International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-6, August 2019

Diversified Quality Aware Ensemble Resource Scheduling (DQAERS) for IAAS with Massive Load of Tasks and Resources in Cloud Computing

B. Ravindra Babu, M. Veera Sekhar Rao

ABSTRACT--- Identifying a deterministic approach to perform resource scheduling in cloud computing is crucial requirement, which is since, the volume of the anomalies and the high dimensionality of the values projected to these anomalies observed during resource scheduling. The volume of tasks that evinces flash-crowd state at resource broker of the IAAS, and high dimensionality of the anomalies projected for resource quality factors are out of scope in regard to contemporary resource scheduling strategies contributed in recent past. Hence' the resource scheduling by contemporary methods in such conditions are insignificant as the resource scheduling optimality observed as probabilistic. In order to optimize the resource scheduling in the context of aforesaid properties high volume of tasks (flash-crowd state at resource broker) and high dimensional projection of anomalies, this manuscript derived an ensemble resource scheduling strategy, which fall in to the category of batch scheduling. The experimental study outlined that the proposal is most prominent and robust to deliver optimal resource scheduling in the context of anomalies of high volume and dimensionality that compared to the contemporary method.

Keywords: cloud computing (CC), resource scheduling (Resource Scheduling), virtual machines (VMs), VM migration, resource management (RM), QoS (Quality of Services).

I. INTRODUCTION

The technology cloud computing (CC) is innovative and virtualized, which deals with various services all over internet. The primary objectives are scalability, utilization monitor, high performance, manage, high availability, economical, provision efficiency and fault-tolerant. Hardware is compressed towards service, included fully and "hardware agnostic software stack" is delivered in the form of alternative for QoS (Quality of Services) [1]. From providers of cloud, huge amount of resources are required to allocate and assign to various cloud users, fairly, dynamically and most significantly in an effective way.

Revised Manuscript Received on August 30, 2019.
* Correspondence Author
B. Ravindra Babu*, Research Scholar, JNTUH, Hyderabad, Telangana, India.
(Email: Sekhar.muddana@gmail.com)
M. Veera Sekhar Rao, Research Scholar, JNTUH, Hyderabad, Telangana, India.
(Email: Bellam.ravindrababu@gmail.com)

In the perspective of cloud users, the economy of user is driven entity while making the decision for using the services of cloud. The important problems, which are interrelated commonly with the IaaS in CC, are data management, virtualization, security, resource management (RM) etc., [2], [3], [4], [5]. In the view of CC, the RM is the process of distributing network, nodes, computing, virtual machines (VMs) and storage towards applications set. It endeavors to attain the objectives of applications performance. The main objective of cloud providers is effective & efficient utilization of resource within the constraints of "SLAs (Service Level Agreements)" with the users of cloud. The effective utilization of resource is managed classically through virtualization methods that allow resources statistical multiplexing through requests & clients (Jennings and Stadler, 2014). The resource scheduling (Resource Scheduling) allocates the accurate & precise task towards network, storage and CPU. The objective of Resource Scheduling is extensive utilization of the resources. Nevertheless, finely organized scheduling is required for both the cloud users & providers [6]. The work [2] presents that the important challenging issue is Resource Scheduling for the IaaS in CC, is managing and providing effective resources utilization. The issue has achieved maximum focus from the groups of research in the past years. Therefore, the limited resources source, locality limitations, resources heterogeneity, and environmental pre-requisites. Hence, it is required to have effective & efficient Resource Scheduling process, which is appropriate for Cloud computing platforms.

II. RELATED WORK

The CC presents basic maintenance system and scalable RM with VMs. With the enhancement in the amount of VMs is being utilized, even though Resource Scheduling is made in an effective way, the maximum VM overhead declines the whole system. For addressing this problem, the scalable deduplication system file is planned in [7] with an aim of lessening the overhead performance at the time of Resource Scheduling. For addressing the issue of computational cost, the task scheduling based on QoS in the environment of cloud is planned in [8]. The work [9] presents the "Distributed load balancing during Resource Scheduling" is addressed by assisting VM migration through networking data.

The work [10] presents Resource Scheduling through the aware of data similarity computation in environment of cloud is examined. The work [11] presents the cuckoo driven model, which is designed in the applications of CC for consuming the prominent resources to assist the query associated operations. The work [12] presents the performance assessment of RM for allocating the resources as per the request of user.

