IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 6, 2019, accepted November 18, 2019, date of publication November 29, 2019,
date of current version December 17, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2956881

Linear Power Modeling for Cloud Data Centers: Taxonomy, Locally Corrected Linear Regression, Simulation Framework and Evaluation

LEILA ISMAIL [1], (Member, IEEE), AND EYAD H. ABED [2], (Fellow, IEEE)
[1]Department of Computer Science and Software Engineering, College of Information Technology, United Arab Emirates University, Al-Ain 15551, United Arab Emirates
[2]Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742, USA

Corresponding author: Leila Ismail (leila@uaeu.ac.ae)

This work was supported by the Emirates Center for Energy and Environment Research, United Arab Emirates University, under Grant 31R101.

ABSTRACT Cloud computing is a novel paradigm, where the limitations of ubiquitous connected devices in terms of computing, data access, networking and storage are solved through the use of cloud infrastructure. The pervasive adoption of cloud computing results in a rising carbon footprint due to the high energy consumption of computing servers. This negatively affects the environment and entails an associated increase in electricity costs and consequently operational costs. Many works proposed scheduling algorithms using software-centric power models in order to predict electric power consumption in underlying data centers and to schedule cloud tasks so as to reduce energy consumption. Linear models which are based on the lowest- and highest-power data points (referred to here as the ''Power Endpoints Model'' - PEM) and the simple linear regression (SLR) model are the most used in the literature. However, these models have traditionally been evaluated using different environments, experimental setups, workloads, and error calculation formulas. In this paper, a unified classification and evaluation for these linear power models is presented, under unified setup, benchmarking applications, and error formula with the main goal being to achieve an objective comparison. A new power model is proposed, named Locally Corrected Multiple Linear Regression (LC-MLR), in order to increase prediction accuracy. A simulation framework for a cloud energy-aware scheduler is introduced. The framework combines the Energy-Aware Task Scheduling on Cloud Virtual Machines (EATSVM) with the LC-MLR power model, and facilitates performance measurement for cloud data centers. The scheduler with the new power model increases energy efficiency without degrading the qualities of service of the system. The workloads used for performance evaluation and comparisons in this work are generated using a diverse set of applications. The results show that LC-MLR outperforms the most-used models for simulation of power consumption of cloud data centers. The detailed performance analysis is elaborated in the paper.

INDEX TERMS Cloud computing, data centers, energy-aware scheduling, energy efficiency, energy simulation, green computing, linear least squares, power models.

I. INTRODUCTION

Cloud computing is an emerging technology enabling on-demand access to a shared pool of configurable computing resources, such as networks, servers, storage, applications and services. It can be rapidly provisioned and released with minimal management effort or service provider interaction [1]. It is envisioned that clouds will provide infrastructure and middleware for future smart cities applications [2]. Consequently, the cloud will be responsible for handling real-time processing and analytics for an enormous amount of streaming data from heterogeneous ubiquitous Internet of Things (IoT) devices, and for managing smart energy hubs [3]. The constantly expanding use of cloud computing, with the associated growth in the size of datacenters, makes energy consumption and related costs a critical issue. According to the Natural Resources Defense Council (NRDC) in the USA [4], data centers used about 91 billion

The associate editor coordinating the review of this manuscript and approving it for publication was Abdullah Iliyasu.

kilowatt–hours of electrical energy in 2013, equivalent to the output of 34 large coal fired plants. This is estimated to reach 140 billion kWh by 2020, a 53% increase. The associated expected annual electricity cost is $13 billion. This will equate to about 100 million metric tons of carbon pollution released into the atmosphere annually.

Energy consumption in data centers can be reduced through deploying energy-efficient algorithms, modifying hardware components architecture [5], using efficient power supply options, designing measures for efficient air handling [6] and cooling measures [6], [7]. In the various energy-efficient algorithms, a power model is used to predict the power consumption of an application and/or to estimate the energy consumed by a server after the execution of an algorithm to assess the effectiveness of the algorithm in achieving energy efficiency. Researchers proposed different power models for servers, based either on a server's hardware such as die temperature, fan speed, heat dissipation, voltage, current, capacitance, motherboard components, and resistance [8]–[14] or its user-level utilization metrics (CPU, memory, disk, and network) [15], [16], [25]–[34], [17], [35]–[44], [18], [45]–[49], [19]–[24]. The hardware-based power models require sensors to measure different variables on a server's hardware. This adds in extra hardware and energy consumption costs incurred by these sensors attached to thousands of servers in a data center. However, the software-based models do not require external sensors to get the values of the model variables. These values are reported by the server's operating system as the performance metrics adding no extra cost. Therefore, in this paper, we focus on the software-based power models. These models are divided into 2 categories: 1) linear [15], [16], [25]–[34], [17], [35]–[38], [43]–[48], [18], [49], [19]–[24], [50] and 2) non-linear [17], [26], [39]–[42], [48]. A linear model has a complexity of O(n) [51] compared to $O(n^2)$ for a non-linear model [52]. In addition, a linear model has lower latency compared to a non-linear one [51], [52]. Consequently, we concentrate on the software-based linear power models.

Most research on linear power models and energy optimizations for servers propose models based on the lowest- and highest-power data points (referred to here as the ''Power Endpoints Model'' or simply PEM) [15], [16], [27], [38], [43]–[49], [53], [17]–[24], or models based on Simple Linear Regression (SLR) [54], [25], [26], [28]–[31], to predict the amount of power consumption. These models use CPU utilization as an independent variable for power prediction. This is based on the studies showing that server power consumption is highly dominated by its CPU utilization compared to memory, disk, and network utilizations [48]. However [32] validated experimentally that memory, disk and network utilizations also contribute to power consumption. Despite their adoption for energy optimization, PEM and SLR have not yet been compared using a unified environment and workload. The present paper attempts to address this void. In addition, we propose a Locally Corrected Multiple Linear Regression (LC-MLR) power model that takes into account CPU, memory, disk and network. LC-MLR, is an extension to the classical MLR model aimed at improving the prediction accuracy. We exemplify our model in the context of a cloud computing data center environment, but it is generic and can be applied in any data center. We present a classification of linear power models and a comparison between PEM, SLR and LC-MLR models. We evaluate their performance in terms of standard error of estimation between the actual power consumption values and the predicted ones using those models. This is done in a unified environment and experimental setup. We use four different tools for the model building and validation and five different applications for testing.

Energy-aware cloud VM placement and task scheduling algorithms in cloud computing rely on power modeling for energy prediction. Moreover, the desired energy efficiency must be achieved while adhering to the Service Level Agreements (SLAs) of the applications. An SLA is an agreement between the service provider and the customer that identifies the required levels of the services offered [55]. Therefore, it is crucial to have a precise power model for more energy savings with the least violations of SLAs. To analyze the performance of energy-aware scheduling algorithms in a cloud data center, we introduce a simulation framework providing a precise power model and an energy-aware scheduler. We modify our energy-aware scheduling algorithm Energy-Aware Task Scheduling on Cloud Virtual Machines (EATSVM) [56] for integration within the framework.

The major contributions of this paper are as follows:

1. We present a taxonomy of linear power models in the data center energy consumption modeling literature. The works on power modeling are then classified into PEM, SLR, and MLR techniques.
2. We propose a new power model, LC-MLR, to predict the power consumption of single servers, which in multiple experiments is found to be more accurate than the mostly used linear power models. It is an extension to the classical MLR model by adding an error correction term.
3. We evaluate the performance of the models by using a diverse set of benchmarks and applications in a unified experimental setup. The experimental results show that the performance of LC-MLR is superior to that of PEM and SLR, and that the precision of an energy-efficient scheduler depends on the power model used. There is a non-negligible impact of memory, disk, and network utilizations on power consumption of CPU-intensive applications, which should be considered by power models for more accurate prediction
4. We present a cloud energy-aware scheduler workflow. We implement it by extending CloudSim [57], a simulation framework for a cloud data center, with a model builder and EATSVM scheduler to evaluate the power models under study. The results show that LC-MLR has the least standard error of estimation in both energy consumption and SLA violations.

5. We consider the impact of an increasing number of hosts in a data center on the performance of the power models.

The remainder of the paper proceeds as follows. Section II provides an overview of related literature. Section III synthesizes a taxonomy of the power models which are based on user-level server utilization metrics. Section IV presents our new LC-MLR model. Section V describes the energy-aware cloud system model used in this work, and an overall architecture and workflow for energy-aware cloud scheduling. The experimental setup, experiments and the performance evaluation, in terms of standard error of estimation for LC-MLR compared to the most used power models PEM and SLR are presented in Section VI. Section VII concludes this paper with lessons learned and possible future research directions.

II. RELATED WORKS

In this section, we provide an overview of related works on energy-efficient scheduling strategies in data centers and the power models used in data center design and/or operation.

A. CLOUD DATA CENTERS VM PLACEMENT AND TASK SCHEDULING ALGORITHMS FOR ENERGY EFFICIENCY

There have been several research efforts aimed at reducing energy consumption through virtual machine (VM) consolidation/placement [43]–[45], [58]–[63] and task scheduling algorithms [2], [20], [49], [64]–[69] in a cloud data center. The works on those algorithms can be classified into two categories: 1) energy-aware VM placement [44], [45], [58]–[60], [62], [63] and task scheduling [2], [20], [49], [64], [66], [68], [69] having power consumption model as an integral part of the algorithms equations, and 2) non-energy-aware VM placement [43, 61] and task scheduling [65], [67], where the model is not part of the algorithms equations, but is used to calculate the power utilized by the underlying servers after the algorithm is executed.

Concerning energy-aware algorithms, Beloglazov et al. [45] proposed Modified Best Fit Decreasing (MBFD) VM placement for energy optimization. MBFD places a VM on a host where the increase in power consumption is the least. Consequently, the algorithm consolidates the VMs, making some servers idle. However, studies show that an idle server consumes 50%-70% of the server's maximum power [48], suggesting that the number of idle servers should be decreased for better energy efficiency. A similar approach to MBFD was used by Sinha et al. [44] and Beloglazov and Buyya [58] for the VM placement. In [60], Chowdhury et al. address the issue of server consolidation based on Modified Worst Fit Decreasing (MWFD), an algorithm which places a VM on the host in which the increase in power consumption is the maximum. MWFD showed more energy savings than MBFD. Li et al. [62] proposed an energy-thermal-aware VM placement algorithm to optimize both energy consumption and SLA Violations (SLAVs). The algorithm places a VM on a host satisfying the following 3 conditions: 1) the increase in power consumption is the minimum, 2) the CPU utilization is below a threshold, and 3) the cooling power consumption required is the minimum. It requires sensors to measure the temperature and a model for predicting the increase in temperature. The proposed algorithm tends to disperse the VMs over the data center to avoid having a thermal hotspot and to optimize SLAVs. Ilager et al. [63] use an algorithm to avoid this dispersion by consolidating the VMs in a condition that there is no thermal hotspot. Their results showed more energy savings than [62] but with more SLAVs. Compared to MBFD [45] which consolidates the VM, [63] has no thermal hotspot, saving more energy, and has fewer SLAVs.

