J Supercomput (2014) 69:429–451
DOI 10.1007/s11227-014-1172-3

An energy-aware heuristic framework for virtual machine consolidation in Cloud computing

Zhibo Cao · **Shoubin Dong**

Published online: 5 April 2014
© Springer Science+Business Media New York 2014

Abstract Virtual machine (VM) consolidation in Cloud computing provides a great opportunity for energy saving. However, the obligation of providing suitable quality of service to end users leads to the necessity in dealing with energy-performance tradeoff. In this paper, we propose a redesigned energy-aware heuristic framework for VM consolidation to achieve a better energy-performance tradeoff. There are two main contributions in the framework: (1) establish a service level agreement (SLA) violation decision algorithm to decide whether a host is overload with SLA violation; (2) minimum power and maximum utilization policy is then proposed to improve the Minimum Power policy in previous work. Finally, we have evaluated our framework through simulation on large-scale experiments driven by workload traces from more than a thousand VMs, and the results show that our framework outperforms previous work. Specifically, it guarantees 21–34 % decrease in energy consumption, 84–92 % decrease in SLA violation, 87–94 % decrease in energy-performance metric, and 63 % decrease in execution time. And we further discuss why the redesigned framework outperforms the previous design.

Keywords Energy-performance tradeoff · Energy saving · VM consolidation · SLA violation

Z. Cao · S. Dong (✉)
School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China
e-mail: sbdong@scut.edu.cn

Z. Cao
e-mail: caozhibo2010@gmail.com

1 Introduction

As stated in the report [1]: "Cloud computing the long-held dream of computing as a utility, has the potential to transform a large part of the IT industry, making software even more attractive as a service and shaping the way IT hardware is designed and purchased." Cloud computing delivers infrastructure, platform and software as services that are made available to end users in a pay-as-you-go model. These services are referred to as infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS) in practice. Moreover, Cloud computing becomes more convenient than conventional computing such as Grid computing [2] due to Virtualization.

Benefits [3] provided by Virtualization include improved utilization, manageability, and reliability of distributed systems. For example, end users with different OS requirements can share a virtualized server more easily, OS upgrades can be conducted by VMs to reduce downtime and failures produced by VMs can be limited to the VMs where they occur. The performance of Virtualization is based on its functional capabilities. And there are three main functional capabilities [3] that encompass a broad range of Virtualization: VM isolation, VM consolidation and VM migration [4]. This work focuses on the VM consolidation and migration. Since the VM migration is part of the VM consolidation, we consider both the consolidation and migration as VM consolidation in this paper.

Moreover, modern data centers are requested to deal with a diversity of applications including serial and parallel types, short and long periods of time types. In this way, data centers will consume huge energy and make higher outlays in Cloud computing. According to statistics [5], each data center around the world consumes as much energy as 250,000 households do on average. Besides, the overall estimated energy bill for data centers is $11.5 billion in 2010 and energy costs in a typical data center double every 5 years [6]. Based on the trends from the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) [7], it has been predicted that infrastructure and energy costs would contribute about 75 %, whereas IT would contribute only 25 % to the total cost of operating a data center [8].

On the other hand, Virtualization provides the ability to consolidate VMs between physical nodes [9]. This enables the dynamic VM consolidation to the minimum physical nodes. As a result, the idle nodes can be turned off for energy saving, but inevitably it will lead to SLA violation. Therefore, it becomes a hotspot to reduce energy consumption using VM consolidation while maintaining a low-level SLA violation. To our knowledge, a VM includes three main parts: virtual CPU, virtual memory and virtual storage. Besides, there is a virtual network among VMs. Accordingly, VM consolidation should take into account these four virtual factors as much as possible in practice. However, since the virtual CPU is the main reason of energy consumption in data centers, we only consider the virtual CPU in this paper for VM consolidation.

In this paper, we propose a redesigned energy-aware framework for VM consolidation in CloudSim [10]. It includes two main contributions.

- Classify the host status overload into two types in Cloud computing: overload with SLA violation, overload with no SLA violation. To decide whether hosts are

overloaded with SLA violation, a SLA violation decision algorithm for hosts in data center is proposed.
- Minimum Power and Maximum Utilization policy for VM migration is then proposed to improve the Minimum Power policy [11], which is used to find a suitable placement for VM migration. And we further carry out experiments for performance evaluation and make discussions.

The remainder of this paper is organized as follows. In Sect. 2 we discuss the related work. In Sect. 3 we discuss the motivation for this paper and investigate two problems for the implemented framework in CloudSim. In Sect. 4 we propose our redesigned framework for VM consolidation. In Sect. 5 we deduced the qualifications for the SLA violation decision algorithm and proved that the Minimum Power and Maximum Utilization policy outperforms the Minimum Power policy. In Sect. 6 we perform experiments for the redesigned framework. Finally, we discuss the future work and conclude this paper in Sect. 7.

2 Related work

Making high efficiency of power in virtual data centers has already become an urgent issue. To this end, lots of methods have been proposed. To our knowledge, the first work to manage power efficiently in virtual data centers has been done by Nathuji and Schwan in [12], where a VirtualPower Management approach (VPM) for VM consolidation was proposed. The VPM supports the isolated and independent operation assumed by guest VMs running on virtualized platforms, and makes it possible to control and globally coordinate the effects of the diverse power management polices applied by guest VMs to virtualized resources. Different from the VPM, Stoess et al. [13] presented a framework for energy management in modular and multi-layered operating system structures which provides a unified model for VM consolidation. The similar works can be found in [14,15].

For performance-driven workload consolidation in virtual environment, Verma et al. [16] proposed an architecture of power-aware application consolidation framework (pMapper), which can incorporate various scenarios including power and performance management using Virtualization. It also provides the solution to the most practical possibility, i.e. power minimization subjected to a fixed performance requirement. Srikantaiah et al. [17] studied the interrelationships among energy consumption, resource utilization, and performance of consolidated workloads. And they modeled the consolidation problem as a modified multi-dimensional bin-packing problem. Cardosa et al. [18] presented a novel suite of techniques for VM consolidation in data centers taking advantage of the min–max and shared features inherent in Virtualization. The techniques provide a smooth mechanism for energy-performance tradeoffs in data centers running heterogeneous applications, wherein the amount of resources allocated to a VM can be adjusted based on available resources, power costs, and application utilities. The similar works can be found in [19–21]. However, the above researches are workload dependent, whereas our research is workload independent and can be deployed in a generic Cloud environment.

There are also some works that treat VM consolidation as a multi-objective optimization problem or use prediction to minimize the number of running servers for energy aware. Xu et al. [22] proposed an improved genetic algorithm with fuzzy multi-objective evaluation for VM consolidation, which obtained better energy-performance tradeoffs than other approaches. In [23], Duy et al. integrated a neural network predictor into a Green scheduling algorithm to predict future resource requirements based on historical data, which made energy saving by turning off unused servers. Since most of the VM consolidation approaches are limited to single resource and hard to distribute, Feller et al. [24] modeled the VM consolidation problem as an instance of the multi-dimensional bin-packing problem and designed a nature-inspired VM consolidation algorithm based on Ant Colony Optimization (ACO). Compared with one frequently applied algorithm (i.e. First-Fit Decreasing), the ACO-based approach achieves better energy-performance tradeoff and can be implemented in a fully distributed environment. The similar works can be found in [25–28].

In [29], Dupont et al. proposed a flexible and energy-aware framework for VM consolidation in a data center. The core element of the framework is an optimizer which is used to deal with SLA requirements, the interconnection among different data centers and energy consumption. Finally, experimental results demonstrated that the framework obtained a good energy-performance tradeoff. Anton et al. [30] defined an architectural framework and principles for energy-aware Cloud computing, and developed algorithms for energy-aware mapping of VMs to suitable Cloud resources in addition to dynamic VM consolidation. The process of the VM consolidation is as follows: firstly, set a fixed upper utilization threshold for hosts in data centers; secondly, probe each host's utilization for a period of time. If it exceeds the threshold, it is denoted as overload; finally, choose VMs from those overload hosts to migrate. However, the fixed threshold is not suitable for virtual environment with variable workloads. Therefore, in [11], they illustrated that VM consolidation should be optimized continuously in an online manner due to the variability of workloads experienced by modern applications. Then, they proposed novel adaptive heuristics for dynamic VM consolidation based on the analysis of historical data. Experimental results show that the allocation and selection algorithms can immensely save energy. However, we think that the SLA violation and energy consumption produced by the framework can be further improved. Therefore, to obtain a better energy-performance tradeoff, we propose a novel redesigned energy-aware heuristic framework for VM consolidation in CloudSim.

