

# A Classification-Based Virtual Machine Placement Algorithm in Mobile Cloud Computing

**Yuli Tang, Yao Hu and Lianming Zhang**

College of Physics and Information Science

Key Laboratory of Internet of Things Technology and Application

Hunan Normal University, Changsha 410081, China

[email:423441058@qq.com, huyao9403@163.com, zlm@hunnu.edu.cn]

\*Corresponding author: Lianming Zhang

*Received December 10, 2015; revised February 20, 2016; accepted March 26, 2016;  
published May 31, 2016*

---

## **Abstract**

In recent years, cloud computing services based on smart phones and other mobile terminals have been a rapid development. Cloud computing has the advantages of mass storage capacity and high-speed computing power, and it can meet the needs of different types of users, and under the background, mobile cloud computing (MCC) is now booming. In this paper, we have put forward a new classification-based virtual machine placement (CBVMP) algorithm for MCC, and it aims at improving the efficiency of virtual machine (VM) allocation and the disequilibrium utilization of underlying physical resources in large cloud data center. By simulation experiments based on *CloudSim* cloud platform, the experimental results show that the new algorithm can improve the efficiency of the VM placement and the utilization rate of underlying physical resources.

---

**Keywords:** mobile cloud computing, virtual machine placement, resource equilibrium, task response time, placement efficiency, wireless network

---

This research is supported in part by the grant from the National Natural Science Foundation of China (No.61572191 and No.61402170), the Hunan Provincial Science and Technology Program Project of China (No.2013FJ4051), and the Hunan Provincial Education Department Scientific Research Fund of China (No.13B065).

## 1. Introduction

In recent years, with the integration and development of the computer and the network communication technology, smart phones and mobile Internet are developing rapidly, and cloud computing services, based on smart phones or other mobile terminals, arise at the historic moment. The existing computing power and storage capacity of terminals and some services of mobile Internet have been more difficult to meet the needs of users. However, cloud computing has advantages of mass storage capacity and high-speed computing power, and it can meet various application needs of mobile users. So cloud computing applications of mobile terminals will be an inevitable trend in future development [1] [2].

To meet the needs of different types of mobile users, the development in mobile cloud computing (MCC) is now so rapid. Paper [3] has designed a new architecture for emerging heterogeneous mobile network. Paper [4] has proposed a framework for resource allocation to the mobile applications. A lightweight service migration framework for computational offloading in the deployment of runtime distributed platform has been proposed in Ref. [5]. The latest developments in MCC also boost a growing interest in the deployment of vehicular cyber-physical systems (VCPS) and Things of Internet [3], [6]. Enhancing the energy efficiency and battery power of mobile devices is also the challenge of MCC [7], [8].

For cloud computing and MCC, virtualization is a key technology, so the virtual machine (VM) is a kernel component of virtualization. Virtualization is a mechanism to create virtual instances of equipments or resources. Under a performance guarantee, virtualization can run multiple operating systems on one host at the same time. The VM placement and migration technology can be used in resource equilibrium, server consolidation, and hotspot migrate, etc [9][10]. Therefore, the VM placement problem is the hinge of scheduling and management in cloud data center. It is to decide to how put the VMs on which physical host to work, thus it can make the most efficient use of the available resources, reduce energy consumption and ensure the quality of service, etc. So a good VM placement algorithm has very profound significance.

According to the dynamic changes of the load application, and through the VM migration and dynamic local-scaling technology which maps the virtual machine's resources to the physical hardware resources, and the VM placement technology aims to gain the improvement of the VM service response performance, resource utilization, energy efficiency, and to maintain the stability of the VM service as much as possible. The existing algorithms for the VM placement are mainly to optimize and calculate the effectiveness and efficiency of using resources. The VM is allowed to share computing power and other resources of physical host, and maximize use of computing resources. Another method is to

calculate the minimum cost of computing resources, or reduce energy consumption, etc.

In this paper, we will discuss the problem of the VM placement for MCC, which provides limited-bandwidth and other limited resources. The technical aim of the paper is to propose an algorithm for the VM placement in MCC. In comparison with the existing literature on similar work, this paper has the following major contributions:

- To classify the VMs according to request resources, and to propose a new algorithm for virtual machine placement in MCC.
- To improve the quality of user access and reduce the response time for applications, by reducing the number of scanning for the VMs, and completing the first optimization of the VM scheduling.
- To improve the equilibrium utilization of physical machine resources to avoid physical machine unnecessary wastage of resources.

In the following sections, we first discuss existing research work on the algorithms for the VM placement in MCC in Section 2. Section 3 presents a classification-based virtual machine placement (CBVMP) algorithm in MCC. The experimental results and performance analysis of the algorithm are presented in Section 4. The paper is concluded in Section 5.

## 2. Related works

The existing algorithms for the VM placement include random packing [11], integer programming [12], constrained programming [13][14], randomized greedy [15], simulated annealing [16], genetics and evolution [19], ant colony algorithms and so on. These algorithms are trying to solve disequilibrium in resource use of physical hosts in different dimensions, high consumption, and high rate of resource waste, and so on [20]. Therefore, in general, the VM placement problem needs to consider the influence of network factors, the resource needs, the VM hardware conditions, and the expected allocation target. For resource needs, the VMs not only consider one kind of resource needs, but also take a variety of resource needs into account, and undoubtedly, CPU and memory are the two most important resources. For high energy consumption problem, generally we should consider server-side constraints. At present, there are already some heuristic methods for the VMs integration problem. These methods, based on the needs of the VM resources and the capacity of physical hosts, decide which host should be a VM instance put on, and then achieve the purpose of using the least physical hosts [21][22]. Van et al. [14] define the VM batch deployment in cloud computing as a constraint satisfaction problem, and it is a typical NP-hard problem. Hyser et al. [23] point out that VMs allocation is a complex problem in large-scale data center. An improved best-fit decreasing algorithm is proposed in Ref. [24], and this is used for the power-aware VM allocation and the heuristic method of adaptive migration based on threshold. Paper [25] puts forward a new optimization problem called min cut ratio-aware VM placement (MCRVMP) and introduces several heuristics to solve it.

