Computing (2016) 98:303–317
DOI 10.1007/s00607-015-0467-4

Energy-efficient migration and consolidation algorithm of virtual machines in data centers for cloud computing

Hongjian Li[1] · **Guofeng Zhu**[1] ·
Chengyuan Cui[1] · **Hong Tang**[1] · **Yusheng Dou**[2] ·
Chen He[3]

Received: 5 November 2014 / Accepted: 16 July 2015 / Published online: 26 July 2015
© Springer-Verlag Wien 2015

Abstract In this paper, we developed a dynamic energy-efficient virtual machine (VM) migration and consolidation algorithm based on a multi-resource energy-efficient model. It can minimize energy consumption with Quality of Service guarantee. In our algorithm, we designed a method of double threshold with multi-resource utilization to trigger the migration of VMs. The Modified Particle Swarm Optimization method is introduced into the consolidation of VMs to avoid falling into local optima which is a common defect in traditional heuristic algorithms. Comparing with the popular traditional heuristic algorithm Modified Best Fit Decrease, our algorithm reduced the number of active physical nodes and the amount of VMs migrations. It shows better energy efficiency in data center for cloud computing.

✉ Hongjian Li
lihj@cqupt.edu.cn

Guofeng Zhu
464799127@qq.com

Chengyuan Cui
844279104@qq.com

Hong Tang
tanghong@cqupt.edu.cn

Yusheng Dou
Yusheng.dou@nicholls.edu

Chen He
che@cse.unl.edu

[1] Department of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

[2] Department of Physical Sciences, Nicholls State University, PO Box 2022, Thibodaux, LA 70310, USA

[3] Department of Computer Science & Engineering, University of Nebraska-Lincoln, Lincoln, NE 68588, USA

Keywords Virtual machines · Energy efficiency · MPSO · Multi-resource · Migration

Mathematics Subject Classification 68Q85 · 68U20

1 Introduction

To satisfy the requirements of Quality of Service (QoS), load balancing, auto-scaling, and energy conservation, the migration and consolidation of virtual machines (VMs) mechanism for dynamic allocation of virtual resources is a popular topic in cloud computing [1–4]. In particular, the following VMs migration problems are mainly investigated: when to migrate, which VMs to migrate, and where to migrate the VMs selected for migration [5,6]. The energy efficiency of the system will be improved by dynamically assigning virtual resources to various cloud tasks and adjusting the mapping between VMs and physical nodes in real time. Migration of VMs can reduce the energy consumption of the data center. At the same time, the number of VMs migration and Service Level Agreements (SLAs) violation can be reduced.

Nowadays VMs migration and consolidation algorithms are mostly based on energy consumption model with single system resource constraint, i.e. CPU. These algorithms may not consider the impact of other resources [7–15]. Some researchers [16,17] believe that the energy consumption of whole server varies approximately linearly with the CPU utilization. However, it still consumes more than 70 % of its peak energy, even if a server is completely idle [18]. Beloglazov et al. [7,8,10] proposed an energy-efficient resource allocation algorithm based on energy consumption model of CPU. Double threshold method based on CPU utilization was designed to trigger VMs migration and the reallocation of VMs selected for migration was also based on CPU utilization. This algorithm effectively reduced energy consumption while satisfying the Quality of Service (QoS). Nevertheless, it did not consider resources other than CPU. In some applications, all kinds of IT resources, such as CPU, memory, bandwidth, and disk, have influence on energy consumption and system performance. Srikantaiah et al. [19,20] studied the impacts of system resources utilization on energy consumption and performance. They proposed a multi-resource energy-efficient model depending on both the CPU and disk utilization. Comparing with previous energy-efficient models depending only on CPU utilization, this model is closer to real situation. In addition, the energy-efficient model unified the Quality of Service (QoS) and energy conservation, which simplified multi-objective optimization problem into a single objective optimization problem. However, they did not focus on the VMs migrations in data center for cloud computing. Pallavi [21] and Rajyashree [22] separately proposed live migration of VMs according to the current power consumption by considering multiple resources. Traditional heuristic algorithms were used in their studies. But double threshold method was only based on CPU utilization in the study of Pallavi [21]. Rajyashree considered CPU, RAM and bandwidth with equal weight for calculating the upper threshold [22], they did not investigate the optimal operating points between energy consumption and resource utilization.

Reallocation of VMs plays an important role in dynamic VMs scheduling. VMs reallocation algorithm decides where to allocate the VMs selected for migration. Heuristic algorithm based on greedy algorithm [7,8,10,14,23,24] is widely applied in VMs reallocation and consolidation. Beloglazov et al. [7,8,10] proposed modified best fit decrease (MBFD) algorithm for VMs reallocation. In this algorithm, all VMs were sorted in decreasing order of their current CPU utilization, and each VM was allocated to a host that provided the least increase of the power consumption caused by the allocation. This heuristic algorithm which was based on the traditional greedy algorithm could optimize the allocation of VMs, but it was easy to fall into the local optima and hard to achieve global optimal with a single point of search strategy. Then some researchers [25,26] proposed an adaptive VMs allocation genetic algorithm to avoid falling into local optima. However, the process of genetic algorithm was too complex and did not fit the reallocation in the real cloud environment. Recently, Particle Swarm Optimization (PSO) algorithm [27] which is similar with genetic algorithm has got widely attention as it has fewer parameters, faster convergence and other advantages. Kenney et al. [28] proposed the discrete particle swarm algorithm based on PSO to solve combinatorial optimization problem. Xu et al. [29] proposed an improved binary PSO to improve the discrete particle swarm algorithm [28], and applied it into knapsack problem. PSO algorithm has been developed and applied widely. It is a feasible choice to optimize the VMs reallocation in cloud computing.

