Resource Scheduling for Offline Cloud Computing Using Deep Reinforcement Learning

Hatem M. El-Boghdadi[1†] and Rabie A. Ramadan[2††],

Islamic University of Madinah, Saudi Arabia Hail University, Saudi Arabia

Summary
Many organizations around the world use cloud computing for their services. Cloud Computing is mainly based on the concept of on-demand delivery of computations, storage, applications, and other resources. It depends on delivering users services through Internet connectivity. It also uses a pay-as-you-go business model to handle users' services. It has some essential characteristics including on-demand service, resource pooling, rapid elasticity, virtualization, and measured services. At the same time, there are different types of virtualization such as full virtualization, para-virtualization, emulation, OS virtualization, and application virtualization. Resource scheduling in the cloud computing is one of the most challenging jobs where resources have to be allocated to the required tasks/jobs according to the required Quality of Services (QoS) of the cloud applications. Due to the cloud environment, uncertainty, and maybe heterogeneity, resource allocation cannot be addressed with the existing policies. The problem still a major concern of most of the cloud providers where they face troubles in selecting the appropriate resource scheduling algorithm for a specific workload, especially the workload might be dynamic. In this paper, we use one of the Artificial Intelligence (AI) emergent algorithms, Deep Reinforcement Learning, (Deep Reinforcement Learning for Cloud Scheduling (DRLCS)), to solve the problem of resource scheduling in cloud computing.

Keywords:
Cloud computing, Scheduling, Artificial Intelligence, reinforcement learning.

1. Introduction

Cloud computing is an emergent technology that is used by most of the current organizations. Cloud Computing is mainly based on the concept of on-demand delivery of computations, storage, applications, and other resources. It depends on delivering services to users through Internet connectivity. It also uses a pay-as-you-go business model to handle users' services. However, the development of cloud computing is facing number of challenges including security and scheduling. Figure 1 show a cloud basic architecture proposed by Madni et al. [11] that consists of three main layers including the data centers, virtual machines, and the user layer. In the data center layer resides all of the hardware and storage as well as the operating systems. In the virtual machines layer, the cloud administrator as well the automated systems are capable of creating different virtual machines with different operating systems that could be different from the host operating system. The third layer contains the cloud users and their jobs and tasks to be submitted to the cloud.

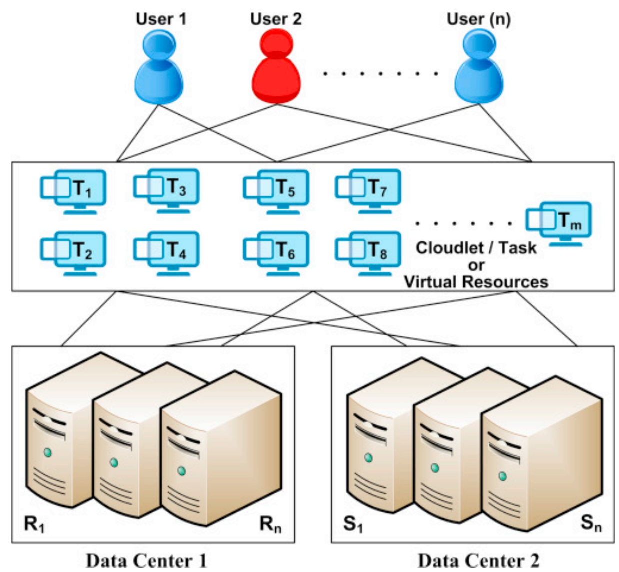


Fig. 1 Cloud Basic architecture (Madni et al., 2016)

