JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

A Deep Reinforcement Learning Based Framework for Task Scheduling for Enhancing Efficiency in Cloud Computing

Ravi Prasad Ravuri
Application Developer, Sriven Technologies, Ashburn, VA, USA

Abstract

In this article, we proposed a learning based methodology for task scheduling in cloud computing towards leveraging performance of cloud infrastructure. It also enables consumers to have their Service Level Agreements (SLAs) satisfied with resource optimization. Deep Reinforcement Learning (DRL) technique is used in the proposed methodology for making scheduling decisions. Towards this end an algorithm known as Intelligent Learning based Task Scheduling (ILTS)is proposed and implemented. The proposed algorithm exploits agent based iterative approach that makes use of action-space, state-space and reward function to make well informed scheduling decisions. With the action-feedback iterations, the algorithm can make accurate scheduling decisions that improve energy efficiency and improve Quality of Service (QoS) in execution of jobs in cloud. DRL involves state transition in each stage and there is update of its Q-table through underlying deep neural network. We made experiments with different workloads. Our empirical study has revealed that the proposed system is better than existing methods in terms of success rate, energy efficiency and execution time.

Keywords – Task Scheduling, Service Level Agreements, Cloud Computing, Machine Learning, Deep Reinforcement Learning

1. INTRODUCTION

Cloud computing has emerged to offer solutions to real world problems with its scalable computing resources. Due to plethora of benefits, organizations are increasingly using cloud infrastructure for storage and computing. As a result, number of users of given cloud platform has experienced exponential growth. In presence of Service Level Agreements (SLAs) between consumers and service provider, there is need for optimization of cloud infrastructure usage to bring equilibrium between service provider and users in terms of utility and satisfaction [2]. In presence of dynamic characteristics of user workloads (jobs), proper scheduling is indispensable to ensure energy efficient scheduling and meet Quality of Service (QoS) requirements of users. Many existing heuristics based methods suffer from mediocre performance in task scheduling as the environment is highly dynamic and heterogeneous in nature. Artificial Intelligence (AI) has emerged to solve problems in many real world applications [3].

Many contributions were found in the literature on task scheduling in cloud. Hongjia et al. [2] explored different DRL applications, corresponding frameworks and underlying implementations. Zhaolonget al. [5] considered 5G enabled vehicular network integrated with IoT. They explored DRL based approach to control traffic. Ji et al. [7] exploited DRL technique for MEC environment. They proposed a methodology for computational offloading and the offloading decisions are made based on DRL. Junfeng et al. [8] proposed a methodology based on regional resource scheduling. This technique has potential to deal with diversified workloads. However, their methodology is based on DRL technique in edge computing scenario. Ali et al. [9] considered different workflows that are scientific in nature. Those workflows re scheduled in such a way that their method shows better performance. From the review of literature, it is understood that there is need for more efficient DRL based approach for efficient task scheduling. Our contributions in this paper are as follows.

1. We proposed a learning based methodology for task scheduling in cloud computing towards leveraging performance of cloud infrastructure. It also enables consumers to have their Service Level Agreements (SLAs) satisfied with resource optimization.
2. An algorithm known as Intelligent Learning based Task Scheduling (ILTS) is proposed and implemented. The proposed algorithm exploits agent based iterative approach that makes use of action-space, state-space and reward function to make well informed scheduling decisions.
3. A prototype is built to evaluate proposed methodology and ILTS algorithm.

The remainder of the paper is structured as follows. Section 2 reviews related works. Section 3 presents the methodology of our system. Section 4 presents results of our study. Section 5 concludes our work and gives scope for future work.

2. RELATED WORK

This section presents review of literature on existing scheduling methods that are based on deep learning. Mingxi et al. [1] proposed a DRL based methodology for task scheduling. It could leverage performance due to consideration of runtime situations in making decisions. Hongjia et al. [2] explored different DRL applications, corresponding frameworks and underlying implementations. Ding et al. [3] focused on Q-learning process which is crucial in making good decisions at runtime in terms of dynamic scheduling. Their approach was found to be energy efficient and could cater to dynamically changing workloads. Qu et al. [4] focused on edge-cloud where cloud resources and edge resources are integrated. Their method makes use of edge cloud resources in order to improve latency. They built an offloading method for efficient scheduling of jobs. Zhaolonget al. [5] considered 5G enabled vehicular network integrated with IoT. They explored DRL based approach to control traffic.