3 Motivation

Generally speaking, in virtual data center, the host's statuses include three types: Overload, Underload and Idle. Overload means that the host may generate SLA violation; Underload indicates that the host is in use but generates no SLA violation; Idle means that the host is available. VM consolidation requires three main steps in CloudSim [11]:

- Step 1. Choose Overload hosts in data center, and then select VMs from those hosts to migrate (denote the VMs as VmsToMigrate) until the Overload hosts become Underload.

– Step 2. Choose suitable placements for VmsToMigrate on Underload or Idle hosts based on the Minimum Power policy [11]. As a result, some Underload hosts may be turned off and become Idle, and some Idle hosts may be turned on and become Underload.
– Step 3. Denote the overall Underload hosts as OUH, exclude those Underload hosts (denoted as EUH) who receive any migration from VmsToMigrate, and denote the remaining Underload hosts as RUH. Finally, turn off the hosts of the RUH which can migrate all their VMs to other RUH or the EUH.

Following these steps, the proposed framework [10] in CloudSim significantly reduces energy consumption, while ensuring a high level of adherence to the SLA. Although the framework performs well on energy consumption and SLA violation reduction, we still consider that it can be further improved. For the framework, we have two problems:

– In step 1, the framework selects VMs from an Overload host until the host becomes Underload. Is it necessary to select a VM until the host becomes Underload? The Overload host does not mean that the host will generate SLA violation, while it only means that the host may generate SLA violation. If the Overload host does not generate SLA violation, then the migration will be in vain and result in higher energy consumption. Therefore, we need a more suitable method to decide the VM selection in this step rather than Overload decision algorithms in CloudSim.
– In step 2, the framework selects suitable placement in the Underload or Idle hosts for each migration based on the Minimum Power policy. However, different hosts have distinct energy consumption model (denoted as PowerModel), while the same hosts always have non-linear PowerModels in practice [31]. Therefore, the policy may greatly increase the number of EUH in step 3. As a result, the number of RUH in step 3 will decrease, and there will be a few hosts to be turned off, which will lead to low utilization and high energy consumption in virtual data center.

In view of the above two problems, we redesign the framework for VM consolidation in the next section.

4 The redesigned framework for VM consolidation

As shown in Table 1, we first define some notations for the hosts' statuses in virtual data center. According to the two problems discussed in Sect. 3, we redesign the framework in CloudSim. The steps about the redesigned framework are as follows:

– Step I. Choose Over hosts in data center by Overload decision algorithms proposed in CloudSim, and classify them into OverS hosts and OverNS hosts. Then select VMs from the OverS hosts until they become OverNS or Critical, and put the VMs into the VmsToMigrate. Finally, select VMs from the OverNS hosts using the proposed SLA violation decision algorithm until they become Critical, and put the VMs into the VmsToMigrate.
– Step II. Choose a maximum request utilization VM in the VmsToMigrate to migrate, and then select a host in the Under or Idle hosts to receive the VM based on the Minimum Power policy (denoted as MinPower). If the selected host does not become Critical after the first migration, then select a minimum request utilization

Table 1 Notations for the hosts' status in virtual data center

Notation	Description
Over	The Overload host's status
OverS	The Over host with SLA violation
OverNS	The Over host with No SLA violation
Under	The Underload host's status
Idle	The Idle host's status
Critical	The host does not send or receive any migration, it is a status between OverNS and Under

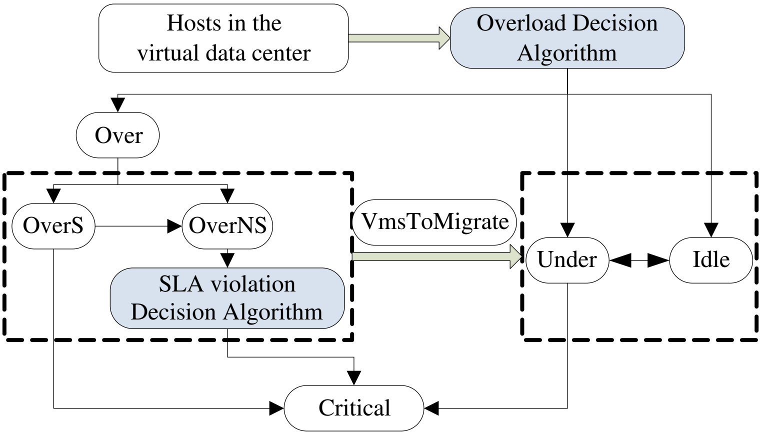

Fig. 1 The process of the redesigned framework for VM consolidation

VM in VmsToMigrate and migrate it to the host until the host become Critical (denoted as MaxUtilization). Repeat this step until there is no VM in VmsToMigrate need to migrate.
– Step III. It is the same as the step 3 of Sect. 3.

Figure 1 describes the redesigned framework on the whole. The detailed implementation of the SLA violation decision algorithm and the MinPower and MaxUtilization policy will be discussed in the next section, where we will also briefly illustrate the Overload decision algorithm, VM selection algorithm and the MinPower policy implemented in CloudSim.

5 Methodology and metrics for the redesigned framework

5.1 SLA violation decision algorithm

For simplicity, the algorithm can be abbreviated to SLAVDA. The SLAVDA aims to decide whether a host generates SLA violation with great probability. Some neces-

Table 2 The description of the parameters defined in algorithm SLAVDA

Parameter	Description
H_i	The ith host in the data center
V_{ij}	The jth virtual machine on the ith host
N	The number of hosts in the data center
M_i	The number of virtual machines on the ith host
$\mathrm{MU}H_i$	The Maximum Utilization (MIPs) of the host H_i
$\mathrm{AU}H_i$	The total Allocated Utilization (MIPs) for VMs by H_i
$\mathrm{MU}V_{ij}$	The Maximum Utilization (MIPs) of the jth VM on H_i
$\mathrm{RU}V_{ij}$	The Request Utilization (MIPs) of the jth VM on H_i
$\mathrm{RU}V_j$	The Request Utilization (MIPs) for the jth VM in VmsToMigrate

sary parameters are defined to deduce the qualifications for SLAVDA as shown in Table 2.

First of all, we need to find out the necessary and sufficient condition for SLA violation. According to [11], when the total request utilization of the VMs exceeds the allocated utilization of them on H_i, H_i will generate SLA violation. If they are equal, it will be assumed to generate no SLA violation. Then, we can easily deduce Eq. (1).

$$\begin{cases} \frac{\sum_{j=1}^{M_i} \mathrm{AU}V_{ij}}{\mathrm{MU}H_i} < \frac{\sum_{j=1}^{M_i} \mathrm{RU}V_{ij}}{\mathrm{MU}H_i}, & \text{SLA violation} \\ \frac{\sum_{j=1}^{M_i} \mathrm{AU}V_{ij}}{\mathrm{MU}H_i} = \frac{\sum_{j=1}^{M_i} \mathrm{RU}V_{ij}}{\mathrm{MU}H_i}, & \text{No SLA violation} \\ \frac{\sum_{j=1}^{M_i} \mathrm{AU}V_{ij}}{\mathrm{MU}H_i} > \frac{\sum_{j=1}^{M_i} \mathrm{RU}V_{ij}}{\mathrm{MU}H_i}, & \text{impossible} \end{cases} \tag{1}$$

$$\begin{cases} 1.0 < x_i, & \text{SLA violation} \\ 0 \le x_i \le 1.0, & \text{No SLA violation} \\ x_i < 0, & \text{impossible} \end{cases} \tag{2}$$

Actually, the total allocated utilization of the VMs on H_i can never exceed the maximum utilization of the host. It means that if the request utilization of the VMs on H_i exceeds the maximum utilization, the host will definitely generate SLA violation. For simplicity, the ratio of $\sum_{j=1}^{M_i} \mathrm{RU}V_{ij}/\mathrm{MU}H_i$ is set to x_i(for each H_i). Then, Eq. (2) can be derived from Eq. (1).

