J Supercomput (2020) 76:1903–1930
https://doi.org/10.1007/s11227-018-2431-5

A stochastic process-based server consolidation approach for dynamic workloads in cloud data centers

Hossein Monshizadeh Naeen[1] · **Esmaeil Zeinali**[1] · **Abolfazl Toroghi Haghighat**[1]

Published online: 1 June 2018

Abstract With the development of information technology, there is a need for computational works everywhere and every time. Thus, people should be able to carry out their heavy computations without having the burden of purchasing expensive hardware and software. Cloud computing is an attractive solution to such needs, but the high energy consumption of physical machines in a Cloud data center is a matter of great concern. Therefore, some of the low-loaded machines can be turned off or switched into low energy mode using server consolidation approaches. In this paper, a Stochastic Process-Based Dynamic Server Consolidation (SB-DSC) policy is developed to reduce the total cost of data centers while satisfying the required quality of service. A novel algorithm, which we call it Stochastic Process-Based BFD (SBBFD), is employed in SB-DSC policy to perform virtual machine placements over time. SBBFD overcomes most drawbacks of other algorithms proposed in the literature. The simulation results on real workload data show that SB-DSC leads to a noticeable reduction in total cost in terms of power consumption, SLA violations, number of mode switching and number of migrations.

Keywords Cloud computing · Dynamic server consolidation · Virtual machine placement · Stochastic workload analysis

✉ Esmaeil Zeinali
zeinali@qiau.ac.ir; zeinali_es@yahoo.com

Hossein Monshizadeh Naeen
monshizadeh@qiau.ac.ir

Abolfazl Toroghi Haghighat
haghighat@qiau.ac.ir

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

1 Introduction

Cloud computing is a relatively new paradigm in computing which uses different existing technologies and concepts such as data centers, internet, hardware virtualization [1]. Cloud provides a set of resources that can be shared to run applications [2]. It leverages virtualization as a key technology. A virtual machine monitor (VMM) is placed as a layer on the main hardware of each physical machine (PM), shielding it completely. Instead, it provides the complete instructions of that hardware or other hardware as an interface. As a result, it is possible to have multiple, and different virtual machines (VMs) run independently and concurrently on a PM [3]. Thus, VMs increase the flexibility of the center and responsiveness to requests due to the fact that every VM meets the needs of several users or customers. In addition, portability, also known as migration, has significantly increased through the use of this technology because VMMs could provide further separation between hardware and software. Thus, an environment can be completely transferred from a machine to another one [3].

In a data center, most running servers only operate on a small part of the total available resources [4]. It can lead to a scenario in data centers in which multiple underutilized servers take up more space and allocate more resources than can be justified by their workloads. This problem is called server sprawl. Virtualization and live migration can be used to dynamically consolidate to a limited number of PMs and switch idle ones off [5]. Increasing resource utilization and, thus, reducing the number of active PMs in data centers have considerable advantages such as saving energy and cost [6].

Research works related to the subject of this study can be divided into two categories [7]: (i) theoretical studies and (ii) practical implementations in Cloud management systems. It is focused on the first category of works in this study. Consolidation algorithms can be implemented in existing open source Cloud management platforms, such as OpenStack,[1] OpenNebula[2] and snooze.[3] There are some studies such as [7] and [8] that have worked on practical implementations.

In case of dynamic server consolidation, the algorithms called VM placement algorithms are used over time for determining the physical hosts of VMs. VM placement algorithms use the required resources of VMs and capacity of PMs as input to determine the proper host for each VM in order to minimize the total number of active PMs. In this regard, several effective and heuristic methods such as [9–11], evolutionary methods such as [12–14], and other methods such as [15,16] are presented.

Heuristic algorithms, which are the focus of this study, are proposed with the purpose of finding the proper physical host for each VM. The proper host is selected based on the policy of the system. There exist various policies such as energy-based, cost-based and QoS-based policies [17]. The type of policy differs from a Cloud provider to one another. There is an obvious contradiction between the objectives of some poli-

[1] The OpenStack Cloud platform. http://openstack.org/.

[2] OpenNebula Cloud manager. https://opennebula.org/.

[3] The Snooze Cloud manager. http://snooze.inria.fr/.

cies. For example, saving the energy consumption sometimes leads to ignoring the QoS or vice versa. The QoS requirements are expressed via service-level agreements (SLAs) that determine the required performance levels, such as minimal throughput and maximal response time or acceptable latency of the system [12]. Monitoring the activities of VMs continuously and using advanced policies for placement based on dynamic workloads are needed to use the resources efficiently and guarantee a certain degree of QoS for users [18,19]. On the other hand, due to the importance of energy in the world and its costs as well as its environmental effects, many solutions are presented for reducing the energy consumption. Therefore, for many Cloud providers, it is important to consider policies that multiple objectives are taken into consideration. For example, they aim at reducing total cost while satisfying QoS requirements. In addition to the electricity price due to power consumption by computational resources, there are other costs that Cloud provider should manage. Some of them are the costs incurred due to SLA violations, switching servers on and off over time [20], VM migrations, cooling systems [10], etc.

The present study provides a Stochastic Process-Based Dynamic Server Consolidation (SB-DSC) policy to minimize data center costs while satisfying QoS requirements. Since these kinds of optimization problems are NP-hard, heuristic algorithms are developed to address them [21]. Some costs that are ignored in many studies, such as switching cost, are considered in the proposed solutions of this paper. More importantly, unlike the previous works that usually make decisions based on the current workload or the predicted behavior of workloads, in this paper, we consider the loads on VMs as stochastic processes. The main contributions of this paper are:

- Analyzing heterogeneous dynamic workloads from stochastic processes perspective. For this purpose, CPU utilization by each VM is considered as a non-stationary process, and then we analyze the behavior of all the VMs residing on a physical node as a whole.
- Taking into account multiple cost factors such as energy, SLA violation, migration and switching cost in the proposed heuristics for the problem of dynamic server consolidation.
- Analytically showing that migrations happen due to overutilization of resources lead to extra energy consumption, and thus, it is important to consider the overload probability before VM placement.
- Proposing a novel VM placement algorithm namely SBBFD that works with the stochastic processes of dynamic workloads. This algorithm considers new parameters such as overload probability when consolidating VMs on servers.
- Extending the definition of some known concepts, such as overloaded and underloaded hosts, to match the stochastic process viewpoint of this paper. We use sliding window idea and also define a novel concept namely unstabled hosts to deal with the non-stationary characteristics of the workloads.
- Proposing a comprehensive metric for evaluating dynamic consolidation systems that not only considers the cost of energy consumption and SLA violations due to resource shortage and migrations but also computes the switching cost.

This paper continues by reviewing related works in Sect. 2. Section 3 describes the system model including data center model and explains about cost factors that this study aims at minimizing them. In Sect. 4 an analysis of stochastic workloads is presented and their effect on resource utilization of PMs is studied. Novel heuristics based on the stochastic process of CPU utilization are presented to solve dynamic server consolidation problem in Sect. 5. Section 6 describes the experimental design and setup using CloudSim simulator; then it presents evaluation metrics and analysis of results. Finally, in Sect. 7 a summary of the work and future possible directions of our research is presented.

2 Related works

Many heuristic methods such as [21–24] are presented that approximate optimal solution for dynamic consolidation of VMs. For example, Beloglazov and Buyya [21] presented an energy-aware placement algorithm called modified best fit decreasing (MBFD) for a Cloud with heterogeneous systems in which a VM is greedily allocated to a host with the minimum energy consumption increase caused by the allocation. They have merely considered energy cost while their proposed model can lead to switching physical hosts on and off frequently, resulting in high risks and depreciation of server hardware. Gao et al. [5] have studied the dynamic energy management by considering QoS and resource utilization in the system optimally to reduce the energy usage. They have presented a BFD-based heuristic providing a high-quality mapping of virtual machines on physical machines. In other work, Arianyan et al. [25] have presented a method in which different criteria such as residual resources, potential, bandwidth, RAM capacity and power consumption in servers are considered for selecting the destination of a virtual machine. According to their model, hosts are scored by using a fuzzy system based on the mentioned criteria and the machine with the best score is selected as the new place of a virtual machine. In [5] and [25], the costs of switching server modes are not considered, and decision-making for the place of virtual machines is only based on the current workload. The present study not only does consider the current workload, but it also considers the stochastic process of resource utilization on each VM to make a decision on VM consolidation

The authors in [20] have investigated the costs of switching server mode from on to off or to low power modes. They have presented an algorithm called LCP which uses a forecast window with the length of ω and predicts the future workload behavior. This work considers mode changing costs to find a relatively optimal solution for determining the number of required servers for data processing. Their proposed model operates on only one workload, requiring cluster computing technology. Thus, they used homogenous servers for their processing model while the present study works on several workloads in a heterogeneous environment.

