Wireless Networks
https://doi.org/10.1007/s11276-018-1874-1

An adaptive heuristic for managing energy consumption and overloaded hosts in a cloud data center

Rahul Yadav[1] · Weizhe Zhang[1,2] · Keqin Li[3] · Chuanyi Liu[4] · Muhammad Shafiq[1] · Nabin Kumar Karn[1]

Abstract
In this paper, we address the problems of massive amount of energy consumption and service level agreements (SLAs) violation in cloud environment. Although most of the existing work proposed solutions regarding energy consumption and SLA violation for cloud data centers (CDCs), while ignoring some important factor: (1) analysing the robustness of upper CPU utilization threshold which maximize utilization of resources; (2) CPU utilization prediction based VM selection from overloaded host which reduce performance degradation time and SLA violation. In this context, we proposed adaptive heuristic algorithms, namely least medial square regression for overloaded host detection and minimum utilization prediction for VM selection from overloaded hosts. These heuristic algorithms reducing CDC energy consumption with minimal SLA. Unlike the existing algorithms, the proposed VM selection algorithm consider the types of application running and it CPU utilization at different time periods over the VMs. The proposed approaches are validated using the CloudSim simulator and through simulations for different days of a real workload trace of PlanetLab.

Keywords Cloud computing · Data center · Energy consumption · Host overloaded detection · Service level agreements · and VM selection

1 Introduction

The demand for computing resources is growing rapidly, thereby requiring large-scale data centers. These data centers require enormous amount of electric power to provide an essential service to cloud users, and such consumption increases carbon dioxide (CO_2) emissions and operating costs. According to statistics, data centers consume approximately 1.3% of the total worldwide electricity supply; the rate is predicted to increase by 8% in 2020 [1]. Therefore, CO_2 is also increasing substantially, which brings direct impacts to the environment.

The main reason for this extremely high energy consumption is not just the use of computing resources in large-scale and the energy inefficiency of hardware, but rather lies in the inefficient usage of these resources. Unfortunately, large amounts of electric energy are wasted by hosts during low workload. Utilization data of host resources collected from more than 5000 production hosts over a six-month period show that most of time hosts operate at 10–50% of their full capacity, thereby leading to wasting energy during low utilization of resources [2, 3].

✉ Rahul Yadav
rahul@stu.hit.edu.cn

✉ Weizhe Zhang
wzzhang@hit.edu.cn

Keqin Li
lik@newpaltz.edu

Chuanyi Liu
cy-liu04@mails.tsinghua.edu.cn

Muhammad Shafiq
muhammadshafiq@hit.edu.cn

Nabin Kumar Karn
karnnabin@hit.edu.cn

1 School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China

2 Pengcheng Laboratory, Shenzhen, China

3 Department of Computer Science, State University of New York, New Paltz, NY 12561, USA

4 School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China

Published online: 20 November 2018

Another reason is the narrow dynamic power range of servers: even completely idle servers still consume about 70% of their peak power [4]. Therefore, keeping servers underutilized is highly inefficient from the energy consumption perspective.

An important question is how to minimize CDC energy consumption while ensuring the service level agreements (SLAs) delivered by the cloud provider. In order to address the high energy consumption in CDC, we leverage the capabilities of the virtualization technology. In a cloud computing platform, a host is installed on more than one heterogeneous virtual machines (VMs) on parallel computing platforms called data center [5, 6]. The virtualization technology provides administrative privileges to VM users within the guest operating system in that they can customize their runtime resources according to their specific needs. These VMs can run different types of application simultaneously [7, 8]. Dynamic VM consolidation is a significant method for optimally utilizing computing resources from data centers. In this approach, the VMs are reallocated using live migration according to the current resource requirement to minimize the number of active hosts required to handle the workload [9, 10]. The idle hosts are switched to energy saving mode with fast transition times to eliminate static energy and reduce total energy consumption. The hosts are reactivated when workload demand increases. This method possesses two main objectives: one is to achieve efficient consumption of electric power, and the other is to minimize the violation of service level agreements (SLAs).

In summary, efficient energy consumption with minimal SLA violation generally involves the host overloaded detection in CDC, select VMs according to application running on VM. Existing methods discussed in related work also focus on host overload detection algorithm and VM selection algorithm. But these host overloaded detection algorithms are not robust for outlier. Moreover, for selection of VMs are not considering CPU utilization for running different type of application on VM and thus may not sufficiently reduce energy consumption of the CDC and minimize SLA violation rate under a different workload.

In this context, we introduce a novel regression-based algorithm called *LmsReg* to create an upper threshold for detecting overloaded hosts. From these hosts, some VMs are migrated to another appropriate host to minimize performance degradation. A novel dynamic algorithm called minimum utilization prediction (*MuP*) is introduced to VM selection to balance the tradeoff among electric power consumption, number of migrations, performance of hosts, and total number of shut-down hosts. These algorithms estimate the upper threshold and selection of VMs on the basis of statistical analysis of the CPU utilization history of hosts. The main contributions of this study are listed as follows:

- An adaptive heuristic algorithm called *LmsReg* is proposed to detect overloaded hosts. This algorithm aims to minimize total electric energy consumption and avoid SLA violation.
- A policy called *MuP* for VM selection is introduced to allocate VMs from overload hosts or underloaded hosts. This algorithm aims to minimize SLAs by selecting VMs from overloaded hosts.
- The efficiency of *LmsReg* and *MuP* is evaluated using the CloudSim simulator, and their performance is compared against that of other proposed approaches in literature.

The rest of the paper is organized as follows. Section 2 discusses previous works related to CDC resources and energy efficiency management. Section 3 presents the architecture of the Infrastructure as a Service (IaaS) cloud platform and its energy model. Section 4 describes the algorithm for overloaded host detection and the policy for VM selection. Section 5 proposes an energy efficiency metric for measuring the effectiveness of the proposed algorithms in the cloud environment. Section 6 introduces the experimental setup for the proposed algorithms. Section 7 analyzes and compares the simulation results of the proposed algorithms. Finally, Sect. 8 elaborates the conclusions of the study and future research direction.

2 Related work

The prior works concerning the energy consumption management in CDC can be broadly divided into three categories: (1) dynamic VM consolidation; (2) adaptive heuristics based threshold; (3) non-adaptive heuristic based threshold. These all are discussed in detail following subsection.

2.1 VM consolidation

The problem of VM consolidation is considered NP-hard [11]. Therefore, the cost of finding the optimal solution in large-scale virtualized data centers (large number of hosts and VMs) is high. Pahlavan et al. [12] proposed a VM consolidation algorithm for energy-aware application in CDC in consideration of structural features, such as racks and network topology. They also focused on the structure of the network and cooling system in CDC that hosts the physical machines during VM consolidation. Specifically, racks and routers are employed without compromising the SLAs. In this way, low traffic routers or idle routers and cooling equipment can be turned off to minimize electricity consumption. A common method for this purpose is the dynamic VM consolidation approach, which is also called dynamic VM management and dynamic resource allocation [13, 14]. However, the consolidation of VMs from overloaded and underloaded hosts optimizes the electric energy consumption of a CDC. Ranganathan et al. [15] described a method for power management of servers at the collective system level instead of individual server level. This approach permits active servers to steal power from inactive servers.

