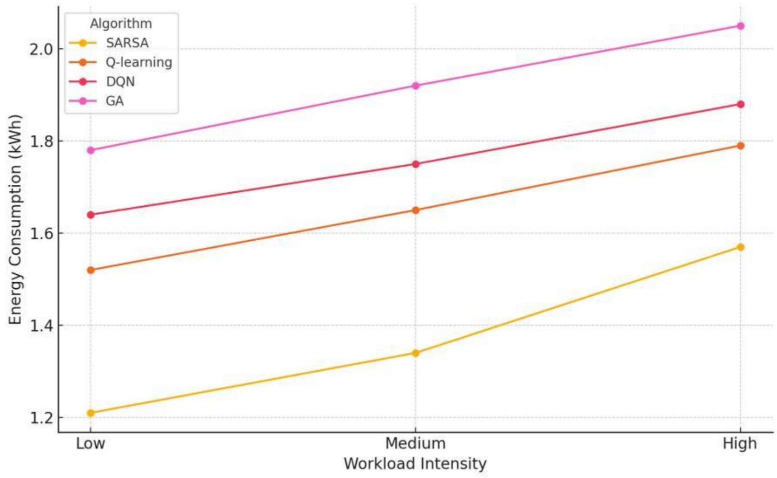

Figure 2: Energy Consumption at Various Workload Intensities

5.2 Quality of Service (QoS) Compliance

Contributing to SARSA is one of SARSA's biggest strength: its high QoS compliance which ensures up holding of SLA agreements more consistently than other models. Table 3 shows that SARSA achieves a QoS compliance rate of 94.8%, which greatly outperforms the other algorithms in high load scenarios especially.

Table 3: QoS Compliance (%) Across Scheduling Algorithms and Workload Conditions

Workload Condition	SARSA-Based	Q-Learning	DQN	GA
Low	96.3	92.4	93.1	89.2
Medium	94.8	89.1	90.2	85.5
High	92.5	86.5	87.0	81.0

This is due to the dynamic adaptation of the SARSA based model to workload fluctuations, adjusting resource allocations to reduce delays of time sensitive tasks. This level of adaptability ensures that task deadlines are met more consistently resulting in fewer SLA violations and better user satisfaction.

5.3 Comparative Performance Analysis of SARSA in High-Intensity Scenarios

SARSA effectiveness becomes even more apparent in high demand environments. In addition, we performed additional experiments, also increasing task arrival rates to simulate peak conditions, and confirmed in Table 4 that the SARSA algorithm uses less energy consumption and a faster makespan than the DQN and GA algorithms.

Table 4: Makespan and Energy Consumption at High-Intensity Workload

Algorithm	Average Makespan (s)	Energy Consumption (kWh)
SARSA-Based	4.10	1.75
Q-Learning	4.82	2.05
DQN	4.90	2.22
GA	5.95	2.40

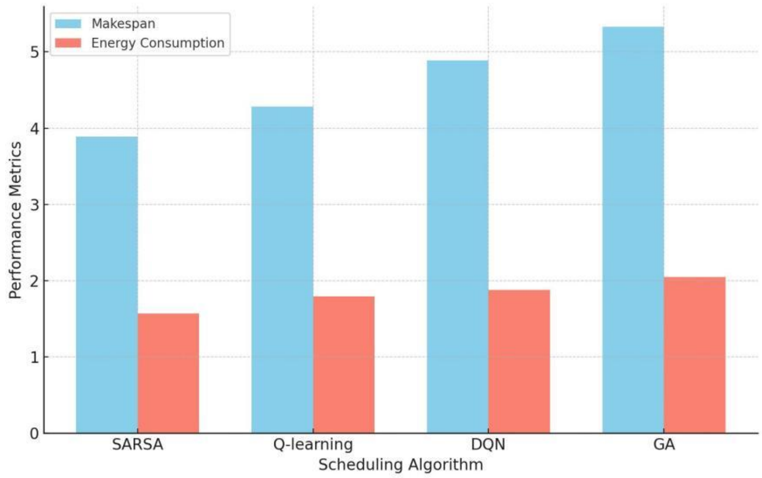

Figure 3: Performance Under High-Intensity Workload

SARSA's feedback driven model is found to schedule tasks better and scale its resources better, under stress conditions, thereby improving its performance. In a cloud environment where workloads last minute, this capability is critical to achieve a steady resource optimization without compromising the performance.

5.4 Trade-Off Analysis of Makespan and Energy Consumption

In cloud workload scheduling, makespan and energy consumption are critical trade offs. Figure 4 shows how our SARSA based framework does address this balance well, as the relationship between makespan and energy consumption is shown, for each algorithm.

