Front. Comput. Sci., 2015, 9(2): 322–330
DOI 10.1007/s11704-015-4286-8

RESEARCH ARTICLE

Virtual machine selection and placement for dynamic consolidation in Cloud computing environment

Xiong FU ((✉)), Chen ZHOU

School of Computer Science and Technology, Nanjing University of Posts & Telecommunications, Nanjing 210003, China

Abstract Dynamic consolidation of virtual machines (VMs) in a data center is an effective way to reduce the energy consumption and improve physical resource utilization. Determining which VMs should be migrated from an overloaded host directly influences the VM migration time and increases energy consumption for the whole data center, and can cause the service level of agreement (SLA), delivered by providers and users, to be violated. So when designing a VM selection policy, we not only consider CPU utilization, but also define a variable that represents the degree of resource satisfaction to select the VMs. In addition, we propose a novel VM placement policy that prefers placing a migratable VM on a host that has the minimum correlation coefficient. The bigger correlation coefficient a host has, the greater the influence will be on VMs located on that host after the migration. Using CloudSim, we run simulations whose results let draw us to conclude that the policies we propose in this paper perform better than existing policies in terms of energy consumption, VM migration time, and SLA violation percentage.

Keywords cloud computing, dynamic consolidation, VM migration, energy consumption

1 Introduction

The number of cloud data centers that can support large scale Internet services is increasing quickly and the operation cost grows due to the rising energy consumption of these data centers [1]. Based on a report from Microsoft [2], the energy consumed by physical resources, e.g., CPU, Memory, Storage, in a data center accounts for 45% of the operating costs have multiplied in the past five years. So any cloud providers who want to survive fierce market competition must commit themselves to reducing energy consumption to cut down the high operation cost [3].

At present, virtualization is widely used in most physical machines in cloud data centers. Resources requested by users are be packed as virtual machines (VMs) and then placed in different hosts based on specific criteria, such as meeting the Service Level Agreement (SLA) requirements between cloud providers and users, improving the utilization of resources, reducing the number of VM migrations and so on.

Each VM requires a certain amount of resources, such as CPU, memory, storage and bandwidth, to support application performance, and multiple VMs can run on the same physical machine (PM) using virtualization technology: this is helpful to improve resource utilization and reduce energy consumption. Moreover, virtualization can also help cloud providers orderly deploy resources on-demand, which provides an effective solution to the flexible resource management and low energy consumption. However, unnecessary VM migrations introduce extra management cost, e.g., virtual machine reconfiguration, online VM migration, and creation and destruction of VMs, which causes extra energy consumption. Therefore, we attempt to reduce the number of VM migrations to reduce energy consumption.

One method to reduce energy consumption is dynamic consolidation of VMs in which VMs are periodically reallocated to minimize the number of active hosts that use live

Received June 24, 2014; accepted November 6, 2014

E-mail: fux@njupt.edu.cn

migration. Nevertheless, application performance should also be considered when placing these VMs. That is to say, if we keep all VMs on a single server, the server's performance will be degraded because of its limited physical resources. In that case, the first condition for VM migration is that if the resource utilization exceeds a certain value, VMs on the PM cannot meet the SLA between customers and providers. Therefore, we set an upper threshold of CPU utilization to avoid overloaded hosts and maintain the SLA agreement.

The second method is the turning off PMs with low utilization rate. As reported in Google data centers [4] the average utilization of the whole data center is only 30%, which encourages us to set a low threshold. When a host's resource utilization is lower than the threshold, all the VMs on that PM are migrated and the now unused host is turned off, resulting in fewer active hosts of which each one is highly utilized. Both of these optimizations are considered in our work.

The process of VM dynamic consolidation refers to the setup of a CPU utilization threshold, the selection of VMs, and the VM placement. Because VM placement is an NP hard problem and the workload is unstable and unpredictable, it makes dynamic VM consolidation even more complicated, so we divide the problem into four subproblems: 1) detection of overloaded hosts; 2) finding underloaded hosts; 3) criteria selection for migratable VMs and 4) selection of suitable target hosts to place these VMs.

We first introduce the energy consumption model, VM migration cost model, and the definition of SLA. Based on these models, we present an improved virtual machine selection policy called MP to reduce the SLA violation rate that maintains a low power consumption. In addition, we have designed a virtual machine placement policy based on the correlation coefficient that represents the relationship between the migrating VM and each host. For a host, a greater correlation coefficient indicates a greater performance degradation for other VMs on this host due to the migration. So the host with the minimum correlation coefficient will be the optimal one to place the VM.

