


Virtual machine migration method based on load cognition

Junwu Zhu¹  · Jinjin Wang¹ · Yonglong Zhang¹ · Yi Jiang¹

Published online: 1 November 2018
© Springer-Verlag GmbH Germany, part of Springer Nature 2018

Abstract

Virtualization technology plays an important role in cloud computing. Virtual machine (VM) migration not only enables load balancing of hosts in data center to avoid overload anomalies, but also reduces the cost of cloud computing data centers. Our work mainly focused on the communication costs of VMs migration in data center. In this paper, a double auction-based VM migration algorithm is proposed, which takes the cost of communication between VMs into account under normal operation situation. The algorithm of VM migration is divided into two parts: (i) selecting the VMs to be migrated according to the communication and occupied resources factors of VMs and (ii) determining the destination host for VMs which to be migrated. In the first process of VM migration, we proposed VMs greedy selection algorithm (VMs-GSA) to select VMs. A VM Migration Double Auction Mechanism was applied to the second process of VM migration to obtain the mappings between VMs and underutilized hosts. The simulation result shows that the proposed VM migration algorithm-based heuristic is efficient. The traffic generated by VMs-GSA is 35% less than the random algorithm, and the success rate of VM migration is very high.

Keywords Virtualization · Virtual machine migration · Double auction · Greedy selection algorithm

1 Introduction

With the rapid development of information and data in the Internet age, computational domain including science, engineering, and business need to process large-scale, massive data. The demand for computing power in these areas goes far beyond the computational power of their own information technology architecture, which requires increasing system hardware input to achieve system scalability (Rings et al. 2009). In this case, in order to save costs and achieve system scalability, the concept of cloud computing is proposed.

Cloud computing is a further development of distributed computing, parallel processing, and grid computing (Wang et al. 2010). It is a system based on Internet computing that can provide hardware services, infrastructure services, platform services, software services, storage services to a variety of Internet applications (Lu et al. 2017a, 2018a).

Virtualization is a key technology of cloud computing, which can turn a host into multiple virtual hosts which have different computer systems resources that can support applications. Virtualization has lots of advantages. By reducing the number of physical hosts by turning the physical host into a virtual host, energy consumption is reduced and energy efficiency is achieved. In addition, Virtualization is a cost-effective technology (Goldberg 1974). However, load balancing is a challenge. Suppose that users have some requirements including CPU, memory and hard disk to run their applications. Then, configure the corresponding VM to hosts according to the user's requirements in the data center. After configuring the virtual machine for the user on the host, there will exist overloaded hosts and underutilized hosts. Host resources, such as CPU, memory and hard disk, are limited. When too many VMs are running on one host, the host can become overloaded and cause exceptions. Virtual machine migration can solve this problem. In this case,

Communicated by A. K. Sangaiah, H. Pham, M.-Y. Chen, H. Lu, F. Mercaldo.

✉ Junwu Zhu
jwzhu@yzu.edu.cn

Jinjin Wang
18361311742@163.com

Yonglong Zhang
ylzhang@yzu.edu.cn

Yi Jiang
jiangyi@yzu.edu.cn

¹ College of Information Engineering, Yangzhou University, Yangzhou, Jiangsu, China

we must select one or more VMs to migrate once the host becomes overloaded and find the best destination host for these selected VMs to be migrated. Double auction is widely used in the field of artificial intelligence to solve the problem of resource competition (Lu et al. 2017b, 2018b; Xu et al. 2018; Li et al. 2018).

In the data center, there are generally three kinds of costs including energy consumption, migration cost and communication cost (Tao et al. 2016). The energy consumption in this paper mainly refers to the number of open hosts, which is fixed. The migration cost is closely related to the size of the VM to be migrated and the distance between the source host and the destination host (Zhang et al. 2017). Communication cost is also the major factor of VMs migration, which mainly includes internal communication between VMs on the same host and external communication between VMs on the different hosts. Internal communication is generally ignored because internal communication is generally communicated through random memory instead of the network. However, VM migration has a huge impact on communication costs. Therefore, our work is mainly focused on the communication costs of VMs migration. The main contributions of this paper can be summarized below:

- (1) Proposing a traffic-aware VMs-GSA, which takes the communication costs between VMs and the occupied resources for VMs into account.
- (2) Applying VMM-DAM (VM Migration Double Auction Mechanism) to VM migration to find the most appropriate destination hosts for VMs to be migrated.
- (3) In the auction process, we match the host for the VMs to be migrated based on the traffic generated by the VM migration and the revenue increment, which further reduces the communication cost.

The rest of this paper is organized below. Section 2 reviewed some related works. In Sect. 3, we detailed system model. Proposing VMs-GSA to select VMs to migrate and introducing VMM-DAM to obtain the mappings between VMs and underutilized hosts in Sect. 4. Section 5 shows the simulation results, and Sect. 6 summarizes the paper.

2 Related work

In recent years, VM migration in the data center has been widely studied and has become a hot topic. The process of VM migration mainly involves energy consumption and communication costs of the data center. Shrivastava et al. (2011) studied the problem of VM migration in data centers and proposed a novel VM migration algorithm named AppAware, which takes the communication dependencies among VMs and the underlying data center network topology into account. The AppAware algorithm is greedy and

allocates a VM to the host each time with the goal of minimizing costs. The cost of AppAware was defined as the delay or the number of hops between hosts multiplied by the communication requirements between virtual machines. In Zhang et al. (2017), the authors also used this calculation method to calculate the communication cost of a virtual machine migrating from one host to another. The AppAware was a dynamic VM migration algorithm because its purpose was to find the best migration destination host for each VM. However, this algorithm ignored the energy consumption of VM migration. Heller (2010) believed that network communication and energy consumption caused by virtual machine migration are directly proportional.

In Meng et al. (2010), the authors studied traffic-aware VM placement to improve the network scalability, i.e., communication between VM is proportional to the distance through virtual machine migration. They turned VM migration into an optimization problem considering the traffic rate, the external traffic rate, and communication cost. However, the authors did not consider the limited resources of each host or server.