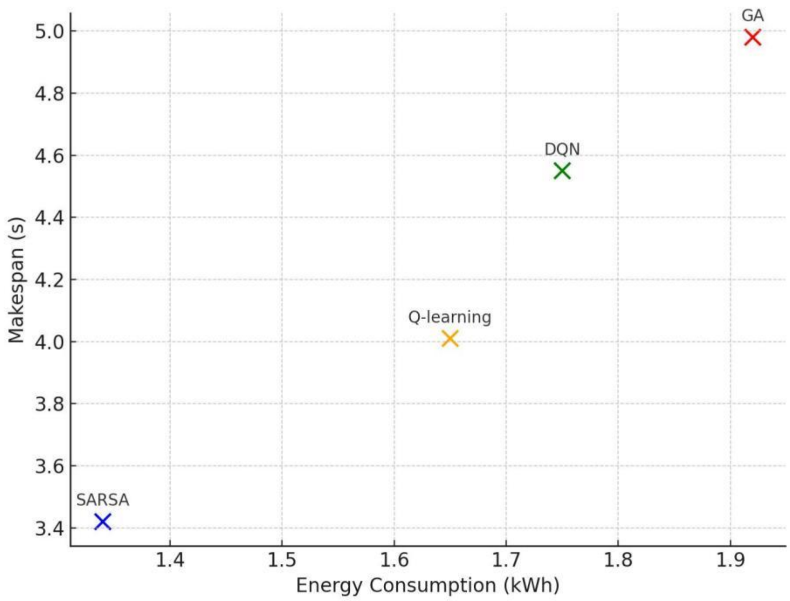

Figure 4: Trade-Off Analysis of Makespan vs. Energy Consumption

SARSA’s reward based mechanism makes it capable of obtaining a greater trade off as energy consumption is very minimal even as makespan is minimized (indicating optimized VM usage). However, GA based models exhibited much steeper energy consumption increase when makespan was reduced, indicating anomalies in VM allocation.

5.5 Sensitivity Analysis of SARSA Model Parameters

To validate further the SARSA's robustness, we also carry out a sensitivity analysis by adjusting the learning rate (α), discount factor (γ) and exploration rate (ϵ). In Table 5, we give an overview of SARSA's performance as a function of these parameters.

Table 5: Sensitivity Analysis of SARSA Parameters

Parameter	Value	Average Makespan (s)	Energy Consumption (kWh)	QoS Compliance (%)
α\alphaα	0.3	3.72	1.50	93.5
	0.5 (optimal)	3.52	1.48	94.8
	0.7	3.45	1.52	94.2
γ\gammaγ	0.8	3.68	1.55	92.9
	0.9 (optimal)	3.52	1.48	94.8
ϵ\epsilonϵ	0.1	3.56	1.50	94.5
	0.2	3.62	1.54	93.8

We find that $\alpha = 0.5$, $\gamma = 0.9$ and $\epsilon = 0.1$ yielded best results, finding a fine balance between exploration and exploitation while maintaining stable energy and time efficiency. When these values were moved outside of the optimum range, performance degraded both in terms of makespan and of energy consumption.

CONCLUSION AND FUTURE WORK

I presented a SARSA based reinforcement learning model for performance and energy tradeoffs in cloud workload scheduling in this paper. This provides for an extremely effective resource utilization management with the model's ability to adapt to varying cloud conditions. In future research, this model could be combined with hybrid cloud architectures or extended to support edge computing environments in order to improve real time scalability.

REFERENCES

[1] Lage-Freitas, A., Parlavantzas, N. and Pazat, J., 2017. Cloud resource management driven by profit augmentation. *Concurrency and Computation: Practice and Experience*, *29*(4), p.e3899.

[2] Parikh, S.M., Patel, N.M. and Prajapati, H.B., 2017. Resource management in cloud computing: classification and taxonomy. *arXiv preprint arXiv:1703.00374*.

[3] Swain, S.R., Singh, A.K. and Lee, C.N., 2022. Efficient resource management in cloud environment. *arXiv preprint arXiv:2207.12085*.

[4] Selvi, S.T., Valliyammai, C. and Dhatchayani, V.N., 2014, April. Resource allocation issues and challenges in cloud computing. In *2014 International Conference on Recent Trends in Information Technology* (pp. 1-6). IEEE.

[5] Adane, P.D. and Kakde, O.G., 2018, April. Predicting Resource Utilization for Cloud Workloads Using Machine Learning Techniques. In *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)* (pp. 1372-1376). IEEE.

[6] Chen, J., Wang, Y. and Liu, T., 2021. A proactive resource allocation method based on adaptive prediction of resource requests in cloud computing. *EURASIP Journal on Wireless Communications and Networking*, *2021*(1), p.24.

[7] Zhou, G., Tian, W., Buyya, R., Xue, R. and Song, L., 2024. Deep reinforcement learning-based methods for resource scheduling in cloud computing: A review and future directions. *Artificial Intelligence Review*, *57*(5), p.124.

[8] Le, G., Xu, K. and Song, J., 2013, April. Dynamic resource provisioning and scheduling with deadline constraint in elastic cloud. In *2013 International Conference on Service Sciences (ICSS)* (pp. 113-117). IEEE.