Verma et al. [22] have focused on the problem of dynamic consolidation of VMs in heterogeneous environments as a continuous optimization problem. In their proposed method, VM placement is optimized and efficiency increases by considering the migration cost along the energy cost in every time period. Similar to [26], they have presented a heuristic method for bin-packing with bins of different sizes and costs. None of these two studies have considered the quality of service and SLAs establishment.

Some studies have used control theoretic approaches for solving the problem of server consolidation. For example, Wang et al. [27] used the combination of feedback control system with optimization methods to reduce the energy consumption in data centers through server consolidation and DFVS technique. Their proposed power management solution contains two levels. At the application level, an MPC controller is designed to set the fraction of CPU time allocated to VMs and the frequency of the processor. At the cluster level, a power optimizer monitors the CPU resource demand of VMs and then finds a power efficient VM-PM mapping that satisfies CPU resource requirements of virtual machines. Berral et al. [28] have studied the placement of real-time applications. Similar to [27], they determine a deadline in the SLAs with the customers. In [28], machine learning techniques are used for optimizing the power usage and QoS simultaneously. Both [27] and [28] studies are designed for specific environments, such as high-performance computing (HPC), where given tasks have deadline constraints.

Garg et al. [18] have presented a method working with batch data for HPC environments and with continuous data for transaction environments such as websites. They assume a limited Cloud capacity, and thus, there is an admission controller that decides about accepting or rejecting a new application based on present and future resource availability. The assumption of a limited capacity is for private Clouds and it’s not a general condition for public Clouds. In addition, they have focused on batch data and SLAs in their proposed algorithms, but no migration policy is considered. The present study aims at conducting server consolidation and reducing the total cost in an unlimited Cloud by using live migrations over time.

As mentioned by Wolke et al. [29], dynamic resource allocation in Cloud environment is beyond a simple bin-packing problem, requiring advanced strategies. All conducted studies apply VM placement strategies based on the current utilization of resources or they use prediction methods to forecast some issues, such as overload conditions. In contrast to the previous works, we consider the stochastic behavior of the workloads instead of only focusing on the current status the resources in the Cloud. To the best of our knowledge, this is the first work that considers the loads on VMs as stochastic processes to solve the server consolidation problem. Accordingly, this study analyzes the problem from a new point of view, many solution paths are reinspected, and then new heuristics for dynamic server consolidation are proposed to reduce the costs and increase the quality of services.

3 System model

The system we consider in this paper includes a data center with heterogeneous physical machines, which heterogeneous VMs run on them. The system uses a network-attached storage (NAS) to enable live migration of VMs. A high-level view of the system model is shown in Fig. 1. Workloads are continuously applied to the system by multiple independent users; they are considered as non-stationary stochastic processes. Data center manager redirects the requests to the relevant virtual machine based on the desired application of each user.

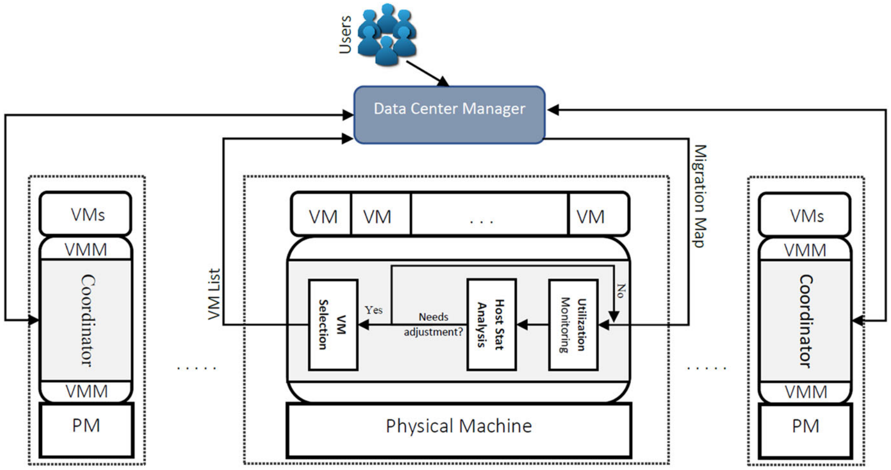

Fig. 1 System model

There is a coordinator as a module of the VMM on each physical machine. It monitors the resource utilization process on its host, and then analyzes the state of the host based on the observed utilization. For this purpose, the non-stationary process of the total utilization on a host is approximated as a sequence of stationary workloads, and then the central limit theorem (CLT) [30] is employed to analyze the state of the host. The residing coordinator on a host sends an adjustment request to the data center manager whenever it identifies the host state as one of the following conditions:

(i) Resource demand on the host exceeds its capacity, or the probability of this event is high according to the utilization process.
(ii) Resource utilization on the host is too low, and it is below a predefined threshold.
(iii) The current utilization process on the host has changed significantly relative to its previously known state, and it is a lower level of utilization than it used to be.

The adjustment request contains a list of VMs for relocation. Note that the state detection algorithms are executed locally on each host, and thus, the scalability of the system improves. The data center manager, upon receipt of adjustment requests, calls the VM placement algorithm named SBBFD and issues migration commands to VMMs using the list of triples (sourceID, vmID, destID) obtained from SBBFD. The objective of SBBFD is to reduce the total number of active hosts while considering the overload probability. The system works in a centralized form only when it employs the VM placement algorithm through the global manager.

3.1 Cost model

For a Cloud service provider, the total cost includes many parameters. This study considers the computation cost incurred by active servers, SLA violation and switching cost to toggle a server into and out of active mode. The combination of these costs gives the total cost equation as follows.

$$\text{Cost} = C_p \sum_{h=1}^{n} \int_{t=0}^{T} w_h\,(t)\mathrm{d}t + C_s \sum_{t=1}^{T} \left(|h|_t - |h|_{t-1}\right) + C_v \sum_{h=1}^{n} \int_{t=0}^{T} v(h)\mathrm{d}t \tag{1}$$

where $C_p$is the energy cost coefficient and $w_h\,(t)$ is the amount of energy usage at the time t on the host h, as given in Eq. (2). The constant C_s is the switching cost which can be interpreted based on different criteria and $|h|_t$ is the number of active hosts at time t. C_v is the SLA violation cost coefficient indicating the importance of SLAs. If there is any violation in the host h, then *v(h)* is equal to one; otherwise, it is set to zero.

3.2 Energy consumption

If the utilization of a host h is defined as the ratio of CPU utilization of the host to its total capacity, then the energy consumption can be assumed as a linear function of utilization. When a host is not used it is assumed to consume no power. Thus, energy consumption is calculated as follows.

$$w_h(t) = \begin{cases} \alpha_h + \beta_h u_h\,(t)\ , & u_h(t) > 0 \\ 0 \quad , & u_h(t) = 0 \end{cases} \tag{2}$$

where α_h and β_h are the constant coefficients depending on the characteristics of the host and $u_h\,(t)$ is the host utilization at the time t.

3.3 Switching cost

Most studies assume switching a server mode has no cost while in the real world it can have big costs and risks and Cloud service providers avoid it as much as possible [20] If a host is idle, i.e., when there is no load on it, it is switched off or switched to sleep mode. Toggling a server back and forth between active and non-active modes has some costs, such as [20]:

1. The energy used for switching the mode.
2. The delay caused by data migration, communications, etc.
3. The wear and tear due to switching the mode.
4. The Risk associated with server-mode changes.

If only 1 and 2 are important, then the switching cost (C_s) is on the order of the cost of running the server for a few seconds. If 3 is included, then the cost is like running the server for minutes to an hour. If all items are taken into consideration, then C_s may be like the cost of running the server for many hours. You can read more information on this in [20,31,32]. In this study, the cost C_s of toggling a server is modeled as being incurred when the server is returned to an active mode. Using the assumptions stated in [20], we consider this cost on the order of running a server for an hour utilizing half of its capacity.