A scheduling algorithm is proposed based on scores in Ref. [26], and it puts each VM in the physical machine of the maximum score, so it is actually a crawler algorithm. Van et al. [14] propose a constrained programming algorithm that under the constraints that meet service-level agreement and operation cost, it can reduce the number of physical hosts, and at the same time make the maximize global utility function.

The disequilibrium resource utilization of physical hosts means, at the allocation of the VMs, the data center has disequilibrium utilization between different dimensions of resources (such as CPU and bandwidth) in physical machines, and then it causes a resource waste problem. Sun et al. [27] establish a double objective optimization model, and propose a VM allocation algorithm based on matrix transformation. This algorithm firstly sets up a VM request queue matrix, cluster matrix and initialization allocation matrix, and then through matrix transformation to get optimal results. Wang et al. [27] put forward a multi-objective VM allocation algorithm based on genetic evolution, under the consideration of genetic evolutionary algorithm, through heuristic iterative variation to do virtualized allocation. Lu et al. [29] propose a two-stage VM merge algorithm. In first stage, they use a polynomial approximate resource equilibrium scheme to adjust the allocation of the different VMs, and in second stage, through the queuing model to adjust the load difference between underlying physical resources to reduce the disequilibrium.

With the development of MCC, how to carry out the placement of the VMs is needed to consider special environments of MCC. The existing VM placement strategies focus on resource utilization and effectiveness, but often ignore the influence of networks to the user experience, and these allocation strategies will lead to assign the VMs which run applications far from the related host.

### 3. A classification-based virtual machine placement algorithm

In this section, we firstly present the system architecture for MCC, and it takes users' accessing quality and resource equilibrium into account. And then, we propose a new classification-based virtual machine placement (CBVMP) algorithm. Firstly, the algorithm classifies the VMs that request resources, and completes the VM scheduling objective optimization for the first time. Secondly, based on the multidimensional natures of physical host resources (CPU, memory, etc.), the algorithm matches the resources that VMs need with the resources of the rest physical machines. Finally, the algorithm finds the most appropriate VM to complete resource allocation and ends the VM placement job.

#### 3.1 System architecture for MCC

Paper [30] has proposed a VM placement algorithm by network awareness based on MCC architecture. This algorithm creates a new VM that is assigned to a computing node (CN). The target is, in a vulnerable, poor reliability mobile network, as far as possible to reduce the

service response time between storage nodes (SNs) and CNs. In strictly accordance with the proposed architecture, this algorithm stores data on the SNs and places the VMs in the CNs. Assuming that a mobile application data is divided to  $m$  data pieces (DPs), and these DPs are respectively stored in  $n$  SNs. The distribution of data can be represented as a matrix  $D_{DP,SN}$ :

$$D_{DP,SN} = \begin{pmatrix} d_{11} & \dots & d_{1n} \\ \vdots & \ddots & \vdots \\ d_{m1} & \dots & d_{mn} \end{pmatrix}$$

Here,  $d_{ij}$  represents the size of DPs  $i$  stored on SN  $j$ .

For CNs, the bandwidth between them and SNs is known, so it can use another matrix  $B_{CN,SN}$  to express the bandwidth between each CN and SN:

$$B_{CN,SN} = \begin{pmatrix} b_{11} & \dots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{m1} & \dots & b_{mn} \end{pmatrix}$$

Here, the  $b_{ij}$  represents the size of the bandwidth between CN  $i$  and SN  $j$ .

Then, through formula (1) can get the service response time matrix  $T_{DP,CN}$  of each VM:

$$T_{DP,CN} = \frac{D_{DP,SN}}{B_{CN,SN}} \quad (1)$$

It can be seen from matrix  $T_{DP,CN}$  that, under the condition of that the amount of requested data is fixed, larger the bandwidth between CNs and SNs, shorter the service response time. Thus, we have found the most important factor of affecting service response time is bandwidth.

### 3.2 VMC process

Through the above analysis, this paper determines the first step and scheduling goal is that, by looking up the biggest bandwidth requesting one in the VMs and classifying it, to decrease the number of scanning times and improve the efficiency of the VM scheduling, thus achieve the goal of optimizing overall service response time.

Optimizing scheduling efficiency means that how to quickly find out the most meet customers' requirements one in a large number of the VMs. In order to improve the efficiency of scheduling algorithm and reduce the time of finding a suitable VM, we can reduce the number  $n$  of the VMs. The number  $n$  is smaller, the time of handling request is shorter, and the scheduling performance is higher.

Considering that reduces the number of the VMs to improve VM scheduling effectiveness,

the first step of the new algorithm is improved on the basis of ISODATA [31]. We have developed a VM classification (VMC) to classify the VMs according to the resource types. This process will divide a huge number of the VMs into  $K$  similar physical collections, so in dealing VM scheduling, the cloud data center will only send access request to each VM collection. Because of the similarity of the VM resources, they become a new type of similar resources, so it can greatly reduce the number of different types of the VMs and reduce the request time when allocating.

