

A Novel Method for Virtual Machine Placement Based on Euclidean Distance

Shukun Liu¹, Weijia Jia²

¹ School of Information Science and Engineering, Central South University, 410083 Changsha, China
[e-mail: liu_shukun@csu.edu.cn]

² Department of Computer Science and Engineering, Shanghai Jiao Tong University, 200240 Shanghai, China
[e-mail: weijiaj@gmail.com]

*Corresponding author: Shukun Liu

*Received November 5, 2015; revised January 31, 2016; revised March 1, 2016; accepted May 31, 2016;
published July 31, 2016*

Abstract

With the increasing popularization of cloud computing, how to reduce physical energy consumption and increase resource utilization while maintaining system performance has become a research hotspot of virtual machine deployment in cloud platform. Although some related researches have been reported to solve this problem, most of them used the traditional heuristic algorithm based on greedy algorithm and only considered effect of single-dimensional resource (CPU or Memory) on energy consumption. With considerations to multi-dimensional resource utilization, this paper analyzed impact of multi-dimensional resources on energy consumption of cloud computation. A multi-dimensional resource constraint that could maintain normal system operation was proposed. Later, a novel virtual machine deployment method (NVMDM) based on improved particle swarm optimization (IPSO) and Euclidean distance was put forward. It deals with problems like how to generate the initial particle swarm through the improved first-fit algorithm based on resource constraint (IFFABRC), how to define measure standard of credibility of individual and global optimal solutions of particles by combining with Bayesian transform, and how to define fitness function of particle swarm according to the multi-dimensional resource constraint relationship. The proposed NVMDM was proved superior to existing heuristic algorithm in developing performances of physical machines. It could improve utilization of CPU, memory, disk and bandwidth effectively and control task execution time of users within the range of resource constraint.

Keywords: Virtual machine, particle swarm, multi-dimensional, low energy consumption, Euclidean distance

This work is supported in part by the China National 973 Project Grant Number 2015CB352401, National Natural Science Foundation of China under Grant Numbers 61532013 and the Scientific Research Fund of Hunan Provincial Education Department (No.14C0680). We express our thanks to the editor and all the reviewers for our manuscript.

1. Introduction

Due to increasingly expanding network application services and limited physical resources, there is a strong desire of high-efficiency resource integration under current network environment. Such resource integration could enhance reasonable distribution of physical resources, lower cost per unit resources and increase users' utilization of desired physical resources. In current distributed computing environment, functions of physical resources desired by users are mainly reflected as encapsulated virtual machines. Therefore, resource scheduling process could be viewed as a process of searching virtual machines [1][9,10]. In the process of resource scheduling [30], one of the key problems is how virtual machines choose corresponding physical nodes quickly and reasonably under the premise of meeting all different application service levels. If the supporting physical machine could be chosen quickly during virtual resource deployment according to comprehensive factors, physical resource utilization and users' satisfaction will increase significantly. Most existing researches viewed the multi-dimensional vector packing as the mapping problem from virtual machine to physical machine [2][11,12]. This is a NP-hard problem which has optimal solution. But it's hard to find out the optimal solution within a reasonable time limit. At present, it is often solved by heuristic algorithm [3][13]. However, existing researches concerning virtual machine deployment in cloud computing mainly focus on optimization of one certain dimension, such as guarantee of service level goal, reduction of physical nodes, cost reduction for virtual machine migration, reduction of energy consumption, etc. Chokoufe Nejad et al. [4] and J. Xu et al. [14] dealt with static virtual machine deployment based on genetic algorithm, but they didn't take expenses for virtual machine migration into account. M. Chen et al. [15] integrated virtualized nodes in cloud computing data center, which was described as a random packing optimization problem. It only considered CPU resources, but neglected other dimensional resources (e.g. Memory). ZP. Peng et al. [5] and MF. Li et al. [16] proposed a management framework for virtual machine deployment in cloud computing, but they neglected expenses and system energy consumption which users were highly concerned. Bhutani et al. [6] and S. Sawant [17] proposed a virtual machine deployment mechanism based on genetic algorithm. Supported by historical data of cloud platform and state of the current system, this mechanism could reach a good load balance of a specific-dimensional resource at a low cost for virtual machine deployment. Nevertheless, it failed to reflect resource utilization and related physical energy consumption in cloud platform. Cerroni Walter et al. [7], J. Xu et al. [14] and MF. Li et al. [16] regarded virtual machine deployment as single-dimensional packing and multi-dimensional goal optimization, respectively. But they neither discussed resource costs for virtual machine migration nor involved complicated dynamic virtual machine deployment. They only coped with static deployment of virtual machines. Virtual machine deployment was decomposed into combinational optimization of specific-dimensional resources [8][18,19]. Based on the genetic algorithm, they still emphasized on static virtual machine deployment and didn't take dynamic migration of virtual machines as an influencing factor.

Although tremendous research results on virtual machine deployment in cloud platform have been achieved, some over emphasized on utilization of physical resources and neglected energy consumption of physical machines. This paper combined energy consumption and physical resource during virtual machine deployment. Additionally, most researches on virtual machine deployment strategy made single-dimensional optimization based on a certain rule,

failing to get the universal optimal solution. A high-efficiency virtual machine deployment strategy shall consider dependence of multi-dimensional constraints comprehensively.

Main contributions of this paper include four aspects below:

(1) In this study, we proposed an improved PSO-based virtual machine deployment method involving multi-dimensional user demands. It analyzes virtual machine deployment from improvement of resource utilization and reduction of energy consumption of physical machines.

(2) We proposed an improved first-fit algorithm based on resource constraint (IFFABRC) in this paper, which can be used to generate the initial positions of virtual machines efficiently.

(3) We proposed a measurement criteria of virtual machine deployment based on Euclidean distance which is multi-dimensional, involving CPU, Memory, Disk and Bandwidth. This criteria is different with other traditional ways which are always in a single dimension (CPU or Memory).

(4) In our deployment method, cloud computing center can analyze desired physical resources of users. In this way we can deal with the resource requirements of user task easily. From this view, a multi-dimensional resource constraint relationship that could maintain normal system operation was proposed.

Rest of the paper is organized as follows: Section 2 provides the problem description. Section 3 describes virtual machine deployment method based on energy consumption criterion. Section 4 describes experimental environment setting and result comparison. Section 5 concludes the paper and describes our future work.

2. Problem Description

Energy consumption is an important challenge for the resources management of cloud data center. With the quick development of data centers, energy consumption problem is increasing prominently. Data center not only wastes the physical energy, but also has negative impact on the environment. The unreasonable resource allocation and the increase of infrastructure construction of data center are the main factors of energy consumptions. Research methods of efficient virtual machine deployment for cloud data center have important influence on reducing energy consumption and improving the utilization rate of resources.

Resources of physical node consist of CPU, memory, disk, bandwidth, etc. At present, with the purpose of energy saving, some virtual machine placement methods only take consideration of single-dimension resource (for example, CPU or Memory) which ignored influences of other system resources on energy consumption. In fact, any dimensional resource (such as CPU, memory, bandwidth, disk) of the physical node has important influence on the comprehensive utilization of energy efficiency for physical nodes. In this paper, resource constraint relationships based on the four dimensions of resources (CPU, memory, bandwidth, and disk) are established. The constraint relationships among the resources are established. The Euclidean distance of ideal virtual machine deployment point combined with CPU, memory, disk and bandwidth of all the enabled physical nodes in the data center is used to measure the deviation degree between current system and the best system during the process of virtual machine deployment.

The smaller the deviation degree is, the better the virtual machine deployment effect is. That is to say, the four dimensional physical resources are all be saved. So energy consumption

saving can be achieved by reducing the number of the enabled physical nodes. The definition of Euclidean distance is defined in section 2.4.2.

