Cost Efficient Scheduling Through Auction Mechanism in Cloud Computing

Huda Althumali, Masnida Hussin, Zurina Mohd. Hanapi
Faculty of Computer Science and Information Technology, University Putra Malaysia.
huda-thm@hotmail.com

***Abstract*—Cloud resources provided different types of virtual machine (VM) instances that are assigned to users for particular periods of time. Currently, the process of VM scheduling in Cloud environments is determined by fixed-price scheduling algorithm in which the user pays a fixed amount per unit time in order to obtain the resources. However, such scheduling algorithm is not effective for Cloud since the Cloud's resources are dynamically allocated and released. To address this issue, an adaptive scheduling algorithm for VM allocation in Cloud environments is proposed. Our market-based scheduling algorithm uses the principle of auction mechanism in order to increase the Cloud providers' and uses' satisfaction. Moreover, the scheduling algorithm considers other factors such as auction deadline and network bandwidth with the aim of enhancing the resources utilization and the quality of service (QoS) in Cloud. The simulation experimental results show that the adaptive auction-based scheduling is able to effectively enhance quality of service, profit of Cloud service provider and resource utilization.**

***Index Terms*—Cloud Computing; VM Scheduling; Auction-Based Scheduling; Resource Allocation.**

I. INTRODUCTION

Cloud Computing (CC) trend has been growing very significantly in the recent years. As reported, there are 91% of the organizations in US and Europe agreed that decrease in cost is a main reason for them to migrate to Cloud environment [12]. Moreover, other reasons that encourage such organizations to choose Cloud services are resource accessibility, unlimited storage, security and recovery of information. In fact, one of the most essential characteristics of Cloud is the resource pooling in which user can access a specific resource, use it and release it once finished. The idea of resource sharing helps to reduce the cost of resources is a great deal. In addition, the use of virtualization enhances the effectiveness of resource sharing in Cloud environments. Virtualization refers to the creation of virtual machines (VMs) which acts as a real individual computer. In Cloud environments, data centers consist of millions of devices that hosts the VMs. These VMs considered as one of the most essential resources in Cloud since that all users' tasks are performed on these VMs.

Currently, resource scheduling has become a critical issue in Cloud computing field. This is due to that the effectiveness of the chosen resource scheduling algorithm has a great impact on the entire performance of Cloud. Thus, it is very significant for researchers to pay more efforts in the issue of resource scheduling in Cloud environments. This work is proposed to address the issue of VMs scheduling in Cloud data centers in order to achieve higher performance of Cloud services. VMs resource scheduling refers to the process of assigning the users' tasks to the most suitable VMs that matches both users' and providers' requirements. From a market point of view, there are many current Cloud providers [1-5] are using the fixed-price scheduling which allows the user to get a specific resource in time duration. However, due to the Cloud resources are dynamically allocated and released such scheduling approach is less suitable to apply anymore. The use of the fixed-price scheduling is leading to a serious wasting of Cloud resources when some users occupy the Cloud resources for a specific duration without using them for the whole duration. This is due to that they pay a fixed amount per time unit for the entire duration. Thus, there is a demand for creating adaptive and dynamic scheduling mechanism for Cloud in order to achieve higher resource utilization while increasing Cloud performance.

II. RELATED WORKS

Cloud computing, as a recent computing trend, has extensively attracted attentions of researchers from both industry and academia. Many studies have been conducted in the area of Cloud computing involving general challenges service models. More precisely, most of researches have been accomplished great efforts in VM allocation approaches and resource provision models (e.g., [6, 7, 9]) in order to achieve significant performance in Cloud services. In fact, the most significant goal of Cloud service providers is to raise their revenues. Currently, there are two available renting techniques to obtain the VM instances, which are *pay as you go* and *long-term deal* [3-5]. In both cases, the users have to pay fixed-prices for each unit of time for the resources usage; the single variance is that by choosing a long-term deal, the amount will be lesser in order to use the same resource. However, these fixed-price strategies for resource allocation are difficult to reach well-adjusted prices between market demand and supply which leads to decline service providers' revenue.

