USC Viterbi EE 454 Final Project Phase III, Part I: Academic Papers Survey of Machine Learning Task Scheduling Algorithms

Stephen Hornyak (1075791723), Victor Hui (3051128112), and Joshua Williams (8139286920)

***Abstract*—Task scheduling is an important part of the OS of an SoC. It impacts the order that tasks are scheduled onto processing units, which has a profound impact on the success of the device. Machine learning algorithms have been utilized to provided optimization to this task scheduling problem. This paper provides a summary and comparison of three such implementations of machine learning optimizations for the task scheduling problem.**

***Keywords*—machine learning, task scheduling, optimization, trade-off, deep reinforcement learning (DRL), deep Q-network (DQN).**

I. Introduction

A computer system's processing cores are designed to execute the instructions of a given task. However, a scheduling algorithm is needed to first schedule a given task onto a given processing element. Figure 1 shows a summary of how tasks are given to a scheduling algorithm and assigned to a VM, or processing element, that has specific CPU and memory resources available. Scheduling algorithms come in many different forms and are each designed to provide different optimizations and trade-offs. The scheduling algorithm is directly responsible for an SoC's quality of service, with parameters including resource utilization, energy consumption, operating cost, and execution time. A scheduling algorithm must be used to assign tasks to the processing elements, and a

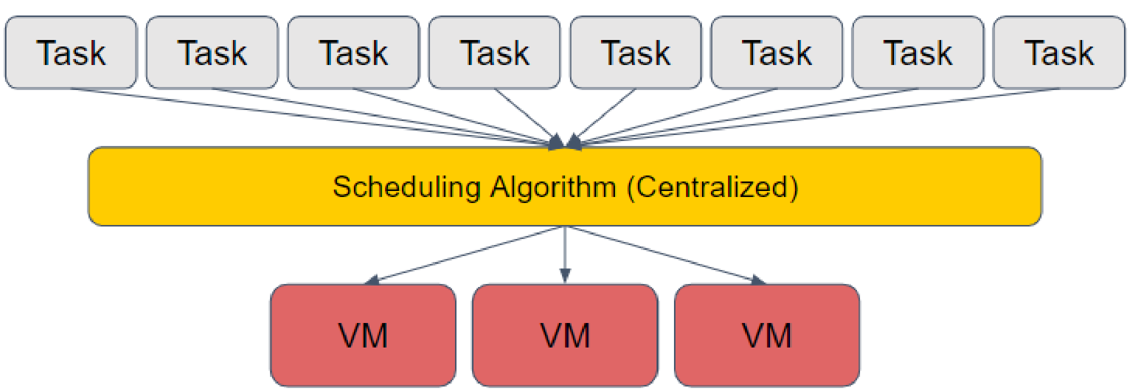


Fig. 1. Diagram of tasks being assigned to VMs using a centralized scheduling algorithm.

specific scheduling algorithm must be chosen to optimize required parameters while sacrificing others.

To best optimize task scheduling, AI techniques and machine earning methods have been applied to successfully schedule tasks and allow for proper optimization in many areas, from resource utilization, energy consumption, scalability, cost of operation, and execution time. Along with these many factors to optimize, there are also a great amount of input that needs to be considered into determining the most optimal technique to apply which would result in the most optimized parameter. Since machine learning algorithms can be trained to learn how to best utilize a given set of inputs, they are an appropriate choice for optimizing the scheduling of variable task inputs. This paper will explore a small subset of methods that utilize different machine learning algorithms to optimization specific parameters of the task scheduling problem. Advantages and drawbacks will be presented for each method surveyed, as well as direct comparisons between the methods.

