Received May 30, 2017, accepted June 17, 2017, date of publication June 29, 2017, date of current version July 24, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2721548

Energy-Efficient Virtual Resource Dynamic Integration Method in Cloud Computing

YINGYOU WEN[1,2], ZHI LI[2], SHUYUAN JIN[3], CHUAN LIN[1], AND ZHENG LIU[1]

[1]School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China
[2]Neusoft Research, Shenyang 110179, China
[3]School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China

Corresponding author: Zheng Liu (liuzheng@mail.neu.edu.cn)

This work was supported in part by the National High Technology Research and Development Program of China under Grant 2015AA016005 and in part by the National Natural Science Foundation of China under Grant 60903159, Grant 61173153, and Grant 61300196.

ABSTRACT In recent years, with the development of cloud computing technology, the size of a data center is expanding rapidly. To minimize the energy consumption of a data center, we propose an energy-efficient virtual resource dynamic integration (VRDI) method. In the proposed VRDI method, first, by monitoring the load patterns of the physical machines (PMs) and the corresponding thresholds of PMs calculated using the statistical data, we propose a PM selection algorithm to find a set of PMs, which should be integrated. Furthermore, we propose a virtual machine (VM) selection algorithm based on minimum migration policy to select the VMs that are deployed on the integrated PMs. Finally, to solve the target VM placement, we propose a VM placement algorithm based on an improved genetic algorithm. Using the encoding, crossover and mutation operations of the genetic algorithm, we obtain an effective solution for the VM placement problem. The experiments show that the proposed VRDI method can reduce the energy consumption of data center and ensure the quality of service of the cloud applications developed on the VMs.

INDEX TERMS Cloud computing, energy consumption, genetic algorithm, green data center, VM migration.

I. INTRODUCTION

Cloud computing has been receiving considerable attention from both the computing industry and academia. Cloud computing is based on virtualization technology, and abstracts the physical resources of a data center as virtual resources that are isolated from each other. Therefore, a single physical machine (PM) can virtualize multiple independent virtual machines (VMs), and provide different services. Cloud computing provides a simple pay-as-you-go business model for customers. Using virtualization technology, a service provider can build a flexible, transparent, resilient and scalable computing environment that meets the requirements of various applications, and increase resource utilization [1]–[3]. With the development and popularity of social networks, e-commerce, streaming media, search engines and other technologies, the demand for computing resources is increasing, and the scale of a data center is also gradually increasing [4], [5]. On the other hand, many enterprises and organizations deploy their services and applications in the public cloud, and purchase resources in the form of VMs. Using this approach, they can reduce the cost of constructing and maintaining a data center. As a result, the size of a data center and its energy consumption is increasing [6].

In recent years, green computing has received much attention from the computing research community [7]–[9]. The concept of green computing is not only limited to the energy consumption of computing devices such as servers, but also includes the energy consumption of network equipment, cooling equipment and so on. Considering the energy consumption of other equipment is out of the scope of the present contribution. In this work, we consider only the high energy consumption of a data center caused by the inefficient allocation of resources. The high energy consumption of a data center and the scale of the expansion of the data center have a direct relationship. Moreover, inefficient allocation of resources also contributes to a lot of energy consumption. According to the study of Barroso *et al.* [10], the data collected from more than 5000 production servers over a six-month period showed that the utilization of PMs reached only 10-50% of their full capacity most of the time. However, the energy consumption is about 70% compared with the fully loaded PM. Therefore, this practical phenomenon leads to a lot of energy wastage. Most of the research about a green data center is based on resource integration. Using resource integration, the PM which has lower resource utilization can be powered off and the VMs developed on it can be migrated to other PMs, and the energy consumption of data center can be reduced. Although resource integration and VM migration technology has made a great progress, it still has the following drawbacks:

1) Many VM migration algorithms do not consider the impact of VM migration on the QoS of cloud applications.
2) Many scholars use genetic algorithms to solve the problem of VM placement, and set the number of iterations of the genetic algorithm based on experience. However, the accuracy of the results depends on chosen parameters of the genetic algorithm. So, the strategy of setting the number of iterations should be improved.
3) The energy consumption model established by researchers is based on the linearity of CPU utilization. However, with the changing of the hardware in a data center, the linearity model may not accurately estimate the energy consumption of the data center.

For allocating the virtual resources of a data center, this paper proposes a novel energy-efficient virtual resource dynamic integration (VRDI) method to meet the QoS requirements of cloud applications. Firstly, according to the load pattern of the PMs of a data center, we find a PM set to be integrated; Secondly, for each PM in the integrated PM set, the migrated VM set is determined based on the load pattern and the relationship between the VM load and the PM load; Finally, for each VM in the migrated VM set, we find a new PM which can satisfy its resource requirements. Based on the above description, the main contributions of this paper are as follows:

1) We propose a PM selection algorithm to reduce energy consumption. For this purpose, we set the thresholds of resource utilization based on the load pattern of PMs.
2) Because the VM migration time will affects the QoS of the cloud applications, in this paper, we propose a VM selection algorithm based on minimum migration policy which effectively minimizes the number of VMs to be migrated.
3) To solve the problem of VM placement, we propose a VM placement algorithm based on an improved genetic algorithm. The proposed solution effectively solves the NP-Hard problem of VM placement.

The reminder of this paper is organized as follows: we provide a literature survey of the existing data center resource integration methods in Section 2. In Section 3, we present the system architecture, and introduce the three algorithms proposed in this work. In Section 4, we describe the experimental design and present the experimental results. Finally, we present conclusions and future research directions in Section 5.

II. RELATED WORKS

With the increasing popularity of the concept of green computing, a lot of research has been carried out with a focus on the energy consumption in a data center. In [11], the authors proposed the Bayesian migration heuristic (BMH) method. The BMH method is a heuristic resource integration method based on Bayesian networks. Using the Bayesian estimates, the BMH method generated the VM set to be migrated. The experimental results showed that the BMH could effectively reduce the energy consumption of a data center.