2.2 Adaptive heuristic based threshold

An adaptive based upper CPU utilization threshold is more efficient than traditional one. Beloglazov et al. [4] proposed Beloglazov et al. proposed different types of threshold that have been calculated through different statistical measures, like: The Random Selection Policy, Median Absolute Deviation, Local Regression, and The Maximum Correlation Policy. Here all the sub problems are addressed. Most of the heuristics are designed in such a way that they are adaptive to changes and keep their threshold changing based on different scenario in different time so that they can still provide the functionality and consolidation decision in changed environment. This adaption process allows the system to be dynamic. In [16], Beloglazov *et al.* introduced a heuristic-based energy-aware approach. This approach focuses on the statistical analysis of CPU utilization history to determine an upper threshold for detecting overloaded hosts. However, placing a VM on the host that will be overloaded causes performance degradation for this VM. As a solution, a VM controller should gather all information and provide an appropriate host during VM consolidation. Verma *et al.* [17] introduced a heuristic based models, namely, median migration time, smallest void detection, and maximum fill technique. Zhu *et al.* [18] introduced a static CPU utilization threshold for detecting overloaded hosts. If the utilization of the host is more than 85% of its total capacity, then this host is detected as an overloaded host. This approach is unsuitable for dynamic workload because it cannot adapt to workload changes. Several current works are focusing on decision making based on statistical analysis of historical data

2.3 Non-adaptive heuristic based threshold

Kusic et al. [19] proposed the idea of limited lookahead control where they sequential optimization a VM placement scheme using limited lookahead control. They used Kalman filter to predict the number of impending arrivals to deal with time-varying nature of workload and to perform necessary re-allocations. Simulation based learning needed for the application-specific adjustments. These kind of adjustments cannot be implemented by real Cloud providers. Moreover, the time complexity of their model is very high. Therefore, it is well executable for large-scale cloud systems. On the other hand, we propose adaptive heuristic model goal to achieve high performance with less energy consumption and minimal SLA violation for large-scale CDC and does not need simulation-based learning prior to deployment of application in real cloud system.

Von et al. [20] introduced many interpolation methods like linear interpolation, polynomial interpolation, Shepard interpolation, and nearest-neighbor interpolation to measuring the energy consumption and compared them to determine the accuracy of the proposed methods to select the best interpolation method using independent reference dataset containing a large set of data points. In this paper, we used the standard energy consumption details on given CPU utilization of hosts to estimate energy consumption for specified duration [21].

Nathuji et. al. [22] introduced a host resource management polciy for VM placement that achieve efficiency through local and global policies using live migration of VMs from overloaded servers. At the local level, the system applies power management policies of guest OSs. On the other hand, global manager keeps an eye the current resource allocation by the local manager. Global manager implements its policy on local resources to decide whether the VM placement is required. They did not considered any specific policy for automatic resource management at the global level. In the next section we will discuss about cloud computing system architecture in detail.

3 System architecture

Cloud computing is an on-demand network access based on virtualized platform and provides large-scale data center resources to users. The submissions of multiple requests for VM provisioning are allocated to hosts simultaneously. The allocation of VMs on hosts depends on the CPU utilization of host. The dynamic nature of VM means the workload of VM variability over time. The main components in the data center that consume electric energy are the CPU, primary storage, secondary storage, cooling unit, and network interface. The CPU consumes more electric energy than does other components [16]. The energy consumed by the CPU is linearly proportional to its utilization [23]. Therefore, the total resources of the host and utilization of VM resources are categorized by a single parameter called the CPU performance. The CPU performance is measured in million instructions per second (MIPS). The efficient consolidation of VMs reduces the consumption of electric energy and the rate of SLA violation. When the running VM cannot obtain its provision resources (such as MIPS and memory) from the CDC, SLA violation occurs. In this case, the cloud service provider must be pay penalty costs to the cloud server users. After the overloaded host is identified, VMs for migration from the overloaded host to the appropriate host are selected and are applied iteratively to the host until it is not considered overloaded.

The target cloud computing platform is shown in Fig. 1. This platform possesses two main components, namely, the central controller and the local controller, which are the same as those described in [4, 24]. The central controller is directly connected to the end users and the local controller, and the local controller is directly connected to the VM controller. The resource management of the data center is done by the central controller, which allocates VMs to hosts on the basis of a predefined policy. The VM controller is responsible for resizing VMs depending on their resource requirements, migrating VMs from one host to another host, and activating and reactivating hosts. The local controller resides in a separate VM on the virtualization layer, is directly connected to the central controller, and is responsible for monitoring the current state of hosts and sending all gathered information to the central controller. We propose an algorithm for overloaded host detection and an approach for VM selection to minimize electric energy consumption while satisfying SLA requirements. This approach first detects an overloaded host by using the dynamic upper CPU utilization threshold and then selects VMs from the overloaded host to reduce performance degradation. Finally, the selected VMs are migrated to the appropriate host to fulfil the requirements. Turning the idle hosts into energy saving mode is also helpful in reducing the energy consumption in the data center.

4 Energy-aware heuristic algorithms

Heuristic-based models provide better result than that of traditional ones, such as static threshold based energy-aware techniques. We propose a heuristic-based upper

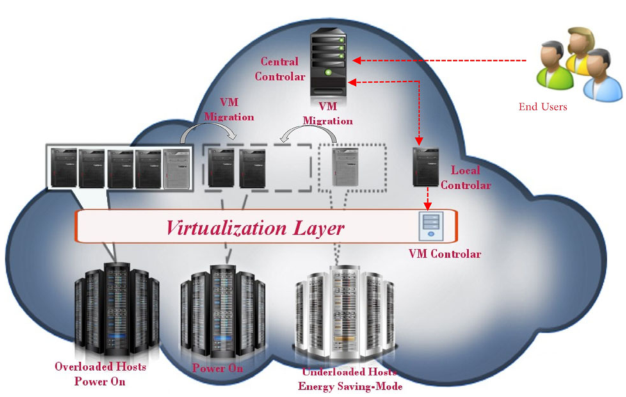


Fig. 1 Cloud IaaS architecture

CPU utilization threshold for detecting overloaded hosts and dynamic consolidation of VMs. The estimation of the upper threshold is based on the statistical analysis of the CPU utilization history data [25]. The proposed flowchart for *IaaS* resource management based on *LmsReg* for overloaded host detection and *MuP* for VM selection from overloaded hosts is shown in Fig. 2. In the first phase, the proposed algorithm lists the hosts, applies *LmsReg* for overloaded host detection, and compares the CPU utilization of hosts by using the estimated upper threshold. If the CPU utilization exceeds the upper threshold, then this host is predicted as an overloaded host, and vice versa. After detecting the overloaded host, all VMs of this host are identified and *MuP* for VM selection is applied to select the VMs that need to be migrated from the overloaded host. After the vmMigrateList is updated, the MBFD VM placement scheme [4] is actively applied on the selected VMs from the overloaded host and the VMs are allocated to the appropriate host. In the second phase, the method proposed in Beloglazov *et al.* [4] is applied to find underloaded hosts. Thereafter, all the VMs from these hosts are migrated to the appropriate host. We discuss *LmsReg* for overloaded host detection and *MuP* for VM selection in the following sections.

4.1 LmsReg for overloaded host detection

The dynamic nature of the cloud environment is a major concern for cloud service providers. The constant value of the processor utilization threshold is not an optimal solution for an unpredictable or dynamic workload of the cloud framework. Therefore, we propose an algorithm for this dynamic nature. The proposed method automatically adjusts the upper CPU utilization threshold on the basis of the statistical analysis of the past CPU utilization of the hosts. A robust regression model, which is more efficient than the traditional one, is used to construct the proposed algorithm. The variability of CPU utilization directly impacts the upper CPU utilization threshold. If variability is small, the CPU utilization reaches 100% utilization, which causes performance degradation or SLA violation. Unlike traditional approaches, robust regression techniques provide a more efficient optimal solution. These techniques do not directly influence by the outliers in the CPU utilization dataset, which makes them robust and reliable for the dynamic cloud environment. Least median of squares regression is commonly used to minimize the median of squared residuals [26]. *LmsReg* is a more robust estimator than other estimators, such as median, standard deviation, variance, and ordinary least squares (OLS). In OLS, a single corrupt data point can provide the resulting regression line with an arbitrarily large slope [27]. For calculating the *LmsReg* model, we need to calculate the OLS model and the relationship