Lee and Zomaya's Energy Conscious Task Consolidation (ECTC) schedules an incoming task on a machine in which the energy consumption to execute that task is the minimum. It assigns the task to the server which has the maximum overlapping time between the incoming task and the ongoing one(s) [20] aiming to maximize CPU utilization. The algorithm considers only a homogeneous data center where the execution time of the task is the same on all hosts. In our previous works [2], [56], we extended the ECTC algorithm to consider a heterogeneous data center, and the increase in the execution time(s) of the ongoing task(s) due to overlaps with the incoming task. Our results showed up to 35% of energy savings compared to ECTC and a better application performance in terms of execution time. Huai et al. [66] compared the Power Best Fit (PBF) and the Load Balancing (LB) algorithms for task scheduling. PBF schedules a task on the server with the least increment in power consumption, while LB, a non-energy-aware algorithm, distributes a task in a way to maintain equal CPU utilizations of the servers. PBF consumes more energy as it results in a bigger number of idle servers than LB, because the latter avoids task consolidation. Ying and Yu [64] propose a genetic algorithm to formulate a bi-objective optimization problem for task scheduling. The algorithm considers a fitness function for each server and places the task on the server with the maximum fitness value. The fitness function takes into account the power consumption and the execution time of the task on each server.

Concerning non-energy-aware algorithms, Bagheri and Zamanifar [43] address the issue of power consumption of idle servers by consolidating the VMs on the active servers and turning off the idle servers. However, the VM placement algorithm does not consider the server's turn-on and turn-off time durations and their corresponding energy consumptions and does not account for SLAVs. Farahnakian et al. [61] use an ant colony approach to optimize both energy consumption and SLAVs by reducing the number of VM migrations and increasing the number of servers in the sleep state. A server in sleep mode is switched on only when a VM cannot be placed on the active servers due to the utilization requirements of the VM. As in [66], Mehdi et al. [65] propose a load balancing task scheduling algorithm. However, they consider the active servers only as long as the task can be executed within the required deadline. If the deadline cannot be satisfied,

TABLE 1. Evaluation of past works on PEM and SLR power models.

Power Model	Work	Experimental Setup	Error Calculation Formula	Error Value
$P_{MIN} + (P_{MAX} - P_{MIN}) * u$	[48] (2007)	100 heterogeneous servers chosen from a google data center. The workload specifications are not mentioned.	Not Reported	Not Reported
$\alpha + \beta * u$	[26] (2013)	Two different servers with workloads from SPEC power benchmark, Linux utility program and gcc compiler.	$\frac{\sum_{i=1}^{n} \frac{\lvert actual_i - Predicted_i \rvert * 100\%}{actual_i}}{n}$	4.09%

then the task is scheduled on an idle server. Wu et al. [67] consider both the energy and the SLAVs optimizations by using the Dynamic Voltage and Frequency Scaling (DVFS) technique. The algorithm places a task on a server where the frequency required by the task lies between the minimum and the maximum servers' frequencies. The minimum ensures the performance of the task, while the maximum prevents server over-utilization. Singh et al. [68], [69] optimize both the energy consumption and the SLAVs by using fuzzy logic. Qureshi [49] propose a scheduling algorithm for application workloads or tasks based on the Application Profiles (APs). An AP includes the application's arrival and completion times, the CPU and memory utilizations, and the power consumption. Experimental results showed that for energy efficiency there should be a tradeoff between the number of VMs to execute an application and the application's completion time. To improve the energy-efficiency the scheduler considers running multi-tenancy of VMs on a single physical server [70].

The works on energy-efficient VM placement and task scheduling algorithms have focused on resource optimization strategies that also reduce energy consumption. The algorithms use power models either within the algorithm and/or after its execution. However, the accuracy of these algorithms depends on the power model used for power prediction. The works on energy-efficient algorithms in the literature use different power models. Next, we give an overview of power models used to predict data centers energy consumption.

B. POWER MODELING FOR ENERGY EFFICIENCY IN DATA CENTERS

Several recent research efforts have aimed at reducing energy consumption in data centers at the circuit, processor, memory/storage, and server levels, as well as at the overall data center level [71]. Power consumption models are pivotal for designing and optimizing energy-efficient operations to curb energy consumption in data centers. Several authors have proposed various power models [71], either to be used in simulation as a tool in designing energy-efficient data centers [16], [25], [28], [32], [36], [42], [48], [72], [73], or for server-level optimization [15], [17], [36], [37], [43], [45], [48], [49], [59], [72]–[74], [20]–[25], [27], [32]. The works on server-level optimization can be divided into two categories: 1) hardware-based on the server's die temperature, fan speed, heat dissipation, voltage, current, capacitance, motherboard components, and resistance [8]–[14], and 2) software-based, reflecting the server's user-level utilization metrics such as CPU, memory, disk, and network [15], [16], [25]–[34], [17], [35]–[44], [18], [45]–[49], [19]–[24]. The software-based models can be divided into linear [15], [16], [25]–[34], [17], [35]–[38], [43]–[48], [18], [49], [19]–[24] and non-linear [17], [26], [39]–[42], [48] models. A non-linear model has multiple local minima that usually require numerical optimization algorithms for determining model parameters. For a linear model, the sum of the squared error is a convex function, allowing for a closed form equation; this may not be the case for a non-linear model. For these reasons, the linear models are less complex and consume less time compared to non-linear models [51], [52], which explains their popular use. Consequently, in this paper, we consider the linear power model approaches. To predict the value of power consumption of a server using a linear approach, most research efforts proposed either power models based on SLR [25], [26], [28]–[31] or PEM [15], [16], [27], [38], [43]–[49], [53], [17]–[24]. These models use CPU only to predict the amount of power consumption, following [48] which identifies the CPU as the dominant factor in determining software utilization.

However, there is no comparative evaluation of the performance of these two most widely used approaches. The evaluations, reported in the literature for those models, are conducted using different setups, whether workloads, environments and experimental testbeds (Table 1). Consequently, one can't assess the relative quality of the models based on these evaluations. For instance, [48] does not report the error value from the predicted model, while [26] reports a 4% error value. In this work, we conduct an experimental evaluation of these models in a unified setup using the same workload and analyze their standard error of estimation. We also introduce an error correction to the standard LR algorithm to increase power modeling accuracy; we call this modified version of LR by the name LC-MLR (Locally Corrected Multiple Linear Regression); this accounts for CPU, memory, disk and network predicting the level of power consumption. We compare

TABLE 2. Taxonomy of linear power consumption models.

Power Model	Technique	Purpose	Works	Limitations
$P_{min} + (P_{max} - P_{min}) * U_{cpu}$	PEM	Designing a data center to avoid thermal emergencies based on server CPU utilization.	[16]	The power model is based only on the minimum and the maximum server power consumptions and does not take into consideration the power consumptions of server's CPU utilizations.
		Energy-efficient resource management based on server CPU utilization.	[48], [17], [18], [19], [53],	
		Energy-efficient virtual machines placement and task scheduling in cloud computing based on server CPU utilization.	[20], [21], [22], [23], [24], [50], [27], [38], [43], [46], [45], [44], [68], [69]	
$\alpha + \beta * U_{cpu}$	SLR	Energy-efficient resource management based on server CPU utilization.	[28–31], [63]	The accuracy of the model depends on how close the data are to the fitted regression line.
		Power consumption modelling based on server CPU utilization.	[25], [26]	
$\alpha + \beta_1 * U_{cpu} + \beta_2 * U_{mem} + \beta_3 * U_{disk}$	MLR	Power consumption modelling based on server's CPU, memory and disk utilization	[34], [35]	The accuracy of the model depends on how close the data are to the fitted plane.
$\alpha + \beta_1 * U_{cpu} + \beta_2 * U_{mem} + \beta_3 * U_{disk} + \beta_4 * U_{net}$	MLR	Power consumption modelling based on server's CPU, memory, disk and network utilization	[32], [33], [36]	The accuracy of the model depends on how close the data are to the fitted hyperplane.
		Energy-efficient resource management in data center	[37]	
$\alpha + \beta_1 * N_i + \beta_2 * M_{access} + \beta_3 * C_{active}$	MLR	Energy-efficient resource management in data center	[59]	The accuracy of the model depends on the applications used to generate the training data set for building the model. This is because the independent variables used by the model, i.e., the number of retired operations, the memory accesses, and the active cores are application-dependent.

P_{min} and P_{max} → Server's minimum and maximum power consumptions
U_{cpu}, U_{mem}, U_{disk}, and U_{net} → CPU, memory, disk, and network utilizations
N_i, M_{access}, and C_{active} → Number of retired instructions, memory accesses, and active cores

the performance of our model (LC-MLR) with the most used PEM and SLR, and evaluate the performances by using the energy-aware scheduling algorithm EATSVM in terms of standard error of variation in energy consumption and SLAVs of a cloud system. This is by using a diverse set of benchmarks and applications.

III. TAXONOMY OF LINEAR POWER CONSUMPTION MODELS

Linear models to predict the power consumption of a computing server aim to reduce the energy consumption of data centers. The models are used either at the data center design stage or during operation such as in the case of a cloud computing data center where loads have to be automatically and dynamically distributed by a cloud scheduler so as to reduce the overall energy consumption. In this section, we present a taxonomy (Table 2) of the linear models used in the literature.

A. PEM TECHNIQUE

In the PEM technique, the power model assumes a linear relationship between power consumption and CPU utilization of a computing server. The linear function based on the calculation of a line segment, where the slope and the intercept are based on the data endpoints, namely the maximum power consumption P_{max} (at the peak 100% usage of CPU resources) and the minimum power consumption P_{min} (at the CPU idle state) [16], [48]. The independent variable in PEM is taken to be CPU utilization, considered the dominant metric in server power consumption [48]. Heath et al. [16] give a linear power model where P_{min} is the intercept, meaning that it is a power consumption value that is always counted irrespective of utilization, and the difference between P_{max} and P_{min} gives the slope defining the rate of increase in power consumption per degree of utilization. Specifically, the model in [16] is:

$$P = P_{min} + (P_{max} - P_{min})U_{cpu} \quad (1)$$

This model appears again in the work by Fan et al. [48]. The model has significantly influenced recent energy-efficient cloud computing data center research [15], [20], [45], [46], [49], [21]–[24], [27], [38], [43], [44]. Qureshi et al. [17] and Cheung et al. [19] use PEM to model the power consumption at the data center level, and [18] use the model for energy-aware resource management. References [44], [45], [47] use PEM taking P_{min} to be 70% of P_{max}.

B. SIMPLE LINEAR REGRESSION (SLR) TECHNIQUE

In the SLR technique, collected experimental data on power consumption, corresponding to a server's single resource utilization, are used to calculate the slope and the intercept of a linear equation. Raghavendra et al. [28] propose a multi-level power management system for data centers. The proposed management system includes a Virtual Machine Controller (VMC), which reads as input the resource utilizations of the individual virtual machines (VMs) and implements an optimizer that creates a VMs-to-servers mapping to minimize the aggregate power for the whole data center. The mapping of VMs-to-servers is based on the energy consumption of VM on that particular server, which is predicted using the following formula:

$$\mathrm{P} = \alpha + \beta^{*}\mathrm{U_{cpu}} \tag{2}$$

Pedram et al. [25] and Zhang et al. [26] used SLR to model and predict power consumption for a multi-core server system with virtualization. References [29]–[31] use the model for energy-aware resource management in a data center based on CPU utilization.