In Section 2, related work is discussed. Section 3 presents the three models used in the following parts, including the energy consumption model, VM migration cost model, and the negotiated SLA. After that, the VM selection policy and the VM placement policy are explained in Section 4 and Section 5, respectively. Experimental results and analysis are shown in Section 6. Finally, Section 7 presents conclusions and future work.

2 Related work

With the rapid growth of cloud computing, cloud providers now are paying more attention to the cost and efficiency of data centers. To attract users, cloud providers must provide high quality services at the lowest cost, which means they should reduce their energy consumption of physical machines as much as possible and continue to meet SLAs at the same time. Therefore many researchers have begun to study energy-efficient policies.

Dong et al. [3] described a VM allocation policy based on limited physical resources (such as CPU and Memory), its purpose was to reduce the number of active hosts. The VM allocation policy was abstracted as a combination of the Bin-Packing problem and quadratic assignment problem (QAP), which are both classic NP-Hard combinatorial optimization problems [1].

Nathuji et al. [5] implemented an energy management architecture aimed at virtualized data centers. Their system is divided into two parts: local resource management and global resource management, which performs VM consolidation. The VM consolidation problem is regarded as a sequential optimization problem in [6], and addressed using limited lookhead control (LLC). Contrary to our approach, the proposed algorithms do not handle SLA violations: SLAs are strictly required by users.

Verma et al. [7] use a heuristic bin packing algorithm to solve the problem of dynamic VM placement. However, the algorithm cannot meet the requirements of SLAs because of instability and unpredictable workloads, and is likely to result in SLA violations.

Srikantaiah et al. [8] present a modified bin packing problem to model the VM consolidation problem, considered from the aspect of CPU and disk optimization. Experimental results showed that the model can effectively make a trade-off between energy consumption and performance. However, this method is based on a specific application and is not suitable for a general virtual environment.

Authors in [9] studied dynamic VM consolidation, and set a static upper threshold of 85% for CPU utilization. They introduced a heuristic method to determine whether a host was overloaded. The static threshold of 85% was proposed for the first time in [10], based on their workload study. In their recent work [11], the authors introduce a dynamic CPU utilization threshold.

Beloglazov et al. [11] proposed a heuristic method based on energy-aware resource allocation and consolidated VMs

in [12]. First, they set a fixed upper threshold of CPU utilization for hosts, and then over a constant period they check each host's utilization. If it exceeds the threshold, the host is marked as overloaded. Then VMs are selected to migrate from the overloaded hosts. However, it is not suitable to use a fixed threshold in a virtual environment, because it cannot well reflect the complexity and instability of workloads. Therefore, in their later work [13], the fixed threshold was replaced by a variable one. The VM placement policy called modified best fit decreasing (MBFD) [11] only allocates a VM to a host that has the least increase of energy consumption after the allocation, while our VM placement will select a physical machine that has the least correlation coefficient with the migrated VM to avoid influencing other VMs due to the allocation.

Four VM allocation policies are depicted in [13] to decide which host is suitable for placing VMs, and three VM selection polices are proposed to select the VMs from the oversubscribed host. Experimental results show that VM selection and placement policies can effectively save energy. But in the process of VM migration, these policies have little effect on getting lower SLA violation rate. To solve these problems, they implemented a new VM allocation and selection policy that takes into account SLA violation rate. The authors of [14] designed a novel VM selection strategy that selects a VM whose utilization has the maximum positive correlation coefficient with the total VMs on the host. However, once a VM has been migrated, the most and worst impact will be on the other VMs due to the maximum correlation between them.

In addition, the [15] and [16] study VM migration. They mainly solve how to choose a migratable VM and find the right target host to place the migrated VM. The goal of the method is to improve the utilization of resources and guarantee the performance of the application at the same time.

In this article, we dynamically set the parameter in the overloaded host testing policy. We propose a VM selection policy based on the degree of performance satisfaction, and describe a VM reallocation policy based on minimum correlation coefficient. In our experiments, we will estimate different policies from three aspects: energy consumption, VM migration times, and SLA violation. The comparison of the results highlights the advantages of our proposed policies.

3 Metric definition

3.1 Energy consumption model

The work in [17] and [18] shows that power consumption by physical machines can be accurately described by a linear relationship of CPU utilization. They also point out that a free physical machine uses about 70% of its energy consumption when it is fully utilized. Therefore, we can define power consumption as a CPU utilization function. The function is showed in Eq. (1):

$$P(u) = k \cdot P_{\max} + (1 - k) \cdot P_{\max} \cdot u. \tag{1}$$

$P_{\max}$ is the maximum power of a host in the running state; k is the percentage of power consumed by an idle physical machine; The CPU utilization is denoted by u. Because the utilization of a CPU changes over time, we define it as a function $u(t)$ of time. Therefore, total energy consumption can be obtained by Eq. (2):

$$E = \int_t P(u(t))dt. \tag{2}$$

According to this function, the energy consumption of a physical machine is determined by the CPU utilization. Thus, to reduce the whole energy consumption, we take the CPU utilization into consideration in the following VM selection and allocation policies.