Liu *et al.* [12] proposed the GreenCloud architecture, which utilized a heuristic algorithm to optimize the placement strategy of VMs using the VM dynamic migration technology. The experimental results showed that the proposed solutions saved 27% of the energy consumption compared with the traditional cloud computing architecture.

In [13], the modified best fit decreasing (MBFD) algorithm for VM placement is proposed. First, the MBFD algorithm arranged the VMs in the decreasing order of CPU utilization. Further, the MBFD algorithm migrated VMs by combining the energy consumption of PMs and the QoS of VMs. Bobroff *et al.* [14] used a threshold for CPU utilization. When the CPU utilization reached the threshold limit, the resource integration strategy is triggered. The approach in [14] resulted in a 20% reduction in SLA violation for cloud applications and a 30% reduction in the resource consumption of the data center.

In [15], it is shown that the load among VMs is correlated. For example, the I/O intensive applications, while deployed on the same PM will result in resource competition and increase the response time. Therefore, the authors proposed an algorithm that integrates the applications with fewer associations to reduce resource competition between the applications. Dhiman *et al.* [16] allocated virtual resources based on memory utilization. However, the approach used in [16] did not consider the impact of the CPU resources on the energy consumption.

In [17]–[19], the authors implemented the virtual resource allocation algorithms by predicting the load of VMs. In [17], the authors used the gray prediction model to predict the future load of a cloud application. They optimized the allocation strategy of the virtual resource according to the predicted load to achieve the load balance and reduce the energy consumption. A comprehensive energy saving resource scheduling algorithm is proposed in [18]. This algorithm is based on load forecasting and convex optimization theory, and guarantees the QoS of cloud applications. The algorithm in [18] integrated the virtual resources by VM migration, and then closed the idle PMs to achieve the energy savings. In [19], the authors forecasted the load of a cluster using Wiener filtering approach. The results showed that the forecasting result is very accurate for the applications with a stable load, and the method does not apply to dynamically changing cloud applications.

In [20], the authors proposed the global power aware best fit decreasing (GPABFD) method. The GPABFD method focused on the initial VM placement, so it is useful only for performing the first step of the resource integration.

In [21], a dynamic integration algorithm of virtual resources based on the ant colony system (ACS) is proposed, and this method is called the ACS-VMC method. This algorithm transformed the virtual resource integration problem into a multi-objective optimization problem. The objective functions of the multi-objective optimization problem included minimizing energy consumption, minimizing the number of VM migrations, and avoiding SLA violations. In addition, according to the PM load, the authors divided the PMs into four sets: P_{normal},P_{over},$\widehat{P}_{over}$,P_{under}. Moreover, the ACS-VMC method consisted of two types of agents: local and global agent. A local agent (LA) resided in a PM to solve the PM status detection sub-problem by observing the current resource utilization of the PM. The global agent (GA) acted as a supervisor to optimize the VM placement. While transferring the VMs, the method used the ant colony algorithm to achieve the global optimal migration.

In order to reduce the energy consumption of a data center, the research focusing on hardware of a data center has also made a great progress. In relation to optimizing the hardware, the dynamic voltage frequency regulation (DVFS) and dynamic power management (DPM) technologies have been widely used in literature. The algorithms proposed by Tang *et al.* [22] and Zhang *et al.* [23] are based on the DVFS technology, in which the frequency and voltage of the server can be dynamically adjusted according to the CPU utilization. In [24], an effective energy-efficient resource allocation algorithm called T-Alloc is proposed. The T-Alloc algorithm aimed at the traditional data center and replaced a PM with a single-core processor with a multi-core processor PM. Further, according to the load of the VM, the T-Alloc algorithm dynamically adjusted the processor count to reduce the energy consumption.

In the single-core CPU period, a lot of research has been carried out to model the power consumption of a CPU. The results show that the power consumption by a PM can be accurately described using a linear relationship between the power consumption and the CPU utilization [25], [26]. However, in recent years, with the development of computer hardware structure, the complexity and diversity of a PM has changed very much. Especially, with the development and popularity of multi-core processor technology, it is difficult and unrealistic to determine a precise model to characterize the relationship between the server energy consumption and its resource utilization [21], [27]. Therefore, in the VRDI method, we use the SPECpower [28] to obtain the real data of the energy consumption of the server. The SPECpower can be combined with the current server performance metrics to accurately measure its real power. In the next section, we present the basic framework of the proposed VRDI method.

III. SYSTEM STRUCTURE

The VRDI method proposed in this paper is based on the live migration of VMs. The live migration can transfer a VM from one PM to another without impacting the normal service. Using the VM live migration, we integrate the virtual resources and close the idle PMs to reduce energy consumption. For clarity of description, we provide the following definitions used in the paper.

Definition 1: Load pattern is used to describe the resource usage of a PM or a VM, and it is defined using the vector $\mathrm{U} = (U^{cpu}, U^{mem})^T$. In this paper, the variables U^{cpu} and U^{mem} represent the CPU and memory utilization, respectively.

Definition 2: The vector $\mathrm{R} = (R^{cpu}, R^{mem})^T$ describes the resource requirement of a VM. The variables R^{cpu} and R^{mem} represent the CPU and memory utilization, respectively. When the resources provided by the cloud service provider meet the resource requirements of the VM, then the QoS requirements of the VM can be guaranteed.

Definition 3: The vector $\mathrm{P} = (P^{cpu}, P^{mem})^T$ describes the maximum resources that a PM can provide. The variables P^{cpu} and P^{mem} represent the maximum CPU and memory resources, respectively. Note that the resource requests for all VMs deployed on a PM cannot exceed the maximum resources that it can provide.