The work [13] presents the allocation of resources utilizing Nash Equilibrium by implementing auction-based model. The work [14] presents the design of deterministic algorithm to allocate the task dynamically as user requested to enhance the allocation of resources in cloud environment. The work [15] presents the effective data structure for the RM is examined by deliberating the various job types and availability of resources. Nevertheless, computational cost at the time of Resource Scheduling remained to be unaddressed. The work [16] presents addressing of optimization of cost with heterogeneous VM instances aimed at scientific workflows.

In order to address these constraints, a cuckoo search-based optimal resource scheduling strategy for IAAS (ORS-CS) [17] has been devised in recent past of the contemporary literature. Though this contribution successfully addressed the most of constraints observed in other contemporary methods, the event of high volume of tasks reported to resource broker and high dimensionality of anomalies related to resource quality factors are still not in the scope.

In the examined manuscripts, they employ diverse techniques for Resource Scheduling; however, there were no variations in the quality factors of the resources which are available. Moreover, the flash crowd of the tasks and the dimensionality of the anomalies projected for the diversified quality factors of the resources are completely fall out of the scope in contemporary resource scheduling strategies. In regard to these observations, the contribution of this manuscript endeavored to define an optimal batch scheduling strategy that deals the flash crowd state of the tasks reported to resource broker and the high dimensionality of the anomalies projected to the quality factors of the resources. The suggested work in this manuscript considered both accessibility & Resource Scheduling utilizing “Diversified QoS metrics” to enable the optimal resource scheduling for IAAS of the cloud computing, which is explored in detail in following sections.

III. BATCH SCHEDULING FOR TASKS AND OPTIMAL RESOURCE PAIRING

This segment discusses the procedure of proposed deterministic batch scheduling algorithm that schedules virtual machines as optimal resources to multiple tasks at scheduler. The entire procedure contains the cluster depiction information overlapped scheduling intervals, clusters formation depicting the tasks arriving the scheduler at time interval and associates the clusters corresponding towards resources and tasks. Later, schedules the resources depicted in the cluster towards tasks represented in corresponding cluster. Further discussion elaborates each of these procedures incorporated in the represented method.

Clustering resources and Tasks

In respect to cluster the resources, the fitness function of clustering algorithm is built by using distance between schedule intervals of the respective resources. Each of the resultant cluster projects the set of resource having overlapped schedule intervals. Similarly, the tasks also clustered in to set of groups, such that each of the cluster projects set of tasks that are overlapping about time interval among arrival time & expiration time. The clustering procedure fitness function tends to cluster the tasks relies on the time interval among arrival & expiration time of respective tasks. Each of the clusters projected from resources represented by a resource as centroid that contains max schedule interval that overlaps with the schedule intervals of the other resources of the respective cluster. Likewise, each of the clusters formed from tasks represents by a task as centroid that contains max time interval between arrival and expiration time, which overlaps with the other tasks of the corresponding cluster in regard to time interval between arrival and expiration times of each task.

The cluster formation of the resources and tasks is continuous process. The resources in sequence that are having schedule intervals will be framed as a cluster, which is only if their schedule intervals are overlapped. Further, the depicted cluster will be refined by selecting the resource with maximum schedule interval as new centroid of that cluster and possessing the other resources in to corresponding cluster, which are overlapping with this centroid. In respective manner, the clusters also framed from tasks. The task that arrives first in sequence will be considered as initial centroid of the cluster and possess the other tasks, those are overlapping with the centroid of the corresponding cluster, which is in regard to the interval between arrival and expiration time each task. Further refines the cluster by selecting new centroid of the cluster, which is a task having maximum interval between arrival and expiration time that compared to the other tasks of the corresponding cluster. Then possess the other tasks in to the corresponding cluster those are overlapping with newly selected centroid, which is in regard to interval between arrival and expiration time. The cluster will be finalized if no new centroid is being selected, which is in both cases of clustering resources and clustering tasks. The process of cluster formation from resources and tasks is continuous, which is since, the arrival of tasks to scheduler is continuous, also the depiction of schedule intervals of the resources is continuous. This practice of cluster formation proposed helps to avoid the loss of task and scheduling the resources inappropriately are important confines in the existing methods. This practice of clustering process enables to cluster the time series data that is streaming. The core principle of this clustering process is that the new entry at scheduler either possessed by the most recent cluster or used as initial centroid to establish the new cluster.