Jiechao et al. [6] used a methodology based on machine learning in order to predict workloads in cloud. It is important because of the prediction of workload; the system can have its planning for proper scheduling of jobs. Ji et al. [7] exploited DRL technique for MEC environment. They proposed a methodology for computational offloading and the offloading decisions are made based on DRL. Junfeng et al. [8] proposed a methodology based on regional resource scheduling. This technique has potential to deal with diversified workloads. However, their methodology is based on DRL technique in edge computing scenario. Ali et al. [9] considered different workflows that are scientific in nature. Those workflows re scheduled in such a way that their method achieves both load balancing and efficient resource provisioning. Zhaolong et al. [10] proposed a DRL based technique to deal with traffic associated with smart vehicular networks. Jun et al. [11] investigated UAV networks and used RL towards cluster task scheduling.

Wenhan et al. [12] performed offloaing of tasks to edge cloud in the presence of Mobile Cloud Computing (MCC). It was done based on DRL and offloading of computations. Zhiyuan et al. [13] studied on the energy efficiency in resource allocation in cloud environment using deep learning models. Ning et al. [14] considered intelligent resource application approach in presence of blockchain technology for both security and privacy preserving actions. It is based on the RL enabled approach. Mushu et al. [15] considered vehicular networks and enabled them with DRL technique in order to have collaborative computing in edge cloud environment. This was done towards more efficient usage of vehicular networks. Ying et al. [16] proposed a methodology for DRL based resource allocation in cloud with dynamically changing environment in place. They also investigated on edge resource. From the review of literature, it is understood that there is need for more efficient DRL based approach for efficient task scheduling.

3. PROPOSED TASK SCHEDULING SYSTEM

We proposed a novel scheduling approach based on DRL which has potential to improve performance of cloud infrastructure. It also benefits cloud consumers in terms of meeting SLAs and getting improved Quality of Service (QoS). The following sub sections provide more details about our proposed task scheduling system.

3.1 Problem Definition

Provided number of user jobs arrived at cloud infrastructure, efficient scheduling to see that the jobs are executed with efficiency in terms of energy conservation and meeting QoS needs of users is the main problem considered in this research. Since cloud infrastructure gets several thousands of jobs form multiple users, allocating resources to those jobs and ensure reduction in consumption of energy through efficient scheduling is given paramount importance. Therefore, this research is aimed at building a deep learning based framework for energy-efficient and QoS-aware task scheduling in cloud environment. Success of the proposed framework is measured in terms of energy efficiency and meeting QoS requirements.

3.2 System Model

Our system model includes a cloud platform, data centre with number of physical machines and VMs. There are number of users or cloud consumers from whom jobs arrive from time to time. Such jobs are to be scheduled efficiently in such a way that it saves energy of cloud infrastructure besides improving QoS for consumers. It also conserves cloud resources with improved utilization. The workload of cloud users contains tasks with different requirements that are to be considered by the proposed system. In other words, our system has to satisfy users' QoS requirements. Therefore, energy efficiency and meeting SLAs is given importance in our system model which is illustrated in Figure 1. The novelty of our system is that it exploits learning based approach at runtime instead of using traditional heuristics based methods. In the wake of emerging AI to solve problems in different fields, our system makes use of AI based runtime assessment to make scheduling decisions. While making decisions, our system considers a server, which consumes less energy and meets desired SLAs, for scheduling.