User submits task request to the cloud computing center through a specific port. To protect successful accomplishment of this task, matched physical resources shall be allocated to this user mainly in virtual machines. Supply excess will cause resource waste, whereas supply shortage will result in failure of the task. If only physical resource demanded by virtual machines are offered, the task performance will be affected significantly even though it could be accomplished. In practice, many users submit applications of different physical resources simultaneously. To avoid mutual influence of users, virtual machines shall be independent from each other. The cloud computing center has to provide various resources to accomplish user tasks, including CPU, memory, hard disk, network resources and so on. Therefore, virtual machine deployment can be viewed as a problem that how physical resources meet multi-dimensional goal. Each dimension is one of physical resources of physical machines and resources demanded by every virtual machine are the corresponding multi-dimensional vector.

The high-efficient reasonable deployment of virtual machines in this paper is to put multiple virtual machines onto multiple reasonable physical nodes according to different requirements of users. This ensures both high utilization of physical resource and low energy consumption.

In order to state application context and problem clearly, we add two notation tables below:

Table 1. Notation table (1)

SYMBOL	DESCRIPTION
H	number of data centres
VM	virtual machine set
vm_j	the j^{th} virtual machine
PM	physical machine set
pm_i	the i^{th} physical machine
ts	task set of a virtual machine
$R_{D(h)}$	resource set of the h^{th} data centre
r_i	resource set of i^{th} physical machine
$r_i(t)$	the t^{th} dimension resource of the i^{th} physical node
SR	useful resource service vector
sr_i	the useful service ratio of r_i
tr	task generation ratio set of virtual machine
tr_j	task generation ratio of vm_j
$P_{n \times m}$	probability distribution matrix of data source
$S_{n \times m}$	cloud resource scheduling matrix

Table 2. Notation table (2)

SYMBOL	DESCRIPTION
$U_{i,t}$	i^{th} dimension resource utility of pm_i
$U_{j,t}^i$	utility of t^{th} resource of vm_j which is assigned to pm_i
$T_{i,t}$	the threshold of t^{th} dimension resource of pm_i
R	remaining resource biggest matrix of physical node
VR	virtual machine resource matrix
MP	mapping matrix of vm_j to pm_i
T	the max number of resource dimension
E	the Euclidean distance between four-dimensional resources
u_i^c	actual utility ratio of CPU on the i^{th} physical node
u_i^m	actual utility ratio of Memory on the i^{th} physical node
u_i^d	actual utility ratio of Disk on the i^{th} physical node
u_i^b	actual utility ratio of Bandwidth on the i^{th} physical node
u^{bc}	utility ratio of CPU of physical node under the ideal state
u^{bm}	utility ratio of Memory of physical node under the ideal state
u^{bd}	utility ratio of Disk of physical node under the ideal state
u^{bb}	utility ratio of Bandwidth of physical node under the ideal state

2.1 Application Context

A cloud platform comprises a number of data centers. In this study, we suppose that H data centers(DC) exist in a cloud platform and denote all resources in the data centers as $\sum_{h=1}^H R_{D(h)}$.

One data center comprises n physical nodes. $R_{D(h)}$ denotes resource set of the h^{th} data centre which is denoted as $R_{D(h)} = \{r_1, r_2, r_3, \dots, r_i, \dots, r_n\}$. And $r_i = \{r_i(1), r_i(2), r_i(3), \dots, r_i(t)\}$ denotes a set of t different dimensions resource of pm_i . During an entire service period, which is denoted as SR , the useful service time obeys the Poisson distribution [27], and all the resources are independent of one another, i.e., $\forall r_i \in R_{D(h)}, \bigcap_{i=0}^n r_i = \phi$. In the resource set of a data center, which is denoted as $R_{D(h)}$, the useful service vector is denoted as $SR = \{sr_1, sr_2, \dots, sr_n\}$, and the useful service rate of r_i is denoted as sr_i ($0 \leq i \leq n$).

A virtual machine set is denoted as $VM = \{vm_1, vm_2, vm_3, \dots, vm_m\}$, which is placed on physical nodes. The set comprises m virtual machines. In this study, we suppose that vm_j only has one task ts_j at any time. The j^{th} task, which is denoted as $ts_j = \{ts_j(1), ts_j(2), ts_j(3), \dots, ts_j(t)\}$, $0 \leq j \leq m$, includes t different attributes. The task generation rate of a virtual machine can be described as $tr = \{tr_1, tr_2, tr_3, tr_4, \dots, tr_j, \dots, tr_m\}$, $0 \leq j \leq m$, where tr_j denotes the task production rate of the j^{th} virtual machine. All independent tasks satisfy $\forall ts_j \in VM, \bigcap_{j=0}^m ts_j = \phi, \forall r_i \in R_{D(h)}, ts_j \in VM$. If r_i can satisfy all the requirements of ts_j , then

ts_j can be assigned to r_i , and all operations can be run. In some cases, many r_i can satisfy the requirements of ts_j . To achieve an effective balance among all resources, ts_j is allocated to r_i with a probability (P_{ij}). The probability distribution matrix of data resources is denoted as $P_{n \times m} = (P_{ij})_{n \times m}$, with the condition that $P_{ij} \geq 0$ and $\sum_{i=0}^n P_{ij} = 1$. The cloud resource scheduling matrix is denoted as $S_{n \times m} = (S_{ij})_{n \times m}$. If vm_j is assigned to pm_i , then $S_{ij} = 1$; otherwise, $S_{ij} = 0$. During all the allocation processes, the following conditions must be fulfilled:

$$\begin{cases} \sum_{i=1}^n p_{ij} \cdot s_{ij} = 1 \\ \sum_{i=1}^n p_{ij} \cdot s_{ij} \cdot tr_j \leq sr_i \\ \sum_{i=1}^n \sum_{j=1}^m p_{ij} \cdot s_{ij} \cdot tr_j \leq \sum_{i=1}^n sr_i \end{cases}$$

2.2 Multi-dimensional Optimization Goal

Description to the deployment problem of multi-dimensional resource constraint vector VMs: there are n physical nodes ($PM = \{pm_1, pm_2, pm_3, \dots, pm_n\}$, n is number of physical machine nodes in the cluster). Resources in this paper mainly include four dimensions, namely, CPU, Memory, disk and network bandwidth resources. There are m VMs that have to be deployed ($VM = \{vm_1, vm_2, vm_3, \dots, vm_m\}$, m is number of VMs that has to be allocated). Resource demands of these VMs include memory, CPU, disk and network bandwidth resources. At this moment, it is necessary to find a reasonable mapping between VM and physical node to meet resource demands of VMs, reduce physical nodes that have to be started as much as possible and increase resource utilization and decrease energy consumption of physical nodes while maintaining healthy operation of users' tasks. The total resource demand of VMs in the mapping shouldn't exceed resource capacity of the corresponding physical node.

Definition 1: the maximum resource vector that pm_i could provide is $r_i = (r_i(1), r_i(2), r_i(3), \dots, r_i(t))^T$, where $r_i(t)$ is the maximum amount of t -type resource provided by pm_i , $1 \leq i \leq n$. Then, the residual maximum resource matrix of physical node set is defined as:

$$R = \begin{bmatrix} r_1(1) & r_1(2) & r_1(3) & \dots & r_1(t) \\ r_2(1) & r_2(2) & r_2(3) & \dots & r_2(t) \\ r_3(1) & r_3(2) & r_3(3) & \dots & r_3(t) \\ \dots & \dots & \dots & \dots & \dots \\ r_n(1) & r_n(2) & r_n(3) & \dots & r_n(t) \end{bmatrix}^T$$

Definition 2: vm_j needs a certain amount of resources for execution: $vr_j = (vr_j(1), vr_j(2), vr_j(3), \dots, vr_j(t))^T$, where $vr_j(t)$ is the amount of t -type resources needed by vm_j , $1 \leq j \leq m$. The resource demand matrix of VR set is defined as:

$$VR = \begin{bmatrix} vr_1(1) & vr_1(2) & vr_1(3) & \dots & vr_1(t) \\ vr_2(1) & vr_2(2) & vr_2(3) & \dots & vr_2(t) \\ vr_3(1) & vr_3(2) & vr_3(3) & \dots & vr_3(t) \\ \dots & \dots & \dots & \dots & \dots \\ vr_n(1) & vr_n(2) & vr_n(3) & \dots & vr_n(t) \end{bmatrix}^T$$

Definition 3: the mapping matrix from vm_j to pm_i is MP :

$$MP = \begin{bmatrix} mp_{11} & mp_{12} & mp_{1j} & \dots & mp_{1m} \\ \dots & \dots & \dots & \dots & \dots \\ mp_{i1} & mp_{i2} & mp_{ij} & \dots & mp_{im} \\ \dots & \dots & \dots & \dots & \dots \\ mp_{n1} & mp_{n2} & mp_{nj} & \dots & mp_{nm} \end{bmatrix}, 1 \leq i \leq n, 1 \leq j \leq m, mp_{ij} \in \{0,1\}.$$

In MP , mp_{ij} reflects whether v_j is put on the physical node p_i . $mp_{ij} = 1$ is yes and $mp_{ij} = 0$ is no.