3.2 Cost of VM live migration

Online migration of virtual machines allows transferring VMs between hosts without suspension in a short down time. However, online migration has a bad influence on the performance of applications. According to [19], the VM migration interferes with VMs on both the migration source and destination. Thus, the number of migrations should be reduced. The survey finds that a reduction in performance and down time depends on the behavior of applications, for example: how many memory pages are updated during execution time. In order to avoid performance degradation, we use the virtual machine migration cost model proposed in [20] to help us choose migratable VMs. As the authors say in [20], a single VM Migration can cause performance degradation and can be estimated by an extra 10% of CPU utilization, and this implies that each migration may cause SLA violations. So we should reduce the number of VM migrations, select the virtual machine using the least memory, and try to improve the available network bandwidth. Thus in our experiment, we define the performance degradation of VM j in Eqs. (3) and (4):

$$U_{d_j} = 0.1 \int_{t_0}^{t_0+T_{m_j}} u_j(t)dt, \tag{3}$$

$$T_{m_j} = \frac{M_j}{B_j}, \tag{4}$$

where $U_{d_j}$is the total performance degradation of VM j; t_0 refers to the time when the migration starts; T_{m_i} is the time spent to complete the migration; CPU utilization at time t can be denoted as $u_j(t)$; M_j is the amount of memory used by VM j, and B_j is the available network bandwidth.

3.3 SLA violation metric

In a cloud computing environment, there are many users competing for resources. Each cloud service provider should ensure the satisfaction of application demands, namely meeting the requirement of the user's quality of service (QoS), which is usually defined in the form of an SLA. According to different application conditions, an SLA can be defined in different forms, such as the minimum throughput or maximum response time. Therefore, we should define an SLA metric that has been referred to in [11] that is independent of the loads in Infrastructure as a Service (IaaS) platforms. In our experiment, SLA violations will be measured in two aspects: i) the percentage of time when the host experiences CPU utilization of 100%, SLA violation time per active host (SLATAH); and ii) performance degradation due to migrations (PDM). SLATAH and PDM can be calculated by Eqs. (5) and (6):

$$\text{SLATAH} = \frac{1}{N}\sum_{i=1}^{N}\frac{T_{s_i}}{T_{a_i}}, \tag{5}$$

$$\text{PDM} = \frac{1}{Q}\sum_{j=1}^{Q}\frac{C_{d_j}}{C_{r_j}}. \tag{6}$$

The number of hosts and VMs are denoted as N and Q in a data center respectively. T_{s_i} is the time when the host's utilization reaches 100% which will lead to an SLA violation. T_{a_i} is the time during which the host i is in active state. C_{d_j} is an estimate of the performance degradation caused by VM migrations. C_{r_j} is the total CPU utilization requested by VM j. In our experiment, we set C_{d_j} to 10% of the CPU utilization during the total migration time of VM j. Here we introduce the SLAV (SLA Violation) metric, which is an integration of the host's workload and the influence of VM migrations. Its computation formula is in Eq. (7):

$$\text{SLAV} = \text{SLATAH} * \text{PDM}. \tag{7}$$

4 VM selection Policy

4.1 Host overloading detection

In this part, we focus on solving the first subproblem of VM consolidation. The subject of this problem is to determine whether a host is overloaded or not and when to migrate VMs from the host. As mentioned above, many methods have been proposed to choose the moment to migrate the VMs in order to prevent a potential SLA violation. One of the most widely used methods is to set upper and lower utilization thresholds for hosts and keep the total utilization by all the VMs between these thresholds. If the CPU utilization exceeds the thresholds, it will invoke the VM selection and placement policies, such as median absolute deviation (MAD), interquartile range (IQR), local regression (LR) and robust local regression (IQR). You can learn more about these polices in [11].

4.2 VM selection

Since more than one virtual machine runs on a host, we need to consider how to choose a migratable VM when a host's CPU utilization exceeds the upper threshold. If a random selection policy is adopted in our VM consolidation, some VMs that are operating efficiently will be moved while those that use many resources and provide poor efficiency still run normally. This will not only increase the energy consumption of the data center, but also reduce the utilization of resources. Therefore, a VM selection policy is needed in the dynamic VM consolidation.

At present, there are four types of VM selection policy: maximum correlation (MC), minimum migration time (MMT), minimum utilization (MU), and random selection (RS). MC migrates a VM that has the maximum correlation coefficient compared to the other VMs on the same host. The MMT migrates those VMs that will take the least time to move. MU selects a VM with the lowest utilization. And, RS migrates VMs randomly without any rules.