C. MULTIPLE LINEAR REGRESSION (MLR) TECHNIQUE

In the MLR technique, the experimental measures of the power consumptions of several computing server metrics are used to predict the power consumption of a running task on a computing server. Economou et al. [32] introduce Mantis, a method for modeling full system power consumption and real-time power prediction. Mantis incorporates user-level server utilization metrics such as CPU, disk, memory, and network utilization as follows:

$$\mathrm{P} = \alpha + \beta * \mathrm{U_{cpu}} + \gamma^{*}\mathrm{U_{mem}} + \zeta^{*}\mathrm{U_{disk}} + \delta^{*}\mathrm{U_{net}} \tag{3}$$

Here, α, β, γ, ζ, and δ are regression coefficients, and $\mathrm{U_{cpu}}$, $\mathrm{U_{mem}}$, $\mathrm{U_{disk}}$, and $\mathrm{U_{net}}$ are the current CPU utilization, memory access count, hard disk I/O rate, and network I/O rate respectively.

Davis et al. [36] evaluate the accuracy of formula (3) using a Map Reduce-style workload. Nagasaka et al. [73] apply this technique for estimating power consumption for GPU applications. They use the exposed GPU performance counters as independent variables for applications. Kim et al. [59] apply this technique for energy-efficient cloud data centers by using the number of retired operations, memory accesses, and active cores as utilization metrics.

As noted above, most work in the literature uses a power model based on either PEM or SLR. In particular, these two models were integrated in energy-efficient load scheduling algorithms in cloud computing data centers. To our knowledge, this is the first study to classify linear power models and consistently evaluate the performance of these two most commonly used modeling approaches. In this work, we experimentally evaluate the models on three different servers with different power consumption profiles, and compare their performance with the proposed LC-MLR model.

IV. LC-MLR POWER MODEL

The LC-MLR model is an extension to the classical MLR model [75], involving the addition of an error correction term. The LC-MLR model is piecewise linear with complexity $O(n)$. The error correction term for a power consumption value, predicted using classical MLR, is calculated locally on MLR training data set intervals bounding the user-level server utilization metrics values corresponding to the predicted power consumption value. Thus, LC-MLR leverages local correlation between the actual and the predicted data values, bringing more precision to the prediction as compared to PEM, LR, and MLR models. The introduced corrections differ in different local regions of the data plot, leading to a piecewise linear approximation, as opposed to the purely linear forms in MLR and PEM.

To predict the server's power consumption value for an application having CPU, memory, disk, and network utilization values of $\mathrm{CPU_a}$, $\mathrm{mem_a}$, $\mathrm{disk_a}$, and $\mathrm{net_a}$ respectively using the LC-MLR model, we use the following formula:

$$P_{CPU_a,mem_a,disk_a,net_a} = \acute{\mathrm{P}}_{CPU_a,mem_a,disk_a,net_a} + \text{Ø}_{CPU_a,mem_a,disk_a,net_a} \tag{4}$$

where $\acute{\mathrm{P}}_{CPU_a,mem_a,disk_a,net_a}$ is the predicted power consumption using the MLR regression model given in Equation (5) below, and $\text{Ø}_{\mathrm{CPU_a,mem_a,disk_a,net_a}}$ is an error correction term. To calculate the regression coefficients (α, β_1, β_2, β_3, and β_4) of the MLR model (Equation 5), a training data set is used consisting of actual power consumptions corresponding to the user-level server utilization metrics values (CPU, memory, disk, and network). In a real data center scenario, this data set is generated from the server's history of power consumption and utilization metrics obtained while running different applications. However, in our laboratory environment, the training data set is obtained experimentally. We run different tools to stress the server's user-level utilization metrics. While the server is being stressed, we measure the utilization values of CPU, memory, disk, and network by using the Linux perf utility [76] and the collectd tool [77]. We simultaneously measure the server's power consumption values by using a LabVIEW program [78] which we implemented to extract data from an oscilloscope [79] connected to the server.

$$\acute{P}_{CPU_a,mem_a,disk_a,net_a} = \alpha + \beta 1 * CPU_a + \beta 2 * mem_a + \beta 3 * disk_a + \beta 4 * net_a \tag{5}$$

To calculate the error correction term corresponding to utilization values $\mathrm{CPU_a}$, $\mathrm{mem_a}$, $\mathrm{disk_a}$, and $\mathrm{net_a}$, the LC-MLR model uses the training data set to determine the intervals where $\mathrm{CPU_a}$, $\mathrm{mem_a}$, $\mathrm{disk_a}$, and $\mathrm{net_a}$ lie ($CPU_a \in [CPU_k, CPU_{k+1}]$, $mem_a \in [mem_l, mem_{l+1}]$, $disk_a \in [disk_m, disk_{m+1}]$, and $net_a \in [net_n, net_{n+1}]$). These interval lengths are used to calculate a linear model between the server's actual power consumption values for all segments defined by interval endpoints. This is done by calculating the slope for each line segment and its intercept. This slope is obtained by dividing the difference between the classical

Input: Training data set $D = \{(CPU_1, mem_1, disk_1, net_1, \ddot{P}_1), (CPU_2, mem_2, disk_2, net_2, \ddot{P}_2), \ldots, (CPU_n, mem_n, disk_n, net_n, \ddot{P}_n)\}$

Application having CPU, memory, disk, and network utilization values of CPU_a, mem_a, $disk_a$, and net_a respectively

Output: Predicted power consumption value $P_{CPU_a mem_a disk_a net_a}$

// **Calculate the regression coefficients $\alpha, \beta_1, \beta_2, \beta_3$, and β_4 for the MLR model using the training data set D.**

$$\hat{P} = \begin{bmatrix} \hat{P}_1 \\ \hat{P}_2 \\ \vdots \\ \hat{P}_n \end{bmatrix} \quad X = \begin{bmatrix} CPU_{11} & mem_{12} & disk_{13} & net_{14} \\ CPU_{21} & mem_{22} & disk_{23} & net_{24} \\ \vdots & \vdots & \vdots & \vdots \\ CPU_{n1} & mem_{n2} & disk_{n3} & net_{n4} \end{bmatrix} \quad \beta = \begin{bmatrix} \alpha \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix};$$

// **$X_{1,1} \ldots, X_{n,1}$ are the CPU utilization values, $X_{12} \ldots, X_{n,2}$ are the memory utilization values, $X_{1,3} \ldots, X_{n,3}$ are the disk utilization values, and $X_{1,4} \ldots, X_{n,4}$ are the network utilization values obtained from the training data set D.**

Minimize : $\sum_{i=1}^{n} (\hat{P}_i - \alpha - \beta_1 X_{i,1} - \beta_2 X_{i,2} - \beta_3 X_{i,3} - \beta_4 X_{i,4})^2$

// **$\acute{P}$ is the power consumption using the MLR model.**

$$\acute{P}_{CPU_a, mem_a, disk_a, net_a} = \alpha + \beta_1 * CPU_a + \beta_2 * mem_a + \beta_3 * disk_a + \beta_4 * net_a$$

//**Calculate the error correction term $\Phi_{CPU_a, mem_a, disk_a, net_a}$**

for $CPU_a \in [CPU_k, CPU_{k+1}]$, $mem_a \in [mem_l, mem_{l+1}]$, $disk_a \in [disk_m, disk_{m+1}]$, and $net_a \in [net_n, net_{n+1}]$ **do**

Equation (6) - (11)

end for

// **For the application, predict the power consumption $P_{CPU_a, mem_a, disk_a, net_a}$ using LC-MLR.**

$$P_{CPU_a, mem_a, disk_a, net_a} = \acute{P}_{CPU_a, mem_a, disk_a, net_a} + \Phi_{CPU_a, mem_a, disk_a, net_a}$$

return $P_{CPU_a, mem_a, disk_a, net_a}$

FIGURE 1. LC-MLR model building to predict power consumption of an application.

power prediction errors of the interval endpoints by the corresponding interval length. The power prediction errors are calculated for each metric utilization by fixing the remaining utilization metrics to their lower interval endpoints. The intercept is the power prediction error for the lower interval endpoints. Consequently, the differences in prediction errors between the interval's endpoints and the corresponding interval length, as well as the low-interval endpoints power prediction error affect the value of the correction term, depending on the training data set. The error correction term is calculated as explicated in Equation (6), as shown at the bottom of this page.

$$\begin{aligned} \Phi_{CPU_a, mem_a, disk_a, net_a} = e_{CPU_k, mem_l, disk_m, net_n} &+ \left[\frac{\left(e_{CPU_{k+1}, mem_l, disk_m, net_n} - e_{CPU_k, mem_l, disk_m, net_n}\right)(CPU_a - CPU_k)}{(CPU_{k+1} - CPU_k)} \right] \\ &+ \left[\frac{(e_{CPU_k, mem_{l+1}, disk_m, net_n} - e_{CPU_k, mem_l, disk_m, net_n})(mem_a - mem_l)}{(mem_{l+1} - mem_l)} \right] \\ &+ \left[\frac{(e_{CPU_k, mem_l, disk_{m+1}, net_n} - e_{CPU_k, mem_l, disk_m, net_n})(disk_a - disk_m)}{(disk_{m+1} - disk_m)} \right] \\ &+ \left[\frac{(e_{CPU_k, mem_l, disk_m, net_{n+1}} - e_{CPU_k, mem_l, disk_m, net_n})(net_a - net_n)}{(net_{n+1} - net_n)} \right] \end{aligned} \tag{6}$$

Here, $e_{CPU_k,mem_l,disk_m,net_n}$, $e_{CPU_{k+1},mem_l,disk_m,net_n}$, e_{CPU_k}, mem_{l+1}, $disk_m$, net_n, $e_{CPU_k,mem_l,disk_{m+1},net_n}$ and e_{CPU_k,mem_l}, $disk_m$, net_{n+1} are the errors calculated by taking the difference between the actual power consumption value obtained from the training data set and the predicted power consumption value obtained from the MLR model. They are calculated as follows:

$$e_{CPU_k,mem_l,disk_m,net_n} = \hat{P}_{CPU_k,mem_l,disk_m,net_n} - \acute{P}_{CPU_k,mem_l,disk_m,net_n} \quad (7)$$

$$e_{CPU_{k+1},mem_l,disk_m,net_n} = \hat{P}_{CPU_{k+1},mem_l,dissk_m,net_n} - \acute{P}_{CPU_{k+1},mem_l,disk_m,net_n} \quad (8)$$

$$e_{CPU_k,mem_{l+1},disk_m,net_n} = \hat{P}_{CPU_k,mem_{l+1},disk_m,net_n} - \acute{P}_{CPU_k,mem_{l+1},disk_m,net_n} \quad (9)$$

$$e_{CPU_k,mem_l,disk_{m+1},net_n} = \hat{P}_{CPU_k,mem_l,disk_{m+1},net_n} - \acute{P}_{CPU_k,mem_l,disk_{m+1},net_n} \quad (10)$$

$$e_{CPU_k,mem_l,disk_m,net_{n+1}} = \hat{P}_{CPU_k,mem_l,disk_m,net_{n+1}} - \acute{P}_{CPU_k,mem_l,disk_m,net_{n+1}} \quad (11)$$

Here, $\hat{P}_{CPU_k,mem_l,disk_m,net_n}$ is the actual power consumption value and $\acute{P}_{CPU_k,mem_l,disk_m,net_n}$ is the predicted power consumption value corresponding to CPU, memory, disk, and network utilization values of CPU_k, mem_l, $disk_m$, and net_n, respectively. Figure 1 shows the pseudocode for building the LC-MLR power model. In the Appendix we show an example for the calculation of Equations (4)-(11) to predict the power consumption.