Definition 4: $P = \sum_{i=1}^n mp_{ij}$ is the number of needed physical nodes to accomplish the user task.

Definition 5: $U_{i,q}$ is utilization of resource q on pm_i .

When users submit VM deployment task to the cloud computing center, the optimization goal is:

$$\begin{cases} \min E \\ \max \sum_{i=1}^n \sum_{q=1}^t U_{i,t} \end{cases} \quad (1)$$

where E is Euclidean distance between dimensions of resources ($E = [(u_i^c - u^{bc})^2 + (u_i^m - u^{bm})^2 + (u_i^d - u^{bd})^2 + (u_i^b - u^{bb})^2]^{1/2}$, which is defined in section 2.4.2). At this moment, utilization of resources could not exceed the ideal critical values (For example, $CPU \leq 70\%$, $Hard\ disk\ resources \leq 50\%$, $Memory\ resources \leq 50\%$ and $Network\ resources \leq 70\%$).

2.3 Multi-dimensional Resource Constraint Relationship

Generally speaking, the total resource demand of VM couldn't exceed the available physical resources in physical machine cluster at any moment. The total resource demands of VMs on every physical machine shall be limited within the total physical resources of the physical machine. Expressions of specific constraints are shown in equation (2) and equation (3):

$$\begin{cases} \forall r_i \in R_{D(h)}, \bigcap_{i=0}^n r_i = \phi \\ \sum U_{j,t}^i \leq T_{i,t} \\ \forall t, \sum_{i=1}^n \sum_{j=1}^m mp_{ij} \times vr_j \leq r_i \\ \sum_{j=1}^m vr_j \leq R_{D(h)} \end{cases} \quad (2)$$

In equation (2), intersection among any resource on every physical node is a null set. That is to say that the same physical resource cannot be used at the same time. $T_{i,t}$ is the threshold value of pm_i to t-type resource and $U_{j,t}$ is t-type resource utilization of vm_j on pm_i . vr_j is the amount of resources needed by vm_j . r_i denotes the resource set of i^{th} physical machine.

$$\left\{ \begin{array}{l} \sum_j vm_{ij}^c * mp_{ij} \leq 0.7 * pm_i^c \\ \sum_j vm_{ij}^m * mp_{ij} \leq 0.5 * pm_i^m \\ \sum_j vm_{ij}^d * mp_{ij} \leq 0.5 * pm_i^d \\ \sum_j vm_{ij}^b * mp_{ij} \leq 0.7 * pm_i^b \end{array} \right. \quad (3)$$

In equation (3), system of inequalities represents threshold values of CPU, Memory, bandwidth and disk resources of all VMs on pm_i . vm_{ij}^c denotes the required CPU resource of vm_j which will be assigned to pm_i . vm_{ij}^m , vm_{ij}^d and vm_{ij}^b denote the required memory resource, disk resource and bandwidth resource of vm_j which will be assigned to pm_i . pm_i^c , pm_i^m , pm_i^d and pm_i^b denote the maximum value of CPU resource, memory resource, disk resource and bandwidth resource on pm_i . In this paper, maximum resource usages were: 70% CPU, 50% Hard disk, 50% Memory and 70% Network resources. And all the physical resource of datacentre must exceed the required resources which all the virtual machines needed at the same time.

2.4 Low Energy Parameter Selection Based on Euclidean Distance

2.4.1 Reason of Euclidean Distance Selection for VM Deployment

How to measure the differences between the actual deployment situation and supposed deployment situation is a very complexed problem. When we consider the problem, there are two choices: Euclidean distance and Mahalanobis distance.

Euclidean distance is usually a distance definition which is often used in many ways. It denotes the real distance between two points which are in the m dimensions space. In this paper, we calculate the similarity degree of resource utilization using Euclidean distance. The closer the distance is, the more similar the resource utility is. In this article, with the closer Euclidean distance, the actual utilization of the physical resource is more closer to the desired deployment result. On the other hand, with the more far distance, the actual virtual machine is more and more deviated from the deployment level required. In this paper, the virtual machine placement problem is considered with four dimensional resources. And at the same time, according to the practice experience value, we set a maximum use of the threshold for each dimension resource respectively. At this point, we are sure that Euclidean distance is very proper to be used.

At first we took consideration of Mahalanobis distance as a parameter to solve this problem because it could not be affected by the concrete dimension. But we found that the value of Mahalanobis distance was affected by some tiny variables easily. That is a fatal weakness. That is why we do not select Mahalanobis distance as the energy consumption parameter now.

With the closer Euclidean distance, the numbers of physical nodes which are enabled will be reduced. So the required physical nodes and other related resources are all will be decreased. From this view, some energy can be decreased quickly. So we select Euclidean distance as a parameter in energy consumption.

2.4.2 Main Calculation Method of Euclidean Distance

Comprehensive use of CPU, Memory, disk and bandwidth resources on physical machines in the could platform influences energy consumption and overall performances of physical machines significantly. YS.DOU et al. [20] implemented a deep experimental analysis on correlation between high performance of physical machines and multi-dimensional resources. Q. Li et al. [19] and YS.DOU et al. [20] pointed out that client application could be controlled by physical machine allocation. For instance, utilization of multi-dimensional physical resources can be measured by loading a corresponding tracer and energy consumption of physical machines could be tested by a power meter. In the experiment, utilization of resources increased gradually. Utilization of CPU, Memory, disk space and network bandwidth increases from 10% to 90% at a steplength of 20%. On this basis, performances of different application programs and energy consumption of physical nodes under different resource utilization could be concluded. According to the experimental results in [9][20], the minimum energy consumption of physical nodes and satisfying task performance were achieved when utilization of four resources (CPU, Memory, disk and network bandwidth) reaches 70%, 50%, 50% and 70%, respectively [9][20,21]. In technical report of Microsoft 2008, system performances stably when network bandwidth and Memory use are limited within 70% and 50%, respectively. Therefore, this paper defined 70%, 50%, 70% and 50% as the ideal fitting point of CPU utilization, disk utilization, network resource utilization and Memory utilization. Here, energy consumption and performances of physical nodes were measured by combining actual utilization of four resources on physical machines after VM deployment and Euclidean distance of ideal fitting points. The comprehensive Euclidean distance of actually started physical nodes was used as the main criterion of energy consumption of the whole system:

$$E = [(u_i^c - u^{bc})^2 + (u_i^m - u^{bm})^2 + (u_i^d - u^{bd})^2 + (u_i^b - u^{bb})^2]^{1/2} \quad (4)$$

In equation (4), n is number of started physical nodes; u_i^c , u_i^m , u_i^d and u_i^b are actual utilization of four-dimensional resources (CPU, Memory, disk and network bandwidth) on the i^{th} physical node after VMs deployment; u^{bc} , u^{bm} , u^{bd} and u^{bb} are utilizations of four-dimensional resources (CPU, Memory, disk and network bandwidth) of physical node to reduce energy consumption under the ideal state. E is the Euclidean distance between four-dimensional resources (CPU, Memory, disk and network bandwidth) utilization on all started physical machines in the could platform and the ideal fitting points after VMs deployment. This constraint, together with measured actual resource utilization, could help to determine utilization thresholds of four-dimensional resources (CPU, Memory, disk and network bandwidth) at the minimum energy consumption. This is closer to real cloud environment. The proposed PSO-based VM deployment strategy that takes multi-dimensional resource constraints as premise is mainly established on the basis of equation (3) (u_i^{bc} , u_i^{bm} , u_i^{bd} and u_i^{bb} in equation (4) were determined 70%, 50%, 50% and 70% according to experimental results in [20] and the Microsoft report 2008).