3.4 SLA violation

SLA violation means that the QoS requirements of users are not met and services are having a quality less than what was established between customers and the Cloud provider in the agreement [18] In this paper SLA violation occurs in two cases:

1. If the total requested CPU on a PM exceeds the available capacity, then the VMs residing on the PM will provide services with less quality than what was specified in SLAs. In this case, the applications running on these virtual machines will encounter a reduction in performance.
2. When a VM is transferred from a PM to another PM in a live migration, applications running on this VM will encounter a reduction in performance while the migration process is taking place. The performance degradation depends on different factors such as the number of memory pages of the applications on the VM, bandwidth between the two machines, and so on [33]. In this study, the average reduction of quality is assumed to be 10% of VM's CPU utilization during the migration. Furthermore, the duration of migration can be calculated as follows.

$$\mathrm{MD}_{pm_j}^{vm_i} = \frac{Al_{vm_i}^{m}}{Av_{pm_j}^{b}} \tag{3}$$

where $\mathrm{MD}_{pm_j}^{vm_i}$ determines the migration duration of VM_i to PM_j. $Al_{vm_i}^{m}$ is the size of memory used by VM_i, and $Av_{pm_j}^{b}$ is the available bandwidth on PM_j. Note that this study ignores some issues like reliability and availability, which are key components of data center service-level agreements.

4 Stochastic workload analysis

The model introduced in Sect. 3 considers the workloads of requested resources as stochastic processes. Despite the dynamic behavior of workloads, it is assumed that they are not completely random. This assumption is true because many studies like [11] and [34] have successfully worked on predicting the behavior of workloads based on the past data. Each workload enters an application which is placed on a VM, which means that the CPU usage by a VM arbitrarily varies over time. Thus, in this study, the CPU utilization of any VM is considered as a continuous-state process and can be illustrated with a bivariate function like $X_{vm}(t,\zeta)$. In this function t $(\in \mathrm{T})$ is a member of parameter space and ζ $(\in \mathrm{S})$, which is a member of state space, is the CPU utilization based on million instructions per second (MIPS). We shall use $X_{vm}(t)$ to represent the aforementioned process omitting its dependence on ζ. For the sake of clarity, the parameters used throughout the paper and their definition are summarized in Table 1.

4.1 Stationary workloads

In this section, for the sake of clarifying the main idea of this study, it is assumed that the workloads are stationary. In the next sections, it is discussed about non-

Table 1 Symbols and definitions

Symbol	Definition
C_p	Energy cost coefficient
C_s	Switching cost coefficient
C_v	SLA violation cost coefficient
$w_h(t)$	Energy consumption of the host h at the time t
$\|h\|_t$	Number of active hosts at the time t
$v(h)$	Indicates whether the host h is experiencing SLA violations
$\alpha_h,\ \beta_h$	Constant coefficients of the host h in energy consumption equation
$u_h(t)$	CPU utilization of the host h at the time t
$u_t^{\{vm_k\}}$	CPU utilization $\{$of the virtual machine $k\}$ during the tth time interval
$Al_{vm_i}^m$	Size of the allocated memory to the virtual machine i
$Av_{pm_j}^b$	Size of the network bandwidth available on the host j
$X_{vm}(t,\zeta) = X_{vm}(t)$	Stochastic process of CPU utilization on a virtual machine
$Y(t)$	Stochastic process of CPU utilization on a host
$X_{vm\,i}$	Random variable of CPU utilization on a virtual machine
$Y_{\{pm_j\}}$	Random variable of CPU utilization on a host
$\sigma_{\{vm\|pm\}}$	Standard deviation of CPU utilization on a $\{$VM or PM$\}$
$\mu_{\{vm\|pm\}}$	Mean of CPU utilization on a $\{$VM or PM$\}$
$\hat{\sigma}$	Estimated standard deviation of CPU utilization on a host
$\hat{\mu}$	Estimated mean of CPU utilization on a host
$l_{\mathbf{w}}$	Sliding window length
λ	Mean rate of migration due to oversubscription
λ'^{-1}	Mean or expected value of an exponentially distributed random variable
α	Shape parameter of a gamma-distributed random variable
β	Scale parameter of a gamma-distributed random variable
C_{host}	CPU capacity of a host in MIPS
th_s	Safety threshold for VM placement
th_o	Overload probability threshold
th_u	Underload threshold
H_0	Null hypothesis
$\triangle_0$	Difference between the means of two sets of observations
n_s	Number of stable state's observations
n_c	Number of current state's observations
m_s	CPU utilization mean of stable state
m_s	CPU utilization mean of current state
S_s	CPU utilization standard deviation of stable state
S_c	CPU utilization standard deviation of current state
α_p	Threshold value for p value (significance level of the test)

stationary workloads and how to handle them for VM consolidation problem. Physical servers in Cloud data centers have enough capacity for hosting several virtual machines simultaneously. Assume that VM$_i$ ($i = 1,\ldots,$n) is placed on host PM$_j$ ($n \gg 1$). Looking at VM$_i$ CPU utilization at a certain point in time (like t_1), a random variable will be observed (i.e., $X_v m_i\ (t_1) = X_v m_i = Rv$). Furthermore, in the real world, VMs and their loads are independent of each other.

$$X_{vm\,1} \perp\!\!\!\perp X_{vm\,2} \perp\!\!\!\perp \ldots\ \perp\!\!\!\perp X_{vm\,n} \tag{4}$$

Hence, there are n independent random variables on PM$_j$ at a time like t_1. The utilization of PM$_j$ is equal to the sum of the utilization of all VMs on it; thus, it is a random variable.

$$Y = Y_{pm\,j} = X_{vm\,1} + X_{vm\,2} + \cdots + X_{vm\,n} \tag{5}$$

where Y is the random variable of CPU utilization on the host at a certain time. It is obvious that workloads in real systems are not identically distributed. However, we claim that the CLT [30] holds, and the aggregate usage of resources (CPU in this case) on a host has approximately a normal distribution. Based on Lindeberg–Feller Theorem [35], in general case (not necessarily in the conditions where the random variables are i.i.d), with simplification, the sufficient and necessary condition for the limiting distribution to converge in distribution to normal is defined as follows.

$$\forall i\ :\ \lim_{n\to\infty} \frac{\sigma_i^2}{\sum_{k=1}^{n} \sigma_k^2} = 0 \tag{6}$$

The interpretation of this equation is that if no sentence is dominant and the variance of each variable compared with the total variance is insignificant, then the sum of independent variables with different distributions will approach a normal distribution. Papoulis and Pillai in [36] have argued that the approximation error for values of small n will be very low. In addition, the placement of some VMs on a host shows that none of the VMs has a very big (or infinite) utilization compared with others, and thus, Eq. (6) holds for consolidation problem. From Eq. (4) and since the conditions of Eq. (6) are established, it can be concluded that the distribution of Y converges to a normal distribution.

$$Y \sim N(\mu_y, \sigma_y^2) \tag{7}$$

Since the random variables $X_{vm\,i}$ are of continuous state space, the density function $f_Y\,(y)$ approaches a normal density.

$$f_Y(y) \cong \frac{1}{\sigma_y \sqrt{2\pi e^{-(x-\mu_y)^2/2\sigma_y^2}}} \tag{8}$$

where $\mu_y = \mu_{\mathrm{vm}_1} + \cdots + \mu_{\mathrm{vm}_n}$, and $\sigma_y^2 = \sigma_{\mathrm{vm}_1}^2 + \cdots + \sigma_{\mathrm{vm}_n}^2$. By assuming the stationary condition, CPU utilization of each PM is a stationary normal stochastic process with the first-order moment $E\,(Y\,(t)) = m_y\,(t) = \mu_y$ (where $m_y\,(t)$ is the process mean), and the covariance $C_Y\,(\tau) = E\left(\tilde{y}^2\right) = \sigma_y^2$. If we define $\mathrm{Z} = \frac{Y-\mu_y}{\sigma}$ then:

$$F_Z\,(z) \underset{n\to\infty}{\rightarrow} G(z) \tag{9}$$

$$f_Z(z) \underset{n\to\infty}{\rightarrow} \frac{1}{\sqrt{2\pi}} e^{-(z)^2/2} \tag{10}$$

4.2 Non-stationary workloads

The analysis presented in Sect. 4.1 was for stationary workloads, but the problem we face in the real world data is that they are non-stationary and unknown a priori. Non-stationary workload means that some properties of process change over time. Thus, in case of consolidation problem, it means that the distribution function parameters of PMs utilization are not constant and may change over time. One approach that adapts known stationary workloads to the conditions of unknown non-stationary workloads is Sliding Window workload estimation method [37]. The main idea is that a non-stationary workload can be approximated as a sequence of stationary workloads. In other words, time is segmented into some intervals assuming the workload is stationary at each time interval.