Recently, auctioning strategy has started to gain interest in the Cloud computing as a smart and dynamic scheduling approach to address the problem of resources allocation. There are several market-based scheduling approaches that are using the principles of auction strategy in their works to improve service providers' profits. The novel negotiation model between resource manager and scheduler has been proposed based on Continuous Double Auction (CDA) [12]. The authors combined the negotiation model with the CDA model to manage the available resources. Their auction strategy recognized the estimation behavior patterns that can be implemented by the scheduler to reduce the application running cost in regards execution time and utilization. However, the auction strategy does not include a user demand as part of their auction factor. To address the user demand issue, the effective allocation algorithm that uses auction principles has been introduced in [5]. Their work achieves a great improvement compared to the traditional auction mechanism. They proposed two combinatorial auction-based allocation techniques where considered the user demand in a particular request and the greedy extension mechanism. Their results illustrate that the proposed combinatorial auction-based algorithm evidently beat the fixed-price strategy not only in term of increasing the providers' income but also in improving resource utilization and allocation efficiency. Another auction mechanism which uses the greedy combinatorial-auction mechanism is introduced in [17] to solve the resource allocation issue. They utilized the price group that represents truthfulness in the auction process. The simulation results show that the auction mechanism indicate a good improvement in performance than the fixed price mechanisms in terms of both users' and providers' utility. On the other hand, the scheduling algorithm needs to consider the suitable compute matching between demand and supply in heterogeneous environment. Dynamic pricing reverse auction mechanism is designed in order to achieve an efficient allocation of Cloud resources [8]. In their model, the bidders choose the best resource which offers the shortest turnaround time and lower completion cost. The provider who offers lowest price is the winner. At the end of each auction, the Cloud providers update their products price based on the trade situation. The experimental results show that the dynamic pricing auction accomplished higher provider's revenue and lower financial cost for users while improving the resources utility.

There are researchers that come out with new market model which relies on the idea of double auction instead of the normal auction [13]. The authors in [13] proposed the Combinatorial Double Auction Resource Allocation model (CDARA) that is an extension of combinatorial model in [5]. In this model, both users and providers send their bids to the auctioneer who is responsible for the auction process. Then, the auctioneer matched each user's bid with the appropriate provider's bid according to the requirements of both. Therefore, each user's task is allocated to the most suitable VM to be executed.

The authors in [2] proposed the multi-attribute combinatorial double auction model with fairness. In their work, both providers and users send their bids to the auctioneer. The auctioneer decides the winner not only based on the offered price but also considers different parameters such as reputation, fairness and the offered QoS. According to the auction results, each offered resource will be allocated to the suitable user to accomplish its tasks. After releasing the resource, the user sends the feedback to the auctioneer about the received QoS so that the auctioneer can impose a penalty on the provider in case that the QoS is not meeting the user satisfaction. Although that all previous auction strategies provided good improvement in increasing the performance, they still ignore some factors such as auction deadline and network bandwidth. The ignorance of these factors leads to some wasting in VM resources. The auction deadline helps to detect what called "zombie clients" who often send a service request to the provider but the provider cannot get any response from them for a long time. So that, the adoption of auction deadline will make the provider able to delete these clients after the competition deadline and in turn save the resources from wasting. Moreover, adding an auctioneer entity that is responsible for the auction process could enhance the resource scheduling procedure.

III. System Design

Our scheduling algorithm relies on auction-based market strategy. It considers network bandwidth and auction deadline in auction process. The system model of the Cloud is constructed as follows:

A. *Cloud Service Provider Sp:*

- The quantity of VM resource types provided by S_p is K.
- The total VM resource quantity is *N*.
- The total amount of network bandwidth is *B*.
- The running and the idle maintenance costs of the VM resource $S_{k,i}$ ($1 \le k \le K$, $1 \le i \le N$) are $C_{k,i,R}$ and $C_{k,i,I}$, respectively.