3. VM Deployment Method Based on Euclidean Distance

Most optimization problems in actual physical environment can be summarized as a multi-objective optimization problem. However, different sub-objectives are often incompatible. In other words, improving one sub-objective is very likely to cause harmful effects on another or several other sub-objectives. Therefore, it is extremely difficult or even impossible to achieve the optimum state of multiple sub-objectives at the same time. In practice, it can only coordinate multiple sub-objectives to optimize the overall objective as much as possible [24,29].

In general, virtual machine deployment is optimized based on a single dimension resource. We combined CPU, memory, disk with bandwidth together. From the view of four dimensional resource, virtual machine is placed. Our main goal is raising the resource efficiency by providing physical resource utilization efficiency. In this way, the enabled physical nodes required can be reduced, so the physical energy consumption is saved also. Achieving the deployment of virtual machine is based on the perspective of multidimensional resources in this paper. And the energy standard is based on the Euclidean distance of the multidimensional resources. It is a multi-objective optimization problem. During the process of deployment, Virtual machine is considered as a separate particle. Our deployment is mainly divided into three periods. First, particle positions will be initialized according to the improved first-fit algorithm. Second, the particle positions will be updated according to the evaluation criteria. Third, positions rationalization of the virtual machines will be evaluated. Particle initialization process is showed in section 3.3.2. The process of updating particle positions is showed in section 3.3.3. The judgment of positions rationalization is described in algorithm 3. In section 3.4, we show the whole virtual machines deployment algorithm.

3.1 Related Concepts

Definition 6: Multi-objective optimization

Multi-objective optimization is a problem to optimize multiple objectives on the given region [27]. Without loss of generality, this paper defined optimization as minimization. $\min f(x) = (f_1(x), f_2(x), \dots, f_k(x))$, where the decision vector x is composed of n -dimensional decision variables; the objective vector $f(x)$ is composed of k -dimensional objectives; and the objective function $f(x)$ is the mapping from n -dimensional decision space to the k -dimensional objective space.

Definition 7: Weak Pareto dominance

If $f(x) \leq f(y)$ on any objective, y is dominated by x weakly.

Definition 8: Pareto dominance

For set decision vectors x and y , if $f(x) \leq f(y)$ on any objective and $f(x) < f(y)$ on at least one objective, y is dominated by x . If there's mutual dominance between x and y , x and y could be compared. If $f(x) = f(y)$ on all objectives, x and y are equivalent. x and y couldn't be compared if no mutual dominance or equivalence is found between them [22, 23].

Definition 9: Optimal solution set of Pareto

In general cases, there are many decision vectors in the decision space. If there is no decision vector that could dominate x , x is called as one Pareto optimal solution. The set covering all Pareto optimal solutions is the Pareto [22, 23] optimal solution set. Fig. 1 is a two-dimensional objective space. No mutual dominance was observed among A, B, C and D. Besides, they were not dominated by any solutions. Solution set represented by the arc ABCD

is the Pareto optimal solution set and the arc ABCD is the leading edge of Pareto. B dominates F and E is dominated by A, but there's no dominance relation among G, H, E and F. Moreover, the final goal of multi-objective optimization is to get a group of solutions with the best convergence and diversity in the objective space.

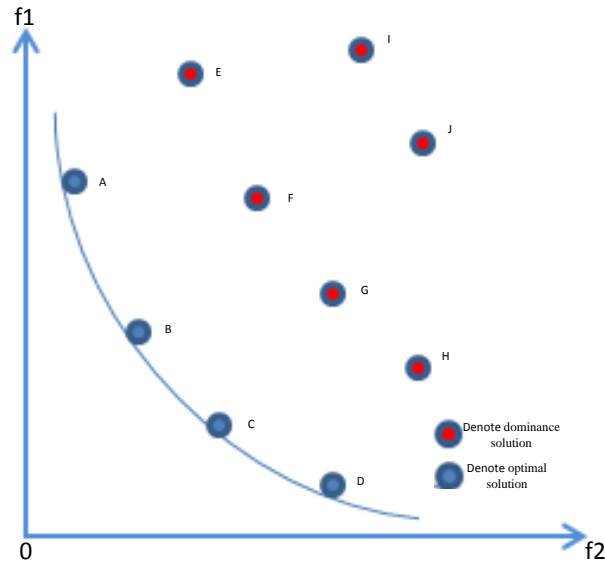


Fig. 1. Two-dimensional map of objective spatial distribution

3.2 First-fit Algorithm Based on Resource Constraint

Allocate M virtual machines onto N physical nodes. Generally speaking, M is far higher than N . Generate a serial sequence including M virtual machines randomly. The random sequence is then placed into physical nodes one by one by the priori matching heuristic algorithm. Firstly, the first physical node in all used physical nodes is chosen. If four-dimensional resources (CPU, Memory, Disk and network bandwidth) amount of this physical node could satisfy four-dimensional resources requirements of the first virtual machine node, the first virtual machine is selected as the physical node. Otherwise, match it with following used physical nodes one by one until finding the satisfying physical node (suppose N physical nodes could meet allocation requirement of M virtual machines). If no appropriate physical node could be found in all used physical nodes, the first unused physical node is chosen to place the virtual machine (all physical nodes are unused in the beginning). Place other virtual machines into physical nodes through the same way. The improved first-fit algorithm with resource constraint is presented in Algorithm 1:

Algorithm 1: Improved First-fit Algorithm based on Resource Constraint

```

For( $i=0; i <= n; i++$ )
{
  VMFlag=0;
  For( $j=0; j <= m; j++$ )
  {
    If ( $VM_{pj} == 1$ )
    {

```

```

If (  $R_{pj}(0.7c,0.5m,0.5d,0.7b) > R_{vi}(c,m,s,b)$  )
  {
    VMi stored in Pj;
    VMFlag=1;
  }
Else
  j++;
}
}
If (VMflag==0)
  VMi stored in the first Pj which is not used
}

```

3.3 Position Updating Strategy Based on Improved PSO

Particle swarm optimization (PSO), an optimization algorithm which was established based on foraging behavior law of birds in late 1990s, could be used for continuous space optimization. It emphasizes on promoting begin development of particle swarm through the information share mechanism. Compared to other evolution algorithms, PSO has fewer parameters, quicker convergence and simpler operation. It is highly appreciated by researchers. Since VM deployment under specific conditions could be viewed as a discretization combinational optimization problem [20, 22, 23], it could be solved effectively by re-defining particles in the particle swarm and the particle updating process.

3.3.1 Definition of Fitness Function

The objective design fitness function of VM deployment is:

$$f(E) = \sum_{i=1}^n [(u_i^c - u^{bc})^2 + (u_i^m - u^{bm})^2 + (u_i^d - u^{bd})^2 + (u_i^b - u^{bb})^2]^{1/2} \quad (5)$$

where $f(E)$ is Euclidean distance [20,21] sum between four-dimensional resources utilization (CPU, Memory, disk and network bandwidth) on every physical node and the ideal fitting points of the physical node.