In this case, distribution function will be a function of current stationary workload. Such a time division requires complete information on workloads and the times that their properties change. In fact, we don't have such information and decision-making can only be performed based on previous observations. Thus, heuristic methods for approximation of such solution should be presented. In this study, by defining a new concept called unstable state for physical servers, we will realize whenever the current workload has significantly changed compared to the previously known properties of the workload on the same node. However, based on the sliding window method, a time window of length l_w moves over time and captures the workload information of this time interval. Considering the workload on a PM as a stationary process during the time window length, maximum likelihood estimation (MLE) method can be used to estimate the parameters of the process as follows.

$$\hat{\mu} = \frac{1}{l_w} \sum_{j=1}^{l_w} u_{t-j} \tag{11}$$

$$\hat{\sigma} \;\; = \sqrt{\frac{1}{l_w} \sum\nolimits_{j=1}^{l_w} \left(u_{t-j} - \hat{\mu}\right)^2} \tag{12}$$

where the set $\{u_{t-l_w}, u_{t-l_w+1}, \ldots, u_{t-1}\}$ is the sequence of the last observed utilizations of the PM by the time t. Different values for l_w are tested and the effect on the performance of the proposed approaches are evaluated using CloudSim simulation toolkit [38] to find the suited length for l_w.

5 Proposed stochastic process-based solutions for dynamic server consolidation

One of the main drawbacks of the server consolidation solutions proposed in the literature is that their *VM placement algorithms* work based on the current resource utilization. This problem may lead to extra migrations due to resource oversubscription in the next time steps. In this section, at first, it is shown that these migrations lead to extra energy consumptions in the data center; then in the following subsections, novel heuristic algorithms are presented for the consolidation problem. The proposed algorithms consider workloads as stochastic processes and also take into account overload probability when performing VM placement.

5.1 Analysis of the extra energy consumption due to migrations from overloaded hosts

This section calculates the average energy consumption happens due to host overloads, and thus, it is shown that to reduce its cost it is important to control the overload probability of hosts. If the event of migration due to oversubscription (MDO) happens with the mean rate λ in a data center with homogenous PMs, then the number of MDOs can be assumed as a Poisson process. Showing this process with $M(t)$, the expected number of MDOs over t time intervals is calculated as follows.

$$E\left(M\left(t\right)\right)=E\left(M_{0,t}\right)=\lambda t \tag{13}$$

If the probability that a new host has to be activated (due to insufficient resources in the current active hosts) to migrate a VM from an overloaded host is shown with p, then the probability that no new host has to get activated is $q = 1 - p$. Let the random variable H represent the number of activated hosts due to MDOs over t time intervals, then H has a binomial distribution, i.e., $H \sim B(\lambda t, p)$; therefore,

$$E\left(H\right)=\lambda pt \tag{14}$$

where $E\left(H\right)$ is the expected number of activated hosts. Let L be the lifetime of a newly activated host during which the host remains active. If L is assumed to have an exponential distribution with the mean ${\lambda'}^{-1}$ (i.e., $L \sim \exp(\frac{1}{\lambda'})$), then the sum of time that extra hosts remain active is $S = L_1 + L_2 + \cdots + L_n$. Considering activation duration of extra hosts independent and identically distributed, the probability density function of S is:

$$f_S\left(s\right)=f_{L_1+L_2+\cdots+L_n}\left(s\right)=\lambda' e^{-\lambda' s}\frac{\left(\lambda' s\right)^{n-1}}{(n-1)!} \tag{15}$$

which means that the variable S has a gamma distribution with parameters $\alpha{=}n$ and $\beta{=}\lambda'$; i.e., $S{\sim}\Gamma$(n, λ'); therefore,

$$E\left(S\right)=E\left(\sum_{i=1}^{\lambda pt}L_i\right)=\frac{\lambda pt}{\lambda'} \tag{16}$$

where *E(S)* is the expected value of S. From (16) and the energy equation presented in Eq. (2), the expected value of extra energy consumption due to MDOs over t time intervals is calculated as follows.

$$E(w)=\int_{t=0}^{\frac{\lambda pt}{\lambda'}}w(t)\mathrm{d}t=\int_{t=0}^{\frac{\lambda pt}{\lambda'}}\alpha_h+\beta_h u_h\left(t\right)\mathrm{d}t \tag{17}$$

It can be concluded from the above statements that MDOs may lead to extra energy consumption in a Cloud data center. This extra cost can be controlled by reducing the overload probability of a PM before placing a VM on it, and thus, the number of migrations mean density λ will be reduced. Therefore, it is important to consider both the overload probability and the current resource utilization in VM placement algorithms.

5.2 A new heuristic algorithm for virtual machine placement

In this section, we present a novel heuristic algorithm that overcomes the drawbacks of previous VM placement algorithms in many cases. Heuristic-based VM placement policies are based on the idea of mapping the placement problem to bin-packing problem. It is done by considering PMs as bins and VMs as objects, and the main goal is to reduce the total number of the used bins. In the proposed heuristic here, a Stochastic Process-Based BDF (SBBFD) algorithm is presented that performs the allocation of VMs to PMs based on the stochastic process of CPU utilization of VMs. The algorithm aims to maximize the utilization of resources on each active machine, which results in using less physical machines, while the oversubscription probability is held lower than a safety threshold. Since the PMs are heterogeneous, the best fitting PM is defined as a machine with the least expected value (EV) of residual space. For this purpose, after sorting VMs in the decreasing order of the expected value of utilization, the algorithm places each VM on a host with the least EV of residual space.

Unlike the previous VM placement algorithms that treat the current load on the virtual machines like momentary events, SBBFD considers the stochastic process of CPU utilization of both VMs and PMs during l_w time intervals. In line 7, based on the current CPU utilization of pm_j and the requested share of CPU by vm_i, the feasibility of allocation is testes. Next, if there are enough resources, the vector of total CPU utilization at each time over the last sliding window on pm_j, i.e., $[u_{t-}l_w,\, u_{t-l_w+1},\, \ldots,\, u_{t-1}]$, will be obtained in line 8. By using MLE, EV of residual space after the placement of vm_i is estimated in line 9. Then, in line 10, the probability of CPU utilization to exceed host capacity is calculated. Next, the overload probability is checked to be lower than the safety threshold th_s in line 11. By conducting this algorithm completely, the best PM for hosting each given VM is obtained.

Algorithm 1: Stochastic Process-Based BFD (SBBFD) algorithm for Dynamic VM Placement

```
input  : pmList, vmList
output: Migration Map: (sourceID, vmID, destID)
Function SBBFD (pmList, vmList) begin
    vmList.sortDecreasingExpectedValue()
    for i=1 to vmList.Size() do
        minResidual←Inf
        allocatedPM←NULL
        for j=1 to pmList.Size() do
            if $pm_j$ has enough resource for $vm_i$ then
                $U^{(t')}_{pm_j,vm_i} \leftarrow \sum_{k=1}^{n} u^{vm_k}_{t'}, \ \forall t' \in \{t - l_w, t\}$
                $\mu_{\text{residual}} \leftarrow C_{pm_j} - MLE_{\hat{\mu}}(U^{(t')}_{pm_j,vm_j})$
                $P_{over} \leftarrow P(utilization\left(pm_j,\ vm_i\right) > C_{pm_j})$
                if $\mu_{\text{residual}} < minResidual \ \&\& \ P_{over} < th_s$ then
                    allocatedPM← $pm_j$
                    minResidual← $\mu_{\text{residual}}$
                end
            end
        end
        if allocatedPM ≠ NULL then
            migrationMap.add($vm_i.getHostID$, $vm_i.getID$, $allocatedPM.getID$)
        end
    end
    return migrationMap
end
```

In contrast to SBBFD, one of the disregarded issues in many proposed VM placement algorithms such as PABFD [11] is the probability of overutilization after VM placements. It is important to take this issue into consideration because of the problem discussed in Sec. 5.1. Also, if PMs become overloaded, then extra costs of SLA violations will be imposed on the service provider. The stochastic nature of SBBFD prevents some unnecessary server-mode changes. In a scenario, the data center manager may be able to accommodate all of a low-loaded host's VMs on other machines at a time interval and turn off the host, but this placement might not be possible at the next time interval of utilization; thus, a new PM should be turned into active mode again. In contrast, in this paper, it is avoided to identify underloaded PMs based on momentary utilizations by considering stochastic processes of CPU utilization and the overload probability.

5.3 VM placement time

It is important to specify the time when VM placement algorithm should be employed. In other words, we should determine the best times that VM migrations should take place to minimize the total cost. In many studies such as [11,25], it is performed in two situations: (i) when a host is overloaded or (ii) when it is underloaded. Since this work investigates server consolidation problem from the ”stochastic processes” viewpoint, these situations should be redefined. In addition, a new concept named ”unstabled PMs” will be introduced. If a PM is recognized as unstabled, it means that the utilization process has dominantly changed compared to the last time a placement was applied to the VMs of this PM, and thus, it’s VMs are candidates for replacement.