Clustering the resources

The stages incorporated in the clustering procedure of resources in the form of diverse groups are represented in the following:

- Order the resources in increasing sequence of their schedule interval begin time.
- Select the set of resources in the same order an initial cluster, such that first resource as centroid and the other resources that overlapping with the first resource, which is in regard to schedule intervals of the corresponding resources.
- Choose the new cluster centroid, which is a resource with maximum schedule interval. If change observed in centroid, then move all the resources from the list to the cluster, which are schedule intervals that overlap with the schedule interval of the centroid.
- If centroid is not changed, then finalize the cluster.
- This process continuous, till no resource left in sorted list.

The pseudo code representation of the cluster formation from resources with schedule intervals is depicted in Table 1.

Table 1: Clustering resources with schedule intervals

step 1. The list of resources $R=\{r_1,r_2,...,r_{|R|}\exists b(r_i)\le b(r_{i+1})\}$ arranged in increasing sequence of idle interval time begin.
step 2. while $(|R|>0)$ Begin // while order list of the resources R is not empty
step 3. $cr_j = R\{1\}$ // consider the first entry of the resources list R as initial centroid of the j^{th} cluster
step 4. For each resource $\{r_i \exists r_i \in R \wedge i=1,2,...|R|\}$ Begin
step 5. If $(b(r_i)<e(cr_j))$ Begin // if idle time begin time $b(r_i)$ of the resource r_i is less than the idle time end time $e(cr_j)$ of the resource $\{cr_j \exists cr_j \in R\}$, which is centroid of the j^{th} cluster
step 6. $rcl_j \leftarrow r_i$ // move the resource r_i in to j^{th} cluster rcl_j
step 7. End // endif in step 5
step 8. End // of for in step 4
step 9. Sort the resources fall in cluster rcl_j in decreasing sequence of their schedule interval-end time
step 10. If $(cr_j \neq \{rcl_j\{1\}\}$ Begin // if the present centroid cr_j of the cluster rcl_j is not the first entry $rcl_j\{1\}$ of the cluster rcl_j
step 11. $cr_j \leftarrow rcl_j\{1\}$ // select the first entry $rcl_j\{1\}$ of the cluster rcl_j as new centroid cr_j of the j^{th} cluster
step 12. $rcl_j \setminus rcl_j$ // empty the cluster rcl_j
step 13. Go to step 4
step 14. End // of if in step 10
step 15. Else Begin // of condition in step 10
step 16. $R \setminus rcl_j$ \\ prune the entries of the j^{th} cluster rcl_j from ordered resources list R
step 17. $j=j+1$
step 18. End // of while

Clustering the Tasks to be scheduled

The data packets those buffered at access point are transformed to tasks by the controller, such that each task consists pool of data packets. Further, controller informs to scheduler, which is about these tasks. Upon receiving the information about tasks, the scheduler groups these tasks, such that each group consists the tasks that are overlapping about their arrival time and transmission completion time of the respective tasks. The formation of the task groups is like formation of the groups of resources with overlapping schedule intervals, which is briefed in following:

- List the tasks of arrival time in increasing sequence.
- Select tasks from the list as a cluster, such that the cluster consists all the tasks that are overlying with completion time of transmission of cluster first entry, which is said to be initial centroid of resultant cluster.
- Further, select entry of the cluster depicted, which required maximum task completion time as novel cluster centroid. If the newly selected centroid is not identical to the current centroid, then empty the entries in cluster and collect the tasks from list as cluster entries, such that each of the cluster entry overlaps about its arrival time with task completion time of the centroid.
- If newly selected centroid is identical to the current centroid lock the cluster and remove all cluster entries from the list.
- Continue this process till the list is not empty.

The algorithmic flow of the clustering process depicted as pseudo code in Table 2.