B. *Clients Group Cg:*

- The number of clients in the group is *n*.
- Each client has one broker *Br* which is responsible for auctioning process and tasks execution.
- The service request (bid) sent from a broker $Br(p)$ $(1 \le p \le n)$ is $(S_{k,i,p}, b_p, t_p, P_p)$. $S_{k,i,P}$ denotes that the required number of the k_{th} VM resource the client $Cg(p)$ requires is *i*. b_p is the network bandwidth demand of the client $Cg(p)$. t_p is the competitive payments moment of the client $Cg(p)$. P_p denotes the highest price the client $Cg(p)$ can afford for the current service request. It is significant that t_p must be located at the competition deadline which announced by the Cloud auctioneer *Ca*, else the service request of $Cg(p)$ is invalid. Moreover, the final payment *PP'* must satisfy $P_{p'} \le P_p$. The client group *Cg* should also satisfy the following requirement:

$$\sum_{p=1}^{n} bp \le B \qquad (1)$$

C. *Cloud Auctioneer Ca:*

- Auction starting time is T_{min}.
- Auction closed time is T_{max}.
- The auction deadline of the VM resource at a certain time is $[T_{min}, T_{max}]$.

Cloud auctioneer is the one that important and responsible for the auction process. It starts the auction at T_{min} and announces that the auction will be closed at T_{max}. The auctioneer then computes the allocation of resources based on some criteria and announces the results to Cloud users and Cloud service provider. Figure 1 shows the basic system model for the auction based scheduling algorithm. It optimistically expected that the execution time will be improved since processing overhead is migrated to the auctioneer instead of Cloud service provider.

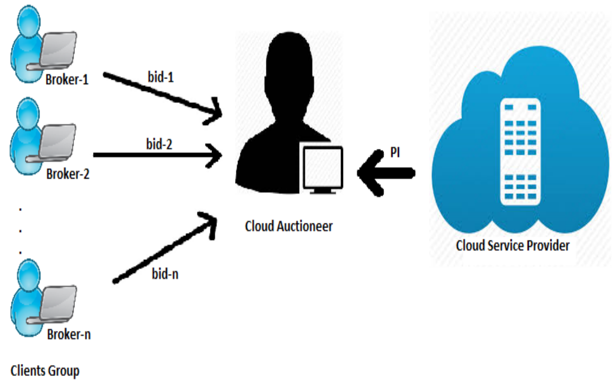

Figure 1: System Model

Furthermore, the QoS parameters can be monitored by the auctioneer to ensure that the Cloud provider is delivering the claimed QoS level. The auction deadline is determined by the auctioneer in order to prevent *"zombie clients"* from wasting the Cloud resources. Zombie clients often occupy the bandwidth resource and send service requests to the Cloud service provider. Meantime, the Cloud service provider did not received any response from them for some period of time. Therefore, the Cloud auctioneer in our system model is able to announce and refresh the auction deadline, so that the zombie clients can be deleted.

IV. AUCTION-BASED SCHEDULING ALGORITHM

The scheduling algorithm in our work is composed of two parts. The first one is the resources availability evaluation and VM configuration, and the other one is the client auction payment mechanism.

A. Resources property evaluation and VM configuration:

This first part is responsible for evaluating the availability of Cloud resources (VMs) based on the indicator called performance index (PI) whose value is in the range [0,1]. There are four indexes used in computed the performance index (PI). Those four indexes are described as follow:

- RI: The number of bids received by Cloud auctioneer in unit time.

$$\text{RI}= \sum_{i=1}^{n} \text{RI}i \tag{2}$$

- DI: The number of user requests accomplished by VM resources in unit time.

$$\text{DI}= \sum_{i=1}^{n} \text{DI}i \tag{3}$$

- LI: The ratio between the average completion time of client's request and client's request interval time.

$$\text{LI} = \frac{\text{ti}}{\text{td}} \tag{4}$$

- CI: Maintenance cost which is in range between the maintenance cost of the idle VMs and the maintenance cost of the running VMs.