In this paper, a dynamic energy-efficient migration and consolidation algorithm is developed. It is based on the multi-resource energy-efficient model and uses a multi-resource double threshold method replaces the CPU double threshold method [7,8] to trigger the migration of VMs. Modified PSO algorithm (MPSO) is introduced into VMs reallocation to improve energy efficiency in data center. Two different migration and consolidation algorithms MBFD and MPSO are investigated. Comparing with MBFD, MPSO algorithm achieves better energy efficiency, reduces the number of VMs migration and active physical nodes effectively, and makes the utilization of system resources more balanced.

2 Energy-efficient model with multi-resource

Different kinds of system resources, such as CPU, memory, bandwidth, and disk, may play a role in energy consumption in cloud data center. To study the impact of energy efficiency with multiple resources, Srikantaiah [19,20] measured performance and energy while varying both CPU and disk utilizations. In their study, an application was combined with workloads of varying CPU and disk utilizations, ranging from 0.1 to 0.9 in each resource. Energy consumption and performance of applications were measured at different utilization in each resource. The experiment results show that the system has the minimum energy consumption and effectively ensure mission performance when CPU utilization and disk utilization of the physical node are 0.7 and 0.5, respectively. The author regarded 0.7 and 0.5 as the optimal operating points of CPU utilization and disk utilization.

In our energy-efficient model, Euclidean distance between CPU utilization and disk utilization is used as an energy efficiency factor to evaluate merits of energy efficiency after allocating VMs in each physical node based on the study of Srikantaiah [19,20]. The energy consumption of system is evaluated by the total Euclidean distance of all active physical nodes at the same time. The smaller Euclidean distance is, the better energy efficiency is. The resource utilization of one physical node becomes zero if this physical node is turned down. Energy efficiency factor of active physical node is defined as Euclidean distance δ_h given by (1).

$$\delta_h = \sqrt{\sum_{j=1}^{d} (u_j - ubest_j)^2} \tag{1}$$

where j denotes resources such as CPU, disk, memory and bandwidth and so on, u_j and $ubest_j$ represent current utilization and best utilization of resource j for energy efficiency in each physical node, respectively. Each resource has its optimal point of utilization, such as 0.7 for CPU and 0.5 for disk usage, according to the experimental result from the study of Srikantaiah [19,20].

The system energy efficiency factor δ^t at time t is defined as the total Euclidean distance of all active physical nodes at this time and given by (2).

$$\delta^t = \sum \delta_h \tag{2}$$

In this paper, the energy-efficient VMs migration and consolidation algorithm with multiple resources is based on this energy-efficient model. To evaluate the system energy efficiency within a given period of time T, resource utilizations will be monitored and the energy efficiency factor of the system at different time t will be calculated periodically in the system. The energy efficiency of the system within T is given by (3).

$$\delta = \sum_{t=0}^{T} \delta^t \tag{3}$$

3 Algorithm design

In this section, a VMs migration algorithm is designed to satisfy Service Level Agreements (SLAs) and to further optimize the energy efficiency of system. There are three steps to be determined in this algorithm: when to migrate, which VM to migrate and where to migrate the VMs selected for migration. A multi-resource double threshold method is used to trigger the VMs migration. Then the migration policy is applied to select which VMs to migrate. Finally, modified Particle Swarm Optimization method is designed and introduced into VMs reallocation algorithm to reduce energy consumption of the whole system.

3.1 Multi-resource double threshold method

The resource utilization of physical nodes is monitorred in each monitoring period. According to the study of Srikantaiah [19,20], two IT resources, CPU and disk, are considered in our method. Three statuses of physical nodes are defined: overloaded, normal and underloaded. VMs migration will be triggered, once the status of one physical node converts to overloaded or underloaded. The status of physical nodes is described as (4) and (5):

$$\left(\sum_j r_j^{CPU} * x_h^j > high\_threshold * c_h^{CPU}\right) \wedge \left(\sum_j r_j^{DISK} * x_h^j > high\_threshold * c_h^{DISK}\right) \tag{4}$$

$$\left(\sum_j r_j^{CPU} * x_h^j \neq 0\right) \wedge \left(\sum_j r_j^{DISK} * x_h^j \neq 0\right) \wedge (\delta_h < low\_threshold) \tag{5}$$

High_threshold and low_threshold denote overloaded and underloaded threshold, respectively, which triggers VMs migration. r_j^{CPU} and r_j^{DISK} are resources required of CPU and disk of VM j. c_h^{CPU} and c_h^{DISK} are capacities of CPU and disk of the physical node h. x_h^j indicates whether the VM j is assigned to physical node h. If VM j is assigned to the physical node h, $x_h^j = 1$, otherwise $x_h^j = 0$. If the resources requirement for VMs cannot be satisfied, the physical node is considered to be overloaded, as seen in formula (4). If the energy efficiency factor of the active physical node is below the underloaded threshold, the node is considered to be underloaded, as seen in formula (5).