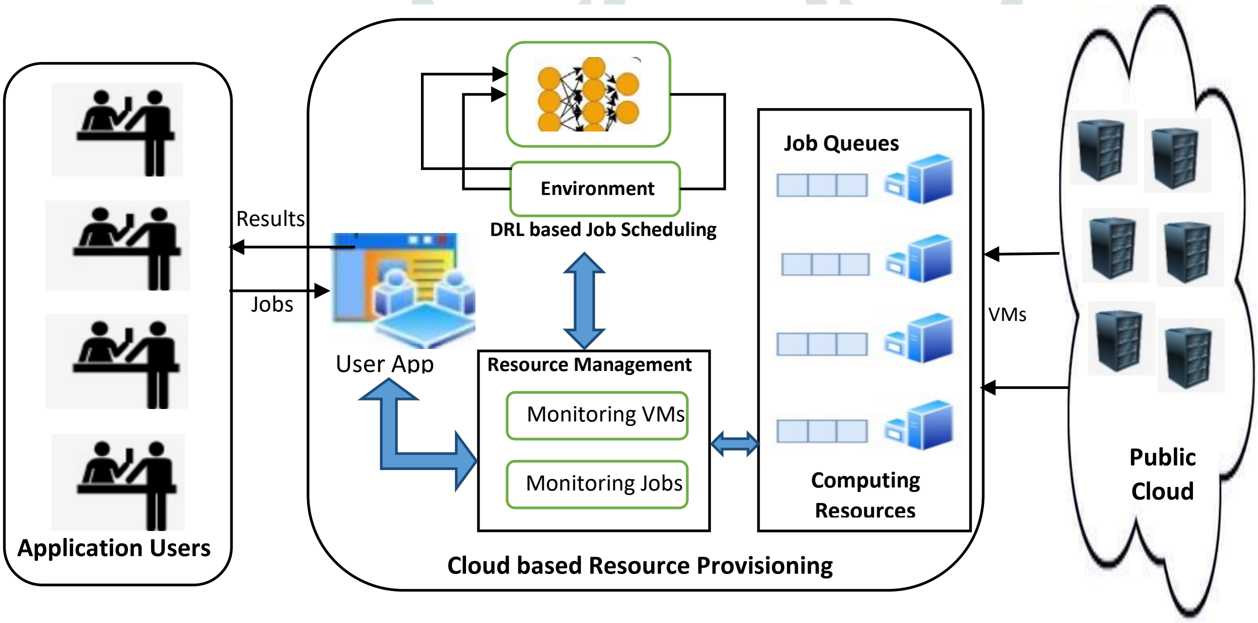

Figure 1: Our system model for DRL based task scheduling

End users are connected with the system through an application. They submit jobs with different requirements. The jobs are to be scheduled in cloud infrastructure with energy efficiency and meeting QoS needs. In presence of several hundreds of servers and underlying VMs in cloud infrastructure the scheduling process is carried out with efficiency. Resource management module continuously monitors VMs and jobs being executed in computing resources. It has provision to interact with user application from which jobs arrive and also computing resources. The jobs are given priority and as per it they are scheduled. However, scheduling decisions are made by DRL based job scheduling module which considers runtime situation and feedback on actions towards optimization in decision making. The system model is designed to improve efficiency in scheduling in terms of energy conservation in cloud infrastructure and improving QoS for end users.

3.3 Reinforcement Learning

RL is widely used in ML for intelligent decision making. RL exploits an agent to learn from training data. The training data, unlike supervised learning, has not labels for decision making. Agent need to explore training data and runtime conditions to make an intelligent decision. At every time step t, the state S_t is observed by the agent and it makes an action A_t accordingly. Then the state is transitioned from St to S_{t+1}. Agent is then given a reward denoted as R_t. This process is known as RL. We exploited deep learning for realizing this leading DRL.

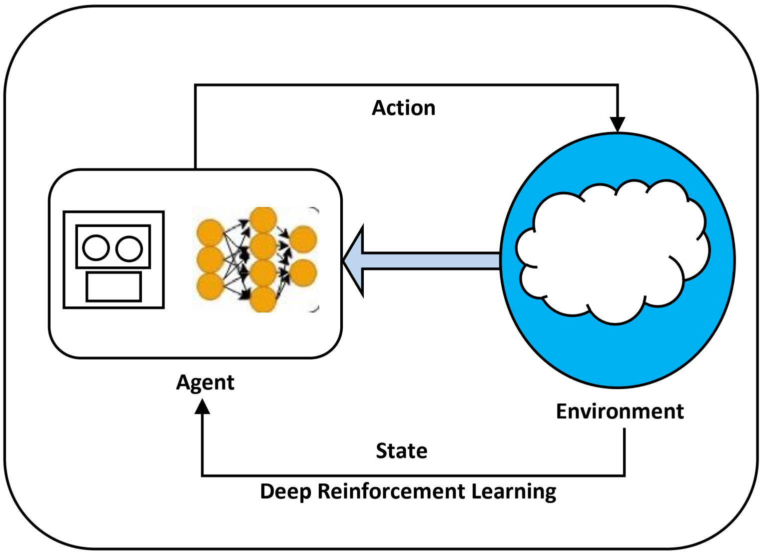

Figure 2: Illustrates the process involved in DRL

Agent is supposed to improve decision making from time to time for optimization in energy and QoS. The crucial aspect in RL is that action of agent is based on specific state that helps in maximum reward. Based on its action, reward is given either positively or negatively. The reward function plays an important role in problem solving. Therefore, RL has its goal to maximize reward over time as expressed in Eq. 1.