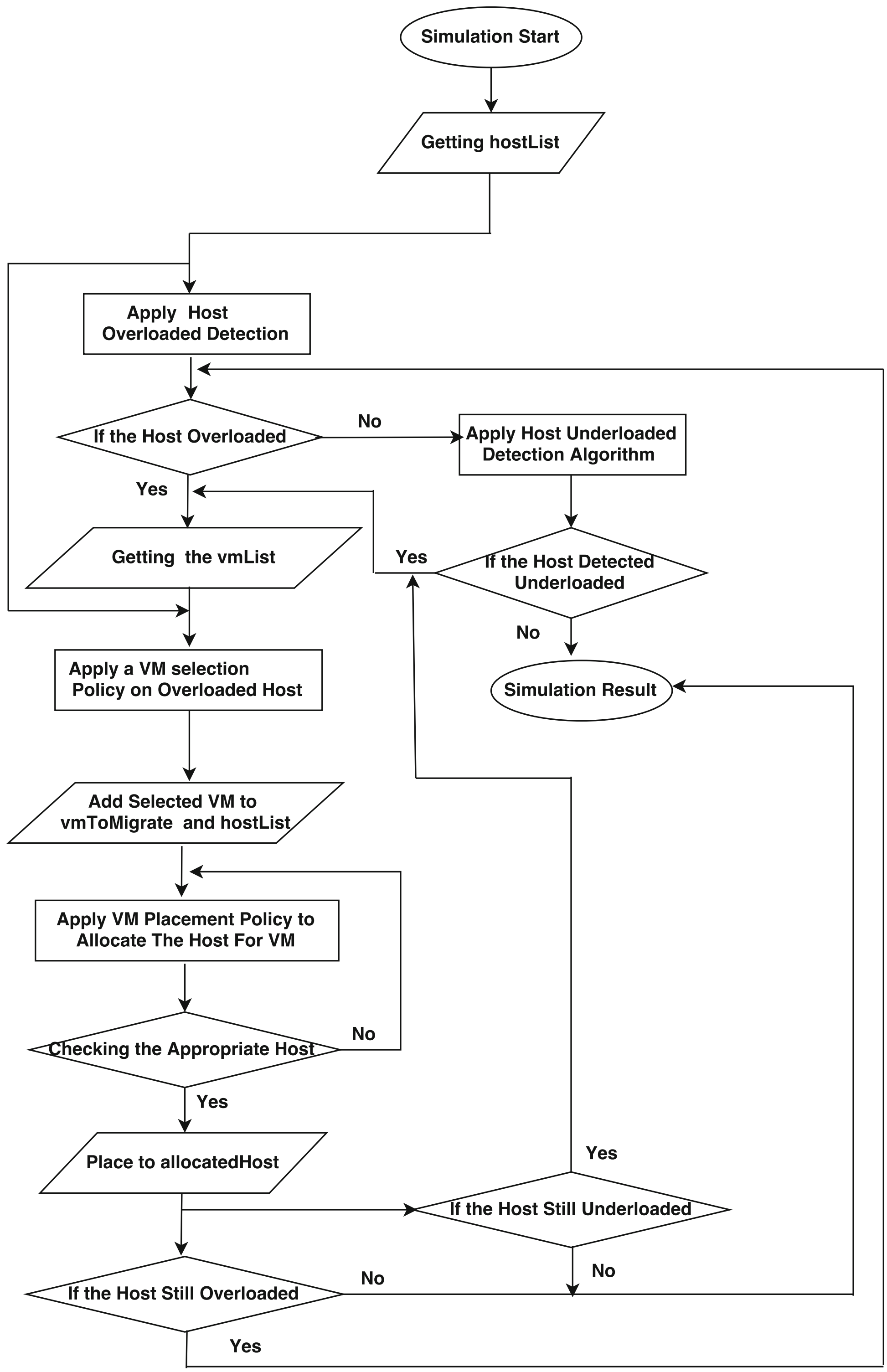


Fig. 2 Proposed flowchart for IaaS resources management

between the value of input data x and the value of the output data y by the straight line as follows:

$$y = \alpha + \beta x + \varepsilon \tag{1}$$

$$\varepsilon = y - (\alpha + \beta x) \tag{2}$$

where α and β are regression model parameters. ε is an independent variable called residuals. We use the estimated regression equation as follows.

$$\hat{y} = a + bx \tag{3}$$

$$\varepsilon_i = y_i - \hat{y}_i \tag{4}$$

where a and b are used to denote the estimates α and β parameters respectively. ε_i is the difference between predicted output variable $\hat{y}_\mathrm{i}$ and real output variable y_i in the i^{th} host CPU utilization data point. This model aims to minimize the value of ε_i. If the value of all residuals ε_i reaches zero, then an optimal model is found for given hosts CPU utilization data points. $i \in H$, where H is set of CPU utilization of all m hosts in the data center. The goal is to estimate the parameters, a and b, which is usually called the intercept and slope of the fitted line in the given data set, respectively. In OLS, a line is fitted in a given dataset by estimating the value of a and b to minimize the sum of squared residuals (SR) as described below:

$$\min_{\alpha,\beta} SR = \sum_{i=1}^{n} \Big(y_i - (a + bx_i) \Big)^2 \tag{5}$$

Let the least square estimator α and β, are a and b respectively. For minimizing the values of α and β, the Eq. (6) partially differentiated with respect to α and β.

$$\begin{aligned} \frac{\partial SR}{\partial \alpha} &= -2 \sum_{i=1}^{m} \Big(y_i - (a + bx_i) \Big) = 0 \\ &= \sum_{i=1}^{m} \Big(y_i - (a + bx_i) \Big) = 0 \end{aligned} \tag{6}$$

$$\begin{aligned} \frac{\partial SR}{\partial \beta} &= -2 \sum_{i=1}^{n} \Big(y_i - (a + bx_i) \Big) x_i = 0 \\ &= \sum_{i=1}^{m} \Big(y_i - (a + bx_i) \Big) x_i = 0 \end{aligned} \tag{7}$$

By simplifying Eq. (7), the value a can be defined as follows:

$$\begin{aligned} &\sum_{i=1}^{m} y_i - \sum_{i=1}^{m} a - \sum_{i=1}^{m} bx_i = 0 \\ &\sum_{i=1}^{m} \alpha = \sum_{i=1}^{m} y_i - \sum_{i=1}^{m} bx_i \\ &ma = \sum_{i=1}^{m} y_i - \sum_{i=1}^{m} bx_i \\ &a = \bar{y} - b\bar{x} \end{aligned} \tag{8}$$

By simplifying Eq. (8) and putting the value of a, the value of b can be defined as follows:

$$b = \frac{\sum_{i=1}^{m} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{m} (x_i - \bar{x})^2} \tag{9}$$

where $\bar{y}$ and $\bar{x}$ are the means of the variable y_i and x_i observations respectively. Formally, the least median of squares fit is determined by the median of the residuals after plotting the values of a and b as follows:

$$LmsReg = \underset{\forall i}{median} \Big(y_i - (a + bx_i) \Big)^2 \tag{10}$$

The overloaded host determines by the upper CPU utilization threshold metric used in [16]. We extended this metric by using *LmsReg* for overloaded host detection as follows:

$$upT = 1 - p \times LmsReg \tag{11}$$

where p (0.5) is a safety parameter of the algorithm and define how fast the system is consolidating the VMs. A small value of the safety parameter p implies low energy consumption but high SLA violation, and vice versa [16].

The pseudo-code in Algorithm 1 gets a $m \times 1$ vector H, which consist of a history of CPU utilization. The values of α and β are obtained by applying OLS. To find an optimal linear regression line we use the parameters α (line 9) and β (10). Afterwards, based on the predicted value of $\hat{y}$, calculating the error or residual for each data point given (line 12). Then, the median of the square of the residuals (line 14), which is helpful in finding the upper CPU utilization threshold, is obtained. If the current CPU utilization of the host is greater than the calculated upper threshold, then this host is detected as an overloaded one and thus violates the SLAs.