A cloud has to be able to efficiently schedule the resources according to user requests. The scheduling problem should consider different user inputs such as deadlines, performance issues, execution cost, transmission cost, energy efficiency, Load Balancing, and Makespan. In addition, it is important to take Service Level Agreements (SLAs) with users into account during the scheduling process. Moreover, during the execution process, there is a possibility of a resource to go offline or becoming invalid or there is a delay due to network congestion or latency. A full reschedule or partial schedule could be required for customer satisfaction. Nevertheless, tasks dependency is required to be taken into consideration.
A good scheduler is the one can adapt to the environmental change and the cloud load. In addition, the scheduler needs to efficiently utilize the cloud resources and maintains a high quality of services. On the other hand, Artificial Intelligence (AI) is an emergent technology that has been used in many applications including military, transportation, networks, and many other fields. AI involves many techniques and algorithms such as case-based reasoning,

Manuscript received April 5, 2019
Manuscript revised April 20, 2019

rule-based systems, artificial neural networks, fuzzy models, genetic algorithms, cellular automata, multi-agent systems, swarm intelligence, reinforcement learning and hybrid systems. In addition, one of the important algorithms that made a great change in the field of AI is the deep learning. Although deep learning is not a new algorithm it has been recently reused differently to solve many of the hard problems.

This paper uses the deep learning to solve the cloud scheduling problem. Specifically, we use the reinforcement learning to solve the cloud scheduling problem. We introduce Deep Reinforcement Learning for Cloud Scheduling (DRLCS).

The next section presents the related work. In section 3, we introduce the concepts of reinforcement learning. Section 4 proposes the new scheduling model. In section 5, we present our experimental results. Finally, Section 6 makes some concluding remarks.

2. Literature Review

Cloud scheduling attracted a large number of researchers. In this section, we review the most recent clouding scheduling solutions. The authors of [1] proposed an approach for task scheduling algorithm. They proposed an idea based on load balancing in cloud computing. The algorithm is based on two levels, and the target was not only to meet the user's requirements but also to satisfy the high quality of resource utilization. In [2], the authors proposed an algorithm for cloud scheduling problem based on a combination between genetic and simulated annealing. The algorithm considered different parameters such as the QoS requirements like completion time, bandwidth, cost, distance, the reliability of different type tasks. A hieratical scheduling algorithm is proposed in [3] where users' Service Level Agreement (SLA) is achieved in minimum time. The authors used high priority jobs first. The priority here is based on the jobs deadline where the completion time is estimated by the algorithm based on the available resources. The authors of [4] proposed what is called Activity Based Costing (ABC) algorithm. The main idea is to assign a priority for each task along with the cost.

Similarly, paper [5] presents transaction intensive cost constraint cloud workflow scheduling algorithm. The algorithm considers both execution cost and time as the two key parameters. It tries to minimize the cost of giving the task deadline. The authors of [6] utilized the CloudSim simulator implementing a new VM Load Balancing Algorithm. They try to assign the best VM for the required tasks. The work in [8] also utilized the CloudSim for cloud schedule using the basic algorithm OS like FCFS, Priority Scheduling and Shortest Job First.

Ant colony optimization is also used to solve the cloud scheduling problem [7]. The main idea is to let the randomized optimization search allocate the incoming jobs to the available VMs, and the positive feedback leads the next assignments. In 2018, the authors of [16] proposed a hybrid GA-PSO algorithm for cloud resource scheduling. They take into consideration some of the critical parameters such as makespan, cost, and load balancing.

Recently, in 2017 and 2018, there are a large number of algorithms are proposed to solve the cloud resource scheduling problem. For instance, in [13], the authors proposed a heuristic approach that combines the modified analytic hierarchy process (MAHP), bandwidth aware divisible scheduling (BATS) + BAR optimization, longest expected processing time preemption (LEPT), and divide-and-conquer methods to perform task scheduling and resource allocation. In [12], another research article, the authors proposed what is called “crowd-funding” while idle resources are collected from a pool of cloud resources. Then, they proposed a genetic algorithm to allocate the resources based on the crowd-funding findings.