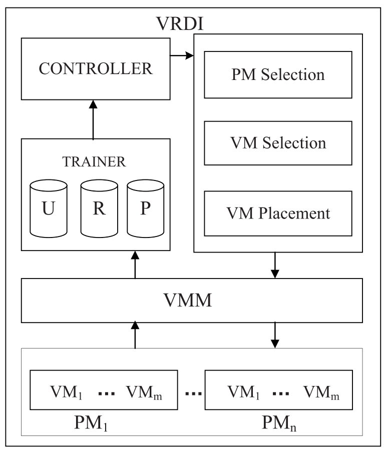


FIGURE 1. The framework of VRDI.

A. THE FRAMEWORK OF VRDI

Fig. 1 shows the framework of VRDI.

As shown in Fig. 1, the virtual machine monitor(VMM) component is used to monitor and collect data from the data center. Currently, most of the data centers use shared storage, therefore, in this work, we do not consider the disk utilization. Moreover, we use CPU utilization and memory utilization to determine the timing of resource integration.

Because different variables have different dimensions, in order to ensure the accuracy of the VRDI, it is necessary to standardize the data. The TRAINER component uses Z-Score to standardize the original data. Assuming $O = \{O_1, O_2, \cdots, O_n\}$ represents the set of data collected by the VMM, where each element $O_i \in O$, with O_i^{cpu} and O_i^{mem} represent the CPU and memory utilization of the PM_i, respectively. The set O can be represented using a matrix as shown below:

$$O = \begin{bmatrix} o_1^{cpu} & o_1^{mem} \\ \vdots & \vdots \\ o_n^{cpu} & o_n^{mem} \end{bmatrix} \tag{1}$$

The average of the jth column of the matrix O can be calculated as:

$$\bar{o}_j = \frac{1}{n}\sum_{i=1}^{n} o_i^j \tag{2}$$

The standard deviation of the jth column of the matrix O can be calculated as:

$$s^j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(o_i^j - \bar{o}_j\right)^2} \tag{3}$$

Using the Z-Score standard transformation, each element $o_i^j \in O$ can be transformed to a dimensionless data in the interval [0,1] using:

$$\left(o_i^j\right)^* = \begin{cases} \dfrac{o_i^j - \bar{o}_j}{s^j}, & s^j \neq 0 \\ 0, & others \end{cases} \tag{4}$$

Using the TRAINER component, the load pattern of a PM and VM is stored in U, the resource requirement of a VM is stored in R, and the maximum resource a PM can provide is stored in P. Based on the values of U, R and P, the CONTROLLER component decides whether to consolidate virtual resources in the data center.

By integrating the virtual resources, the VRDI can reduce the energy consumption of the data center. The VRDI method comprises of the following three algorithms:

1) PM selection algorithm: according to the resource utilization of PMs and the corresponding pre-defined thresholds of the PMs, we generate a set of PMs which should be integrated resources to save energy.
2) VM selection algorithm: according to the resource utilization of the PMs in the first step, we consider two migration scenarios. In the first scenario, all the VMs deployed on the PM should be migrated. In the second scenario, only a part of the VMs should be migrated.
3) VM placement algorithm: according to the results of the VM selection algorithm, for the VMs that need to be migrated (the target VMs), the VM placement algorithm needs to select another PM to place the VMs.

B. THE PM SELECTION ALGORITHM

In order to reduce the energy consumption of the data center, we should collect the performance data of the PMs and VMs in real time to justify the necessity of resource integration. In the first case, the PM resource utilization is low or the PM is already in the idle state. In this scenario, as shown in [10], the energy consumption is close to 70% of the full load. For PM_i, the load pattern is denoted using $U_i = \left(U_i^{cpu}, U_i^{mem}\right)^T$. We denote the lower threshold of resource utilization using $Lower_i = \left(Lower_i^{cpu}, Lower_i^{mem}\right)^T$. If the relationship between the resource utilization and the lower threshold of the PM satisfies,

$$\left(U_i^{cpu} < Lower_i^{cpu}\right) \cap \left(U_i^{mem} < Lower_i^{mem}\right) \tag{5}$$

it is necessary to integrate the virtual resource of the PM. In this scenario, we need to migrate all the VMs deployed on the PM to another PM, and the PM shall be set to the sleeping or shutdown state. We denote the integration performed in this case using ALL.

In the second case, when the PM resource utilization is close to the full load, it may affect the performance of the VM and violate the SLA requirements of the cloud applications. We denote the upper threshold of resource utilization using $Upper_i = \left(Upper_i^{cpu}, Upper_i^{mem}\right)^T$. If the relationship between resource utilization and upper threshold of the PM satisfies,

$$\left(U_i^{cpu} \geqslant Upper_i^{cpu}\right) \cup \left(U_i^{mem} \geqslant Upper_i^{mem}\right) \tag{6}$$

it is necessary to integrate the virtual resource of the PM. As long as one of the U_i^{cpu} or U_i^{mem} is higher than the upper threshold, the VRDI should start. Because one of the CPU or memory is overloaded, it will affect the QoS of the VMs deployed on the PM. In this scenario, we migrate only a part of the VMs to other PMs in the data center to reduce the resource utilization of the PM. We denote the integration performed in this case using PART.

In view of the two different scenarios described above, there are two schemes of integration, namely, all integration (ALL) and partial integration (PART). For the two scenarios described above, the PM selection algorithm is presented in Algorithm 1.

Algorithm 1 The PM Selection Algorithm

Input: pmList // PM Set
Output: ALL,PART // PM Set to be integrated

$ALL \leftarrow \phi$,$PART \leftarrow \phi$;
for each pm in pmList **do**
    calculate load mode U_{pm} of PM;
    if U_{pm} satisfies equation (5) **then**
        $ALL \leftarrow ALL \cup$ pm; //Migrating all VMs on PM
    else if U_{pm} satisfies equation (6) **then**
        $PART \leftarrow PART \cup$ pm; //Migrating part VMs on PM
    else
        continue;
    end if
end for
return ALL $\cup$ PART;

If the number of PMs is n and the number of VMs is m, the algorithm complexity of the PM selection algorithm is $O(n * m)$.

