The Journal of Supercomputing (2019) 75:2126–2147
https://doi.org/10.1007/s11227-018-2709-7

A reliable energy-aware approach for dynamic virtual machine consolidation in cloud data centers

Monireh H. Sayadnavard[1] · **Abolfazl Toroghi Haghighat**[2] · **Amir Masoud Rahmani**[1]

Published online: 4 December 2018

Abstract

To achieve energy efficiency in data centers, dynamic virtual machine (VM) consolidation as a key technique has become increasingly important nowadays due to the significant amounts of power needed to operate these data centers. Most of the existing works on VM consolidation have been focused only on reducing the number of active physical machines (PMs) using VM live migration to prevent inefficient usage of resources. But on the other hand, high frequency of VM consolidation has a negative effect on the system reliability. Indeed, there is a crucial trade-off between reliability and energy efficiency, and to optimize the relationship between these two metrics, further research is needed. Therefore, in this paper a novel approach is proposed that considers the reliability of each PM along with reducing the number of active PMs simultaneously. To determine the reliability of PMs, a Markov chain model is designed, and then, PMs have prioritized based on their CPU utilization level and the reliability status. In each phase of the consolidation process, a new algorithm is proposed. A target PM selection criterion is also presented that by considering both energy consumption and reliability selects the appropriate PM. We have validated the effectiveness of our proposed approach by conducting a performance evaluation study using CloudSim toolkit. The simulation results show that the proposed approach can significantly improve energy efficiency, avoid inefficient VM migrations and reduce SLA violations.

✉ Abolfazl Toroghi Haghighat
at_haghighat@yahoo.com; haghighat@qiau.ac.ir

Monireh H. Sayadnavard
mo.sayadnavard@gmail.com

Amir Masoud Rahmani
rahmani@srbiau.ac.ir

[1] Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

[2] Faculty of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

Keywords Cloud computing · VM consolidation · Live migration · Energy efficiency · Reliability · Markov chain

1 Introduction

Cloud computing has emerged in recent years as one of the most interesting developments in technology, and it is changing the way we access and retrieve information [1]. The ever-increasing demand for computing resources has resulted in the establishment of large-scale data centers, which require enormous amount of power and hence consume a lot of energy. Statistics of the worldwide data center electricity consumption show nonlinear growth during the last decade, and a similar trend is expected for the upcoming years [2]. The amount of computing resources and the inefficient use of these resources could lead to huge energy wastage. An effective way to improve the resource utilization and energy efficiency in cloud data centers is VM consolidation that has been widely studied in recent years [3, 4]. During the consolidation process, VMs are periodically reallocated using live migration according to their current resource demand to minimize the number of active physical servers and the idle PMs are switched to low-power modes to reduce the energy consumption [5].

Since most modern applications experience dynamic patterns of resource consumption because of highly variable workloads, VM consolidation in clouds is a complicated operation. Unconstrained VM consolidation may lead to degraded performance when an application is faced with increasing demand and resource usage. If the resources requirements of an application are not met, the response time will increase. The QoS guarantee defined in the service-level agreements (SLAs) between the cloud provider and their users is essential. Hence, the cloud providers must consider the trade-off between performance and energy consumption in order to fulfill QoS requirements [6].

On the other hand, high VM consolidation has a negative effect on the reliability of the system [4, 7, 8]. Most of the existing researches on consolidation have focused on the performance–energy trade-off. There are some works that consider the relationship between system reliability and energy efficiency in cloud environment [8, 9], and still, there is a distinct need for more research on the mentioned challenge. In fact, there is a critical trade-off between reliability and energy efficiency.

Server consolidation may increase the probability of server failure and compromise the reliability of the system by increasing the load on some servers and shutting down some of them. It is noteworthy that increasing resource redundancy can improve the reliability of the system, but it is in contradiction to the concept of consolidation that leads to reduce the number of active servers. Hence, we need to consolidate servers in a flexible manner with considering both energy efficiency and reliability to cover different operating conditions and scenarios.

In this paper, we present a novel approach to dynamic VM consolidation by considering both reliability and energy efficiency simultaneously. We try to manage energy consumption along with considering the reliability of each PM in every phase of consolidation to reach equilibrium between these two metrics. We have calculated the reliability under the probability of failures occurrence in a heterogeneous environment. A computational model for PM reliability prediction is presented, and based on the results of this phase and the CPU utilization level of each PM, they are divided into different categories. Then, we can make a proper decision to VM migration to the realization of our purpose. Specifically, the major contributions of this paper can be summarized as follows:

- Propose a framework for predicting and analyzing PM reliability using a Markov chain model for VM consolidation implementation to optimize the relationship between energy efficiency and reliability.
- Design a technique to dynamic VM consolidation which considers two important metrics including the reliability of PMs and their CPU utilization to improve energy efficiency and keep the reliability of the system at a desirable level.
- Perform experimental evaluation to validate the effectiveness of proposed approach utilizing the CloudSim as simulation framework.

The remainder of the paper is organized as follows: The current and past researches on VM consolidation are reviewed in Sect. 2. The system model and problem statement are explained in Sect. 3. Our Markov chain-based reliability model and the proposed approach for dynamic VM consolidation are described in detail in Sect. 4. The experimental setup and results are shown and discussed in Sect. 5. Finally, conclusions are presented in the last section.

2 Related works

This paper presents an approach for dynamic VM consolidation in data centers, considering optimizing the relationship of energy efficiency and system reliability to provide energy-efficient and highly reliable computing environments. There are several research works that address the VM consolidation. In this section, we review relevant works in the literature related to the similar issues.

Many studies formulated the VM consolidation as a well-known NP-hard bin packing problem [6, 10–13]. Various heuristics like greedy algorithms are utilized to approximate the optimal solution of this NP-hard problem. These include worst fit and best fit in [10], first-fit decreasing (FFD) and best-fit decreasing (BFD) [11]. The authors in [12] have divided VM consolidation into the four following phases: host overload detection, selection of VMs that should be migrated, VM placement and running PM shrinking. Due to the complexity of VM consolidation, the VM consolidation issues in [12] were separated into several subproblems, and then, they have proposed the novel adaptive heuristics for each subproblem. They proposed the modified best-fit decreasing (MBFD) algorithm to VM placement by considering power consumption and SLA violation. In this algorithm, the VMs are first sorted in decreasing order based on their utilization. Then, these VMs are allocated to the hosts having a minimum increase in energy consumption.

In [13], a VM consolidation framework is proposed to minimize the performance–energy trade-off. The VM placement problem is resolved using semi-online multi-dimensional bin packing. The authors in [14] have considered rack, cooling structure and network topology when consolidating VMs. In this paper, the MBFD algorithm is improved, and then, three structure-aware VM placement methods are proposed to consolidate VMs in the servers to minimize the number of active racks that results in turning off idle routing and cooling equipment in order to reduce the energy consumption. In [15], a burstiness-aware server consolidation algorithm, QUEUE, is proposed. First, the burstiness of workload is captured using a two-state Markov chain; then, some extra resources on each PM are reserved to avoid live migrations.

