A Double Auction VM Migration Approach

Jinjin Wang, Yonglong Zhang, Junwu Zhu, and Yi Jiang

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. In this case, the concept of cloud computing is proposed. Cloud computing is a further development of distributed computing, parallel processing, and grid computing [13]. It is a system based on Internet computing that can provide hardware services, infrastructure services, platform services, software services, and storage services to a variety of Internet applications [6, 7].

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 lot 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 [3]. However, load balancing is a challenge. There are overloaded hosts and underutilized hosts. When too many VMs are running on one host, the host will become overloaded and cause exceptions. VM migration can solve this problem. In this case, we must select one or more VMs to migrate once the host become overloaded and find the best destination host for these selected VMs which to be migrated. Double auction is widely used in the field of artificial intelligence to solve the problem of resource competition [5, 14]. In this paper,

J. Wang · Y. Zhang · Y. Jiang
College of Information Engineering, Yangzhou University, Yangzhou, Jiangsu, China

J. Zhu (✉)
College of Information Engineering, Yangzhou University, Yangzhou, Jiangsu, China

Department of Computer Science, University of Guelph, Guelph, ON, Canada
e-mail: jwzhu@yzu.edu.cn

H. Lu, L. Yujie (eds.), *2nd EAI International Conference on Robotic Sensor Networks*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-17763-8_13

we proposed a VM migration algorithm based on double auction, which considers communication cost.

The rest of this paper is organized below. Section 2 reviews some related works. In Sect. 3, we detail system model. We introduce VMs-GSA and VMM-DAM in Sect. 4. Section 5 is simulation results and conclusion.

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. In order to improve energy efficiency of the data center, Tao et al. [10] 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 [10] targets these three factors has two parts. The first part generates bucket codes. 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. Beloglazov et al. [2] proposed energy-aware allocation heuristics which provides the resources of servers or hosts to client applications while guaranteeing quality of service (QoS). In [8], the authors proposed three VM migration schemes which take the traffic factor and VM clustering into account, which are the improvement of papers [12]. VM clustering includes two steps. First step forms the VM graph on same host, in which vertexes are VMs and edges are communications between VMs. Then, form clusters. In addition, many studies have considered communication factors in the migration of VMs, such as [4] and [11].

Zhang et al. [15] apply the genetic algorithm (GA) and artificial bee colony (ABC) to the problem of VM migration problem and aim to find an approximate optimal solution through repeated iterations. The GA first generated a population of PopSize chromosomes which is a vector represented 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. In [1], 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.

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 $H = \{h_{1,}h_{2,...,}h_k\}$ 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 the communication distance.

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. $V(h_j) = \{v_1^j, v_2^j, \ldots, v_n^j\}$ is the set of VMs residing on h_j. The communication relation of VMs residing on h_j is represented by an undirected graph $G = (V, E)$, where $V = V(h_j)$ and $E = \{(v_i^j, v_l^j) | \text{where } v_i^j \text{ and } v_l^j \text{ exist communications}\}$ is the set of edges. The weight $W'(v_i^j, v_l^j)$ represents VM traffic. $\forall v_l^j \in V(h_j)$ that communicate with v_i^j are called v_i^js 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. $C(v_i^j) = \sum_{v_l^j \in N(v_i^j)} W'(v_i^j, v_l^j)$ is the traffic of v_i^j. The residing set of overloaded hosts is represented by $H^+ = \{h_j | \sum_{i \in V(h_j)} O(i) > \tau(h_j), h_j \in H\}$ and $H^- = \{h_j | \sum_{i \in V(h_j)} O(i) \leq sl(h_j), h_j \in H\}$ 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.