$$\mathbb{E}[\sum_{t=0}^{\infty} \gamma^{t} r_t] (1)$$

In RL, agent's decision reflects probability distribution π(a | s). with given environment space S, there is action space associated with specific policy. In many real world problems, probability distribution is very large and generally an approximator is found to be feasible to manage such probability distribution. Approximator is a function that deals with many parameters and they can be tuned as per policy as expressed in Eq. 2.

$$\pi(a \mid s) (2)$$

In RL problems, deep learning models are used to play the role of an approxmator as discussed in [12]. Often Q learning is used as RL model exploiting deep learning. It has provision to explore for best action based on the current state. Agent maintains a value function denoted as Q(s, a) for each action-state pair. The value function assists the agent to determine Q-values for each action. In the long run, agent uses Q values towards achieving maximum reward for its action. Q-table is maintained to hold Q-values. DRL is found suitable for intelligent decision making in gaming and other domains. The value function is updated after completion of each step as expressed in Eq. 3.

$$Q_{t+1}(s_{t,}a_t) = Q_t(s_{t,}a_t) + \alpha(r_{t+1,}\gamma_{a_{t+1}}^{max} Q_t(s_{t+1,}a_{t+1}) - Q_t(s_{t,}a_t)) \qquad (3)$$

Here the learning rate is set to some value denoted by α. Our aim in this paper to reduce energy consumption by data centres. The proposed system schedules jobs in public cloud in presence of unpredictable characteristics of jobs with varies size of jobs and time of arrival. These constrained are fixed and they vary for different jobs and RL is expected to achieve best scheduling results in presence of sub dynamic jobs. RL has potential to learn from the runtime system without prior knowledge to arrive at intelligent scheduling decisions.

3.4 Task Scheduling

Deep Q learning is used in order to make decisions on scheduling. Every job is scheduled by DRL agent to a best VM depending on Q-values generated by deep neural networks. Based on the reward function, agent makes decisions on scheduling given job to an appropriate VM meeting QoS requirements of the job. The reward function and an iterative process involved in DRL helps agent to make an optimized decision. In our system model, which is based on public cloud infrastructure model, jobs are very dynamic in nature and decisions on scheduling are to be made without prior knowledge. Each job is different from other jobs and to be scheduled to a specific VM. When a VM is running a job, it is considered busy and any job to be assigned to this needs to be in queue. When a job job_i is assigned to a VM, its wait time , T_i^{wait} is equal to the actual execution time of already running job job_{i-1}.

There is an action space associated with DRL based system. For a given job, DRL assigns it to a resource named VM_j, in the given action space expressed in Eq. 4.

$$A=[VM\_id_{1,}VM\_id_{2,}VM\_id_{3,}\ldots..VM\_id_M] \quad (4)$$

Here the total number of VMs is denoted as M. In the same fashion, there is state space in the proposed system. It is the space explored by agent to make decisions. Based on the runtime information available, agent makes an action that strives to ensure integrity of the action. Therefore, state space should be very informative to help agent in making intelligent decisions. In our system, the state space is expressed as in Eq. 5.

$$S=[Job\_size_{i,}arrival\_Time_{i,}QoS_{i,}T_1^{wait},T_2^{wait},T_3^{wait},.\ \ldots,T_M^{wait}](5)$$

State space is large as it needs to accommodate different possible states. In our system, reward function is another crucial aspect used to estimate reward for agent's actions. Agent tries to choose a state that renders highest reward towards improving accuracy in decision making. The best reward is the one which maximizes energy efficiency and improves QoS. Reward function is computed as in Eq. 6 and Eq. 7.

$$T_i^r=QoS_i/T_i(6)$$

$$s_i^r=1/T_i^{EXE}.E_i(7)$$

The resultant s_i^r and T_i^r is used to compute final reward that satisfies QoS needs. Based on the Eq. 6 and Eq. 7, reward function can be formulated as in Eq. 8.

$$R_i=T_i^r s_i^r(8)$$

In essence, our system is based on DRL with underlying deep neural network which generates states that are exploited by agent in making intelligent decisions based on rewards for actions. Agent takes several episodes to arrive at accurate decisions.

3.5 Proposed Algorithm

We proposed an algorithm, as shown in Algorithm 1, known as Intelligent Learning based Task Scheduling (ILTS). An agent, according to our algorithm, explores state and makes an action. For every action, it receives a reward based on values on Q-table. In every step, Q-table gets updated. The state transition is maintained in memory space and it has its capacity. Two deep networks are used in our system. They are known as evaluation and target networks respectively. Their structure is same but parameters differ. Target network is used to generate Q-table in each step. As explored in [16], evaluation network's parameters are copied to target network in every step. Every time, best case that considers energy efficiency maximization is considered for decision making.