According to Eq. (2), the necessary and sufficient condition for SLA violation is $x_i > 1.0$, where x_i is less than or equal to the ratio. When $x_i > 1.0$, H_i will be in OverS. When $0 < x_i \le 1.0$, H_i will be in OverNS or Under. And when $x_i = 0$, H_i will be in Idle. The SLAVDA decides whether the OverNS hosts transform into OverS ($x_i > 1.0$) with great probability.

Afterwards, we need to find out the condition transforming OverNS into OverS. First of all, the request utilization changes timely, so x_i is a random variable, where the mean μ_i and standard deviation σ_i of x_i can be calculated with the historical records. Denote $p(x_i)$ as the probability density function of x_i. Then, we can calculate the probability of the condition next time using Eq. (3).

$$
\begin{aligned}
P\{x_i \geq 1\} &= P\{x_i - \mu_i \geq 1 - \mu_i\}(\text{set}1 - \mu_i = \varepsilon_i) \\
&= P\{x_i - \mu_i \geq \varepsilon_i\} \\
&\leq \int_{x_i - \mu_i \geq \varepsilon_i} \left(\frac{x_i - \mu_i}{\varepsilon_i}\right)^2 p(x_i)\mathrm{d}x_i \\
&= \frac{1}{\varepsilon_i^2} \int_{\varepsilon_i + \mu_i}^{\infty} (x_i - \mu_i)^2 p(x_i)\mathrm{d}x_i \\
&\leq \frac{1}{\varepsilon_i^2} \int_{0}^{\infty} (x_i - \mu_i)^2 p(x_i)\mathrm{d}x_i \\
&= \frac{\sigma_i^2}{\varepsilon_i^2}
\end{aligned} \tag{3}
$$

According to Eq. (3), if the mean is greater than 1, H_i will be assumed to be OverS in the next time. If the mean is smaller than 1, H_i's status will be decided by $\sigma_i^2/\varepsilon_i^2$. Since probability cannot exceed 1, the host will generate SLA violation with great probability when the ratio is greater than 1. Therefore, the qualifications of the condition can be formulated by Eq. (4).

$$
\begin{cases}
\forall \sigma_i, \quad \mu_i \geq 1.0 \\
\sigma_i \geq 1 - \mu_i, \quad \mu_i < 1.0
\end{cases} \tag{4}
$$

5.2 The overload decision algorithm

For simplicity, the algorithm can be abbreviated to ODA, which aims to decide a host Over or not. Up to now, four ODAs have been implemented in CloudSim, i.e. Interquartile Range (IQR), Local Regression (LR), Robust Local Regression (LRR) and Median Absolute Deviation (MAD).

IQR As a measure of statistical dispersion in descriptive statistics, the IQR, also called the midspread or middle fifty, is equal to the difference between the upper (75 %) and lower (25 %) quartiles.

LR In linear algebra, regression aims to find a trend line for a large set of data points. The LR aims to find a trend line by minimizing the sum of the absolute weighted distances between the line and the points.

LRR The LR is vulnerable to the outliers caused by heavy-tailed or other distributions. To make it robust, the LRR [34] is proposed to assign an additional weight to each absolute distance in the LR so that it can weaken the ourliers.

MAD The process of MAD is as follows: firstly, calculate the median value of a set A. Then, take the absolute distances between the median and the points into set B. Finally, figure out the median value of the set B.

5.3 The VM selection algorithm

For simplicity, the algorithm can be abbreviated to VMSA, which aims to select VMs from Over hosts and prevent them from being Over. Four VMSAs have been imple-

Input: V_j in the VmsToMigrate, HUI(the set of the Under and Idle hosts), $P(AUh)$ the PowerModel for the host h, minPower (Double.MAX_VALUE).

Output: allocatedHost (a host in the set HUI)

1.**FOR** H_k **IN** HUI

2. diffPower=$P(RUV_{kj}+AUH_k)$-$P(AUH_k)$;

3. **IF** minPower > diffPower

4. minPower=diffPower;

5. allocatedHost=H_k;

6. **END**

7.**END**

8.**return** allocatedHost;

Fig. 2 The pseudo-code for the MinPower policy

mented in CloudSim, i.e. Minimum Migration Time (MMT), Minimum Utilization (MU), Random Selection (RS) and Maximum Correlation (MC).

MMT The MMT aims to select a VM from Over host with the least migration time.

MU The MU aims to select a VM from Over host with the least request utilization.

RS The RS aims to randomly select a VM from Over host.

MC The MC aims to select a VM with the maximum correlation with other VMs on Over host.

5.4 The MinPower and MaxUtilization policy

For simplicity, the policy can be abbreviated to MPMU. The MinPower policy (MP) has been proposed in CloudSim, whereas the MaxUtilization policy is proposed in this paper. Different from the SLAVDA, the MPMU aims to find suitable placement for VmsToMigrate.

The MP is used to select a host for a VM with the least energy consumption and the pseudo-code of it is shown in Fig. 2. As discussed in Sect. 3, the MP may greatly increase the number of EUH and leads to higher energy consumption. To this end, we propose MPMU (1) sort VmsToMigrate into the set DescVms in descending order and the set AscVms in ascending order, respectively; (2) choose the first VM of DescVms and find a suitable host for it through the MP, then remove the VM from the DescVms and the AscVms; (3) choose the first VM of AscVms. If the VM would not lead the suitable host into Over, then migrate the VM to the host, and remove the VM from the DescVms and the AscVms; (4) repeat steps (2) and (3) until there are no VMs in DescVms and AscVms. The pseudo-code of the MPMU is shown in Fig. 3.

For illustrating the better performance of MPMU, we should prove that the number of EUH produced by MPMU is less than or equal to the one produced by MP, then the MPMU will turn off more hosts than the MP and consumes less energy.

```
Input: VmsToMigrate; MinPower,which is a function for VM and return a suitable host for the VM.
Output: MigrationMap, is a type of Map<Host,Vm>.
AscVms=sortAscending(VmsToMigrate);
DescVms=sortDescending(VmsToMigrate);
FOR V_j IN DescVms
    allocatedHost=MinPower(V_j);
    AscVms.remove(V_j);
    DsecVms.remove(V_j);
    FOR V_k IN AscVms
        IF NOT allocatedHost.isHostOverUtilized(V_k);
            MigrationMap.put(allocatedHost,V_k);
            allocatedHost.create(V_k);
            AscVms.remove(V_k);
            DsecVms.remove(V_k);
        ELSE
            break;
        END
    END
END
return MigrationMap;
```

Fig. 3 The pseudo-code for the MPMU policy

First of all, denote the size (the number of VMs) of VmsToMigrate as n, denote the number of EUH produced by MP as $N(n)_{\mathrm{MP}}$ and the one produced by MPMU as $N(n)_{\mathrm{MPMU}}$.

Theorem 1 *For any size of VmsToMigrate n, the condition $N(n)_{MPMU} \leq N(n)_{MP}$ holds.*

Proof (a) When $n = 1$, the MPMU degenerates to the MP, so $N(1)_{\mathrm{MPMU}} = N(1)_{\mathrm{MP}}$. (b) Given $n = k$, there is $N(k)_{\mathrm{MPMU}} \leq N(k)_{\mathrm{MP}}$. Then, we should prove that when $n = k + 1$, there is $N(k+1)_{\mathrm{MPMU}} \leq N(k+1)_{\mathrm{MP}}$. Compared to $n = k$, there will be an additional VM needs to be migrated to the EUH or the RUH when $n = k + 1$. Then, there will be two cases for the additional VM: the EUH does not have enough room for the VM and the EUH has enough room for the VM.

1. The EUH does not have enough room for the additional VM. In this case, the MP needs to migrate the VM to the RUH. So does the MPMU, thus there is $N(k+1)_{\mathrm{MP}} = N(k)_{\mathrm{MP}} + 1$ and $N(k+1)_{\mathrm{MPMU}} = N(k)_{\mathrm{MPMU}} + 1$. Then, we can get $N(k+1)_{\mathrm{MPMU}} = N(k+1)_{\mathrm{MP}}$.