5.3.1 Overloaded hosts

It is a condition when the requested resources on a physical host are more than its capacity. This leads to performance degradation of applications, and thus, it is necessary to avoid its occurrence. Overloading conditions for a PM are defined as follows:

(a) When the total CPU demand on a PM is larger than its capacity, that machine will be known as an overloaded machine.

(b) IF the stochastic process of CPU utilization of a PM is shown with *Y(t)* whenever the probability that *Y(t)* can take values greater than the PM’s capacity is more than a threshold th_o, the PM is considered as an overloaded machine. The value th_o shows the importance of SLAs. As its value is less, it shows the higher importance of SLAs.

$$P\left(Y\left(t\right) > C_{\text{host}}\right) > th_o \tag{18}$$

For this purpose, we use Eqs. (11) and (12) to estimate the parameters of *Y(t)* over the last sliding window.

(c) When the predicted utilization value according to the last observations is greater than the available CPU capacity, the PM is added to the set of overloaded PMs. To predict the short-term CPU utilization of a PM, we employ SVR regression with a linear kernel.

After determining overloaded PMs, in the next step, VMs residing on each overloaded PM are selected one by one for migration based on minimum migration time (MMT) policy defined in [11] until elimination of the hot spots. Following that, in order to find new hosts for the selected VMs, they are given to SBBFD placement algorithm. All the VMs that the SBBFD has found a proper destination for them are added to the migration map.

5.3.2 Underloaded hosts

A PM which all of its VMs can get transferred to other PMs in a way that no new violations occur is called an underloaded machine, and thus, it is switched off. Two PMs with utilizations below 50% can get merged safely in a homogeneous server environment. In this work, for heterogeneous servers with different capacities, when the expected value of a PM utilization is less than $th_u = 50\%$ of its capacity, the data center manager checks the feasibility of placing all VMs from this PM on other active hosts keeping them not overloaded. To determine the underloaded PMs, the list of candidate machines is sorted by average utilization ascending, then iteratively for each PM, SBBFD is called with all the VMs residing on it. This model also improves the scalability of the system, since the candidate PMs are found locally.

5.3.3 Unstabled hosts

Each time a PM is involved in a VM placement process, i.e., it is the source/destination of a migration, the PM is assumed to be in a stable state after the placement procedure is done. When utilization process changes significantly on a PM relative to its known stable state, it is considered as an unstabled host. VMs allocated to a low-load unstabled PM (a PM which has a lower utilization than its stable state) need to be relocated. To do this, all the VMs on such PMs are given to the placement algorithm as input and it's examined whether the total number of active PMs can be reduced. If this can be accomplished, the VMs are set for migration to the determined target PMs, otherwise, no action will be taken. Note that, by defining unstabled hosts and taking appropriate actions about them, the system is implicitly adapted to the non-stationary behavior of the workloads.

Unstabled host detection The purpose of this section is to determine if the current CPU utilization has changed significantly in comparison with the stable state of the server. Since utilization process is non-stationary and variance may change over time, Welch t-test is employed. The null hypothesis is defined as $H_0{:}\mu_1 - \mu_2 = \Delta_0$ in the general form of Welch t test, but there is not an exact t statistic available for testing it. However, if H_0 is true, the statistic t_0^* is distributed approximately as t for the current and the stable state period.

$$t_0^* = \frac{m_s - m_c - \Delta_0}{\sqrt{\frac{S_s^2}{n_s} + \frac{S_c^2}{n_c}}} \tag{19}$$

m_s and m_c are the CPU utilization means of the stable state and the current period respectively. Here, Δ_0 is equal to zero, the number of observations $n_s = n_c = l_w$ and S_c stands for the standard deviation of the current utilization and S_s is the standard deviation of the stable state's utilization. The degree of freedom is calculated as follows.

$$df = \frac{\left(\frac{S_s^2}{n_s} + \frac{S_c^2}{n_c}\right)^2}{\frac{\left(\frac{S_s^2}{n_s}\right)^2}{n_s - 1} + \frac{\left(\frac{S_c^2}{n_c}\right)^2}{n_c - 1}} - 2 \tag{20}$$

Omitting minus 2 and since $n_s = n_c = l_w$ the above formula can be simplified to

$$df \approx \frac{\left(\frac{S_s^2 + S_c^2}{l_w}\right)^2}{\frac{(S_s^4 + S_c^4)}{l_w{}^2(l_w - 1)}} = \frac{(l_w - 1)(S_s^4 + S_c^4 + 2S_s^2 S_c^2)}{S_s^4 + S_c^4} \tag{21}$$

Finding the p value for the values calculated in Eqs. (19) and (21) for a host, one can judge about the significance of the difference in utilization means. If the p value exceeds $\alpha_p = 0.05$, the null hypothesis cannot be rejected. That is, we do not have a strong evidence to conclude that the average of current CPU utilization differs from the average of the stable state, or it can be interpreted that the workload on the host is still stationary. On the other hand, $\alpha_p < 0.05$ means that server is not working in a stable condition. One of the two following situations is expected:

(i) $m_s > m_c$: means that the server is working with a load lower than the optimal resource utilization. Therefore, VMs on the server need placement optimization.
(ii) $m_s < m_c$: means that the load on the PM has increased compared with the stable state condition. Since the server is not recognized as overloaded, it is assumed as a safe condition and there is no change needed.

Whenever the condition (i) holds for a PM, VM placement algorithm is invoked to see if all the VMs of this host can get migrated to other hosts. Note that the PMs which fit the condition (i) can be added to the set of underloaded hosts defined in Sect. 5.3.2, but some researchers may be keen to study the other condition, i.e., and (ii) hence, in this paper, unstable state is studied and treated in isolation.

6 Experimental design and setup

In this section, we discuss the performance evaluation of the server consolidation model proposed in this paper. Repeatable large-scale experiments and evaluation of proposed algorithms on real systems are extremely difficult due to many limitations such as expense, and the effects of experiments on system availability, security and risks. Thus, we set up the experimental environment using an extension of CloudSim toolkit [38]. CloudSim is a discrete event simulation platform which enables us to perform repeatable experiments of resource provisioning and server consolidation on large-scale virtualized data centers.

It is important to simulate a large number of physical servers and virtual machines for performance evaluation of the proposed system. In our infrastructure setup we use real configuration; i.e., a data center including 800 heterogeneous PMs is simulated with two server configurations: HP ProLiant ML110 G4, HP ProLiant ML110 G5 with the characteristics shown in Table 2. The energy consumption of servers is computed based on the information described in Sect. 3.2. α_h and β_h parameters for each server type are extracted from the data provided in CloudSim toolkit [38] by applying a linear regression. There are also four types of VMs inspired form Amazon EC2 instance types as shown in Table 3.

Table 2 Configuration of servers along with α_h and β_h parameters

Server type	CPU model	Cores	Freq. (MHz)	RAM (GB)	Max. power (W)	α_h	β_h
HP ProLiant G4	Intel Xeon 3040	2	1860	4	117	86	0.3259
HP ProLiant G5	Intel Xeon 3075	2	2660	4	135	93.7	0.3953

Table 3 Four VM types (Amazon EC2 VM types) [25]

VM type	CPU (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

Table 4 Workload data characteristics [39]

Date	# VMs	Mean (%)	Median (%)	SD(%)
March 3, 2011	1052	12.31	6	17.09
March 6, 2011	898	11.44	5	16.83
March 9, 2011	1061	10.7	4	15.57
March 22, 2011	1516	9.26	5	12.78
March 25, 2011	1078	10.56	6	14.14

Since our proposed model is defined based on non-stationary workloads it's needed to conduct experiments using real workload data. Real workload traces are provided as a part of the CoMon project, a monitoring infrastructure for PlanetLab [39] In this project, the CPU utilization data is collected every five minutes from more than a thousand VMs that are distributed at more than 500 servers around the world. In fact, the workload is representative of an IaaS Cloud environment such as Amazon EC2, which several independent users create and manage VMs. We have chosen 5 days from the dataset which the characteristics of the data for each day including the number of VMs as well as the mean, median and standard deviation of the utilization are shown in Table 4. During the simulations, each VM is randomly assigned a workload trace from one of the VMs from the corresponding day. The proposed algorithms are publicly available online.[4]

6.1 Evaluation metrics

In order to evaluate the efficiency of the algorithms several parameters like energy consumption, QoS, the number of VM migrations and the number of mode switchings should be considered. However, some of them are negatively correlated. For example, energy consumption usually increases when SLA violation decreases. Therefore, appropriate metrics should be used to evaluate all the aspects of the consolidation performance properly. The metrics used for evaluating the efficiency of the proposed algorithms are explained as follows.