V. OVERALL ARCHITECTURE AND WORKFLOW FOR ENERGY-AWARE CLOUD SCHEDULER

A cloud middleware is a software supporting a set of processes and workflow to ensure that users connect to cloud Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) based on users' SLAs [1]. Scheduling algorithms are then used by the middleware to satisfy users' SLAs. An energy-aware cloud middleware schedules users' requests in a way estimated to minimize the cloud energy consumption. In this section, we present the energy-aware cloud system model considered in this work. We also describe the overall architecture and workflow of our simulation-based energy-aware cloud scheduler for cloud operation. The cloud system model we consider in this work consists of a set m heterogeneous VMs connected to an energy-aware cloud broker as shown in Figure 2. Each VM has information about its own clock speed (CS) in GHz and processing speed in Million Instructions Per Second (MIPS) as in the Amazon Elastic Cloud [80].

Figure 3 shows the overall architecture and workflow for our energy-aware cloud scheduler. The power model is built for all the heterogeneous server architectures under study. The data workflow for developing the power model follows:

1. A workload stressing the user-level server utilization metrics (CPU, memory, disk and network) is executed on each server. This is by running a variety of benchmarks.

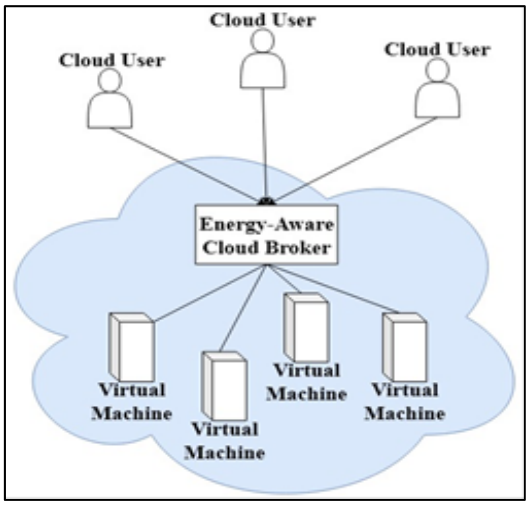

FIGURE 2. Energy-aware cloud system model.

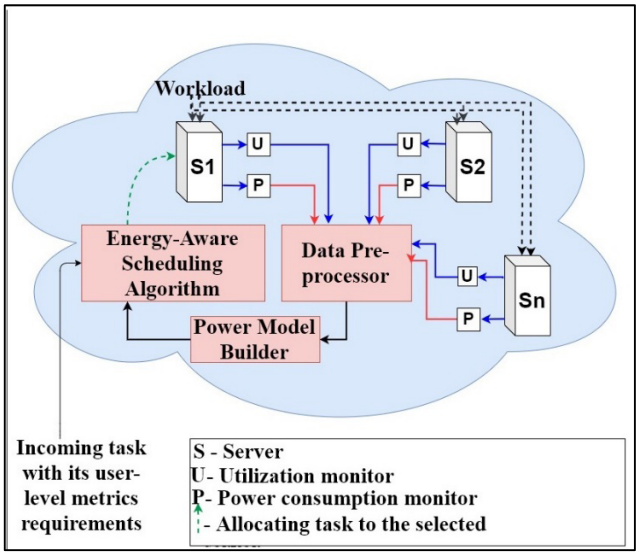

FIGURE 3. Overall architecture and workflow for proposed energy-aware scheduler.

2. The utilization monitor (U) and the power consumption monitor (P) record the user-level server utilization metrics and the power consumption values, respectively, while the workload is being executed. These values are written in a file and sent to the data pre-processor where they are synchronized and averaged to develop the data set.
3. 55% of the data set values, selected randomly, constitute the training data set which is used to build the power model by the model builder component. The builder then generates a prediction formula using LC-MLR, SLR, or PEM.
4. The model builder uses 100% of the data set values as validation data set to determine the accuracy of the developed model.
5. An energy-aware scheduling algorithm uses the prediction model developed by the power model builder in order to predict the power consumption of an incoming task to the cloud.

In our experiments, we modified our energy-aware scheduling algorithm EATSVM [56] to evaluate the developed power models under study in a cloud-computing simulation environment. In order to increase the energy-efficiency of a cloud system, EATSVM uses the power formula generated by the model builder component, to predict the increase in power consumption caused by potentially placing the task on one of the cloud servers. To schedule an incoming task having its CPU, memory, disk, and network utilization requirements on one of the cloud's VMs, EATSVM proceeds as follows:

- ➢ It calculates the completion time of the task on a VM based on the length of the task in terms of Millions of Instructions (MI) and the speed of the VM in terms of Million Instructions per Second (MIPS).
- ➢ If the VM is idle, the algorithm calculates the value of the energy function for that task on that VM. The energy function is the multiplication of the power consumed by the task and the calculated completion time of the task. The power consumption is calculated using the power model.
- ➢ If the VM is active (i.e., having ongoing task(s)), the algorithm first calculates the increase in the completion time of the ongoing task(s) and then calculates a combined energy function for the incoming task and the ongoing one(s).
- ➢ The algorithm calculates the value of the energy function and places the task on the VM having the minimum value of the energy function.

The runtime of the scheduler depends on the number of VMs in the data center and the number of tasks to be scheduled.

VI. PERFORMANCE ANALYSIS

In this section, we analyze the performance of our proposed LC-MLR power consumption model on three different servers, as compared to PEM and SLR under the same environment, experimental setup, and workload. We evaluate and compare their performance in terms of standard error of estimation using:

$$e_{est} = \sqrt{\frac{\sum_{i=1}^{n} (Actual\ value_i - Predicted\ value_i)^2}{n}} \quad (12)$$

where n is the length of the validation data set.

A. EXPERIMENTAL ENVIRONMENT

To evaluate the performance of LC-MLR, SLR and PEM models, we use a testbed of three heterogeneous servers; this testbed is part of our research laboratory as described in Table 3. The three different server types are chosen to have different architectures and capabilities.

To evaluate a model, we compare the predicted power consumptions to their actual values at different user-level server utilization metrics (CPU, memory, disk and network) and hence different workloads. A training data set of loads stressing the utilization metrics and measuring the corresponding power consumptions is thus required for each server

TABLE 3. List of server types used in the experiments.

Server 1	Sun Fire Intel_Xeon CPU core of 2.80 GHz, Dual core, with 512 KB of cache and 4 GB of memory for each core, CPU voltage rating 1.5 V, OS version CentOS 6.8(i686).
Server 2	Sun Fire X4100 with AMD_Operaton252 CPU of 2.59 GHz, dual CPU, single core, with 1MB of cache and 2GB of memory for each core, CPU voltage rating of 3.3-2.9 V, OS version Red Hat Enterprise Linux Server release 7.3 (Mapio).
Server 3	CELSIUS R940power 2 x Intel Xeon E5-2680v4 CPU (2.40 GHz, 14 cores), 8 x 32GB DDR4, 2 x HDD SAS 600GB, OS version Redhat Enterprise Linux Server RHEL 7.4 – 64-bit.

to build the studied models and validating and testing data sets are required to evaluate the accuracy of the models. The training, validating and testing data sets are obtained experimentally by running different tools and benchmarks on our servers. To generate the training and validating data sets, we use the CPU Load Generator 1.0.0 [81], the Stress 1.0.4 [82], the Vdbench 5.04.06 [83], and the iperf 3.1.3 [84] to stress the CPU, the memory, the disk I/O, and the network I/O, respectively. CPU Load Generator uses a script that generates a fixed CPU load for a finite user-defined time duration. Stress uses a defined number of VM workers of a specific memory allocation size for a defined time interval to stress the memory. Vdbench generates a configurable amount of disk I/O workloads on a server using a curve parameter of a specific Vdbench's run definition file. iperf3 generates a configurable network I/O rate between the server under study and a remote host server. We run the tools on each server and measure the values of CPU and memory utilizations using Linux perf utility [76], and disk I/O and network I/O values using the collectd tool [77].

We use a two-channel digital oscilloscope of type Tektronix – TD2012B [79] 100 MHz with 1GS/s sampling to obtain the power consumption data of these servers while the tools are running. We connect the oscilloscope to a current probe [85] and a high differential voltage probe [85] for acquiring the current and voltage signals, respectively, in real-time by running a program that we developed. Our program is implemented using the G programming language of the LABVIEW 2016 software. Our program computes the power consumption by multiplying the current and the voltage signals. We use the R environment 3.5.1 [86] for building the studied models.

To evaluate the models in an energy-aware cloud computing system, we implemented our cloud energy-aware scheduler in CloudSim 3.0.3, written in Java programming language. Therefore, we extend the power consumption class of CloudSim to include the PEM, SLR, and LC-MLR, generated by the model builder of our scheduler. We create a heterogeneous data center made of hosts of the three different server types (Table 3). We use four VM types (Table 4).

We experimentally generate in our Lab synthetic workloads for the scheduler to typify real-life applications. The workloads are generated using different benchmarks and

TABLE 4. Specifications of VM types used in the experiments.

	VM_Type 1	VM_Type 2	VM_Type 3	VM_Type 4
MIPS	2500	2000	1000	500
RAM (MB)	870	1740	1740	613

applications such as the Sysbench 1.0.17 benchmark [87], MEncoder 1.2.1 application [88] whose source code is included in the Mplayer project version SVN-r31628-4.8.5 [88], PARSEC 3.0 benchmark's Black Scholes model and Streamcluster [89], and ensemble clustering application using Weka 3.8.1 [90]. Sysbench stresses the CPU of a server by calculating the prime numbers between zero and a user-defined number. MEncoder stresses the CPU and memory by performing a video compression operation on a specified video file. We use MPEG-4 video format [91] with 1920 × 1080 resolution. The Black Scholes model utilizes the CPU, memory, and disk I/O by calculating the prices of European options' portfolio analytically using partial differential equations (PDE). Streamcluster utilizes the CPU, memory, disk I/O, and network I/O by solving an online clustering problem. The data mining ensemble clustering stresses the CPU, memory, and disk I/O by performing k-means clustering on a specific data set. We use forest cover data sets [92] consisting of geospatial descriptions of various forest types. The data contains 581,000 instances, 7 classes, and 54 attributes. We run these benchmarks and applications on each server of our experimental testbed, and measure the values of the different utilization metrics and the corresponding power consumptions. The measured utilization metrics values are then used as resource requirements for the cloud workloads.

B. EXPERIMENTS

We perform two buckets of experiments. One bucket of experiments is performed to obtain the training, validation, and testing data sets for the models under study. The second bucket simulates an energy-aware cloud data center that runs our energy-aware scheduling algorithm EATSVM with each of the models under study to compare their performance.