3.3.2 Initial Positions of Particles in Particle Swarm

Position vector of particles $x_s^r = \{x_s^{r1}, x_s^{r2}, x_s^{r3}, \dots, x_s^{rj}, \dots, x_s^{rm}\}$ is the r^{th} feasible solution to VM deployment, where x_s^{rj} is serial number of the physical machine to which virtual machine j is allocated during particle position updating of the s^{th} -generation VM. Among r feasible solutions to VM deployment, $x_s^{ri}, x_s^{rj}, x_s^{rk}$ value (x, y, z) . It means that virtual machines i, j and k are deployed on physical nodes x, y and z , respectively. When updating particle positions, particle positions were transformed into a 0, 1 matrix:

$$MP = \begin{bmatrix} mp_s^{11} & mp_s^{12} & \dots mp_s^{1i} & \dots & mp_s^{1n} \\ \dots & \dots & \dots & \dots & \dots \\ mp_s^{j1} & mp_s^{j2} & \dots mp_s^{ji} & \dots & mp_s^{jn} \\ \dots & \dots & \dots & \dots & \dots \\ mp_s^{m1} & mp_s^{m2} & \dots mp_s^{mi} & \dots & mp_s^{mn} \end{bmatrix}$$

In position updating of the s^{th} -generation particles, whether the j^{th} virtual machine is allocated to the i^{th} physical node is expressed by mp_s^{ji} . If yes, $mp_s^{ji} = 1$; otherwise, $mp_s^{ji} = 0$. The j^{th} -dimensional value of x_s^r is an absolute influencing factor of mp_s^{ji} . Since one virtual machine can only be placed onto one physical node, $\forall j \in \{1, 2, 3, \dots, m\}$, $\sum_{i=1}^n mp_s^{ji} = 1$.

3.3.3 Particle Position Updating Mechanism

In PSO, particle position updating is determined by the optimal solution of individual particle and the global optimal solution of all particles together [20][22][23]. $pb_s^r = (pb_s^{r1}, pb_s^{r2}, pb_s^{r3}, \dots, pb_s^{rj}, \dots, pb_s^{rm})$ is the optimal solution of the r^{th} particle after s iterations, where pb_s^{rj} is the coordinate position of the j^{th} -dimension of r 's optimal solution. $gb_s = (gb_s^1, gb_s^2, gb_s^3, \dots, gb_s^j, \dots, gb_s^m)$ is the global optimal solution of all VM position in VM particle swarm after s iterations. gb_s^j is the j^{th} -dimensional position value of global optimal solution of all VMs. In particle swarm updating of VM, pb_s^r and gb_s^r could be expressed by position matrices. Position value of the j^{th} and i^{th} rows are expressed by pb_s^{ji} and gb_s^{ji} , respectively.

This paper assumed that the probability for the j^{th} virtual machine to be deployed on physical node i during position updating of the $(s+1)^{\text{th}}$ -generation particles under any state meets $p(mp_{s+1}^{ji} = 1) = 0.5$, $p(mp_{s+1}^{ji} = 0) = 0.5$. Here, p_p^b represents reliability of individual optimal solution and p_g^b is the reliability of global optimal solution. For the convenience of description, two parameters (λ_1, λ_2) were used. They could be calculated easily by combining ideal probability of different dimensional resources and the Bayes formula.

$$\begin{cases} \lambda_1 = P(mp_{s+1}^{ji} = 1 | pb_s^{ji} = 1, gb_s^{ji} = 1) = (p_p^b \times p_g^b) / [p_p^b \times p_g^b + (1-p_p^b)(1-p_g^b)] \\ \lambda_2 = P(mp_{s+1}^{ji} = 1 | pb_s^{ji} = 1, gb_s^{ji} = 0) = [p_p^b \times (1-p_g^b)] / [p_p^b \times (1-p_g^b) + p_p^b \times (1-p_p^b)] \\ 1-\lambda_2 = P(mp_{s+1}^{ji} = 1 | pb_s^{ji} = 0, gb_s^{ji} = 1) = [p_g^b \times (1-p_p^b)] / [p_p^b \times (1-p_g^b) + p_g^b \times (1-p_p^b)] \\ 1-\lambda_1 = P(mp_{s+1}^{ji} = 1 | pb_s^{ji} = 0, gb_s^{ji} = 0) = [(1-p_p^b) \times (1-p_g^b)] / [p_p^b \times p_g^b + (1-p_p^b)(1-p_g^b)] \end{cases} \quad (6)$$

Generally, particle swarm only can generate the individual optimal solution and global optimal solution after multiple iterations. Therefore, the optimal solution probability gained in the proposed algorithm shall meet $0.5 < p_p^b < 1$ and $0.5 < p_g^b < 1$. To make scatter search of particles and prevent local optimization, let $p_g^b < p_p^b$, that is, $0.5 < p_g^b < p_p^b < 1$. In this paper, $p_g^b = 0.75$ and $p_p^b = 0.85$. So $\lambda_1 = 0.944$ and $\lambda_2 = 0.654$. Position updating strategy of particle swarm is shown in Algorithm 2.

Algorithm 2: Position updating algorithm of particle swarm

Input: γ , pb_s^{ji} , gb_s^{ji}
Output: mp_{s+1}^{ji}
 $\gamma = \text{random}()$
if $((pb_s^{ji} = 1) \& \& (gb_s^{ji} = 1))$
{

```

    If (  $\gamma < \lambda_1$  )  $mp_{s+1}^j = 1$ 
    Else  $mp_{s+1}^j = 0$ 
  }
  If ( (  $pb_s^j = 1$  ) && (  $gb_s^j = 0$  ) )
  {
    If (  $\gamma < \lambda_2$  )  $mp_{s+1}^j = 1$ 
    Else  $mp_{s+1}^j = 0$ 
  }
  If ( (  $pb_s^j = 0$  ) && (  $gb_s^j = 0$  ) )
  {
    If (  $\gamma < 1 - \lambda_1$  )  $mp_{s+1}^j = 1$ 
    Else  $mp_{s+1}^j = 0$ 
  }
  Else
  {
    If (  $\gamma < 1 - \lambda_2$  )  $mp_{s+1}^j = 1$ 
    Else  $mp_{s+1}^j = 0$ 
  }

```

3.4 NVMDM Algorithm Based on Particle Swarm

(1) Population initialization

Generate a random sequence of virtual machines that have to be deployed. Virtual machine is placed on the first physical node that meets the resource constraint according to Algorithm 1. If there are n virtual machines, a total of n deployment schemes could be gained, that is, n particles in particle swarm.

(2) Calculate fitness value of initial population of NVMDM, so individual optimal solution and the global optimal solution of particles are acquired.

(3) Calculate the latest position of each particle according to particle position updating mechanism (Algorithm 2).

(4) Judge whether the updating result satisfies position updating requirement of particles in view of the latest position value of particles. Iteration times increases by 1.

(5) Determine whether the new position of every particle is reasonable according to rationality of new position of the particle (Algorithm 3). If all particle states meet the constraint, update particle position; otherwise, keep the old position.

(6) Judge whether iteration times have reached the maximum. If yes, end the iteration and output the global optimal solution of VM deployment; otherwise, turn to Step (2).

Algorithm 3: Rationality Judgment of Particle New Position

For (j=1; j<=m; j++)

{

 If ($\sum_{i=1}^n mp_s^i > 1$)

 {

```

Show the list of physical machine(LPM) which can satisfy the requirement of  $v_j$ ;
Record the sequence number of the physical machine on which  $v_j$  was placed;
Set  $\forall i, mp_s^i = 0$ ;
Select the first physical node of LPM as the first choice deployment node of  $v_j$ ;
Set Flag=1;
Update the position information of  $v_j$ ;
}
If ( $\sum_{i=1}^n mp_s^i = 1$ )
{
Select the physical node as the deployment node of  $v_j$ ;
Set flag=1;
Update the position information of  $v_j$ ;
}
If ( $\sum_{i=1}^n mp_s^i = 0$ )
{
Keeping the old position value information of  $v_j$ ;
}
}

```

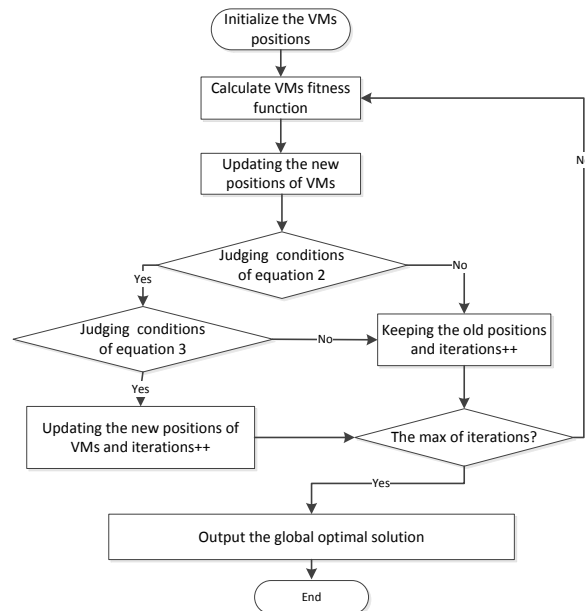


Fig. 2. A flow chart of VMs locations updating process

The proposed PSO-based NVMDM algorithm generates particles by the improved first-fit algorithm with resource constraint (Algorithm 2) tries to place particles on used physical nodes. This satisfies resource constraint of particles and ensures full utilization of physical node resources. Particles were evaluated by Euclidean distance and evolved toward high

energy efficiency. Position data of particles were updated by the improved binary discrete PSO. It determines new position of particles through probability directly rather than speed transform. This is more efficient than traditional methods. Additionally, new position data of particles that satisfy two constraints in Algorithm 2 are kept, which assures feasibility of new particle position effectively during the evolution process. As a result, the final search results can be regarded as feasible solutions. The proposed algorithm gets the final VM deployment through particle comparison and evolution, and gives more attention to integrity of final deployment.