Algorithm 1 Least Median of Square Regression (LmsReg)

1: **Input:** H // *History of Hosts CPU utilization*
2: **Output:** Boolean // *Host is overloaded or not*
3: // *Approximate the LmsReg value from regression function based on m hosts utilization history*
4: **for** *i=1 to m* **do**
5: $\mathrm{x_i} \leftarrow$ CpuUtilHistory(i);
6: $\mathrm{y_i} \leftarrow$ CpuUtilHistory(i+1);
7: $\hat{y}_\mathrm{i} = \alpha + \beta\ \mathrm{x_i}$;
8: // *Calculate the* α *and* β
9: $\alpha \leftarrow \bar{y} - \beta\bar{x}$
10: $\beta \leftarrow \frac{\sum_{i=1}^{m}(x_\mathrm{i}-\bar{x})(y_\mathrm{i}-\bar{y})}{\sum_{i=1}^{m}(x_\mathrm{i}-\bar{x})^2}$
11: // *Calculate the residuals of all data points using the updated value of* α *and* β
12: $\epsilon_\mathrm{i} \leftarrow \Big(y_\mathrm{i} - \hat{y}_\mathrm{i} \Big)^2$
13: **end for**
14: *LmsReg* $\leftarrow$ median of ϵ_i
15: *upT* $\leftarrow$ 1- *p*(*LmsReg*)
16: **if** *Current utilization* $>$ *upT* **then**
17: **return** True;
18: **else**
19: **return** False;
20: **end if**

4.2 Minimum utilization prediction VM selection policy (MuP)

The selection of VM is an immediate task after the host detected overload and decision needs to be made to select VMs for migration from overloaded host to appropriate host. Selection of VM efficiently is an important task in both prospective: performance degradation; and SLA violation. We proposed Minimum Utilization Prediction (*MuP*) approach for VM selection. The idea of behind is that, most of the applications are utilizing the different amount of VM CPU at each time frame t. Selection of VM is determine by it CPU utilization over the period of time and select a VM who's CPU utilization is minimum than other VMs on same overloaded host. This approach significantly reduce SLA violation and performance degradation at VM migration time. In this scenario, a proposed algorithm collect CPU utilization of all VM at each time frame t such as $VMu_{m,t-n}$, where m is a total number of VMs and $t-n$ are the time frame $n \in (0, 1, \ldots, n-1, n)$. The collection matrix of all VM CPU utilization (*VMu*) at each time frame t is describe as follows:

$$U = \begin{bmatrix} VMu_{1,t} & VMu_{1,t-1} & \ldots & VMu_{1,t-n} \\ \vdots & \vdots & \ddots & \vdots \\ VMu_{m,t} & VMu_{m,t-1} & \ldots & VMu_{m,t-n} \end{bmatrix}$$

After collected all *VMu*. The proposed algorithm basically analysis past history of each VM CPU utilization acquired by application running on overloaded host VMs at the time frame t. For analysis, we used robust statistics method provides better estimation than other classical statistical methods. The motivation is to produce estimation value of each VM CPU utilization that are not unduly affected by small departures from model assumptions. The Median Absolute Deviation (*MAD*) is a measure of statistical dispersion. It is a more robust estimation technique than the sample variance or standard deviation, as it behaves better with distributions without a mean or variance, such as the normal distribution. The MAD is a robust statistic, being more resilient to outliers in a data set than the standard deviation. In the standard deviation, the distances from the mean are squared, so on average, large deviations are weighted more heavily, and thus outliers can heavily influence it. In the MAD, the magnitude of the distances of a small number of outliers is irrelevant.

For a univariate data set $VMu_{1,t-i}, VMu_{2,t-i}, \ldots, VMu_{m,t-i}$, the *MAD* is defined as the median of the absolute deviations from the datas median:

$$VMu_i^{MAD} = med(|VMu_{i,t-j} - med(VMu_{i,t-j})|) \tag{12}$$

The predicted *MAD* value of each VM on ith host is calculated by the Eq. (13). It,s determine over a period of time which VM workload is lesser than the other VMs.

$$VMu^{MAD}(p) \leq VMu^{MAD}(q), \quad p \in V_i | \forall \in V_i; \tag{13}$$

The choosing a minimum *MAD* value VM using Eq. (14) for migration because application utilizing selected VM resources over the period is much lesser, thereby performance deg radiation and SLA violation occur during VM migration is significantly reduces.

Algorithm 2 MuP VMs Selection

1: **Input:**vmList
2: **output:** Selected VM
3: VmUtilHistory [][] $\leftarrow Null$
4: minMetric $\leftarrow MIN$
5: **for each** i in vmList **do**
6:     **for each** t in n **do**
7:         VmUtilHistory [i][t]+= CPUMips (clock-time(t));
8:     calculate $VMu_i^{MAD}.list \leftarrow$ VmUtilHistory-MAD();
9:     $VMu_i^{MAD}.list$.sort(); //sort by ascending order
10: **return** select minimum value VMu^{MAD} VM

The pseudocode that for understanding the work-flow of *MuP* VMs selection is presented as Algorithm 2. After the overloaded host is detected, *MuP* lists all the VMs from this host. Collected CPU utilization of all VMs is checked at fixed time frame t (line 7) and calculated the MAD value (13) of each VM CPU utilization. At last, select the VM who's MAD value is minimum than the other VMs on same overloaded host. The time complexity of this algorithm is $O(n^2)$.

5 Efficiency metrics

To evaluate the effectiveness of the algorithms, their results are compared with those of other methods by using different metrics. The first metric is called the total electric energy consumed by the data center resources at different application service workloads, and the data are provided by the cloud service provider. The second metric is the average percentage of SLA violation. Such violation occurs when the provisioned VMs are not requested resources (or the average computing power of the shared host is not allocated to the requested VMs). This metric directly influences the level of quality of service, which is not negotiated between the cloud provider and users. If an SLA violation occurs, then the cloud service provider must pay penalty costs to users to compensate for the SLA violation.

5.1 Energy consumption metric

Some components of the computing system (such as CPU, network, and memory) in the data center consume larger amounts of electric energy than do other components. Current studies show that the electric energy consumed by the processor of the host is directly propositional to the utilization of the processor. The utilization of the processor depends on the workload of the host and changes depending on the variability of the workload [23]. Therefore, the utilization of the processor is a function of time. The overall electric energy consumption by the host can be defined as an integral function of power consumed by the host at a given period of time, and this function is described as follows [4]:

$$E = \int_{t_0}^{t_1} P(u(t))dt. \tag{14}$$

where, E represents the total electric energy consumed by the server. $P(u(t))$ is a continuous function of the workload utilization at time t.

Moreover, we consider four different types of hosts namely, Fujitsu M1, Fujitsu M3, Hitachi TS10, and Hitachi SS10. The features of these hosts are shown in Table 2. The energy consumption of the considered servers are obtained from SPECpower [21]. The electric energy consumption of these hosts at different workloads is shown in Table 1.

5.2 SLA violation metric

The SLA defining the quality attributes such as QoS—Quality of Service. SLA metrics are used to measure the performance characteristics of the service objects. The value of SLA violations is vital for the energy-aware algorithms, and this metric could be defined as follows:

$$SLA = \frac{VM^{requested}_{utilization} - VM^{allocated}_{utilization}}{VM^{requested}_{utilization}} \tag{15}$$

where $VM^{requested}_{utilization}$ and $VM^{allocated}_{utilization}$ represents requested MIPS and allocated MIPS by all VMs respectively.

5.3 Performance ratio metric (*PRM*)

The performance ratio metric (PRM) describe the overall performance of proposed algorithm *LmsRegMuP* on CDC. By using this metric is able to understand the efficiency in energy consumption, and SLA violation. The lager value of PRM means overall performance of algorithm is more efficient in CDC. Therefore, if PRM value of proposed algorithm *LmsRegMuP* is greater than other algorithms discussed in related work, the proposed algorithm called more efficient in CDC. We introduced PRM, which is described as follows:

$$PRM = \frac{1}{SLA} + \frac{1}{E} \tag{16}$$

where PRM represents the overall performance ratio metric, and E is the total electric energy consumption of the CDC. *SLA* represents the a percentage of SLA violation in the data center. The next section describe the experiment setup of CDC.

6 Experiment setup

Cloud computing is a large-scale virtualized resources for providing virtual computing resources to cloud service users. We basically focus on IaaS platform. However, deploying a real large-scale virtualized IaaS platform is costly and brings difficulty in conducting repeated experimental analysis and comparing the results of the proposed algorithm. Therefore, simulation is the best choice for evaluating efficiency of proposed algorithms. In this work, we implemented *LmsRegMuP* algorithm on the CloudSim toolkit [28], analyse and comparing the performance of overloaded host detection and VM selection with other literature work. The CloudSim is a modern open source simulator and provides an IaaS cloud computing framework for repeatable experiments.