Our work is mainly focused on the communication costs of VMs migration in data center. Let mapping function $\sigma : VM \rightarrow H^-$ denotes the mapping between VMs and underutilized hosts. Let vm_i denote the selected VM from overloaded host. The communication cost that vm_i matches $h_j^- \in H^-$ is defined as below

$$Cost_i = \sum_{v_l \in N(vm_i)} W'(vm_i, v_l) W(h_{\sigma(i)}^-, h_{\sigma(l)}^-) \tag{1}$$

4 VM Migration Algorithm

In this section, VM migration algorithm based on heuristic consists of two parts: (1) selecting VMs from overloaded hosts to migrate and we proposed VMs-GSA to determine these VMs, (2) obtaining the mapping between VMs and underutilized hosts and we employed VMM-DAM to obtain it.

4.1 VMs-GSA Design

The idea of VMs-GSA is to select the 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 residing on h_j^+ are sorted by $O(v_i^j)C(v_i^j)$ in the ascending order. 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.

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: for $u = 1$ to $|V(h_j)|$ do
3:     Assume the u-th total communication is $O(v_u^j)C(v_u^j)$
4:     if $h_j^+$ is overloaded then
5:         $W = W \cup \{v_u^j\}$
6:     end if
7: end for
8: return $W$
```

4.2 VMM-DAM Design

Auction Model We consider that the allocation process of finding the destination hosts for VMs which to be migrated is modeled as an auction process. Auction market consists of three entities. The buyers refer to the VMs which to be migrated. The sellers are the underutilized hosts. The third-party auctioneer mainly solves the mapping problem and final payment. Buyer $vm_i \in VM$ 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. $B = \{B_1, B_2, \ldots, B_N\}$ denotes the bids of all buyers. Seller $h_j^- \in H^-$ submits a bid S_j to the auctioneer. S_j is denoted as a 2-tuple: $(o_j, p_j(m_j))$, 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 is the quantity sold of h_j^-. The bids of all sellers are denoted as $S = \{S_1, S_2, \ldots, S_M\}$.

Allocation Algorithm Design Let $d_i = v_i/\sqrt{O(vm_i)}$ be vm_i's bid density. Firstly, we sort the VMs according to d_i in the descending 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 host h_{j*}^- which satisfies two conditions of vm_i with maximum revenue increment $RI_{ij} = v_i - O(vm_i)p_j(m_j)$ is matched with vm_i. One condition is that the host can meet demand of vm_i and another is that the ask price of the host is more than vm_i's valuation. Next, the algorithm finds other VMs that reside on the vm_i's original host from L and put them in list $L_i' \subset L$. 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*}^-. Otherwise, the algorithm finds other hosts which satisfy conditions of vm_u. The host with maximum $\varphi_{uk} = \alpha RI_{uk} - \beta Cost_u$ becomes the destination host of vm_u, where α and β are the weight coefficients. The algorithm loops until all the VMs in L are matched.

Algorithm 2 VMs allocation algorithm