Algorithm: Intelligent Learning based Task Scheduling (ILTS)

1. For each episode e in E
2. Set environment to initial state
3. Choose an action a_j randomly
4. Schedule job j
5. Update state
6. Update reward
7. IF j ≥ t and j ≡ 0 mod f Then
8. IF j ≡ 0 mod η Then
9. Reset Qˆ= Q
10. End If
11. Choose samples S_Δ
12. For each transition
13. Compute target
14. Update reward
15. End For
16. Make scheduling decision
17. End If
18. End If

Algorithm 1: Intelligent Learning based Task Scheduling (ILTS)

As presented in the proposed algorithm, it is observed that an agent has an iterative process as part of DRL to study action-space, state-space and reward function in making well informed decisions. The action taken by algorithm is given response in the form of reward. Then, the state transition takes pace. This process continues for each job to be scheduling until there is convergence which makes good decision that improves energy efficiency, success rate and also takes less execution time.

4. RESULTS AND DISCUSSION

This section presents experimental results of the proposed system compared with different existing methods. The observations are made in terms of execution time for different workloads, success rate and energy efficiency dynamics.

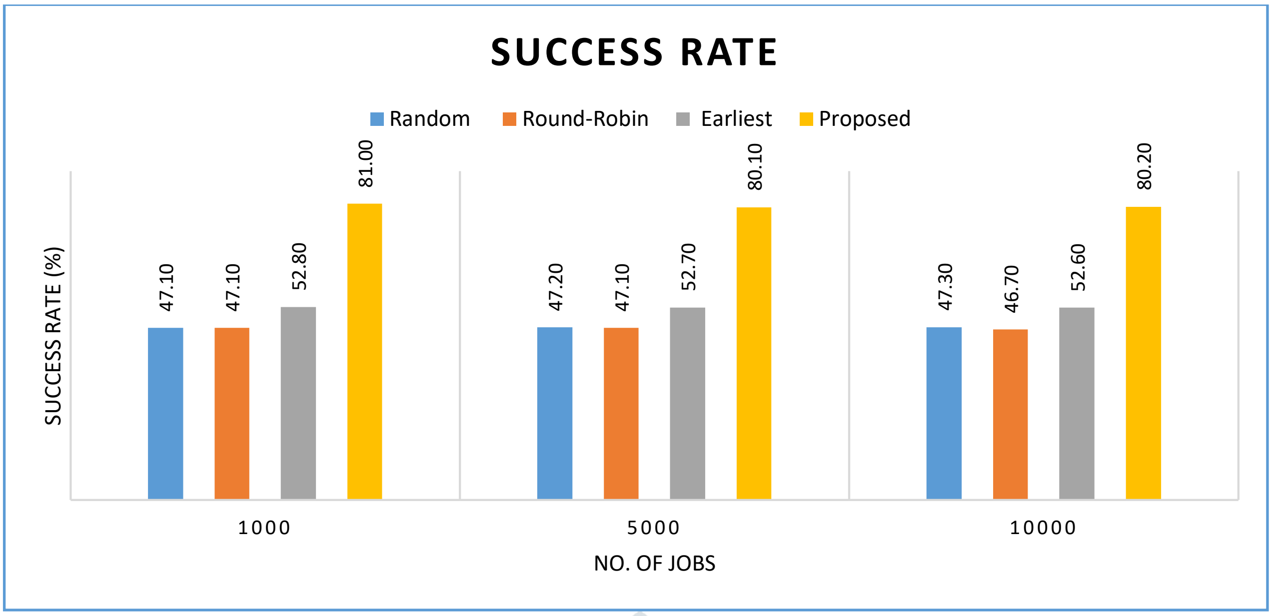

Figure 3: Shows performance comparison in terms of success rate

As presented in Figure 3, it is observed that different workloads are used in the experiments. When workload has 10000 jobs, Random method showed 47.30% success rate, Round-Robin 46.70%, Earliest 52.60% and the proposed method exhibited 80.20% success rate. Therefore, the experimental results revealed that the proposed method has highest success rate in efficient scheduling of different workloads.