In order to improve the energy efficiency of the data center, Tao et al. (2016) proposed a new algorithm named BGM-BLA for VM migration, which considered three factors including energy consumption, communication cost, and migration cost. The algorithm in Tao et al. (2016) that targets these three factors has two parts. The first part generates bucket codes, that is, divide the overloaded VMs into groups, and then, evaluate every bucket code. The second part is to learn and mutate to reduce communication cost and migration cost based on the original bucket codes and Output the Pareto set of solutions. However, the authors ignored the original placement of the VM, which increased migration costs. Beloglazov et al. (2012) proposed energy-aware allocation heuristics which provides the resources of servers or hosts including CPU, memory and disk storage to client applications while guaranteeing Quality of Service (QoS). In Reguri et al. (2016), the authors proposed three VM migration schemes which take the traffic factor and VM clustering into account, which is the improvement in papers (Huang et al. 2014; Vu and Hwang 2014). VM clustering includes two steps. The first step forms the VM graph on the same host, in which vertexes are VMs and edges are communications between VMs. Then, delete edges that are less than the average weight and the connected VMs form clusters. However, the VM cluster algorithm only considered the communication of the VMs within the host and did not consider external communication. In addition, if the underutilized hosts have very little free memory, it is difficult to satisfy the VM cluster occupancy. In addition, many studies have considered communication factors in the migration of VMs, such as Kansal and Chana (2016), Gao et al. (2017) and Chen et al. (2012).

Zhang et al. (2017) applies the genetic algorithm (GA) and artificial bee colony (ABC) to the problem of VM migration

problem and aims to find an approximately optimal solution through repeated iterations. The GA first generated a population of PopSize chromosomes which is a vector represented by the mapping of VMs and servers. Then, cross and mutate the chromosomes and compare with the previous chromosome to select smaller chromosomes with smaller fitness value. The ABC and the GA are basically the same. Mutation is another type of method applied to VM migration in Tao et al. (2016) and Zhang et al. (2017). In Azougaghe et al. (2017), migration algorithm is investigated, which is a new system based on matching game theory. The paper applied firefly algorithm to energy-aware VM migration, which migrates the maximally loaded VM to the least loaded active node.

In this paper, we apply VMM-DAM to the problem of VM migration that is a very novel approach. Double auction based on resource allocation and its optimization have been widely used in task schedule (Marahatta et al. 2018), cloud computing (Zhang and Sangaiah 2018; Abdel-Basset et al. 2018), e-governance (Medhane and Sangaiah 2018) and virtual networks (Sun et al. 2018) and its optimization. We considered the traffic factor between VMs and the resource constraints of hosts, and turned the VM migration problem into a constant optimization problem. Apply auctions to the problem of VM migration in an incentive way to find the best destination host for the selected VMs. We also consider selecting VMs with small occupancy for migration to reduce migration time and energy consumption during migration. However, considering the complexity of the external communication of the VMs, we did not consider the external communication of the VMs, only considering the internal communication of the VMs in the host.

3 System model

We consider that there are K hosts in data center, and the relation between these hosts can be denoted as an undirected graph $DC = (H, D)$, where $H = \{h_1, h_2, \dots, h_k\}$ represents the set of hosts and $\forall (h_i, h_j) \in D$ represents the edge between hosts h_i and h_j . The weight $W(h_i, h_j)$ is communication distance, i.e., physical distance, or the number of hops between h_i and h_j .

Each host $h_j \in H$ can be represented by a 3-tuple $(\tau(h_j), sl(h_j), V(h_j))$, where $\tau(h_j)$ is the threshold of h_j , i.e., if h_j 's load exceeds this value $\tau(h_j)$, h_j is overloaded. $sl(h_j)$ indicates the safety limit of h_j , i.e., if h_j 's load is lower than this value $sl(h_j)$, h_j is underutilized. The shaded area in Fig. 1 is the host's load. Therefore, as is shown in Fig. 1, h_i is underutilized, h_l is fully utilized and h_j is overloaded. $V(h_j) = \{v_1^j, v_2^j, \dots, v_n^j\}$ is the set of VMs residing on h_j . v_i^j represents the VM i

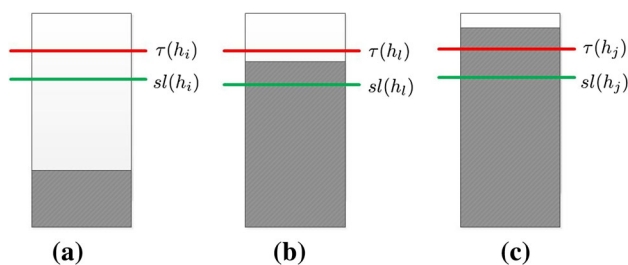


Fig. 1 An example of hosts threshold and safety limit

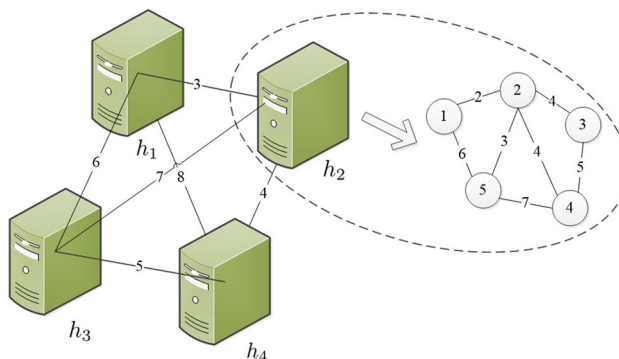


Fig. 2 The relation graph of hosts and VMs

residing on host j . The communication relation of VMs residing on h_j is represented by an undirected graph $G = (V, E)$, where $V = V(h_j)$ is the set of vertices and $E = \{(v_i^j, v_l^j) | \text{where } v_i^j \text{ and } v_l^j \text{ exists communications, } v_i^j, v_l^j \in V(h_j)\}$ is the set of edges. The weight $W'(v_i^j, v_l^j)$ represents VM' traffic, which is proportional to their actual communication. $\forall v_l^j \in V(h_j)$ that communicate with v_i^j are called v_i^j 's neighbor, i.e., $N(v_i^j) = \{v_l^j | (v_i^j, v_l^j) \in E, \forall v_l^j \in V(h_j)\}$.

$v_i^j \in V(h_j)$ have two attributes denoted by a 2-tuple $(O(v_i^j), C(v_i^j))$, where $O(v_i^j)$ denotes the size of occupied resources of v_i^j . In this paper, we assume that there are communications between the VMs on the same host and there are no communications between the VMs on different hosts. Therefore, the traffic of v_i^j is $C(v_i^j) = \sum_{v_l^j \in N(v_i^j)} W'(v_i^j, v_l^j)$, i.e., the communications with other VMs on the same host where it resides on, which is the sum of the weights of the edges connected with v_i^j . Fig. 2 shows an example of VMs communication residing on h_2 . For example, the traffic of v_1^2 is $6 + 2 = 8$ and the traffic of v_2^2 is $2 + 3 + 4 + 4 = 13$.