We propose a novel virtual machine selection policy which is different from the above policies. First, compare host's utilization deviation *dev* over the upper threshold, and compare this with the utilization of VMs on the host and a corresponding strategy is selected on the basis of different comparison results. The policy will make the host's utilization closer to the upper threshold after migration, which will reduce number of migrations needed. At the same time, the aspect of resource satisfaction is taken into consideration and the VM with the lowest satisfaction will get priority to be migrated. We call this policy meets performance (MP).

Step 1 Initialize the host list $HostList = \{H_1, H_2, \ldots, H_i, \ldots, H_n\}$ and define the CPU utilization of each host as $hUtil$. There are m VMs placed on each host and they are represented as $vmList = \{vm_1, vm_2, \ldots, vm_j, \ldots, vm_m\}$

Step 2 Traverse the $hostList$ and see whether each host's

hUtil is above the upper threshold $THRESH\_UP$. If a host's $hUtil > THRESH\_UP$, then go to Step 3, otherwise go to Step 8;

Step 3 Sort the VMs on the overloaded host H_i in descending order of their current CPU utilization and get the VM list of this host as $vmList = \{vm_1, vm_2, \ldots, vm_j, \ldots, vm_m\}$. The CPU utilization of VM j is $util_{vm_j}$ and this host's dev can be obtained by the equation $dev = hUtil - THRESH\_UP$, where dev is the part of CPU utilization that exceeds the upper threshold.

Step 4 Select the first VM in the $vmList$ vm_j, and then we denote the result of $util_{vm_j} - dev$ as t, namely $t = util_{vm_j} - dev$.

Step 5 If $t \geqslant 0$, the VM is selected to be migrated and pushed into the queue $ToMigrateList$ and then end the policy. This step will make the CPU utilization lower but much closer the upper threshold.

Step 6 If $t < 0$, calculate the degree of resource satisfaction sla for each VM in $vmList$ using Eq. (8):

$$sla = (Util_{\text{req}} - Util_{\text{alloc}})/Util_{\text{req}}, \tag{8}$$

where $Util_{\text{req}}$ and $Util_{\text{alloc}}$ represent the VM's respective requested and allocated CPU in MIPS.

Step 7 Sort the VMs in an ascending order of sla value. Select the first VM and push it into the queue $ToMigrateList$. Update the host's utilization and repeat Steps 2–7;

Step 8 If the host's utilization is smaller than the lower threshold $THRESH\_LW$, i.e., $hUtil \leqslant THRESH\_LW$, then all VMs on the host will be selected to be pushed into the queue $ToMigrateList$ and the source host will be turned off after completing all the migrations.

5 VM placement policy

The VMs that need to be migrated are acquired by the implementation of the VM selection policy in Section 4, and then we need a policy to select in which host to place the VMs. One method used in [11] is the power aware best fit decreasing (PABFD). It allocates each VM to a host that provides the least increase of power consumption due to this allocation. In our work, we propose a new policy called the minimum correlation coefficient (MCC). The correlation coefficient is used to represent the degree of association between a chosen VM and the target host. The greater the correlation coefficient, the greater the influence on the performance of the other VMs when the chosen VM is migrated to the target host. A VM will be migrated to a host with the minimum correlation coefficient to avoid performance degradation on other VMs.

Assuming that the hosts in a data center can be denoted as the set U_{hs}, and the target host can be obtained when both of the following two conditions are satisfied: 1) a VM can be migrated to a host only when the remaining physical resources of that host can satisfy the VM's request; 2) a VM will be migrated to a host with the minimum correlation coefficient. Finally, a host that meets the two conditions above will be chosen to place the VM. The policy is implemented by the following steps:

Step 1 Initialize the set $H = \{H_1, H_2, \ldots, H_n\} \subseteq U_{hs}$ where H refers to the set of hosts that satisfy the first condition

Step 2 Select a host H_i from H, and assume that it contains m VMs. The CPU utilization of the m VMs are collected during p time slices and values are stored in the matrix

$$util_i[m][p] = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1p} \\ u_{21} & u_{22} & \cdots & u_{2p} \\ \vdots & \vdots & u_{jk} & \vdots \\ u_{m1} & u_{m2} & \cdots & u_{mp} \end{bmatrix},$$

where u_{jk} refers to the CPU utilization of VM j on host H_i during time slice k.