In [14], the authors proposed a PerfGreen as a dynamic cloud resource scheduling with the main concern is saving the cloud energy. Therefore, Perfgreen is energy-aware as well application placement technique that is based on a heuristic approach. In [15], the same authors provided a quantitative analysis of virtualization overheads for two hypervisor-based (XEN, KVM) and two OS-based (LXC, Docker) platforms.

More algorithms and techniques are summarized in the recent surveys indicating the state-of-the-art are presented in [9] and [10]. These two articles show that the cloud scheduling problem still needs more investigation. In addition, they show that AI techniques could be a suitable solution. One of the recent algorithms used for resource scheduling the DeepRM [18]. The DeepRM tries to solve the generic resource scheduling problem using reinforcement learning. We utilize a modified version of the DeepRM to fit the offline cloud scheduling problem.

3. Deep Reinforcement Learning

Reinforcement learning, as shown in Figure 2, is based on the concepts of agents, environment, states, actions, and rewards. The agent is responsible for taking the actions. The actions are the set of all possible moves that the agent can make. However, the agent has to choose from among a set of possible actions. There is also what is called a “discount factor”, that is multiplied by a future reward for reducing the effect of the accumulated rewards on the agent actions. The environment is the world that its borders restrict the agent. The environment input is the agent current state and its action. It returns the agent reward and next state. The state is the immediate configuration that agent discovers or what is returned from the environment. The reward is the

feedback by which the success or failure of an agent's actions is measured. There are other terms such as:
Policy (π): it is considered as the strategy that the agent follows to determine its next action based on the current state.
Value (V): it is the expected long-term return of the current state under policy π.
Q-value or action-value (Q): it is similar to the value (V) except it considers the current action. It maps the state and action to rewards.
Trajectory: it is just the sequence of states and actions that influence the states.

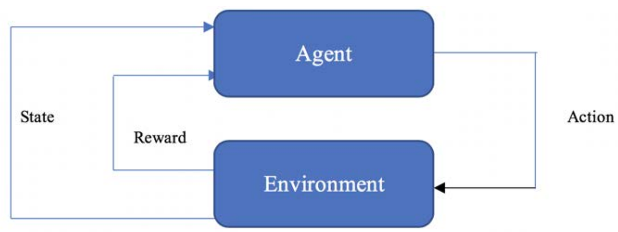


Fig. 2 Reinforcement learning architecture

The generic objective function of the reinforcement learning could be as follows:

$$\sum_{t=0}^{t=\infty} \gamma^t r(x(t), a(t))$$

It is basically the sum of the reward function r over the time steps t. Also, x represents the state at a given time step, a is the state action, and r is the reward function for state x and action a. In this case Neural networks could be used as the agent that learns to map state-action pairs to rewards.

Convolutional neural networks have been used in many applications to recognize an agent's state. Certainly, the performance of the neural networks is based on finding the right coefficients, or weights. In this paper, we follow the footsteps of DeepRM2 [19] in designing the deep reinforcement learning. The change made in DeepRM2 was basically in convolutional neural networks structure as shown in Table 1.

Table 1: DeepRM2 convolutional neural network structure

Layer name	Output size	DeepRM2
Conv1	20x224	5x5, 8, stride 1, relu
Pool1	10x112	2x2 avg pool, stride 2
Conv2	10x112	5x5, 16, stride 1, relu
Pool2	5x56	2x2 avg pool, stride 2
FC1	1x1	72-d fc, tanh
FC2	1x1	11-d fc, softmax

4. Our Proposed Model

In cloud scheduling, there are different parameters to be considered such as the required CPU, memory, job deadline, VM load balancing. We follow the same approach used by DeepRM and DeepRM2 [18]. However, DeepRM and DeepRM2 considered only the CPU and memory parameters. Therefore, DeepRM and DeepRM2 are modified to fit the cloud scheduling problem. Let V be the number of available virtual machines with the following configurations:

U_x - the VM CPU,
M_x – the VM memory, and
S_x– the VM storage.