$$\text{CI} = [\ \sum_{i=1}^{n} \text{CiI}\ , \sum_{i=1}^{n} \text{CiR} \tag{5}$$

The larger value of PI indicates that lesser resources are available and the overall performance of Cloud resources is low. Suppose the VM resources set denoted by S= {S_1, S_2, ... ,S_n}, and n is the number of VM resources, $1\leq i \leq n$. The following mathematical expression describes the performance index (PI):

$$\text{PI} = \begin{cases} 1, & \text{if } RIi = \text{R}* \text{ or DIi} \geq \text{D}* \text{ or LIi} \geq \text{L}* \text{ or CIi} \geq \text{C}* \\ \max[\frac{RIi}{R*}, \frac{DIi}{D*}, \frac{LIi}{L*}, \frac{CIi}{C*}], & else \end{cases} \tag{6}$$

In the Equation (6), RI_i, DI_i, LI_i, CI_i are the four indexes for the current VM where R*, D*, L*, C* are the largest quantities of the service requests, dealing ability, loading and the cost that the current VM can afford. This equation indicates that if any one of the four indexes reaches its maximum value, then PI is equal to 1, which means that there are no resources are available and the performance of the Cloud is very poor. Otherwise, the VM resources are still available.

The resource property evaluation depends on both the clients' service requests and the performance index (PI). This part can be described by the following steps:

- Cloud auctioneer *Ca* starts the auction process at time T_{min}.
- Cloud service provider *Sp* computes the performance index (PI) and sends it to the Cloud auctioneer *Ca*.
- The clients $Cg(p)$ send bids ($S_{i,p}$, b_p, t_p, P_p) to the Cloud auctioneer, where $S_{i,p}$, b_p, t_p, P_p are the required quantity of VMs, the required bandwidth, the request moment (time) and the highest price offered by the client respectively.
- Cloud auctioneer closes the auction at time T_{max}.
- Cloud auctioneer examines the bid moment for all clients. Only the bids that are sent within the deadline are considered.
- For all clients who meet the deadline, Cloud auctioneer examines the offered price P_p and the requested resources ($S_{i,p}$, b_p) according to the following requirements:
  - $Pp \geq C_{k,i,R}$
  - $\sum_{i=1}^{K} \sum_{p=1}^{n} Si,p \leq N$
  - $\sum_{p=1}^{n} bp \leq B$
- Cloud auctioneer then obtains the final client group *Sc* who are auctioning successfully.
- For the final clients group, Cloud auctioneer computes the resources allocation according to the highest price P stated in the client service request.
- Cloud service provider *Sp* allocates the VM resources to clients group $Cg(p)$.

The Cloud auctioneer maintains three filtering processes. First, it made upon the auctioning process by announcing the determine deadline [T_{min}, T_{max}] when the client sends the service request. This step aims to guarantee that the client's request is filtering within the period of time. It also to insure the zombie clients cannot occupy the resources for long time whiles the legal clients waiting the resource. The second filtering step is based on the price afford by the client. The price must be greater than the maintenance cost so that the provider can accomplish the minimum profit. Thus, all clients that wish to pay less than the maintenance cost will be deleted from the auctioning client group. The third filtering process is conducted based on the required resources requested by the client. If the client requests a quantity of VMs which is greater than the total number of VMs offered by the Cloud provider, the client will be deleted from the client group. The required bandwidth is examined as well to ensure that the requested bandwidth is available. As a result for the three filtering process, the client group auctioning successfully is conducted and the resources are allocated to these clients based on the highest price offered. The looser clients have to bid again in the next auction process. This resource property evaluation process provides the base for the second part of the auction-based scheduling algorithm.