Step 3 Let the array $Util_i[k]$ denote the CPU utilization of host H_i at time slice k ($k = 1, 2, \ldots, p$) and then it can be obtained by the following formula:

$$Util_i[k] = \sum_{j=1}^{m} util_i[j][k]. \tag{9}$$

Step 4 Calculate the correlation coefficient between the VM and each host in set H according to Eq. (1):

$$\rho_{\text{i}} = \frac{E[(u - \frac{1}{p}\sum_{k=1}^{p} u_k)(U - \frac{1}{p}\sum_{k=1}^{p} Util_i[k])]}{\sqrt{V(u)}\sqrt{V(U)}}. \tag{10}$$

The sign u and U refer to the current CPU utilization of the chosen VM j and the host H_i, respectively. $\frac{1}{p}\sum_{k=1}^{p} u_k$ and $\frac{1}{p}\sum_{k=1}^{p} Util_i[k]$ refer to the respective average CPU utilization of the VM j and the host H_i during the past p time slices. The variance of the CPU utilization of VM and the host H_i are represented as $V(u)$ and $V(U)$ accordingly: they are computed by Eqs. (11) and (12):

$$V(u) = E[(u - \frac{1}{p}\sum_{k=1}^{p} u_k)^2], \tag{11}$$

$$V(U) = E[(U - \frac{1}{p}\sum_{k=1}^{p} Util_i[k])^2]. \tag{12}$$

Step 5 Compute all the correlation coefficients of the hosts in the set H and acquire the set of squares of the correlation coefficients $\rho = \{\rho_1^2, \rho_2^2, \ldots, \rho_n^2\}$, then select the host that has

the minimum squared correlation coefficient and migrate the VM to it.

Step 6 Allocate the VMs from the queue *ToMigrateList* according to Steps 1–5. The *ToMigrateList* is obtained by the VM selection policy mentioned in Section 4.

6 Performance evaluation

6.1 Experimental setup

Carrying out experiments on a real world system would be prohibitively expensive and complicated. Also it is difficult to carry out repeatable experiments in complicated system conditions and user deployments to evaluate the performance of cloud provisioning policies. Therefore, we use CloudSim [12, 21] for modeling and simulation of cloud computing environments. CloudSim is an extensible simulator developed by Melbourne University whose goal is to enable modeling and simulation of cloud computing systems. It can simulate virtualized resources and cloud entities like data centers, virtual machines, and physical hosts. And, we can implement different resource allocation policies and evaluate policy performance.

We use CloudSim-3.0 and simulate a data center that comprises 800 physical hosts, half of which are HP ProLiant ML110 G4 servers (Intel Xeon 3040, 2cores×1 860 MHz, 4 GB), and the other half consists of HP ProLiant ML110 G5 servers (Intel Xeon 3075, 2cores×2 660 MHz, 4 GB). There are 500 VMs on the data center and they are divided into four types: High-CPU Medium Instance (2 500 MIPS, 0.85 GB); Extra Large Instance (2 000 MIPS, 3.75 GB); Small Instance (1 000 MIPS, 1.7 GB) and Micro Instance (500 MIPS, 613 MB). At the beginning, different amounts of resources are requested by different types of VMs, and can be changed in real time according to the VM workload traces: this creates an opportunity for dynamic VM consolidation.

In order to make the results more realistic, we use the CPU utilization traces collected from more than a thousand VMs operating in more than five hundred locations from around the world and collect the data every five minutes. In our experiments, a randomly generated set of VMs and CPU utilization traces is allocated to the host.

6.2 Performance metrics

There are many metrics to measure the efficiency and superiority of various algorithms. One metric is the energy consumption consumed by the data center which can be calculated according to the energy model discussed in Section 3.1. In this model, the parameter $P_{\max}$ is equal to 250 w since a host consumes 250 w when its CPU utilization is 100% according to [11]. And the value of coefficient k is set to 0.7 [22]. Another metric is the SLA violation percentage, which is defined as the ratio of SLA violation time to the whole running time. SLAV (SLA violation), SLATAH (SLAV time per active host), and PDM (performance degradation due to migration), [11], are introduced to evaluate the SLA performance which is negotiated by the users and cloud providers. The third metric is the total number of VM migrations in the data center.

6.3 Simulation results and analysis

We compare our work with the algorithms IQR, MAD, LR and LRR, and VM selection policies MC, MNT, MMT, and RS [11] (we briefly described in Section 4.1) Comparing our work and these four algorithms using the four different selection policies gives a total of 20 combinations. In addition, the parameters in the four host overloading detection are set according to Beloglazov et al. [11]. The simulation results and analysis are displayed from the following four aspects:

1) In our first group of experiments, we compare the different VM selection policies with the default VM allocation policy PABFD, mentioned in Section 5.1, which allocates each VM to a host that has the least increase of energy consumption after this allocation. These experiment results highlight the superiority of the MP VM selection policy. For each host overload detection algorithm, we compare the performance of the five VM selection policies in three aspects: the energy consumption, the number of VM migrations, and SLA violation percentage. The results are displayed in Figs. 1–3 and the observations are summarized as follows: i) there is no remarkable difference between the IQR and MAD algorithms, nor between the two local regression algorithms (LRR and LR), the two local regression based algorithms exhibit less energy consumption, VM migration times and SLA violation