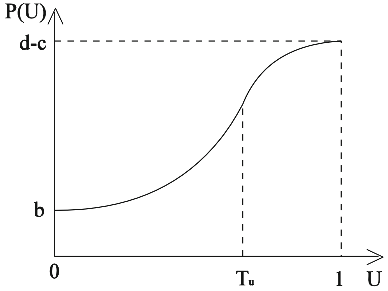

Fig. 4 The assumed PowerModel

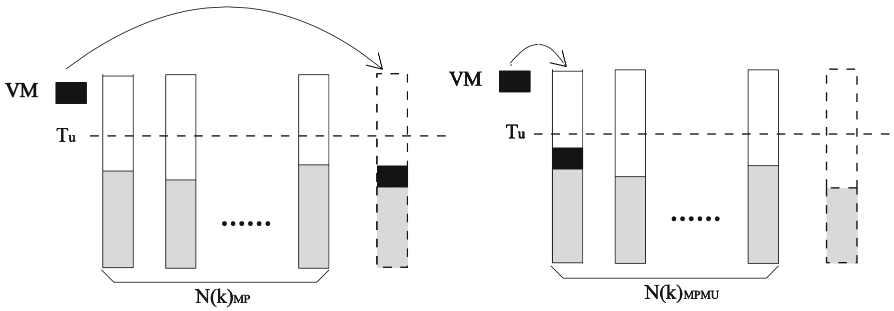

Fig. 5 The comparison between the MP and the MPMU with enough capacity for the additional VM

$$P(U) = \begin{cases} aU^2 + b, & 0 \le U < T_u (a, b \in R^+) \\ -cU^2 + d, & T_u \le U \le 1.0 (c, d \in R^+) \end{cases} \tag{5}$$

$$\frac{\mathrm{d}P(U)}{\mathrm{d}U} = \begin{cases} aU, & 0 \le U < T_u (a \in R^+) \\ -cU, & T_u \le U \le 1.0 (c \in R^+) \end{cases} \tag{6}$$

2. The EUH has enough room for the additional VM. In this case, the MPMU only needs to migrate the VM to EUH, so there is $N(k+1)_{\mathrm{MPMU}} = N(K)_{\mathrm{MPMU}}$. It means that the MPMU does not need to occupy the RUH. Therefore, once we find a situation to ensure the MP will occupy the RUH, we will fully prove the theorem. Suppose the OUH have the same PowerModel which is available on the website [32]. Since the EUH belongs to the OUH, it also has the same PowerModel. And the relationship between the energy consumption (P) and the utilization (U) of the host in OUH can be assumed to Eq. (5) and Fig. 4. In Eq. (5), T_u is the critical point of OUH's utilization. Then, the first derivative of $P(U)$ can be derived from Eq. (5) as shown in Eq. (6). When $0 \le U < T_u$, the first derivative is monotonically increasing, and when $T_u \le U \le 1.0$, the first derivative is monotonically decreasing. It means that if the host's utilization of the OUH is the least and less than T_u, then it will obtain the VM. If it belongs to the EUH, then we can get $N(k+1)_{\mathrm{MP}} = N(k)_{\mathrm{MP}}$ for the MP. If it belongs to the RUH, then we can get $N(k+1)_{MP} = N(k)_{\mathrm{MP}} + 1$ for the MP (see Fig. 5). So there is $N(k+1)_{\mathrm{MPMU}} \le N(k+1)_{\mathrm{MP}}$.

In a word, we can easily prove the Theorem 1 according to the above analysis. □

5.5 Metrics

Meeting QoS requirements is very important for Cloud computing. QoS requirements are commonly formalized as SLAs, which can be determined in terms of characteristics such as minimum throughput, maximum response time or minimum bandwidth and so on. These characteristics are workload or application dependent. However, the redesigned framework belongs to IaaS layer in Cloud computing and should be workload independent. Therefore, we use those SLA-related metrics defined in [11] to evaluate the redesigned framework in our experiments. Simultaneously, we also use some other metrics including energy consumption, migrations and execution time.

(a) SLATAH (SLA violation Time per Active Host): The percentage of time, during which active hosts have experienced the CPU utilization of 100 %. When a host experiences 100 % utilization, it will not be able to allocate enough CPU to the VMs on it, so it will generate SLA violation. The SLATAH can be calculated using Eq. (7), where T_{s_i} is the SLA violation time and T_{a_i} is the active time for H_i.

$$\text{SLATAH} = \frac{1}{N}\sum_{i=1}^{N}\frac{T_{s_i}}{T_{a_i}} \tag{7}$$

(b) Migrations: The total number migrations happened in the N hosts in data center.
(c) Energy: The total energy consumption in data center.
(d) PDM (Performance Degradation due to Migrations): The PDM can be calculated using Eq. (8), where M is the number of VMs in data center, C_{d_j} is the estimate of the migration performance degradation of VM j, C_{r_j} is the total CPU utilization requested by j during its lifetime and C_{d_j} is estimated as 10 % of the requested CPU utilization of j during its migration.

$$\text{PDM} = \frac{1}{M}\sum_{j=1}^{M}\frac{C_{d_j}}{C_{r_j}} \tag{8}$$

(e) SLAV (SLA Violation) and ESV (Energy and SLA Violation): The SLATAH is used to evaluate host-level SLA violation due to host overloading. And the PDM is used to evaluate VM-level SLA violation due to the VM migration. Since the two metrics are independent, a combined metric is needed to evaluate the two SLA violations. As a result, the SLAV is proposed to evaluate the two SLA violations, which can be calculated using Eq. (9). Since the Energy and the SLAV are two main metrics, the ESV is proposed to combine the two metrics, which can be calculated using Eq. (10).

$$\text{SLAV} = \text{SLATAH} \cdot \text{PDM} \tag{9}$$

$$\text{ESV} = \text{SLAV} \cdot \text{Energy} \tag{10}$$

(f) ETF (Execution Time of the Framework): The execution time of the frameworks.

Table 3 Power consumption at different load levels in Watts

PowerModels	0 %	10 %	20 %	30 %	40 %	50 %	60 %	70 %	80 %	90 %	100 %
The G4 (W)	86	89.4	92.6	96	99.5	102	106	108	112	114	117
The G5 (W)	93.7	97	101	105	110	116	121	125	129	133	135

Table 4 The four VM types used in our experiments

VM types	Cores	Capacity (MIPs)	RAM (MB)	Storage (GB)	Bandwidth (Mbit/s)
Large	1	2,500	870	2.5	100 (Gaussian)
Medium	1	2,000	1,740	2.5	100 (Gaussian)
Small	1	1,000	1,740	2.5	100 (Gaussian)
Micro	1	500	413	2.5	100 (Gaussian)

6 Performance evaluation

As the targeted system is IaaS in Cloud computing, it is better to evaluate the redesigned framework on a large-scale virtualized data center. However, it is very difficult to conduct repeatable large-scale experiments on a real infrastructure. Therefore, to ensure the repeatability of the experiments, simulations have been chosen as a suitable way to evaluate the redesigned framework. CloudSim has been chosen in our experiments, which has been developed by the Cloud Computing and Distributed Systems (CLOUDS) Laboratory, University of Melbourne. It supports both system and behavior modeling of Cloud system components such as data centers, resource provisioning policies and energy-efficient management.

The statistical data are obtained from the cluster located in Guangdong Key Laboratory of Computer Network: Sun Blade 6000 which consists of one master and four slaves, each node is equipped with Dual-core AMD Opteron (tm) Processor 2218 2.6 GHz $\times$ 2, DDR2 8 G of memory.

6.1 Experiments setup

We simulated a data center with 800 nodes, half of which are HP ProLiant ML110 G4 (denote as G4) servers, and the other half are HP ProLiant ML110 G5 (denote as G5) servers. To guarantee better performance evaluation of VM consolidation, both the servers have two cores. The CPU frequency of each core of G4 is 1,860 MIPS with a memory of 4,096 MB. And the CPU frequency of each core of G5 is 2,660 MIPS with a memory of 4,096 MB. Both of them are modeled with a bandwidth of 1 Gbit/s. The PowerModels of them are available on the website [33] as shown in Table 3. According to Amazon EC2 Instance Types [36], we modeled the four types of VMs as shown in Table 4 and each type has 1 core. Initially, the VM are all allocated with the maximum requested resource according to its type. During their lifetime, each VM is allocated with the dynamic requested resource according to the workload (see Sect. 6.2).