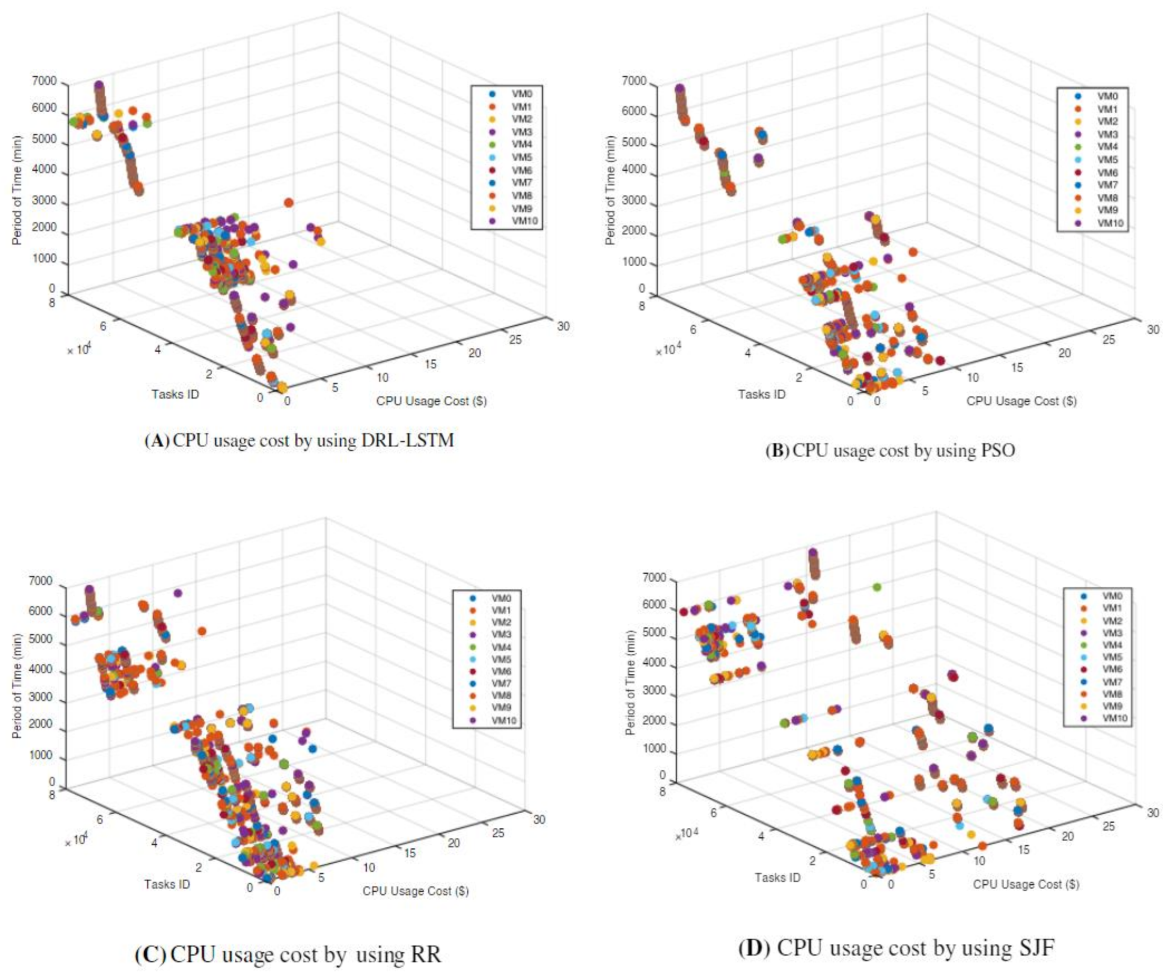
II. DRL-LSTM

The article "Deep and Reinforcement Learning for Automated Task Scheduling in Large-Scale Cloud Computing Systems," by Rjoub Gaith, Jamal Bentahar, Omar Abdel Wahab, and Ahmed Saleh Bataineh, provides four different methods of using machine learning (ML) to optimize task scheduling in cloud computing [1]. The goal of the article is to provide better alternatives to the task scheduling problem to branch away from more traditional methods of task scheduling, such as Round Robin (RR) and Shortest Job First (SJF) [1].

According to Gaith et al., the current field of task scheduling has two main approaches: traditional and intelligent approaches [1]. Traditional approaches focus on implementing and further modifying conventional scheduling methods, such as RR and SJF. However, the downfall to the traditional approach occurs when optimizing parameters, as the traditional approach can only optimize a limited number of parameters [1]. This does not bode well for cloud computing due to the many different parameters that must be optimized simultaneously, such as bandwidth, memory, and CPU costs [1]. Gaith et al. also describes in high-level how cloud computing is performed, as there are a certain number of VMs that perform the tasks and have a certain number of given resources to that VM. (1) Therefore, the question that is discussed in the article is how to ensure that the task execution performance is maximized while the resource cost is minimized.