The set of overloaded hosts is represented by $H^+ = \{h_j | \sum_{i \in V(h_j)} O(i) > \tau(h_j), h_j \in H\}$, where $\sum_{i \in V(h_j)} O(i)$ is the load of h_j . $H^- = \{h_j | \sum_{i \in V(h_j)} O(i) \leq sl(h_j), h_j \in H\}$ denotes the set of underutilized hosts. We use $h_j^+ \in H^+$ to denote overloaded host h_j and $h_j^- \in H^-$ to denote underutilized host h_j .

The process of VM migration is to select some VMs from overloaded hosts and migrate them to underutilized hosts, which in order to ensure load balancing of the hosts. The result of selected VMs from all overloaded hosts is represented by $\mathbf{S} = (\vec{s}^1, \vec{s}^2, \dots, \vec{s}^{|H^+|})$, where $\vec{s}^j = (s_1^j, s_2^j, \dots, s_{|V(j)|}^j)$ is the result of selected VMs from h_j^+ , $|V(j)|$ represents the total number of VMs residing on h_j^+ . If v_i^j is selected to migrate, $s_i^j = 1$, i.e., VM i residing on h_j^+ is selected to migrate. otherwise, $s_i^j = 0$.

$$s_i^j = \begin{cases} 1 & \text{if } v_i^j \text{ is selected} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Suppose there are N VMs selected to migrate and we define the set of VMs to be migrated selected form all overloaded hosts as $VM = \{vm_1, vm_2, \dots, vm_N\}$. We employ $r(vm_i)$ to denote source host of vm_i . Let $M = |H^-|$ represent the number of underutilized hosts in data center. A $N \times M$ matrix for indicating the result of the allocation between VMs and underutilized hosts is $X = (x_{ij} | i = 1, 2, \dots, N; j = 1, 2, \dots, M)$, where

$$x_{ij} = \begin{cases} 1 & \text{if } vm_i \text{ is migreted to } h_j^- \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$x_{ij} = 1$ means that the demand of vm_i is satisfied by h_j^- and the number of VMs that allocated to h_j^- is $\sum_{i=1}^N x_{ij}$.

As we have mentioned in the introduction section, our work is mainly focused on the communication costs of VMs migration in data center. Let mapping function $\sigma : VM \rightarrow H^-$ denote the mapping between VMs and underutilized hosts, i.e., if $\sigma(i) = j, x_{ij} = 1$, which means that the destination host of vm_i is h_j^- . The communication cost that vm_i matches $h_j^- \in H^-$ is defined as below:

$$Cost_i = \sum_{vm_l \in N(vm_i)} W'(vm_i, vm_l)W(h_{\sigma(i)}^-, h_{\sigma(l)}^-) \quad (3)$$

If $\sigma(i) = \sigma(l)$, then $W(h_{\sigma(i)}^-, h_{\sigma(l)}^-) = 0$. If the neighbor of the VM to be migrated is not selected to migrate, the neighbor's "destination" host is the host where it resides. Thus, we define the problem of VM migration as an optimization problem as follows:

$$(S^*, X^*) = \arg \min_{S, X} \sum_{vm_i \in VM} Cost_i \quad (4)$$

$$s.t. \quad \sum_{j=1}^M x_{ij} \leq 1, \forall 1 \leq i \leq N \quad (5)$$

$$\sum_{i=1}^N x_{ij} O(vm_i) \leq o_j, \forall 1 \leq j \leq M \quad (6)$$

$o_j = \tau(h_j) - \sum_{i \in V(h_j)} O(i)$ is the size of resources provided by h_j^- , i.e., idle resources. Noted one VM can only match at most one underutilized host. One underutilized host can be matched with multiple VMs. However, the idle resources owned by each underutilized host are limited. The total demand of VMs that matched with the h_j^- can't exceed the h_j^- 's idle resources, otherwise, it is not feasible. Therefore, Eqs. (5) and (6) are constraints for generating mappings between VMs and underutilized hosts.

Traffic-aware VM migration problem is NP-complete (Zhang et al. 2017). Therefore, we proposed VM migration algorithm based on heuristic to solve the problem (4). This algorithm is described in detail in Sect. 4.

4 VM migration algorithm

In this section, VM migration algorithm based on heuristic consists of two parts. The first part is to select VMs from overloaded hosts to migrate, and we proposed VMs-GSA to determine these VMs. The second part is to obtain the mapping between VMs and underutilized hosts, and we employed VMM-DAM to obtain it.

4.1 VMs-GSA design

In this section, we proposed VMs-GSA to select VMs from overloaded hosts to migrate, which takes the communication and the size of VMs' occupied resources into account. The idea of VMs-GSA is to select VMs resided on $h_j^+ \in H^+$ with smaller traffic and smaller occupied resource to migrate, thereby reducing communication costs. Therefore, the algorithm will be executed as follows. First of all, the VMs on h_j^+ are sorted by $O(v_i^j)C(v_i^j)$ in ascending. Then, starting from v_i^j with the highest $O(v_i^j)C(v_i^j)$ value, select the VMs in sequence and put them in the list W until the host is not overloaded. Finally, these selected VMs are the VMs to be migrated

The sorting needs $O(n \log n)$ and the for-loop of selecting VMs to be migrated needs $O(n)$, where $n = \max_{h_j^+ \in H^+} |\vec{s}^j|$.

Therefore, time complexity of VMs-GSA is $O(n \log n)$. The time complexity of selecting VMs from all overloaded hosts is $O(Mn \log n)$.

An example of the relation of VMs on h_2 is shown in Fig. 2. Suppose $\tau(h_2)$ is 12 and we can know the load of h_2 is $3+2+3+4+7=19$ from Table 1. The result of the sorted VMs on h_2 is $\{v_2^2, v_1^2, v_3^2, v_4^2, v_5^2\}$. Firstly, we select v_2^2 in the sorted list and put it in W . h_2 's load now is $19 - 2 = 17 > 12$, and h_2 is still overloaded. Therefore, we continue to select the v_1^2 and the load of h_2 now is $17 - 3 = 14 < 12$. We continue to select the v_3^2 and the load of h_2 now is $14 - 3 = 11 < 12$. Therefore,

Table 1 VM’s parameters on host h_2

v_i^j	$O(v_i^j)$	$C(v_i^j)$	$O(v_i^j)C(v_i^j)$
1	3	8	24
2	2	11	22
3	3	9	27
4	4	16	64
5	7	26	182

Algorithm 1: VMs-GSA (one host h_j^+)

```

Input: The set of VMs on  $h_j^+$ ,  $V(h_j)$ 
Output: The set of VMs selected from  $h_j^+$ ,  $W$ 
1 Sort VMs by  $O(v_i^j)C(v_i^j)$  in ascending order;
2  $W \leftarrow \emptyset$ ;
3 for  $u = 1$  to  $|V(h_j)|$  do
4   Assume the u-th total communication is  $O(v_u^j)C(v_u^j)$ ;
5   if  $h_j^+$  is overloaded then
6      $W = W \cup \{v_u^j\}$ ;
7   end
8 end
9 Return  $W$ ;
    
```

the VMs selected by VMs-GSA are $W = \{v_2^2, v_1^2, v_3^2\}$. VMs-GSA is implemented on each overloaded host and finally gets all the selected VMs.