Table 2: The group formation of the tasks

step 1. The list of tasks $T=\{t_1,t_2,...,t_{|T|}\exists a(t_i)\le a(t_{i+1})\}$ arranged in increasing sequence of the arrival time.
step 2. while $(|T|>0)$ Begin // while order list of the tasks T is not empty
step 3. $ct_j = T\{1\}$ // consider the first entry of the tasks list T as initial centroid of the j^{th} cluster
step 4. For each task $\{t_i \exists t_i \in T \wedge i=1,2,...|T|\}$ Begin
step 5. If $(a(t_i)<e(ct_j))$ Begin // if arrival time $a(t_i)$ of the task t_i is less than the transmission completion time $e(ct_j)$ of the task $\{ct_j \exists ct_j \in T\}$, which is centroid of the j^{th} cluster tcl_j
step 6. $tcl_j \leftarrow t_i$ // move the task t_i in to j^{th} cluster tcl_j
step 7. End // end of if in step 5
step 8. End // of for in step 4
step 9. Sort the tasks fall in cluster tcl_j in decreasing sequence of their job completion time
step 10. If $(ct_j \neq \{tcl_j\{1\}\}$ Begin // if the present centroid ct_j of the cluster tcl_j is not the first entry $tcl_j\{1\}$ of the cluster tcl_j
step 11. $ct_j \leftarrow tcl_j\{1\}$ // select the first entry $tcl_j\{1\}$ of the cluster tcl_j as new centroid ct_j of the j^{th} cluster
step 12. $tcl_j \setminus tcl_j$ // empty the cluster tcl_j
step 13. Go to step 4
step 14. End // of if in step 10
step 15. Else Begin // of condition in step 10
step 16. $R \setminus tcl_j$ \\ prune the entries of the j^{th} cluster tcl_j from ordered tasks list R
step 17. $j=j+1$
step 18. End // of while in step 2

Scheduling by correlating task-clusters and resource-clusters

Later the process, for each cluster respective to tasks, find the average task completion time required, and respective root mean square distance of tasks projected in corresponding cluster. Likewise, for each cluster respective to resources, find the average schedule interval, and respective root mean square distance of the resources projected in corresponding cluster.

Further procedure of the suggested method, associates each cluster of the tasks with possible number of clusters related resources, such that the least arrival time observed in task cluster must be greater than or equal to the least begin time of the schedule interval observed in corresponding cluster of the resources. The correlation aspect for associating the resource cluster & task cluster is determined in the following way:

For a specified cluster related to tasks tcl, a cluster rcl that projecting resources will be associated as optimal under following conditions:

- Sort the transmission widows projected by the cluster tcl in increasing sequence of the arrival time, also sort resources projected by the cluster rcl in increasing sequence of their schedule interval begin time.
- if arrival-time $at_1(tcl)$ of the first task in cluster tcl is approximately equal to schedule interval begin time $bt_1(rcl)$ of the first resource in cluster rcl, which can be depicted as $(at_1(tcl) \cong bt_1(rcl))$, and
- If the average schedule interval $\langle si \rangle_{rcl}$ associated to the cluster rcl is greater than or equal to the average task completion time $\langle tct \rangle_{tcl}$ of the cluster tcl $(\langle si \rangle_{rcl} \geq \langle tct \rangle_{tcl})$,
- If the root mean square distance $d_{\langle si \rangle}(rcl)$ observed for schedule intervals spanned over the resources projected by cluster rcl is less than the root mean square distance $d_{\langle tct \rangle}(tcl)$ observed for required task completion times of the tasks projected in cluster tcl $\left(d_{\langle si \rangle}(rcl) \leq d_{\langle tct \rangle}(tcl)\right)$,

If the cluster tcl and cluster rcl strongly correlates according to the constraints given above, further process schedules an optimal resource from the cluster rcl to each of the task projected by the cluster tcl as follows.

The scheduler selects an optimal resource to the task depicted in cluster, such that, the resource bandwidth is competent against load and schedule interval of the resource is greater than the required task completion time.

The criteria to allocate an optimal resource for each task is as follows: If required schedule interval is significantly lower than the available schedule interval, then the scheduler verifies the usage frequency of that resource, if found to usage frequency is high, then look forward to select either a resource that is not frequent to be scheduled and having overwhelmed schedule interval that compared to required schedule interval. If not succeed at this, then selects the resources that already engaged in transmission, but having significant idle time, more than the desired task completion time, then schedules. If none of the above conditions succeed, then the respective task will be dropped and the same will be acknowledged to the controller.

IV. EXPERIMENTAL SETUP AND EMPIRICAL ANALYSIS & RESULTS

Empirical Study

The empirical study for comparing the results of proposed Resource Scheduling scheme DQAERS and other existing method "Optimal Resource Scheduling using Cuckoo Search (ORS-CS) for IAAS cloud computing" [17] are simulated by Cloudsim [18], which allows the simulating high dimensional Cloud computing platforms of network. Here, Synthesizing input tasks, so that there could be no priority order towards corresponding jobs. The confines are executed for performing the experiment from 1 processor towards the other; interrupting & preempting at the time of performance were not enabled. The VMs are in the form of scheduled resources, which utilizes the suggested & other existing methods chosen aimed at the analysis of performance. The suggested method DQAERS and the other existing method ORS-CS, which are deliberated for the analysis of performance tends to attain the "minimal resource failure ratio", and "minimal delay in the task completion". Therefore, in respect to this, these methods fitness is evaluated by comparing outcomes that are attained for metrics (a) Response Failure Ratio (RFR), (b) ratio of job completion delay (JCD), and (c) Load Disparity Ratio (LDR).