Based on the heterogeneity of the VMs, the VMC should dynamically create a parameters expert database, and the VM parameters is as follows:  $U$  is the expected number of the VM types,  $\theta_c$  is the minimum distance between two cluster centers,  $\theta_n$  is the lower bound of the number of the VMs in each category,  $L$  is the largest cluster number that is allowed to merge each iteration, and  $T$  is the allowed maximum number of iterations. Due to the value of  $U$  will be greatly subjective, and it may cause lower actual classification performance. Therefore, compared with ISODATA, we have made following optimization in the VMC process:

1) according to matrix  $T_{DP,CN}$  in formula (1), it can be seen the bandwidth impact, so this VMC process uses the bandwidth between CNs and the SNs as cluster attributes and calculating coordinates to complete host classification.

2) It can reduce multiple initialized parameters as follows: a) initializing cluster center. Because the parameters expert database in ISODATA has stored the number  $U$  of virtual host types, the VMC process can extract from different classes of hosts to initialize cluster center at any time. b) According to the actual classification requirements, the VMC process uses the distance between the virtual hosts as the standard that whether splitting.

The VMC process is divided into three stages: ① initializing environment, and computing cluster information, ② judging split, and merging operation, ③ doing split operation or doing merge operation.

The algorithm of the VMC process is as follows:

---

**Algorithm 1.** VMC process

---

**Input:**  $K // K \leftarrow U$

$P\{p_1, p_i \in P\}$  // virtual host set

$Z\{z_1, \dots, z_k\}$  // initial cluster center

**Output:**  $K$  virtual machine sets

- 1: **for**  $T$  do
- 2:     according to the Euclidean distances of virtual hosts and initial cluster center to classify the VM neighbors together
- 3:     adjusting cluster center  $z_k = \frac{1}{N_k} \sum_{p \in z_k} p, k = 1, 2, \dots, K$

```

4:      calculating  $\overline{D}_k = \frac{1}{N_k} \sum_{p \in z_k} \|p - z_k\|, k = 1, 2, \dots, K$  //the distance between each virtual host
                                                and cluster centers
5:      calculating total average distance  $\overline{D} = \frac{1}{N} \sum_{k=1}^K N_k \overline{D}_k$ 
6:      for  $K$  do
7:          if the cluster's host number  $N_k < \theta_n$  then
8:              cancel this category
9:               $K \leftarrow K - 1$ , and jump out of this loop
10:         else if  $K \leq U/2$  or  $(\overline{D}_k > \overline{D}$  and  $N_k > 2(\theta_n + 1))$  then // split cluster
11:             calculating  $Differ_k^{max}$  //the maximum value of  $Differ_k^{Ban}$  difference value between
                                                the bandwidth of VMs and that of cluster center
12:             splitting  $z_k$  into two new cluster centers  $z_k^+ \leftarrow z_k + Differ_k^{max}$ ,  $z_k^- \leftarrow z_k -$ 
                                                 $Differ_k^{max}$ 
13:              $K \leftarrow K + 1$ , and jumping out of this loop
14:         else if the number of iterating times is even number or  $U \geq K \geq U/2$  then // merge cluster
15:             calculating all cluster centers' distance  $D_{ij} = \|z_i - z_j\|, i = 1, 2, \dots, K - 1; j = i +$ 
                                                 $1, \dots, K$ 
16:             comparing with  $D_{ij}$  and  $\theta_c$ , arrange the cluster that meet  $D_{ij} < \theta_c$  of the ascending order,
                                                then create set  $\{D_{i_1 j_1}, D_{i_2 j_2}, \dots, D_{i_L j_L}\}$ . Here,  $D_{i_1 j_1} < D_{i_2 j_2} < \dots < D_{i_L j_L}$ 
                                                merging two cluster centers  $z_{i_k}$  and  $z_{j_k}$  that distance equals  $D_{i_k j_k}$  to get a new center
17:              $z_k^* = \frac{1}{N_{i_k} + N_{j_k}} [N_{i_k} z_{i_k} + N_{j_k} z_{j_k}], k = 1, 2, \dots, L$ 
18:              $K \leftarrow K - 1$ , and jumping out of this loop
19:         end if
20:     end for
21:     if it is the last iteration calculation then
22:          $\theta_c \leftarrow 0$ 
23:     end this algorithm
24:     else  $T \leftarrow T + 1$ 
25:     end if
26: end for

```

---

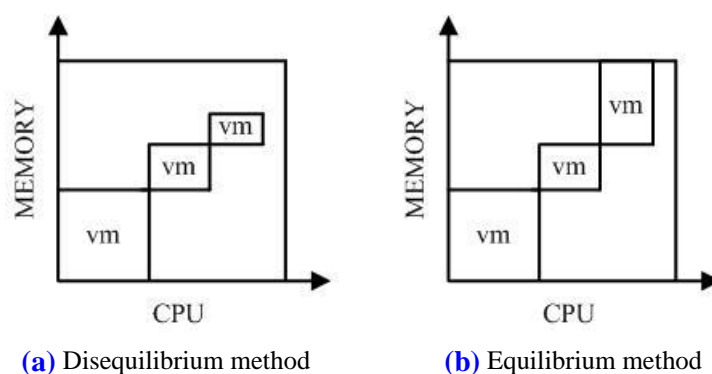
### 3.3 Resources equilibrium

The first step in the VM placement algorithm, we have completed the initial classification of the VMs, and it can quickly find the suitable VM sets from huge VM clusters according to resource requests. But it is not enough, after the VMC by the algorithm, and the output result is still a set of VMs. How to find the most appropriate VM from the optimized VM set to complete VM placement will be the second step to be considered.