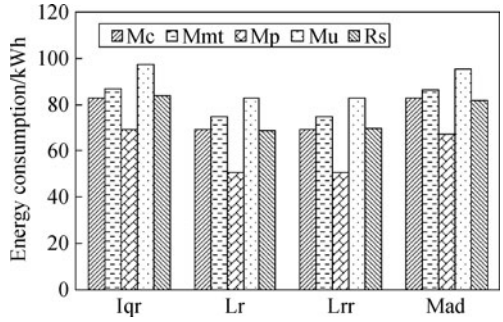

Fig. 1 Energy consumption PABFD

percentage compared with the other algorithms like IQR and MAD; ii) it is obviously that the MP algorithm consumes less energy consumption, VM migration times than the other VM selection policies no matter which host overloading detection algorithm is adopted; iii) although combinations of IQR-MP and MAD-MP generate more SLA violations, the other two combinations of LR-MP and RLR-MP have the fewest SLA violation (Fig.3).

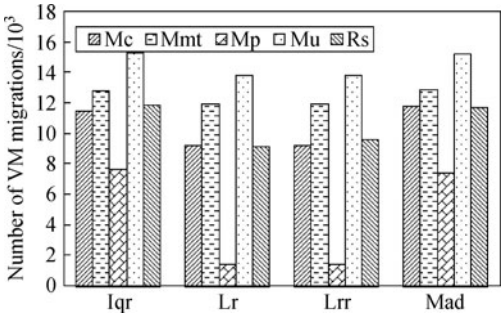

Fig. 2 The number of VM migrations PABFD

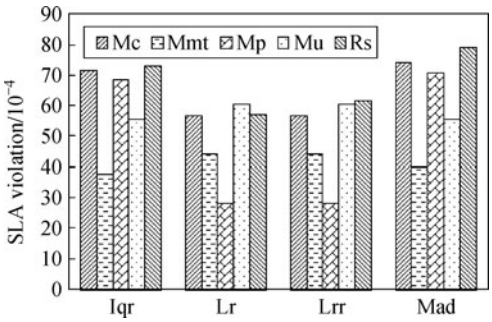

Fig. 3 SLA violations

2) Unlike the first group experiments, the VM placement policy deployed in this group is MCC. The data for the three metrics discussed in Section 3 are shown in Figs. 4–6. The observations are summarized as follows: i) MP exhibits the least energy consumption; ii) VM migration times are greatly reduced, almost half that of other VM selection algorithms, showing the superiority of our MP algorithm; iii) MP exhibits fewer SLA violations than the other VM selection policies.

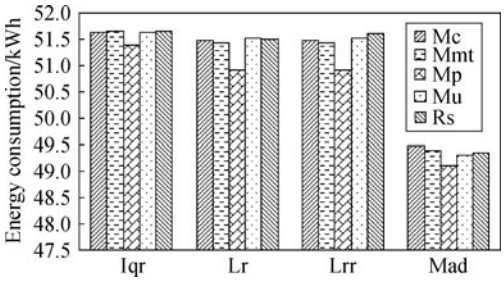

Fig. 4 Energy consumption MCC

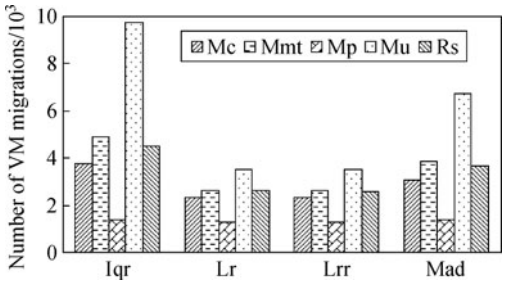

Fig. 5 The number of VM migrations MCC

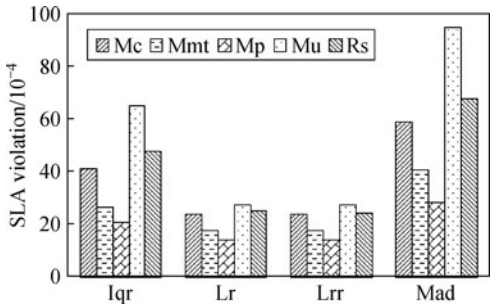

Fig. 6 SLA violations MCC

3) A comparison of MCC and PABFD VM allocation algorithm is shown in Figs. 7–9, we only consider the default four VM selection policies, the MP is not included. This can avoid the effect of MP. The results are estimated from three points of view: energy consumption, number of VM migrations and SLA violations. We can conclude that the MCC algorithm performs better than the PABFD algorithm in energy consumption no matter which combination of algorithms is used. In addition, the number of VM migrations is nearly reduced by 50% with the use of MCC algorithm when compared with the PABFD algorithm. Except for the combination of IQR-MU, the other combinations that apply the MCC VM placement algorithm have better results than the PABFD from the angle of SLA violations.