The Performance Analysis

At the time of simulation procedure, the log updates of the event are conducted in averse to every attempt of the scheduling VMs in the form of resources. Here, values for every of above mentioned metrics are noticed during fixed time intervals of simulation that are evaluated from the listed values in the log of event. Comparing the values attained for all these metrics at the time of performance of the proposed method and other existing method. The RFR noticed at fixed time intervals for the DQAERS is in between 1.3-1.6%, which is definitely lower than RFRs that are noticed in between 2-6% of existing ORS-CS method. The results signifying that the DQAERS is stable & minimal in respect to RFR (in Figure 1).

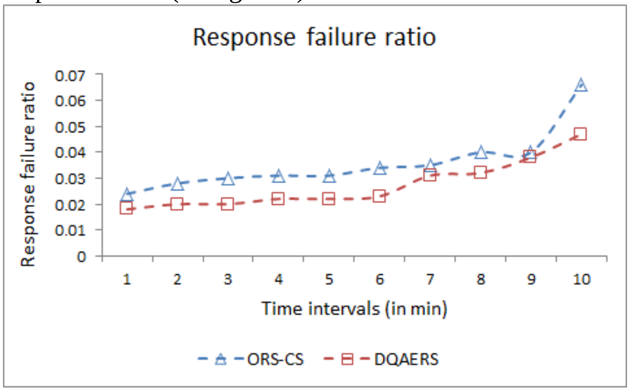

Figure 1: The response failure ratio noticed at diverse intervals of time

The LDR represented for the ORS-CS, DQAERS are compared in Figure 2 that signifies the DQAERS possessing minimum LDR when compared with the contemporary ORS-CS method. The noticed disparity ratios were 6.5 % & 3.1% for ORS-CS & DQAERS respectively.

Figure 3 representing the values compared for the other performance evaluation metric known as JCD that is noticed from the simulations conducted on ORS-CS & DQAERS. The proposed DQAERS method depicted the delay in respect to the completion of job. It is comparatively lower when compared with the ratio of JCD noticed for contemporary ORS-CS method. Here, average delay noticed for ORS-CS & DQAERS are 14 % & 9% in respective order.

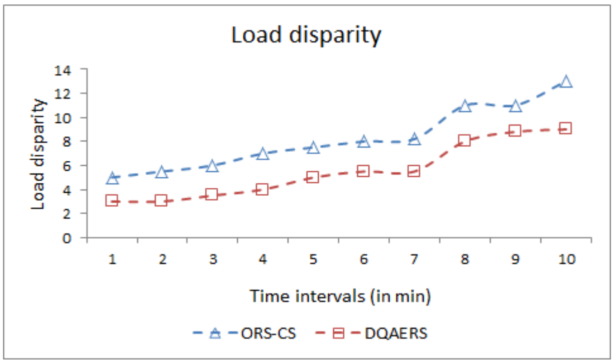

Figure 2: Load Disparity noticed at diverse intervals of time

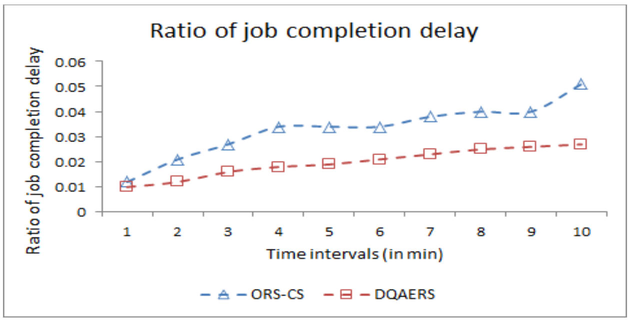

Figure 3: Ratio of job completion delay noticed at distinct intervals of time

V. CONCLUSION

This manuscript proposed a Batch scheduling strategy that pairs the group of tasks with the batch of resources available. The suggested DQAERS method objective is to achieve maximum utilization of resource as well as with maximum task completion. The method suggested in this paper is tasks scheduling in the form of groups and every group of scheduled tasks in a hierarchy sequence. The initial process of the depicted model clusters the tasks to schedule, which clusters the tasks based on their arrival time, clusters the resources with schedule interval based on their begin time of the indolent interval. Further, the tasks observed in each group schedules over resources with schedule interval noticed in respective resource group that interconnects. The experimental study evincing that the suggested method simplifies the procedure of scheduling and increases the optimal selection of resources. The inputs which are collected in this research motivate to explore the opportunity to scheduling parallel group with conflict optimization in further study.

