Hindawi
Wireless Communications and Mobile Computing
Volume 2017, Article ID 6741972, 11 pages
https://doi.org/10.1155/2017/6741972

Research Article

MeReg: Managing Energy-SLA Tradeoff for Green Mobile Cloud Computing

Rahul Yadav and Weizhe Zhang

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

Correspondence should be addressed to Rahul Yadav; rahul@stu.hit.edu.cn and Weizhe Zhang; wzzhang@hit.edu.cn

Received 1 August 2017; Revised 13 October 2017; Accepted 1 November 2017; Published 17 December 2017

Academic Editor: Javier Bajo

Mobile cloud computing (MCC) provides various cloud computing services to mobile users. The rapid growth of MCC users requires large-scale MCC data centers to provide them with data processing and storage services. The growth of these data centers directly impacts electrical energy consumption, which affects businesses as well as the environment through carbon dioxide (CO_2) emissions. Moreover, large amount of energy is wasted to maintain the servers running during low workload. To reduce the energy consumption of mobile cloud data centers, energy-aware host overload detection algorithm and virtual machines (VMs) selection algorithms for VM consolidation are required during detected host underload and overload. After allocating resources to all VMs, underloaded hosts are required to assume energy-saving mode in order to minimize power consumption. To address this issue, we proposed an adaptive heuristics energy-aware algorithm, which creates an upper CPU utilization threshold using recent CPU utilization history to detect overloaded hosts and dynamic VM selection algorithms to consolidate the VMs from overloaded or underloaded host. The goal is to minimize total energy consumption and maximize Quality of Service, including the reduction of service level agreement (SLA) violations. CloudSim simulator is used to validate the algorithm and simulations are conducted on real workload traces in 10 different days, as provided by PlanetLab.

1. Introduction

Mobile devices, such as smartphones and tablets, are becoming essential to human life as the most effective computational and convenient communication tools are not bounded by time and place. These devices are replacing desktop or laptop computers by using the cloud computing environment or mobile cloud computing (MCC). The MCC is a combined infrastructure of cloud computing and mobile computing in which data processing and storage are performed on the cloud, and mobile devices are mainly used as client to communicate with the application and retrieve processed results from the cloud [1]. The rapid growth of mobile computing usage is evident in the study of Juniper Research, which states that the consumer and enterprise market for cloud-based mobile applications increased to $9.5 billion by 2014 [2], directly impacting cloud infrastructure. Cloud computing is leveraged on existing technologies and ideas, such as data centers and virtualization technology. This new perspective revolutionized traditional information technology (IT) business by helping developers and companies overcome lack of hardware capacity (such as CPU, memory, and storage) by allowing users to access on-demand resources through the Internet [3, 4].

Cloud computing is mainly divided into three types of service models, namely, Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Moreover, cloud computing has four types of deployment models such as private, public, hybrid, and community clouds [5, 6]. Provision of MCC services to users requires large-scale cloud computing platform, which drains enormous amount of electric power and increases MCC operational costs, CO_2 emissions. Data centers consume approximately 1.3% of the total worldwide electricity supply, which is predicted to increase to 8% by 2020 [7]. Therefore, CO_2 also increase substantially, which directly impacts the environment. Unfortunately, large amounts of electrical energy are wasted by servers during low workload. The server resources

utilization data collected from more than 5000 production servers over a six-month period have shown that most of the time servers operate at 10% to 50% of their full capacity, leading to wasting the energy on low utilization of resources [8].

The Quality of Service (QoS) constraint plays an important role between mobile cloud service providers and users. Meeting QoS requirements is determined via Service Level Agreements (SLAs) that describe the required performance levels, such as minimal throughput and maximal response time or latency of the system. Therefore, the main challenge is to minimize power consumption of mobile cloud data centers while satisfying QoS requirements [9].

Hardware virtualization technology transforms traditional hardware to the new paradigm. This technology consolidates workload, called virtual machine (VM) consolidation, and exploits low-power hardware states. Most current studies have minimized the overall energy consumption through two widely used techniques, such as VM consolidation and dynamic server provisioning [10, 11]. Dynamic server provisioning methods reduce electric power consumption by reducing the computational resources during low workloads [12]. This reduction means turning the unnecessary servers to sleep-mode when the workload demand decreases. Similarly, when data processing and data storage demands increase, these servers are reactivated according to requirements [13, 14]. The server shares its resources among multiple performance-isolated platforms called VMs by using hypervisor technology. Each VM runs more than one task simultaneously. Dynamic VM consolidation also plays an important role in minimizing overall energy consumption in mobile cloud data centers. The VM consolidation occurs when a server (host) detects overload or underload, during which VM migrates one by one from the overloaded host to another appropriate host until the overload returns to its normal state. Similarly, when the host detects underload, all VMs migrate to appropriate hosts and turn this host to sleep-mode [15, 16]. Basically, these approaches have two main objectives: minimizing overall energy consumption and maximizing the QoS. The QoS requirements are formalized via SLA metric and such features are described as minimal throughput and maximal response time or latency delivered by the deployed system [17].

The basic task of efficient energy consumption in mobile cloud data centers is divided into five parts as follows:

(1) Determine when a host is considered overloaded so that some VMs would migrate one by one to other efficient hosts under SLA constraint until the host returns to normal state. To detect overloaded hosts, we used *MeReg* algorithm, which is introduced in this paper.

(2) Determine when a host is considered underloaded so that all VMs would migrate from it to the appropriate hosts and it will turn into sleep-mode. To detect underloaded host, we used constant lower CPU utilization threshold proposed in Beloglazov and Buyya [18].

(3) Select VMs from an overloaded host that should have migrated from it. To select, we used our previous work in Yadav et al. [19].

(4) Select all VMs from an underloaded host that should have migrated from it. To select, we used our previous work in Yadav et al. [19].

(5) Find a new VM allocation where selected VMs from overloaded and underloaded hosts would be placed to activate or reactivate hosts. We used the modified best fit decreasing (MBFD) algorithm proposed in Beloglazov et al. [16] for VM placement.

In this study, we proposed a regression-based adaptive heuristic algorithm for estimating an upper threshold to detect the overloaded hosts of mobile cloud data center. From these hosts, several VMs are migrated to another host to minimize the performance degradation. We used a novel MuMs dynamic VM selection algorithm to balance trade-offs among electric power consumption, number of migrations, performance of host, and total number of hosts that were shut down. These algorithms estimate the upper threshold and selection of VMs based on the statistical analysis of CPU utilization history of hosts. The following are the main contributions of this paper:

(i) An adaptive heuristic *MeReg* algorithm to estimate upper CPU utilization threshold using recent CPU utilization history for detecting overloaded hosts is introduced. This algorithm mainly aims to minimize overall power consumption under the required SLA of mobile cloud data center.