3.2 Selecting migration VMs

Once the physical node is overloaded, find which resource requirement cannot be satisfied. Then VMs are sorted in descending order according to this resource. A VM is selected from the descending VMs queue and added into migration queue in sequence until the resource requirement of the rest VMs on this physical node can be met. In contrast, once a physical node is underloaded, all VMs on this physical node are added into the migration queue. The algorithm periodically monitors workload and decides which nodes should be turned on or off to minimize power consumption by the system. The algorithm always consolidates VMs to the minimum number of active physical nodes without causing a SLAs violation.

3.3 Consolidation algorithm based on PSO

Particle Swarm Optimization algorithm [27] (PSO) is proposed by Kenney and Eberhart in 1995 to solve optimization problems in continuous space. Compared with other group evolutionary algorithms such as Genetic Algorithm (GA), PSO has many advantages: fewer parameters, faster convergence and easier implementation. Recently, PSO has gained more and more attentions and been wildly applied in many fields. In this section, Modified Particle Swarm Optimizer (MPSO) is introduced into VMs reallocation algorithm. Assume that there are m VMs and n physical nodes in cloud data center, and w VMs are in the migration VMs queue. When $w = 1$, MBFD algorithm is applied to VMs consolidation. When $w > 1$, MPSO migration algorithm is used to optimize energy efficiency of the system. The main steps of MPSO are as follows.

3.3.1 Population initialization

N VMs requests sequences are randomly generated. For each request sequence, the VM is assigned on the first physical node meeting the resources requirement by First Fit algorithm. Then N distribution plans are obtained in this step. Each distribution plan is sorted in descending order, and N particles are generated to constitute the initial population. The position vector of particles is defined as $X_l^r = (x_{l1}^r, \ldots, x_{lj}^r, \ldots, x_{lw}^r)$, where l is the lth possible solution, $l \leq N$, the serial number of VM $j \leq w$, and r is the iteration number. For example, $X_l^r = (x_{l2}^r,\ x_{l4}^r,\ x_{l5}^r)\ = (1,\ 2,\ 1)$ means that VM 2, 4, 5 are respectively allocated to node 1, 2, 1. While the particles are updating, the position vector of particles X_l^r transfers to a (0,1)-matrix S as seen in formula (6).

$$S = \begin{bmatrix} s_{11}^r & \cdots & s_{1h}^r & \cdots & s_{1n}^r \\ \vdots & & \vdots & & \vdots \\ s_{j1}^r & \cdots & s_{jh}^r & \cdots & s_{jn}^r \\ \vdots & & \vdots & & \vdots \\ s_{w1}^r & \cdots & s_{wh}^r & \cdots & s_{wn}^r \end{bmatrix} \tag{6}$$

If VM j is allocated to node h at iteration r, $s_{jh}^r = 1$; otherwise, $s_{jh}^r = 0$. A VM can be allocated to only one physical node, so a constrain exists for s_{jh}^r, as seen in formula (7).

$$\sum_{h=1}^{n} s_{jh}^r = 1, \quad \forall j \in \{1, 2, \ldots, m\} \tag{7}$$

3.3.2 The definition of the fitness function

In order to achieve minimal energy consumption after VMs being reallocated, the fitness function is designed as seen in formula (8).

$$f(\delta^t) = \sum \delta_h \tag{8}$$

where f(δ^t) is the sum of the energy efficiency factor of all active physical nodes at the time $t$after reallocation of VMs in migration queue. As described in formula (2), δ_h is the energy efficiency factor of physical node h. The objective of dynamic VMs consolidation is to minimize the energy consumption caused by all physical nodes, so we need to minimize the fitness function f(δ^t).

3.3.3 Updating the positions of particles

The fitness of the particle population is calculated by the fitness function. Then the personal best of each particle and the global best of all particles are obtained.

In our MPSO algorithm, $Pbest_l^r = (pbest_{li}^r, \ldots, pbest_{lj}^r, \ldots, pbest_{lo}^r)$is the personal best of particle l at iteration r. Each particle keeps track of its coordinates in the solution space which are associated with the best solution that has achieved so far by that particle. $Gbest^r = (gbest_i^r, \ldots, gbest_j^r, \ldots, gbest_o^r)$ is the best value obtained so far by all particles.

Suppose the probability that VM$j$is allocated to node h is 0.5 at iteration $r + 1$, $P(s_{jh}^{r+1} = 1) = P(s_{jh}^{r+1} = 0) = 0.5$. p_p and p_g denote, respectively, the trust degree of personal and global best which means the probability of finding the optimal solution. According to the Bayesian formula, the probabilities of personal and global best should be higher than average probability, $0.5 < p_p < 1, 0.5 < p_g < 1$. To avoid falling into local optimization, let $p_g < p_p$, then $0.5 < p_g < p_p < 1$. In this study, the parameters are set as $p_p = 0.8$, $p_g = 0.7$, according to the study of Xu [29].