REFERENCES

1. Sandholm, Thomas, and Dongman Lee. "Notes on Cloud computing principles." Journal of Cloud Computing 3, no. 1 (2014): 21.
2. Manvi, Sunilkumar S., and Gopal Krishna Shyam. "Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey." (2013).
3. Mustafa, Saad, Babar Nazir, Amir Hayat, and Sajjad A. Madani. "Resource management in cloud computing: Taxonomy, prospects, and challenges." Computers & Electrical Engineering 47 (2015): 186-203.
4. Durao, Frederico, Jose Fernando S. Carvalho, Anderson Fonseka, and Vinicius Cardoso Garcia. "A systematic review on cloud computing." The Journal of Supercomputing 68, no. 3 (2014): 1321-1346.
5. Yang, Haibo, and Mary Tate. "A descriptive literature review and classification of cloud computing research." CAIS 31 (2012): 2.
6. Achar, Raghavendra, P. Santhi Thilagam, D. Shwetha, and H. Pooja. "Optimal scheduling of computational task in cloud using virtual machine tree." In 2012 Third International Conference on Emerging Applications of Information Technology, pp. 143-146. IEEE, 2012.
7. Zhao, Xun, Yang Zhang, Yongwei Wu, Kang Chen, Jinlei Jiang, and Keqin Li. "Liquid: A scalable deduplication file system for virtual machine images." IEEE transactions on parallel and distributed systems 25, no. 5 (2013): 1257-1266.
8. Bansal, Nidhi, Amitab Maurya, Tarun Kumar, Manzeet Singh, and Shruti Bansal. "Cost performance of QoS Driven task scheduling in cloud computing." Procedia Computer Science 57 (2015): 126-130.
9. Xie, Ruitao, Yonggang Wen, Xiaohua Jia, and Haiyong Xie. "Supporting seamless virtual machine migration via named data networking in cloud data center." IEEE Transactions on Parallel and Distributed Systems 26, no. 12 (2014): 3485-3497.
10. Hua, Yu, Xue Liu, and Dan Feng. "Data similarity-aware computation infrastructure for the cloud." IEEE Transactions on Computers 63, no. 1 (2013): 3-16.
11. Hua, Yu, Bin Xiao, Xue Liu, and Dan Feng. "The design and implementations of locality-aware approximate queries in hybrid storage systems." IEEE transactions on parallel and distributed systems 26, no. 11 (2014): 3194-3207.
12. Batista, Bruno Guazzelli, Julio Cezar Estrella, Carlos Henrique Gomes Ferreira, Dionisio Machado Leite Filho, Luis Hideo Vasconcelos Nakamura, Stephan Reiff-Marganiec, Marcos José Santana, and Regina Helena Carlucci Santana. "Performance evaluation of resource management in cloud computing environments." PloS one 10, no. 11 (2015): e0141914.
13. Nezarat, Amin, and G. H. Dastghaibifard. "Efficient nash equilibrium resource allocation based on game theory mechanism in cloud computing by using auction." PloS one 10, no. 10 (2015): e0138424.
14. Priya, V., and C. Nelson Kennedy Babu. "Moving average fuzzy resource scheduling for virtualized cloud data services." Computer Standards & Interfaces 50 (2017): 251-257.
15. Li, Guangshun, Shuzhen Xu, Junhua Wu, and Heng Ding. "Resource scheduling based on improved spectral clustering algorithm in edge computing." Scientific Programming 2018 (2018).
16. Malawski, Maciej, Kamil Figiela, Marian Bubak, Ewa Deelman, and Jarek Nabrzyski. "Scheduling multilevel deadline-constrained scientific workflows on clouds based on cost optimization." Scientific Programming 2015 (2015): 5.
17. Madni, Syed Hamid Hussain, and Muhammad Shafie Abd Latiff. "Optimal resource scheduling for IaaS cloud computing using cuckoo search algorithm." Sains Humanika 9, no. 1-3 (2017).
18. Calheiros, Rodrigo N., Rajiv Ranjan, Anton Beloglazov, César AF De Rose, and Rajkumar Buyya. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms." Software: Practice and experience 41, no. 1 (2011): 23-50.