In CDC simulation setup, four different categories hosts are taken such as Fujitsu M1, Fujitsu M3, Hitachi TS10, and Hitachi SS10. The features of these hosts are shown in Table 2. We are installing 800 heterogeneous hosts of real configuration describe in Table 2. The energy consumption of these hosts at different workload are shown in Table 1.

The CPU clock speeds of the hosts are mapped into MIPS ratings. The cores of Fujitsu M1, Fujitsu M3, Hitachi TS10, and Hitachi SS10 host are mapped as 2700, 3500, 3500, and 3600 MIPS respectively. The network bandwidth of the each host is 1 GB/s. The corresponding VM types are Amazon EC2 VM types, as shown in Table 3.

This simulation-based experiment relies upon real workload traces from real servers that are available publicly. We used data provided by CoMon project for monitoring infrastructure of PlanetLab servers [29]. This set of data was included in CloudSim simulation toolkit for 10 random days shown in Table 4.

These real workload traces contain CPU utilizations of more than 1000 VMs deployed on more than 500 PlanetLab server located at worldwide. In the next section we will analyze the simulation results.

Table 1 The electric energy consumed by the considered servers at different level of workload in Watts (W)

Server	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Fujitsu M1	13.3	18.3	21.1	23.4	26.5	29.6	34.7	40.7	46.8	57.4	60
Fujitsu M3	12.4	16.7	19.4	21.4	23.4	26.1	29.7	34.8	41	47.1	51.2
Hitachi TS10	37	39.9	43.2	45.5	48.8	52.8	57.8	65.1	73.8	80.8	85.2
Hitachi SS10	36	38.8	41.2	43.7	46.3	49.4	53.1	58.8	64.2	67	69.7

7 Simulation result and analysis

We evaluate the performance of *LmsReg* for overloaded host detection and *MuP* for VM selection on considered simulation framework with real CPU utilization data of heterogeneous servers described in Table 4. We get the results of energy consumption metric, total SLA violation metric, number of host shutdown, and number of VMs migration. Similarly, we have collected the results of performance ratio metric. The proposed algorithms are simulated, and their results are analyzed and compared with other algorithms for overloaded host detection and VM selection. The compared algorithms for overloaded host detection are median absolute deviation (Mad), linear regression (Lr), and inter quartile range (Iqr); for comparison of VM selection algorithms, minimum migration time (Mmt), maximum correlation (Mc), and minimum utilization (Mu) are used [16]. We discuss the simulation results and their comparison in the following subsections.

7.1 Energy consumption

Recent studied [16], show that the energy is consumed by hosts in datacentres related to its CPU and memory utilization, even when the DVFS technique is applied. However, with the equipment of multicore CPUs, large-capacity memory, and big hard disk, the traditional linear model is not capable of depicting the energy consumption of a host accurately. The simulation results of the proposed algorithms analyze and compare with *Mad*, *Lr*, and *Iqr* in Beloglazov et al. [16]. The real 10 days workload trace is used in the experiment that are shown in Table 4. The simulation results of the energy consumption are presented in Fig. 3. Notably, energy consumption is significantly reduces using the proposed algorithm for overloaded host detection with minimum utilization prediction called *LmsRegMuP* compared with other techniques describe in literature work. The Fig.3(a) showing superiority of *LmsRegMuP* according to 10 days energy consumption of the CDC. The 10 days variation of energy consumption by using *LmsRegMuP* is from 26 kWh to 41 kWh. In other words, the overall energy consumption of the data center is significantly reduced by using *LmsRegMuP* The energy consumption of the data center by using the techniques in Beloglazov et al. [16] is greater than the median of *LemRegMuP*. We also implement LmsReg for host overloaded detection and MuP for VM detection combined with the algorithms and policies in Beloglazov *et al.* [16], respectively.

The energy consumption of the data center by implementing *LmsReg* with *Mc, Mmt,* and *Mu* in Beloglazov

Table 3 Amazon EC2 VM types

VM Types	MIPS	Memory (MB)
Hight-CPU instance	2500	850
Extra-large instance	2000	3750
Small instance	1000	1700
Micro instance	500	613

Table 4 The characteristics of workload dataset [16]

Date	No. of VMs	Mean (%)	SD (%)
03-03-2011	1052	12.31	17.09
06-03-2011	898	11.44	16.83
09-03-2011	1061	10.7	15.57
22-03-2011	1516	9.26	12.78
25-03-2011	1078	10.56	14.14
03-04-2011	1463	12.39	16.55
09-04-2011	1358	11.12	15.09
11-04-2011	1233	11.56	15.07
12-04-2011	1054	11.54	15.15
20-04-2011	1033	10.43	15.21

Table 2 The characteristics of the hosts

Server	CPU	Core	Clock speed (GHz)	Memory (GB)
Fujitsu M1	Xeon 1230	4	2.7	8
Fujitsu M3	Xeon 1230	4	3.5	8
Hitachi TS10	Xeon 1280	4	3.5	8
Hitachi SS10	Xeon 1280	4	3.6	8

et al. [16] is also compared. The performance of *MuP* with *Mad, Lr,* and *Iqr* in Beloglazov et al. [16] is compared as well. Figure 3(b) shows the implementation of *MuP* with *Mad, Iqr,* and *Lr*. The figure shows that the energy consumption of the data center is significantly reduced by the proposed algorithms compared with the combined ones.

7.2 SLA violation

High upper utilization threshold for detecting overloaded hosts allows aggressive VM consolidation, thereby directly influencing the cost of SLA violation.

As shown in Fig. 4, the percentage of SLA violation of the proposed algorithms is significantly less than that of the combined techniques. The average median of SLA violation results of the combined algorithms is 9.9%. Therefore, the average percentage of SLA violation of the proposed algorithms is 6.9%, which is 30% less than that of the combined algorithms.

Figure 4(b) shows that *LrMuP*, *MadMuP*, and *IqrMuP* do not reduce the average percentage of SLA violation. Therefore, the proposed algorithms reduce the significant amount of energy consumption and average percentage of SLA violation simultaneously.

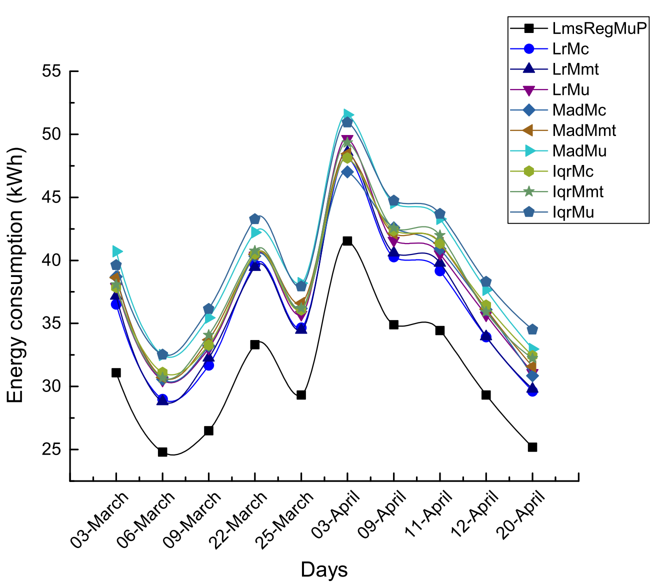
7.3 PRM (performance ratio metric)

We discuss the overall performance of the CDC by using PRM. The cloud provider aims to maximize the overall performance with minimized electric energy consumption, and SLA violation. Figure 5(a) shows that the PRM of proposed heuristic algorithm is grater than other algorithm PRM. Therefore, *LmsRegMuP* performance on considered CDC is more efficient. Similarly, we also implemented *LmsReg* host overloaded detection algorithm with *mc, mmt*, and *mu*. The Fig. 5(b) Shows that PRM of *LmsRegMc* is grater than other algorithms combination.