3.3.4 Checking the new positions of particles

As seen in formula (9)–(11), new positions of particles should meet these constraints,

$$x_h^j \in \{0, 1\} \tag{9}$$

$$\sum_h x_h^j = 1, \quad \forall j \tag{10}$$

x_h^j indicates whether the VM j is assigned to physical node h. If VM j is assigned to the physical node h, $x_h^j = 1$, otherwise $x_h^j = 0$. Formula (9) indicates that each VM can be assigned on only one physical node.

$$\left(\sum_j r_j^{CPU} * x_h^j \leq c_h^{CPU}\right) \wedge \left(\sum_j r_j^{DISK} * x_h^j \leq c_h^{DISK}\right) \tag{11}$$

In formula (11), when a couple of VMs are assigned to physical node h, resources of the VMs cannot exceed resources capacities of physical node$h$.

If all the constraints are satisfied, the particles are updated to the new positions; otherwise the value of the original particle remains.

a. If $\sum_{h=1}^{n} s_{jh}^r > 1$, VM j is allocated to multiple physical nodes. Then set $\forall h,\ sjh^r{}_{= 0}$ and sort physical nodes in ascending order. The VM j is allocated to the first physical node which satisfies formula (11). If all recorded physical nodes do not satisfy formula (11), the result is unreasonable and the particles will not be updated.

b. If $\sum_{h=1}^{n} s_{jh}^{r} = 1$, VM j is allocated to a physical node, then check whether constraint (11) is satisfied. If it is satisfied, the particles are updated to new positions; otherwise the particles are not updated.

c. If $\sum_{h=1}^{n} s_{jh}^{t} = 0$, VM j is not assigned to any physical node. The new positions do not satisfy the conditions, and the particles are not updated.

This VMs migration and consolidation algorithm is based on multi-resource energy-efficient model and Modified Particle Swarm Optimization algorithm (MPSO). The energy efficiency can be improved, while QoS requirements are constrained. SLAs violation and energy efficiency are both considered in this study. The VMs which most likely cause SLAs violation in physical node will be selected for migration. MPSO algorithm is proposed in this part. First Fit algorithm is used to generate particles for reducing active physical nodes, and a Modified Particle Swarm Optimization algorithm [29] is introduced and designed to update the positions of particles. Additionally, Euclidean distance is defined to evaluate energy efficiency after migration.

In our MPSO algorithm, we assume that there are m VMs and n physical nodes in cloud data center, and w VMs are in the migration VMs queue. N is the scale of the particle swarm and Q is the maximum number of iterations. N and Q are both constants. There are several parts of calculations in VMs allocation as follows: (1) Particles initialization: generate N particles at random. All VMs are allocated to the nodes by the First Fit algorithm, which requires N*m operations. N operations are required to initialize the particle's pbest and gbest position, separately; (2) Updating velocities and positions of particles: mn^2 operations; (3) Checking if all the constraints are satisfied: mn^2 operations; (4) Other iterations require $(\mathrm{Q} - 1) * (N + 4mn^2)$. So the latency of algorithms execution as a function of number of nodes is $g(n, m) = (m + Q + 1)N + 4(Q - 1)mn^2$. The computational complexity of MPSO allocation is $o(mn^2)$. The analysis of operations of VMs migration is similar to VMs allocation. There are w VMs in the migration VMs queue and the computational complexity of MPSO migration is $o(wn^2)$.

4 Simulation and results analysis

4.1 Experiment setup

Since it is very complicated to conduct repeatable large-scale experiments on a real cloud environment, we perform simulations in CloudSim to ensure the repeatability and reproducibility of experiments.

Cloudsim [30] is a framework for modeling and simulation of cloud computing infrastructures and services. It has been widely used to evaluate algorithms, applications, and policies before actual development of cloud products. We have simulated a data center that comprises 250 heterogeneous physical nodes, half of which are HP ProLiant ML110 G4 servers, and the others consist of HP ProLiant ML110 G5 servers. The frequency of the servers' CPUs is mapped onto MIPS ratings. The storages of two servers are 640 and 1000 GB. There are four VM types in the data center. The configuration (MIPS, storage) of each VM type is High-CPU Medium Instance (2500 MIPS, 350 GB), Extra Large Instance (2000 MIPS, 1690 GB), Small Instance (1000 MIPS, 850 GB) and Micro Instance (500 MIPS, 160 GB).

Discrete-event simulation has been chosen to evaluate the performance of the proposed algorithms and the number of VMs ranges from 50 to 500 by 50. Runs were performed using a MBFD [7,8] algorithms and our MPSO method to compare. The total running time is 100 s and the monitoring period is 5 s. The size of the particle swarm is 20 and the maximum iteration number is 30. Take the average values of 10 replicates as the final results.