C. THE VM SELECTION ALGORITHM

After selecting the set of PMs to be integrated, we need to select one or more VMs, which should be migrated out from the PM to reduce the load or shutdown the PM to save energy. According to the result obtained by the PM selection algorithm, it is necessary to deal with the ALL and PART cases separately.

For the ALL case, all of the VMs need to be migrated out of the PM, and the PM needs to be set in the sleeping state. For the PART case, it is necessary to select a set of VMs to be migrated. The process of migration of VMs will consumes energy and may impact the QoS of the cloud applications. Therefore, in order to reduce the cost of migrating VMs, in this paper, we propose a minimum migration (MM) policy to minimize number of VMs that need to be migrated. In the MM policy, we calculate the Euclidean distance between the VM's load pattern and PM's load pattern. Note that, the larger the distance, the effect of the PM is dominant. The Euclidean distance d_{ij} of VM's load pattern and PM's load pattern is calculated as:

$$d_{ij} = \frac{1}{\sqrt{\left(u_i^{cpu} - u_j^{cpu}\right)^2}} + \frac{1}{\sqrt{\left(u_i^{mem} - u_j^{mem}\right)^2}} \tag{7}$$

where u_i^{cpu} and u_i^{mem} represent the CPU and Memory utilization of VM_i respectively. Similarly, u_j^{cpu} and u_j^{mem} represent the CPU and memory utilization of PM_j, respectively. According to (7), we can get a set $\mathrm{D} = \{d_{1j}, d_{2j}, \cdots, d_{nj}\}$, which represents the Euclidean distance between PM_j and VMs developed on it.

According to the above description, the migrated VM selection algorithm is presented in Algorithm 2. Using the MM policy, by only transferring the VMs which consume more resources, the VM selection algorithm can effectively reduce the number of VMs to be migrated. If the number of PMs to be integrated is n and the number of VMs is m, the algorithm complexity of the VM selection algorithm is $\mathrm{O}(n * m)$.

Algorithm 2 The VM Selection Algorithm

Input: ALL,PART // PM Set to be integrated
Output: M //VM Set of VMs to be migrated

$M \leftarrow \phi$;
for each pm in ALL **do**
    vmList ← pm.getVMList(); //get VMs developed on pm
    $M \leftarrow M \cup$ vmList; //VMs to be migrated
end for
for each pm in PART **do**
    vmList ← pm.getVMList(); //get VMs developed on pm
    $D \leftarrow \phi$; // Set of Euclidean distance
    for each vm in vmList **do**
        calculating the Euclidean distance d between pm and vm;
        $D \leftarrow D \cup$ d;
    end for
    Sort D and get newVMList;
    for each vm in newVMList **do**
        pm.removeVM(vm); // remove vm
        $M \leftarrow M \cup$ vm;
        calculating the load pattern U_{pm} of pm;
        if U_{pm} satisfies equation (6) **then**
            continue; // pm still overloading, continue remove vm
        else
            break;
        end if
    end for
end for
return M;

D. THE VM PLACEMENT ALGORITHM

The VM placement is a key problem of the virtual resource integration. The VM placement is generally described as a bin packing problem. The bin packing problem is an NP-hard problem, and most of researchers use the global optimization tools to find a solution [29]. As a classical algorithm for solving optimization problems, genetic algorithm has been researched extensively [30], [31]. The genetic algorithm simulates the biological evolution process by selecting, encoding, crossing and mutating to generate the best individual. By calculating the fitness of each individual iteratively, the genetic algorithm eliminates the individual with the minimum fitness. When the maximum number of iterations is reached, the individual with the best fitness is taken as the optimal solution of the problem. For selecting the number of iterations, most of the researchers use the empirical approach. However, this approach has the following problems: on one hand, if the iteration number is low, the algorithm cannot obtain the optimal solution; on the contrary, if the number of iterations is high, the algorithm has low efficiency. Therefore, in this paper, we propose a VM placement algorithm based on improved genetic algorithm, denoted IGAVP. In the IGAVP approach, we terminate the algorithm if the optimal fitness of the offspring individual is no longer increasing. Using this approach, we avoid the subjectivity of the results.

1) ENCODING

Assuming the target VM set is $\mathrm{M} = \{VM_1, VM_2, \cdots, VM_m\}$, PM set is $\mathrm{S} = \{PM_1, PM_2, \cdots, PM_n\}$, the encoding of a chromosome is denoted using an array E. The elements of E represent a mapping of a VM in the set M to a PM from the set S. The size of the array E is m. The initial population has some influence on the efficiency and accuracy of a genetic algorithm. In this paper, we use the first fit algorithm (FFA) to generate the encoding of the initial population. The encoding process is as follows: first, we select VM_i randomly from the set M, then find PM_j from S. If PM_j can satisfy the resource request of VM_i, we continue selecting the next VM from the set M, otherwise, we pick a new PM in P. This process is repeated until the set M is empty. Assuming that the maximum resource provided by the PM is $P = (1, 1)$ and the size of S is 4, for the VM set in Table 1, we can get the chromosome: 123241. Table 2 and Fig. 2 show the encoding scheme and the VM placement scheme, respectively.

TABLE 1. Partial VMs load patterns.

VM	Load pattern
VM_1	$(0.6, 0.5)$
VM_2	$(0.5, 0.3)$
VM_3	$(0.8, 0.3)$
VM_4	$(0.2, 0.5)$
VM_5	$(0.5, 0.2)$
VM_6	$(0.1, 0.3)$

TABLE 2. Encoding scheme.