SLA violations Beloglazov and Buyya [11] proposed a workload independent metric called SLA Violation (*SLAV*) to evaluate the QoS delivered to any VM in an IaaS Cloud. It is a multi-parameter metric which represents both the SLA Violation Time

[4] http://monshizade.com/sb-dsc/.

Per Active Host (SLATAH) due to overutilization, and SLA Violations due to Migrations (SLAVM). The SLATAH and SLAVM metrics are independent and assumed with equal importance. Therefore, the significance of the SLA violation by the infrastructure is defined by the combined metric (SLAV) that describes the total performance degradation, which is calculated as follows.

$$\mathrm{SLAV} = \mathrm{SLATAH}\ \times \mathrm{SLAVM} \tag{22}$$

SLATAH represents the percentage of time, during which active hosts have experienced the CPU utilization of 100%.

$$\mathrm{SLATAH} = \frac{1}{|\mathrm{PM}|}\sum_{i=1}^{|\mathrm{PM}|}\frac{T_{oi}}{T_{ai}} \tag{23}$$

where |PM| is the total number of PMs; T_{oi} is the total time that the host i has experienced the utilization of 100% leading to an SLA violation. T_{ai} is the total time during which the host i has been in active state. SLAVM is the performance degradation experienced by VMs, and it is defined as follows.

$$\mathrm{SLAVM} = \frac{1}{|\mathrm{VM}|}\sum_{j=1}^{|\mathrm{VM}|}\frac{D_{vm\,j}}{R_{vm_j}} \tag{24}$$

where |VM| is the total number of VMs in the system; $D_{vm\,j}$ is the performance degradation that VM_j faces due to migration. In this study, the average reduction in performance is estimated to be 10% of CPU utilization during all migrations of VM_j. R_{vm_j} is the total CPU capacity in MIPS requested by VM_j during its lifetime.

Energy consumption One of the major performance metrics is energy consumption. The total energy consumption of physical nodes is calculated as defined in Sect. 3.2.

Energy consumption and SLA violations The main metrics used in the literature are energy consumption and SLAV. The objective is to minimize both of them; however, there is a trade-off between them. For this reason, we use Energy and SLA Violations (ESV) metric to evaluate the simultaneous optimization of energy and QoS of the proposed method.

$$\mathrm{ESV} = \mathrm{Energy}\ \times \mathrm{SLAV} \tag{25}$$

Switching cost, energy consumption, and SLA violations The main objective of the proposed management system in this study is to reduce the total cost. Here, the cost of returning a server to active mode is also considered as described in Sect. 3.3. Thus, to evaluate the performance of the proposed system considering all the cost factors introduced in Sect. 3.1, the Costs of Energy and Switching and SLA Violation (CESSV) metric is defined as follows.

$$\mathrm{CESSV} = (\mathrm{Energy} + \mathrm{switching\ Cost})\ \times \mathrm{SLAV} \tag{26}$$

6.2 Experimental results

In this section, we compare the proposed SB-DSC policy with four best standard benchmark heuristic policies reported in [11] for dynamic server consolidation problem in Cloud data centers. The reference solutions consist of a local regression (LR), local regression robust (LRR), interquartile range (IQR) and median absolute deviation (MAD) policies for detecting overloaded PMs, a simple method (SM) for determining underloaded PMs, MMT policy for VM selection from overloaded hosts and a power-aware algorithm named PABFD for VM placement of the VMs selected for migration.

First of all, we have to decide about the sliding window length. In fact, the sliding window length for which good estimates can be made depends on the workloads. For this purpose, we simulated the proposed system with different values of l_w and observed the impact on various performance metrics. The details of the results are summarized in Table 5. We decided to choose *CESSV* metric to designate the best length for sliding windows because it has information about all the cost factors that are important in this paper. The window of length 35 had the best result, so we consider $l_w = 35$ in the rest of the simulation experiments. The impact of different l_w sizes on *CESSV* metric for the 5 days in March 2011 is illustrated in Fig. 2. Note that in this study we consider the safety threshold for VM placement and the overload threshold for overloaded host detection equal to 0.05. The values of the parameters that are used in the simulation experiments are summarized in Table 6.

Results indicate that our proposed method outperforms the other four standard benchmark algorithms. This proves that is the heuristics presented in this study have a realistic assumption about the real workload traces, i.e., considering them as stochastic processes rather than considering them as events that randomly happen at a moment in time. Figure 3 shows that the proposed SB-DSC approach reduces the *SLATAH* more than the other algorithms. Therefore, there is a noticeable reduction in SLA Violations as shown in Fig. 4. The main reason is that SB-DSC considers the probability of overloading before a VM placement; thus, it enables the system to prevent the SLA violations in advance of VM placements, while the other consolidation methods take action to avoid violations only after placement.

As depicted in Fig. 5, the system we introduced in this study results in about 23% reduction in the total power consumption compared with the bests of the benchmark ones, i.e., LR-MMT and LRR-MMT. It shows the energy efficiency of the proposed heuristics in SB-DSC. More precisely, it can be inferred that adopting SBBFD placement algorithm, which tries to minimize residual spaces while caring about overload probability, along with underloading detection method and low-load unstabled hosts idea defined in Sect. 5.3.3 leads to more efficient utilization of resources with more compact consolidations. It is clear when there is a reduction in *SLAV* and energy consumption, the *ESV* metric reduces as well, as shown in Fig. 6. Furthermore, the trade-off between energy consumption and SLA violation can be adjusted by varying the safety and overload thresholds.

The number of VM migrations and the number of overloaded hosts are positively correlated. Since we care about overload probability before placement by using SBBFD, and after placement by using overloading detection method explained in

Table 5 Output results for different sliding window length (median values)

l_w	CESSV ($\times 10^{-5}$)	ESV ($\times 10^{-3}$)	Energy (kWh)	SLAV ($\times 10^{-5}$)	SLAVM	SLATAH (%)	#VM Mig. ($\times 10^{3}$)	#Reac. ($\times 10^{2}$)
20	24.028	1.24	113.44	1.04	0.02	4.33	9.00	9.06
25	25.306	1.29	114.59	1.1	0.02	4.57	9.07	9.07
30	25.822	1.39	114.21	1.2	0.02	4.8	8.71	9.09
35	20.664	1.17	115.08	0.97	0.02	4.13	8.91	9.23
40	20.894	1.09	113.69	0.97	0.02	4.14	8.77	9.11
45	21.855	1.11	113.96	1.14	0.02	4.47	8.41	9.12
50	23.450	1.17	114.41	1.22	0.02	4.48	8.477	9.03

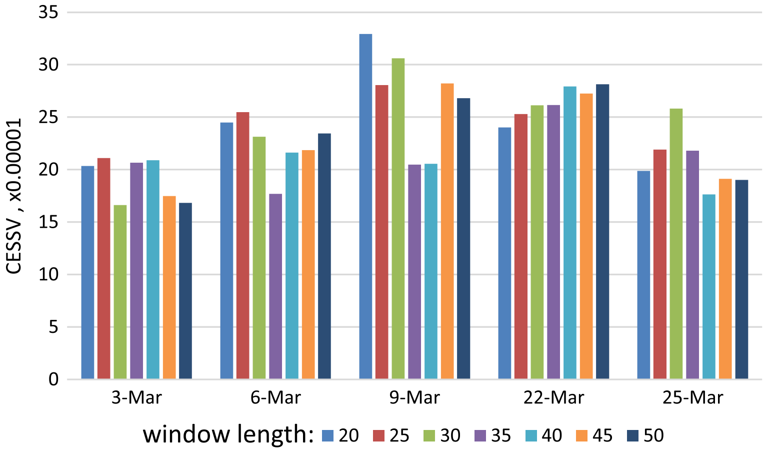

Fig. 2 *CESSV* for different sliding window length

Table 6 Values of parameters in the simulation experiments

l_w	th_s	th_o	th_u	α_p
35	0.05	0.05	50%	0.05

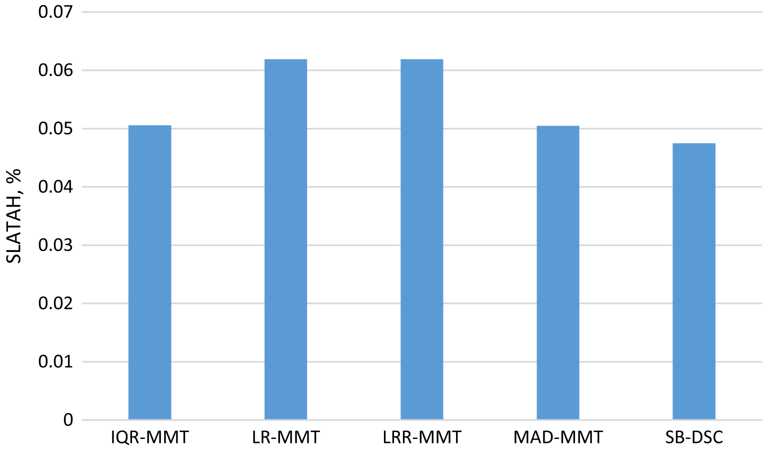

Fig. 3 *SLATAH* for server consolidation policies

Sect. 5.3.1, the number of migrations has prominently reduced (i.e., up to 66% reduction), as shown in Fig. 7. Furthermore, the level of SLA violations due to migrations (*SLAVM* metric) decreases when the number of VM migrations reduces. The results of *SLAVM* metric are shown in Fig. 8.