The first step of the algorithm is to solve the problem of task response time, likely, it will assign multiple tasks in larger bandwidth the VM, but the smaller bandwidth VM will be assigned rarely tasks, then it will cause disequilibrium distribution. So the second step is to balance all kinds of resource allocation in the VM.

According to the multi-dimension of physical host resources, the VM placement should be resource equilibrium between each degree resources, and guarantee the similarity of them.

**Fig. 1** shows the comparison of resources equilibrium situation.



**Fig. 1.** Multi-dimensional resource equilibrium of physical hosts

In **Fig. 1(a)**, CPU utilization has been as high as 90%, but memory utilization is only 80%. Because each VM request includes CPU and memory resources, it makes that CPU utilization is full that the physical host cannot bear more VMs, and then causes 20% of memory resources are idle. In **Fig. 1(b)**, the utilizations of CPU and memory all have reached 90%, and the utilization of each dimension is in equilibrium, and physical hosts have played the best benefit. Therefore, in physical hosts, more balanced each dimension resources, more fully used can be physical resources.

### 3.4 CBVMP algorithm

In order to make the VM resources that wait to be distributed matching with mobile equipment remaining resources to some extent, and find the right VM from the VM set that be output of the first step of placement algorithm. We have proposed the CBVMP algorithm. It uses the angle between two given vectors to measure the similarity between the VMs and the remaining resources of mobile devices and complete the VM placement at same time. With shielding physical storage properties, the VM requests are a two-dimensional vector  $\vec{V}\langle CPU, MEM \rangle$ , and the remaining resources of mobile equipments are also a two-dimensional vector  $\vec{P}\langle CPU, MEM \rangle$ . As long as  $\vec{V}$  and  $\vec{P}$  are two paralleled vectors, it can guarantee, after physical hosts bearing the VM, each dimensional resources are relatively balanced. Therefore, in this paper, we use the angle between two given vectors to measure effects, that smaller the angle, more balanced the resources. The formula of the angle is as



follows:

$$\cos\langle\vec{v},\vec{p}\rangle = \frac{\vec{v}\cdot\vec{p}}{|\vec{v}|\cdot|\vec{p}|} \quad (2)$$

Therefore, in order to optimize the equilibrium rate, we need to minimize the angle between two given vectors.

In the VMC process, the VM requests are dynamically divided into  $K$  classes, and these  $K$  classes are concurrently allocated to the VMs. The algorithm puts  $K$  as the input parameter, and divides  $n$  sets into  $K$  clusters, that ensure the high similarity in clusters and low similarity between clusters. At the same time, it uses Euclidean distance to complete neighbor clusters, and then repeatedly uses Euclidean distance formula to shrinking clusters, finally gets the optimal mapping set of the VMs and physical machines.

The specific process of the CBVMP algorithm is as follows:

---

#### Algorithm 2. CBVMP algorithm

---

**Input:**  $V\{v_1, \dots, v_n\}$  // VM request set

**Output:** mapping set of the target VM ID and physical machine ID

- 1: **if**  $V > K$  then
  - 2:     convert the VM requests to  $\vec{V}\langle CPU, MEM \rangle$
  - 3:     calculating the value of the angle  $|\vec{v}|$  using the formula  $\cos\langle\vec{v},\vec{p}\rangle = \frac{\vec{v}\cdot\vec{p}}{|\vec{v}|\cdot|\vec{p}|}$  and the fixed vector  $\langle 1, 1 \rangle$
  - 4:     sorting  $|\vec{v}|$  in ascending order
  - 5:     choosing  $K$  initial points as the center in ladder way
  - 6:     **while**  $K$  is one of the initial points **do**
  - 7:         traversing the set  $V\{v_1, \dots, v_n\}$
  - 8:         calculating the Euclidean distance between the VM and the  $K$  center points
  - 9:         completing neighbors cluster, and forming  $K$  clusters
  - 10:        recalculating each cluster center
  - 11:     **end while**
  - 12: **end if**
- 

## 4. Performance evaluation and discussion

### 4.1 Simulation Setup

We tested the CBVMP algorithm in simulation platform *CloudSim3.0*. For the two steps of this algorithm, firstly, we have used classification process to the VMs, and run the greedy

algorithm and the genetic algorithm based on the pre-processed classification of the VMs, and compared the two algorithms performance between before and after, and we can see that the classification process can reduce the time consumption in the VM allocation. Secondly, we have compared the CBVMP algorithm with the greedy algorithm and the genetic algorithm, and the results show that the CBVMP algorithm not only can greatly reduce the VM placement time, but also can achieve the equilibrium utilization of resources.

We have created a data center, and each data center contains several hosts, each host is described by CPU, memory, bandwidth. Here, CPU performance consists of the number of CPU cores and the MIPS of CPU. The parameters are as follows:

#### 1) Bandwidth

According to the features of wireless networks [18], we set up a bandwidth set of the VMs, and it is as follows: {2.4, 3.2, 3.5, 5, 6, 7, 7.2, 8, 8.5, 9, 10, 11, 12, and 16}.

#### 2) Physical machine

Using randomly generated strategy, we define a new class: *DatacenterCharacteristics*, and it can generate corresponding data centers and physical hosts. We randomly generate four types of physical hosts to simulate the natural heterogeneous underlying environment. The eight types of parameters are as shown in **Table 1**. Based on four types of physical hosts, we can generate multiple physical hosts using random strategy again.