4.2 VMM-DAM design

In this section, we consider that the allocation process of finding the destination hosts for VMs to be migrated is modeled as an auction process. The auction process is described in detail below.

4.2.1 Auction model

Auction market consists of three entities, including buyers, sellers, and auctioneer. The buyers refer to the VMs to be migrated and need to buy the resources it needs. The sellers are the underutilized hosts, which sell its idle resources. The third-party auctioneer mainly solves the mapping problem between the buyers and the sellers and their final payment.

Buyer, i.e., VM to be migrated $vm_i \in VM$ submits a bid represented by B_i to auctioneer and B_i can be denoted by a 2-tuple: $(O(vm_i), v_i)$, where $O(vm_i)$ is the size of vm_i 's resource demand and v_i indicates vm_i 's valuation, i.e., The highest price vm_i is willing to pay for using these resources $O(vm_i)$. $B = \{B_1, B_2, \dots, B_N\}$ denotes the bids of all buyers, N is the total number of selected VMs from all overloaded hosts.

Seller, i.e., the underutilized host $h_j^- \in H^-$ submits a bid S_j to the auctioneer. S_j is denoted as a 2-tuple: $(o_j, p_j(m_j))$,

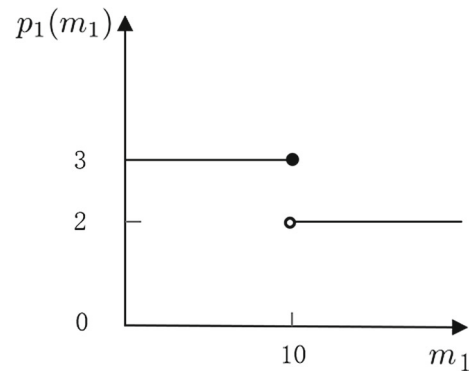


Fig. 3 The unit price $p_1(m_1)$ of h_1^-

where $p_j(m_j)$ denotes the unit price of the resource o_j provided by h_j^- , which is piecewise constant function. $m_j = \sum_{i=1}^N x_{ij} O(vm_i)$ is the quantity sold of h_j^- . Figure 3 is an example of h_1^- unit price $p_1(m_1)$. The $p_1(m_1)$ is 2 when m_1 is more than 10. The unit price decreases as the quantity sold increases, which is a reflection of the discount and can motivate buyers to migrate to this host for lower unit prices. The bids of all sellers are denoted as $S = \{S_1, S_2, \dots, S_M\}$, M is the total number of underutilized hosts.

After collecting the bids B and S submitted by the buyers and sellers, the auctioneer starts to match buyers and sellers and determines the winners of buys and sellers. The matching result generated by the auctioneer is represented by matrix X [see Eq. (2)]. $x_{ij} = 1$ represents buyer vm_i is matched seller h_j^- . vm_i couldn't be migrated, if $\sum_{j=1}^M x_{ij} = 0$. The winner set of buyers is $W^B = \{vm_i | x_{ij} = 1, \forall 1 \leq i \leq N, \exists 1 \leq j \leq M\}$, and the winner set of sellers is $W^S = \{h_j^- | x_{ij} = 1, \forall 1 \leq j \leq M, \exists 1 \leq i \leq N\}$.

After generating the mappings between buyers and sellers, the auctioneer is responsible for determining the payments b_j to h_j^- and charge c_i for vm_i . The utility of vm_i is represented by U_i , which is the difference between the buyer's valuation v_i and payment c_i :

$$U_i = \begin{cases} v_i - c_i & vm_i \in W^B \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

The utility of h_j^- is defined as the difference between the total income b_j from the buyers and the discount ask price \dot{b}_j for his resources:

$$U_j = \begin{cases} b_j - \dot{b}_j & h_j^- \in W^S \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where $\dot{b}_j = m_j p_j(m_j)$. In this auction model, the goal of the auctioneer is to maximize social welfare, which is the sum

of utility of buyers and sellers as follows:

$$SW = \sum_{i=1}^N U_i + \sum_{j=1}^M U_j \tag{9}$$

4.2.2 Allocation algorithm design

In this section, we introduce the allocation algorithm to obtain the mappings between buys and sellers. The idea of this algorithm is that the VMs which selected from the same overloaded host are migrated to the same underutilized host as much as possible, which can reduce the communication cost and lower the cost of buyers in terms of the seller’s unit price.

Let $d_i = v_i / \sqrt{O(vm_i)}$ be vm_i ’s bid density. Firstly, VMs are sorted according to d_i in decreasing order. Let L be the sorted VMs list. Then, select VM from the sorted list L in turn to match seller. Suppose the selected VM currently is vm_i . The algorithm traverses all underutilized hosts to find hosts which can satisfy two conditions of vm_i . One condition is that the host can meet the resource demand of vm_i and another is that the ask price of the host is not less than vm_i ’s valuation v_i . Let $H_i = \{h_j^- | o_j \geq O(vm_i), v_i \geq O(vm_i)p_j(m_j), 1 \leq j \leq M\}$ be the set of the hosts that satisfy the two conditions of vm_i . if $H_i = \emptyset$, vm_i match failed. If $H_i \neq \emptyset$, the host