Gaith et al. explore four different methods that will be tested for optimizing the cloud scheduling task, all of which are deep or reinforcement learning-based scheduling approaches: reinforcement learning (RL), deep Q-networks (DQN), recurrent neural network long short-term memory (RNN-LSTM), and deep reinforcement learning combined with LSTM (DRL-LSTM) [1]. Each of these methods provides different ways of exploring the inputs provide the optimized output. RL learns how to recognize and respond to changes to its environment [1]. Deep Q-Networks combine Q-learning and deep learning to maximize reward by exploring and exploiting the solution space [1]. RNN-LSTM uses learning to predict what the next state will be given the current inputs, and it uses the prediction to optimize the current state [1]. DRL-LSTM learns from previous states to predict future states where tasks must be scheduled to optimize rewards [1]. As Gaith et al. explored the four different methods, their overall goal was to optimize CPU and RAM usage to minimize cost [1].

After testing the four different machine learning methods, Gaith et al. determined that the most efficient approach was the DRL-LSTM method; it provided the scheduling mechanism with the most accurate prediction of which VM should receive the incoming task [1]. Their research found that the DRL-LSTM method minimizes CPU utilization cost by 67% compared to SJF and by 35% compared to RR and particle swarm optimization (PSO) approaches [1]. Along with a CPU cost minimization, RAM utilization cost was also minimized, with improvements as great as 72% compared to SJF, 65% compared to RR, and 31.25% compared to PSO [1]. These CPU and RAM improvements are visualized in Figure 2 and Figure 3. However, Gaith et al. achieved these results with a considerable drawback in the computation time of the DRL-LSTM method. The LSTM layer must check the full history of the states, requiring significant processing times to explore the full solution space [1]. As a potential solution of this considerable drawback, Gaith et al. proposes using distributed and federated learning techniques to reduce this overhead.