The number of times that we have to turn on a server after it is turned off is implicitly correlated with the success of the proposed algorithms to avoid unnecessary server-mode changes. Figure 9 shows the average number of times that servers

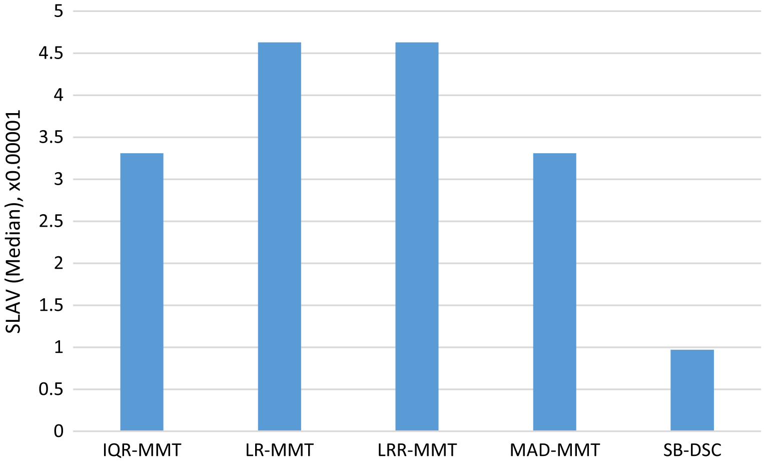

Fig. 4 *SLAV* for server consolidation policies

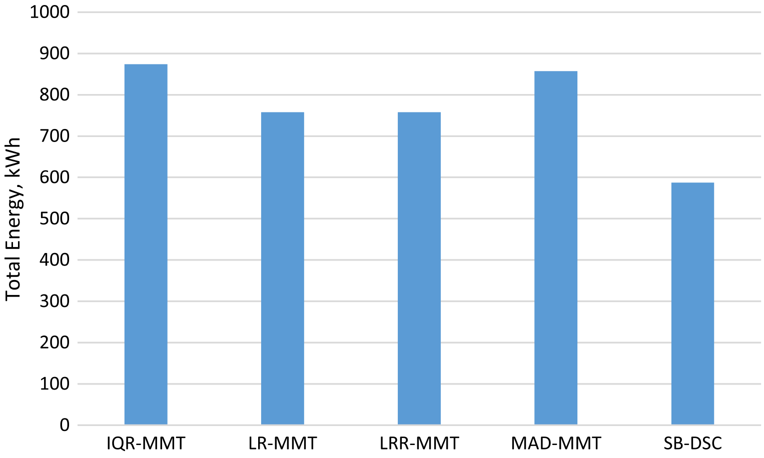

Fig. 5 Energy consumption for server consolidation policies

are switched to the active state in the five days of the simulations. Our proposed model has led to a noticeable reduction in this metric. The reason is that in comparison with the other solutions, SB-DSC reduces the MDOs by using SBBFD, and thus, fewer hosts have to be activated due to server oversubscriptions; and also the stochastic process-based nature of the solutions prevents momentary underloaded PMs detection.

Finally, to perform a comprehensive comparison between our proposed system and benchmark policies, we employ *CESSV* metric. As mentioned previously, this metric considers all the cost factors that are important in this study. We can conclude from the results which are illustrated in Fig. 10 that our proposed model has succeeded in reducing the main costs in the system. We expected this because our proposed model

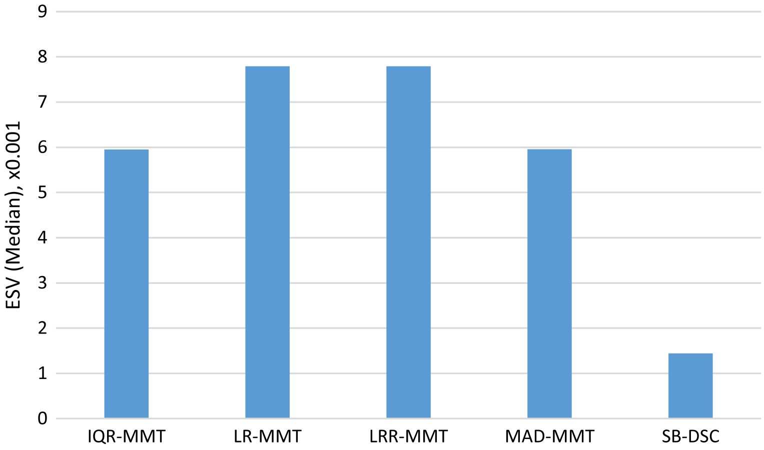

Fig. 6 *ESV* for dynamic server consolidation policies

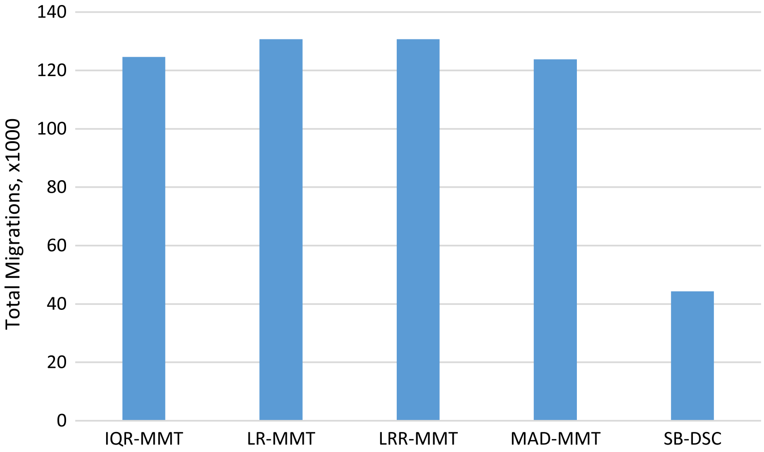

Fig. 7 Total migrations for server consolidation policies

has reduced energy consumption, SLA violation, the number of VM migrations and the number of mode switching.

7 Summary and future works

In this paper, an analysis of real workload data that are non-identically distributed was presented and then we proved that the distributions of CPU utilization on PMs converge to normal distributions. We presented a novel dynamic server consolidation approach based on the stochastic process of CPU utilization of virtual machines. It reduces energy consumption and switching cost while preserving SLAs requirements. We evaluated the performance of our proposed approach by conducting extensive

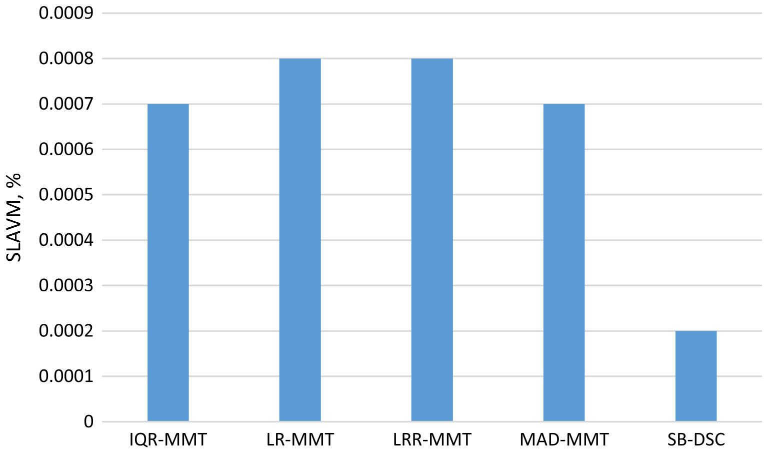

Fig. 8 *SLAVM* for server consolidation policies

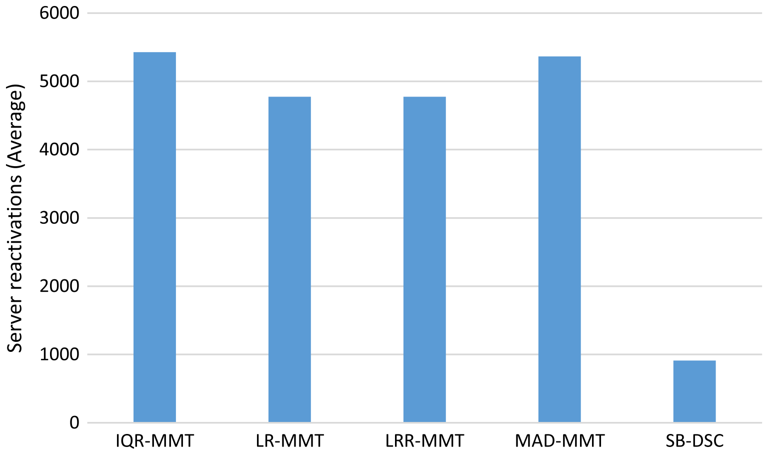

Fig. 9 Average number of host reactivations for server consolidation policies

simulations using CloudSim Toolkit with real workload data. Comparing the results to the benchmark heuristic policies, the performance of the system has improved in SLA violations, *CESSV*, *ESV*, number of migrations and switchings, and other metrics that was introduced in this study.