However, these resources are considered available in a pool or a cluster regardless their VM or location.

At the same time, we assume the resource profile for each job is *j*. The profile contains:
U_i - job *i* required CPU,
M_i - job *i* required memory,
T_i - job deadline, and
R_i - job expected running time.

The objectives of the proposed resource scheduling algorithm are:

1) Assign the given jobs to the appropriate VMs to minimize of the job completion time,
2) Satisfy jobs deadline, and
3) Minimize the slowdown jobs

5. Reinforcement Learning Representation

In this section, the problem of cloud scheduling is structured to fit the reinforcement learning representation. Reinforcement learning usually works on states. So, the cloud resources in this paper is formed as states in distinct images as shown in Figure 3. The cloud image shows the available cloud resources. Other images, job slots, are the required jobs to be scheduled and their resource requirements. Those jobs are arranged according to their time stamps. Therefore, since it is offline scheduling, jobs could be sorted according to their deadlines. Ideally, we assume that we have N jobs waiting to be scheduled at a certain point of time. The output of the scheduler is either the job Schedule, Postpone, Missed, or Rejected. The "Postpone" decision means that the current available resources do not satisfy the job requirements while "Missed" means that the scheduler will not be able to satisfy the job deadline. On the other hand, "Reject" decision means that there is no way to satisfy of the job due to the limitation of the cloud resources. In other word, the available cloud resources do not fit the requirement of the job.

Jobs are assumed preemptive where if a job started it is assumed not to stop till it is completed. Therefore, the required resources will not be available till the job finishes. The scheduler agent works on a job by job scheduling. Job schedule could be parallelized if the number of scheduled jobs, N, is minimized. However, we believe that the reward function would work better with sequential scheduling. The reward function is used to guide the scheduler toward better schedule based on the problem objectives. The reward function is considered as the controller of the convergence process.

Job	**1**	**2**	**3**	**4**
Deadline	5	10	9	11
Expected running time	3	1	5	1

Fig. 3 Example on state representation

6. Evaluation

In this section, we generate a set of jobs and cloud resources randomly; 80% of the generated data are used for training and the 20% are used for testing. We also implemented four different algorithms, Shortest Job First (SJF), Longest Job First (LJF), Tetries [20], and Random algorithms. In Shortest Job First (SJF), Longest Job First (LJF), and Tetries, there is no training phase. For each of these algorithms, we measure the discounted total reward and the average job slowdown. The discounted total reward is about 150 points. In addition, the average job slowdown is about 10% of the number of the jobs.

We generate jobs randomly and measure the performance. Also, the available resources are fixed during the multiple trails. Figures 4 and 5 show the results obtained. As can be seen, Figures 4 and 5 show the discount reward and the job slowdown values for Deep Reinforcement Learning for Cloud (DRLC) Scheduling algorithm during the testing phase. The figures show that the DRLC seems to perform well while the reward is high, and the slowdown is very low. It is also much better than the other three algorithms. However, Tetries slowdown values seem to be very high, even higher than the SJF and LJF.

Another performance measure is to evaluate the SDLC algorithm against an algorithm with two different extremes where in one case the whole cloud resources are available and the number of the jobs are either small or very high. We noticed that when small number of jobs are planned, DRLC is almost the same as other algorithms while on the other extreme, DRLC performs much better than the three other algorithms (SJF, LJF, and Tetries). In other words, when small jobs arrived for a schedule, the four scheduler preform almost the same where the number of the delayed jobs are the same. On other hand, increasing the number of jobs to be scheduled, the number of delayed jobs are much higher using the SJF, LJF, and Tetries than the ones delayed using DRLC as shown in Figure 6.