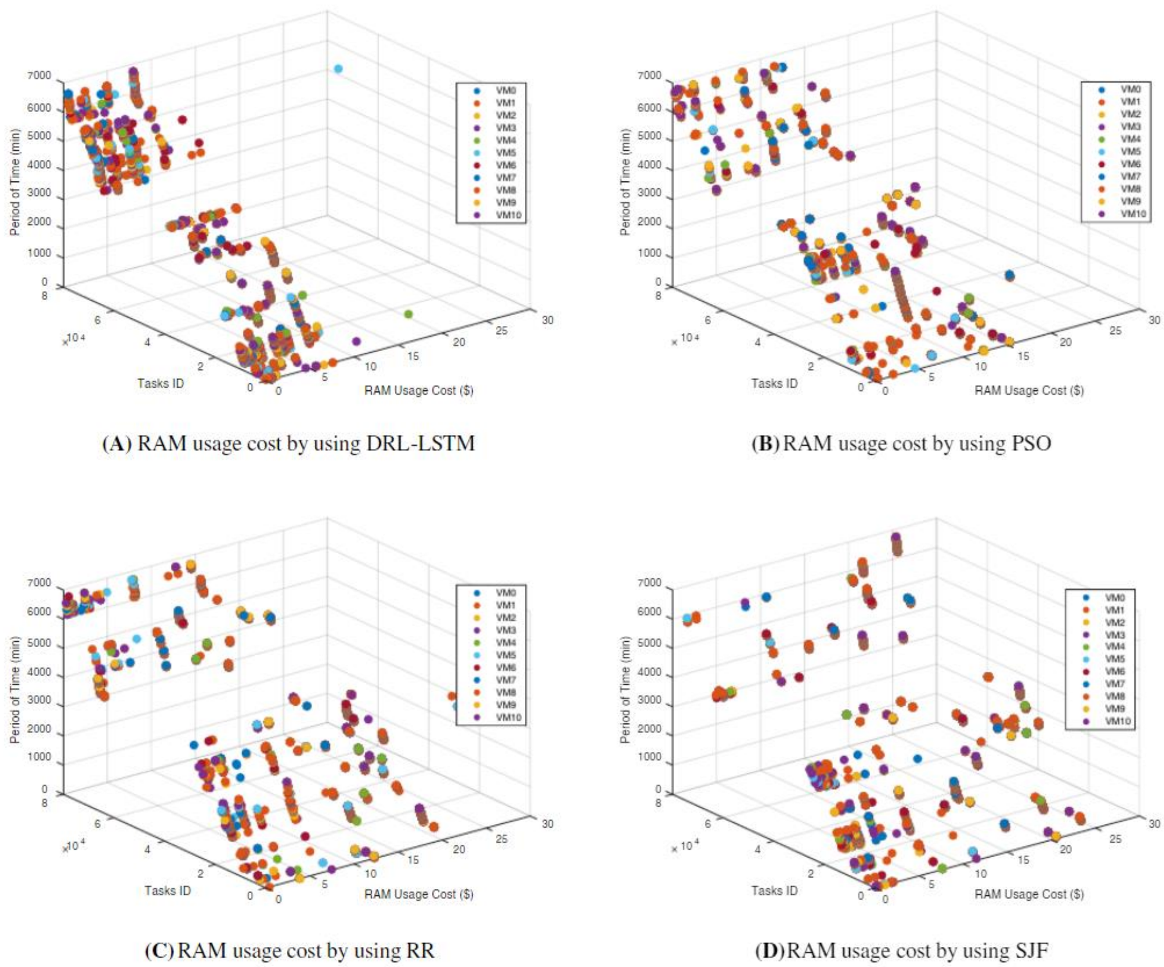
(A) CPU usage cost by using DRL-LSTM

(B) CPU usage cost by using PSO

(C) CPU usage cost by using RR

(D) CPU usage cost by using SJF

Fig. 2. CPU cost of the DRL-LSTM algorithm compared to traditional task scheduling algorithms [1].



(A) RAM usage cost by using DRL-LSTM

(B) RAM usage cost by using PSO

(C) RAM usage cost by using RR

(D) RAM usage cost by using SJF

Fig. 3. RAM cost of the DRL-LSTM algorithm compared to traditional task scheduling algorithms [1].

Furthermore, they argue that the distributed learning models do not account for the reliability of the VMs. They note that adding the trust and performance of each VM as a metric can further optimize their solution to cloud computing task scheduling [1].

III. DRL-CLOUD

The second paper, “DRL-cloud: Deep reinforcement learning-based resource provisioning and task scheduling for cloud service providers,” by Mingxi Cheng, Ji Li, and Shahin Nazarian, further discusses the topic of cloud computing and task scheduling in terms of cloud service providers (CSP) [2]. In this article, Cheng et al. discuss a novel DRL, Resource Provisioning (RP) and Task Scheduling (TS) system. The parameters that are being optimized in this article are energy cost for the large-scale CSPs with extremely large number of servers that receive vast amounts of user requests per day. The DRL-cloud technique uses target network and experience replay techniques to provide parallelization and improve runtime, while it applies DRL to RP and TS through Semi-Markov Decision Process (SMDP) formulation to learn how to best optimize cost [2].

DRL-Cloud can provide considerable runtime improvements without sacrificing optimization. The method, when compared to FERPTS, another currently implemented efficient algorithm, achieves a maximum energy cost efficiency improvement of 320% and maintains a low rejection rate [2]. In terms of runtime, the DRL-Cloud method achieves a 144% runtime reduction, 218% energy cost efficiency improvement and a 249% lower reject rate as compared to a RR baseline [2].

Cheng et al. also discuss the difficulties that appear in terms of reducing energy and electric cost. The difficulty is primarily related to the scalability of expenditure control due the large size of server farms and the vast amounts of incoming requests [2]. A further difficulty is that the incoming request patterns are variable and changing in both the long term and short term. To solve the scalability issue of the cloud servers, Cheng et al. has proposed a DRL-based system to combat the scalability issue along with helping with the adaptability of the RP and TS system [2]. The solution is broken into two main sections: a sorting mechanism that accepts the user request and then passes onto the appropriate job and task ready queue, and a DRL-based two-stage RP-TS processor that optimizes the energy cost and converges the results based on training techniques in deep Q-learning [2]. Figure 4 provides a diagram of the overall DRL-Cloud system. Through these techniques, DRL-Cloud achieves a much more optimized solution for the cost and execution time parameters compared to a RR baseline and a state-of-the-art method such as FERPTS [2].