To generate the training and validation data sets, we stress the CPU, memory, disk, and network individually by running 4 experiments on each server in our Lab. This is by using the tools described in the above foregoing subsection. In a first experiment, we produce 30 configurable CPU loads between 0-100% at random intervals using the CPU Load Generator. For multi-core servers, we generate the CPU load on all the cores simultaneously. In a second experiment, we use Stress tool to populate the server's memory using VM workers of 30 random memory sizes. In a third experiment, we generate 30 I/O rates between 0% and 100% of the maximum I/O rate of the server under experiment at random intervals using Vdbench. In a fourth experiment, we ping the server under experiment from a remote server using 30 bandwidths between 0% and 100% of the maximum bandwidth of the server under experiment at random intervals using iperf3. Every experiment runs for 5 minutes during which we measure the values of utilization metrics and the corresponding power consumption every 1-second, and write them to a file. We calculate the average utilizations for each metric and the average power consumption over the 5 minute period. We repeat each experiment 25 times and compute the average of all the averages. The power model builder randomly selects 55% of the data set generated experimentally as the training data set to develop the models under study; 100% of the generated data set is used for validation.

Several works in Cloud data centers propose energy-aware VM placement [44], [45], [58]–[60], [62], [63] and task scheduling [2], [20], [49], [64], [66], [68], [69] algorithms for energy savings. These algorithms use a power model to predict the value of energy consumption for energy-aware optimal scheduling. Therefore, we verify the performance of the studied power models in this work using our energy-aware scheduling algorithm to compare the performance of the power models in a cloud data center using the same environmental setup and workload. We first simulate the data center with an increasing number of hosts (50, 250, 500, 800, and 1000). The host types used for the simulation of the data center are of the same specifications as the ones used in our Lab, and are equally distributed in the simulated data center. We then create 1500 VMs with the four VM types as shown in Table 4 equally distributed.

We generate 5 synthetic workloads by running 5 different benchmarks and applications on the servers' testbed in our Lab to produce the testing data set. The workloads represent a diverse set of applications which stress one or more user-level utilization metrics. We run Sysbench to generate a CPU-intensive workload by calculating the prime numbers between 0 and 20,000,000 using different number of threads. The number of threads is increased randomly from zero to the total number of threads of the server under test. To produce a CPU and memory-intensive workload, we run the MEncoder application which compresses a video file. To assess the performance of the models against increasing sizes, we compressed 10 different AVI format [93] video files with increasing sizes from 5GB to 50GB at an interval of 5GB using the compression function of MEncoder. To generate an intensive workload of CPU, memory, and disk, we use the Black Scholes application to calculate the prices of a 65,536 European options portfolio. We also use the ensemble clustering application to perform k-means clustering of data sets with different number of instances (2799, 279000, 2790000, and 5580000). We use the Streamcluster application to generate an intensive workload of CPU, memory, disk, and network by performing an online stream clustering for native input options having 1,000,000 inputs points and 218 dimensions. We collect the utilization metrics and the corresponding power consumption values every 1 second for each experiment and calculate the average. We repeat every experiment 25 times and calculate the average of the averages. The data set generated for each benchmark/application

on each server is replicated 100 times and shuffled randomly to create a large workload.

To evaluate the impact of each of the models under study on energy-aware scheduling, we calculate the standard error of estimation of the predicted cloud data center energy compared to the actual one post scheduling. We also calculate the standard error of estimation of SLAVs when using predicted power versus actual. In the experiments, we first schedule the workload on the hosts randomly. While the workload is still running, we schedule the same workload on the host by using EATSVM. We repeat this scheduling in 2 scenarios: 1) when the scheduler uses a power model, and 2) when the schedule uses the actual power consumption values from the testing data set. In each scenario, we measure the energy consumption and SLAVs of the data center and calculate the corresponding errors of estimation. We repeat both scenarios 3 times (referred to as a run), and calculate the average errors of estimations for energy and SLAVs. The run is repeated for each of the 5 workloads and for each power model under study in a dynamic data center. The energy consumptions of the running VMs, based on our Lab experimental power consumption baseline, are added to find that of the data center after the execution of the workload. The VM's energy consumption is calculated by multiplying its power consumption of the machine and the time it has been running. The SLAVs of the VMs are also added up to calculate the SLAVs of the cloud.

We measure the SLAVs of the VMs (due to migrations) post scheduling. This is in terms of the ratio of the CPU capacity given to the VMs relative to the total capacity of a host experiencing 100% CPU utilization using the equation [94]:

$$\text{SLAV} = \text{OTF}^{*}\text{PDM} \tag{13}$$

where OTF (Overload/Time Fraction) is the fraction of the time during which, the active hosts experience 100% CPU utilization and PDM (Performance Degradations due to Migrations) is the overall performance degradation due to the migration of VMs. OTF and PDM are calculated using Equations (14) and (15), respectively:

$$\text{OTF} = (\frac{1}{N}) \sum_{i=1}^{N} \frac{\text{T}_{\text{si}}}{\text{T}_{\text{ai}}} \tag{14}$$

$$\text{PDM} = \left(\frac{1}{\text{M}}\right) \sum_{k=1}^{M} \frac{C_{dk}}{\text{C}_{\text{rk}}} \tag{15}$$

Here, N is the number of hosts, T_{si} is the total time during which host i has experienced utilization of 100% leading to SLA violations, T_{ai} is the total time of the host i being active, M is the total number of VMs, C_{dk} is the estimate of the performance degradation of VM_{k} caused by migrations, C_{rk} is the total CPU capacity requested by VM_{k} during its lifetime, C_{dk} is estimated to be 10% of the CPU utilization in MIPS during all migrations of the VM_{k}.

C. EXPERIMENTAL RESULTS ANALYSIS

In this section, we evaluate our experimental results comparing between the studied models LC-MLR, SLR and PEM using our servers' testbed as well as for a dynamic energy-aware cloud computing system. We also give insights on and conclusions of these evaluations. In particular, we explain the reasons for the rationale behind the models' performance.

Figure 4 shows that PEM has the highest average standard error of estimation compared to SLR, and LC-MLR, while LC-MLR has the least error. This is because PEM is based only on the power consumptions endpoint values P_{max} and P_{min} and consists of a straight line model in which all the possible predicted values lie. Therefore, the model does not consider the implications of other power consumption data between P_{min} and P_{max} for its predictions. However, the SLR and LC-MLR models compute a linear regression plane to best fit data distribution while minimizing the sum of the squares of vertical deviations from each data point to the plane.

Comparing the performance of LC-MLR with SLR, based on our results (Figure 4), LC-MLR has the least standard error of estimation. This is thanks to the piecewise planar linearization in LC-MLR, whereas SLR computes a single regression line for the entire data set. This indicates an advantage of modeling the power consumption data collected on our experimental testbed as linear within intervals rather than over the entire range (i.e., the advantage of using a piecewise linear model).

In summary, LC-MLR model has the least standard error of estimation compared to the SLR and PEM models. For the validation data set, the standard error of estimation in server 1 is 4.3142, 6.7853 and 8.1312, in server 2 is 4.0913, 5.0252 and 7.2509, and in server 3 is 5.5454, 8.0324 and 11.8538, using LC-MLR, SLR and PEM respectively.

Figure 5 shows that our energy-aware scheduling algorithm using PEM has the highest standard error of estimation of energy consumption, compared to using SLR and LC-MLR. This is due to its high error in the prediction of power consumption. On the other hand, our algorithm using LC-MLR predicts the energy consumption with the least standard error of estimation, very close to real energy data.

Our experimental results (Figure 5) reveal that the PEM, SLR errors are higher using MEncoder, Portfolio, K-means, and Streamcluster applications than Sysbench. This is because SLR and PEM models use only CPU utilization as an independent variable. Sysbench heavily involves the CPU during its execution. On the other hand, MEncoder involves CPU and memory utilizations, Portfolio and K-means involve CPU, memory, and disk I/O operations, and Streamcluster involves CPU, memory, disk and network I/O operations.

Figure 6 shows the standard error of estimation in SLA violations of our energy-aware scheduling algorithm using the studied models. It shows that our algorithm using PEM has the highest standard error of estimation, while using LC-MLR has the least. This is due to the low accuracy of the prediction of power consumption by PEM for a running

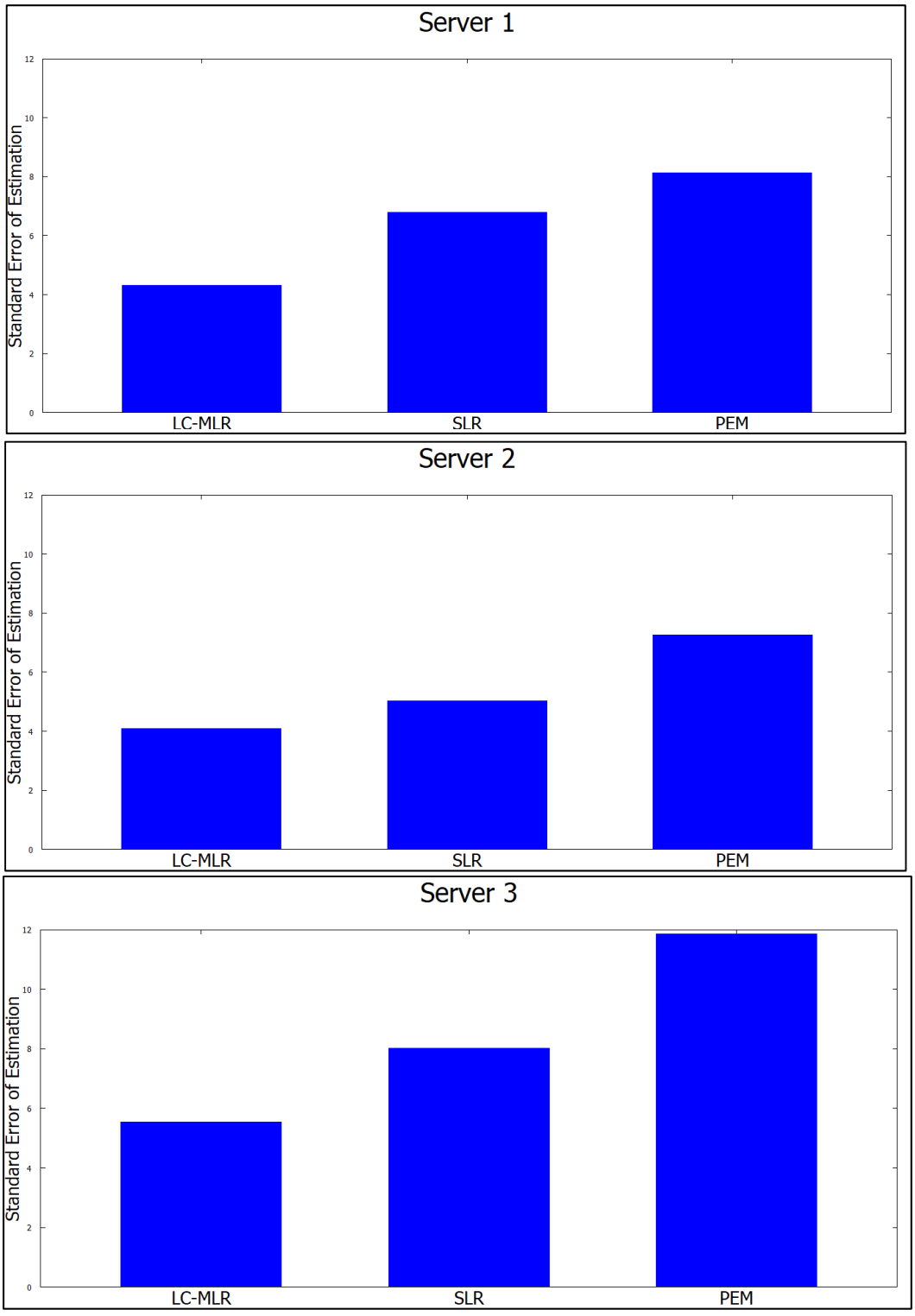

FIGURE 4. Standard error of estimation of LC-MLR. SLR and PEM using the validation data set.

task, leading to more server consolidation. Consequently, this leads to more SLA violations.