$$h_{j^*}^- = \arg \max_{h_j^- \in H_i} RI_{ij} \tag{10}$$

is matched with vm_i and $x_{ij} = 1$. j^* indicates the host j that the vm_i matches. where $RI_{ij} = v_i - O(vm_i)p_j(m_j)$ is the revenue increment of vm_i matching h_j^- . Next, The algorithm finds other VMs that reside on the vm_i ’s original host from L and put them in list $L'_i \subset L$, where $L'_i = \{vm_u | s_u^{r(vm_i)} = 1, \forall vm_u \in V(r(vm_i)) \setminus vm_i\}$. $r(vm_i)$ is the source host of vm_i , and $s_u^{r(vm_i)} = 1$ indicates vm_i resided on $r(vm_i)$ is selected to migrate. Then, select VM from L'_i in turn to match seller. Suppose the selected VM currently is vm_u . If $h_{j^*}^-$ satisfies the conditions of vm_u , vm_u is matched with $h_{j^*}^-$ and $x_{uj} = 1$. Otherwise, the algorithm traverses all underutilized hosts to find hosts which satisfy conditions of vm_u . If $H_u = \emptyset$, vm_u failed to match any seller. If $H_i \neq \emptyset$, the host

$$h_{k^*}^- = \arg \max_{h_k^- \in H_u} \varphi_{uk} \tag{11}$$

becomes the destination host of vm_u and $x_{uk} = 1$, where $\varphi_{uk} = \alpha RI_{uk} - \beta Cost_u$. α and β are the weight coefficients. The algorithm selects the next VM in L'_i to match until all VMs in the L'_i are matched. Then, remove L'_i from L and start to match the next VM in L . The algorithm loops until all the VMs in L are matched. Therefore, we finally obtain the matching result matrix X .

The time complexity of sorting buyers is $O(N \log N)$, where N is the number of all buyers. The time complexity of the rest of the algorithm is $O(NM |L'|)$, where $|L'| = \max_{h_j^- \in H^+} |n_j - 1|$, $n_j = \{s_i^j | s_i^j = 1 \wedge s_i^j \in \vec{s}^j, 1 \leq i \leq N, 1 \leq j \leq M\}$.

Therefore, the time complexity of VMs allocation algorithm of VMM-DAM is $O(\max\{N \log N, NM |L'|\})$.

Algorithm 2: VMs Allocation Algorithm

Input: The set of buyers’ bids, B ; The set of sellers’ bids, S

Output: The mappings of buyers and sellers, X

1 Sort buyers by d_i in descending order and put the sorted buyers in L ;

2 **for** $vm_i \in L$ **do**

3 Assume the i -th bid density is d_i ;

4 **if** exist hosts which satisfy two conditions of vm_i **then**

5 $x_{ij} = 1$;

6 Find other VMs that reside on the vm_i ’s original host from L and put them in list L'_i ;

7 **for** $vm_u \in L'_i$ **do**

8 **if** h_j^- satisfy two conditions of vm_u **then**

9 $x_{uj} = 1$;

10 **else**

11 **if** exist hosts which satisfy two conditions of vm_u **then**

12 Choose k that φ_{uk} is maximized;

13 $x_{uk} = 1$;

14 **end**

15 **else**

16 vm_u failed to match host

17 **end**

18 **end**

19 **end**

20 **end**

21 **end**

22 **else**

23 vm_i failed to match host

24 **end**

25 $L = L \setminus L'_i$;

26 **end**

27 **Return** X ;

An example of allocation algorithm is shown below. the relation of hosts in data center is shown in Fig. 4, where h_4 and h_5 are overloaded hosts, h_1, h_2 and h_3 are underutilized host. The bids of VMs are shown in Table 2 and the bids of underutilized hosts are shown in Table 3. Assume the relation of VMs residing on h_5^- is the same as Fig. 2. Thus, the result of the VMs-GSA on h_5^- is $W = \{v_2^5, v_1^5, v_3^5\}$. Assume the result of the VMs-GSA on h_4^- is $W = v_4^4$. We put these VMs in the same set and renumber them. We renumber v_2^5 as vm_1 , v_1^5 as vm_2 , v_3^5 as vm_3 and v_4^4 as vm_4 . Firstly, we rank these VMs by bid density in decreasing order and the sorting result is $L = \{vm_1, vm_2, vm_4, vm_3\}$. Starting from the first vm_1 in sorted list L , traverse all the underutilized hosts. The ask

Table 2 The bids of the buyers

vm_i	v_i	$O(vm_i)$	d_i
1	8	5	3.58
2	9	8	3.18
4	7	5	3.13
3	7	6	2.85

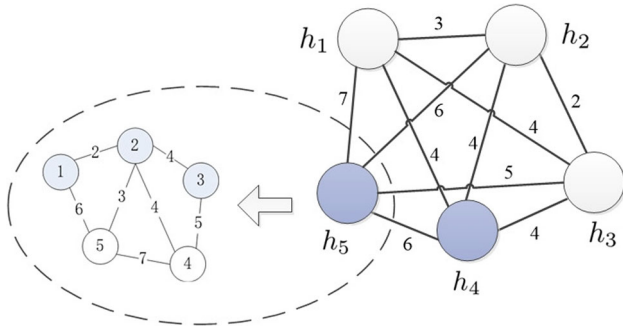


Fig. 4 the relation of hosts in data center

Table 3 The bids of the sellers

h_j^-	$p_j(m_j)$	o_j
1	$\begin{cases} 1.1 & m_3 > 5 \\ 1.5 & \text{otherwise} \end{cases}$	7
2	$\begin{cases} 0.8 & m_4 > 8 \\ 1.3 & \text{otherwise} \end{cases}$	13
3	$\begin{cases} 1 & m_5 > 5 \\ 2.0 & \text{otherwise} \end{cases}$	12

price of h_1^- for vm_1 is $5 * 1.5 = 7.5 < 8$ and resources $o_1 = 7 > 5$. Therefore, In the same way, we can obtain the set of satisfy the conditions of vm_1 is $H_1 = \{h_1^-, h_2^-\}$. So, $RI_{11} = 8 - 7.5 = 0.5$ and $RI_{12} = 8 - 1.3 * 5 = 1.5$. Therefore, vm_1 is matched h_2^- and $x_{12} = 1$. Then, find other VMs that reside on the vm_1 's original host from L and put them in list L'_1 , $L'_1 = \{vm_2, vm_3\}$. We match the VMs in L'_1 in turn. The ask price of h_2^- for vm_2 is $8 * 0.8 = 6.4 < 7$ and resources o_2 is $13 - 5 = 8$. Therefore, h_2^- satisfies the conditions of vm_2 and $x_{22} = 1$. The h_2^- ' resources o_2 now is $13 - 5 - 8 = 0 < 6$; thus, h_2^- can't satisfy the conditions of vm_3 . Then, the algorithm traverses the rest underutilized hosts and the set of satisfy the conditions of vm_3 is $H_1 = \{h_1^-, h_3^-\}$. We assume that the values of α and β are both 1. If vm_3 , i.e., v_3^5 , is migrated to h_1^- ,

$$\begin{aligned} \varphi_{31} &= RI_{31} + Cost_3 \\ &= RI_{31} + (W'(vm_3, v_2^5)W(h_1^-, h_2^-) \\ &\quad + W'(vm_3, v_4^5)W(h_1^-, h_5^-)) \\ &= 7 - (1.1 * 6) + 4 * 3 + 5 * 7 \\ &= 47.4 \end{aligned}$$