IV. H20-Cloud

The final paper, "H2O-Cloud: A Resource and Quality of Service-Aware Task Scheduling Framework for Warehouse-Scale Data Centers - A Hierarchical Hybrid DRL (Deep Reinforcement Learning) based Approach," by Mingxi Cheng, Ji Li, Paul Bogdan, and Shahin Nazarian, discusses the topic of task scheduling while optimizing the Quality of Service (QoS) and energy usage of data centers through machine learning [3]. Due to the large appeal of cloud computing to outsource computing and storage needs, the data centers that store these resources consume a vast amount of power and thus place a huge barrier in terms of energy consumption. The article details two primary areas where data centers consume the most power: during sub-capacity times and during idle times [3]. In both conditions, the servers of the data center continue to operate with no

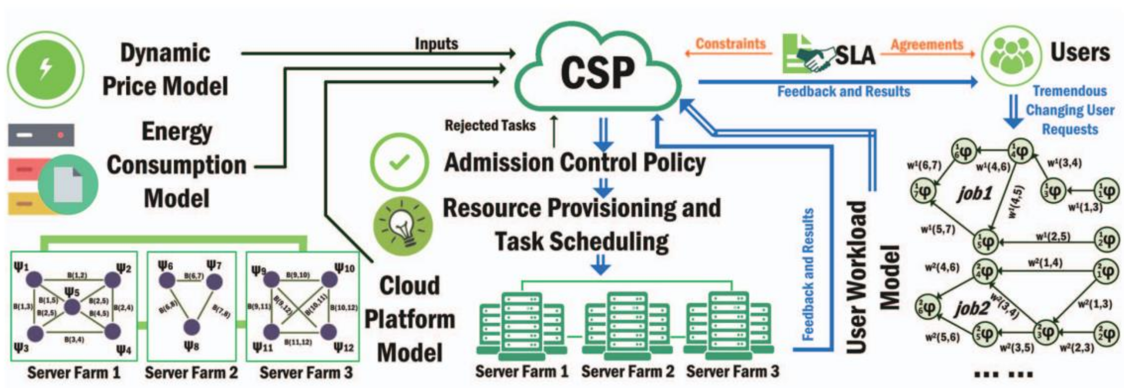


Fig. 4. Diagram of the DRL-Cloud system [2].

despite having no major tasks to service, thus consuming energy resources while performing no computation. Along with attempting to optimize the energy consumption resources, Cheng et al. are also attempting to maintain QoS as a parameter of measurement, which is defined as the combination of the task rejection rate, deadline violation rate, and the reward rate, which is defined as the rate that the priority requirement of the user-designated task is if fulfilled [3].

To achieve this optimization, Cheng et al. propose a task scheduler based on deep reinforcement learning (DRL) as it is most effective against problems with a high dimension state space and a limited action space [3]. The solution provided in the paper is defined as a Hierarchical and Hybrid Online (H2O) task scheduler. The hierarchical portion is used to deal with the enormous action space and separates them into further sub-action spaces. Each task will be assigned to the sub-action by one Deep Q-Network (DQN) [3]. This improves the efficiency by allowing many DQNs to work and schedule tasks faster than one large DQN, while also improving the decision quality [3]. An incoming task is scheduled to a server farm, which is a collection of servers, and a pricing model determines which server would be best suited to perform the task while optimizing the cost [3]. The model can be improved by adding more DQN layers or more information inputs [3]. The hybrid portion of the method is applied when the DRL generates invalid actions in some corner cases, acting as a backup method to be implemented in the case that the DRL method fails [3]. As a result, the H2O method maintains the robustness of the framework and simultaneously achieves high performance in the resource utilization metric [3]. Compared to baselines of RR and DRL-Cloud task scheduling algorithms, the H2O-Cloud method improves resource efficiency and QoS significantly [3]. The areas where H2O-Cloud performed the best compared to the baseline methods was the large-scale configuration and high variance scenarios, decisively optimizing the fields of soft-deadline violation rate and both energy efficiency and cost efficiency. H2O-Cloud gained an improvement by 201.17% in the energy cost efficiency field, 47.88% in the energy efficiency field and 551.76% in the reward rate field [3].