(ii) The performance and effectiveness of the *MeReg* algorithm are evaluated using the CloudSim simulator on real and random workload traces and compared with other proposed approaches in the literature.

The rest of this paper is organized as follows: In Section 2, we discussed some previous literature related to mobile cloud data center resources and energy efficiency management. In Section 3, we presented the mobile cloud platform architecture. Section 4 is a key part of this paper where we discussed host overload detection. In Section 5, we proposed energy efficiency metric for measuring the effectiveness of the proposed algorithms in the cloud environment. In Section 6, the experiment setup for proposed algorithms is discussed. In Section 7 results of the proposed algorithms are analysed and compared, and in Section 8, the study is concluded by a summary with future research direction.

2. Related Work

Researchers have examined the design of mobile cloud models and its associated software architecture [20]. A paradigm shift is evident from traditional to mobile cloud computing which requires large-scale of cloud data center, wherein the cost of computational resources is no longer the major portion of the overall cost. However, the cost of power consumption and cooling infrastructure are still considered primary cost drivers. Power consumption and CPU

utilization in servers or mobile are directly proportional to one another [21, 22]. Therefore, recent techniques for minimizing power consumption and maximizing QoS are discussed in this study. In one of the first works introduced by Zhang et al. [23], dynamic efficient energy techniques for mobile computing that schedule multiple computing tasks are dynamically reconfigured and selectively turned off to minimize overall energy consumption in mobile computing.

Esfandiarpoor et al. proposed a VM consolidation algorithm that efficiently reduces energy in cloud data center by considering structural features, such as racks and network topology. Moreover, they focused on the cooling and network structure of cloud data center hosting the physical machines when consolidating VMs. Few racks and routers are employed without compromising the SLA so that idle routing and cooling equipment could be turned off to reduce energy consumption [24]. Zhu et al. [25] investigated the dynamic VM consolidation problem and applied a static host CPU utilization threshold of 85%, which is determined if the host is overloaded when CPU utilization threshold exceeded 85%. However, static CPU threshold is unsuitable for systems with dynamic workload, as this static model does not adapt to system workload changes. In this study, we introduced a dynamic adapt threshold value according to the statistical analysis of workload history.

Nathuji and Schwan [26] proposed dynamic VM consolidation to minimize the energy consumption of hosts in data centers. They investigated energy management techniques in the large-scale virtualized resources of data center. They proposed a new energy management method for virtualized resources of data center called Soft Resource Scaling. In addition, the authors suggested dividing the resource management problem into two levels: local and global. At the local level, the algorithms handle the energy management of guest VMs. By contrast, global policies coordinate multiple physical machines. They also explored the benefits of efficient energy consumption using live migration and found that total energy consumption can be significantly reduced.

Beloglazov et al. [16] proposed a cloud computing architectural framework and the provision of mobile cloud data center resources in power efficient manner, while meeting SLA requirements. They established two parts of the VM consolidation problem: (1) submission of new requests for VM provisioning and allocation of VMs on hosts; (2) significant use of current VM allocations. To solve the problem of VM placement on hosts, they used the MBFD algorithm. This algorithm first sorts current CPU utilization of all VMs in decreasing order and allocates each VM to a host, which provides efficient energy consumption environment. In another work, Beloglazov and Buyya [18] introduced a heuristic-based energy-aware approach, which focused on the statistical analysis of CPU utilization history to determine an upper threshold for detecting overloaded hosts

Ranganathan et al. [27] described server power management method at the collective systems level instead of the individual server level. This approach permits active servers to borrow power from inactive servers. Similarly, Venkatachalam et al. [28] introduced an efficient energy technique for minimizing the overall energy consumed by the server CPU at a given period. They also focused on GPU electric energy consumption.

The energy consumption of the data centers is broken down in [29, 30]. Most studies have considered energy consumption modeling at the CPU level: however, network devices also consume considerable amount of energy in terms of data center energy consumption. Therefore, load balancing of data center network devices is important to minimize the energy consumption cost. Shang et al. [31, 32] introduced a distributed green-routing algorithm which consider computation, communication, and thermal temperature within the data center. The future decision of the proposed load-balancing algorithm requires a full energy model including networks and servers in the data center. Liu et al. [33] introduced a distributed flow scheduling (DFS) for efficient energy consumption in data center network devices. However, this approach did not consider the nature of communication sources, sinks, and corresponding computation.

3. System Architecture

The general architecture of MCC includes mobile devices, network connection, and cloud computing data center. In Figure 1, mobile devices are directly connected to the base station using the mobile network. The base station establishes and controls the air connection between mobile devices and the network [34] and communicates with the cloud data center via the Internet to complete the task of the mobile users such as data processing and storage. The cloud data center includes numerous virtualized resources to improve performance of the services. These resources consist of n heterogeneous hosts. Wherein each host contains multicore CPU, primary memory, secondary memory, and network I/O. The CPU performance is determined in terms of millions of instruction per second (MIPS). The submission of multiple requests for VM provisioning is allocated to hosts simultaneously. The allocation of VMs to hosts is based on CPU utilization of the host. The energy consumed by the CPU is linearly proportional to its utilization [18]. Therefore, efficient consolidation of VM would reduce the electric energy consumption and the SLA violation rate. When the running VM cannot obtain its resources from the cloud data center such as MIPS and memory, then SLA violation would occur. In this case, a cloud service provider should pay cloud service users penalty, when an overloaded host is confirmed. The next step is selecting VMs for migration from the overloaded host to appropriate host and apply iteratively to the host until it is no longer considered overloaded.