In a similar way, we can calculate the φ_{33} of vm_3 migrate to h_3^- is 34. Therefore, vm_3 matches h_3^- and $x_{33} = 1$. Remove L'_1 from L and the result is $L = \{1, 4\}$. The algorithm allocates vm_4 next. The set of satisfy the conditions of vm_4 is only h_3^- . Therefore, vm_4 is matched with h_3^- and $x_{43} = 1$.

The matching result of buyers and sellers in this example

$$\text{is } X = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

4.2.3 Scheme of payment

After the mapping of buyers and sellers is generated, we discuss the payment of buyers and sellers. We employ ‘‘Vickery’’ price to the payment of buyers (Zaman and Grosu 2010). The ‘‘Vickery’’ price of vm_i is defined as the value of the size of vm_i 's demand multiplied by the highest bid density of vm_i among losers who would become the winner if vm_i would not participate in the auction. The winner vm_i 's payment c_i to its matched seller is defined as the maximum between the Vickery price and the asking price of the seller as follows.

$$c_i = \begin{cases} \max\{d_i\sqrt{O(vm_i)}, O(vm_i)p_j(m_j)\} & \exists vm_l \in (VM \setminus W^B) \\ O(vm_i)p_j(m_j) & \text{otherwise} \end{cases} \quad (12)$$

The payment \dot{b}_j to the winner h_j^- is

$$\dot{b}_j = \sum_{i=1}^N x_{ij}c_i \quad (13)$$

which is the sum of payments of all the VMs that matched the h_j^- .

4.2.4 Analysis

In this section, we prove that VMM-DMA achieves individual rationality, budget balance and truthfulness.

Theorem 1 VMM-DMA achieves individual rationality.

Proof The final price for winner $vm_i \in W^B$ is shown in (12). If vm_i 's payment is ‘‘Vickery’’ price: $U_i = v_i - d_i\sqrt{O(vm_i)} = v_i - (v_i/\sqrt{O(vm_i)})\sqrt{O(vm_i)} > v_i - (v_i/\sqrt{O(vm_i)})\sqrt{O(vm_i)} = v_i - v_i = 0$. if vm_i 's payment is the ask price of its matched seller h_j^- : $U_i = v_i - O(vm_i)(p_j(m_j)) \geq 0$. Because h_j^- must satisfy the conditions that seller's asking price is less than the buyer's valuation.(See line 4, line 8 and line 11 in Algorithm 2). Therefore, the buyers achieve individual rationality. The final price for winner $h_j^- \in W^S$ is the sum of the payments of all buyers matching him. The buyer's payment will not exceed

the seller's asking price regardless of the price of "Vickery" or the seller's ask price. Therefore, the sellers achieve individual rationality. \square

Theorem 2 *VMM-DMA achieves budget balance.*

Proof The payment to seller is $\hat{b}_j = \sum_{i=1}^N c_i$, which is the sum of the payments of all the VMs that matched him. All sellers are $\sum_{j=1}^M \hat{b}_j = \sum_{j=1}^M \sum_{i=1}^N c_i$, which is the total payments of all winning buyers. So, the VMM-DAM achieves budget balance. \square

Theorem 3 *VMM-DMA achieves truthfulness.*

Proof For a winning buyer vm_i , if exists the loser $vm_l \in VM \setminus W^B$, the price vm_i pays is "Vickery" price and Vickery is truthful (Sun et al. 2015). If there is no loser, vm_i 's payment to his matched seller h_j^- is $O(vm_i)(p_j(m_j))$, which is independent of the bids. Because the demanded of vm_i is truthful. Therefore, VM-DMA achieves truthfulness. \square

5 Evaluation results

In this section, we simulated VMs migration in the data center. Four performance indicators are considered as follows: (i) the selected VM's average total communications generated by VMs-GSA, (ii) success rate of VM migration, which is the ratio of the number of winning buys to the total number of VMs selected from all overloaded hosts, (iii) average load of overloaded hosts (before and after migration), (iv) the overload ratio of the host (before and after the migration).

The simulation code is written in Java, using Eclipse development tool, and running on a local computer (Intel Core 2.4 GHZ, 8 GB of RAM).

The experimental setup is shown below. The demand $O(vm_i)$ of vm_i is randomly generated within [8, 15], and the valuation v_i is randomly selected in the interval [35, 40]. Edges between VMs randomly generated, and weights $W'(v_i, v_j)$ and $W(h_i, h_j)$ are randomly generated within [1, 21]. The threshold $\tau(h_j)$ of h_j is randomly selected in the interval [40, 47], and the safety limit $sl(h_j)$ of h_j is generated randomly from interval [32, 39]. The unit price $p_j(m_j)$ of h_j before discount is randomly generated in the interval [2, 3] and after discount is randomly generated in the interval [1, 2]. There are 8 hosts in Fig. 6 and 26 VMs in Figs. 7 and 8. In order to reduce the impact of the randomness of the simulated data, experiment run over 200 times.

We used two algorithms including enumeration algorithm (best algorithm) and random algorithm to compare with VMs-GSA in terms of the communications of VMs in Fig. 5. The best algorithm is to list the combination of all the VMs

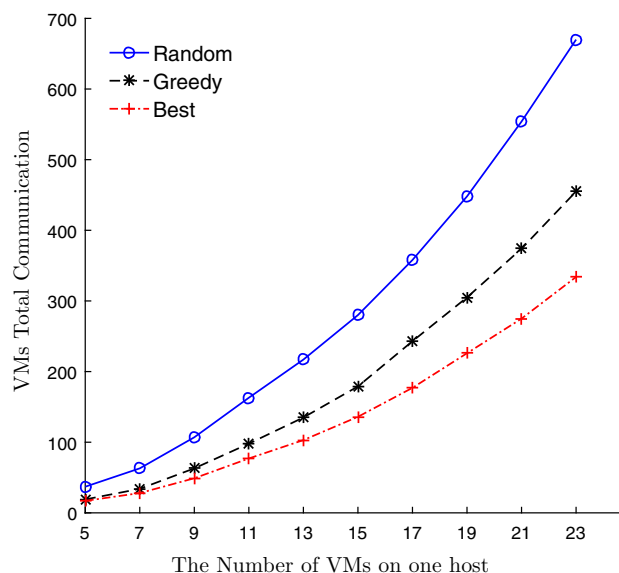


Fig. 5 Comparison of the VM total communications generated by three VM selection algorithm