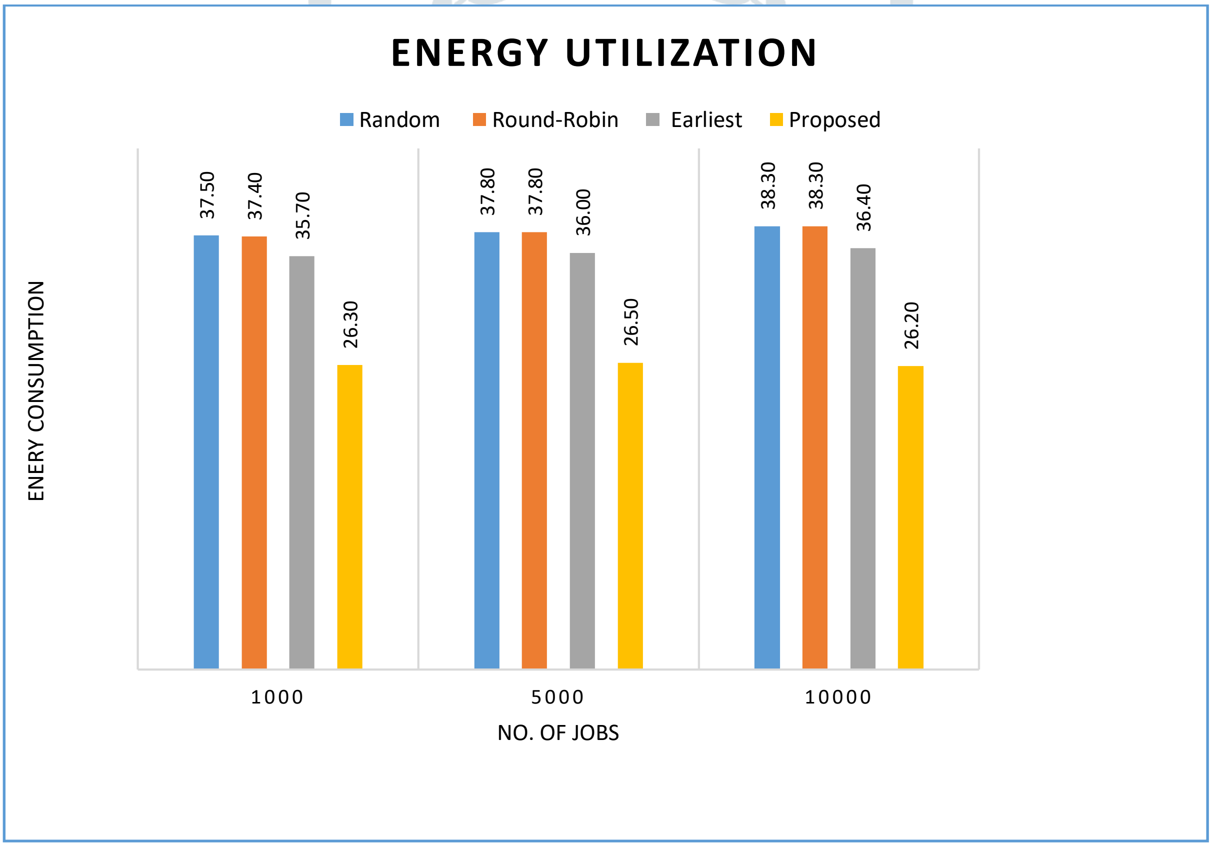

Figure 4: Shows performance comparison in terms of energy consumption

As presented in Figure 4, it is observed that different workloads are used in the experiments. When workload has 10000 jobs, Random method showed 38.30 energy consumption, Round-Robin 38.30, Earliest 36.40 and the proposed method exhibited 26/20energy consumption. Therefore, the experimental results revealed that the proposed method has highest level of energy conservation in efficient scheduling of different workloads.