In this MCC model, three main important players handle all workflows within cloud data centers. The key players are global controller, local controller, and virtual machine manager (VMM). A local controller resides in each host as a separate VM and is tasked to monitor the status of the VM, and CPU utilization as well as decide what time VM should be migrated from the host. The global controller resides on a single master host and gathers all information from the local controller to maintain overall resources utilization. Moreover, it decides where VM should be optimally placed. Finally, the VMM resides along the hypervisor and helps in resizing the

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

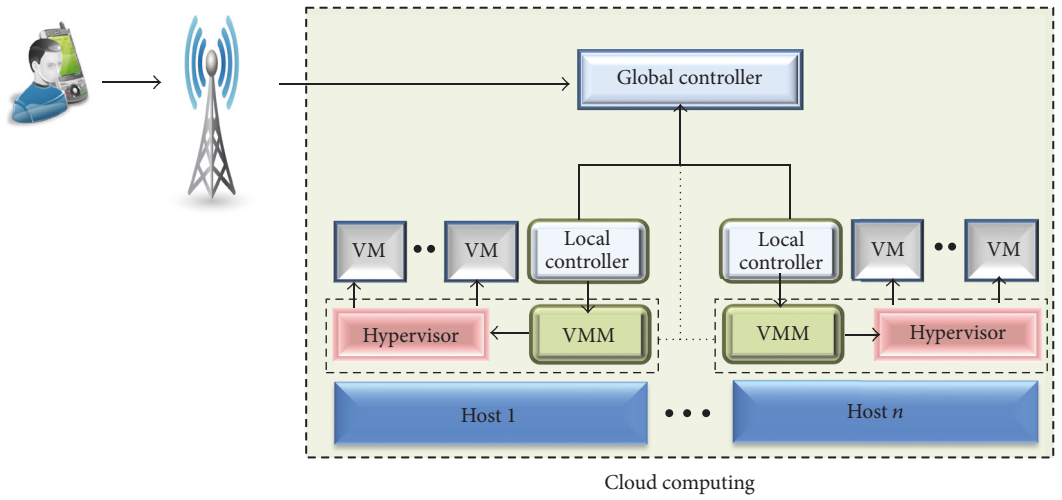


Figure 1: Mobile cloud computing system architecture.

VM and changes the power state of the host, which helps efficiently utilizing energy.

3.1. Energy Model. Relative to other types of equipment, the major energy consumers of mobile cloud data center components are CPU, network, and memory. Recent works show that the electric power consumed by the host's processor is directly proportional to its utilization. Utilization of the processor depends on the workload of the host and changes according to the variability of the workload [35]. Therefore, utilization of the processor is a function of time, and its value changes according to workload variability. The overall electric energy consumption by the host can be defined as an integral function of the power consumed by the host at a given period and is described as follows [16]:

$$E = \int_{t_0}^{t_1} P\left(u\left(t\right)\right) dt, \tag{1}$$

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

Moreover, we considered 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 these servers is obtained from the SPECpower [36]. The electric energy consumption of these hosts at different workloads is shown in Table 1.

Table 2: Characteristics of the hosts.

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

4. *MeReg* Host Overloaded Detection

The mobile cloud computing platform has recently become popular worldwide because of its dynamic nature. However, the dynamic characteristics of mobile cloud computing pose a big concern for cloud service provider (CSP). Therefore, the constant CPU utilization threshold is unsuitable for detecting an overloaded host in cloud environments. We proposed a novel algorithm for host overload detection based on a regression model called *M* estimator regression model. This algorithm dynamically estimates the upper CPU utilization threshold based on the historical dataset of CPU utilization, which is automatically adjusted according to the historical CPU workload.

Robust regression techniques provide more efficient optimal solution than traditional approaches. These techniques are not directly influenced by the outlier in the dataset, which makes it more robust and trustworthy for the dynamic environment of the cloud. The "M estimation Regression" *(MeReg)* generates a regression line in which the median

of the squared residuals is minimized [37]. The *MeReg* is a more robust estimator than the median, standard deviation, variance, and ordinary least squares estimators. "Ordinary least squares (OLS) have the following disadvantages: (1) a single corrupt data point can give the resulting regression line an arbitrarily large slope; (2) it can behave badly when the residual distribution is not normal, particularly when the residuals are heavily tailed" [38, 39]. To initialize the *MeReg* algorithm, we first need to generate the OLS model representing the relationship between input data X and the value of the output data Y using line the straight as follows:

$$\begin{aligned} Y_i &= \theta_1 + \theta_2 X_i + \varepsilon_i, \\ \varepsilon_i &= Y_i - (\theta_1 + \theta_2 X_i), \end{aligned} \tag{2}$$

where ε_i is the independent variable called residuals. This model mainly aims to minimize the value of residuals ε_i. If the values of all residuals ε_i converge to the zero, then an optimal model is generated, wherein all given data points lie on this model. $i \in V$, where V is set of all VMs CPU utilization dataset of the data center. The goal is to minimize the sum of distance between the estimated linear parameter and actual CPU utilization data point. The objective function of estimation can be defined as follows:

$$\begin{aligned} \min \mathcal{F}(\widehat{\varepsilon}_i) &= \sum_{i=1}^{m} \frac{(Y_i - (\theta_1 + \theta_2 X_i))}{\sigma}, \\ \sigma &= \frac{\text{median}\left|\varepsilon_i - \text{median}(\varepsilon_i)\right|}{0.6745}, \end{aligned} \tag{3}$$

where σ represents a residuals standard deviation of CPU utilization data point. To make this model more robust, Tukey's bisquare function as an objective function of M estimation is used, where $\widehat{\varepsilon}_i$ is the residual divided by residuals standard deviation, and constant c is called a *tuning constant*. The small value of c produce increases resistance to outliers but at the expense of very low efficiency when the residuals are normally distributed. Therefore, the value of $c = 4.685$ is usually selected to provide 95% efficiency when the residuals are normally distributed [39]. The $\mathcal{U}(\widehat{\varepsilon}_i)$ bisquare objective function is given as follows:

$$\mathcal{U}(\widehat{\varepsilon}_i) = \begin{cases} \dfrac{\widehat{\varepsilon}_i^2}{2} - \dfrac{\widehat{\varepsilon}_i^4}{2c^2} + \dfrac{\widehat{\varepsilon}_i^6}{6c^4}, & \left|\widehat{\varepsilon}_i\right| \le c \\ \dfrac{\widehat{\varepsilon}_i^2}{6} & \left|\widehat{\varepsilon}_i\right| > c. \end{cases} \tag{4}$$

To define the *weight function* of the residuals, we should obtain the partial differentiation of this equation with respect to θ_2. Let ψ be the first derivative function of $\mathcal{F}(\widehat{\varepsilon}_i)$, which define the weight function

$$\begin{aligned} \sum_{i=1}^{m} X_i \psi\left(\frac{(Y_i - (\theta_1 + \theta_2 X_i))}{\sigma}\right) &= 0, \\ w(\widehat{\varepsilon}_i) &= \frac{\psi(\widehat{\varepsilon}_i)}{\widehat{\varepsilon}_i}. \end{aligned} \tag{5}$$

The weight function w of this model also changed according to observations.

$$w_i = \begin{cases} \left(1 - \left(\dfrac{\widehat{\varepsilon}_i}{c}\right)^2\right)^2, & \left|\widehat{\varepsilon}_i\right| \le c \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

To determine the optimal solutions or values of θ_1 and θ_2 by Tukey's bisquare weighted function,

$$\sum_{i=1}^{m} X_i w_i \left(\frac{(Y_i - (\theta_1 + \theta_2 X_i))}{\sigma}\right) = 0. \tag{7}$$

We utilize this approach to fit a trend polynomial model to all observations of the CPU utilization of VMs. In every iteration, weight function is defined according to new residuals that is called iteratively reweighed least squares and is repeated until it converges to the optimal values of θ_1 and θ_2, which determine the minimum value of $\mathcal{U}(\widehat{\varepsilon}_i)$ metric. This minimum value is called *MeReq*, which estimates the upper threshold of CPU utilization.