Table 5 The four types of frameworks for our experiments

The frameworks	Description
Origin	The original framework
SLAVDA	The Origin with SLAVDA improvement
MPMU	The Origin with MPMU improvement
SDAMU	The SLAVDA with MPMU improvement

Table 6 The signs used in following figures for the 16-type combinations of the ODA and the VMSA

Combination	Sign	Combination	Sign	Combination	Sign	Combination	Sign
IQR_MC_1.5	A	LR_MC_1.2	E	LRR_MC_1.2	I	MAD_MC_2.5	M
IQR_MU_1.5	B	LR_MU_1.2	F	LRR_MU_1.2	J	MAD_MU_2.5	N
IQR_MMT_1.5	C	LR_MMT_1.2	G	LRR_MMT_1.2	K	MAD_MMT_2.5	O
IQR_RS_1.5	D	LR_RS_1.2	H	LRR_RS_1.2	L	MAD_RS_2.5	P

6.2 Workload data and the evaluated frameworks

To make the results more convincing, the real workload provided by CloudSim is adopted in our experiments. The only characteristic recorded in the workload is the CPU utilization of VMs. It is a part of CoMon project [35], which is collected from more than a thousand VMs from the servers located at more than 500 places around the world. It has 10-day records from March to April in 2011 and a time interval of 5 min between two consecutive records. In addition, Table II of [11] gives a brief analysis of the workload.

In the redesigned framework, the SLAVDA is an improvement for the step 1 (Sect. 3) and the MPMU is an improvement for the MP in the step 2. Therefore, for different improvements, the redesigned framework can be classified into four types as shown in Table 5.

6.3 Results and analysis

In the four frameworks, the four ODAs and the four VMSAs are equipped. Therefore, there are 16-type combination policies for the ODA and the VMSA. According to the results of [11], we set constant parameters for the ODA, i.e. 1.5 for IQR, 1.2 for LR, 1.2 for LRR and 2.5 for MAD, respectively. For simplicity, each combination is represented by a sign as shown in Table 6.

(a) The SLATAH evaluation: In Fig. 6, we use the 10-day workload to evaluate the four frameworks for the 16-type combination policies. In each subfigure, each candlestick represents 10 experimental results of the 10-day workload, within which the hollow cylinder represents 80 % of the results (the same below). For better comparison, we use the average of the 10 results as an evaluation value (the same below). In this way, the minimum and the maximum evaluation values for the Origin can be easily figured out which are 5.5 and 7.87 %, respectively. For the

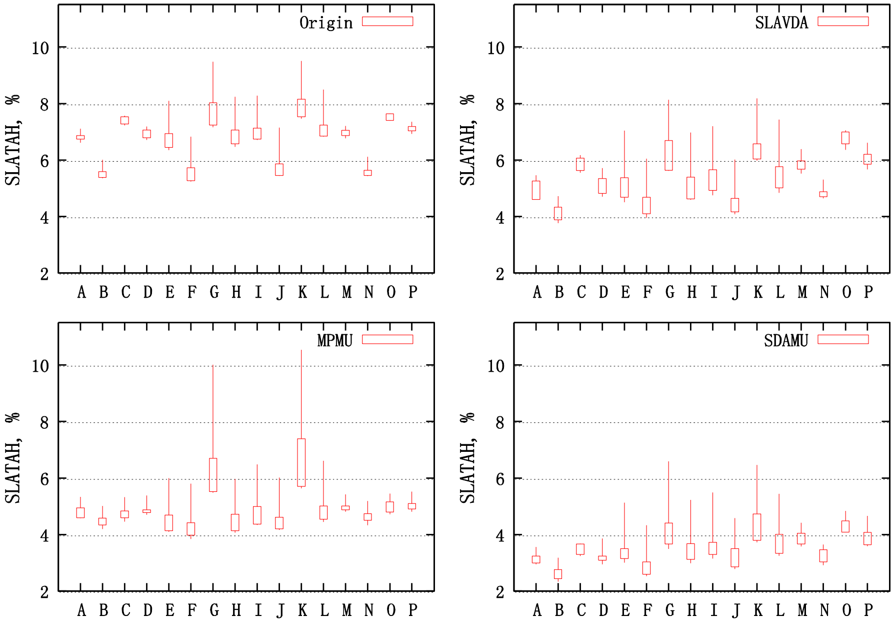

Fig. 6 The SLATAH evaluation for the four frameworks

SLAVDA, the values are 4.1 and 6.74 %, respectively. For the MPMU, the values are 4.29 and 6.65 %, respectively. And for the SDAMU, the values are 2.66 and 4.36 %, respectively. Compared with the Origin, the SLAVDA obtains a decrease of 25.4–47.8 %, the MPMU obtains a decrease of 18.2–42.8 %, and the SDAMU obtains a decrease of 51.6–66.2 % on the SLATAH for the 16-type combinations, respectively.

(b) The Migrations evaluation: In Fig. 7, for the Origin, the minimum and maximum evaluation values are 31,701 migrations and 65,762 migrations, respectively. For the SLAVDA, the values are 15,615 migrations and 33,417 migrations, respectively. For the MPMU, the values are 16,383 migrations and 47,311 migrations, respectively. And for the SDAMU, the values are 8,466 migrations and 17,588 migrations. Compared with the Origin, the SLAVDA obtains a decrease of 33.3–67.9 %, the MPMU obtains a decrease of 10.8–57 %, and the SDAMU obtains a decrease of 68.4–84.7 % on the Migrations for the 16-type combinations, respectively.

(c) The Energy evaluation: In Fig. 8, for the Origin, the minimum and maximum evaluation values are 141.933 and 171.444 kWh, respectively. For the SLAVDA, the values are 121.839 and 129.886 kWh, respectively. For the MPMU, the values are 114.875 and 120.463 kWh, respectively. And for the SDAMU, the values are 109.125 and 111.965 kWh. Compared with the Origin, the SLAVDA obtains a decrease of 11.8–27 %, the MPMU obtains a decrease of 15.2–29.8 %, and the SDAMU obtains a decrease of 21.2–34.7 % on the Energy for the 16-type combinations, respectively.