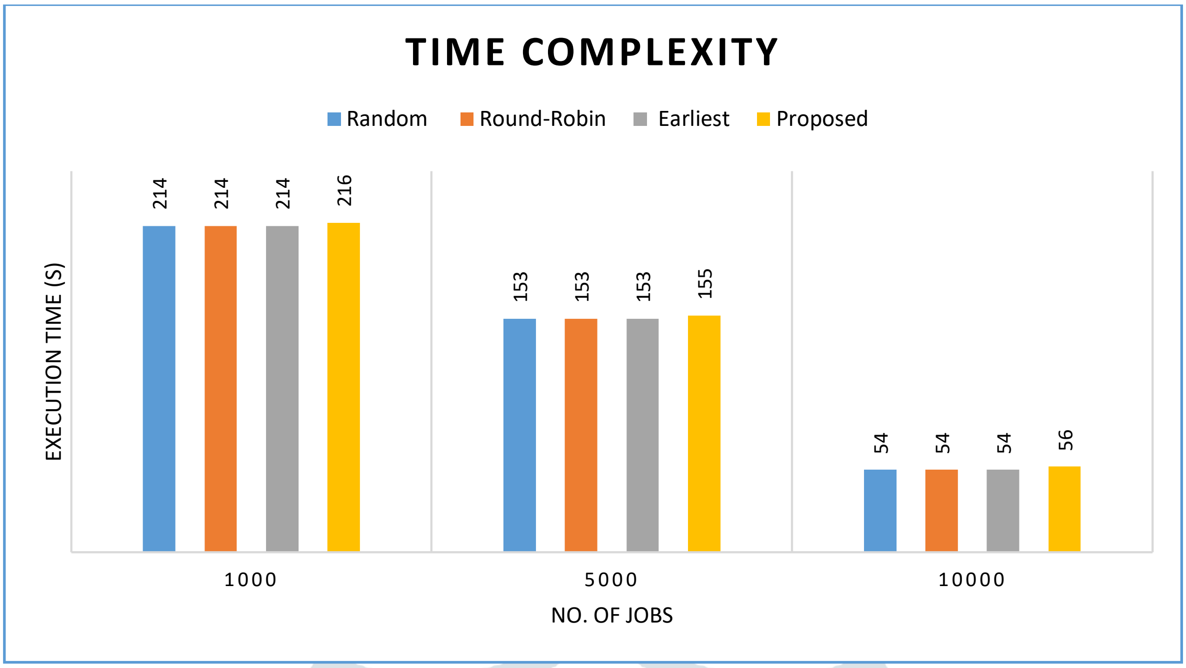

Figure 5: Shows performance comparison in terms of execution time

As presented in Figure 4, it is observed that different workloads are used in the experiments. When workload has 10000 jobs, Random method showed 54 secondsexecution time, Round-Robin 54, Earliest 54 and the proposed method exhibited 56seconds. Therefore, the experimental results revealed that the proposed method has efficiency in execution time, though it takes relatively little bit more time over existing methods, even in presence of complex DRL operations.

5. CONCLUSION AND FUTURE WORK

We proposed a learning based methodology for task scheduling in cloud computing towards leveraging performance of cloud infrastructure. It also enables consumers to have their Service Level Agreements (SLAs) satisfied with resource optimization. Deep Reinforcement Learning (DRL) technique is used in the proposed methodology for making scheduling decisions. Towards this end an algorithm known as Intelligent Learning based Task Scheduling (ILTS) is proposed and implemented. The algorithm has an iterative process in which underlying DRL network plays crucial role in understanding runtime situation using state-space and produce action-space based on reward function. Based on the reward or feedback for every action, the algorithm eventually makes best decision in task scheduling. The proposed algorithm is compared with many existing methods such as Random, Round-Robin and Earliest. Experiments results showed that the success rate of proposed method is 80.20%, energy utilization % is 26.20 and execution time is 56 seconds when workload as 1000 jobs. This performance is better than all existing methods. In future we intend to improve our method further with optimizing parameters involved in deep learning.

References

[1] Cheng, Mingxi; Li, Ji and Nazarian, Shahin (2018). 23rd Asia and South Pacific Design Automation Conference (ASP-DAC) - DRL-cloud: Deep reinforcement learning-based resource provisioning and task scheduling for cloud service providers, 129–134. http://doi:10.1109/ASPDAC.2018.8297294.

[2] Li, Hongjia; Wei, Tianshu; Ren, Ao; Zhu, Qi and Wang, Yanzhi (2017). IEEE/ACM International Conference on Computer-Aided Design (ICCAD) - Deep reinforcement learning: Framework, applications, and embedded implementations: Invited paper, 847–854. http://doi:10.1109/ICCAD.2017.8203866.

[3] Ding, Ding; Fan, Xiaocong; Zhao, Yihuan; Kang, Kaixuan; Yin, Qian and Zeng, Jing (2020). Q-learning based dynamic task scheduling for energy-efficient cloud computing. Future Generation Computer Systems, S0167739X19313858–. http://doi:10.1016/j.future.2020.02.018.

[4] GuanjinQu;HuamingWu;Ruidong Li and Pengfei Jiao; (2021). DMRO: A Deep Meta Reinforcement Learning-Based Task Offloading Framework for Edge-Cloud Computing . IEEE Transactions on Network and Service Management. http://doi:10.1109/tnsm.2021.3087258.