B. Client Auction Payment Mechanism:

The auction payment mechanism is constructed in order to ensure the client can obtain the required resource with the minimum cost and the provider can achieve a good level of profit. The objective of this part is to find the final price *Pp*'.Upon the payment, the clients need to consider several factors such as the average price in clients group and the competitive prices offered by other clients. Suppose that the client group *Sc* contains *x* clients, then the following price identification take into consideration:

- For each client, Cloud auctioneer computes the average price *Avg* offered by other clients.
- Cloud auctioneer then finds the price *Pz* offered the client S_z who is the most serious competitor toward the current client.
- Final Price P_p , is the maximum value between *Avg* and P_z.

The Cloud auctioneer makes a search within the client group elements based on two factors. The first factor is the average payment level of the client group *Sc* which reflects the payment ability of the whole group. Second, the auctioneer evaluates the price of the most serious competitor client. This factor is involved in order to consider the competitive payment which helps in turn to increase the level of provider's profit. Naturally, the final price must be the maximum between the average payment and the price offered by the most serious competitor client.

V. SIMULATION EXPERIMENTS

The scheduling algorithm in this work is simulated using CloudSim 3.0.3 which is a Java-based library used to simulate different Cloud environments. The simulation is conducted on a PC whose configurations are as follows: 2.4 GHz CPU, 8 GB Memory, 512 GB hard disk. Eclipse is used as an environment to run the CloudSim library. The setting of the simulation parameters are shown in Table 1.

Table 1
Simulation Parameters Setting

Parameter	Value
The total quantity of VMs offered by the Cloud provider	40,000
Number of clients (Cloud users)	20,000
The total network bandwidth	11 Mbit/s
The maintenance cost of a running VM	0.5 cent/h
The maintenance cost of an idle VM	0.15 cent/h
The auction deadline interval	20 min.

The simulation of the auction-based scheduling algorithm is executed in a scalable simulation program that models the fundamental Cloud environment. It involves the creation of the following Cloud entities:

i. Cloud service provider that is represented by the datacenter class.
ii. Cloud users who are represented by the broker class.
iii. Cloud auctioneer which represented by auctioneer class.

After the basic entities have been created, the creation of VMs and Cloudlets is done consequently. The proposed algorithm is developed inside the auctioneer entity who is responsible for the auctioning process and the VMs allocation process. The simulation process is started and the results are printed after the simulation has finished.

VI. PERFORMANCE EVALUATION AND RESULTS

The performance of the proposed algorithm has been evaluated based on three performance parameters:

i. *Profit of Cloud service provider:* which can be expressed as follows based on the system design mentioned earlier:

$$\sum_{P=1}^{n} \mathrm{Pp}'\lambda p-\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{P=1}^{n} \mathrm{Ck,i,R}\,\lambda\mathrm{p}-\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{P=1}^{n} \mathrm{Ck,i,I}(1-\lambda\mathrm{p}) \qquad (7)$$

where λp denotes the service response coefficient. If the Cloud service provider accepts the service request from the client, λp = 1, else λp = 0.

ii. *Resource Utilization Rate*; which can be computed based on the running time and overall simulation time of VM resource. The utilization rate for a given VM in this work can be expressed as follow:

$$U=\frac{\mathit{Running\ Time}}{\mathit{Ovrall\ Simulation\ Time}} \times 100 \qquad (8)$$

iii. *Execution time*; which refers to the time spent to execute the client service by the VM. This parameter can be expressed as follow:

$$\mathit{Execution\ Time} = \mathit{Completion\ Time} - \mathit{Submission\ Time} \qquad (9)$$

Moreover, in order to evaluate the effectiveness of the auction-based scheduling algorithm, it is compared to the fixed-price-based VM resource allocation. The fixed-price used here is 30 cent/h [2].

The relation between the number of clients and the provider's profit for the auction-based scheduling algorithm and the fixed-price scheduling algorithm is shown in Figure 2. It shows that the profit in the auction-based scheduling algorithm is increased continuously than the fixed price scheme. This is due that the fixed price scheme merely allows clients to pay the same amount while the auctioning prices are divers according to the average price offered by different clients.