In [16], DVFS-aware consolidation procedure is presented to eliminate the inconsistencies between consolidation and DVFS techniques. This paper also has proposed PSFWT as a fuzzy DVFS-aware multi-criteria and objective resource allocation solution for VM placement in cloud data centers that simultaneously optimize important objectives including energy consumption, SLA violation and the number of VM migrations. Beloglazov and Buyya [5] investigated the problem of overloaded host detection using a Markov chain model. A specified QoS goal is defined to maximizing the mean time between VM migrations for any known stationary workload. The unknown non-stationary workloads are also handled using a multi-size sliding window workload estimation. In [17], a heuristics-based multi-phase approach for server consolidation is proposed which effectively reduces residual resource fragmentation along with reducing the number of active PMs. Residual resource fragmentation refers to the state where a sufficient amount of residual resources is available but are fragmented and distributed across multiple active PMs.

The authors in [18] studied the influence of four aspects on energy consumption and QoS, namely the dynamic workload, CPU utilization, times of VM migrations and opportunity of VM migration from nine related factors. They created a Bayesian network-based estimation model (BNEM) for dynamic VM migration using these factors that each node represents one aspect of VM migration. Khani et al. [19] proposed a distributed mechanism for dynamic consolidation of virtual machines using a non-cooperative game for reducing power consumption in data centers with heterogeneous PMs.

There are also various metaheuristic algorithms that have been proposed to solve the VM consolidation problem in cloud computing environments. These algorithms rely on a probabilistic approach to find near optimal solutions to the problems. In [20], ant colony optimization (ACO) method is used to pack the VMs into the least number of physical machines while preserving quality of service requirements. A multi-objective function is defined that considers both the number of dormant PMs and the number of migrations. The GABA approach [21] is a genetic algorithm (GA)-based algorithm that dynamically finds the optimum reconfiguration for a set of VMs according to the predicted future demand of the running workload. The algorithm decreases the number of PM significantly and converges within reasonable time. In [22], a VM consolidation approach is proposed based on the particle swarm optimization (PSO) algorithm, which considered reducing energy consumption and improving resource utilization in the data center as the optimization objective.

Deng et al. in [8] presented a Reliability-Aware server Consolidation stratEgy (RACE) to address a multi-objective problem with considering hardware reliability and energy efficiency. A utility model has been formulated that uses three parameters U_{SLA}, U_r and U_e to determine the best VM-to-PM mapping. U_{SLA} ensures that there are enough resources to support the SLA, U_r value shows the impacts of turning servers on and off and temperature variation on reliability and lifetime, and U_e estimates the amount of power usage reduction. Finally, the mapping that has the maximum value of the sum of these three parameters is chosen to provide an optimized solution of the problem.

There are also some works that considered energy efficiency and reliability in cloud computing at the same time that a review of them has been provided in [7]. But most of these works have not specifically addressed the issue of VM consolidation and focus on resource allocation in a reliable and energy-efficient manner.

However, to the best of our knowledge, this is the first paper that provides a reliability model of PMs to use in consolidation process with the aim of saving unnecessary wastage of energy that will be required to restart all the running process that were interrupted during the failure.

3 System model and problem statement

We consider a system that consists of a single data center with heterogeneous resources as the scope of our work is restricted to migrations within a data center. Let $\mathrm{PM} = \{\mathrm{PM}_1, \mathrm{PM}_2, \ldots, \mathrm{PM}_i, \ldots, \mathrm{PM}_m\}$ be the set of active PMs in the current state of the data center and $\mathrm{VM}_i = \{\mathrm{vm}_1, \mathrm{vm}_2, \ldots, \mathrm{vm}_j, \ldots, \mathrm{vm}_n\}$ be the set of deployed VMs that in $\mathrm{PM_i}$. Each PM is characterized by the CPU performance defined in millions of instructions per second (MIPS), amount of RAM, network bandwidth and disk storage. But the disk storage space in any PM is usually large, and dynamic variations in disk space requirements are usually not observed. Hence, it can be safely neglected.

At any given time, a cloud data center usually serves many simultaneous users. Users submit their requests for provisioning n heterogeneous VMs, which are allocated to the PMs and characterized by requirements of resources. The length of each request is specified in millions of instructions (MI). It is assumed that each of the n VMs is initially placed in a random PM in the data center based on its resource requirements. The problem is to minimize the number of PMs used, by maximizing the resource utilization in each PM using live migration of VMs so that the freed PMs can be set to a power saving state. To reduce energy consumption without compromising reliability, we can abstract the VM consolidation problem as the following.

$$\text{minimize}: \qquad E_{\text{total}} = \sum_{j=1}^{n} E_{\mathrm{PM}_j} \tag{1}$$

$$\text{maximize}: \qquad R_{\text{sys}}\,(t) = \prod_{j=1}^{n} R_{\mathrm{PM}_j}(t) \tag{2}$$

Subject to

$$\sum_{j=1}^{n} x_{ij} = 1\ .x_{ij} = 0 \text{ or } 1\ .i = [1.2\ \ldots\ .m] \tag{3}$$

$$\sum_{i=1}^{m} l_i^{\mathrm{CPU}} x_{ij} < C_j^{\mathrm{CPU}} \cap \sum_{i=1}^{m} l_i^{\mathrm{mem}} x_{ij} < C_j^{\mathrm{mem}} \cap \sum_{i=1}^{m} l_i^{\mathrm{bw}} x_{ij} < C_j^{\mathrm{bw}} \tag{4}$$

The first objective is to minimize energy consumption. E_{total} is the summation of energy consumption of all PMs. Equation (2) represents the second objective related to the reliability of the system. $R_{\mathrm{sys}}\,(t)$ is the reliability of system and $R_{\mathrm{PM}_j}(t)$ is the reliability of jth PM. Constraint presented in Eq. (3) assigns each VM to only one PM. Constraint described in Eq. (4) ensures that the allocated resources do not exceed the total available resource in that particular PM.

4 Proposed approach

The framework of our proposed approach to resolving the aforementioned issues, as shown in Fig. 1, consists of two main components: prediction module and decision-making unit. Prediction module observes energy consumption caused by VMs and PMs and collects historical data of past failures that can be utilized in a Markov chain-based prediction model described in the following subsection. The module is executed on each PM locally and sends obtained information to decision-making unit. Decision-making unit manages resource allocation and VM placement on PMs in the data center. According to the received PM messages and state analysis, this unit determines each PM belongs to one of the critical, optimal and suboptimal categories. Then, VM selection and target PM selection algorithms are carried out and appropriate decisions are made to solve the consolidation problem.