4.2 Thresholds analysis

The influence of high and low thresholds is evaluated on energy efficiency in data center which consists of 50 VMs. In migration algorithm, the upper threshold is set to avoid the SLAs violations. Each physical node periodically executes an overload detection strategy to trigger migration. A physical node is overload, when the resource utilization reaches the upper threshold. The upper threshold should be adjusted, depending on the specific system requirements, to avoid performance degradation and SLAs violations. Pallavi set 30–70 % for ensuring percentage of SLA violation of 1 % [21] and Rajyashree set dynamic upper thresholds to avoid overloaded situation [22]. The value of upper threshold can be set to any value in range from 0 to 1. In this algorithm, the upper threshold is set to 1 which is the same with the study of MBFD [7,8].

A physical node is considered to be underloaded, when the resource utilization is under the lower threshold. The lower threshold significantly affects the energy efficiency and the amount of migrations. In order to find an optimal lower threshold for this simulation, MBFD and MPSO algorithms are respectively tested while the value of lower threshold is varied from 0.0 to 0.9 by 0.1. The system energy consumption and the number of migrations are measured within 100 s as seen in Figs. 1 and 2.

As shown in Figs. 1 and 2, the system energy consumption and the number of migrations vary with the value of lower threshold. The tendencies of the system energy consumption obtained from the two algorithms are similar, as well as the tendencies of the number of migrations. As seen in Fig. 1, when the value of lower threshold increases from 0 to 0.5, there is no obvious change of the system energy consumption. However, it increases fast from 0.5 to 0.9. As seen in Fig. 2, when the value of lower threshold increases from 0 to 0.2, the value of the number of migrations keeps at a high level and remains stable. It drops gradually from 0.2 to 0.9. Considering both energy consumption and the number of migrations, the lower threshold is set to 0.5 in the following simulations.

According to formula (1), $0 \leq \delta_h \leq 0.86$. When low threshold is small, δ_h should be small and u_j is close to $ubest_j$. In contrast, when low threshold is large, δ_h could be large but smaller than 0.86. In Fig. 2, when low threshold is larger than the maximum value of δ_h0.86, the CPU and disk utilization could be in range from 0 to 1, and the number of migrations could decrease to 0. However, when low threshold is small, such as 0, δ_h is small, and the CPU and disk utilization should be close to 0.7 and 0.5 respectively. Then the number of migrations will increase.

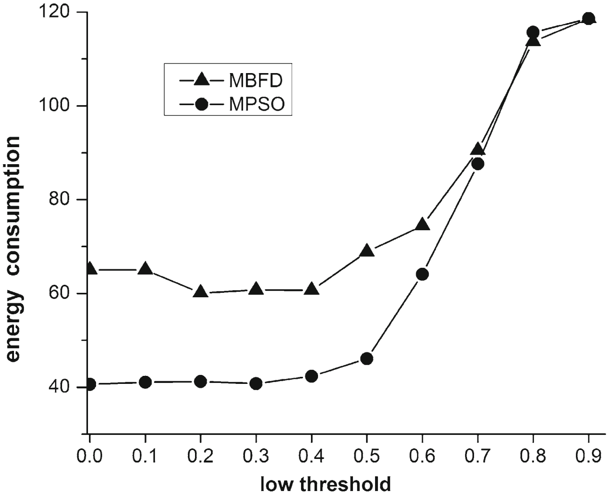

Fig. 1 Comparison of system energy consumption with different low threshold

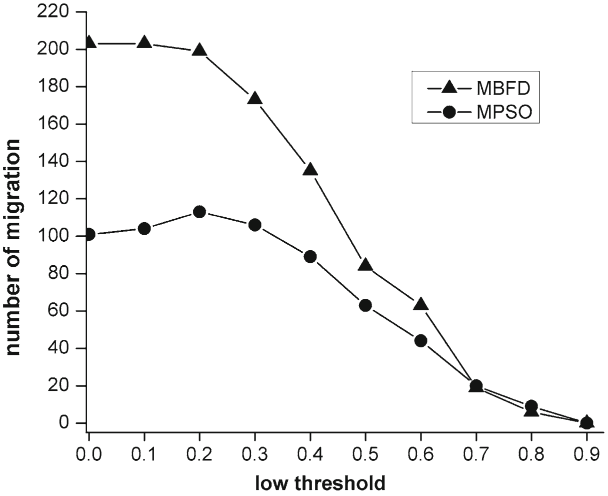

Fig. 2 Comparison of the number of migrations with different low threshold

4.3 Evaluation of energy consumption

The system energy consumption of a cloud computing is evaluated by the total Euclidean distance of all physical nodes within running time. To achieve the optimal system energy efficiency, the total Euclidean distance should be minimized. The comparison of the system energy consumption using MBFD and MPSO algorithms is shown in Fig. 3.