The detection of the overloaded host is determined by the upper CPU utilization threshold metric used in [18]. We extended this metric through *MeReg* to detect overloaded hosts shown as follows:

$$UpT = 1 - p \times MeReq, \tag{8}$$

where p is the safety parameter of this algorithm, which define how fast the system is in consolidating VMs. Moreover, the small value of safety parameter p implies low energy consumption but high SLA violation and vice versa [18]. The pseudocode of *MeReg* host overloaded detection algorithm, which helps in understanding the full workflow of the algorithm, is discussed in Algorithm 1.

5. Efficiency Metrics

Various matrices are used to evaluate the results and compare the effectiveness of the algorithm. The first metric is called total energy consumed by the data center resources at different workloads. The second type of efficiency metric is the average percentage of the SLA violation, which only occurs when provision VMs are not obtaining the requested resources (or when the average computing power of the shared host is not allocated to the requested VMs). This metric directly influence the QoS, which is not negotiated between cloud provider and its users. If an SLA violation occurs, then the CSP should pay some penalty to users.

5.1. Performance Metric (Pertric). To maximize the overall performance with minimum energy consumption, average SLA violation, and number of the reactivation hosts, we introduced a performance metric. If the host reactivated from energy saving-mode called reactivated host. These hosts directly affect the energy consumption of the data center. To address this concern, a performance metric is described as follows:

$$Pertric = ASLA \times HS \times E, \tag{9}$$

(1) **Input:** Dataset of the CPU utilization
(2) **Output:** Boolean // Host is overloaded or Not
(3) Initiate the $Y[]$ and $X[]$ // $Y[]$ is the CPU utilization dataset.
(4) **for each** $j \in [1, 100]$ **do**
(5) **for each** $i \in [Y.\text{length}]$ **do**
(6) $\varepsilon_i \leftarrow Y_i - (\theta_1 + \theta_2 X_i)$
(7) **end for**
(8) Calculated the σ
(9) $\sigma \longleftarrow \dfrac{\text{median}|\varepsilon_i - \text{median}(\varepsilon_i)|}{0.6745}$
(10) Initialised $\widehat{\varepsilon}_i[]$ array
(11) **for each** $i \in [Y.\text{length}]$ **do**
(12) $\widehat{\varepsilon}_i \longleftarrow \dfrac{(Y_i - (\theta_1 + \theta_2 X_i))}{\sigma}$
(13) **end for**
(14) Calculate Tukey's bisquare function
(15) **if** $\widehat{\varepsilon}_i \le c$ **then**
(16) $\mathcal{U}(\widehat{\varepsilon}_i) \longleftarrow \dfrac{\widehat{\varepsilon}_i^2}{2} - \dfrac{\widehat{\varepsilon}_i^4}{2c^2} + \dfrac{\widehat{\varepsilon}_i^6}{6c^4}$
(17) **else if** $\widehat{\varepsilon}_i > c$ **then**
(18) $\mathcal{U}(\widehat{\varepsilon}_i) \longleftarrow \dfrac{\widehat{\varepsilon}_i^2}{6}$
(19) Calculate the weighted value
(20) **if** $\widehat{\varepsilon}_i \le c$ **then**
(21) $w_i \longleftarrow \left(1 - \left(\dfrac{\widehat{\varepsilon}_i}{c}\right)^2\right)^2$
(22) **else if** $\widehat{\varepsilon}_i > c$ **then**
(23) $w_i \longleftarrow 0$
(24) Finding the value of θ_1 and θ_2 by using as follows
(25) $\sum_{i=1}^{m} X_i w_i \left(\dfrac{(Y_i - (\theta_1 + \theta_2 X_i))}{\sigma}\right) \longleftarrow 0$
(26) **end for**
(27) MeReg $\leftarrow$ minimum value of $\mathcal{U}(\widehat{\varepsilon}_i)$
(28) $upT \leftarrow p \times$ MeReg
(29) **return** HostUtilisation $> upT$

Algorithm 1: *MeReg* host overloaded detection.

where *Pertric* represents the overall performance metric, *HS* represents the total number of the host shutdowns after applying these algorithms, and E is the total electric energy consumption of the data center. The average SLA violation percentage in the data center is represents as *ASLA*.