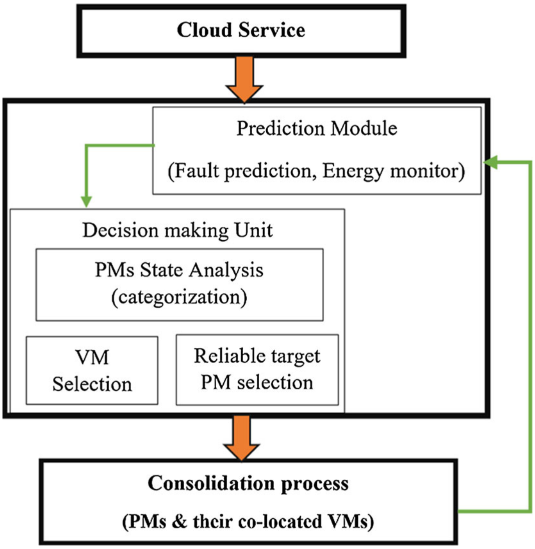


Fig. 1 Proposed framework

4.1 Markov chain-based prediction model

Markov chain model is the most fundamental and general state-based stochastic method that concerns about a sequence of random variables, which correspond to the states of a system, in such a way that the state at one time epoch depends only on the one in the previous time epoch [23].

Markov chains are usually classified into two categories: discrete-time Markov chain (DTMC) and continuous-time Markov chain (CTMC). CTMC, semi-Markov process and stochastic Petri net (SPN) have been used widely for evaluating the performance [24], reliability/availability [25] and performability [26] of computer systems. In this paper, we choose the CTMC model to develop a prediction mechanism to analysis PM reliability. Since the exponential random variable is the only continuous random variable with Markov property and hardware and software faults are commonly modeled as exponential distribution, we assume that the time to transit from a system state to another due to failures and recovery follows an exponential distribution. Reason for its use includes its memoryless (Markov) property and its relation to passion distribution. Thus, the random variables such as time to failure of a component and time required to repair a component will often be modeled as exponential [24]. Figure 2 shows the CTMC model state transition diagram for the probabilistic reliability behavior of each PM in data center. Although in many works only two active and failed states are considered for a host, there are some factors that result in performance degradation.

In this study, we consider that hypervisor or virtual machine monitor (VMM) is affected by software aging. One of the common ways to deal with this problem is software rejuvenation as a proactive fault management technique to prevent or postpone failures in VMMs and VMs. Migrate VM rejuvenation [27] is an effective technique for VMM rejuvenation. In this technique, before triggering the VMM rejuvenation, running VMs are migrated to another host, and then, VMM rejuvenation starts. If we choose a PM with aged VMM as a VM migration destination in the consolidation process, it leads to an increase the number of migrations and waste energy. Therefore, in order to model the reliability of PMs, in addition to hardware failures, VMM failures are also considered which can be extended to other software failures.

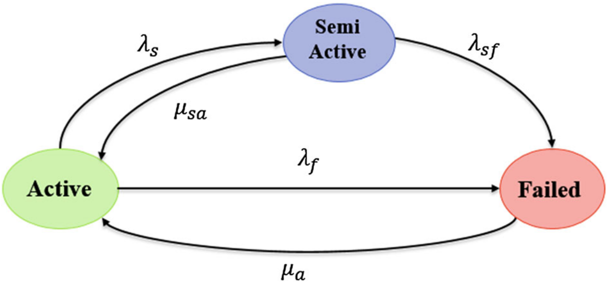


Fig. 2 State transition diagram

As depicted in Fig. 2, the model consists of three states including active, semi-active and failed. Let X(t) with discrete state space S = {A, SA, F} represents the state of PM at time t.

$$X(t) = \{X_A(t),\ X_{\mathrm{SA}}(t),\ X_F(t)\} \tag{5}$$

if $X(t) = X_A(t)$, PM is in proper condition and active state. In other two states that are inactive states, we consider hypervisor failure and hardware failures. Hardware failures are critical and lead PMs to the failed state. Semi-active state is about VMM rejuvenation process during which the PM is not available.

We define μ and λ as the repair rate and the failure rate of a PM, respectively. With these assumptions, the transient process $X(t)$ can be modeled mathematically as a homogeneous CTMC on the state space S. For each time $t > 0$, the probability of a PM in state i is given by $X_i(t) = \Pr\{X(t) = i\}$, $i \in S$. The Markov process is defined by generator Q whose is given by:

$$Q = \begin{bmatrix} -\lambda_s - \lambda_f & \lambda_s & \lambda_f \\ \mu_{sa} & -\mu_{sa} - \lambda_{sf} & \lambda_{sf} \\ \mu_a & 0 & -\mu_a \end{bmatrix} \tag{6}$$

where

$$q_{ij} = \begin{cases} \lim\limits_{\mathrm{d}t \to 0} \left(\frac{\Pr(\text{going from } i \text{ to } j(t.t + \mathrm{d}t))}{\mathrm{d}t} \right) .i \neq j \\ -\sum\limits_{j \neq i} q_{ij} \qquad i = j \end{cases} \tag{7}$$

State probability vector $X(t)$ is a function of time, and to describe the dynamic behavior of the CTMC, the Kolmogorov differential equation can be defined using the Q matrix as:

$$\frac{\mathrm{d}X(t)}{\mathrm{d}t} = X(t)Q \tag{8}$$

So the transient state probability vector can be obtained as follows, where $X(0)$ is initial probability vector:

$$X(t) = X(0)\mathrm{e}^{Qt} \tag{9}$$

$$\mathrm{e}^{Qt} = \sum_{i=0}^{\infty} \frac{(Qt)^i}{i!} \tag{10}$$

Here we use uniformization that is a simple and efficient method to compute the transient probability vector $X(t)$ [28]. Let U be the transition probability matrix and I be the unit matrix; then, the transient state probability vector can be computed as:

$$U = I + \frac{Q}{\gamma} \ \rightarrow\ Q = \gamma(U - I) \tag{11}$$

$$X(t) = \sum_{i=0}^{\infty} \mathrm{e}^{-\gamma t} \frac{\gamma t^{i}}{i!} \pi U^{i} = \sum_{i=0}^{\infty} \mathrm{e}^{-\gamma t} \frac{\gamma t^{i}}{i!} \hat{\pi}(i) \tag{12}$$

where $\gamma \geq \max_i |q_{ii}|$ is uniform rate parameter. Poisson probability estimates are also needed to determine where the summation in Eq. (12) should be truncated to meet a desired accuracy error bounded by ε that is computed using:

$$\varepsilon \leq 1 - \mathrm{e}^{-\gamma t} \sum_{i=0}^{k} (\gamma t)^{i} / i! \tag{13}$$

After uniformization and obtaining transient state probabilities, two metrics that we need for next steps are also computed: occupation time and mean first passage time(mfpt). Indeed, these metrics, along with the PM states, are the outputs of the prediction module that are used in VM consolidation process.