VM_1	VM_2	VM_3	VM_4	VM_5	VM_6
PM_1	PM_2	PM_3	PM_2	PM_4	PM_1

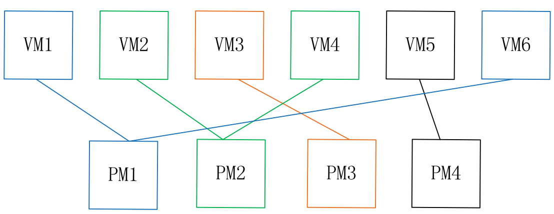


FIGURE 2. VM placement scheme.

2) THE FITNESS FUNCTION

Each chromosome needs to be evaluated so that we can choose an excellent chromosome as a parental collection to cross. In this paper, we build the fitness function based on the resource utilization of the PM. For a chromosome g, its fitness function Fit(g) is given by:

$$Fit\,(g) = \frac{1}{n}\sum_{i=1}^{n}\left(U_i^{cpu} + U_i^{mem}\right) \tag{8}$$

where n is the number of PMs, U_i^{cpu} and U_i^{mem} denote the resource utilization of PM_i. We use $Fit\,(g)$ to calculate the fitness of the chromosomes.

3) SELECTION

For the initial population, we select two chromosomes to generate the offspring. The fitness of offspring depends on the two chromosomes that we selected. Based on the principle of survival of the fittest, we use the roulette wheel to select the individual. The principle of a roulette wheel is that the probability of each individual to be selected is proportional to its fitness function. The probability is calculated as follows:

$$p_i = \frac{Fit\,(i)}{\sum_{j=1}^{m} Fit\,(j)} \tag{9}$$

where m is the size of population. First, we generate a random number r from the interval $\lfloor 0, 1 \rfloor$. If $\sum_{j=0}^{i-1} p_j < r < \sum_{j=0}^{i} p_j$, *where* $i = 1, 2, \cdots, n$ *and* $p_0 = 0$, then the individual i is selected.

4) CROSSOVER

Crossover is the process of combining the genes of the parents to generate the offspring. The purpose of crossover is to generate an offspring who has excellent gene inherited from the parents. In this work, the crossover point is selected randomly. Using FFA, one offspring is generated by inheriting one parent from the beginning to the crossover point, and the remaining part from the crossover point to the end is inherited from the other parent. For example, if the parents are $p_1 = 123241$ and $p_2 = 123341$, when the crossover point ρ is 3, based on FFA, we can get the offspring $c_1 = 123123$ and $c_2 = 132132$. For c_1 and c_1, we calculate their fitness separately, and select the offspring which has the higher fitness.

5) MUTATION

After the crossover, we need to perform the mutation operation. In the mutation operation, two randomly selected points within a chromosome are swapped. For example, by swapping the third and fourth positions of c_1, we can get $c_3 = 121323$.

Using the described procedure, by replacing the original population of the lowest fitness with c_3, we get a new population. Next, we continue the iterative evolution process. When the fitness of the optimal chromosome is no longer increasing after the generation C, the IGAVP process is stopped, and the optimal chromosome corresponds to the solution of the best VM placement. Based on the described ideas, we summarize the specific process of the IGAVP in the Algorithm 3.

Algorithm 3 The VM Placement Algorithm

Input: M,S,N,C
Output: T //The VM placement solution

```
generate the initial population S of N individuals
count ← 0;
calculate Fit(g) of each individual in S;
MAX_FIT ← max (Fit (g)); // get the highest individual
repeat
    select the parents chromosome p1 and p2 based on Roulette wheel;
    get c1 and c2 by crossing p1 and p2;
    get the higher fitness individual c3 from c1 and c2;
    get the new individual T by mutating c3;
    get the new population by replacing the lowest fitness individual of S with T;
    if MAX_FIT ⩾ max (Fit (g))) then
        count++;
    else
        MAX_FIT ← max (Fit (g)));
    end if
until count == C
```

IV. EXPERIMENT AND EVALUATION

In order to verify the effectiveness of the proposed method based on the energy-efficient virtual resource integration, a series of experiments are carried out. In the following, we describe the experimental setup followed by the results of the experiments.

A. EXPERIMENTAL SETUP

In order to perform a large number of repeated experiments to verify the effectiveness of the proposed virtual resource integration method, we used CloudSim [32], a cloud computing simulation toolkit, to simulate the experiment. The CloudSim is the most popular cloud environment simulation framework, and its core component allows its users to monitor and manage the virtual resources and customize the virtual resource allocation strategy. The expanded components can provide energy consumption and statistics of the simulation. Moreover, the toolkit can also simulate the dynamic load of cloud applications.

Using the CloudSim toolkit, we created a data center consisting of 100 PMs. It had two types of PMs, namely, the HP ProLiant ML110 G5 and the IBM X3550. There are 5∼20 VMs which have different load mode on each PM. In order to ensure the accuracy of the results, our experiments were carried out for about six months.

Table 3 summarizes some important parameters used in the IGAVP algorithm.

TABLE 3. IGAVP parameters.

Parameter	Description	Value
S	Population size	100
C	Iteration Counter	20
PS	Policy of Selection	Roulette wheel

As described in [27], the thresholds of PMs are not constant because of the dynamic and unpredictable workloads of cloud applications. Therefore, we used the Local Regression (LR) method in [27] to set the utilization thresholds based on the historical data of the load patterns collected by the VMM. In this work, we performed a series of experiments to estimate the threshold values. Based on our analysis, we set the lower and upper thresholds as (0.30, 0.30) and (0.80, 0.80).

In this work, we define an SLA violation event as a scenario in which a VM cannot get the requested cpu CPU utilization. In case of SLA violation, the cloud providers should pay a penalty to the users.

B. RESULTS OF EXPERIMENTS

In our experiment, by controlling the number of VMs, we compare the VRDI method proposed in this work with the BMH method of [11] and the ACS-VMC method of [21]. By instantiating different number of VMs, and using CloudSim's FFA to complete the initial allocation of VMs, we compared the three methods in terms of four different performance metrics: the total energy consumption of the data center, the number of migrated VMs, the percentage of SLA violations, and the number of shutdown PMs.