6. Experiment Setup

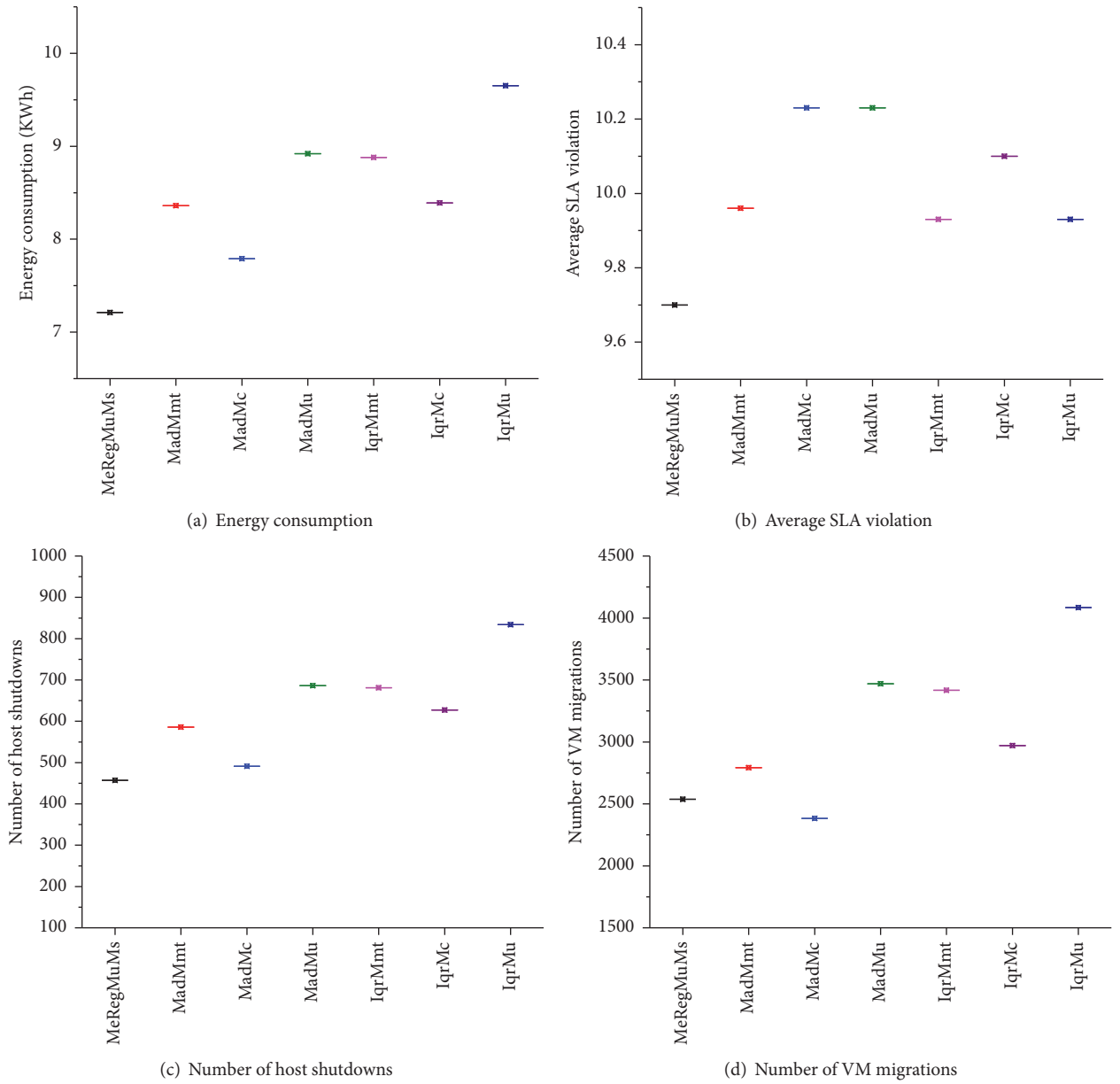
The deployment of real large-scale virtualized infrastructure is very expensive and conducting a repeatable experiment to analyse and compare the result of the proposed algorithm is difficult. Therefore, simulation is a best choice for evaluating VM selection policy to repeat the experiment of the proposed algorithms. We chose the CloudSim toolkit [40] for analysis and compared the performance of the proposed host overloaded detection algorithm. This is a modern open source simulator, which provides an IaaS cloud computing framework that enables us to conduct repeatable experiments for which results can be analysed and compared on large-scale virtualized cloud data centers.

Table 3: Types of Amazon EC2 VM.

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

In our cloud computing simulation setup, we installed 800 heterogeneous servers with real configurations. These hosts are Fujitsu M1, Fujitsu M3, Hitachi TS10, and Hitachi SS10. The features of these servers are presented in Table 2. The electric energy consumption of these servers at different workloads is shown in Table 1.

The CPU clock speed of servers is mapped onto MIPS ratings; that is, each core of the servers Fujitsu M1, Fujitsu M3, Hitachi TS10, and Hitachi SS10 is mapped 2700, 3500, 3500, and 3600 MIPS, respectively. The network bandwidth of each server is modeled to possess 1 GB/s. The corresponding VM types are supported by Amazon EC2 VM, as shown in Table 3.



(a) Energy consumption

(b) Average SLA violation

(c) Number of host shutdowns

(d) Number of VM migrations

Figure 2: Evaluation of the proposed host overload detection algorithm using random workload.

Simulation must be conducted using real workload traces of the data center server, which is applicable on real cloud environment. To achieve this objective, we used the data provided by PlanetLab as part of the CoMon project [41]. We utilized more than a thousand heterogeneous VM CPU utilization data from more than 500 heterogeneous servers placed worldwide. The features of the data daily are discussed in Beloglazov and Buyya [18].

7. Simulation and Analysis

Real time CPU utilization data of heterogeneous servers is used to evaluate the performance of *MeReg* host overloaded detection algorithm. We simulated the proposed algorithm with the *MuMs* VM selection scheme and compared it with the overloaded hosts detection algorithms and VM selection policy described in Beloglazov and Buyya [18]. These overloaded host algorithms are median absolute deviation (MAD), and interquartile range (IQR) with maximum correlation (MC), minimum migration time (MMT), and minimum utilization (MU) of VM selection policy. We used the values of safety parameters (p) 1, 2.5, and 1.5 for *MeRegMuMs*, *MAD*, and *IQR*, respectively.

7.1. Random Workload. In the random workload, every VM runs an application with a variable utilization of CPU, which is generated with a uniform distribution. In Figure 2(a), the electric energy consumption by using *MeRegMuMs* host overloaded detection algorithm must be lesser than the other approaches. Figure 2(b) shows significant reduction

in average SLA violation. Moreover, in Figures 2(c) and 2(d) the number of shutdown hosts and VM migrations are also reduced more efficiently than the other host overloaded detection algorithms.

7.2. Real Workload. The real workload dataset is provided by the PlanetLab as part of the CoMon project. In the CoMon project, data of thousands of VMs CPU utilization worldwide are collected every five minutes and stored in different extension files. We selected this real dataset to evaluate the proposed policy. Analysis of the proposed policy using real workload is discussed in the following subsections.

7.2.1. Evaluation of Energy Consumption. The total electric energy consumption of the resources of the hosts in the data center depends on CPU utilization, primary memory, network devices, and disks. However, numerous studies have revealed that the host CPU consumes more electric energy than other resources in the hosts [29]. Therefore, we are more focused on the CPU utilization of hosts. In this section we analysed the simulation of *MeRegMuMs* host overloaded detection with the MAD and IQR. As shown in Figure 3, electric energy consumption by the proposed algorithm is 17.3% lesser than means of other algorithms.