As a future work, we plan to further extend the proposed system by considering other resources in Cloud such as memory and bandwidth. We also intend to develop a metaheuristic algorithm for dynamic VM consolidation problem that work with complex workload models, such as workloads with Markov properties.

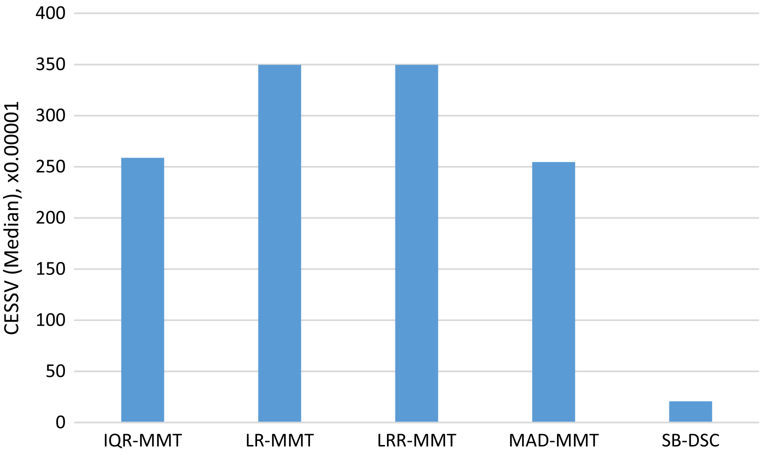

Fig. 10 *CESSV* for server consolidation policies

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Mell P, Grance T (2011) The NIST definition of cloud computing. National Institute of Standards and Technology NIST Special Publication 800-145
2. Rimal BP, Choi E, Lumb I (2009) A taxonomy and survey of cloud computing systems. NCM 9:44–51
3. Tanenbaum AS, Van Steen M (2007) Distributed systems: principles and paradigms, 2nd edn. Prentice-Hall, Upper Saddle River, pp 80–82
4. Barroso LA, Hzle U (2007) The case for energy-proportional computing. Computer 40(12):33–37
5. Gao Y, Guan H, Qi Z, Song T, Huan F, Liu L (2014) Service level agreement based energy-efficient resource management in cloud data centers. Comput Electr Eng 40(5):1621–1633
6. Lee YC, Zomaya AY (2012) Energy efficient utilization of resources in cloud computing systems. J Supercomput 60(2):268–280
7. Beloglazov A, Buyya R (2015) OpenStack Neat: a framework for dynamic and energy sefficient consolidation of virtual machines in OpenStack clouds. Concurr Comput Pract Exp 27(5):1310–1333
8. Feller E, Rilling L, Morin C (2012) Snooze: a scalable and autonomic virtual machine management framework for private clouds. In: Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012). IEEE Computer Society, pp 482–489
9. Rao KS, Thilagam PS (2015) Heuristics based server consolidation with residual resource defragmentation in cloud data centers. Future Gener Comput Syst 50:87–98
10. Song W, Xiao Z, Chen Q, Luo H (2014) Adaptive resource provisioning for the cloud using online bin packing. IEEE Trans Comput 63(11):2647–2660
11. 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
12. Farahnakian F, Ashraf A, Pahikkala T, Liljeberg P, Plosila J, Porres I, Tenhunen H (2015) Using ant colony system to consolidate VMs for green cloud computing. IEEE Trans Serv Comput 8(2):187–198
13. Rajabzadeh M, Haghighat AT (2017) Energy-aware framework with Markov chain-based parallel simulated annealing algorithm for dynamic management of virtual machines in cloud data centers. J Supercomput 73(5):2001–2017

14. Tang M, Pan S (2015) A hybrid genetic algorithm for the energy-efficient virtual machine placement problem in data centers. Neural Process Lett 41(2):211–221
15. 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
16. Zhang L, Zhuang Y, Zhu W (2013) Constraint programming based virtual cloud resources allocation model. Int J Hybrid Inf Technol 6(6):333–344
17. Masdari M, Nabavi SS, Ahmadi V (2016) An overview of virtual machine placement schemes in cloud computing. J Netw Comput Appl 66:106–127
18. Garg SK, Toosi AN, Gopalaiyengar SK, Buyya R (2014) SLA-based virtual machine management for heterogeneous workloads in a cloud datacenter. J Netw Comput Appl 45:108–120
19. 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
20. Lin M, Wierman A, Andrew LL, Thereska E (2013) Dynamic right-sizing for power-proportional data centers. IEEE/ACM Trans Netw 21(5):1378–1391
21. 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
22. Verma A, Ahuja P, Neogi A (2008) pMapper: power and migration cost aware application placement in virtualized systems. In: Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware. Springer-Verlag New York, Inc., pp 243–264
23. Luo L, Wu W, Tsai W-T, Di D, Zhang F (2013) Simulation of power consumption of cloud data centers. Simul Modell Pract 39:152–171
24. Quan DM, Basmadjian R, De Meer H, Lent R, Mahmoodi T, Sannelli D, Mezza F, Telesca L, Dupont C (2011) Energy efficient resource allocation strategy for cloud data centres. In: Computer and Information Sciences II. Springer, pp 133–141
25. 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
26. 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, 2008. San Diego, California, pp 1–5
27. Wang Y, Wang X (2014) Performance-controlled server consolidation for virtualized data centers with multi-tier applications. Sustain Comput Inf Syst 4(1):52–65
28. Berral JL, Goiri, Nou R, Juli F, Guitart J, Gavald R, Torres J (2010) Towards energy-aware scheduling in data centers using machine learning. In: Proceedings of the 1st International Conference on energy-Efficient Computing and Networking, 2010. ACM, pp 215–224
29. Wolke A, Tsend-Ayush B, Pfeiffer C, Bichler M (2015) More than bin packing: dynamic resource allocation strategies in cloud data centers. Inf Syst 52:83–95
30. Montgomery DC (2009) Introduction to Statistical Quality Control, 6th edn. Wiley, New York
31. Thereska E, Donnelly A, Narayanan D (2009) Sierra: a power-proportional, distributed storage system. Microsoft Research Ltd, Tech Rep MSR-TR-2009 153
32. Bodik P, Armbrust MP, Canini K, Fox A, Jordan M, Patterson DA (2008) A case for adaptive datacenters to conserve energy and improve reliability. University of California at Berkeley, Technical Report UCB/EECS-2008-127
33. Voorsluys W, Broberg J, Venugopal S, Buyya R (2009) Cost of virtual machine live migration in clouds: a performance evaluation. CloudCom 9:254–265
34. Farahnakian F, Pahikkala T, Liljeberg P, Plosila J (2013) Energy aware consolidation algorithm based on k-nearest neighbor regression for cloud data centers. In: 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing (UCC). IEEE, pp 256–259
35. Hunter DR (2014) Notes for a Graduate-Level Course in Asymptotics for Statisticians. Penn State University, Pennsylvania
36. Papoulis A, Pillai SU (2002) Probability, Random Vvariables, and Stochastic Processes. Tata McGraw-Hill Education, New York
37. Chung E-Y, Benini L, Bogliolo A, Lu Y-H, De Micheli G (2002) Dynamic power management for nonstationary service requests. IEEE Trans Comput 51(11):1345–1361

38. Calheiros RN, Ranjan R, Beloglazov A, De Rose CA, 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
39. Park K, Pai VS (2006) CoMon: a mostly-scalable monitoring system for PlanetLab. ACM SIGOPS Oper Syst Rev 40(1):65–74