In summary, the average standard error of estimation of energy consumption when scheduling tasks using our energy-aware scheduling algorithm for workloads in all our scenarios of a dynamic data center is 5.0096, 25.8878, and 34.6062 using LC-MLR, SLR, and PEM respectively. The average standard error of estimation in SLA violations is

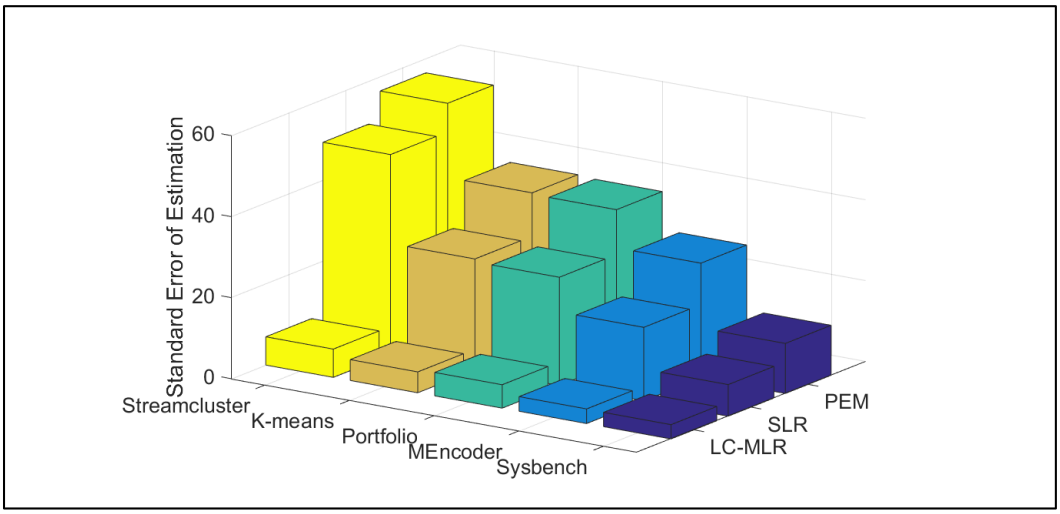

FIGURE 5. Standard error of estimation of energy consumption using LC-MLR, SLR and PEM for EATSVM with increasing number of hosts for different workloads.

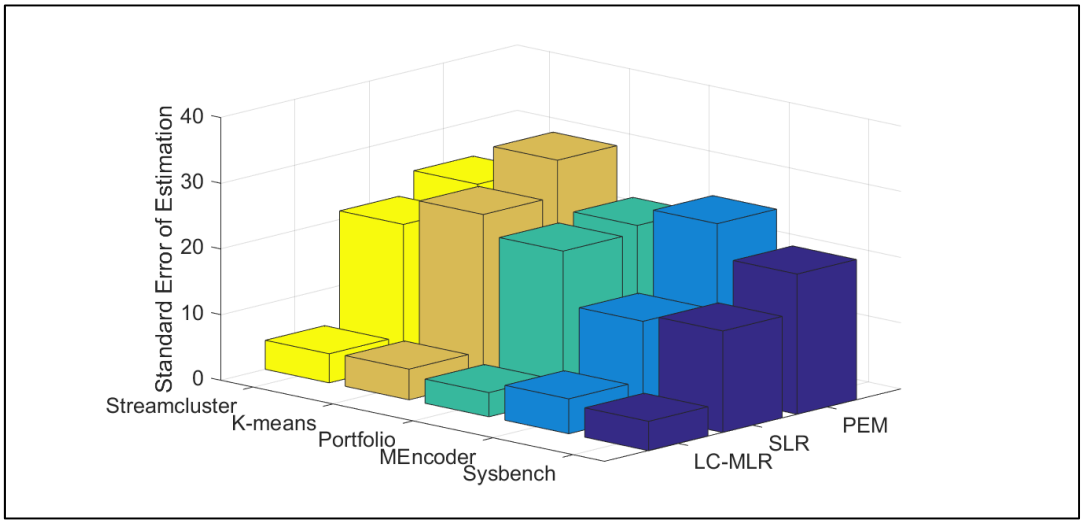

FIGURE 6. Standard error of estimation of SLA violations using LC-MLR, SLR and PEM for EATSVM with increasing number of hosts using different workloads.

4.5134, 19.8068, and 25.3911 using LC-MLR, SLR, and PEM respectively.

VII. CONCLUSION AND SUMMARY

Cloud computing is a technology that is being rapidly adopted and will be extensively used, especially with the emergence of IoT and Big Data analytics on enormous amounts of data collected from ubiquitous devices. Power consumption of cloud computing infrastructure thus becomes a crucial issue. Along with this trend, the need for green computing environments is also becoming more pressing. Therefore, many research works propose cloud computing scheduling algorithms to reduce cloud energy consumption. Most such algorithms use PEM to predict power consumption of cloud applications. PEM uses CPU utilization as the independent variable because CPU use is considered the dominant factor in server power consumption. Other linear power models were also proposed in the literature. Very few works validate those models, but these haven't employed identical experimental setups and workload data. This makes it difficult to perform comparisons of efficacy of the models from existing validations.

In this work, we have presented a classification of linear power models based on user-level server utilization metrics, such as CPU, memory, network and disk utilizations. We evaluated CPU-based PEM and SLR models, using identical experimental setups and workload data. In addition, we developed LC-MLR, an extended version of MLR, and compared its performance to SLR and PEM, the latter having been used most frequently in research works in green computing. To perform the comparisons, we utilized a testbed with several available hardware architectures, and used it for several

experiments for the evaluation. We also proposed an energy-aware simulation framework which allowed us to implement and experiment with energy-aware scheduling algorithms using different power models.

Our experimental results show that LC-MLR gives the least standard error of estimation of power consumption with up to 5.5454 using the validation data set. The results show that the relative performance of these models persists in the different servers' architecture considered in our experimental testbed. The results remain valid for energy consumption and SLAs violation using our energy-aware scheduling algorithm giving the least standard error of estimation of energy prediction with up to 6.9832, and of SLAs violation with up to 5.3141.

When developing energy-aware algorithms in cloud computing, it is important to choose an accurate power consumption model to enhance energy savings and reduce SLAs violations. When developing a power model to predict the power consumption of a running application on a server, the following requirements should be considered:

1) *Considering Data Range*: the local regression model (LR) has an important implication in accurately predicting power consumption.
2) *Piecewise Linearity*: linearity in different local regions of the data plot (LC-MLR) that replaces purely LR leads to more accurate prediction than LR.
3) *User-Level Server Utilization Metrics*: other server's utilization metrics such as memory, disk, and network contribute to a server's power consumption. Metrics other than CPU utilization should be then considered by the power models.

In future research work, we propose investigations in the following directions. It would be valuable to validate and compare the different linear and non-linear power models in the literature, as the accuracy of the model directly impacts the accuracy of the energy-efficient algorithms using it. In addition, energy-efficient VM placement and task scheduling algorithms in cloud data centers should take into consideration the energy consumption of building the power modes. Therefore, analytical and experimental evaluation of building power models in terms of energy consumption is then crucial for the selection of the power model in cloud computing data centers. Furthermore, the runtime of energy-efficient algorithms and corresponding energy consumption should be studied.

APPENDIX
EXAMPLE FOR PREDICTING THE POWER CONSUMPTION OF AN APPLICATION USING LC-MLR

In this example, we would like to predict the power consumption, using LC-MLR (Equation 4), of an application having CPU utilization $\text{CPU}_\text{a} = 50.03\%$, memory utilization $\text{mem}_\text{a} = 50.31\%$, disk utilization $\text{disk}_\text{a} = 49.73\%$, and network utilization $\text{net}_\text{a} = 49.89\%$. We first predict the power consumption using MLR. Then, we calculate the error correction term.

TABLE 5. Training data set obtained from real experiments.

CPU Utilization	Memory Utilization	Disk utilization	Network utilization	Power consumption ($\hat{P}$)
38.87%	40.98%	38.85%	38.30%	243.5366
59.08%	40.98%	38.85%	38.30%	261.2763
38.87%	58.72%	38.85%	38.30%	244.9026
38.87%	40.98%	60.01%	38.30%	246.0179
38.87%	40.98%	38.85%	62.62%	252.6219

1. *Prediction of Power Consumption using MLR*: we calculate the values of the regression coefficients (α, β_1, β_2, β_3, and β_4) by using the training data set (Table 5). We use the R programming language to obtain these coefficients values. We calculate the predicted power consumption for the application by substituting the coefficients and the utilization metrics by their corresponding values in Equation 5 as follows:

$$\begin{aligned}\acute{P}_{CPU_a,mem_a,disk_a,net_a} &= 204.2955 + (95.42945 * 50.03\%) \\ &+ (-13.1792 * 50.31\%) \\ &+ (0.46406 * 49.73\%) \\ &+ (24.13667 * 49.89\%) = 257.2194\end{aligned}$$

2. *Calculation of the Error Correction Term*: to calculate the error correction term (Equation 6), we proceed as follows:
   a. The MLR model uses the training data set to find the intervals where CPU, memory, disk, and network utilization values of the application lie. In our example, the intervals are:

$$\begin{aligned}CPU_a &= 50.03\% \epsilon [CPU_k \\ &= 38.87\%, CPU_{k+1} \\ &= 59.08\%] \\ mem_a &= 50.31\% \epsilon [mem_l \\ &= 40.98\%, mem_{l+1} \\ &= 58.72\%] \\ disk_a &= 49.73\% \epsilon [disk_m \\ &= 38.85\%, disk_{m+1} \\ &= 60.01\%] \\ net_a &= 49.89\% \epsilon [net_n \\ &= 38.30\%, net_{n+1} \\ &= 62.62\%]\end{aligned}$$

   b. The error correction term involves linear functions between the interval endpoints for each utilization. These are evaluated by calculating the intercept and slope for each line. The intercept is calculated using Equation (7) by substituting the values of the actual

power consumption and the predicted one using MLR for the lower interval endpoints as follows:

$$e_{CPU_k,mem_l,disk_m,net_n} = 243.53660 - 245.0511 = -1.5145$$

The slope for each line is calculated by dividing the difference between the MLR prediction errors at the interval endpoints by the interval length. The prediction errors for the interval endpoints for CPU, memory, disk, and network utilizations are calculated using Equations (8), (9), (10), and (11), respectively. This is done by substituting the values of the actual power consumption and the predicted one using MLR for the upper interval endpoints as follows:

$$e_{CPU_{k+1},mem_l,disk_m,net_n} = 261.27630 - 264.3374 = -3.0611$$

$$e_{CPU_k,mem_{l+1},disk_m,net_n} = 247.90260 - 242.7118 = 5.19083$$

$$e_{CPU_k,mem_l,disk_{m+1},net_n} = 246.01790 - 244.9529 = 1.06503$$

$$e_{CPU_k,mem_l,disk_m,net_{n+1}} = 252.62190 - 250.9235 = 1.69838$$

Consequently, the error correction term is calculated using Equation (6) as follows:

$$Ø_{CPU_a,mem_a,disk_a,net_a} = -1.5145 - 0.8877 + 3.5265 + 1.3263 + 1.4730 = 3.9236$$

The results from Steps 1) and 2) above are used in Equation (4) to obtain the predicted power consumption for the application as follows:

$$P_{CPU_a,mem_a,disk_a,net_a} = 257.2194 + 3.9236 = 261.143$$

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments that helped to improve the content, quality, and presentation of the article.