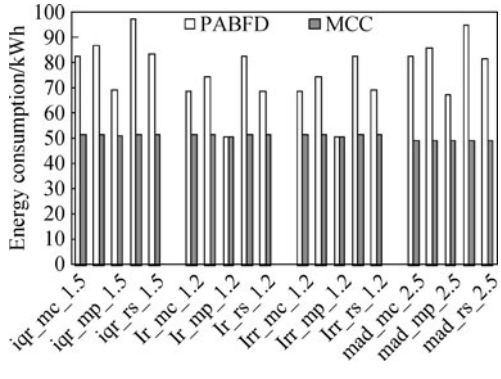

Fig. 7 Energy consumption MCC/PACFD

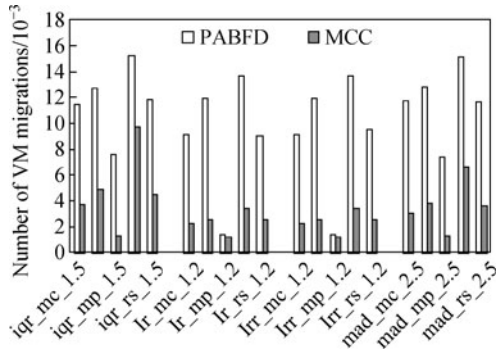

Fig. 8 The number of VM migrations MCC/PACFD

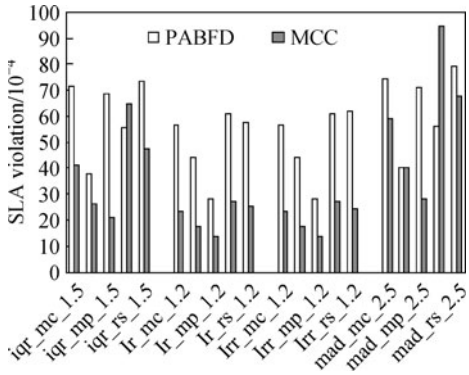

Fig. 9 SLA violation MCC/PACFD

4) Comprehensively considering the above comparisons, the combination LR-MP performs the best in energy consumption, number of VM migrations, and SLA violations. However, the performance of the combined algorithm will change as different parameters are applied. In this experiment, the parameter of LR is increased from 0.4 to 1.4 in increments of by 0.1. The result is shown in Fig. 10, in which the left *y*-axis represents the product of the energy consumption and SLA violation and we denote it as ESV, and the right *y*-axis refers to the number of VM migrations. The performance of LR-MP algorithm achieves an optimal value both in ESV and VM migrations when the parameter is 1.2.

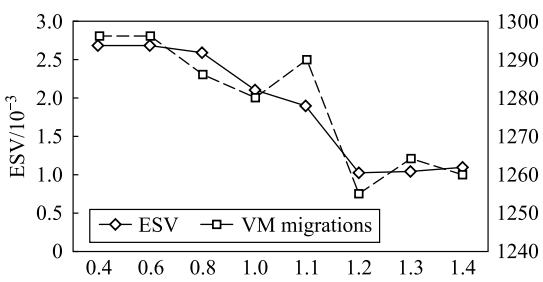

Fig. 10 The value of parameter

7 Concluding remarks and future direction

It is necessary to reduce the energy consumption without the SLA violation degradation in virtualized data centers. In this paper, we design a new VM selection policy (MP) which considers the degree of resource satisfaction and can reduce energy consumption, VM migration times and SLA violation. In addition, a VM placement policy (MCC) is proposed to search the target host that has the least correlation coefficient with the migratable VM. Experimental results show that the VM selection and VM placement policies proposed in this paper have the optimal performance in the three aspects. The performance of each combination of algorithms varies with the changing value of the parameter.

Although the policies we present have better performance in the simulated environment, we still do not know their effects in a real cloud infrastructure. In future work, we will extend them to a real-word cloud environment like OpenStack in order to evaluate the proposed policies.