**Table 1.** Parameters of physical hosts

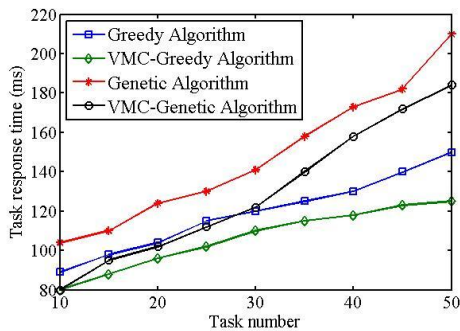
Class	CPU number	CPU speed	Memory capacity
G1	1	1000	1
G2	2	1000	3
G3	3	1000	6
G4	4	1000	10
G5	1	1500	1
G6	2	1500	3
G7	3	1500	6
G8	4	1500	10

#### 3) VM allocation request

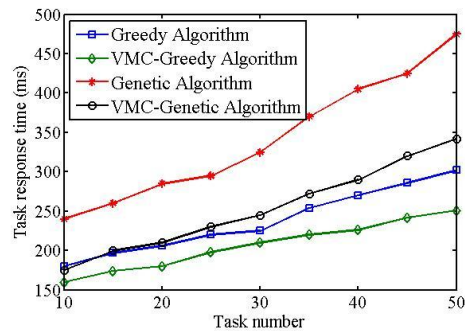
According to the randomly generating condition of physical machines, using random strategy again to generate a VM allocation request queue, and get the VM queue *DatacenterBroker* that meets *CloudSim* platform. For example, the number of CPUs is randomly generated from 1 to 4, and the memory capacity is randomly generated between 1G - 8G. The number of the VMs that request to be allocated is set by 500, 1000, 1500 and 3000.

## 4.2 Results and analysis of the VMC process

We run the greedy algorithm and the genetic algorithm based on the pre-processed classification of the VMs. **Fig. 2** and **Fig. 3** show the impact of the task number of VM clusters on a mobile host on the task response time of VMs based on the greedy algorithm and genetic algorithm before and after the VMC process.



**Fig. 2.** Task response time for task number (VMs number is 500)



**Fig. 3.** Task response time for task number (VMs number is 3000)

From **Fig. 2** and **Fig. 3**, we see a straightforward dependency: when the VM request number is 500, the effect of acceleration is not particularly obvious, and performance improvement is also limited, and also compression between two lines is narrow. But when 3000, accelerating effect is clearly better. Greedy algorithm's own performance is relatively excellent, but genetic algorithm is relatively complex, that involves selection, crossover and mutation process. So when using the VMC process after preprocessing, the optimization effect is relatively obvious, especially when the VMs number is larger, speedup will be more obvious by the VMC process.

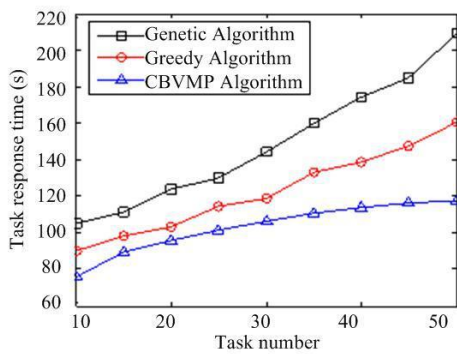
To summarize, it can be seen that the VMC can effectively accelerate greedy algorithm and genetic algorithm. Here, the overall task response time is the sum of time from searching the most appropriate VM to completed tasks, and we control this time through reducing the VM cluster dimension. Because the VMC will always choose the one that has the biggest bandwidth as the target, and so the response time will be gradually increased with better VM being used. But regardless of how to increase the number of the VM cluster, the VMC process we have proposed can accelerate these two algorithms very well.

## 4.3 Results and analysis of the CBVMP algorithm

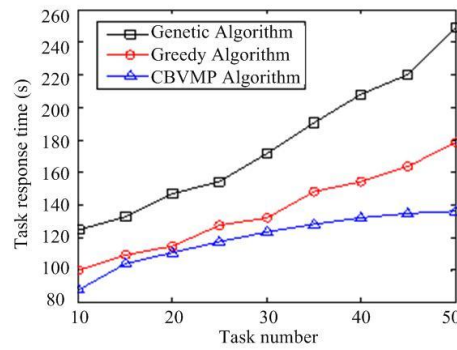
We run the CBVMP algorithm, the greedy algorithm and the genetic algorithm based on the pre-processed classification of the VMs. **Fig. 4-7** show the impact of the task number of the VM clusters on a mobile host on the task response time of the VMs.

From **Fig. 4-7**, we see a straightforward dependency: it can be seen that the CBVMP algorithm, for the overall task response time, compared with other two algorithms, has been

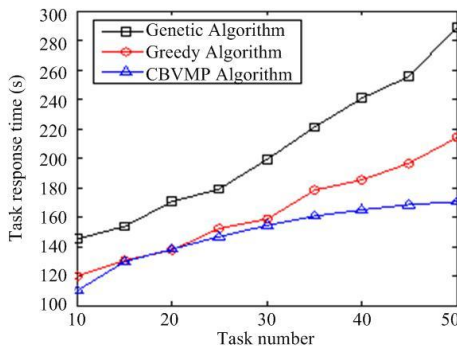
obviously improved. When the VMs number is a given value, with the increase of tasks that submitted to the VM, the growth of task response time of the CBVMP algorithm is not large, but other two algorithms are very large. Meanwhile, when tasks have been more than 40, the CBVMP algorithm have played a big role in reducing response time. It is can be imaged that when we continue to submit tasks, the task response time will no longer obviously increasing. The VMs number grows, and a suitable VM becomes more and more difficult to be found, so the CBVMP algorithm can make a clear advantage, and this has the very vital significance for mobile devices on multitasking improvement.