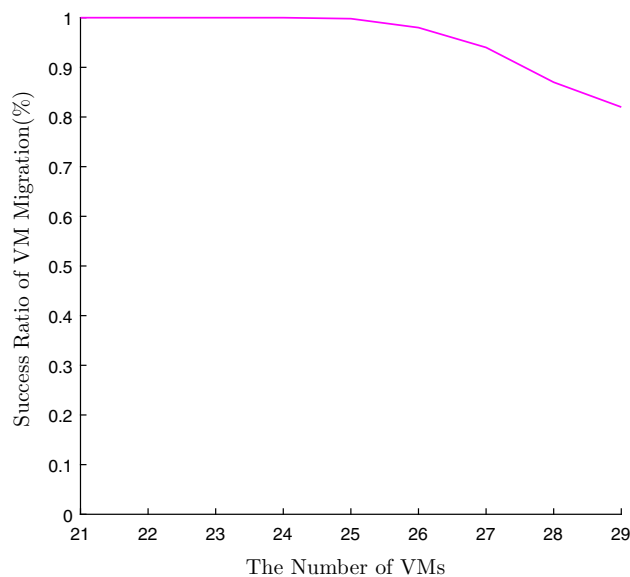


Fig. 6 The success ratio of VM migration

and find out the combination that has the least amount of traffic and makes the host a non-overloaded state. The random algorithm is to randomly select the VM from the overloaded host until the host is not overloaded. Figure 5 shows that compared with the random algorithm, the total amount of communication generated by VMs-GSA reduces about 35%. When the number of VMs is not large, the total amount of communication generated by VMs-GSA is approximately similar to the enumeration algorithm. When the number of VMs is large, the total amount of communication generated by the greedy algorithm increases about 30% than the opti-

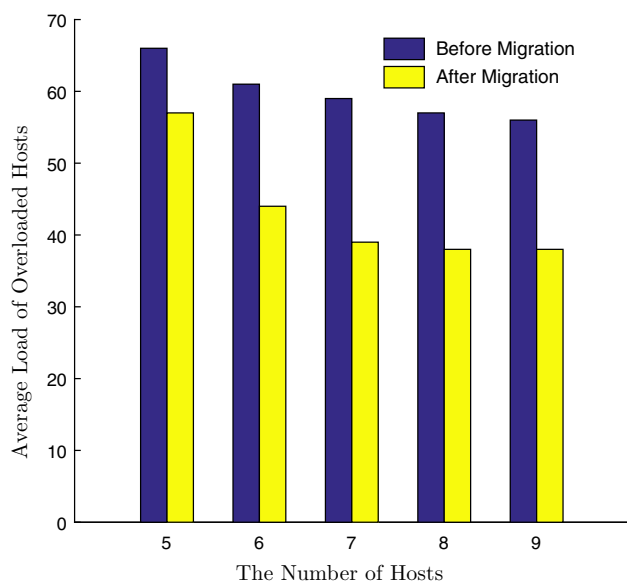


Fig. 7 Comparison of overloaded hosts load (before and after migration)

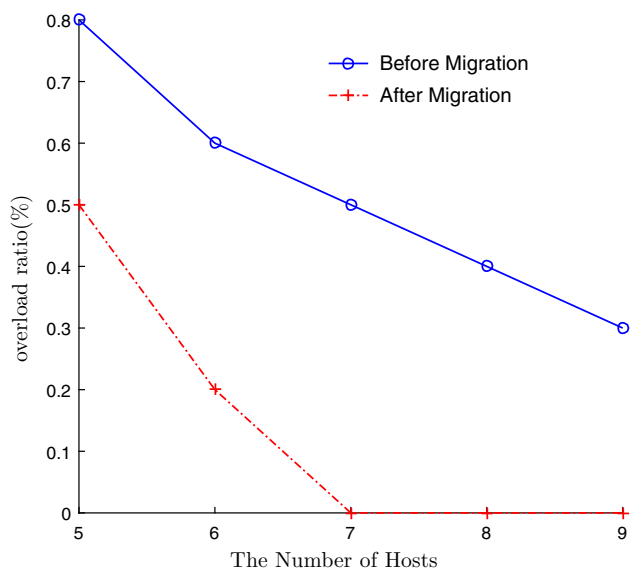


Fig. 8 comparison of hosts' overload ratio (before and after migration)

mal algorithm. However, the computational complexity of the enumeration algorithm is huge. Communication costs are reduced compared to algorithms in Vu and Hwang (2014).

Figure 6 shows VM migration is almost 100% successful when the number of VMs is in the range of 21–25. However, As the number of VMs increases, the number of hosts is constant. The success rate of VM migration is decreasing when the number of virtual machines increases to 26, because the resources of these 8 hosts are limited. Figure 7 shows the average load of overloaded hosts before and after migration. The load of overloaded hosts drops significantly to the threshold after the migration. When the number of

hosts is greater than 6, the hosts in the data center are basically fully utilized. Figure 8 shows the overload rate of the host is reduced by 50% after the migration. When the number of hosts is greater than 7, the overload rate of the host is 0. Experiments proved that the proposed VM migration algorithm based on heuristic reduces communication costs and is efficient.

6 Conclusion

VM migration can avoid exception in the host in the data center due to overload and achievement of load balancing. In this paper, we investigated traffic-aware heuristic VM migration problem in the data center which mainly focuses on reducing communication cost in the process of VM migration. The VM migration algorithm consists of two parts. We designed VMs-GSA in the first part to select VMs with the low communication costs, which greatly reduce the communication generated by VM migration. VMM-DAM is applied to the second part of VM migration to match hosts with the low communication costs and high revenue increment for VMs to be migrated. VMM-DAM allowed that sellers, i.e., the underutilized hosts, provided a discount to incent buyers to trade with himself for a lower unit price, which can make the host fully utilized as much as possible. The allocation algorithm of VMM-DAM made the selected VM from the same host migrate to the same host as much as possible, which reduces communication costs. The VM migration algorithm proposed in this paper achieves the load balancing of hosts and reduces the communication cost in the process of VM migration. Simulation experiment has shown that the VMs-GSA and VMM-DAM are traffic-aware and effective.