Acknowledgements The subject was sponsored by the National Natural Science Foundation of China (Grant No. 61202354)

References

1. Zhu X, Young D, Watson B J, Wang Z, Rolia J, Singhal S, McKee B, Hyser C, Gmach D, Gardner R, Christian T, Cherkasova L. 1000 islands: an integrated approach to resource management for virtualized data centers. Cluster Computing, 2009, 12(1): 45–57
2. Greenberg A, Hamilton J, Maltz D A, Patel P. The cost of a cloud: research problems in data center networks. ACM SIGCOMM Computer Communication Review, 2008, 39(1): 68—73
3. Dong J, Jin X, Wang H, Li Y, Zhang P, Cheng S. Energy-saving virtual machine placement in Cloud data centers. In: Proceedings of the 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid). 2013, 618–624
4. Barroso L A, Hölzle U. The datacenter as a computer: an introduction to the design of warehouse-scale machines. Synthesis lectures on computer architecture, 2009, 4(1): 1–108
5. Nathuji R, Schwan K. Virtualpower: coordinated power management in virtualized enterprise systems. ACM SIGOPS Operating Systems Review, 2007, 41(6): 265–278
6. Kusic D, Kephart J, Hanson J, Kandasamy N, Jiang G. Power and performance management of virtualized computing environments via lookahead control. Cluster Computing, 2009, 12(1): 1–15
7. Verma A, Ahuja P, Neogi A. pMapper: power and migration cost aware application placement in virtualized systems. In: Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware. 2008, 243–264
8. Srikantaiah S, Kansal A, Zhao F. Energy aware consolidation for cloud computing. In: Proceedings of USENIX Workshop on Power Aware

Computing and Systems in conjunction with OSDI. 2008, 1–5

9. Zhu X, Young D, Watson B J, Wang Z, Rolia J, Singhal S, McKee, Hyser C, Gmach D, Gardner T, Cherkasova L. 1000 Islands: integrated capacity and workload management for the next generation data center. In: Proceedings of the 5th International Conference Autonomic Computing (ICAC). 2008, 172–181
10. Gmach D, Rolia J, Cherkasova L, Belrose G, Turicchi T, Kemper A. An integrated approach to resource pool management: policies, efficiency and quality metrics. In: Proceedings of IEEE 38th International Conference Dependable Systems and Networks (DSN). 2008, 326–335
11. Beloglazov A, Buyya R. Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in Cloud data centers. In: Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science. 2010: 4
12. Calheiros R N, Buyya R, Beloglazov A, Rose CAFD, Buyya R. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience, 2011, 41(1): 23–50
13. Beloglazov A, Buyya R. Optimal online deterministic algorithms and adaptive Heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers. Concurrency and Computation: Practice and Experience, 2012, 24(12): 1397–1420
14. Cao Z, Dong S. Dynamic VM consolidation for energy-aware and SLA violation reduction in cloud Computing. In: Proceedings of the 13th International Conference on Parallel and Distributed Computing, Applications and Technologies. 2012, 363–369
15. Bobroff N, Kochut A, Beaty K. Dynamic placement of virtual machines for managing SLA violations. In: Proceedings of the 10th IFIP/IEEE International Symposium on Integrated Network Management. 2007, 119–128
16. Wood T, Shenoy P, Venkataramani A, Yousif M. Black-box and gray-box strategies for virtual machine migration. In: Proceedings of the 4th USENIX Symposium on Networked Systems Design and Implementation. 2007, 229–242
17. Fan X, Weber WD, Barroso LA. Power provisioning for a warehouse-sized computer. In: Proceedings of the 34th Annual International Symposium on Computer Architecture. 2007, 35(2): 13-23
18. Beloglazov A, Abawajy J, Buyya R. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future Generation Computer Systems, 2012, 28(5): 755–768
19. Xu F, Liu F, Liu L, Jin H, Li B. Iaware: making live migration of virtual machines interference-aware in the cloud. IEEE Transactions on Computers, 2014, 63(12): 3012–3025
20. Song Y, Wang H, Li Y, Feng B, Sun Y. Multi-tiered on-demand resource scheduling for VM-based data center. In: Proceedings of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid. 2009, 148–155
21. Calheiros R N, Ranjan R, De Rose C A F, Buyya R. CloudSim: A novel framework for modeling and simulation of cloud computing infrastructures and services. arXiv preprint arXiv, 2009:0903.2525
22. Fan X, Weber W D, Barroso L A. Power provisioning for a warehouse-sized computer. ACM SIGARCH Computer Architecture News, 2007, 35(2): 13–23

Xiong Fu received his BS and PhD in computer science from the University of Science and Technology of China, China, in 2002 and 2007, respectively. He is currently an associate professor in computer science at the Nanjing University of Posts & Telecommunications, China. His research interests include parallel and distributed computing, and cloud computing.

Chen Zhou received her BS in computer science from the Nantong University, Nantong, in 2013. She is a master candidate at Nanjing University of Posts & Telecommunications, China. Her research interests include cloud computing, and computer networks.