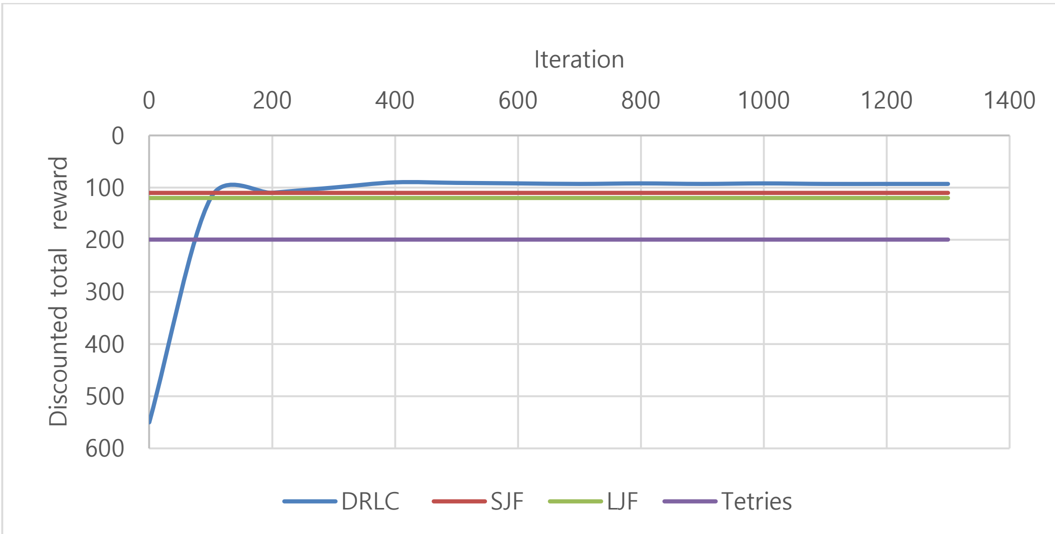


Fig. 4 Discounted total reward

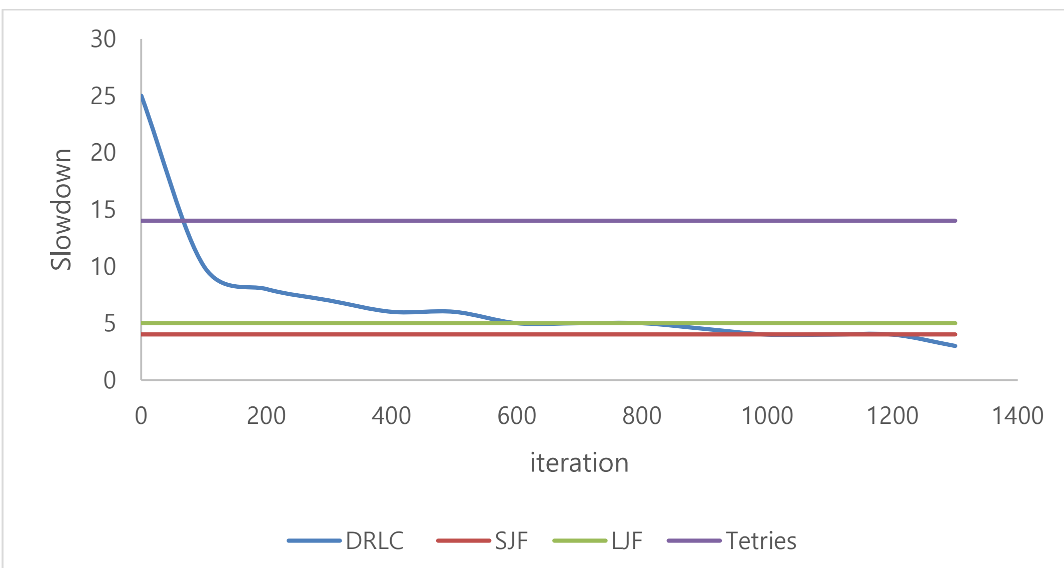
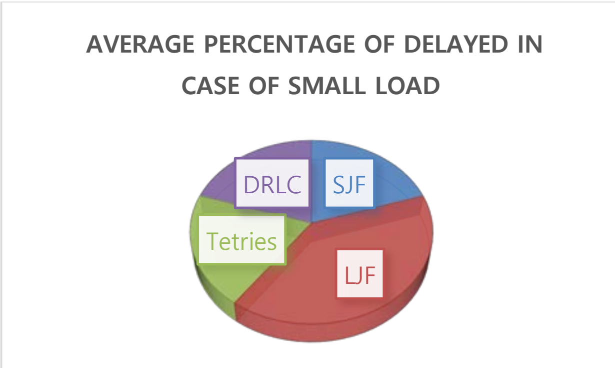
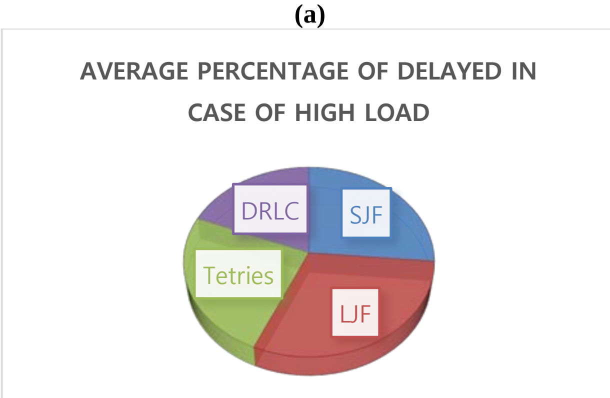