7.2.2. Evaluation of the Average SLA Violation. Maintaining the QoS is an important aspect of cloud computing environment. The required QoS are determined by SLAs [9]. In this section, we analysed and compared the percentage of average SLA violation in overloaded hosts. Cloud users do not want SLA violation and performance degradation. If these situations occur then CSP should pay the penalty to users. Thus, reduced SLA violation is desired among users and CSPs. Figure 4 shows that the percentage of average SLA using the *MeRegMuMs* host overloaded detection is 23.3% lesser than that of traditional algorithms.

7.2.3. Number of Host Shutdowns and VM Migrations. The cost of dynamic live migration of VMs is always high, which includes processing power on the allocated host, and performance degradation [9, 14]. Therefore, minimizing the total number of VMs migrations is one of the objectives of this study. In this section, we analysed and compared the simulation of the number of host shutdowns and VM migrations. If the number of reactivated hosts increase, then energy consumption is maximized. The host is reactivated to allocate new VMs and shutdown when it detect underload.

In the experiment environment, we installed 800 hosts but the number of host shutdowns is greater than 800 due to host reactivation. Figure 5 shows that the proposed algorithm also, minimized 25.9% of host reactivations of hosts relative to traditional MadMmt, MadMc, MadMu, IqrMmt, IqrMc, and IqrMu algorithms.

Meanwhile, the number of migration is directly proportional to performance degradation. If the total number of VM migrations decreases then performance degradation also decreases, which is desired by users and CSPs. The comparison of the proposed policy VM migration with other

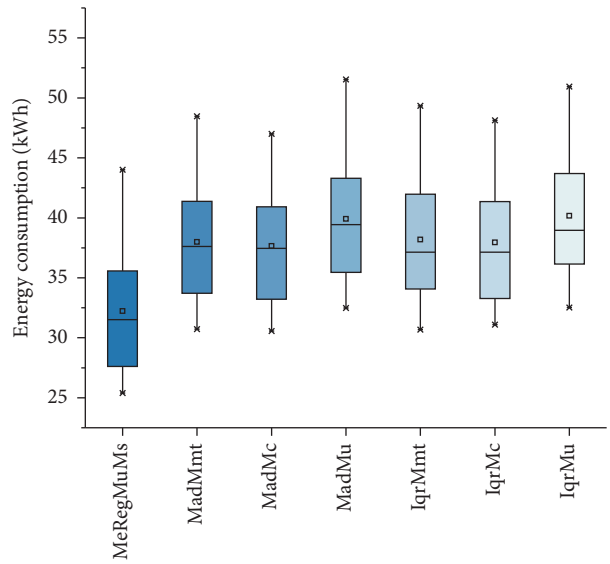


Figure 3: Energy consumption comparison using real workload.

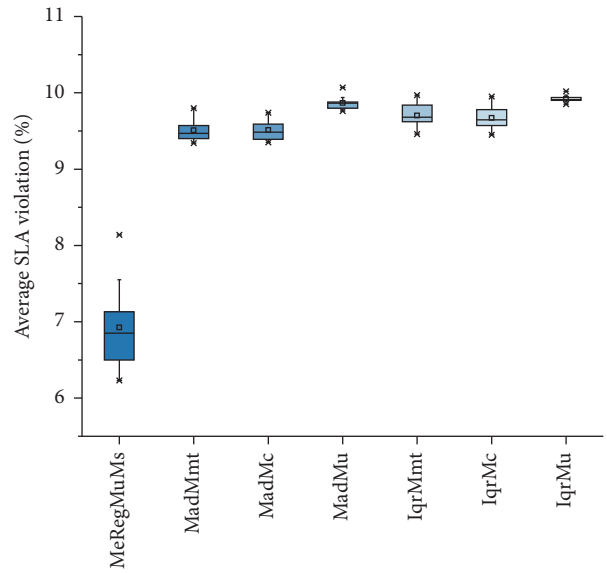


Figure 4: Comparison of the average percentage of SLA violation.

old algorithms proposed in Beloglazov and Buyya [18] is described in Figure 6.

7.2.4. Evaluation of Pertric. In this section, we discussed the overall performance of the cloud data center using proposed *MeReg* host overloaded detection algorithm. The overall performance calculated by the Pertric metric proposed in Section 5.1 is also discussed. The main objective is to propose this metric to analyse the all aspects of energy-awareness in the cloud data center, such as minimization of electric energy consumption, average percentage SLA violation, and number of reactivated hosts for placing new VMs.

Figure 7 shows the effectiveness of the *MeReg* host overload detection algorithms using MuMs VMs selection policy relative to other old host overload detection algorithms

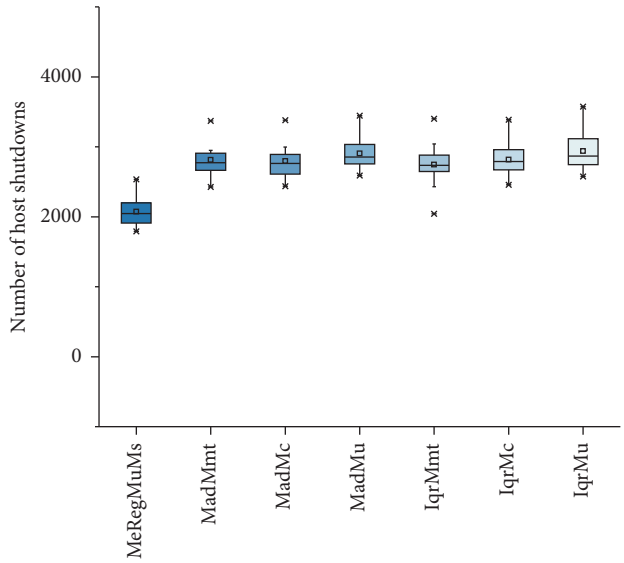


Figure 5: Total number of host shutdowns.

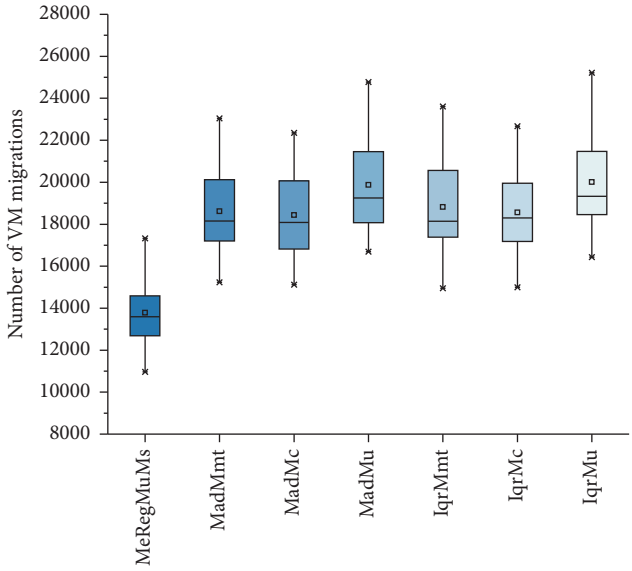


Figure 6: Total number of VM migrations.

using VM selection policies such as MadMmt, MadMc, MadMu, IqrMmt, IqrMc, and IqrMu.