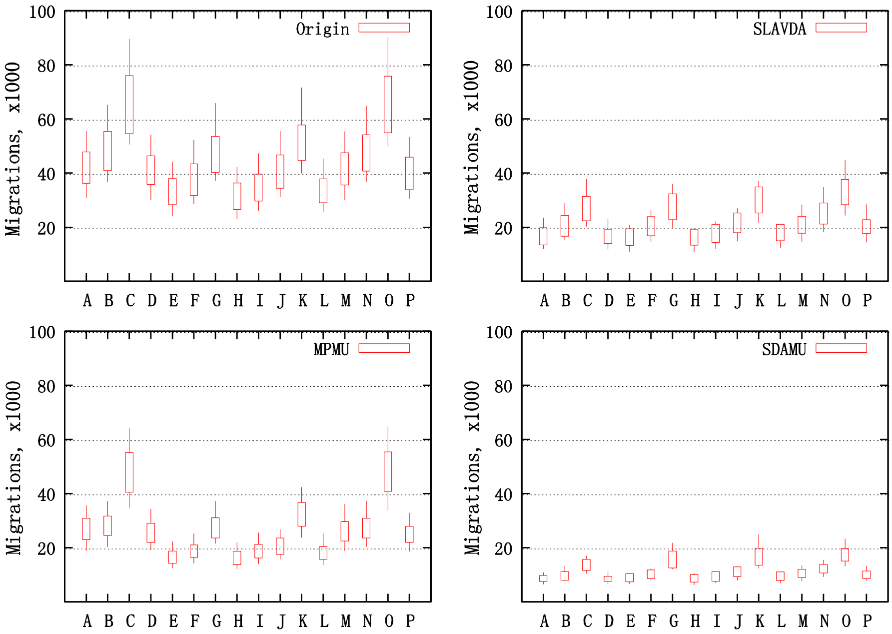

Fig. 7 The Migrations evaluation for the four frameworks

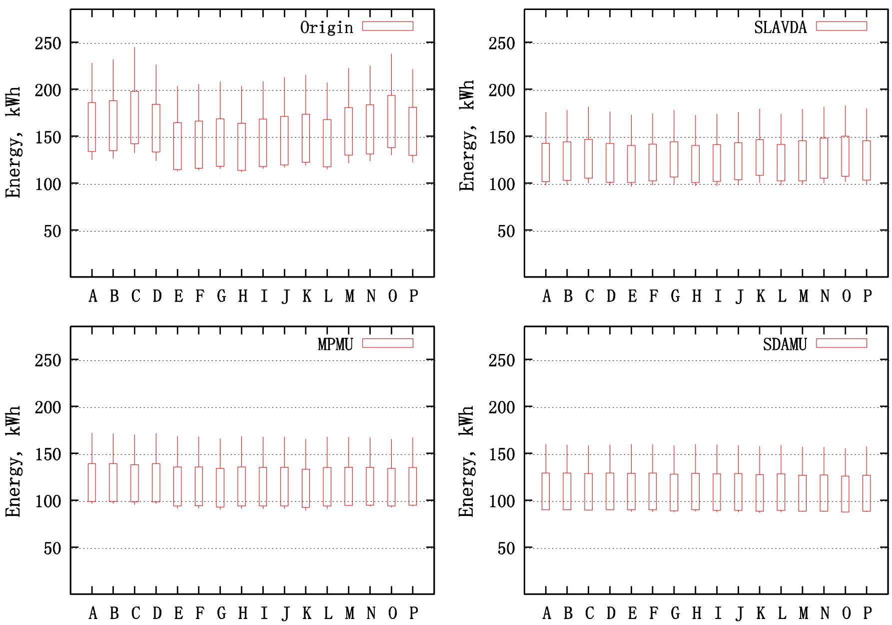

Fig. 8 The Energy evaluation for the four frameworks

(d) The PDM evaluation: In Fig. 9, for the Origin, the minimum and maximum evaluation values are 0.126 and 0.194 %, respectively. For the SLAVDA, the values are 0.072 and 0.109 %, respectively. For the MPMU, the values are 0.069 and

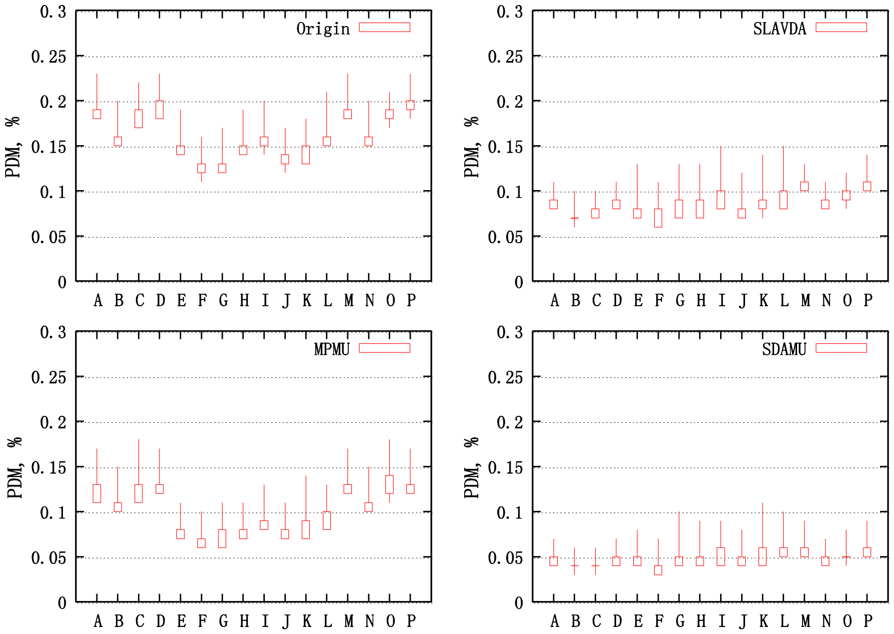

Fig. 9 The PDM evaluation for the four frameworks

0.131 %, respectively. And for the SDAMU, the values are 0.041 and 0.06 %. Compared with the Origin, the SLAVDA obtains a decrease of 42.9–62.9 %, the MPMU obtains a decrease of 15.1–44.8 %, and the SDAMU obtains a decrease of 67.5–78.9 % on the PDM for the 16-type combinations, respectively.

(e) The SLAV evaluation: In Fig. 10, for the Origin, the minimum and maximum evaluation values are 0.707 and 1.379 ‰ respectively. For the SLAVDA, the values are 0.297 and 0.663 ‰ respectively. For the MPMU, the values are 0.301 and 0.655 ‰ respectively. And for the SDAMU, the values are 0.11 and 0.24 ‰. Compared with the Origin, the SLAVDA has 57.9–78.4 % decrease, the MPMU has 31.5–64.9 % decrease, and the SDAMU has 84.4–92 % decrease on the SLAV for the 16-type combinations, respectively.

(f) The ESV evaluation: In Fig. 11, for the Origin, the minimum and maximum evaluation values are 10.06 and 23.25 ‰ respectively. For the SLAVDA, the values are 3.7 and 8.34 ‰ respectively. For the MPMU, the values are 3.43 and 7.56 ‰ respectively. And for the SDAMU, the values are 1.22 and 2.57 ‰. Compared with the Origin, the SLAVDA obtains a decrease of 63.2–84.1 %, the MPMU obtains a decrease of 42.5–75.1 %, and the SDAMU obtains a decrease of 87.8–94.7 % on the ESV for the 16-type combinations, respectively.

(g) The ETF evaluation: For each framework, there are 160 scripts for the 10-day workload and the 16-type combination policies. We record the wall-clock time of the scripts for each framework and repeat the experiment 10 times as shown in Table 7. For the Origin, the ETF is 42,185 s on average. Compared with the Origin, the SLAVDA's ETF is 19,179 s on average and obtains a decrease of 54.5 %, the

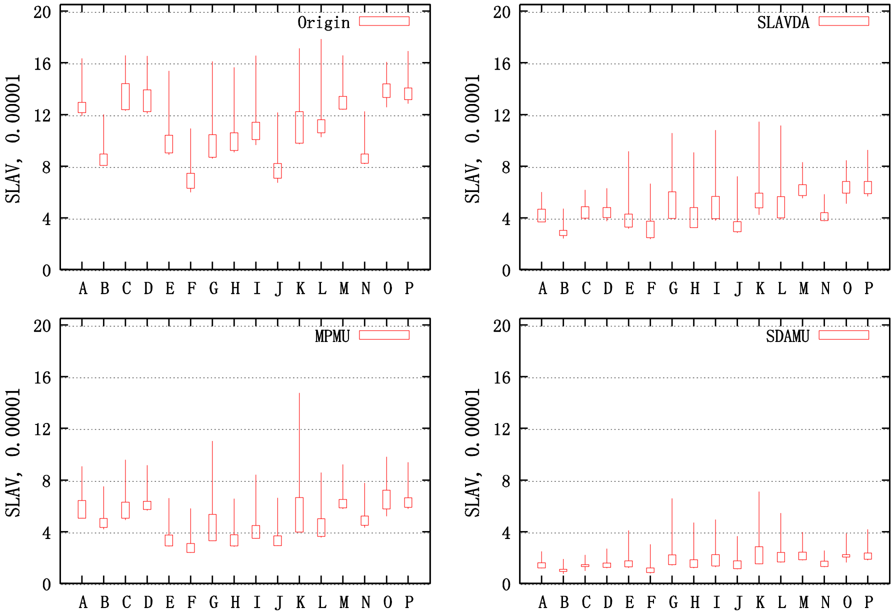

Fig. 10 The SLAV evaluation for the four frameworks

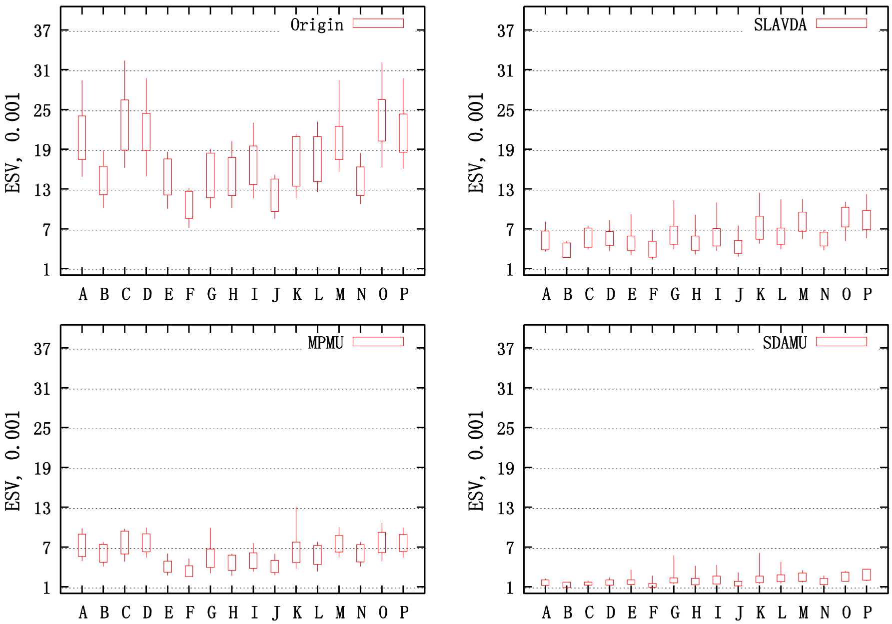

Fig. 11 The ESV evaluation for the four frameworks

MPMU's ETF is 68,423 s on average and obtains an increase of 62.2 %, and the SDAMU's ETF is 15,265 s on average and obtains a decrease of 63.8 % for the scripts, respectively.