Acknowledgements This work was supported by the National Nature Science Foundation of China under Grant 61872313 and Grant 61472344, in part by the Innovation Foundation for graduate students of Jiangsu Province under Grant CXLX12 0916, in part by the Natural Science Foundation of the Jiangsu Higher Education Institutions under Grant 14KJB520041, in part by the Advanced Joint Research Project of Technology Department of Jiangsu Province under Grant BY2015061-06 and Grant BY2015061-08, and in part by the Yangzhou Science and Technology under Grant YZ2017288 and Grant YZ2016245 and Yangzhou University Jiangdu High-end Equipment Engineering Technology Research Institute Open Project under Grant YDJD201707.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

- Abdel-Basset M, Abdle-Fatah L, Sangaiah AK (2018) An improved lvy based whale optimization algorithm for bandwidth-efficient virtual machine placement in cloud computing environment. *Clust Comput* 1:1–16
- Azougaghe A, Oualhaj OA, Hedabou M (2017) Many-to-one matching game towards secure virtual machines migration in cloud computing. In: International conference on advanced communication systems and information security, pp 1–7
- Beloglazov A, Abawajy J, Buyya R (2012) Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Gener Comput Syst* 28(5):755–768
- Chen J, Liu W, Song J (2012) Network performance-aware virtual machine migration in data centers. In: The third international conference on cloud computing, GRIDs, and virtualization Cloud Comput. Nice, France, pp 65–71
- Gao C, Wang H, Zhai L, Gao Y, Yi S (2017) An energy-aware ant colony algorithm for network-aware virtual machine placement in cloud computing. In: IEEE international conference on parallel and distributed systems, pp 669–676
- Goldberg RP (1974) Survey of virtual machine research. *Computer* 7(6):34–45
- Heller B (2010) Saving energy in data center networks. *Nsdi10 Apr*
- Huang J, Wu K, Moh M (2014) Dynamic virtual machine migration algorithms using enhanced energy consumption model for green cloud data centers. In: International conference on high performance computing and simulation, pp 902–910
- Kansal NJ, Chana I (2016) Energy-aware virtual machine migration for cloud computing—a firefly optimization approach. *J Grid Comput* 14(2):327–345
- Li Y, Lu H, Nakayama Y, Kim H, Serikawa S (2018) Automatic road detection system for an airland amphibious car drone. *Future Gener Comput Syst* 85:51–59
- Lu H, Li Y, Mu S (2017a) Motor anomaly detection for unmanned aerial vehicles using reinforcement learning. *IEEE Internet Things J* PP(99):1–1
- Lu H, Li B, Zhu J, Li Y (2017b) Wound intensity correction and segmentation with convolutional neural networks. *Concurr Comput Pract Exp* 29(6):e3927. <https://doi.org/10.1002/cpe.3927>
- Lu H, Li Y, Chen M, Kim H, Serikawa S (2018a) Brain intelligence: go beyond artificial intelligence. *Mob Netw Appl* 23(2):368–375
- Lu H, Li Y, Uemura T, Kim H, Serikawa S (2018b) Low illumination underwater light field images reconstruction using deep convolutional neural networks. *Future Gener Comput Syst* 82:142–148
- Marahatta A, Wang Y, Zhang F, Sangaiah AK, Tyagi SKS, Liu Z (2018) Energy-aware fault-tolerant dynamic task scheduling scheme for virtualized cloud data centers. *Mob Netw Appl*. <https://doi.org/10.1007/s11036-018-1062-7>
- Medhane DV, Sangaiah AK (2018) Pcca: Position confidentiality conserving algorithm for content-protection in e-governance services and applications. *IEEE Trans Emerg Top Comput Intell* 2(3):194–203
- Meng X, Pappas V, Zhang L (2010) Improving the scalability of data center networks with traffic-aware virtual machine placement. In: Proceedings IEEE INFOCOM, San Diego, 14–19 Mar 2010. IEEE, pp 1–9. <https://doi.org/10.1109/INFCOM.2010.5461930>
- Reguri VR, Kogotam S, Moh M (2016) Energy efficient traffic-aware virtual machine migration in green cloud data centers. In: IEEE international conference on big data security on cloud, pp 268–273
- Rings T, Caryer G, Gallop JR, Grabowski J, Kovacicova T, Schulz S, Stokesrees I (2009) Grid and cloud computing: opportunities for integration with the next generation network. *J Grid Comput* 7(3):375–393
- Shrivastava V, Zerfos P, Lee K, Jamjoom H, Liu Y, Banerjee S (2011) Application-aware virtual machine migration in data centers. In: Proceedings IEEE INFOCOM, Shanghai, 10–15 Apr 2011. IEEE, pp 66–70. <https://doi.org/10.1109/INFCOM.2011.5935247>
- Sun Z, Zhu Z, Chen L, Xu H, Huang L (2015) A combinatorial double auction mechanism for cloud resource group-buying. In: Performance computing and communications conference, pp 1–8
- Sun J, Zhu G, Sun G, Liao D, Li Y, Sangaiah AK, Ramachandran M, Chang V (2018) A reliability-aware approach for resource efficient virtual network function deployment. *IEEE Access* PP(99):1–1
- Tao F, Li C, Liao TW (2016) Bgm-bla: a new algorithm for dynamic migration of virtual machines in cloud computing. *IEEE Trans Serv Comput* 9(6):910–925
- Vu HT, Hwang S (2014) A traffic and power-aware algorithm for virtual machine placement in cloud data center. *Int J Grid Distrib Comput* 7(1):350–355
- Wang L, Von Laszewski G, Younge A (2010) Cloud computing: a perspective study. *New Gener Comput* 28(2):137–146
- Xu X, He L, Lu H, Gao L, Ji Y (2018) Deep adversarial metric learning for cross-modal retrieval. *World Wide Web*. <https://doi.org/10.1007/s11280-018-0541-x>
- Zaman S, Grosu D (2010) Combinatorial auction-based allocation of virtual machine instances in clouds. In: IEEE second international conference on cloud computing technology and science, pp 127–134
- Zhang SM, Sangaiah AK (2018) Reliable design for virtual network requests with location constraints in edge-of-things computing. *Eurasip J Wirel Commun Netw* 2018(1):65
- Zhang W, Han S, He H, Chen H (2017) Network-aware virtual machine migration in an overcommitted cloud. *Future Gener Comput Syst* 76:428–442

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.