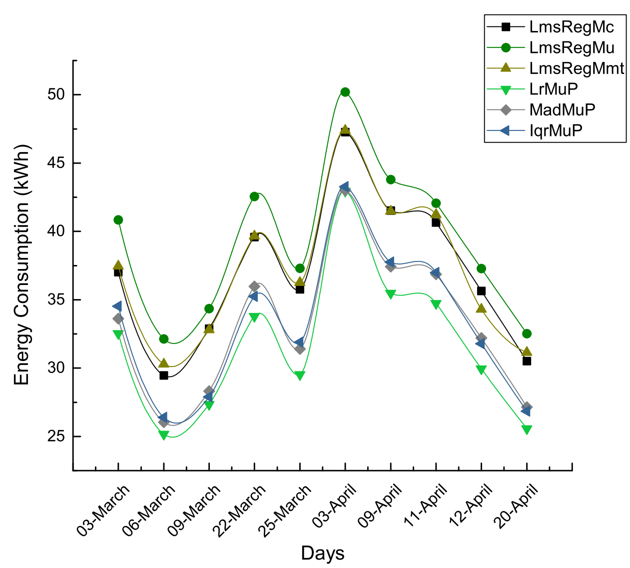
7.4 Number of hosts shutdown

The simulation results on the number of shutdown hosts is analysed and compared. If the number of reactivated hosts increases, thereby the consumption of energy increases as well. The hosts are reactivated for allocating new VMs and are shut down when detected as idle.

We collect the data on the number of hosts turning to energy-saving mode by implementing the proposed algorithms during simulation. As shown in Fig. 6(a), the approximate variation of shutdown host is 1400 to 2200 of the proposed algorithms, and this value is much lesser than other considered algorithms. The simulation results indicate that the proposed algorithms work better than does other algorithms because of the few number of reactivated hosts. Few number of reactivated hosts is directly proportional to energy consumption of the data center. In the experimental environment, we use only 800 hosts. However, the number of shut-down hosts is high because of reactivation of hosts. Fig. 6(a), and (b) shows that using the proposed algorithms significantly minimizes of host reactivation compared with traditional algorithms (Fig. 7).



(a)



(b)

Fig. 3 Comparison of energy consumption on real workload traces of PlanetLab

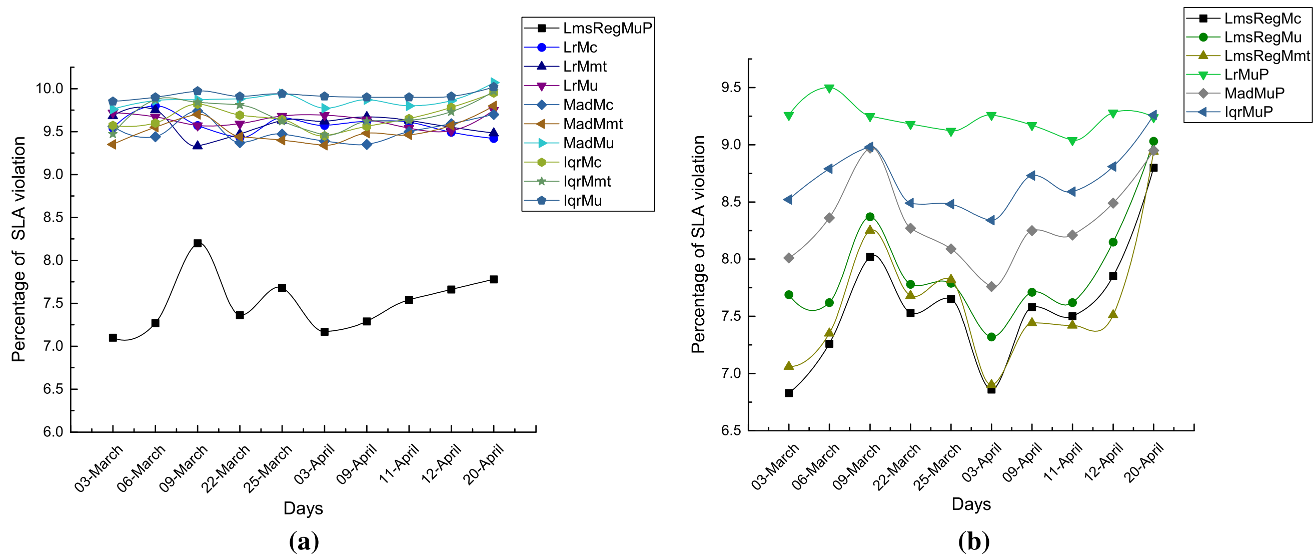


Fig. 4 Comparison of total SLA violation using real workload traces of PlanetLab

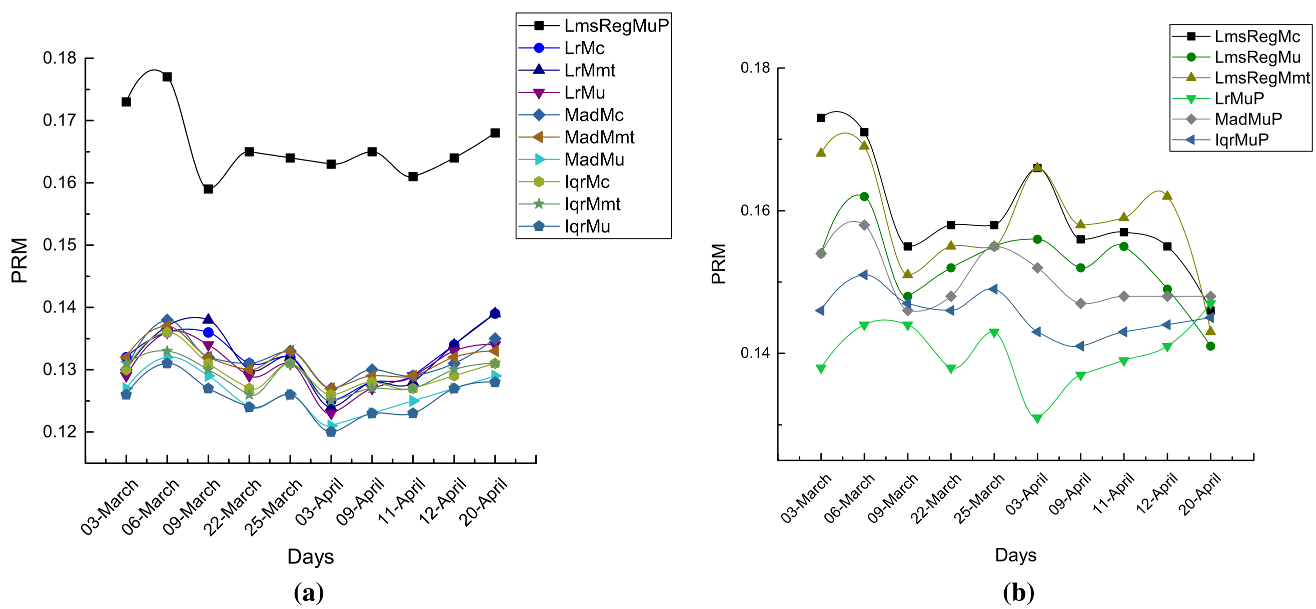


Fig. 5 Comparison of performance ratio metric using real workload traces of PlanetLab

7.5 Number of VM migration

Total VM migration is an important parameter for real environment of cloud computing technology, which involves a huge bandwidth cost for live migrations of VMs. The simulated results of the number of VM migrations by using the proposed algorithms are obtained. The number of VM migrations is directly proportional the performance degradation of VMs during active state of CPU utilization. High number of VM migrations directly influences the SLAs, and such effect is unfavorable for the cloud provider and users. Figure 6(a) and (b) show that the number of VM migrations by using the proposed algorithms is much lesser than that of the algorithms in Beloglazov *et al.* [16]. Therefore, the performance degradation substantially decreases because it is directly proportional to the number of VM migrations.