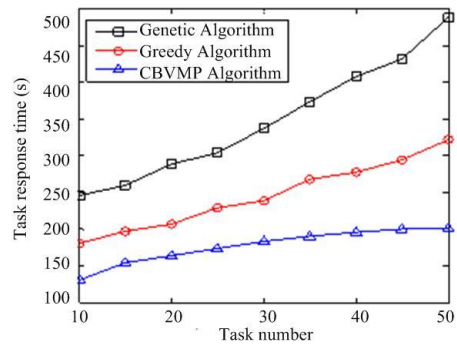
**Fig. 4.** Task response time for task number (VMs number is 500)



**Fig. 5.** Task response time for task number (VMs number is 1000)



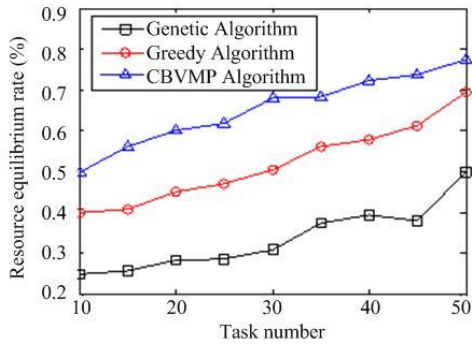
**Fig. 6.** Task response time for task number (VMs number is 1500)



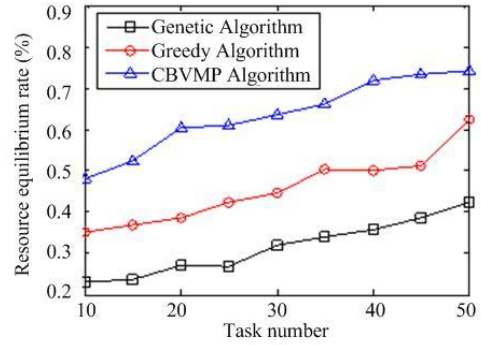
**Fig. 7.** Task response time for task number (VMs number is 3000)

**Fig. 8-11** show the impact of the task number of the VM clusters on a mobile host on the resource equilibrium rate. We see a straightforward dependency: when VMs number increases from 500 to 3000, the improvement of task response time is always very obvious, while the number of the VMs is 3000 and the number of tasks is 50, the task response time is never more than 200s. At the same time, compared with other three groups of test, it can be found that task response time becomes smaller, and the change of task response time of the

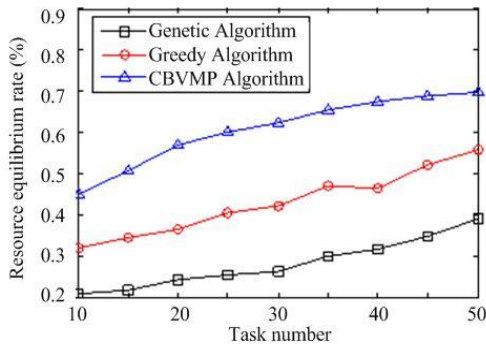
CBVMP algorithm is not more than 100s, while, in other two algorithms, task response time is almost double.



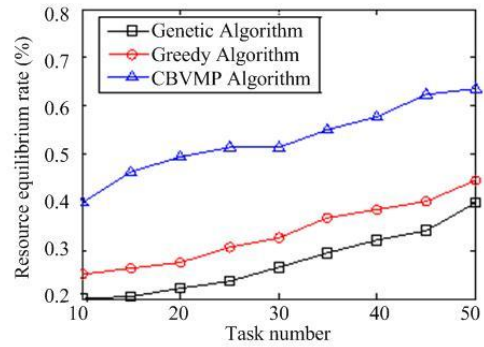
**Fig. 8.** Resource equilibrium rate for tasks number (VMs number is 500)



**Fig. 9.** Resource equilibrium rate for tasks number (VMs number is 1000)



**Fig. 10.** resource equilibrium rate for tasks number (VMs number is 1500)



**Fig. 11.** resource equilibrium rate for tasks number (VMs number is 3000)

From **Fig. 8-11**, we can see the resource equilibrium rate of the CBVMP algorithm is always better than other two algorithms, but as the rise in the VM number, resource equilibrium rate also will continually decline. Also we can see that, when the VMs number is small, these three algorithms in resource equilibrium rate are actually close, and when 3000, one of other two algorithms has a quite obvious decline, but CBVMP algorithm is still relatively stable. So, CBVMP algorithm, in multitasking, huge and complicated network environment, is good to maintain resource equilibrium.

### 5. Conclusion and future work

This paper has proposed a VM placement algorithm called the CBVMP algorithm. It aims at solving low efficiency in the VM allocation in the large cloud data center and disequilibrium utilization of physical machine resources. Firstly, we have discussed influence factors of the

user experience, and obtained that the network bandwidth between CNs and SNs is one of the important factors that affects the response time. While, the VM is placed on the CNs, so the allocated bandwidth of the VM will determines the value of the whole task response time, and the bandwidth will be the key influence factor in the algorithms. Because of this background, we have changed a series of parameters of the existing ISODATA algorithm that decides bandwidth as the key factor to classify the VM cluster. Second, our main purpose is, through the multi-objective optimization, to reach the better performance of the VM placement. In order to solve this problem, we finally chose the CPU and memory as factors. We have arranged the VM allocation requests of ascending order by resource equilibrium rate, and divided the VMs into  $K$  classes by request resources quantity and equilibrium rate. Finally we have concurrently mapped the VM with  $K$  classes. In addition, because each type of the VM resource is relatively balanced, we can further ensure, after mapping, each dimension resource is balanced in underlying hosts.