4. Experimental Environment Setting and Result Comparison

4.1 Simulation Setting

In this part, a simulation experiment was conducted on the Cloudsim platform to verify validity of the proposed NVMDM algorithm. It was confirmed effective in reducing system energy consumption and increase utilization of physical resources. Number of physical machines, system energy consumption and different-dimensional resources utilization of MBFD [21], multi-dimensional resource scheduling DRF algorithm [26] and the proposed NVMDM algorithm were compared. Cloud platform of 50-500 virtual machines was simulated through the CloudSim simulation platform. Each physical node was equipped with two processors (Intel(R) Core(TM) i7-5500U 3.0GHz), 8G RAM, 4MB L3 Cache, two 5400r 500MB disks. All nodes were connected through Gigabit Ethernet. Parameters are listed in Table 1. Population size in the simulation experiment was set 40 and iteration times were set 50.

Table 1. Hardware environment settings of physical nodes

CPU	RAM	Disk	Network bandwidth	Algorithm
2	8G	1T	1.0Gbps	MBFD, DRF, NVMDM

4.2 Result Analysis

Total Euclidean distance of all physical nodes in the cloud platform is the sum of Euclidean distance between CPU utilization, Memory utilization, disk utilization and network resource utilization of physical nodes and the optimum state. Results of three algorithms as the number of virtual machines increases from 50 to 500 are displayed in Fig. 3 to Fig. 7.

The comparison result in Fig. 3 demonstrates the bandwidth utilities of MFDB, DFR and NVMDM in the same experimental conditions. With the variety of virtual machine numbers (range from 50 to 500), the bandwidth utilities of the above three algorithms are all increased respectively. From Fig. 3 we can find that NVMDM algorithm has the best bandwidth utility among MFDB, DFR and NVMDM. In Fig. 4 we can find that the concrete comparison results of different CPU utilities. With the increasing of virtual machine numbers, the CPU utilities of MFDB, DFR and NVMDM are all improved gradually. Under the same experimental conditions, the NVMDM algorithm has the best CPU utility which can reflect the high efficiency of NVMDM. It can be seen from Fig. 5, the utility of memory is changing accompany with the changing of virtual machine number. Among MFDB, DFR and NVMDM, MFDB algorithm has the lowest memory utility. And NVMDM has the highest memory utility. Though the utilities of different resources are increased with the increasing of number of virtual machines, all of them are in the different thresholds we have supposed in this paper. Fig. 6 shows volatility of disk utilization. If the number of virtual machines is small, MFDB and DFR algorithm are all with the low disk utility. But with the increasing number of virtual machines disk utilization has also improved. From the Fig. 6 we can find that NVMDM has

the highest disk utility among the three algorithms. It can be seen from Fig. 7 that Euclidean distance of MBFD algorithm and DRF algorithm increases sharply with the increase of virtual machines, but Euclidean distance of the proposed NMDM algorithm changes slightly. This indicates that the proposed NVMDM algorithm is significantly superior to MBFD algorithm and DRF algorithm in term of Euclidean distance.