REFERENCES

[1] P. Mell and T. Grance, "The NIST definition of cloud computing: Recommendations of the national institute of standards and technology," NIST, Gaithersburg, MD, USA, NIST Special Publication 800-145, 2011, doi: 10.1136/emj.2010.096966.

[2] L. Ismail and A. A. Fardoun, "Energy-aware task scheduling (EATS) framework for efficient energy in smart cities cloud computing infrastructures," *Int. J. Thermal Environ. Eng.*, vol. 13, no. 1, pp. 37–48, 2016, doi: 10.5383/ijtee.13.01.007.

[3] A. Sheikhi, M. Rayati, A. M. Ranjbar, S. Sattari, and S. Bahrami, "A cloud computing framework on demand side management game in smart energy hubs," *Int. J. Elect. Power Energy Syst.*, vol. 64, pp. 1007–1016, Jan. 2015, doi: 10.1016/j.ijepes.2014.08.020.

[4] P. Delforge. (2015). *America's Data Centers Consuming and Wasting Growing Amounts of Energy | NRDC*. Accessed: Aug. 16, 2018. [Online]. Available: https://www.nrdc.org/resources/americas-data-centers-consuming-and-wasting-growing-amounts-energy

[5] D. Meisner and T. F. Wenisch, "Peak power modeling for data center servers with switched-mode power supplies," in *Proc. 16th ACM/IEEE Int. Symp. Low Power Electron. Design*, 2010, pp. 319–324, doi: 10.1145/1840845.1840911.

[6] S. Greenberg, E. Mills, and B. Tschudi, "Best practices for data centers: Lessons learned from benchmarking 22 data centers," in *Proc. ACEEE Summer Energy Efficiency Buildings*, 2006, pp. 76–87, doi: 10.1016/j.energy.2012.04.037.

[7] T. Brey, P. Lembke, J. Prisco, and K. A. Emerson, "Case study: The ROI of cooling system energy efficiency upgrades," White Paper # 39, 2011, pp. 1–42.

[8] O. Sarood, A. Langer, A. Gupta, and L. Kale, "Maximizing throughput of overprovisioned HPC data centers under a strict power budget," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, Nov. 2014, pp. 807–818, doi: 10.1109/SC.2014.71.

[9] D. Shin, J. Kim, J. Choi, S. W. Chung, E.-Y. Chung, and N. Chang, "Energy-optimal dynamic thermal management for green computing," in *Proc. Int. Conf. Comput. Design (ICCAD)*, Nov. 2009, pp. 652–657, doi: 10.1145/1687399.1687520.

[10] B. Mills, T. Znati, R. Melhem, K. B. Ferreira, and R. E. Grant, "Energy consumption of resilience mechanisms in large scale systems," in *Proc. 22nd Euromicro Int. Conf. Parallel, Distrib., Netw.-Based Process. (PDP)*, 2014, pp. 528–535, doi: 10.1109/PDP.2014.111.

[11] T. Horvath and K. Skadron, "Multi-mode energy management for multi-tier server clusters," in *Proc. Int. Conf. Parallel Archit. Compilation Techn. (PACT)*, 2008, pp. 270–279, doi: 10.1145/1454115.1454153.

[12] S.-W. Ham, M.-H. Kim, B.-N. Choi, and J.-W. Jeong, "Simplified server model to simulate data center cooling energy consumption," *Energy Buildings*, vol. 86, pp. 328–339, Jan. 2015, doi: 10.1016/j.enbuild.2014.10.058.

[13] E. N. Elnozahy, M. Kistler, and R. Rajamony, "Energy-efficient server clusters," in *Proc. Int. Workshop Power-Aware Comput. Syst.*, 2003, pp. 179–197, doi: 10.1007/3-540-36612-1_12.

[14] W. Wu, L. Jin, P. Liu, S. X.-D. Tan, and J. Yang, "Efficient power modeling and software thermal sensing for runtime temperature monitoring," *ACM Trans. Des. Automat. Electron. Syst.*, vol. 12, no. 3, 2007, Art. no. 25, doi: 10.1145/1255456.1255462.

[15] R. Buyya, A. Beloglazov, and J. Abawajy, "Energy-efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges," in *Proc. Int. Conf. Parallel Distrib. Process. Technol. Appl.*, 2010, pp. 1–12, doi: 10.1002/cpe.1867.

[16] T. Heath, Jr., A. P. Centeno, L. Ramos, Y. Jaluria, R. Bianchini, and P. George, "Mercury and freon: Temperature emulation and management for server systems," in *Proc. 12th Int. Conf. Architectural Support Program. Lang. Oper. Syst.*, 2006, pp. 106–116, doi: 10.1145/1168857.1168872.

[17] A. Qureshi, R. Weber, J. Guttag, B. Maggs, and H. Balakrishnan, "Cutting the electric bill for Internet-scale systems," *ACM SIGCOMM Comput. Commun. Rev.* vol. 39, pp. 123–134, Oct. 2009, doi: 10.1145/1594977.1592584.

[18] D. Gmach, J. Rolia, L. Cherkasova, and A. Kemper, "Resource pool management: Reactive versus proactive or let's be friends," *Comput. Netw.* vol. 53, pp. 2905–2922, Dec. 2009, doi: 10.1016/j.comnet.2009.08.011.

[19] H. Cheung, S. Wang, C. Zhuang, and J. Gu, "A simplified power consumption model of information technology (IT) equipment in data centers for energy system real-time dynamic simulation," *Appl. Energy*, vol. 222, pp. 329–342, Jul. 2018, doi: 10.1016/j.apenergy.2018.03.138.

[20] Y. C. Lee and A. Y. Zomaya, "Energy efficient utilization of resources in cloud computing systems," *J. Supercomput.*, vol. 60, pp. 268–280, May 2012, doi: 10.1007/s11227-010-0421-3.

[21] P. Raycroft, R. Jansen, M. Jarus, P. R. Brenner, "Performance bounded energy efficient virtual machine allocation in the global cloud," *Sustain. Comput., Inform. Syst.*, vol. 4, pp. 1–9, Mar. 2014, doi: 10.1016/j.suscom.2013.07.001.

[22] X. Dai, J. M. Wang, and B. Bensaou, "Energy-efficient virtual machines scheduling in multi-tenant data centers," *IEEE Trans. Cloud Comput.*, vol. 4, no. 2, pp. 210–221, Jun. 2016, doi: 10.1109/TCC.2015.2481401.