4.1.1 Occupation time

The times spent by the PM in each of states {active, semi-active and failed}during a finite interval of time can be determined using occupation time of the CTMC [29]. Let $W_{i.j}(t)$ be the expected amount of time the CTMC spends in state j during the interval [0, t], starting in state i, and $u_{i.j}(t)$ be the element of the transition probability matrix U. The quantity $W_{i.j}(t)$ is called the occupation time of state j starting from state i, which is computed as

$$W_{i.j}(t) = \int_{0}^{t} u_{i.j}(t) \mathrm{d}t \tag{14}$$

4.1.2 First passage time

Let τ_j be the expected value of random time to reach state j (for the first time), given that it started in state i. These are sometimes referred to as mean first passage time. The first passage time into state$N$ is defined to be

$$T = \min\{n \geq 0 : X(t) = N\} \tag{15}$$

where $\{1, \ldots, N\}$ represent the state space. The expected value $E(T)$ is defined as

$$\tau_j = E((T | X_0 = i)) \tag{16}$$

According to a theorem defined in [28], the expected first passage time satisfies the following relation,

$$r_i \tau_i = 1 + \sum_{j=1}^{N-1} r_{i.j} \tau_j \quad .1 \leq i \leq N-1 \tag{17}$$

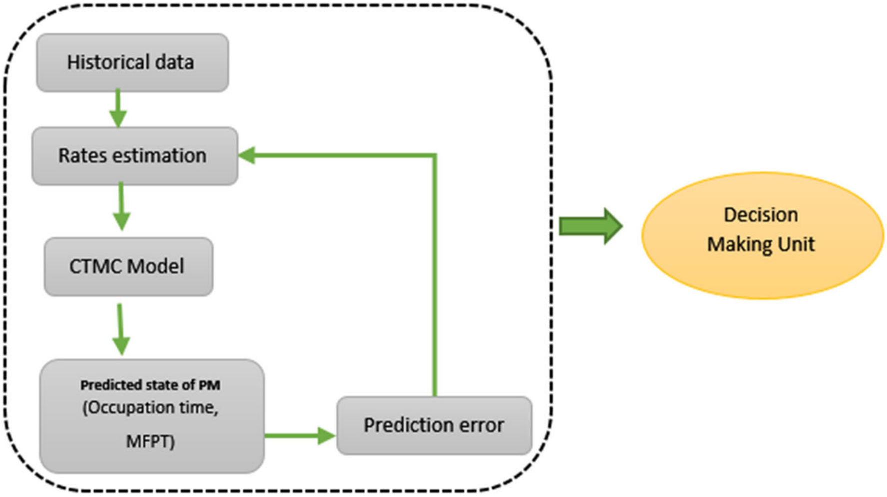


Fig. 3 Prediction steps

where $r_i = \sum_{j=1}^{N} r_{i.j}$ and $r_{i.j}$ is the entry of rate matrix R,

$$R = \begin{bmatrix} 0 & \lambda_s & \lambda_f \\ \mu_{sa} & 0 & \lambda_{sf} \\ \mu_a & 0 & 0 \end{bmatrix} \tag{18}$$

To summarize, we can describe our CTMC-based prediction models and its algorithmic steps as shown in Fig. 3. According to the flowchart, historical data and past failures can be utilized to estimate the λ and μ. Then, the generator matrix is constructed based on the estimated rates and the CTMC transition diagram. In the next step, transient state analysis performs to predict the state of PM and also compute the defined metrics values including expected time for the first occurrence of failure and occupation time. The difference between the predicted and actual values can be used to train and modify the transition rates. Then, the obtained results are sent to decision-making unit and used to classify the PMs for the consolidation process.

4.2 VM consolidation process

After performing the prediction phase, PM status is sent to the decision-making unit. To determine whether the host is overloaded, we apply local regression (LR) method proposed by Beloglazov et al. [30]. Under-loaded PMs can be found by comparing the CPU utilization with a low threshold. Other PMs are considered as well utilized. According to the obtained results from previous steps, each PM will be in one of the six sets: WR, OR, UR, WU, OU and UU. These sets represent the well utilized and reliable, overloaded and reliable, under-loaded and reliable, well utilized and unreliable, overloaded and unreliable, and under-loaded and unreliable PMs, respectively.

Unreliable state is related to semi-active and fail states in the Markov model. Then, these sets are divided into critical, optimal and suboptimal categories.

To select the migration source, the categories whose PMs are in the critical situation are candidate. PMs in OU set have the highest priority. PMs of critical and suboptimal categories are sorted based on MFPT in ascending order. The pseudocode of the PM categorization algorithm is given as algorithm 1. At the end of this phase, the potential source of PMs is determined. It should be noted that if all the PMs are in WR set, no migration will be done.

4.2.1 VM selection

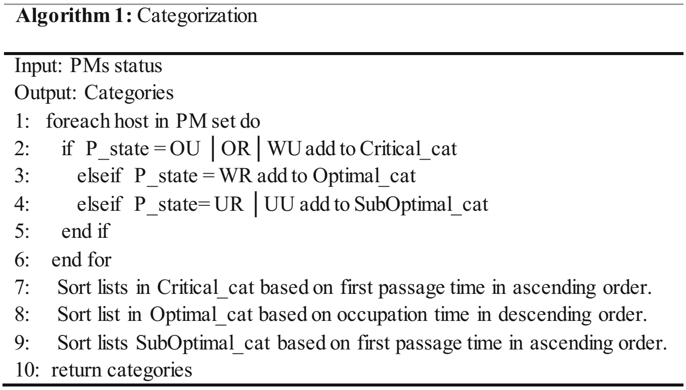
Since more than one virtual machine runs on a host, we need to consider how to choose a VM to migrate to improve QoS and prevention of SLA violation. Algorithm 2 represents the pseudocode of our proposed selection procedure running on the PMs of the critical category. It should be noted that this procedure only runs on OR list. If there was some PM on the other two lists in this category, we try to migrate all of their colocated VMs because of the unreliable state of the PMs. To select the VMs to be migrated in OR list, we use a hybrid heuristic based on CPU usage and memory. Initially, VMs are sorted in ascending order according to their CPU usage. Then, the VM list is sorted once again in ascending order based on memory usage. Afterward, a weight of i is assigned to the VM in the ith position in every sorted list. Indeed, the VM with the highest CPU usage has the heaviest weight and the VM with the lowest memory usage has the lightest weight. Eventually, we use these weights and make a list of VMs according to the CPU usage weight (w_u) to the memory usage weight (w_m) ratio, in descending order. To assign scores to VMs, a selection criterion is defined as follows,

$$\mathrm{SC}_i = \frac{w_{ui}}{w_{mi}} \tag{19}$$

Based on the presented selection criterion of VMs to be migrated, the selection algorithm is performed. This criterion is calculated for all VMs in the list; then, the VM with the highest score is selected to migration and added to the migration list.