In Figure 3 shows the resource utilization rate of the fixed price scheme is not affected by the number of clients. It is because each client is occupying the resource for the whole duration without necessary to utilize them. This leads to lower resource utilization compared to the auction-based scheduling algorithm in which the auctioneer allocates the resources to the clients according to the requested resources and the offered price. Furthermore, it illustrates that in Figure 4 the execution time is exponentially higher in the fixed price scheme when the number of Cloudlets increased. However, this increment is growing slowly in the auctioning scheme where the execution time is steadily increased about merely 10% on average as the number of Cloudlets growth. Consequently, reduction in the execution time in turn improves the performance of Cloud resources.

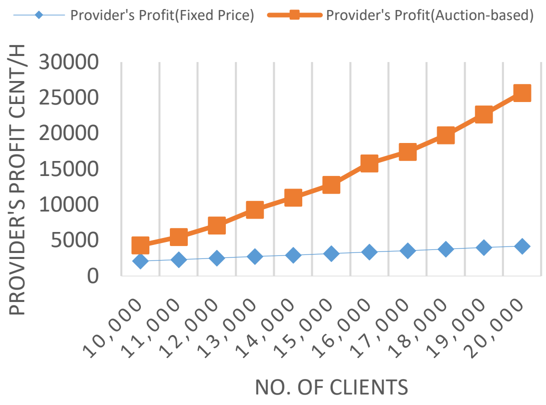

Figure 2: Relation between provider's profit and number of clients.

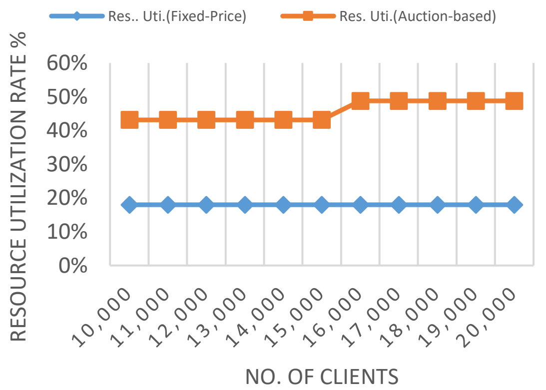

Figure 3: Relation between resource utilization rate and number of clients

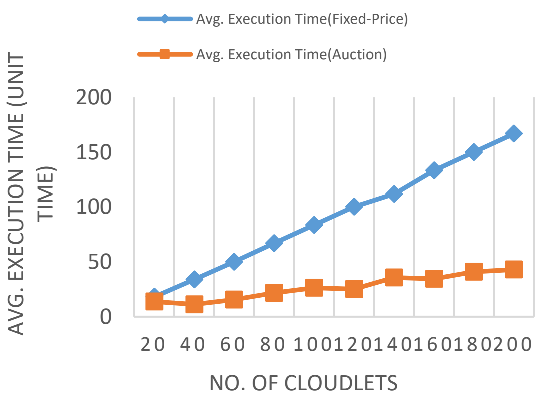

Figure 4: Average execution time.

VII. CONCLUSION AND FUTURE WORK

This work proposed an auction-based scheduling algorithm in which the clients submit bids containing the required resources and the suitable price that they able to pay. These bids are sent to the Cloud auctioneer which already has received information from Cloud provider about the available resources. The auctioneer then examines the clients' bids based on three criteria: quantity of requested resources, bid submission moment and the offered price. The available resources are allocated to the clients who are auctioning successfully. The simulation experimental results show that the proposed scheduling algorithm increases the profit of Cloud provider and QoS level. The auction-based scheduling algorithm can be extended to include some penalty against the Cloud providers who do not provide respective QoS level. Such improvement could enhance the Cloud performance in terms of reliability.