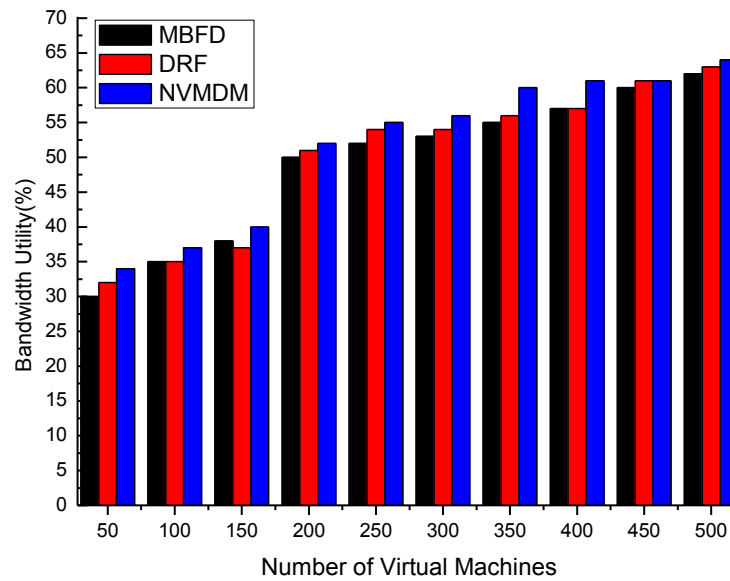


Fig. 3. Bandwidth utilization

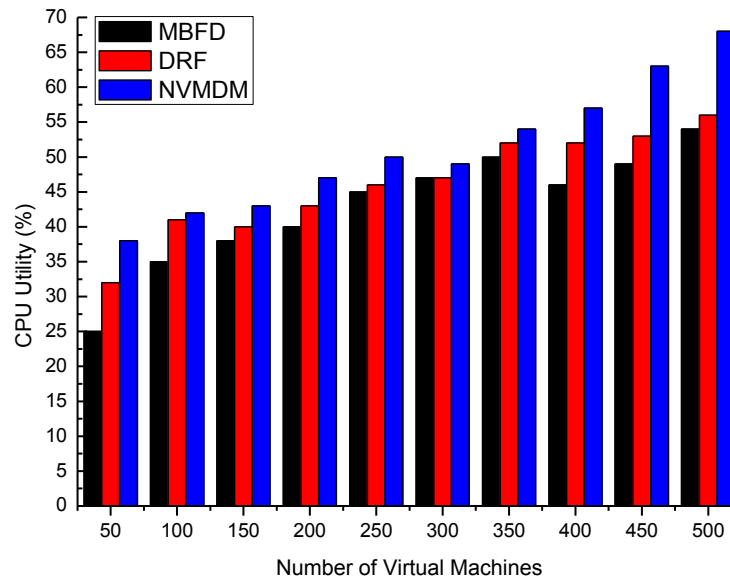


Fig. 4. CPU utilization

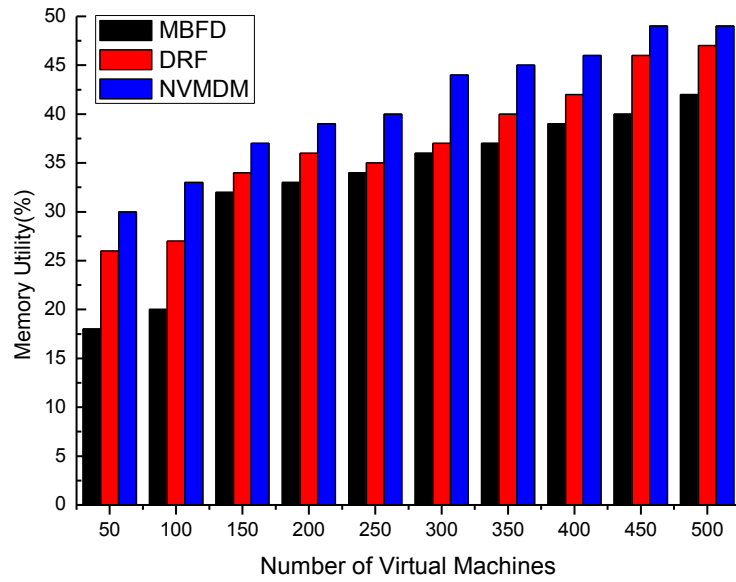


Fig. 5. Memory utilization

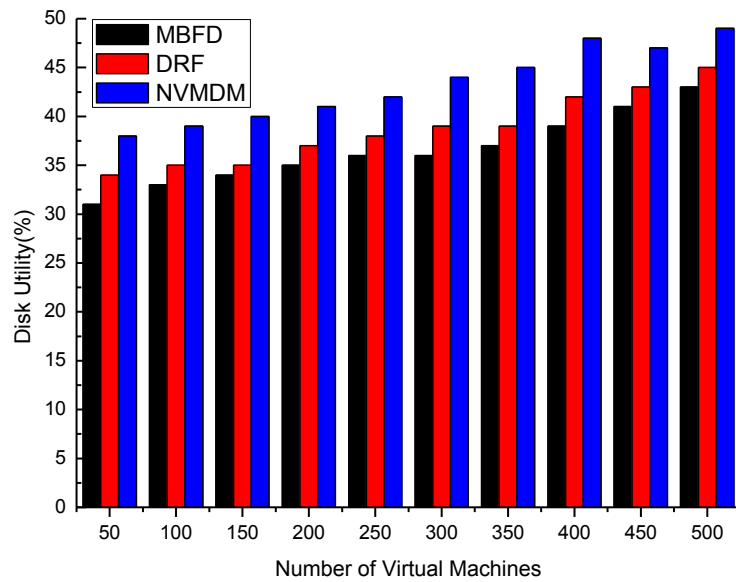


Fig. 6. Disk utilization

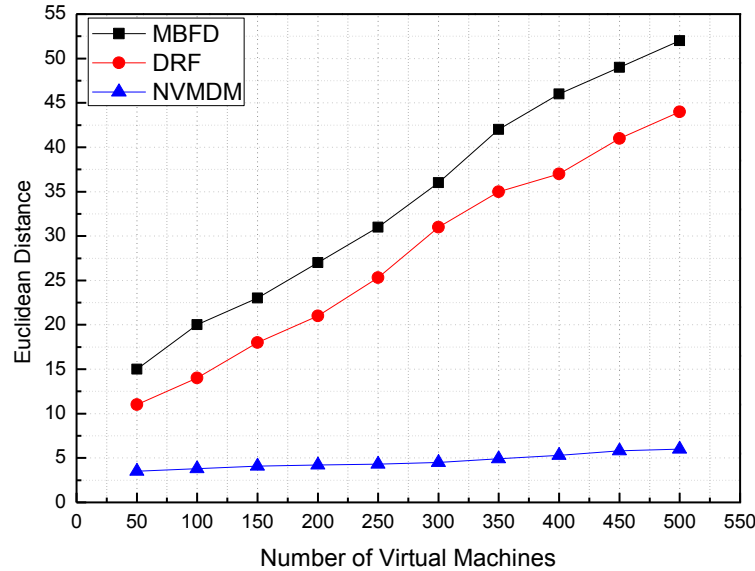


Fig. 7. Euclidean distance comparison

In a word, as virtual machines increase from 50 to 500 gradually, three algorithms have similar network bandwidth utilization when virtual machines increase from 50 to 500 gradually. NVMDM algorithm shows good utilization of network bandwidth when virtual machines exceed 200. The lowest CPU utilization of the proposed NVMDM algorithm is 38% and the maximum CPU utilization doesn't exceed constraint of CPU resources. Although CPU utilization of MBFD algorithm and DRF algorithm increases with the increase of virtual machines, the maximum CPU utilization is 56%, far lower than that of the proposed NVMDM algorithm. Memory utilization of NVMDM algorithm varies from 30% to 49%, while those of MBFD algorithm and DRF algorithm range from 18% to 42% and from 26% to 47%, respectively. Disk utilization is an important index of algorithm performance. Disk utilizations of MBFD algorithm and DRF algorithm are similar, ranging between 31% and 45%. Disk utilization of NVMDM algorithm is between 38% and 49%. Moreover, convergence of three algorithms was compared preliminarily. Although MBFD algorithm and DRF algorithm could converge as time goes on, they have slower convergence than the NVMDM algorithm.

To sum up, in this paper we made a comparative analysis on four-dimensional resources utilization, Euclidean distance and convergence of three algorithms and find that the proposed NVMDM algorithm is significantly superior to MBFD algorithm and DRF algorithm in all aspects.

5. Conclusions and Future Work

This paper mainly studies virtual machine deployment strategy based on Euclidean distance and multi-objective particle swarm optimization. Combining with energy consumption goal (Euclidean distance), an improved first-fit algorithm based on resource constraint (IFFABRC) is proposed under the basic premise of effective constraints on CPU, Memory, disk and network resources. Initial population sequence of particles is generated on the basis of the

IFFRCA. Then, the novel virtual machine deployment method (NVMDM) based on PSO is further put forward. A simulation experiment on MBFD algorithm, DFR algorithm and NVMDM algorithm is carried out. It confirms that NVMDM algorithm is superior to MBFD algorithm and DFR algorithm with respect to multi-dimensional resources utilization and system energy efficiency. However, its superiority in bandwidth resource utilization remains concealed until number of virtual machines reach a certain level. Further research will focus on related causes and solutions.

The utilities of the four dimensional resources (CPU, Memory, Disk and Bandwidth) can be improved in some extent. During the process of virtual machines deployment, the four dimensional resources can be configured efficiently and the enabled physical nodes can be reduced under the same resource requirements with NVMDM algorithm proposed in this paper. But the fault-tolerant mechanism is not considered during the process of VMs deployment. And at the same time the fault-tolerant mechanism is not treated as a metric which can be used to judge the final effect of virtual machine deployment. NVMDM is benefited to improve the resource utility, decrease the enabled physical nodes and promote the stability of cloud platform. Due to the lack of fault tolerant strategy, it can lead to losses to real time virtual machine deployment sometimes.

Therefore, our future work will focus on adding fault-tolerant mechanism properly according to personal requirements and characters of real time problem in the process of virtual machine deployment.

References

- [1] Hong, Hua-Jun, Chen, De-Yu, Huang, Chun-Ying, Chen, Kuan-Ta, Hsu, Cheng-Hsin, "Placing virtual machines to optimize cloud gaming experience," *IEEE Transactions on Cloud Computing*, vol.3, no.1, pp.42-53, 2015. [Article \(CrossRef Link\)](#).
- [2] Sait, Sadiq M., Shahid, Kh. Shahzada, "Engineering Simulated Evolution for Virtual Machine Assignment Problem," *Applied Intelligence*, vol.43, no.2, pp.296-307, 2015. [Article \(CrossRef Link\)](#).
- [3] Sun Gang, Liao Dan, Anand, Vishal, Zhao Dongcheng, Yu Hongfang, "A new technique for efficient live migration of multiple virtual machines," *Future Generation Computer Systems*, vol.55, pp.74-86, 2015. [Article \(CrossRef Link\)](#).
- [4] Chokoufe Nejad, Bijan, Ohl, Thorsten, Reuter, Jürgen, "Simple parallel virtual machines for extreme computations," *Computer Physics Communications*, vol.196, pp.58-69, 2015. [Article \(CrossRef Link\)](#).
- [5] Peng Zhiping, Xu Bo, Gates Antonio Marcel, Cui Delong, Lin Weiwei, "The feasibility and properties of dividing virtual machine resources using the virtual machine cluster as the unit in cloud computing," *KSII Transactions on Internet and Information Systems*, vol.9, no.7, pp.2649-2666, 2015. [Article \(CrossRef Link\)](#).
- [6] Bhutani, Akshi, Jauhari, Isha, Kaushik, Vinay Kumar, "Optimized virtual machine tree based scheduling technique in cloud using K-way trees," in *Proc. of Proceedings-2015 International Conference on Cognitive Computing and Information Processing*, April 30, pp.1-6, 2015. [Article \(CrossRef Link\)](#).
- [7] Cerroni Walter, "Network performance of multiple virtual machine live migration in cloud federations," *Journal of Internet Services and Applications*, vol.6, no.1, pp.1-20, 2015. [Article \(CrossRef Link\)](#).
- [8] Zhang Xiaoqing, Qiu Lan, Qian Qiongfeng, Li Yaqin, "Virtual machines consolidation and placement based on constraint satisfaction in the clouds," *Journal of Computational Information*