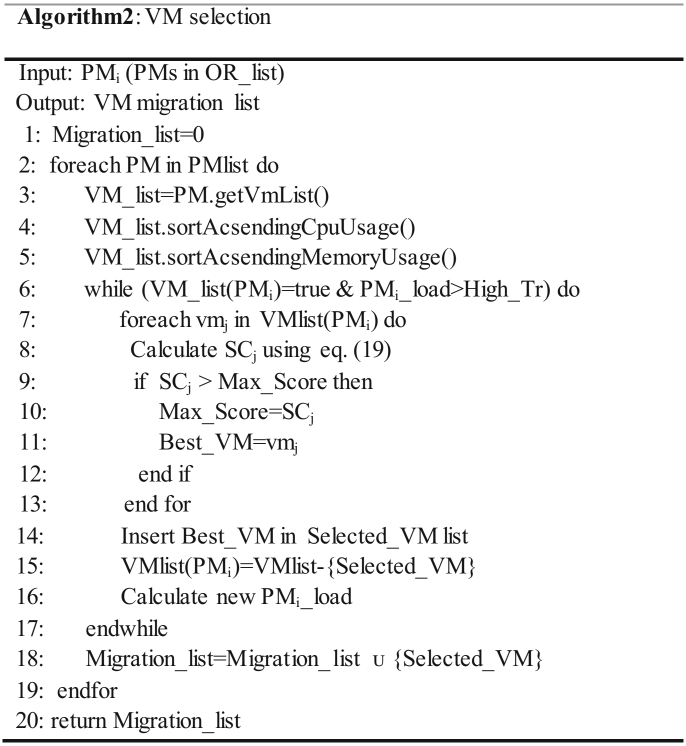
4.2.2 Target selection

After the VMs to be migrated are acquired, we need a policy to select the appropriate target for migrations. When a PM is selected as the destination of VM migration, its state likely changes due to increasing workload and resource usage. Therefore, the proposed algorithm in this phase tries to find a proper host with sufficient residual capacity and considering energy consumption and reliability. In this way, PMs in WR list of the optimal category are explored at first .



Algorithm 1: Categorization

```
Input: PMs status
Output: Categories
foreach host in PM set do
    if P_state = OU | OR | WU add to Critical_cat
        elseif P_state = WR add to Optimal_cat
        elseif P_state= UR | UU add to SubOptimal_cat
    end if
end for
Sort lists in Critical_cat based on first passage time in ascending order.
Sort list in Optimal_cat based on occupation time in descending order.
Sort lists SubOptimal_cat based on first passage time in ascending order.
return categories
```



Algorithm2: VM selection

```
Input: $PM_i$ (PMs in OR_list)
Output: VM migration list
Migration_list=0
foreach PM in PMlist do
    VM_list=PM.getVmList()
    VM_list.sortAcsendingCpuUsage()
    VM_list.sortAcsendingMemoryUsage()
    while (VM_list($PM_i$)=true & $PM_i$_load>High_Tr) do
        foreach $vm_j$ in VMlist($PM_i$) do
            Calculate $SC_j$ using eq. (19)
            if $SC_j$ > Max_Score then
                Max_Score=$SC_j$
                Best_VM=$vm_j$
            end if
        end for
        Insert Best_VM in Selected_VM list
        VMlist($PM_i$)=VMlist-{Selected_VM}
        Calculate new $PM_i$_load
    endwhile
    Migration_list=Migration_list ∪ {Selected_VM}
endfor
return Migration_list
```

If the algorithm fails to find adequate PM, the search process continues in underutilized and reliable PM within the suboptimal category. In our proposed policy, a VM will be migrated to a PM with the highest score that is estimated according to Eq. (20). To specify each PM score, energy cost and reliability are considered. Then, scores are determined using weight assignment to each criterion. α is an adjustable weight to obtain different trade-off points, since each cloud provider will pursue various objectives and business requirements. Indeed, the greater value of α demonstrates that the policy is more reliability biased, and the smaller value of α demonstrates more energy efficiency biased.

According to the proposed scoring method, VMs are assigned to more reliable PMs to prevent additional migrations. The idea behind this is that selecting safer PMs allows to migrate VMs to complete their works on the same server without any interruption or wasting time because of forced migration. Moreover, when we consider the reliability issue during target selection, probability of turning on sleep or off servers in long time is reduced.

$$\text{Score}_i = \alpha \times \left(R_{\text{PM}_i} + \text{occu\_time}_{\text{PM}_i}\right) + (1 - \alpha) \times \left(\text{EC}_i^{\text{curr}} - \text{EC}_i^{\text{after}}\right) \tag{20}$$

in which R_{pm} denotes the reliability of the target server and $\text{occu\_time}_{\text{PM}_i}$ is the times spent by the PM_i in the reliable state. $\text{EC}_i^{\text{after}}$ and $\text{EC}_i^{\text{curr}}$ are the energy cost after and before the vm_i placement, respectively. Estimation of R_{pm} and EC parameters are illustrated in the next two subsections. In order to allocate VMs to PMs, VM migration list will be sorted according to their CPU capacity requirements in decreasing order. Then, the score of each PM in WR list, which is already sorted based on occupation time, is computed (see Algorithm 3).

Algorithm 3: Target PM selection Algorithm

```
Input: Migration_list, WR_list, UR_list
Output: MigrationSchedule
1: Sort Migration_list by resource requierments in desc order
2: foreach $vm_j$ in Migration_list do
3:   best_score = Min
4:   target_PM = Null
5:    foreach $PM_i$ in WR_list do
6:       Calculate $Score_i$ using eq. (20)
7:       if  $Score_i$  >  best_score then
8:          best_score = $Score_i$
9:          target_PM = $PM_i$
10:      endif
11:    end for
12:  if target_PM= Null then
13:    foreach $PM_i$ in UR_list do
13:       repeat steps 7-10
14:      calculate new $PM_i$_load
15:      if $PM_i$_load> LowTR then
16:          Add $PM_i$ to WR_list
17:       endif
18:     endfor
19:   endif
20:    MigrationSchedule.put($vm_j$, target_PM)
21: end for
22: return  MigrationSchedule
```

Reliability model Reliability is defined as an evaluation parameter to measure the system's ability to functioning correctly under certain conditions over a specified interval of time. Given a single component i of a system, its reliability $R_i(t)$ is derived from its failure rate λ as follows:

$$R_i(t) = \mathrm{e}^{-\lambda t} \tag{21}$$

Here the PM reliability is formulated in two steps by considering two important components of a PM, hardware and VMM. First, the reliability of hardware $R_{\mathrm{HW}_i}(t)$ is computed. Then, the VMM reliability $R_{\mathrm{VMM}_i}(t)$ is considered. By using the reliability of these two components, the reliability of a PM can be expressed as [31]:

$$R_{\mathrm{PM}_i}(t) = R_{\mathrm{HW}_i}(t)R_{\mathrm{VMM}_i}(t) = \mathrm{e}^{-(\lambda_{\mathrm{HW}_i}+\lambda_{\mathrm{VMM}_i})t} \tag{22}$$

Suppose PM failures are independent. Then, the overall system reliability is defined as:

$$R_{\mathrm{sys}}(t) = \prod_{i=1}^{m} R_{\mathrm{PM}_i}(t) \tag{23}$$

Energy cost model Power consumption in data centers is mostly determined by the CPU, memory, storage and network as the main power consumer [12]. However, CPU has the most effect and power consumption can be described by a linear relationship of CPU utilization [12, 32]. The following function defines power consumption:

$$P(u) = K \times P_{\max} + (1 - K) \times P_{\max} \times u \tag{24}$$

where $P_{\max}$ is the maximum power of a PM in the running state, k is the fraction of power consumed by an idle PM and u is the CPU utilization. In this model, $P_{\max}$ is set to 250 W, which is a usual value for modern servers and the value of coefficient k is set to 0.7 because on average an idle server consumes approximately 70% of the power consumed by the server running at the full CPU speed [12]. Since the CPU utilization changes over time, it is defined as a function of time. Therefore, the total energy consumption of each PM in a period of time $[t_1, t_2]$ can be obtained as:

$$E = \int_{t_1}^{t_2} P(u(t))\mathrm{d}t \tag{25}$$

4.2.3 Reducing the number of active PMs

After performing the aforementioned steps, we can safely shut down remaining underloaded PMs in UR and UU sets of suboptimal category. First, we attempt to migrate all VMs on the PMs of UU set because of unreliability. If the proper destination PMs was found to hosting the migrated VMs, then source PM is switched to sleep mode. Finally, UR set is explored to reduce the number of active PMs as much as possible.