[5] Ning, Zhaolong; Kwok, Ricky Y. K.; Zhang, Kaiyuan; Wang, Xiaojie; Obaidat, Mohammad S.; Guo, Lei; Hu, Xiping; Hu, Bin; Guo, Yi and Sadoun, Balqies (2020). Joint Computing and Caching in 5G-Envisioned Internet of Vehicles: A Deep Reinforcement Learning-Based Traffic Control System. IEEE Transactions on Intelligent Transportation Systems, 1–12. http://doi:10.1109/TITS.2020.2970276.

[6] Gao, Jiechao; Wang, Haoyu and Shen, Haiying (2020). 29th International Conference on Computer Communications and Networks (ICCCN) - Machine Learning Based Workload Prediction in Cloud Computing, 1–9. http://doi:10.1109/ICCCN49398.2020.9209730.

[7] Li, Ji; Gao, Hui; Lv, Tiejun and Lu, Yueming (2018). IEEE Wireless Communications and Networking Conference (WCNC) - Deep reinforcement learning based computation offloading and resource allocation for MEC, 1–6. http://doi:10.1109/WCNC.2018.8377343.

[8] Zou, Junfeng; Hao, Tongbo; Yu, Chen and Jin, Hai (2020). A3C-DO: A Regional Resource Scheduling Framework based on Deep Reinforcement Learning in Edge Scenario. IEEE Transactions on Computers, 1–1. http://doi:10.1109/TC.2020.2987567.

[9] Asghari, Ali; Sohrabi, Mohammad Karim and Yaghmaee, Farzin (2020). Task scheduling, resource provisioning, and load balancing on scientific workflows using parallel SARSA reinforcement learning agents and genetic algorithm. The Journal of Supercomputing. http://doi:10.1007/s11227-020-03364-1.

[10] Ning, Zhaolong; Dong, Peiran; Wang, Xiaojie; Guo, Lei; Rodrigues, Joel J. P. C.; Kong, Xiangjie; Huang, Jun and Kwok, Ricky Y. K. (2019). Deep Reinforcement Learning for Intelligent Internet of Vehicles: An Energy-Efficient Computational Offloading Scheme. IEEE Transactions on Cognitive Communications and Networking, 1–1. http://doi:10.1109/TCCN.2019.2930521.

[11] Yang, Jun; You, Xinghui; Wu, Gaoxiang; Hassan, Mohammad Mehedi; Almogren, Ahmad and Guna, Joze (2019). Application of reinforcement learning in UAV cluster task scheduling. Future Generation Computer Systems, 95, 140–148. http://doi:10.1016/j.future.2018.11.014.

[12] Zhan, Wenhan; Luo, Chunbo; Wang, Jin; Wang, Chao; Min, Geyong; Duan, Hancong and Zhu, Qingxin (2020). Deep Reinforcement Learning-Based Offloading Scheduling for Vehicular Edge Computing. IEEE Internet of Things Journal, 1–1. http://doi:10.1109/JIOT.2020.2978830.

[13] Xu, Zhiyuan; Wang, Yanzhi; Tang, Jian; Wang, Jing and Gursoy, Mustafa Cenk (2017) IEEE International Conference on Communications (ICC) - A deep reinforcement learning based framework for power-efficient resource allocation in cloud RANs, 1–6. http://doi:10.1109/ICC.2017.7997286.

[14] Zhaolong Ning;ShoumingSun;XiaojieWang;LeiGuo;GuoyinWang;Xinbo Gao and Ricky Y. K. Kwok; (2021). Intelligent resource allocation in mobile blockchain for privacy and security transactions: a deep reinforcement learning based approach . Science China Information Sciences. http://doi:10.1007/s11432-020-3125-y.

[15] Li, Mushu; Gao, Jie; Zhao, Lian and Shen, Xuemin (2020). Deep Reinforcement Learning for Collaborative Edge Computing in Vehicular Networks. IEEE Transactions on Cognitive Communications and Networking, 1–1. http://doi:10.1109/TCCN.2020.3003036.

[16] Chen, Ying; Liu, Zhiyong; Zhang, Yongchao; Wu, Yuan; Chen, Xin and Zhao, Lian (2020). Deep Reinforcement Learning based Dynamic Resource Management for Mobile Edge Computing in Industrial Internet of Things. IEEE Transactions on Industrial Informatics, 1–1. http://doi:10.1109/TII.2020.3028963.