Table 7 The 10-time repeated experiments for the ETF evaluation

Experiment number		The ETF evaluation (s)		
	Origin	SLAVDA	MPMU	SDAMU
1	42,218	19,248	68,512	15,391
2	42,161	19,189	68,538	15,360
3	42,208	19,226	68,726	15,141
4	42,174	19,184	68,116	15,263
5	42,229	19,093	68,464	15,264
6	42,305	19,266	68,155	15,267
7	42,149	19,145	68,236	15,238
8	41,984	19,235	68,358	15,291
9	42,253	19,089	68,529	15,185
10	42,172	19,118	68,595	15,252
Average	42,185	19,179	68,423	15,265

In a nutshell, the SDAMU and the SLAVDA outperform the Origin for the six metrics and the ETF. The MPMU performs better than the Origin for the six metrics, but worse than the Origin for the ETF.

6.4 Discussion

Although the redesigned framework outperforms the Origin, we still need to investigate the reasons leading to the lower SLAV and the lower Energy.

(a) The reasons as to the lower SLAV: The SLAV is the product of the PDM and the SLATAH. Therefore, once we find the reasons to the lower PDM and SLATAH, we can find the reasons to the lower SLAV.

According to Eq. (8), the PDM includes the Migrations, which means that the PDM has some relationship with the Migrations. To find the relationship, we choose a 6-day workload from the 10-day workload. With the experimental results of the 6-day workload, we can draw Fig. 12. As shown in Fig. 12, the PDM grows slowly with violent oscillation with the increase of the Migrations, which means that the Migrations are the main reason leading to the lower PDM. Since there exists violent oscillation, we should also find the corresponding reason. To this end, assume that there are M_j migrations of the jth VM, denote $C_{d_{i,j}}$ as the performance degradation of the jth VM in the ith migration, and denote $C_{t_{i,j}}$ as the requested CPU utilization for the jth VM in the ith migration. Then we can derive Eq. (11) from Eq. (8). In Eq. (11), the PDM includes four parameters, where the product of the M_j and the M equals to the Migrations, and $\sum_{j=1}^{M} C_{r_j}$ is a constant for the workload. Obviously, $C_{t_{i,j}}$ is the main reason leading to the violent oscillation. Therefore, the reason to the lower PDM is that the redesigned framework makes much less migrations and average request CPU utilization for each migration.

$$\text{PDM} = \frac{1}{M}\sum_{j=1}^{M}\sum_{i=1}^{M_j}\frac{C_{d_{i,j}}}{C_{r_j}} = \frac{1}{10M}\sum_{j=1}^{M}\sum_{i=1}^{M_j}\frac{C_{t_{i,j}}}{C_{r_j}} \tag{11}$$

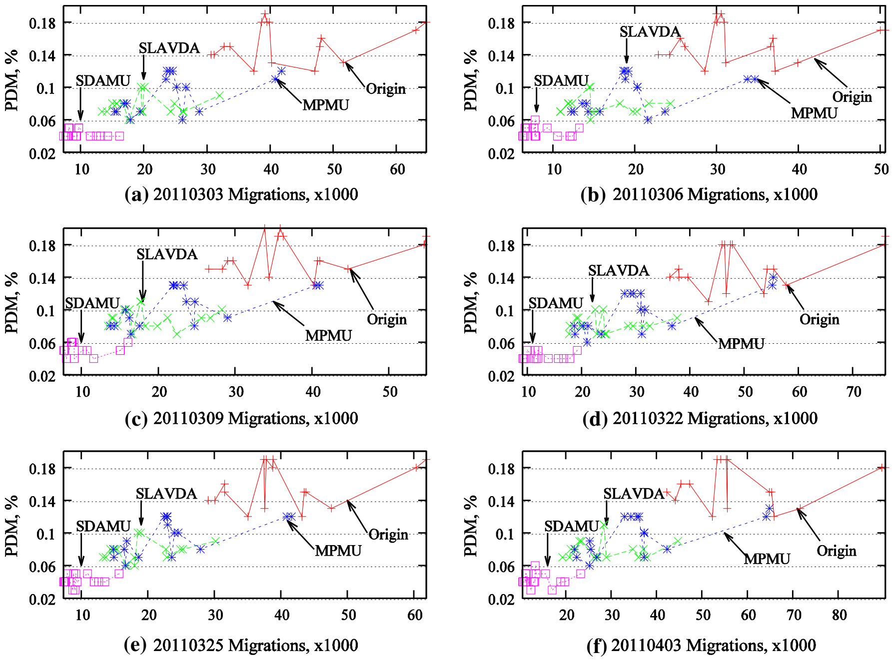

Fig. 12 The relationship between the Migrations and the PDM in the 6-day workload

The SLATAH is the average SLA violation time per active host. Migration will cause performance degradation, which will lead some hosts into SLA violation. With the experimental results of the 6-day workload, we can draw Fig. 13. As shown in Fig. 13, the SLATAH grows slowly with violent oscillation with the increase of the Migrations, which means that the Migrations is also the main reason leading to the lower SLATAH. Since there exists violent oscillation, we should also find the corresponding reason. According to the definition of the SLATAH, the migration time decides how long hosts experience SLA violation, which depends on the requested memory of the VMs. Obviously, the average requested memory of each VM is the main reason to the violent oscillation. Therefore, the reason to the lower SLATAH is that the redesigned framework makes much less migrations and average requested memory.

In a nutshell, the lower PDM and SLATAH lead to the lower SLAV.

(b) The reasons to the lower Energy. For the SLAVDA, it selects much less VMs to migrate that makes less EUH and turns off more hosts, which leads to energy saving. For the MPMU, it also reduces the number of EUH and turns off more hosts resulting in less energy consumption. Since the SDAMU takes advantage of those two, it can also reduce energy consumption.

7 Conclusion and future work

To obtain quick ROI (Return On Investment), Cloud providers should reduce energy consumption as much as possible while keeping a low-level SLA violation in data center, which is also called energy-performance tradeoff. Energy saving means that data

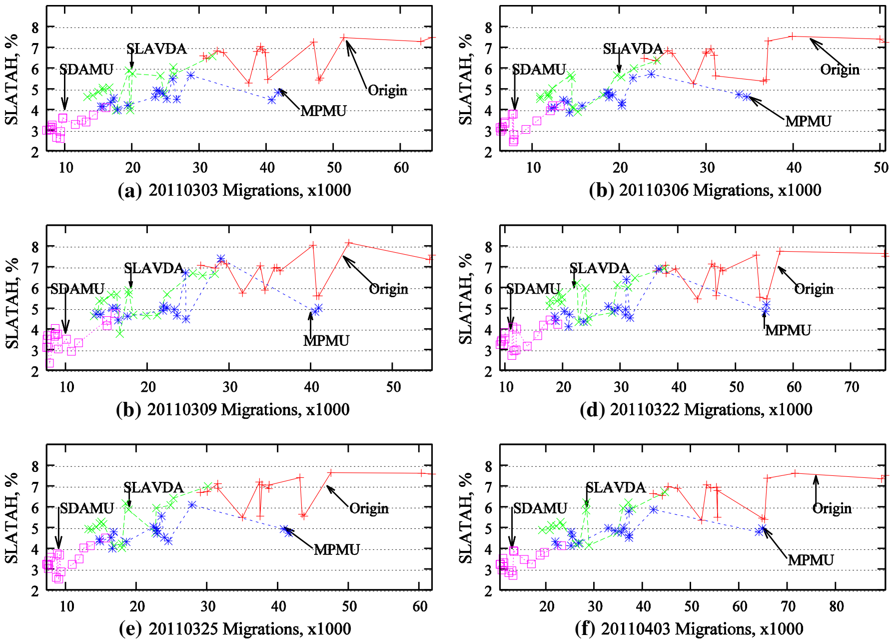

Fig. 13 The relationship between the Migrations and the SLATAH in the 6-day workload

centers have to choose better cooling systems and VM consolidation frameworks. SLA violation indicates that data centers should take user requirements as qualifications for the VM consolidation framework. In this paper, we redesign an energy-aware heuristic framework for VM consolidation to make better energy-performance tradeoff. Firstly, we classify the host status overload into two types, i.e. OverS and OverNS. And design the SLAVDA to decide whether an OverNS host is OverS or not. Secondly, the MPMU is proposed to improve the MP in [11]. Finally, we have evaluated our framework and the Origin framework through simulation on large-scale experiments driven by workload traces collected from more than a thousand PlanetLab VMs, and the results show that our framework gets a better energy-performance tradeoff (the ESV metric), and then discuss why the redesigned framework outperforms the Origin.