Greedy algorithm, genetic algorithm and CBVMP algorithm belong to the class of heuristic algorithms. Greedy algorithms follow the problem solving heuristic of making the locally optimal choice at each stage with the hope of finding a global optimum, and genetic algorithms have a tendency to converge towards a locally optimal choice rather than the global optimum of the problem. In the problem of virtual machine placement for mobile cloud computing, the CBVMP algorithm yields the best one of the three sub-optimal solutions that approximates an optimal solution in a reasonable time. In the future work, we will consider more factors and economic costs, in addition to the response time, we also can take network topology, throughput and other factors into account.

In addition, in the cloud data center, physical hosts have heterogeneous characteristics. Especially under the environment of mobile Internet, smart phones' battery problem still lagged behind the development of the phone itself. In the future, we will focus on saving energy consumption, and combine with green cloud computing.

## References

- [1] Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, "Heterogeneity in mobile cloud computing: taxonomy and open challenges," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 369-392, 2014. [Article \(CrossRef Link\)](#)
- [2] M. R. Rahimi, J. Ren, C. H. Liu, A. V. Vasilakos, and N. Venkatasubramanian, "Mobile cloud computing: a survey, state of art and future directions," *Mobile Networks & Applications*, vol. 19, no. 2, pp. 133-143, 2014. [Article \(CrossRef Link\)](#)
- [3] M. Jo, T. Maksymyuk, B. Strykhalyuk, and C. H. Cho, "Device-to-device-based heterogeneous radio access network architecture for mobile cloud computing," *IEEE Wireless Communications*, vol. 22, no. 3, pp. 50-58, 2015. [Article \(CrossRef Link\)](#)



- [4] R. Kaewpuang, D. Niyato, P. Wang, and E. Hossain, "A framework for cooperative resource management in mobile cloud computing," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 12, pp. 2685-2700, 2013. [Article \(CrossRef Link\)](#)
- [5] M. Shiraz and A. Gani, "A lightweight active service migration framework for computational offloading in mobile cloud computing," *The Journal of Supercomputing*, vol. 68, no. 2, pp. 978-995, 2014. [Article \(CrossRef Link\)](#)
- [6] J. Wan, D. Zhang, Y. Sun, K. Lin, C. Zou, and H. Cai, "VCMIA: a novel architecture for integrating vehicular cyber-physical systems and mobile cloud computing," *Mobile Networks & Applications*, vol. 19, no. 2, pp. 153-160, 2014. [Article \(CrossRef Link\)](#)
- [7] L. Wang, Y. Cui, I. Stojmenovic, X. Ma, and J. Song, "Energy efficiency on location based applications in mobile cloud computing: a survey," *Computing*, vol. 96, no. 7, pp. 569-585, 2014. [Article \(CrossRef Link\)](#)
- [8] H. Ghiasi and M. G. Arani, "Smart Virtual Machine Placement Using Learning Automata to Reduce Power Consumption in Cloud Data Centers," *Smart Computing Review*, vol. 5, no. 6, pp. 553-562, 2015. [Article \(CrossRef Link\)](#)
- [9] C. Liang and F. R. Yu, "Wireless network virtualization: a survey, some research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 358-380, 2014. [Article \(CrossRef Link\)](#)
- [10] F. Xu, F. Liu, H. Jin, and A. V. Vasilakos, "Managing performance overhead of virtual machines in cloud computing: a survey, state of the art, and future directions," in *Proc. of the IEEE*, vol. 102, no. 1, pp. 11-31, 2014. [Article \(CrossRef Link\)](#)
- [11] N. Bobroff, A. Kochut, and K. Beaty, "Dynamic placement of virtual machines for managing SLA violations," in *Proc. of the 10th IFIP/IEEE international Symposium on Integrated Network Management (IM)*, pp. 119-128, Munich, Germany, May. 21-25, 2007. [Article \(CrossRef Link\)](#)
- [12] S. Chaisiri, B-S. Lee, and D. Niyato, "Optimal virtual machine placement across multiple cloud providers," in *Proc. of IEEE Asia-Pacific Services Computing Conference (APSCC)*, pp. 103-110, Biopolis, Singapore, Dec. 7-11, 2009. [Article \(CrossRef Link\)](#)
- [13] H. N. Van, F. D. Tran, and J-M. Menaud, "Autonomic virtual resource management for service hosting platforms," in *Proc. of the ICSE Workshop on Software Engineering Challenges of Cloud Computing (CLOUD)*, pp. 1-8, Vancouver, Canada, May. 16-24, 2009. [Article \(CrossRef Link\)](#)
- [14] X. Meng, V. Pappas, and L. Zhang, "Improving the scalability of data center networks with traffic-aware virtual machine placement," in *Proc. of 29th IEEE International Conference on Computer Communications (INFOCOM)*, pp. 1-9, San Diego, CA, USA, Mar. 14-19, 2010. [Article \(CrossRef Link\)](#)
- [15] A. Geyer-Schulz and M. Ovelgönne, "The randomized greedy modularity clustering algorithm and the core groups graph clustering scheme," *German-Japanese Interchange of Data Analysis Results*, pp. 17-36, 2013. [Article \(CrossRef Link\)](#)
- [16] A. Dhingra and S. Paul, "Green cloud: heuristic based BFO technique to optimize resource allocation," *Indian Journal of Science and Technology*, vol. 7, no. 5, pp. 685-691, 2014.