7.2.5. Statistical Analysis. Statistical analysis validated the proposed algorithm, and the results demonstrated the efficiency of the proposed algorithm compared with other approaches. One-way ANOVA on the Pertric Matrices is conducted to analyse the tradeoff between minimizing the overall energy consumption and maximizing the QoS of the data center demonstrated in Table 4. Based on the One-way ANOVA result, *MeRegMuMs* significantly reduced energy consumption and maximized QoS, compared with *MadMc, MadMmt, MadMu, IqrMc, IqrMmt,* and *IqrMu*. Table 4 shows that the F ratio (10.61) is greater than the F critical value (2.24), which indicates that the null hypothesis is rejected

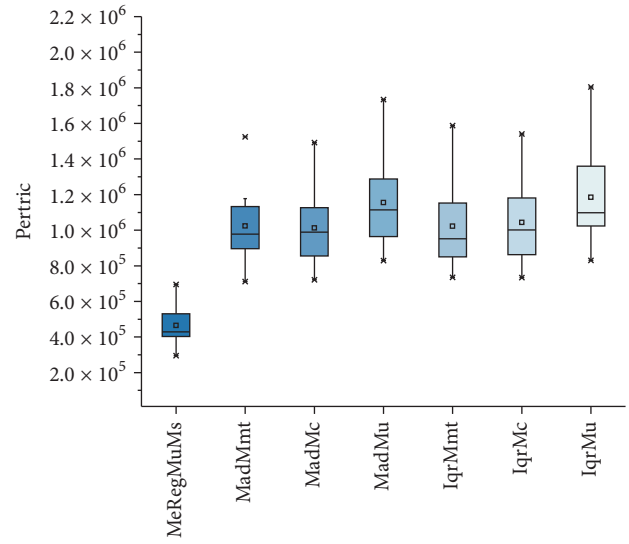


Figure 7: Performance metric (Pertric) comparison.

and the population means are significantly different from one another at the 0.05 level. Therefore, the *MeRegMuMs* algorithm is significantly different from other algorithms, such as *MadMc, MadMmt, MadMu, IqrMc, IqrMmt,* and *IqrMu* with p value of $4.068E - 8$.

One sample t-test of VM migration time duration and host running time is also carried out. The average value of the sample mean times before a VM migration during the host detection underload or overload is 19.67 seconds with a 95% CI: 18.23, 20.12. The average value of the sample means host running time before transition to energy-saving-mode is 21.3 minutes with 95% CI: 20.2, 22.8.

8. Conclusion and Future Work

Mobile cloud computing enables seamless and rich functionality of the cloud computing services to mobile users. Mobile cloud data centers worldwide are growing according to the increasing demand of data processing and storage by mobile users. Therefore, to keep the mobile cloud data centers running, massive amount of electric energy is required, which leads to high operational costs and CO_2 emission. High emission of CO_2 negatively impacts the social environment. In this study, we introduced a novel adaptive heuristic host overload detection algorithm called *MeReg*, which minimizes electric energy consumption and maximize QoS in terms of required SLA of the data center. A host overload problem directly influences VM performance, which is totally against SLA. Therefore, a regression-based technique called M estimation is used to find optimal upper CPU utilization threshold for detecting overloaded hosts. For VM consolidation from overloaded hosts, the approach used in previous study called *MuMs* policy is implemented, which selects VM from overloaded or underloaded hosts and migrates it to appropriate hosts. CloudSim simulator is used in the implementation of the proposed algorithm to obtain the results using 10 different real workload traces.

Table 4: Summary of the one-way ANOVA test.

Source of variation	df	SS ($\times 10^{10}$)	MS ($\times 10^{10}$)	F ratio	p value	F critical
Between groups	6	347	57.8	10.61	$4.07E-8$	2.246
Within groups	63	343	5.45			
Total	69	690				

In the future, we plan to further extend this work by introducing a machine learning based technique called Markov chain for VM consolidation policy, which works better in a dynamic environment such as cloud computing. The implementation of these algorithms in the open source real cloud platform such as OpenStack would also be studied.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The National Key Research and Development Plan under Grant no. 2016YFB0800801 and the National Science Foundation of China (NSFC) under Grants nos. 61672186 and 61472108 support this work.

References

[1] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: Architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, vol. 13, no. 18, pp. 1587–1611, 2013.

[2] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: a survey," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 84–106, 2013.

[3] S. Vakilinia, B. Heidarpour, and M. Cheriet, "Energy efficient resource allocation in cloud computing environments," *IEEE Access*, vol. 4, pp. 8544–8557, 2016.

[4] I. Petri, H. Li, Y. Rezgui, Y. Chunfeng, B. Yuce, and B. Jayan, "A HPC based cloud model for real-time energy optimisation," *Enterprise Information Systems*, vol. 10, no. 1, pp. 108–128, 2016.

[5] B. P. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," in *Proceedings of the 5th International Joint Conference on INC, IMS and IDC (NCM '09)*, pp. 44–51, Seoul, Republic of Korea, August 2009.

[6] G. Motta, N. Sfondrini, and D. Sacco, "Cloud computing: An architectural and technological overview," in *Proceedings of the International Joint Conference on Service Sciences, Service Innovation in Emerging Economy: Cross-Disciplinary and Cross-Cultural Perspective (IJCSS '12)*, pp. 23–27, May 2012.

[7] S. Lambert, W. Van Heddeghem, W. Vereecken, B. Lannoo, D. Colle, and M. Pickavet, "Worldwide electricity consumption of communication networks," *Optics Express*, vol. 20, no. 26, pp. B513–B524, 2012.

[8] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *The Computer Journal*, vol. 40, no. 12, pp. 33–37, 2007.

[9] F. Farahnakian, A. Ashraf, T. Pahikkala et al., "Using Ant Colony System to Consolidate VMs for Green Cloud Computing," *IEEE Transactions on Services Computing*, vol. 8, no. 2, pp. 187–198, 2015.