First, we compare the performance of the three methods in terms of the total energy consumption of the data center. Under the condition of having the same number of VMs, the VRDI, BMH and ACS-VMC algorithms are compared with the FFA. The results are shown in Fig. 3.

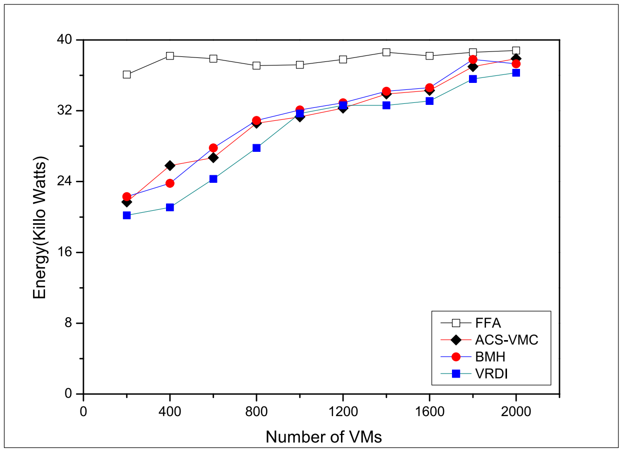


FIGURE 3. Comparison of different algorithms in terms of the energy consumption at the data center.

From Fig. 3, it can be observed that the three methods can have smaller energy consumption at the data center compared with the FFA algorithm. The VRDI method can save about 45% of energy when the resource utilization of PM is less than 50%. The proposed VRDI method integrates the resource utilization of the PM and VM, and considers the problem of virtual resource integration as a multi-objective optimization problem. Hence the performance of the VRDI is better than the BMH and ACS-VMC methods.

Secondly, we compare the three methods in term of the number of migrated VMs. In the process of virtual resource integration, the migration of VM may affect the QoS of the deployed cloud applications. Hence, it is important to reduce the number of VMs to be migrated. Fig. 4 shows the results of experiment for the three methods.

From figure 4, it can be observed that the number of VMs to be migrated in the VRDI method is larger than the BMH and ACS-VMC methods when the number of VMs in data center is less. This is because, when the number of VMs is less, the VRDI approach will migrate all of the VMs developed on the PM and shutdown the PM to save energy. Hence, in the VRDI algorithm, during the VM selection, the ALL case is active. When the number of VMs in data center is high, the PM's resource utilization is higher, and during the VM selection algorithm, the PART case becomes active. Hence, in this case, the VRDI migrates less number of VMs compared with the BMH and ACS-VMC methods as illustrated in Fig. 4. When selecting a set of VMs for migration, the VRDI uses the minimum migration principle to select a VM whose load pattern is identical with that of the PM. Hence, the VRDI approach selects less number of VMs for migration.

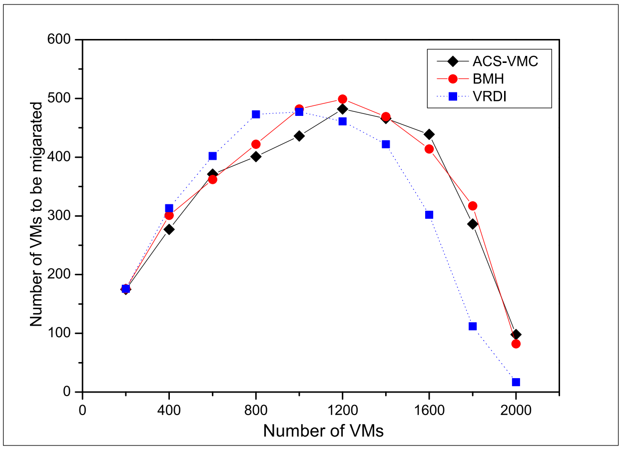


FIGURE 4. Comparison of the number of VMs to be migrated.

Thirdly, we compare the three methods in terms of the percentage of SLA violations. The process of VM migration will increase the utilization of resources and even violate the SLA requirements of the cloud applications. Therefore, it is very important to minimize the time required to perform VM migration in the process of virtual resource integration. Fig. 5 shows the percentage of SLA violation of the VRDI, BMH and ACS-VMC methods.

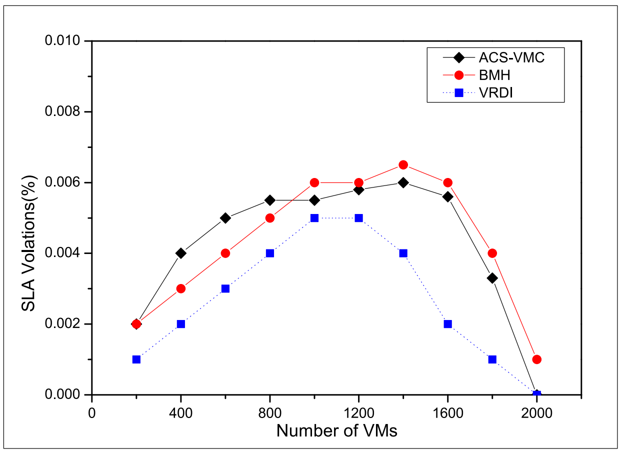


FIGURE 5. Percentage of SLA violation of VMs.

Using the IGAVP algorithm, the VRDI can find an optimal scheme to place target VMs efficiently, hence, it can minimize the time required for VMs migration. The results from Fig. 5 show that the SLA violation of the VRDI method is lower than that of the BMH and ACS-VMC methods.