- Systems*, vol.11, no.14, pp.5251-5258, 2015. [Article \(CrossRef Link\)](#).
- [9] Joshi, Sourabh, Kaur, Sarabjit, "Cuckoo search approach for virtual machine consolidation in cloud data centre," in *Proc. of International Conference on Computing, Communication and Automation, ICCCA 2015*, pp. 683-686, 2015. [Article \(CrossRef Link\)](#).
- [10] M. Stillwell, D. Schanzenbach, F. Vivien, and H. Casanova, "Resource allocation algorithms for virtualized service hosting platforms," *Journal of Parallel and distributed Computing*, vol.70, no.9, pp.962-974, 2010. [Article \(CrossRef Link\)](#).
- [11] J. Xu and J. A. Fortes, "Multi-objective virtual machine placement in virtualized data center environments," in *Proc. of Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on & Int'l Conference on Cyber, Physical and Social Computing (CPSCom)*, pp. 179-188, 2010. [Article \(CrossRef Link\)](#).
- [12] X. Kong, C. Lin, Y. Jiang, W. Yan, and X. Chu, "Efficient dynamic task scheduling in virtualized data centers with fuzzy prediction," *Journal of network and Computer Applications*, vol.34, no.4, pp.1068-1077, 2011. [Article \(CrossRef Link\)](#).
- [13] D. Warneke and O. Kao, "Exploiting dynamic resource allocation for efficient parallel data processing in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol.22, no.6, pp.985-997, 2011. [Article \(CrossRef Link\)](#).
- [14] J. Xu and J. A. Fortes, "Multi-objective virtual machine placement in virtualized data center environments," in *Proc. of Green Computing and Communications (Green Com), 2010 IEEE/ACM Int'l Conference on & Int'l Conference on Cyber, Physical and Social Computing (CPS Com)*, pp.179-188, 2010. [Article \(CrossRef Link\)](#).
- [15] M. Chen, H. Zhang, Y.-Y. Su, X. Wang, G. Jiang, and K. Yoshihira, "Effective VM sizing in virtualized data centers," in *Proc. of Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on*, pp.594-601, 2011. [Article \(CrossRef Link\)](#).
- [16] Li MF, Bi JP, Li ZC, "Resource Scheduling Waiting Aware Virtual Machine Consolidation," *Journal of Software*, vol.25, no.7, pp.1388-1402, 2014. [Article \(CrossRef Link\)](#).
- [17] S. Sawant, "A genetic algorithm scheduling approach for virtual machine resources in a cloud computing environment," *Master's Projects*. Paper 198, 2011. [Article \(CrossRef Link\)](#).
- [18] J. Gu, J. Hu, T. Zhao, and G. Sun, "A new resource scheduling strategy based on genetic algorithm in cloud computing environment," *Journal of Computers*, vol.7, no.1, pp.42-52, 2012. [Article \(CrossRef Link\)](#).
- [19] Q. Li, Q.-F. Hao, L.-M. Xiao, and Z.-J. Li, "Adaptive management and multi-objective optimization for virtual machine placement in cloud computing," *Chinese Journal of Computers*, vol. 34, no.12, pp.2253-2264, 2011. [Article \(CrossRef Link\)](#).
- [20] DOU Yu-sheng, CUI Cheng-yuan, TANG Hong, LI Hong-jian, "An Energy-efficient Virtual Machine Placement Algorithm in Cloud Data Center," *Journal of Chinese Computer Systems*, vol.35, no.11, pp.2543-254, 2014. [Article \(CrossRef Link\)](#).
- [21] Beloglazov A, Abawajy J, Buyya R, "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 \(CrossRef Link\)](#).
- [22] Hui Li, Qingfu Zhang, "Multi-objective Optimization Problems with Complicated Pareto Sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol.13, no.2, pp.284-302, 2009. [Article \(CrossRef Link\)](#).
- [23] Xu heming, "Research of multi-objective particle swarm optimization algorithm" (PHD Dissertation), Shanghai Jiaotong University, 2013.
- [24] Calheiros R N,Ranjan R,Beloglazov A, "CloudSim:a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice*

- and Experience*, vol.41, no.1, pp.23-50, 2011. [Article \(CrossRef Link\)](#).
- [25] Eberhart R, Kennedy J., “A new optimizer using particle swarm theory. Micro Machine and Human Science,” in *Proc. of the Sixth International Symposium on Institute of Electrical and Electronics Engineers*, pp.39-43, 1995. [Article \(CrossRef Link\)](#).
- [26] Ping Guo and Qi li, “Load balance scheduling algorithm based on the load on the server status classification,” *Journal of Huazhong University of science and technology: Natural Science Edition*, vol. 40, no. 1, pp.62-65, 2012. [Article \(CrossRef Link\)](#).
- [27] Dawei Sun, Guiran Chang, Fengyun Li, Chuan Wang, and Xingwei Wang, “Optimizing multi-dimensional QoS cloud resource scheduling by immune clonal with preference,” *Acta Electronica Sinica*, vol. 39, no.8, pp.1824-1831, 2011. [Article \(CrossRef Link\)](#).
- [28] Kumar, M.R.V. and S. Raghunathan, “Heterogeneity and thermal aware adaptive heuristics for energy efficient consolidation of virtual machines in infrastructure clouds,” *Journal of Computer and System Sciences*, vol.82,no.2, pp.191-212, 2016. [Article \(CrossRef Link\)](#).
- [29] Kaur, P. and A. Rani, “Virtual Machine Migration in Cloud Computing,” *International Journal of Grid and Distributed Computing*, vol.8, no.5, pp. 337-342. 2015. [Article \(CrossRef Link\)](#).
- [30] Mann, Z.Á., “Allocation of virtual machines in cloud data centers—a survey of problem models and optimization algorithms,” *ACM Computing Surveys* ,vol.48,no.1, pp.11:1-11:34(Article No.11), 2015. [Article \(CrossRef Link\)](#).



Shukun Liu received the M.S. degree in computer science and technology from University of South China, PR China, in 2007. Currently, He has been a Ph.D. candidate at Central South University since September 2013, China. His major research interests include cloud computing, virtualization technology, performance analysis, computer networks, database technology, data mining and software engineering. He has published nearly fifteen papers in related journals. And he is a member of CCF and a member of ACM.



Prof. Weijia Jia is a Zhiyuan Chair Professor in the Department of Computer Science and Engineering, Shanghai Jiaotong University. He joined German National Research Center for Information Science (GMD) in Bonn (St. Augustine) from 1993 to 1995 as a research fellow. During 1995-2013, he joined Department of Computer Science, City University of HK as a professor. Weijia Jia received BSc and MSc from Center South University, China in 1982 and 1984 and Master of Applied Sci. and PhD from Polytechnic Faculty of Mons, Belgium in 1992 and 1993 respectively, all in Computer Science. He is the guest Professor of Beijing University of Science and Technology of China, Beijing Jiao Tong University, Jinan University, Guangzhou, Chengdu University, China. He has served as the editors of IEEE TPDS and ComCom and PC chairs and members/keynote speakers for various prestige international conferences. He is the Senior Member of IEEE and the Member of ACM. His research interests include next generation wireless communication, protocols and heterogeneous networks; distributed systems, multicast and anycast QoS routing protocols. In these fields, he has a number of publications in the prestige international journals (IEEE Transactions, e.g., ToN, TPDS, TC, TMC etc.), books/chapters and refereed international conference proceedings (e.g. ACM CCS, WiSec, MobiHoc, SenSys, IEEE ICDCS, INFOCOM etc.). He has received the best paper award in a prestige (IEEE) conference.