```
Input: The set of buyers' bids, $B$; The set of sellers' bids, $S$
Output: The matrix which is the mappings result 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:     if exist hosts which satisfy two conditions of $vm_i$ then
 4:         Choose $j$ with the greatest $RI_{ij}$ to matched $vm_i$ and $x_{ij} = 1$
 5:         Find other VMs that reside on the $vm_i$'s original host from $L$ and put them in list $L_i'$,
 6:         for $vm_u \in L_i'$ do
 7:             if $h_j^-$ satisfy two conditions of $vm_u$ then
 8:                 $h_j^-$ is matched with $vm_u$ and $x_{uj} = 1$
 9:             else
10:                 if exist hosts which satisfy two conditions of $vm_u$ then
11:                     Choose $k$ with the greatest $\varphi_{uk}$ to matched $vm_u$ and $x_{uk} = 1$
12:                 else
13:                     $vm_u$ failed to match host
14:                 end if
15:             end if
16:         end for
17:     else
18:         $vm_i$ failed to match host
19:     end if
20:     $L = L \backslash L_i'$
21: end for
22: return $X$
```

Scheme of Payment We employ "Vickery" price to the payment of buyers [9]. 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_l among losers who would become the winner if vm_i would not participate in the auction. The winner vm_i' payment c_i to its matched seller is $c_i = \max\{d_l\sqrt{O(vm_i)}, O(vm_i)p_j(m_j)\}$. The payment $\dot{b}_j$ to the winner h_j^- is the sum of payments of all the VMs that matched h_j^-.

5 Results and Conclusion

In this section, we simulated VMs migration in the data center and summarized the paper. Simulation experiment has shown that compared with the random algorithm, the total amount of communication generated by VMs-GSA reduces about 35% and the VMs-GSA is approximately similar to the enumeration algorithm. However, the computational complexity of the enumeration algorithm is very huge. The load of overloaded hosts drops significantly to the threshold after the migration by VMM-DAM and VM migration is almost 100% successful and the communication generated by VMM-DAM is small.

We investigated traffic-aware VM migration problem in data center. 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 reduces 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. 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 61170201, Grant 61070133, 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 Yangzhou University Jiangdu High-end Equipment Engineering Technology Research Institute Open Project under Grant YDJD201707.

References

1. Azougaghe, A., Oualhaj, O. A., & 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). Piscataway: IEEE.
2. Beloglazov, A., Abawajy, J., & Buyya, R. (2012). Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Generation Computer Systems, 28*(5), 755–768.
3. Goldberg, R. P. (1974). Survey of virtual machine research. *Computer, 7*(6), 34–45.
4. Kansal, N. J., & Chana, I. (2016). Energy-aware virtual machine migration for cloud computing—a firefly optimization approach. *Journal of Grid Computing, 14*(2), 327–345.
5. Lu, H., Li, B., Zhu, J., & Li, Y. (2017). Wound intensity correction and segmentation with convolutional neural networks. *Concurrency and Computation Practice and Experience, 29*(6), e3927.
6. Lu, H., Li, Y., Chen, M., Kim, H., & Serikawa, S. (2018). Brain intelligence: Go beyond artificial intelligence. *Mobile Networks and Applications, 23*(2), 368–375.
7. Lu, H., Li, Y., & Mu, S. (2018). Motor anomaly detection for unmanned aerial vehicles using reinforcement learning. *IEEE Internet of Things Journal, 5*(4), 2315–2322.
8. Reguri, V. R., Kogatam, 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). Piscataway: IEEE.
9. Sun, Z., & Zhu, Z. (2015). A combinatorial double auction mechanism for cloud resource group-buying. In *2014 IEEE 33rd International Performance Computing and Communications Conference (IPCCC)* (pp. 1–8). Piscataway: IEEE.
10. Tao, F., Li, C., & Liao, T. W. (2016). BGM-BLA: A new algorithm for dynamic migration of virtual machines in cloud computing. *IEEE Transactions on Services Computing, 9*(6), 910–925.
11. Tso, F. P., Hamilton, G., Oikonomou, K., & Pezaros, D. P. (2013). Implementing scalable, network-aware virtual machine migration for cloud data centers. In *IEEE Sixth International Conference on Cloud Computing* (pp. 557–564). Piscataway: IEEE.

12. Vu, H., & Hwang, S. (2014). A traffic and power-aware algorithm for virtual machine placement in cloud data center. *International Journal of Grid and Distributed Computing, 7*(1), 21–32.
13. Wang, L., Laszewski, G. V., & Younge, A. (2010). Cloud computing: A perspective study. *New Generation Computing, 28*(2), 137–146.
14. Xu, X., He, L., Lu, H., Gao, L., & Ji, Y. (2018). Deep adversarial metric learning for cross-modal retrieval. *World Wide Web-Internet and Web Information Systems, 22*(2), 657–672.
15. Zhang, W., Han, S., He, H., & Chen, H. (2017). Network-aware virtual machine migration in an overcommitted cloud. *Future Generation Computer Systems, 76*, 428–442.