As shown in Fig. 3, the total energy consumption increases with the number of VMs using both MBFD and MPSO algorithm. However, the total energy consumption is always lower for MPSO especially from 50 to 400. MBFD algorithm just considers one reallocation sequence of VMs selected for migration, and only considers current status

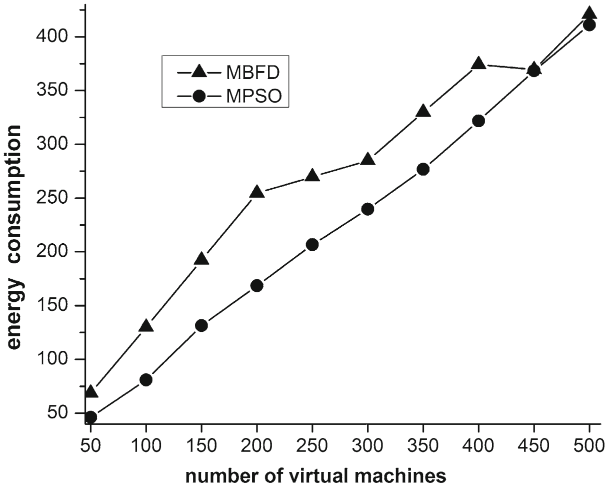

Fig. 3 Comparison of the system energy consumption

of the system when reallocate a VM. However, MPSO algorithm achieves the best result of reallocation by comparing the results of different allocations and evolution, so the result of MPSO algorithm is closer to the optimization of energy efficiency.

4.4 Comparison of the number of active physical nodes

The system maintains a set of active physical nodes selected to serve requests for each VM. Cloud providers optimize resource usage and reduce energy consumption by switching idle nodes to sleep mode. Therefore, the fewer the number of active physical nodes is, the better the energy efficiency is. Fig. 4 shows the comparison of the number of active physical nodes using MBFD and MPSO algorithms.

As shown in Fig. 4, the number of active physical nodes increases with the number of VMs using both MBFD and MPSO algorithm. However, the number of active physical nodes is fewer for MPSO with same VMs. In the Modified Best Fit Decreasing (MBFD) algorithms [19,20], it sorts all VMs in decreasing order of their current CPU utilizations, and allocates each VM to a physical node that provides the least increase of power consumption due to this allocation. MBFD does not consider directly reducing the number of active physical nodes in entire running time (The total running time is 100 s and the monitoring period is 5 s). However, In MPSO algorithm, First Fit algorithm is applied during the initialization of particles, and the VMs are assigned to the first active physical node satisfying the QoS constraints. Thus, this MPSO algorithm effectively reduces the number of active physical nodes.

4.5 VMs migration and load balance

VMs migrate at the beginning of each monitor period. It has negative impacts on performance of the running tasks. We assume that each VM migration costs the same

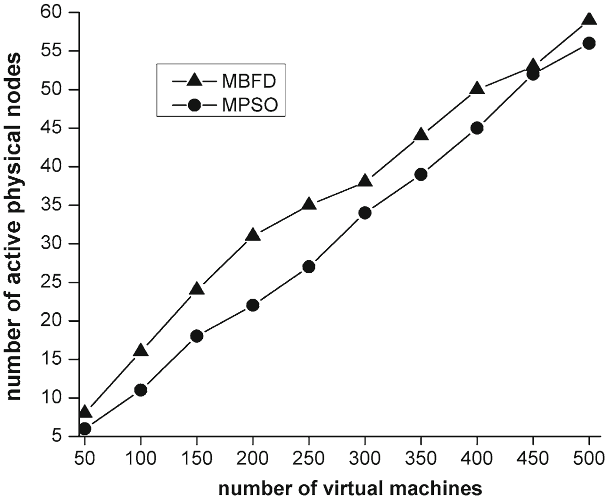

Fig. 4 Comparison of the number of active physical nodes

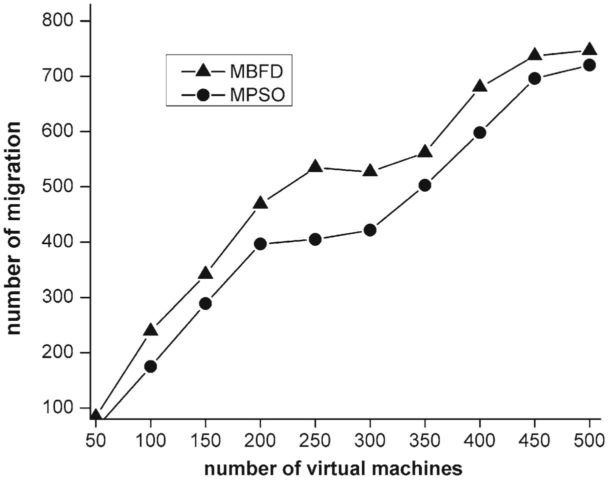

Fig. 5 Comparison of the number of VM migrations

amount of resources, so it is crucial to minimize the number of VM migrations ensuring the quality of service and energy conservation. Figure 5 shows the comparison of number of VM migrations using by MBFD and MPSO.

In Fig. 5, the number of VMs migration increases more slowly for MPSO than MBFD. The number of VMs migration of MPSO is always fewer than that of MBFD with the same number of VMs. Thus, our method has the less cost of VMs migration than MBFD which is important for energy efficiency in cloud computing. The reason is from the resources utilizations become more reasonable by the fitness function for MPSO. Then while preventing to cause a SLAs violation, MPSO is less likely to trigger the migration of VMs in a monitoring period.