8 Test of significance

The combination of four overloading detection methods (LmsReg, Lr, Iqr, and Mad) with 4 VM selection policies (MuP, Mc, Mu, and Mmt) are simulated on CloudSim

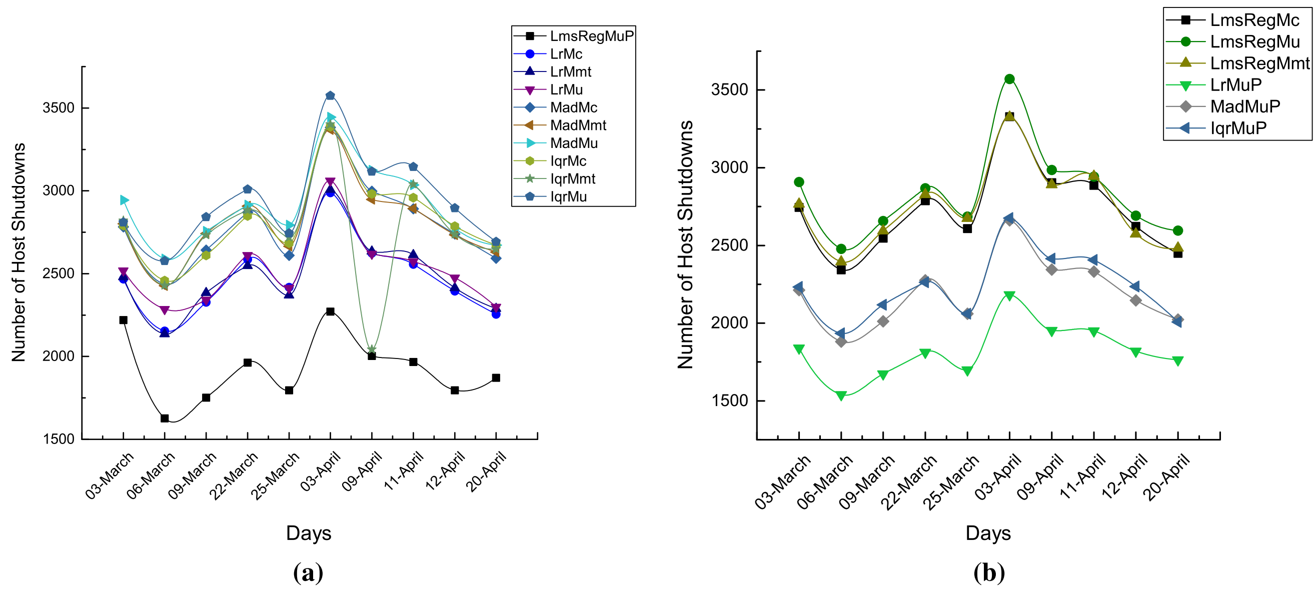


Fig. 6 Comparison of number of hosts shutdown using real workload traces of PlanetLab

toolkit. Moreover, we have taken safety parameter to be 0.5, 1.2, 1.5, and 2.5 for LmsReg, Lr, IQR, and Mad, respectively, to control aggressiveness of the algorithms for consolidating VMs. To perform the test of significance on proposed methods, we select Shapiro–Wilk test [30] of the tests yield p value > 0.05 for all the proposed combinations and pass the test of normality validating that results are statistically significantly different. The proposed algorithms not only reduce the energy consumption but also obtain minimize VM migrations and host shutdowns.

9 Conclusions and future directions

The rapid growth of CDCs worldwide has led to enormous amount of electric energy consumption, thereby increasing CO_2 emissions. This work focuses on the cloud IaaS platform in two ways. First, the overall electric energy consumption, which directly impacts the running cost of the CDC, is minimized. Thus, this study provides a real environment of cloud computing for growing its industries all over the world. Second, SLA violation, number of reactivated hosts, and number of VM migrations, are

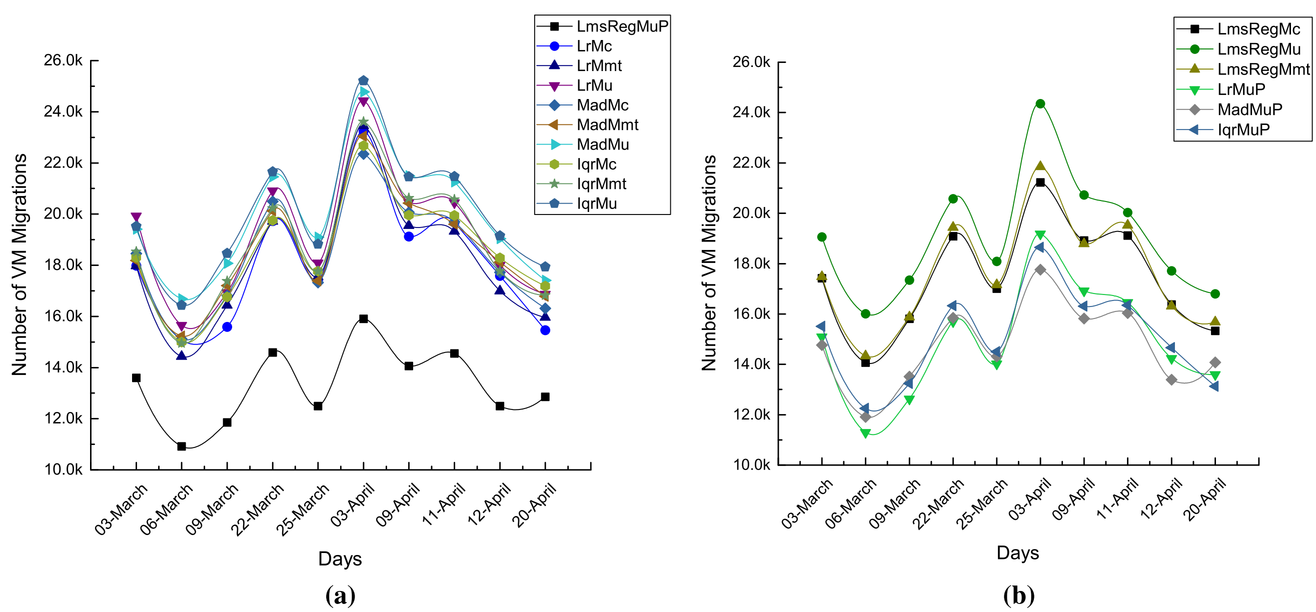


Fig. 7 Comparison of number of VM migration using real workload traces of PlanetLab

reduced. Moreover, a novel algorithm for overloaded host detection called *LmsReg* and a policy for VM selection from overloaded hosts called MuP are proposed. The proposed algorithms are implemented using the CloudSim simulator. The experimental results show that these approaches substantially minimize the electric energy consumption of CDC compared with previous approaches for overloaded host detection and VM selection.

We will implement and evaluate the results on the open source cloud platform called Open-Stack to analyze the effectiveness of the proposed algorithms in a real cloud platform. A few complex machine learning models, such as gradient decent, will also be considered. New algorithms for minimizing the energy consumption of data center under SLA will be developed as well.

Acknowledgements The National Key Research and Development Plan under Grant No. 2017YFB0801801, the National Science Foundation of China (NSFC) under Grant Nos. 61672186, 61472108, support this work.