Fig. 5 Average Job Slowdown



(a)



(b)

Fig. 6 (a) Average small load performance, (b) average high load performance

7. Conclusions

In this paper, we introduced the deep reinforcement learning approach for offline cloud resource scheduling process. We extended the DeepRM and DeepRM2 to be used with more resources' configuration. With different sets of experiments, the proposed method showed a comparable performance over the regular methods. This is considered as a proof of concept that deep reinforcement learning could be used in similar optimization problems. Our future investigation will be on the utilization of reinforcement deep learning in other optimization problems such as routing problems.

Acknowledgment

This work is done under the grant received (400296-53/40) by Deanship of research at Islamic University of Madinah (IUM) for application- based research. We also give special thanks to the administration of IUM for their support in every aspect.

References

[1] Burya R Raman, R. Calheiros, R.N.(2009) "Modeling and Simulation of Scalable Cloud Environment and the Cloud Sim Toolkit: Challenges and Opportunities'', IEEE publication 2009,pp1-11.
[2] Sudha Sadhasivam, R. Jayarani, Dr. N. Nagaveni, R. Vasanth Ram "Design and Implementation of an efficient Twolevel Scheduler for Cloud Computing Environment" In Proceedings of International Conference on Advances in Recent Technologies in Communication and Computing, 2009.
[3] G. Guo-Ning and H. Ting-Lei, "Genetic Simulated Annealing Algorithm for Task Scheduling based on Cloud Computing Environment," In Proceedings of International Conference on Intelligent Computing and Integrated Systems, 2010, pp. 60-63.
[4] Rajkumar Rajavel , Mala T "Achieving Service Level Agreement in Cloud Environment using Job Prioritization in Hierarchical Scheduling" Proceeding of International Conference on Information System Design and Intelligent Application,2012 , vol 132, pp 547-554
[5] Q. Cao, W. Gong and Z. Wei, "An Optimized Algorithm for Task Scheduling Based On Activity Based Costing in Cloud Computing," In Proceedings of Third International Conference on Bioinformatics and Biomedical Engineering, 2009, pp. 1-3
[6] Y. Yang, Kelvin, J. chen, X. Lin, D.Yuan and H. Jin, "An Algorithm in Swin DeW-C for Scheduling Transaction Intensive Cost Constrained Cloud Workflow," In Proceedings of Fourth IEEE International Conference on eScience, 2008, pp. 374- 375
[7] Jasmin James, Dr. Bhupendra Verma "Efficient Vm Load Balancing Algorithm For A Cloud Computing Environment " In Proceeding of International Journal on Computer Science and Engineering (IJCSE) Vol. 4 No. 09, Sep 2012
[8] Medhat A. Tawfeek, Ashraf El-Sisi, Arabi E. keshk, Fawzy A. Torkey "Cloud Task Scheduling Based on Ant Colony Optimization" In Proceeding of IEEE International Conference on Computer Engineering & Systems (ICCES), 2013