[Article \(CrossRef Link\)](#)

- [17] H. Nakada, T. Hirofuchi, H. Ogawa, and S. Itoh, "Toward virtual machine packing optimization based on genetic algorithm," in *Proc. of the 10th International Work Conference on Artificial Neural Networks (IWANN)*, pp. 651-654, Salamanca, Spain, Jun. 10-12, 2009.

[Article \(CrossRef Link\)](#)

- [18] S. Agrawal, S. K. Bose, and S. Sundarrajan, "Grouping genetic algorithm for solving the serverconsolidation problem with conflicts," in *Proc. of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation (GEC)*, pp. 1-8, Shanghai, China, Jun. 12-14, 2009.

[Article \(CrossRef Link\)](#)

- [19] M. Tang and S. Pan, "A hybrid genetic algorithm for the energy-efficient virtual machine placement problem in data centers," *Neural Processing Letters*, vol. 41, no. 2, pp. 211-221, 2015.

[Article \(CrossRef Link\)](#)

- [20] B. Wadhwa and A. Verma, "Energy saving approaches for green cloud computing: a review," in *Proc. of 2014 Recent Advances in Engineering and Computational Sciences (RAECS)*, pp. 1-6, Chandigarh, India, Mar. 6-8, 2014. [Article \(CrossRef Link\)](#)

- [21] S. Srikantaiah, A. Kansal, and F. Zhao, "Energy aware consolidation for cloud computing," in *Proc. of the 2008 conference on Power aware computing and systems (HotPower)*, pp. 10, San Diego, CA, Dec.7-10, 2008. [Article \(CrossRef Link\)](#)

- [22] C. Tang, M. Steinder, M. Spreitzer, and G. Pacifici, "A scalable application placement controller for enterprise datacenters," in *Proc. of the Sixteenth International World Wide Web Conference (WWW)*, pp. 331-340, Banff, Alberta, Canada, May. 8-12, 2007. [Article \(CrossRef Link\)](#)

- [23] C. Hyser, B. McKee, R. Gardner and B. J. Watson, "Autonomic virtual machine placement in the data center," *Technical Report*, HP Laboratories, HPL-2007-189, 2008. [Article \(CrossRef Link\)](#)

- [24] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755-768, 2012. [Article \(CrossRef Link\)](#)

- [25] O. Biran, A. Corradi, M. Fanelli, L. Foschini, A. Nus, D. Raz, and E. Silvera, "A stable network-aware VM placement for cloud systems," in *Proc. of the 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, pp. 498-506, Ottawa, Canada, May. 13-16, 2012. [Article \(CrossRef Link\)](#)

- [26] I. Goiri, F. Julia, R. Nou, J. L. Berral, J. Guitart and J. Torres, "Energy-aware scheduling in virtualized datacenters," in *Proc. of 2010 IEEE international Conference on Cluster Computing (CLUSTER)*, pp. 58-67, Heraklion, Crete, Greece, Sept. 20-24, 2010. [Article \(CrossRef Link\)](#)

- [27] M. Sun, W. Gu, X. Zhang, H. Shi, and W. Zhang, "A matrix transformation algorithm for virtual machine placement in cloud," in *Proc. of 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 1778-1783, Melbourne, Australia, Jul. 16-18, 2013. [Article \(CrossRef Link\)](#)

- [28] S. Wang, H. Gu, and G. Wu, "A new approach to multi-objective virtual machine placement in virtualized data center," in *Proc. of IEEE Eighth International Conference on Networking*,



*Architecture and Storage (NAS)*, pp. 331-335, Xi'an, China, Jul. 17-19, 2013.

[Article \(CrossRef Link\)](#)

- [29] L. Lu, H. Zhang, E. Smirni, G. Jiang, and K. Yoshihira, "Predictive virtual machine consolidation on multiple resources: beyond load balancing," in *Proc. of IEEE/ACM 21st International Symposium on Quality of Service (IWQoS)*, pp. 1-10, Montreal, Canada, Jun. 3-4, 2013. [Article \(CrossRef Link\)](#)

- [30] D. Chang, G. Xu, L. Hu, and K. Yang, "A network-aware virtual machine placement algorithm in mobile cloud computing environment," in *Proc. of 2013 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pp. 117-122, Shanghai, China, Apr. 7-10, 2013. [Article \(CrossRef Link\)](#)

- [31] G. H. Ball and D. J. Hall, "ISODATA, a novel method of data analysis and pattern classification," *Technical Report*, Stanford Research Institute, Menlo Park, CA, 1965. [Article \(CrossRef Link\)](#)



**Yuli Tang** received his BS degree and MS degree from Hunan Normal University, China in 2012 and 2015. He is a member of the Key Laboratory of Internet of Things Technology and Application. Her research interests include network virtualization, and mobile cloud computing.



**Yao Hu** received the BS degree from Hunan Normal University, China in 2014. She is currently a MS student in the College of Physics and Information Science, Hunan Normal University, China, and a member of the Key Laboratory of Internet of Things Technology and Application. Her research interests include wireless communication and network, and cloud computing.



**Lianming Zhang** received his PhD from Central South University, China. He received his BS degree and MS degree from Hunan Normal University, China. He is currently a professor in the school of Physics and Information Science, Hunan Normal University, and leads the Key Laboratory of Internet of Things Technology and Application there. His main research interests include cloud computing, software-defined-networking, complex networks, and network calculus. He completed a two year postdoctoral fellowship in complex network at South China University of Technology. He manages research projects funded by various sources such as National Natural Science Foundation of China and industries. He has published more than 90 papers.