[10] A. Corradi, M. Fanelli, and L. Foschini, "VM consolidation: A real case based on OpenStack Cloud," *Future Generation Computer Systems*, vol. 32, no. 1, pp. 118–127, 2014.

[11] W.-Z. Zhang, H.-C. Xie, and C.-H. Hsu, "Automatic memory control of multiple virtual machines on a consolidated server," *IEEE Transactions on Cloud Computing*, vol. 5, no. 1, pp. 2–14, 2017.

[12] S. E. Dashti and A. M. Rahmani, "Dynamic VMs placement for energy efficiency by PSO in cloud computing," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 28, no. 1-2, pp. 97–112, 2016.

[13] Y.-J. Hong, J. Xue, and M. Thottethodi, "Dynamic server provisioning to minimize cost in an IaaS cloud," in *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS'11*, pp. 147-148, ACM, June 2011.

[14] A. Beloglazov, R. Buyya, Y. C. Lee, and A. Zomaya, "A taxonomy and survey of energy-efficient data centers and cloud computing systems," *Advances in Computers*, vol. 82, pp. 47–111, 2011.

[15] Z. Cao and S. Dong, "Dynamic VM consolidation for energy-aware and SLA violation reduction in cloud computing," in *Proceedings of the 13th International Conference on Parallel and Distributed Computing, Applications, and Technologies, PDCAT '12*, pp. 363–369, IEEE, December 2012.

[16] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755–768, 2012.

[17] R. N. Calheiros, R. Ranjany, and R. Buyya, "Virtual machine provisioning based on analytical performance and QoS in cloud computing environments," in *Proceedings of the 40th International Conference on Parallel Processing, ICPP '11*, pp. 295–304, September 2011.

[18] A. Beloglazov and R. Buyya, "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*, vol. 24, no. 13, pp. 1397–1420, 2012.

[19] R. Yadav, W. Zhang, H. Chen, and T. Guo, "MuMs: Energy-Aware VM Selection Scheme for Cloud Data Center," in *Proceedings of the 28th International Workshop on Database and Expert Systems Applications (DEXA)*, pp. 132–136, Lyon, France, August 2017.

[20] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "Thinkair: dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in *Proceedings of the IEEE INFOCOM*, pp. 945–953, IEEE, March 2012.

[21] S. U. R. Malik, K. Bilal, S. U. Khan, B. Veeravalli, K. Li, and A. Y. Zomaya, "Modeling and analysis of the thermal properties

exhibited by cyberphysical data centers," *IEEE Systems Journal*, vol. 11, no. 1, pp. 163–172, 2017.

[22] W. Zhang, S. Han, H. He, and H. Chen, "Network-aware virtual machine migration in an overcommitted cloud," *Future Generation Computer Systems*, vol. 76, pp. 428–442, 2017.

[23] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4569–4581, 2013.

[24] S. Esfandiarpoor, A. Pahlavan, and M. Goudarzi, "Structure-aware online virtual machine consolidation for datacenter energy improvement in cloud computing," *Computers and Electrical Engineering*, vol. 42, pp. 74–89, 2015.

[25] X. Zhu, D. Young, B. J. Watson et al., "1000 Islands: integrated capacity and workload management for the next generation data center," in *Proceedings of the 5th International Conference on Autonomic Computing, ICAC '08*, pp. 172–181, June 2008.

[26] R. Nathuji and K. Schwan, "VirtualPower: Coordinated power management in virtualized enterprise systems," *ACM SIGOPS Operating Systems Review*, vol. 41, no. 6, pp. 265–278, 2007.

[27] P. Ranganathan, P. Leech, D. Irwin, and C. Jeffrey, "Ensemble-level power management for dense blade servers," *ACM SIGARCH Computer Architecture News*, vol. 34, no. 2, pp. 66–77, 2006.

[28] V. Venkatachalam, M. Franz, and C. W. Probst, "A new way of estimating compute-boundedness and its application to dynamic voltage scaling," *International Journal of Embedded Systems*, vol. 3, no. 1-2, pp. 17–30, 2007.

[29] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," *ACM SIGARCH Computer Architecture News*, vol. 35, no. 2, pp. 13–23, 2007.

[30] C. Lange, D. Kosiankowski, R. Weidmann, and A. Gladisch, "Energy consumption of telecommunication networks and related improvement options," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 17, no. 2, pp. 285–295, 2011.

[31] Y. Shang, D. Li, and M. Xu, "Energy-aware routing in data center network," in *Proceedings of the 1st ACM SIGCOMM workshop on Green networking*, pp. 1–8, ACM, August 2010.

[32] I. Takouna, R. Rojas-Cessa, K. Sachs, and C. Meinel, "Communication-aware and energy-efficient scheduling for parallel applications in virtualized data centers," in *Proceedings of the IEEE/ACM 6th International Conference on Utility and Cloud Computing, UCC '13*, pp. 251–255, IEEE Computer Society, December 2013.

[33] R. Liu, H. Gu, X. Yu, and X. Nian, "Distributed flow scheduling in energy-aware data center networks," *IEEE Communications Letters*, vol. 17, no. 4, pp. 801–804, 2013.

[34] S. C. Shah, "Recent Advances in Mobile Grid and Cloud Computing," *Intelligent Automation and Soft Computing*, pp. 1–13, 2017.

[35] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang, "Power and performance management of virtualized computing environments via lookahead control," *Cluster Computing*, vol. 12, no. 1, pp. 1–15, 2009.

[36] C.-H. Hsu and S. W. Poole, "Power signature analysis of the SPECpower_ssj2008 benchmark," in *Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS '11)*, pp. 227–236, Austin, Tex, USA, April 2011.

[37] Y. Susanti, H. Pratiwi, H. Sulistijowati, and T. Liana, "M Estimation, S estimation, and MM estimation in robust regression," *International Journal of Pure and Applied Mathematics*, no. 3, pp. 349–360, 2014.

[38] H. Edelsbrunner and D. L. Souvaine, "Computing least median of squares regression lines and guided topological sweep," *Journal of the American Statistical Association*, vol. 85, no. 409, pp. 115–119, 1990.

[39] J. Fox, *Applied Regression Analysis and Generalized Linear Models*, Sage, 2015.

[40] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. de Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.

[41] K. Park and V. S. Pai, "CoMon: a mostly-scalable monitoring system for PlanetLab," *ACM SIGOPS Operating Systems Review*, vol. 40, no. 1, pp. 65–74, 2006.