Table 1 Virtual machine instances

VM type	CPU frequency (MIPS)	RAM (GB)
High-CPU medium instance	2500	0.85
Extra large instance	2000	3.75
Small instance	1000	1.7
Micro-instance	500	0.613

5 Performance evaluation

In this section, we explain the simulation environment settings and then evaluate the performance of our proposed approach based on obtained results.

5.1 Simulation environment setup

It is difficult to implement the consolidation algorithm on a real infrastructure for large-scale experiments, so in this study we have chosen CloudSim toolkit [33] as the simulation platform that is a modern simulation framework for cloud computing environments. The experiments simulate a data center comprised of 800 heterogeneous PMs, half of which are HP ProLiant ML110 G4 (Intel Xeon 3040 2 Cores 1860 MHz, 4 GB) servers and the other half are HP ProLiant ML110 G5 (Intel Xeon 3075 2 Cores 2260 MHz, 4 GB). VMs are supposed to correspond to Amazon EC2 instance types with the only exception that all the VMs contain single core, because of the fact that the workload data used for the simulations come from single-core VMs.

There are four types of VMs in the experiments: high-CPU medium instance, extra large instance, small instance and micro-instance. Table 1 shows the VM properties. After creating PM and VM instances on the CloudSim platform, the VMs are deployed to random PMs based on their resource requirements. After each round of VM consolidation, VM resource demands changes according to workload data. We assume high-TR and low-TR thresholds equal to 0.8 and 0.4, respectively. The parameter α is set to 0.4 in our experiments.

5.2 Workload model

In order to make the results of simulation more realistic, it is important to conduct experiments using workload traces from a real system. We have used data that provided as a part of the CoMon project, a monitoring infrastructure for PlanetLab [34]. In this project, the data on the CPU utilization are obtained every five minutes by more than a thousand virtual machines from servers located at more than 500 places around the world. We have chosen 10 different days from the workload traces gathered during March and April 2011, randomly. Table 2 shows the characteristics of the data for each day.

Table 2 Properties of PlanetLab trace (CPU utilization)

Date	Number of VMs	Mean (%)	SD (%)
03/03/2011	1052	12.31	6.68
06/03/2011	898	11.44	6.77
09/03/2011	1061	10.70	7.35
22/03/2011	1516	9.26	6.24
25/03/2011	1078	10.56	6.32
03/04/2011	1463	12.39	7.03
09/04/2011	1358	11.12	6.95
11/04/2011	1233	11.56	7.13
12/04/2011	1054	11.54	7.22
20/04/2011	1033	10.43	8.10

5.3 Performance evaluation metrics

Energy consumption and SLA violations are two important factors in VM consolidation problem. In order to reasonably evaluate the efficiency of our proposed approach, we adopt several metrics that were presented by Beloglazov et al. [30]. These metrics are included: service-level agreement violation time per active host (SLATAH), overall performance degradation caused by migrations (PDM), SLA violations (SLAV), energy consumption (EC), VM migrations and energy and SLA violations (ESV).

QoS requirements are commonly formalized in the form of SLAs, which can be determined in terms of such characteristics as minimum throughput or maximum response time delivered by the deployed system [30]. There are two basic metrics to depict an SLA violation: SLATAH and PDM. The SLATAH is defined as Eq. (26) and measures the percentage of time during which active hosts have experienced CPU utilization of 100%.

$$\text{SLATAH} = \frac{1}{n}\sum_{i=1}^{n}\frac{T_i^s}{T_i^a} \tag{26}$$

where n is the total number of physical machines, T_i^s is the total time of SLAV caused by the CPU resource overload of PM$_i$, T_i^a is the running time of PM$_i$. Another metric PDM is calculated as follows:

$$\text{PDM} = \frac{1}{m}\sum_{j=1}^{m}\frac{C_j^d}{C_j^r} \tag{27}$$

where m is the number of VMs, C_j^d is the unsatisfied CPU required capacity caused by the migration of vm$_j$ and C_j^r is the CPU capacity requested by vm$_j$. SLAV is a

Table 3 Markov model parameters

Parameters	Values (1/h)	Mean time	Parameters	Values (1/h)	Mean time
λ_f	0.000114	1 year	μ_a	0.625	100 min
λ_s	0.00138	1 month	μ_{sa}	1	1 h
λ_{sf}	0.000114	1 year			

Table 4 Simulation results of different algorithms

Method	EC (KWh)	SLAV	ESV (%)	VM migrations
LR-MMT	162.53	0.48243	78.3394	32871
LR-MC	149.25	0.74538	111.2479	27095
LR-RS	147.88	0.78383	115.9127	26540
RE-VMC	118.71	0.13728	16.2965	9122

combined metric of two aforementioned metrics that evaluates a single-day QoS of the data center and is defined as:

$$\text{SLAV} = \text{SLATAH} \times \text{PDM} \tag{28}$$

ESV as described in Eq. (29) is a combined metric which consists of energy consumption of a data center per day and the level of SLA violations. Energy consumption and SLA violations are typically negatively correlated as energy can usually be decreased by the cost of the increased level of SLA violations [30]. A lower estimation of ESV indicates that energy saving is higher than the SLA violations.

$$\text{ESV} = \text{EC} \times \text{SLAV} \tag{29}$$

5.4 Simulation results and analysis

In this section, the results of our experiments are discussed. Since we use LR method to host overload detection, three traditional combination methods, LR-MMT, LR-MC and LR-RS [30], are selected to compare and evaluate our proposed approach. These methods apply PABFD algorithm [30] to target selection for migrated VMs. The safety parameter is set to 1.2 in experiments. The CTMC model parameter default values are chosen as shown in Table 3. These values are found in the literature [27, 31, 35].