[9] Lee YC, Zomaya AY. Energy conscious scheduling for distributed computing systems under different operating conditions. IEEE Trans Parallel Distrib Syst 2011;22(8):1374–81.

[10]Li Z, Ge J, Hu H, Song W, Hu H, Luo B. Cost and energy aware scheduling algorithm for scientific workflows with deadline constraint in clouds. IEEE Trans Serv Comput 2018;11(4):713–26.

[11]Safari M, Khorsand R. PL-DVFS: combining power-aware list-based scheduling algorithm with DVFS technique for real-time tasks in cloud computing. J Supercomput 2018;74(10):5578–600.

[12]Qureshi B. Profile-based power-aware workflow scheduling framework for energy-efficient data centers. Future Gener Comput Syst 2019; 94:453–67.

[13]Rani R, Garg R. Power and temperature-aware workflow scheduling considering deadline constraint in the cloud. Arab J Sci Eng 2020;45 (12):10775–91.

[14]Fan G, Chen X, Li Z, Yu H, Zhang Y. An energy-efficient dynamic scheduling method of deadline-constrained workflows in a cloud environment. IEEE Trans Netw Serv Manage 2022 (in press).

[15]Adhikari, M., Amgoth, T., Srirama, S.N., 2020. Multi-objective scheduling strategy for scientific workflows in cloud environment: A firefly-based approach. Appl. Soft Comput. 106411.

[16]Kondikoppa, P., Chiu, C.-H., Cui, C., Xue, L., Park, S.-J., 2012. Network-aware scheduling of mapreduce framework ondistributed clusters over high speed networks. In: Proceedings of the 2012 Workshop on Cloud Services, Federation, and the 8th Open Cirrus Summit. ACM, pp. 39–44.

[17]Yuan, H., Bi, J., Tan, W., Li, B.H., 2016. CAWSAC: Cost-aware workload scheduling and admission control for distributed cloud data centers. IEEE Trans. Autom. Sci.Eng. 13 (2), 976–985.

[18]Yuan, H., Bi, J., Tan, W., Li, B.H., 2017a. Temporal task scheduling with constrained service delay for profit maximization in hybrid clouds. IEEE Trans. Autom. Sci.Eng. 14 (1), 337–348.

[19]Yuan, H., Bi, J., Zhou, M., Sedraoui, K., 2018. WARM: Workload-aware multiapplication task scheduling for revenue maximization in SDN-based cloud data center. IEEE Access 6, 645–657.

[20] Keshanchi B, Souri A, Navimipour NJ (2017) An improved genetic algorithm for task scheduling in the cloud environments using the priority queues: formal verification, simulation, and statistical testing. J Syst Softw 124:1–21

[21]Wang G-G, Gao D, Pedrycz W. Solving multiobjective fuzzy job-shop scheduling problem by a hybrid adaptive differential evolution algorithm. IEEE Trans Industr Inf 2022;18(12):8519–28.

[22] Tang, Zhiqing; Jia, Weijia; Zhou, Xiaojie; Yang, Wenmian and You, Yongjian (2020).

[23] Representation and Reinforcement Learning for Task Scheduling in Edge Computing. IEEE Transactions on Big Data, 1–1.

[24] Asghari, Ali; Sohrabi, Mohammad Karim and Yaghmaee, Farzin (2020). Online scheduling of dependent tasks of cloud workflows to enhance resource utilization and reduce the makespan using multiple reinforcement learning-based agents. Soft Computing.

[25] Orhean AI, Pop F, Raicu I (2018) New scheduling approach using reinforcement learning for heterogeneous distributed systems. J Parallel Distrib Comput 117:292–302

[26] Liu N, Li Z, Xu J, Xu Z, Lin S, Qiu Q, Tang J, Wang Y (2017) A hierarchical framework of cloud resource allocation and power management using deep reinforcement learning. In: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). IEEE, pp 372–382

[27] Zhang Yu, Yao J, Guan H (2018) Intelligent cloud resource management with deep reinforcement learning. IEEE Cloud Comput 4(6):60–69

[28] Lage-Freitas, A., Parlavantzas, N., & Pazat, J. (2017). Cloud resource management driven by profit augmentation. *Concurrency and Computation: Practice and Experience*, 29(4), e3899.

[29] Parikh, S. M., Patel, N. M., & Prajapati, H. B. (2017). Resource management in cloud computing: classification and taxonomy. *arXiv preprint arXiv:1703.00374*.

[30] Swain, S. R., Singh, A. K., & Lee, C. N. (2022). Efficient resource management in cloud environment. *arXiv preprint arXiv:2207.12085*.

[31] Le, G., Xu, K., & Song, J. (2013). Dynamic resource provisioning and scheduling with deadline constraint in elastic cloud. *IEEE ICSS*, 113-117.

[32] Fan, G., Chen, X., & Zhang, Y. (2022). An energy-efficient dynamic scheduling method of deadline-constrained workflows in a cloud environment. *IEEE Trans Netw Serv Manage* (in press).