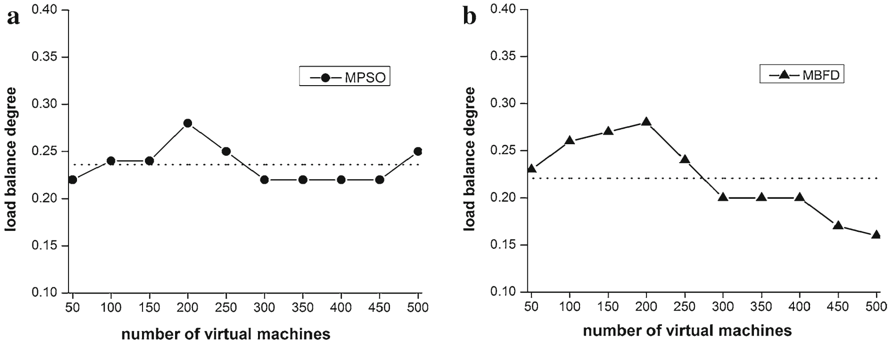

Fig. 6 Comparison of load balance degree

As seen in Fig. 6, the dot line indicates the average load balance degree. When the number of virtual machines changes from 50 to 500, the load balance degree of MPSO algorithm keeps at a stable level. It also shows that MPSO is better at maintaining a high degree of load balance, and has better adaptability than MBFD in the simulations.

Euclidean distance between CPU utilization and disk utilization is used to evaluate merits of energy efficiency in our energy-efficient model. According to the study of Srikantaiah, the optimal balance between resource utilization and energy consumption resides around 0.7 on the CPU and 0.5 storage usage [19,20]. As seen in formula (1), we set important parameters global best as 0.7 and 0.5 for CPU and storage, respectively. Srikantaiah plots the energy consumption with varying combined CPU and disk utilizations. It is a “U”-shaped curve [19,20]. It means that two points in the figure with the same distance from 0.7 and 0.5 have the same energy consumptions. For instance, the point (0.8, 0.4) has the same distance from optimum point (0.7, 0.5) with the point (0.8, 0.6) approximately. Energy consumption is a little more sensitive to variations in CPU utilization than variations in disk utilization [19,20]. Thus, increasing the same distance in CPU utilization may have a bigger effect than that in disk utilization. For example, the decline from 0.7 to 0.3 in CPU utilization may have a bigger effect than that from 0.5 to 0.3 in disk utilization.

Additionally, in this study we compare our method with the recent study from Beloglazov [7,8]. From the observed simulation results, several conclusions can be made: (1) MPSO algorithm has some advantages in VMs allocations. MPSO has fewer parameters and faster convergence than GA algorithm. As described in Sect. 3.3, the computational complexity of MPSO algorithm is $o(mn^2)$. Besides, this algorithm will not fall into a local optimum which is a common defect in traditional heuristic algorithms. (2) In this simulation, we analyze four important factors: the total Euclidean distance, the number of active physical nodes, the number of migrations and load balance degree. These factors are interrelated each other in some degree. It is obviously shown that the performance of our MPSO algorithm is better than MBFD [7,8]. (3) A multi-resource double threshold method is used to replace the CPU double threshold method to trigger the migration of VMs. Inspired by the influence of high and low thresholds on energy efficiency in different applications, the values for high and low thresholds of different IT resources are evaluated and analyzed in the simulation. If the thresholds change, the number of migrations will change, and the Quality of Service (QoS) will also be impacted in some degree in cloud data center for cloud computing.

5 Conclusions

The main contributions of this study include: (1) a multi-resource energy-efficient migration and consolidation algorithm of VMs under dynamic load is designed and implemented; (2) a method of multi-resource utilization double threshold is applied to trigger migration of VMs, and Particle Swarm Optimization is introduced into the VMs reallocation algorithm; (3) analysis is provided between the traditional heuristic algorithms MBFD and MPSO algorithm.

We use CloudSim to simulation our virtual machines migrations and consolidation algorithm with multi-resource under dynamic loads. The results indicate that our algorithm improved energy efficiency, reduced the amount of physical nodes in use and virtual machines migrations, and balanced the system resources. This paper proposed a solution considering not only the impact of multi-resource on energy efficiency but also more accurate scheduling of virtual machines which has practical reference value for virtual machines scheduling in real environment.

Acknowledgments This paper was supported by the Scientific and Technological Research Program of Chongqing Municipal Education Commission (Grant No. KJ130514), Research Program of Chongqing Science and Technology Commission (Grant No. cstc2015jcyjA0420) and Natural Science Foundation of Chongqing University of Posts and Telecommunications (Grant No. A2012-31). We also thanks Linpeng He and Chao Dong from Nicholls State University and Chen He from University of Nebraska-Lincoln for suggestion on this study.