Finally, in Fig. 6, we show the number of PMs that were closed when using the VRDI, BMH and ACS-VMC methods. As mentioned above, the purpose of resource integration of the data center is to transfer the VMs to close some of the PMs which have lower utilization to improve the energy efficiency of the data center. Therefore, the more the number of closed PMs, the more is the effectiveness of an algorithm. From Fig. 6, we can see that when the resource utilization is less than 50%, the VRDI method closes about 38% of the PMs. Whereas, the BMH and the ACS-VMC methods only close about 20% of the PMs. Hence, the VRDI is more efficient in performing the virtual resource integration. On the other hand, the experimental results also show that the energy consumption of the data center is a very serious issue, especially, in a large scale data center. Hence, it is important to integrate resources and save energy. Therefore, the VRDI method proposed in this paper can greatly contribute to the construction of a green data center.

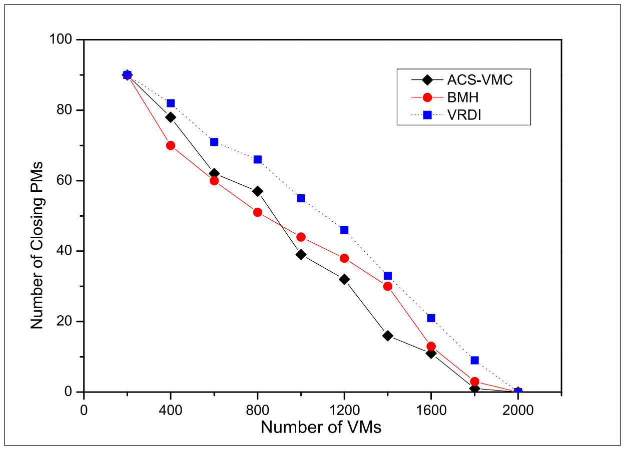


FIGURE 6. Comparison of the number of closed PMs.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed an energy-efficient virtual resource dynamic integration (VRDI) method. The method is based on the live migration technology of VM, which can reduce the energy consumption of a data center by integrating the virtual resources. The proposed VRDI method consists of three parts: (1) based on the resource utilization and the corresponding predefined thresholds of the PMs, we determine the integration timing and the PMs set that need to be integrated; (2) based on the load pattern of the VM, and the Euclidean distance between the VM and a PM, we select a minimal set of VMs which need to be migrated; (3) using the IGAVP, we find an effective VM placement solution to solve the bin-packing problem.

Further, we presented experimental results comparing the proposed VRDI method with the existing solutions. The results showed that the VRDI method has a significant advantage in terms of reducing the energy consumption of a data center. Therefore, the proposed VRDI method can be useful in the construction of a green data center.

In recent years, with the development of the mobile Internet, the energy consumption of network resources in a data center is becoming more and more prominent. In the proposed VRDI method, we did not consider the impact of the network resources on the energy consumption of a data center, and we plan to consider this topic as one of our future works. In addition, we also plan to predict the VMs load pattern in the VM selection algorithm and study its performance.

REFERENCES

[1] M. Chen, H. Zhang, Y.-Y. Su, X. Wang, G. Jiang, and K. Yoshihira, "Effective VM sizing in virtualized data centers," in *Proc. IEEE Int. Symp. Integr. Netw. Manage.*, May 2011, pp. 594–601.

[2] D. Frederico, C. Jose, F. Anderson, and G. Vinicius, "A systematic review on cloud computing," *J. Supercomput.*, vol. 68, no. 3, pp. 1321–1346, 2014.

[3] Y. Bi, H. Zhou, W. Xu, X. Shen, and H. Zhao, "An efficient PMIPv6-based handoff scheme for urban vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 12, pp. 3613–3628, Dec. 2016.

[4] M. A. Vouk, "Cloud computing—Issues, research and implementations," *Int. J. Adv. Res. Comput. Sci. Manage.*, vol. 16, no. 4, pp. 235–246, 2015.

[5] M. N. O. Sadiku, S. M. Musa, and O. D. Momoh, "Cloud computing: Opportunities and challenges," *IEEE Potentials*, vol. 33, no. 1, pp. 34–36, Jan. 2014.

[6] M. Armbrust *et al.*, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[7] S. Sharma and G. Sharma, "A review on secure and energy efficient approaches for green computing," *Int. J. Comput. Appl.*, vol. 138, pp. 25–32, Sep. 2016.

[8] S. Shivam, "Green computing strategies & challenges," in *Proc. Int. Conf. Green Comput. Internet Things (ICGCIoT)*, Oct. 2015, pp. 758–760.

[9] Y. Bi, H. Shan, X. S. Shen, N. Wang, and H. Zhao, "A multi-hop broadcast protocol for emergency message dissemination in urban vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 3, pp. 736–750, Mar. 2016.

[10] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *Computer*, vol. 40, no. 12, pp. 33–37, 2007.

[11] S. Sohrabi, A. Tang, I. Moser, and A. Aleti, "Adaptive virtual machine migration mechanism for energy efficiency," in *Proc. Int. Workshop Green Sustain. Softw. (ACM)*, Sep. 2016, pp. 8–14.

[12] L. Liu *et al.*, "GreenCloud: A new architecture for green data center," in *Proc. 6th Int. Conf. Ind. Session Autonomic Comput. Commun. Ind. Session (ICACINDST)*, New York, NY, USA, 2009, pp. 29–38.

[13] B. Anton, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Generat. Comput. Syst.*, vol. 28, no. 5, pp. 755–768, 2012.

[14] N. Bobroff, A. Kochut, and K. Beaty, "Dynamic placement of virtual machines for managing SLA violations," in *Proc. IEEE Int. Symp. Integr. Netw. Manage.*, Sep. 2007, pp. 119–128.

[15] Q. Zhu, J. Zhu, and G. Agrawal, "Power-aware consolidation of scientific workflows in virtualized environments," in *Proc. ACM/IEEE Int. Conf. High Perform. Comput. Netw. Storage Anal.*, Apr. 2010, pp. 1–12.