Comparison between other methods and RE-VMC algorithm is shown in Table 4. The obtained results indicate that energy consumption is reduced by RE-VMC algorithm compared to LR-MMT, LR-MC and LR-RS, due to decreasing number of migrations and switching the under-loaded and unreliable PMs to sleep mode which leads to energy saving. In terms of SLAV, RE-VMC has optimal SLAV compared to others and LR-RS has the highest SLAV. According to the results, RE-VMC's SLAV is only 17.5% of LR-RS's SLAV. These results reveal that RE-VMC is better than the other algorithms in guaranteeing QoS. The ESV index in Table 4 indicates that

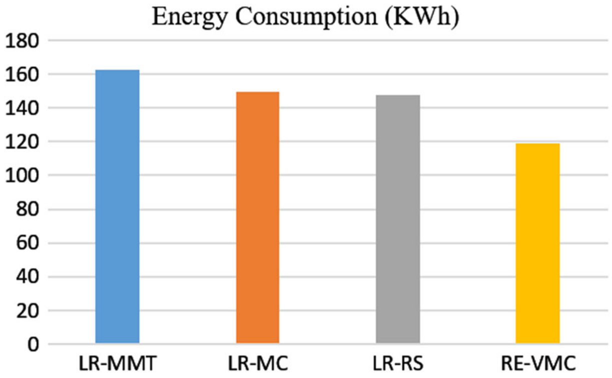


Fig. 4 Energy consumption of algorithms

the comprehensive performance of RE-VMC is considerably higher than others. The ESV of RE-VMC is only 20% of LR-MMT which has the closest value to RE-VMC, 14.6% of LR-MC and 14% of LR-RS. Eventually, the methods are compared in terms of the number of VM migration based on the experimental results. RE-VMC has the lowest number of VM migrations because it avoids additional migration by selecting the proper and reliable destination PMs.

Figure 4 shows the energy consumption of our proposed algorithm and three traditional VM consolidation algorithms that each of them consists of a VM selection policy and a VM placement policy. As can be seen, RE-VMC is better than other algorithms in terms of energy consumption. As mentioned before, minimizing the number of active physical servers along with reducing the VM migrations leads to decreasing energy consumption. In our proposed approach, we have considered these factors to improve the obtained results. However, taking into account reliability may initially lead to an increase in the number of migrations. But in the long run, finding the more reliable targets for VMs will reduce the number of migrations, and as a result, more energy will be saved.

Figure 5 shows the simulation results and comparison of algorithms in terms of SLATAH. It is completely obvious that RE-VMC outperforms the other methods and reduces PM overload risk. The reason is that the proposed approach considers reliability, which effectively leads to proper target PM selection. On the other hand, there is a correlation between the occurrence of failure and the system utilization, and with considering the reliability in consolidation steps, QoS of running PMs is maintained.

Figure 6 compares the RE-VMC and the other algorithms in terms of PDM. As depicted in this figure, RE-VMC has better performance. Prevention of extra migration effects on this parameter directly and migrating VMs to the safer PMs with considering failures and VMM rejuvenation reduce the number of VM migration. Indeed, according to obtained results in experiments, we can conclude that one of our objectives about decreasing VM migration has been achieved.

Figure 7 shows the number of migrations of the proposed algorithm and other algorithms. According to the results, the RE-VMC has a smaller number of migrations

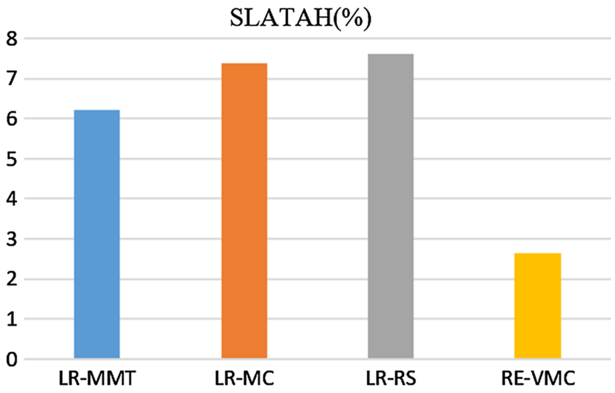


Fig. 5 Comparison of SLATAH

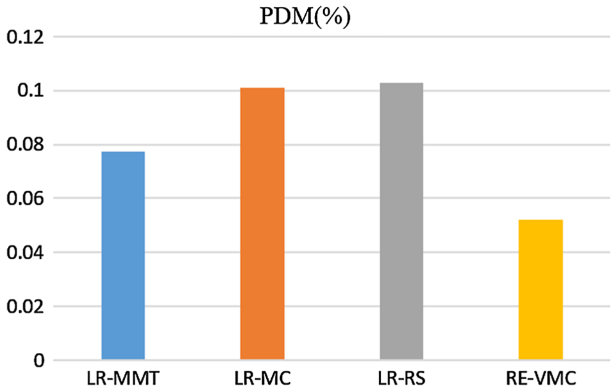


Fig. 6 Comparison of PDM

and outperforms LR-MMT, LR-MC and LR-RS significantly. The reason is that our proposed approach properly selects reliable servers as the destination of migrations and prevents unnecessary migrations by avoiding unprofitable and aggressive reconfigurations. When we consider reliability, the probability of VM migration, because of failure occurrence, is reduced. Therefore, while consuming a lower amount of energy, RE-VMC has a fewer number of migrations.

The variation in the number of VM migrations with respect to the cycle of the ongoing VM consolidation is shown in Fig. 8. The number of VM migrations triggered by the RE-VMC algorithm within the preliminary cycles is greater than other algorithms. The reason is that the proposed approach selects unreliable PMs in addition to overloaded PMs as VM migration source. Therefore, the number of migrations increases in the initial cycles. But it is obvious that after some cycles and by identifying appropriate PMs, VMs are placed on more reliable PMs with normal utilization. So, extra VM migrations are avoided and the number of VM migrations is significantly degraded.