V. Comparison of the Approaches

As is seen by the techniques achieved by the articles previously mentioned, each of the prevailing methods have certain drawbacks and advantages when using them for task scheduling. In Gaith et al., the DRL-LSTM model is great for optimizing the execution performance of each individual VM; however, the drawback is the significant execution time of the method [1]. The second method, DRL-Cloud, is great for being scalable and being able to implement in many different sizes of cloud computing; however, it has the drawback of not having an optimal runtime with smaller data sets [2]. The third technique, H2O-Cloud, allows for great savings in terms of energy consumption but has the drawback of not being optimized for resource consumption. Instead, it is geared specifically towards maintaining QoS and improving the energy consumption of large cloud providers [3]. The three methods are summarized in Table 1.

From the combination of drawbacks and advantages of the three methods, a specific method can be selected to best fit the needs

	DRL Scheduling [1]	DRL-Cloud [2]	H2O-Cloud [3]
Algorithm Type	Machine Learning	Machine Learning	Machine Learning
Learning Type	• Reinforcement Learning (RL) • Deep Q-Networks (DQN) • Recurrent Neural Network with Long Short-term Memory (RNN-LSTM) • Deep Reinforcement Learning with LSTM (DRL-LSTM)	• Novel Deep Reinforcement Learning	• Deep Reinforcement Learning • Deep Q-Network
Metrics of Optimization	• CPU usage cost • RAM usage cost	• Energy cost efficiency • Rejection rate • Runtime	• Energy efficiency • Cost reduction • Reward rate
Runtime	Worst	Better	Best
Data/ Tool Used	Google cluster dataset	Google cluster usage traces	Google cluster usage traces
Tradeoff	Longer Execution Time	Not optimal with smaller data sets	Not exceptional in low variance, small sets of data
Published	2020	2018	2019
Journal	Concurrency and Computation	ASP-DAC	TCAD

Table 1. Summary of the three task scheduling approaches surveyed.

of a task scheduling situation. If the data is smaller and one of the most important optimization parameters is resource utilization, then the DRL-LSTM is best suited. If the data fluctuates from large to sets to huge sets of data or requests and energy consumption is a concern and a high parameter to be optimized, then DRL-Cloud is the more appropriate method to use. Finally, if the tasks need both to consume less energy and to guarantee that priorities are met in terms of the deadlines and execution time, then H2O-Cloud would produce a more optimal solution.

VI. Discussion: Selecting a Task Scheduling Algorithm

This paper only presents a small subset of the many different algorithms for task scheduling. As is often with engineering problems, there is no one best task scheduling algorithm. Every situation is different, and a task scheduling algorithm should be chosen to best fit a specific situation. Some considerations when choosing a task scheduling algorithm for a problem are the inputs to the problem, what must be optimized to solve the problem, and what performance metrics can be sacrificed while solving the problem.

VII. Conclusion

The task scheduling problem is one of the many fields in engineering that where machine learning is being used to find improvements that were not previously possible. As task scheduling algorithms improve, their optimizations will provide improvements to operating systems that can be used for a wide range of applications, giving engineers more options to design systems that best fit consumer needs.

References

[1] G. Rjoub, J. Bentahar, O. Abdel Wahab, and A. Saleh Bataineh, "Deep and reinforcement learning for automated task scheduling in large - scale cloud computing systems," *Concurrency and Computation*, vol. 33, no. 23, 2021, doi: 10.1002/cpe.5919.

[2] M. Cheng, J. Li, and S. Nazarian, "DRL-cloud: Deep reinforcement learning-based resource provisioning and task scheduling for cloud service providers," in *2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2018, vol. 2018, pp. 129–134, doi: 10.1109/ASPDAC.2018.8297294.

[3] M. Cheng, J. Li, P. Bogdan, and S. Nazarian, "H2O-Cloud: A Resource and Quality of Service-Aware Task Scheduling Framework for Warehouse-Scale Data Centers," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 2019, vol. 39, no. 10, pp. 2925-2937, doi: 10.1109/TCAD.2019.2930575