References

1. Nguyen QH, Nam T, Nguyen T (2013) Epobf: energy efficient allocation of virtual machines in high performance computing cloud. J Sci Technol 51(4B):173–182
2. Atefeh K, Saurabh K, Rajkumar B (2013) Energy and carbon-efficient placement of virtual machines in distributed cloud data centers. In: Euro-par 2013 parallel processing. Lecture notes in computer science, vol 8097. Springer, Berlin, pp 317–328
3. Nakku K, Jungwook C, Euiseong S (2014) Energy-credit scheduler: an energy-aware virtual machine scheduler for cloud systems. Future Gener Comput Syt 32:128–137
4. Sheikh H, Tan H, Ahmad I, Ranka S, Bv P (2012) Energy- and performance-aware scheduling of tasks on parallel and distributed systems. ACM J Emerg Technol Comput 8(4):32(1–37)
5. Zhang W, Song Y, Ruan L (2012) Resource management in internet-oriented data centers. J Softw 23(2):179–199
6. Haikun L, Hai J, Cheng X, Xiao L (2013) Performance and energy modeling for live migration of virtual machines. Cluster Comput 16(2):249–264
7. Beloglazov A, Abawajyb J, Buyya R (2012) Energy-aware resource allocation heuristics for efficient management of data center for Cloud computing. Future Gener Comput Syt 28(5):755–768
8. Beloglazov A, Buyya R (2012) Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud computings. Concurr Comput Pract Exp 24(13):1397–1420
9. Bobroff N, Kochut A, Beaty K (2007) Dynamic allocatement of virtual machines for managing sla violations. In: 10th IFIP/IEEE international symposium on integrated network management. IEEE Computer Society Press, Munich, pp 119–128

10. Beloglazov A, Buyya R (2013) Managing overloaded hosts for dynamic consolidation of virtual machines in cloud computing under quality of service constraints. IEEE Trans Parallel Distrib 24(7):1366–1379
11. Wei L, Huang T, Chen J (2013) Workload prediction-based algorithm for consolidation of virtual machines. J Electr Inf Technol 35(6):1271–1276
12. Kusic D, Kephart JO, Hanson J (2009) Power and performance management of virtualized computing environments via look ahead control. Cluster Comput 12(1):1–15
13. Ajiro Y, Tanaka A (2007) Improving packing algorithms for server consolidation. In: International computer measurement group conference. CMG Press, San Diego, pp 399–406
14. Gupta R, Bose SK, Sundarrajan S (2008) A two stage heuristic algorithm for solving server consolidation problem with item-item and bin-item incompatibility constraints. In: Proceedings of the IEEE international conference on service computing. IEEE Computer Society Press, Hawaii, pp 39–46
15. Gergo L, Florian N, Hermann M (2013) Performance tradeoffs of energy-aware virtual machine consolidation. Cluster Comput 16(13):481–496
16. Gandhi A, Harchol-Balter M, Das R et al (2009) Optimal power allocation in sever farms. In: Proceedings of the 11th international joint conference on measurement and modeling of computer systems. ACM, New York, pp 157–168
17. Chen G, He W, Liu J et al (2008) Energy-aware server provisioning and load dispatching for connection-intensive internet services. In: Proceedings of symposium on networked systems design and implementation (NSDI). USENIX Association Berkeley, pp 337–350
18. Fan X, Weber WD, Barroso LA (2007) Power provisioning for a warehouse-sized computer. In: Proceedings of the 34th annual international symposium on computer architecture (ISCA 2007). ACM, New York, pp 13–23
19. Srikantaiah S, Kansal A, Zhao F (2008) Energy aware consolidation for cloud computing. In: Proceedings of the 2008 conference on power aware computing and systems. USENIX Association Berkeley, p 10
20. Srikantaiah S, Kansal A, Zhao F (2010) Energy aware consolidation for cloud computing. In: Proceedings of the IEEE conference on power aware computing and systems. IEEE Computer Society Press, San Diego, pp 577–578
21. Gupta Pallavi, Vishwakarma Lokendra, Patel Awadheshwari (2014) Power—aware virtual machine consolidation considering multiple resources with live migration. Int J Comput Appl 103(17):24–30
22. Rajyashree VR (2015) Double threshold based load balancing approach by using VM migration for the cloud computing environment. Int J Eng Comput Sci 4(1):9966–9970
23. Verma A, Ahuja P, Neogi A (2008) pMapper: power and migration cost aware application allocatement in virtualized systems. In: Middleware 08 proceedings of the 9th ACM/IFIP/USENIX international conference on middleware. Springer, Berlin, pp 243–264
24. Widmer T, Premmand M, Karaenke P (2013) Energy-aware service allocation for cloud computing. In: Proceedings of the international conference on wirtschaftsinformatik. Leipzig, pp 1147–1161
25. Nguyen Q, Pham D, Nguyen H, Nguyen H, Nam T (2013) A genetic algorithm for power-aware virtual machine allocation in private cloud. In: ICT-EurAsia'13 proceedings of the 2013 international conference on information and communication technology. Springer, Berlin, pp 183–191
26. Agrawal S, Bose SK, Sundarrajan S (2009) Grouping genetic algorithm for solving the server consolidation with conflicts. In: Proceedings of the ACM/SIGEVO summit genetic and evolutionary computation. ACM Press, New York, pp 1–8
27. Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science. IEEE, pp 39–43
28. Kennedy J, Eberhart R C (1997) A discrete binary version of the particle swarm algorithm. In: IEEE international conference on systems, man, and cybernetics, vol 5. IEEE, Orlando, pp 4104–4108
29. Xu Y, Xiao R et al (2007) An improved binary particle swarm optimizer. Pattern Recogn Artif Intell 20(6):788–793
30. Calheiro R, Ranjan R, Beloglazov A, Rose C, Buyya R (2011) CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Softw Pract Exp 41(1):23–50