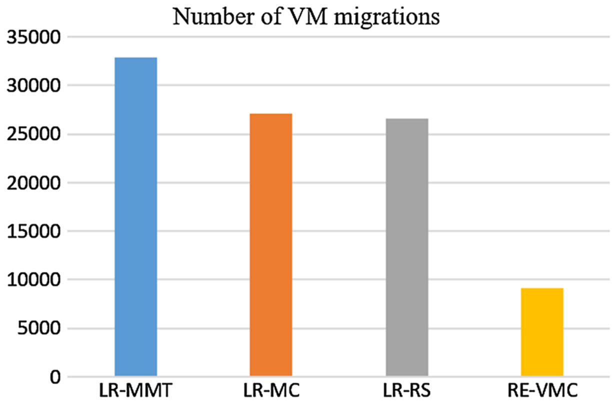


Fig. 7 Comparison of the number of VM migrations

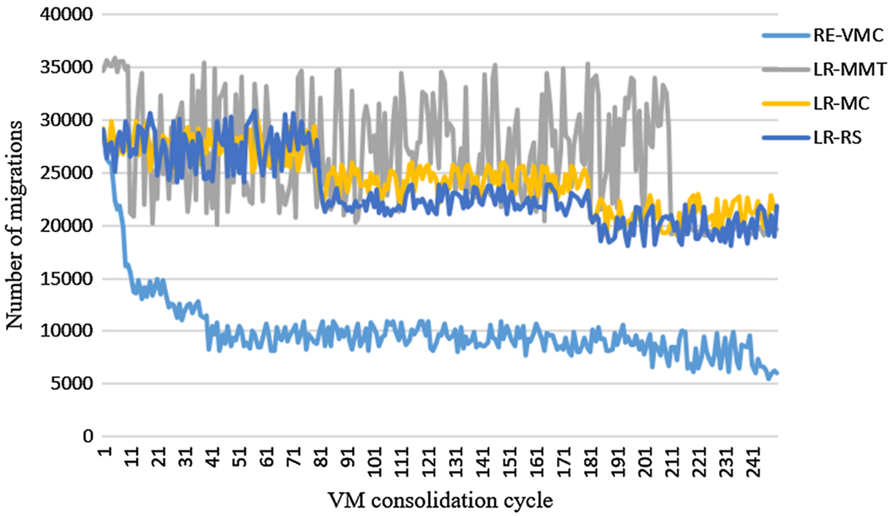


Fig. 8 Variation in the number of VM migrations

6 Conclusions

In this paper, we presented a reliability- and energy-aware framework to dynamic virtual machine consolidation in the cloud data centers. Energy consumption is one of the important issues in these data centers, and many researches have been done to optimize energy management using power control techniques in recent years. Although these methods are very efficient from the point of view of energy management, they ignore the negative impact on the reliability of the system. Frequent turning on or off resources or putting them in sleep mode tends to make them more susceptible to failure and result in increasing the overall response time, service delays and the SLA violation. Hence, we tried to address the challenge of energy management using dynamic consolidation by considering the energy efficiency and reliability and their trade-off in cloud computing.

First, we have introduced a Markov model for reliability estimation of PMs, and then, a classification based on the obtained results and CPU overload detection algorithm (LR) was made. Finally, we consider utilization along with the reliability in consolidation steps to select the source and destination PMs that leads to proper decision making and reduce number of migrations, energy consumption and consequently SLA violation. We have evaluated our proposed approach, and the simulation results show major improvements in RE-VMC in terms of SLATAH, PDM, SLAV, EC, ESV and the number of VM migrations.

As future work directions, further studies can be conducted to address the impact of failure overhead on energy consumption. To achieve better results, studying how to develop a new model for prediction of PM utilization and reliability based on intelligence algorithms is another interesting direction for the future work.

References

1. Armbrust M et al (2010) A view of cloud computing. Commun ACM 53(4):50–58
2. Mills M (2013) The cloud begins with coal. Big data, big networks, big infrastructure, and big power. An overview of the electricity used by the digital ecosystem, Technical report
3. Ahmad RW et al (2015) A survey on virtual machine migration and server consolidation frameworks for cloud data centers. J Netw Comput Appl 52:11–25
4. Varasteh A, Goudarzi M (2017) Server consolidation techniques in virtualized data centers: a survey. IEEE Syst J 11(2):772–783
5. Beloglazov A, Buyya R (2013) Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints. IEEE Trans Parallel Distrib Syst 24(7):1366–1379
6. Beloglazov A (2013) Energy-efficient management of virtual machines in data centers for cloud computing, PhD dissertation
7. Sharma Y et al (2016) Reliability and energy efficiency in cloud computing systems: survey and taxonomy. J Netw Comput Appl 74:66–85
8. Deng W et al (2014) Reliability-aware server consolidation for balancing energy-lifetime tradeoff in virtualized cloud datacenters. Int J Commun Syst 27(4):623–642
9. Varasteh A, Tashtarian F, Goudarzi M (2017) On reliability-aware server consolidation in cloud datacenters. arXiv:1709.00411
10. Grit L, et al (2006) Virtual machine hosting for networked clusters: Building the foundations for autonomic orchestration. In: Proceedings of the 2nd International Workshop on Virtualization Technology in Distributed Computing. IEEE Computer Society
11. Speitkamp B, Bichler M (2010) A mathematical programming approach for server consolidation problems in virtualized data centers. IEEE Trans Serv Comput 3(4):266–278
12. Beloglazov A, Abawajy J, Buyya R (2012) Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future Gener Comput Syst 28(5):755–768
13. Beloglazov A, Buyya R (2010) Energy efficient resource management in virtualized cloud data centers. In: Proceedings of the 2010 10th IEEE/ACM international conference on cluster, cloud and grid computing. IEEE Computer Society
14. Esfandiarpoor S, Pahlavan A, Goudarzi M (2015) Structure-aware online virtual machine consolidation for datacenter energy improvement in cloud computing. Comput Electr Eng 42:74–89
15. Zhang S et al (2016) Burstiness-aware resource reservation for server consolidation in computing clouds. IEEE Trans Parallel Distrib Syst 27(4):964–977
16. Arianyan E, Taheri H, Khoshdel V (2017) Novel fuzzy multi objective DVFS-aware consolidation heuristics for energy and SLA efficient resource management in cloud data centers. J Netw Comput Appl 78:43–61

17. Rao KS, Thilagam PS (2015) Heuristics based server consolidation with residual resource defragmentation in cloud data centers. Future Gener Comput Syst 50:87–98
18. Li Z et al (2017) Bayesian network-based virtual machines consolidation method. Future Gener Comput Syst 69:75–87
19. Khani H et al (2015) Distributed consolidation of virtual machines for power efficiency in heterogeneous cloud data centers. Comput Electr Eng 47:173–185
20. Farahnakian F et al (2015) Using ant colony system to consolidate VMs for green cloud computing. IEEE Trans Serv Comput 8(2):187–198
21. Mi H, et al (2010) Online self-reconfiguration with performance guarantee for energy-efficient large-scale cloud computing data centers. In: Services Computing (SCC), 2010 IEEE International Conference on. IEEE
22. Li H et al (2016) Energy-efficient migration and consolidation algorithm of virtual machines in data centers for cloud computing. Computing 98(3):303–317
23. Fuqua NB (2003) The applicability of Markov analysis methods to reliability, maintainability, and safety. START 2(10):1–8
24. Trivedi KS (2008) Probability and statistics with reliability, queuing and computer science applications. Wiley, New York
25. Goyal A, Lavenberg SS, Trivedi KS (1987) Probabilistic modeling of computer system availability. Ann Oper Res 8(1):285–306
26. Meyer JF (1982) Closed-form solutions of performability. IEEE Trans Comput 7:648–657
27. Machida F, Kim DS, Trivedi KS (2013) Modeling and analysis of software rejuvenation in a server virtualized system with live VM migration. Perform Eval 70(3):212–230
28. Ghosh JK (2012) Introduction to modeling and analysis of stochastic systems, by VG Kulkarni. Int Stat Rev 80(3):487
29. Sericola B (2000) Occupation times in Markov processes. Stoch Models 16(5):479–510
30. Beloglazov A, Buyya R (2012) Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. Concurr Comput Pract Exp 24(13):1397–1420
31. Wei B, Lin C, Kong X (2011) Dependability modeling and analysis for the virtual data center of cloud computing. In: High Performance Computing and Communications (HPCC), IEEE 13th International Conference on 2011. IEEE
32. Fan X, Weber W-D, Barroso LA (2007) Power provisioning for a warehouse-sized computer. In: ACM SIGARCH Computer Architecture News. ACM
33. Calheiros RN et al (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
34. Park K, Pai VS (2006) CoMon: a mostly-scalable monitoring system for PlanetLab. ACM SIGOPS Oper Syst Rev 40(1):65–74
35. Matos RdS et al (2012) Sensitivity analysis of server virtualized system availability. IEEE Trans Reliab 61(4):994–1006