REFERENCES

[1] Ban, Y., Chen, H., & Wang, Z. (2013). 2013 IEEE Seventh International Symposium on Service-Oriented System Engineering EALARM : Enhanced Autonomic Load-Aware Resource Management for P2P Key-value Storage in Cloud. http://doi.org/10.1109/SOSE.2013.38

[2] Baranwal, G., & Vidyarthi, D. P. (2015). The Journal of Systems and Software A fair multi-attribute combinatorial double auction model for resource allocation in Cloud computing. *The Journal of Systems & Software*, *108*, 60–76. http://doi.org/10.1016/j.jss.2015.06.025

[3] Calheiros, R. N., Ranjan, R., Beloglazov, A., & Rose, A. F. De. (2011). CloudSim : a toolkit for modeling and simulation of Cloud computing environments and evaluation of resource provisioning algorithms, (August 2010), 23–50. http://doi.org/10.1002/spe

[4] Cao, J., Wu, Y., & Li, M. (2012). Energy Efficient Allocation of Virtual Machines in Cloud, 137–151.

[5] Comput, J. P. D., Zaman, S., & Grosu, D. (2013). Combinatorial auction-based allocation of virtual machine instances in Clouds. *J. Parallel Distrib. Comput.*, *73*(4),495–508. http://doi.org/10.1016/j.jpdc.2012.12.006

[6] Hassan, M. M., & Alamri, A. (2014). Virtual Machine Resource Allocation for Multimedia Cloud : A Nash Bargaining Approach. *Procedia - Procedia Computer Science*,*34*,571–576. http://doi.org/10.1016/j.procs.2014.07.074

[7] Kong, W., Lei, Y., & Ma, J. (2016). Virtual machine resource scheduling algorithm for Cloud computing based on auction mechanism. *Optik - International Journal for Light and Electron Optics*, *127*(12), 5099–5104. http://doi.org/10.1016/j.ijleo.2016.02.061

[8] Li, X., Liu, X., & Zhu, E. (2015). An Ef fi cient Resource Allocation Mechanism Based on Dynamic Pricing Reverse Auction for Cloud Work fl ow Systems, *1*, 59–69. http://doi.org/10.1007/978-3-319-19509-4

[9] Minarolli, D., & Freisleben, B. (2013). Virtual Machine Resource Allocation in Cloud Computing via Multi-Agent Fuzzy Control. http://doi.org/10.1109/CGC.2013.35

[10] Mustafa, S., Nazir, B., Hayat, A., Khan, R., & Madani, S. A. (2015). Resource management in Cloud computing : Taxonomy , prospects , and challenges q. *Computers and Electrical Engineering*, *47*, 186–203. http://doi.org/10.1016/j.compeleceng.2015.07.021

[11] Nguyen, Q. T., Quang-hung, N., Tuong, N. H., Tran, V. H., & Thoai, N. (2013). Virtual Machine Allocation in Cloud Computing for Minimizing Total Execution Time on Each Machine, 241–245.

[12] Prodan, R., & Wieczorek, M. (2011). Double Auction-based Scheduling of Scientific Applications in Distributed Grid and Cloud Environments, 531–548. http://doi.org/10.1007/s10723-011-9196-x

[13] Samimi, P., Teimouri, Y., & Mukhtar, M. (2014). A combinatorial double auction resource allocation model in Cloud computing. *INFORMATION SCIENCES*. http://doi.org/10.1016/j.ins.2014.02.008

[14] Yang, C., Cheng, H., & Huang, K. (2011). A Dynamic Resource Allocation Model for Virtual Machine Management on Cloud *, 581–590.

[15] Zaman, S., Member, S., Grosu, D., & Member, S. (2013). A Combinatorial Auction-Based Mechanism for Dynamic VM Provisioning and Allocation in Clouds, *1*(2), 129–141.

[16] Zhang, Q., Chen, H., Shen, Y., Ma, S., & Lu, H. (2016). Optimization of virtual resource management for Cloud applications to cope with traffic burst. *Future Generation Computer Systems*, *58*, 42–55. http://doi.org/10.1016/j.future.2015.12.011

[17] Zhang, Y., Li, B., Huang, Z., Wang, J., Zhu, J., & Peng, H. (2014). Strategy-Proof Auction Mechanism with Group Price for Virtual Machine Allocation in Clouds. http://doi.org/10.1109/CBD.2014.17.