References

1. Lambert, S., Van Heddeghem, W., Vereecken, W., Lannoo, B., Colle, D., & Pickavet, M. (2012). Worldwide electricity consumption of communication networks. *Optics Express*, *20*(26), B513–B524.
2. Barroso, L. A., & Hölzle, U. (2007). The case for energy-proportional computing. *Computer*, *40*(12), 33–37.
3. Fawaz, A.-H., Peng, Y., Youn, C.-H., Lorincz, J., Li, C., Song, G., et al. (2018). Dynamic allocation of power delivery paths in consolidated data centers based on adaptive ups switching. *Computer Networks*, *144*, 254–270.
4. Beloglazov, A., Abawajy, J., & Buyya, R. (2012). Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Generation Computer Systems*, *28*(5), 755–768.
5. Ahmed, A., Hanan, A. A., Omprakash, K., Usman, M., & Syed, O. (2017). Mobile cloud computing energy-aware task offloading (mcc: Eto). In *Proceedings of the communication and computing systems: Proceedings of the international conference on communication and computing systems (ICCCS 2016)* (p. 359).
6. Xu, C., Wang, K., Li, P., Xia, R., Guo, S., & Guo, M. (2018). Renewable energy-aware big data analytics in geo-distributed data centers with reinforcement learning. *IEEE Transactions on Network Science and Engineering*, *PP*(99), 1–1.
7. Yadav, R., Zhang, W., Chen, H., & Guo, T. (2017). Mums: Energy-aware vm selection scheme for cloud data center. In *28th International workshop on database and expert systems applications (DEXA), 2017* (pp. 132–136). IEEE.
8. Hu, X., Li, P., Wang, K., Sun, Y., Zeng, D., & Guo, S. (2018). Energy management of data centers powered by fuel cells and heterogeneous energy storage. In *2018 IEEE international conference on communications (ICC)* (pp. 1–6). IEEE.
9. Wang, M., Meng, X., & Zhang, L. (2011). Consolidating virtual machines with dynamic bandwidth demand in data centers. In: *INFOCOM, 2011 Proceedings IEEE* (pp. 71–75). IEEE.
10. Kaiwartya, O., Abdullah, A. H., Cao, Y., Lloret, J., Kumar, S., Shah, R. R., et al. (2018). Virtualization in wireless sensor networks: Fault tolerant embedding for internet of things. *IEEE Internet of Things Journal*, *5*(2), 571–580.
11. Feller, E., Morin, C., & Esnault, A. (2012). A case for fully decentralized dynamic vm consolidation in clouds. In *IEEE international conference on cloud computing technology and science* (pp. 26–33).
12. Esfandiarpoor, S., Pahlavan, A., & Goudarzi, M. (2015). Structure-aware online virtual machine consolidation for datacenter energy improvement in cloud computing. *Computers & Electrical Engineering*, *42*, 74–89.
13. Murtazaev, A., & Oh, S. (2011). Sercon: Server consolidation algorithm using live migration of virtual machines for green computing. *IETE Technical Review*, *28*(3), 212–231.
14. Feller, E., Morin, C., & Esnault, A. (2012). A case for fully decentralized dynamic vm consolidation in clouds. In *IEEE 4th international conference on cloud computing technology and science (CloudCom), 2012* (pp. 26–33). IEEE.
15. Ranganathan, P., Leech, P., Irwin, D., & Chase, J. Ensemble-level power management for dense blade servers. In *ACM SIGARCH computer architecture news* (Vol. 34(2), pp. 66–77). IEEE Computer Society.
16. 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. *Concurrency and Computation: Practice and Experience*, *24*(13), 1397–1420.
17. Verma, J. K., Kumar, S., Kaiwartya, O., Cao, Y., Lloret, J., Katti, C., et al. (2018). *Enabling green computing in cloud environments: Network virtualization approach toward 5g support* (p. e3434). London: Transactions on Emerging Telecommunications Technologies.
18. Zhu, X., Young, D., Watson, B. J., Wang, Z., Rolia, J., Singhal, S., McKee, B., Hyser, C., Gmach, D., & Gardner, R. *et al.* (2008). 1000 islands: Integrated capacity and workload management for the next generation data center. In: *International conference on autonomic computing, 2008. ICAC'08.* (pp. 172–181). IEEE.
19. Kusic, D., Kephart, J. O., Hanson, J. E., Kandasamy, N., & Jiang, G. (2009). Power and performance management of virtualized computing environments via lookahead control. *Cluster Computing*, *12*(1), 1–15.
20. von Kistowski, J., & Kounev, S. (2016). Univariate interpolation-based modeling of power and performance. In *Proceedings of the 9th EAI international conference on performance evaluation methodologies and tools* (pp. 212–215). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
21. All published specpowerssj2008 results. https://www.spec.org/power_ssj2008/results/power_ssj2008.html. Accessed May 12, 2017.
22. Nathuji, R., & Schwan, K. (2007) Virtualpower: Coordinated power management in virtualized enterprise systems. In *ACM SIGOPS operating systems review* (Vol. 41(6), pp. 265–278). ACM.
23. Yadav, R., & Zhang, W. (2017). MeReg: Managing energy-SLA tradeoff for green mobile cloud computing. *Wireless Communications and Mobile Computing*, *2017*, 6741972.
24. Farahnakian, F., Ashraf, A., Pahikkala, T., Liljeberg, P., Plosila, J., Porres, I., et al. (2015). Using ant colony system to consolidate vms for green cloud computing. *IEEE Transactions on Services Computing*, *8*(2), 187–198.
25. Farahnakian, F., Liljeberg, P., & Plosila, J. (2013). Lircup: Linear regression based cpu usage prediction algorithm for live migration of virtual machines in data centers. In: *Euromicro conference on software engineering and advanced applications* (pp. 357–364).

26. Mili, L., Phaniraj, V., & Rousseeuw, P. J. (1991). Least median of squares estimation in power systems. *IEEE Transactions on Power Systems*, *6*(2), 511–523.
27. Edelsbrunner, H., & Souvaine, D. L. (1990). Computing least median of squares regression lines and guided topological sweep. *Journal of the American Statistical Association*, *85*(409), 115–119.
28. Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A., & Buyya, R. (2011). Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and experience*, *41*(1), 23–50.
29. Park, K., & Pai, V. S. (2006). Comon: A mostly-scalable monitoring system for planetlab. *ACM SIGOPS Operating Systems Review*, *40*(1), 65–74.
30. Shapiro, S. S., & Francia, R. (1972). An approximate analysis of variance test for normality. *Journal of the American Statistical Association*, *67*(337), 215–216.

Rahul Yadav is a Ph.D student at the School of Computer Science and Technology, Harbin Institute of Technology, China. He has received M.S. degree in Computer Science from the Department of Computer Science and Mathematics, South Asian University, New Delhi, India, in July 2015. Since September 2015, he has been doctoral candidate and actively involved in research on efficient-energy management, cloud computing, optimal utilization of data center resources, machine learning, and cost-efficient virtual machine consolidation. He has published conference and journal papers.

Weizhe Zhang is currently a professor in the School of Computer Science and Technology, Harbin Institute of Technology, China. His research interests are primarily in parallel computing, distributed computing, cloud and grid computing, and computer network. He has published more than 100 academic papers in journals, books, and conference proceedings. He serves on a number of journal editorial boards. He has edited more than 10 international journal special issues as a guest editor and has served many international conferences as a chair or committee member. He is a member of the IEEE.

Keqin Li is a SUNY Distinguished Professor of computer science in the State University of New York. He is also a Distinguished Professor of Chinese National Recruitment Program of Global Experts (1000 Plan) at Hunan University, China. He was an Intellectual Ventures endowed visiting chair professor at the National Laboratory for Information Science and Technology, Tsinghua University, Beijing, China, during 2011–2014. His current research interests include parallel computing and high-performance computing, distributed computing, energy-efficient computing and communication, heterogeneous computing systems, cloud computing, big data computing, CPU–GPU hybrid and cooperative computing, multi-core computing, storage and file systems, wireless communication networks, sensor networks, peer-to-peer file sharing systems, mobile computing, service computing, Internet of things and cyber-physical systems. He has published over 520 journal articles, book chapters, and refereed conference papers, and has received several best paper awards. He is currently or has served on the editorial boards of IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Computers, IEEE Transactions on Cloud Computing, IEEE Transactions on Services Computing, and IEEE Transactions on Sustainable Computing. He is an IEEE Fellow.

Chuanyi Liu born in May 1982, became member of IEEE in 2011. He received his Ph.D. (2009) in computer science and technology from Tsinghua University, Beijing, China. His research interests include computer architecture, cloud security, and data protection. Chuanyi spent one year as a visiting scholar at the Digital Technology Center of the University of Minnesota, Minneapolis, USA. He is now associate professor of Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, China. He has published more than 30 papers in related conferences and journals.

Muhammad Shafiq Muhammad Shafiq was born in Pakistan. He received the B.S. degree with honor rank in computer science from the Faculty of Computer Science, Malakand University, Chakdara, Pakistan, in 2009, and the M.S. degree in computer science from Faculty of Computer Science, Malakand University, Chakdara, Pakistan, in 2011. He is currently pursuing the Ph.D. degree at the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China. His current research areas of interests include Network Traffic Classification and Network Security.

Nabin Kumar Karn is PhD student in School of Computer Science and Technology at Harbin Institute of Technology Harbin China. His research interests include measuring Internet performance, topology measurement and management, Network Security and Traffic classification. Earlier he had worked 4 years in ISP as a network engineer. He received ME in Computer Science and Technology from Harbin Institute of Technology in 2015. He has published many EI papers.