More work is still underway for the redesigned heuristic framework. It has not been evaluated on large-scale experiments in practice and considered other resource requirements such as IO, bandwidth and storage. Therefore, the proposed framework will be evaluated in real environment taking into account IO, bandwidth and storage in future work.

Acknowledgments Thanks to Anton Beloglazov, who gives us a positive confirmation about the bug [37] in the MinPower policy deployed in CloudSim. As a result, we can correct the policy and finish this paper. This work is partly supported by Natural Science Foundation of China (Grant No. 61070092).

References

1. Armbrust M, Fox A, Grith R, Joseph AD, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I, Za-haria M (2009) A view of Cloud computing. Commun ACM 53(4):50–58

2. Foster I, Zhao Y, Raicu I, Lu S (2008) Cloud Computing and Grid Computing 360-Degree Compared, Grid Computing Environments Workshop, GCE '08, Austin, pp 1–10
3. Uhlig R, Neiger G, Rogers D et al (2005) Inter virtualization technology, IEEE Computer Society. IEEE press, USA, pp 48–56
4. Barham P, Dragovic B, Fraser K, Hand S, Harris T, Ho A, Neugebauer R, Pratt I, Warfield A (2003) Xen and the art of virtualization. In: Proceeding SOSP '03 Proceedings of the nineteenth ACM symposium on operating systems principles. ACM, New York, pp 164–177
5. Kaplan J, Forrest W, Kindler N (2009) Revolutionizing data center energy efficiency. McKinsey
6. http://searchstorage.techtarget.com.au/articles/28102-Predictions-2-9-Symantec-s-Craig-Scroggie
7. ASHRAE Technical COmmittee 99 (2005) Datacom equipment power trends and cooling applications
8. Belady C (2007) In the data center, power and cooling costs more than the equipment it supports. http://www.electronics-cooling.com/articles/2007/feb/a3/
9. Buyya R, Beloglazov A, Abawajy J (2010) Energy-efficient management of data center resources for Cloud computing : a vision, architectural elements, and open challenges. In: PDPTA 2010, proceedings of the 2010 international conference on parallel and distributed processing techniques and applications. CSREA Press, United States of America, pp 6–17
10. Calheiros RN, Ranjan R, Beloglazov A, Rose CAFD, 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
11. Beloglazov A, Buyya R (2011) 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 (CCPE). Wiley Press, New York
12. Nathuji R, Schwan K (2007) Virtualpower: coordinated power management in virtualized enterprise systems. ACM SIGOPS Oper Syst Rev 41(6):265C278
13. Stoess J, Lang C, Bellosa F (2007) Energy management for hypervisor-based virtual machines. In: Proceeding ATC'07 USENIX annual technical conference on proceedings of the USENIX annual technical conference
14. Kansal A, Zhao F, Liu J et al (2010) Virtual machine power metering and provisioning. In: Proceeding SoCC'10 proceedings of the 1st ACM symposium on Cloud computing. ACM, New York, pp 39–50
15. Oh FYK, Kim HS et al (2011) Enabling consolidation and scaling down to provide power management for Cloud computing. In: Proceeding HotCloud'11 proceedings of the 3rd USENIX conference on hot topics in Cloud computing, CA, pp 14–14
16. Verman A, Ahuja P, Neogi A (2008) pMapper: power and migration cost aware application placement in virtualized systems. In: Proceeding middleware '08 proceedings of the 9th ACM/IFIP/USENIX international conference on middleware. Springer, New York, pp 243–264
17. 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, San Diego, pp 10–10
18. Cardosa M, Korupolu MR, Singh A (2009) Shares and utilities based power consolidation in virtualized server environments. In: IFIP/IEEE international symposium on integrated network management, Long Island, pp 327–334
19. Gong ZH, Gu XH (2010) PAC: pattern-driven application consolidation for efficient Cloud computing. In: Proceeding MASCOTS' 10 proceedings of the 2010 IEEE international symposium on modeling, analysis and simulation of computer and telecommunication systems. IEEE Computer Society Washington, DC, pp 24–33
20. Goudarzi H, Pedram M (2012) Energy-efficient virtual machine replication and placement in a Cloud computing system. In: 2012 IEEE fifth international conference on Cloud computing, Honolulu, pp 750–757
21. Bila N, Lara ED et al (2012) Jettison: efficient idle desktop consolidation with partial VM migration. In: Proceeding EuroSys'12 proceedings of the 7th ACM European conference on computer systems. ACM, New York, pp 211–224
22. Xu J, Fortes JAB (2010) Multi-objective virtual machine placement in virtualized data center environments. In: 2010 IEEE/ACM international conference on green computing and communications and international conference on cyber, physical and social computing, Hangzhou, pp 179–188
23. Duy TVT, Sato Y, Inoguchi Y (2010) Performance evaluation of a green scheduling algorithm for energy savings in Cloud computing. In: 2010 IEEE international symposium on parallel and distributed processing, workshops and Phd forum (IPDPSW), Atlanta, pp 1–8

24. Feller E, Rilling L, Morin C (2011) Energy-aware ant colony based workload placement in Clouds. In: Proceeding GRID'11 proceedings of the 2011 IEEE/ACM 12th international conference on grid computing. IEEE Computer Society, Washington, DC, pp 26–33
25. Mills K, Filliben J, Dabrowski C (2011) Comparing VM-placement algorithms for on-demand Clouds. In: 2011 IEEE third international conference on Cloud computing technology and science (CloudCom), Athens, pp 91–98
26. Bobroff N, Kochut A, Beaty K (2007) Dynamic placement of virtual machines for managing SLA violations, IM '07. in: 10th IFIP/IEEE international symposium on integrated network management, Munich, pp 119–128
27. Borgetto D, Casanova H, Costa GD et al (2012) Energy-aware service allocation. Future Generation Computer Systems (FGCS). Elsevier press, Amsterdam 28(5):769–C779
28. Lovsz G, Niedermeier F, Meer HD (2012) Performance tradeoffs of energy-aware virtual machine consolidation. Cluster Computing
29. Dupont C, Schulze T, Giuliani G et al (2012) An energy aware framework for virtual machine placement in cloud federated data centres. In: e-Energy'12 proceedings of the 3rd international conference on future energy systems: where energy, computing and communication meet. ACM, New York
30. Beloglazov A, Abawajy J, Buyya R (2012) Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing. Future generation computer systems (FGCS). Elsevier Science, Amsterdam 28(5):755–768
31. SPECpower_ssj2008 Results. http://www.spec.org/power_ssj2008/results/
32. SPECpower_ssj2008 Results res2012q1. http://www.spec.org/power_ssj2008/results/res2012q1/
33. SPECpower_ssj2008 Results res2011q1. http://www.spec.org/power_ssj2008/results/res2011q1/
34. Cleveland WS, Loader C (1996) Smoothing by local regression: principles and methods. Stat Theory Comput Asp Smooth 1049:10C49
35. Park KS, Pai VS (2006) CoMon: a mostly-scalable monitoring system for PlanetLab. ACM SIGOPS Oper Syst Rev 40(1):74
36. Amazon EC2 Instance Types. http://aws.amazon.com/ec2/instance-types/
37. The Corrected bug in CloudSim. http://code.google.com/p/cloudsim/source/browse/trunk/modules/cloudsim/src/main/java/org/cloudbus/cloudsim/power/PowerVmAllocationPolicyMigrationAbstract.java