[9] Shanthan H. A Survey of Algorithms for Scheduling in the Cloud. Int J Comput Sci Eng. 2018;(April).
[10] 1. Liu Y, Wang L, Wang XV, Xu X, Zhang L. Scheduling in cloud manufacturing: state-of-the-art and research challenges. Int J Prod Res. 2018;7543:1-26. doi:10.1080/00207543.2018.1449978
[11] Madni SHH, Latiff MSA, Coulibaly Y, Abdulhamid SM. Resource scheduling for infrastructure as a service (IaaS) in cloud computing: Challenges and opportunities. J Netw Comput Appl. 2016;68:173-200. doi:10.1016/j.jnca.2016.04.016
[12] Zhang N, Yang X, Zhang M, Sun Y, Long K. A genetic algorithm-based task scheduling for cloud resource crowd-funding model. Int J Commun Syst. 2018;31(1):1-10. doi:10.1002/dac.3394
[13] Gawali MB, Shinde SK. Task scheduling and resource allocation in cloud computing using a heuristic approach. 2018. doi:10.1186/s13677-018-0105-8
[14] S.K. Tesfatsion, E. Wadbro, and J. Tordsson. PerfGreen: Performance and Energy Aware Resource Provisioning for

Heterogeneous Clouds, Technical Report UMINF 18.04, 2018.
[15] S.K. Tesfatsion, C. Klein, and J. Tordsson. Virtualization Techniques Compared: Performance, Resource, and Power Usage Overheads in Clouds. 2018 ACM/SPEC International Conference on Performance Engineering (ICPE), to appear, 2018.
[16] Manasrah AM, Ali HB. Workflow Scheduling Using Hybrid GA-PSO Algorithm in Cloud Computing. Wirel Commun Mob Comput (4, EI, 1899). 2018;2018.
[17] Google. Deepmind. https://deepmind.com/. Published 2018. Accessed July 27, 2018.
[18] Mao, H., Alizadeh, M., Menache, I., & Kandula, S. (2016). Resource Management with Deep Reinforcement Learning. In 15th ACM Workshop on Hot Topics in Networks (pp. 50–56).
[19] Yufei Ye, Xiaoqin Ren, Jin Wang, Lingxiao Xu, Wenxia Guo, Wenqiang Huang and Wenhong Tian, " A New Approach for Resource Scheduling with Deep Reinforcement Learning, " Arvix Artificial Intelligence , 2018 .
[20] Robert Grandl , Ganesh Ananthanarayanan , Srikanth Kandula, Sriram Rao, and Aditya Akella , "Multi-Resource Packing for Cluster Schedulers," Proceedings of the 2014 ACM conference on SIGCOMM, Pages 455-466 ; 2014.

Hatem M. El-Boghdadi is a professor of Computer Engineering Department at Cairo University, Giza, Egypt since 2015. Currently, he is on leave at the Islamic University in Madinah, Saudi Arabia. His research interest include Parallel Architectures and Algorithms, Network-on-Chips, Reconfigurable Computing.

Rabie A. Ramadan is associated with the Computer Engineering Department at Cairo University, Giza, Egypt as an associate professor since 2009. Currently, he is on leave at the University of Hail, Saudi Arabia. His research interest is on IoT, Computational Intelligence and Big Data.