[16] G. Dhiman, G. Marchetti, and T. Rosing, "vGreen: A system for energy efficient computing in virtualized environments," in *Proc. ACM/IEEE Int. Symp. Low Power Electron. Design (ACM)*, Sep. 2009, pp. 243–248.

[17] J.-J. Jheng, F.-H. Tseng, H.-C. Chao, and L.-D. Chou, "A novel VM workload prediction using grey forecasting model in cloud data center," in *Proc. Int. Conf. Informat. Netw.*, 2014, pp. 40–45.

[18] D. M. Bui, Y. Yoon, E. N. Huh, S. Jun, and S. Lee, "Energy efficiency for cloud computing system based on predictive optimization," *J. Parallel Distrib. Comput.*, vol. 102, pp. 103–114, Sep. 2017.

[19] M. Dabbagh, B. Hamdaoui, M. Guizani, and A. Rayes, "Energy-efficient resource allocation and provisioning framework for cloud data centers," *IEEE Trans. Netw. Serv. Manage.*, vol. 12, no. 3, pp. 377–391, Apr. 2015.

[20] T. K. Okada, A. D. L. F. Vigliotti, D. M. Batista, and A. G. vel Lejbman, "Consolidation of VMS to improve energy efficiency in cloud computing environments," in *Proc. 33rd Brazilian Symp. Comput. Netw. Distrib. Syst.*, Vitoria, Brazil, May 2015, pp. 150–158.

[21] F. Farahnakian *et al.*, "Using ant colony system to consolidate VMs for green cloud computing," *IEEE Trans. Serv. Comput.*, vol. 8, no. 2, pp. 187–198, Apr. 2015.

[22] Z. Tang, L. Qi, Z. Cheng, K. Li, S. U. Khan, and K. Li, "An energy-efficient task scheduling algorithm in DVFS-enabled cloud environment," *J. Grid Comput.*, vol. 14, no. 1, pp. 55–74, 2016.

[23] Y. Zhang, Y. Wang, and C. Hu, "CloudFreq: Elastic energy-efficient bag-of-tasks scheduling in DVFS-enabled clouds," in *Proc. IEEE Int. Conf. Parallel Distrib. Syst.*, Sep. 2015, pp. 585–592.

[24] D. M. Quan, F. Mezza, D. Sannenli, and R. Giafreda, "T-Alloc: A practical energy efficient resource allocation algorithm for traditional data centers," *Future Generat. Comput. Syst.*, vol. 28, no. 5, pp. 791–800, 2012.

[25] X. Fan, W. D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *Proc. Int. Symp. Comput. Archit. (DBLP)*, 2007, pp. 13–23.

[26] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang, "Power and performance management of virtualized computing environments via lookahead control," in *Proc. Int. Conf. Autonomic Comput.*, Sep. 2008, pp. 3–12.

[27] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurrency Comput. Pract. Exper.*, vol. 24, no. 13, pp. 1397–1420, 2012.

[28] K. D. Lange, "Identifying shades of green: The SPECpower benchmarks," *Computer*, vol. 42, no. 3, pp. 95–97, Mar. 2009.

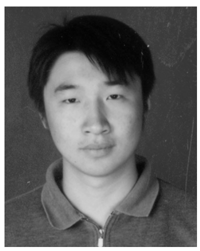
[29] Z. Qi, C. Lu, and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," *J. Internet Services Appl.*, vol. 1, no. 1, pp. 7–18, 2010.

[30] E. G. Coffman, M. R. Garey, and D. S. Johnson, *Approximation Algorithms NP-Hard Problems*. Boston, MA, USA: PWS Publishing Company, 1997.

[31] E. Falkenauer, "A hybrid grouping genetic algorithm for bin packing," *J. Heuristics*, vol. 2, no. 1, pp. 5–30, 1996.

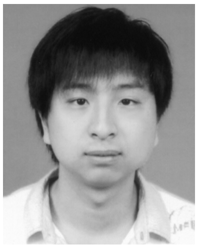
[32] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Softw. Pract. Exper.*, vol. 41, no. 1, pp. 23–50, Sep. 2011.

YINGYOU WEN received the Ph.D. degree in computer science from Northeastern University, Shenyang, China, in 2005. He was a Post-Doctoral Research Fellow with the Neusoft Group from 2005 to 2008. He is currently an Associate Professor with the School of Computer Science and Engineering, Northeastern University. He is also an Associate Director of the Chinese State Key Laboratory of Software Architecture and the Vice President of the Neusoft Group Research. His research interests include cloud computing, network security, and sensor network.

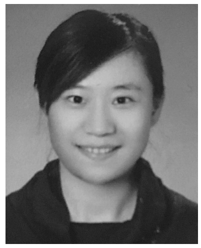


ZHI LI received the B.S. degree in computer science from the College of Information Science and Engineering, Northeastern University, China. He is currently a Graduate Student with the School of Computer Science and Engineering, Northeastern University. His research interests focus on wireless sensor network.

SHUYUAN JIN received the B.Sc. and M.Sc. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 1996 and 1998, respectively, and the Ph.D. degree from The Hong Kong Polytechnic University in 2006. She is currently a Full Professor with Sun Yat-sen University, China. Her main research interests include network security, especially intrusion detection and responses, Internet technologies, machine learning, and pattern recognition techniques and applications.



CHUAN LIN received the B.S. degree in computer science and technology from Liaoning University, Shenyang, China, in 2011, and the M.S. degree in computer science and technology from Northeastern University, Shenyang, in 2013, where he is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering. His research interests include network performance analysis, software defined networking, and network modeling.



ZHENG LIU received the B.Sc., M.Sc., and Ph.D. degrees in computer science from Northeastern University, Shenyang, China, in 2002, 2006, and 2010, respectively. She joined the College of Information Science and Engineering, Northeastern University, as a Lecturer in 2005, where she is currently a Lecturer with the School of Computer Science and Engineering. Her current research interests focus on software network and complex system.