[23] N. K. Sharma and G. R. M. Reddy, "Multi-objective energy efficient virtual machines allocation at the cloud data center," *IEEE Trans. Services Comput.*, vol. 12, no. 1, pp. 158–171, Jan./Feb. 2019, doi: 10.1109/TSC.2016.2596289.
[24] X. Ye, Y. Yin, and L. Lan, "Energy-efficient many-objective virtual machine placement optimization in a cloud computing environment," *IEEE Access*, vol. 5, pp. 16006–16020, 2017, doi: 10.1109/ACCESS.2017.2733723.
[25] M. Pedram and I. Hwang, "Power and performance modeling in a virtualized server system," in *Proc. 39th Int. Conf. Parallel Process. Workshops*, 2010, pp. 520–526, doi: 10.1109/ICPPW.2010.76.
[26] X. Zhang, J.-J. Lu, X. Qin, and X.-N. Zhao, "A high-level energy consumption model for heterogeneous data centers," *Simul. Model. Pract. Theory*, vol. 39, pp. 41–55, Dec. 2013, doi: 10.1016/j.simpat.2013.05.006.
[27] L. Hongyou, W. Jiangyong, W. Junfeng, L. Tang, and P. Jian, "Energy-aware scheduling scheme using workload-aware consolidation technique in cloud data centres," *China Commun.*, vol. 10, pp. 114–124, Dec. 2013, doi: 10.1109/CC.2013.6723884.
[28] R. Raghavendra, P. Ranganathan, Z. Wang, X. Zhu, and V. Talwar, "No power struggles: Coordinated multi-level power management for the data center," *ACM SIGARCH Comput. Archit. News*, vol. 36, pp. 48–59, Mar. 2008.
[29] C. Gong, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao, "Energy-aware server provisioning and load dispatching for connection-intensive Internet services," in *Proc. USENIX Symp Netw. Syst. Design Implement.*, 2008, pp. 337–350, doi: 10.1109/INFCOM.2012.6195719.
[30] J. L. Berral, Í. Goiri, F. Julià, J. Guitart, R. Gavaldà, J. Torres, and R. Nou, "Towards energy-aware scheduling in data centers using machine learning," in *Proc. 1st Int. Conf. Energy-Efficient Comput. Netw.*, 2010, pp. 215–224.
[31] J. L. Berral, R. Gavalda, and J. Torres, "Adaptive scheduling on power-aware managed data-centers using machine learning," in *Proc. IEEE/ACM 12th Int. Conf. Grid Comput.*, Sep. 2011, pp. 66–73.
[32] D. Economou, S. Rivoire, C. Kozyrakis, and P. Ranganathan, "Full-system power analysis and modeling for server environments," in *Proc. Int. Symp. Comput. Archit.*, 2006, pp. 807–812.
[33] J. W. Smith, A. Khajeh-Hosseini, J. S. Ward, and I. Sommerville, "CloudMonitor: Profiling power usage," in *Proc. IEEE 5th Int. Conf. Cloud Comput. (CLOUD)*, Jun. 2012, pp. 3–4.
[34] A. Kansal, F. Zhao, N. Kothari, A. A. Bhattacharya, and J. Liu, "Virtual machine power metering and provisioning," in *Proc. 1st ACM Symp. Cloud Comput. (SoCC)*, 2010, pp. 39–50, doi: 10.1145/1807128.1807136.
[35] Y. Li, Y. Wang, B. Yin, and L. Guan, "An online power metering model for cloud environment," *Proc. IEEE 11th Int Symp. Netw. Comput. Appl. (NCA)*, Aug. 2012, pp. 175–180, doi: 10.1109/NCA.2012.10.
[36] J. D. Davis, S. Rivoire, M. Goldszmidt, and E. K. Ardestani, "CHAOS: Composable Highly Accurate OS-based power models," in *Proc. IEEE Int. Symp. Workload Characterization (IISWC)*, Nov. 2012, pp. 153–163, doi: 10.1109/IISWC.2012.6402920.
[37] I. Alan, E. Arslan, and T. Kosar, "Energy-aware data transfer tuning," in *Proc. 14th IEEE/ACM Int. Symp. Cluster, Cloud, Grid Comput. (CCGrid)*, 2014, pp. 626–634, doi: 10.1109/CCGrid.2014.117.
[38] M. Tang and S. Pan, "A hybrid genetic algorithm for the energy-efficient virtual machine placement problem in data centers," *Neural Process. Lett.*, vol. 41, no. 2, pp. 211–221, 2015, doi: 10.1007/s11063-014-9339-8.
[39] G. Sun, V. Anand, D. Liao, C. Lu, X. Zhang, and N.-H. Bao, "Power-efficient provisioning for online virtual network requests in cloud-based data centers," *IEEE Syst. J.*, vol. 9, no. 2, pp. 427–441, Jun. 2015, doi: 10.1109/JSYST.2013.2289584.
[40] C.-H. Lien, Y.-W. Bai, and M.-B. Lin, "Estimation by software for the power consumption of streaming-media servers," *IEEE Trans. Instrum. Meas.*, vol. 56, no. 5, pp. 1859–1870, Oct. 2007, doi: 10.1109/TIM.2007.904554.
[41] C.-J. Tang and M.-R. Dai, "Dynamic computing resource adjustment for enhancing energy efficiency of cloud service data centers," in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, Dec. 2011, pp. 1159–1164, doi: 10.1109/SII.2011.6147613.
[42] S. Janacek, K. Schröder, G. Schomaker, W. Nebel, M. Rüschen, and G. Pistoor, "Modeling and approaching a cost transparent, specific data center power consumption," in *Proc. Int. Conf. Energy Aware Comput. (ICEAC)*, Dec. 2012, pp. 1–6, doi: 10.1109/ICEAC.2012.6471012.
[43] Z. Bagheri and K. Zamanifar, "Enhancing energy efficiency in resource allocation for real-time cloud services," in *Proc. 7th Int. Symp. Telecommun. (IST)*, 2014, pp. 701–706, doi: 10.1109/ISTEL.2014.7000793.
[44] R. Sinha, N. Purohit, and H. Diwanji, "Power aware live migration for data centers in cloud using dynamic threshold," *Int. J. Comput. Technol. Appl.*, vol. 2, no. 6, pp. 2041–2046, 2011, doi: 10.1.1.658.4169.
[45] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," *Future Gener. Comput. Syst.*, vol. 28, pp. 755–768, May 2012, doi: 10.1016/j.future.2011.04.017.
[46] R. Patel, H. Patel, and S. Patel, "Efficient resource allocation in cloud computing," *Int. J. Technol. Res. Eng.*, vol. 2, pp. 1253–1260, Mar. 2015.
[47] G. Han, W. Que, G. Jia, and L. Shu, "An efficient virtual machine consolidation scheme for multimedia cloud computing," *Sensors*, vol. 16, no. 2, p. 246, 2016, doi: 10.3390/s16020246.
[48] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," *ACM SIGARCH Comput. Archit. News*, vol. 35, pp. 13–23, May 2007, doi: 10.1145/1273440.1250665.
[49] B. Qureshi, "Profile-based power-aware workflow scheduling framework for energy-efficient data centers," *Future Gener. Comput. Syst.*, vol. 94, pp. 453–467, May 2019, doi: 10.1016/j.future.2018.11.010.
[50] R. Buyya, A. Beloglazov, and J. Abawajy, "Energy-efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges," in *Proc. Int. Conf. Parallel Distrib. Process. Techn. Appl. (PDPTA)*, Las Vegas, NV, USA, Jul. 2010, pp. 1–12.
[51] *Complexity of Linear Regression Model*. Accessed: Apr. 14, 2019. [Online]. Available: https://www.thekerneltrip.com/machine/learning/computational-complexity-learning-algorithms/
[52] L. Li, "A new complexity bound for the least-squares problem," *Comput. Math. Appl.*, vol. 31, pp. 15–16, Jun. 1996.
[53] Y. Jin, Y. Wen, Q. Chen, and Z. Zhu, "An empirical investigation of the impact of server virtualization on energy efficiency for green data center," *Comput. J.*, vol. 56, pp. 977–990, Aug. 2013, doi: 10.1093/comjnl/bxt017.
[54] *Find a Linear Regression Equation*. Accessed: Jun. 24, 2019. [Online]. Available: https://www.statisticshowto.datasciencecentral.com/probability-and-statistics/regression-analysis/find-a-linear-regression-equation/
[55] A. Butterfield and G. E. Ngondi, *A Dictionary of Computer Science*. Oxford, U.K.: Oxford Univ. Press, 2016.
[56] L. Ismail and H. Materwala, "EATSVM: Energy-aware task scheduling on cloud virtual machines," *Procedia Comput. Sci.*, vol. 135, pp. 248–258, 2018, doi: 10.1016/j.procs.2018.08.172.
[57] 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," *Softw., Pract. Exper.*, vol. 41, pp. 23–50, Jan. 2011.
[58] 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 Comput. Pract. Exper.*, vol. 24, pp. 1397–1420, Sep. 2012, doi: 10.1002/cpe.1867.
[59] N. Kim, J. Cho, and E. Seo, "Energy-credit scheduler: An energy-aware virtual machine scheduler for cloud systems," *Future Gener. Comput. Syst.*, vol. 32, pp. 128–137, Mar. 2014, doi: 10.1016/j.future.2012.05.019.
[60] M. R. Chowdhury, M. R. Mahmud, and R. M. Rahman, "Implementation and performance analysis of various VM placement strategies in CloudSim," *J. Cloud Comput.*, vol. 4, Nov. 2015, Art. no. 20, doi: 10.1186/s13677-015-0045-5.
[61] F. Farahnakian, A. Ashraf, T. Pahikkala, P. Liljeberg, J. Plosila, I. Porres, and H. Tenhunen, "Using ant colony system to consolidate VMs for green cloud computing," *IEEE Trans. Services Comput.*, vol. 8, no. 2, pp. 187–198, Mar./Apr. 2015, doi: 10.1109/TSC.2014.2382555.
[62] X. Li, P. Garraghan, X. Jiang, Z. Wu, and J. Xu, "Holistic virtual machine scheduling in cloud datacenters towards minimizing total energy," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 6, pp. 1317–1331, Jun. 2018, doi: 10.1109/TPDS.2017.2688445.
[63] S. Ilager, K. Ramamohanarao, and R. Buyya, "ETAS: Energy and thermal-aware dynamic virtual machine consolidation in cloud data center with proactive hotspot mitigation," *Concurrency Comput.*, vol. 31, Sep. 2019, Art. no. e5221, doi: 10.1002/cpe.5221.
[64] Y. Changtian and Y. Jiong, "Energy-aware genetic algorithms for task scheduling in cloud computing," in *Proc. 7th ChinaGrid Annu. Conf. ChinaGrid*, 2012, pp. 43–48, doi: 10.1109/ChinaGrid.2012.15.
[65] N. A. Mehdi, H. Ali, A. Amer, and Z. T. Abdul-Mehdi, "Two-phase provisioning for HPC tasks in virtualized datacenters," in *Proc. Int. Conf. Emerg. Trends Comput. Electron. Eng. (ICETCEE)*, Dubai, United Arab Emirates, 2012, pp. 29–35.

[66] W. Huai, Z. Qian, X. Li, G. Luo, and S. Lu, "Energy aware task scheduling in data centers," *J. Wireless Mobile Netw.*, vol. 4, no. 2, pp. 18–38, 2013.
[67] C.-M. Wu, R.-S. Chang, and H.-Y. Chan, "A green energy-efficient scheduling algorithm using the DVFS technique for cloud datacenters," *Future Gener. Comput. Syst.*, vol. 37, pp. 141–147, Jul. 2014, doi: 10.1016/j.future.2013.06.009.
[68] S. Singh and I. Chana, "EARTH: Energy-aware autonomic resource scheduling in cloud computing," *J. Intell. Fuzzy Syst.*, vol. 30, no. 3, pp. 1581–1600, 2016, doi: 10.3233/IFS-151866.
[69] S. Singh, I. Chana, M. Singh, and R. Buyya, "SOCCER: Self-optimization of energy-efficient cloud resources," *Cluster Comput.*, vol. 19, no. 4, pp. 1787–1800, 2016, doi: 10.1007/s10586-016-0623-4.
[70] B. Qureshi, S. Alwehaibi, and A. Koubaa, "On power consumption profiles for data intensive workloads in virtualized Hadoop clusters," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, May 2017, pp. 42–47.
[71] M. Dayarathna, Y. Wen, and R. Fan, "Data center energy consumption modeling: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 732–794, 1st Quart., 2016, doi: 10.1109/COMST.2015.2481183.
[72] T. Li and L. K. John, "Run-time modeling and estimation of operating system power consumption," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 31, pp. 160–171, Jun. 2003, doi: 10.1145/885651.781048.
[73] H. Nagasaka, A. Nukada, T. Endo, S. Matsuoka, and N. Maruyama, "Statistical power modeling of GPU kernels using performance counters," in *Proc. Int. Conf. Green Comput.*, Aug. 2010, pp. 115–122.
[74] H. Liu, C.-Z. Xu, H. Jin, J. Gong, and X. Liao, "Performance and energy modeling for live migration of virtual machines," in *Proc. 20th Int. Symp. High Perform. Distrib. Comput. (HPDC)*, 2011, pp. 171–182, doi: 10.1145/1996130.1996154.
[75] J. Neter, M. H. Kutner, C. J. Nachtsheim, and W. Wasserman, *Applied Linear Statistical Models*. Chicago, IL, USA: Irwin, 1997.
[76] *Linux Perf Examples*. Accessed: Sep. 2, 2019. [Online]. Available: http://www.brendangregg.com/perf.html
[77] *Collectd*. Accessed: Sep. 1, 2019. [Online]. Available: https://collectd.org/wiki/index.php/Main_Page
[78] *What is LabVIEW?—National Instruments*. Accessed: Sep. 21, 2019. [Online]. Available: https://www.ni.com/en-lb/shop/labview.html
[79] *User Manual TDS1000- and TDS2000-Series Digital Storage Oscilloscope*, Tektronix, Beaverton, OR, USA, 2009.
[80] *What Is Amazon EC2?—Amazon Elastic Compute Cloud*. Accessed: Dec. 4, 2019. [Online]. Available: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html
[81] *Carlucci G CPU Load Generator*. Accessed: Sep. 1, 2019. [Online]. Available: https://github.com/GaetanoCarlucci/CPULoadGenerator
[82] *Stress Project Page*. Accessed: Sep. 1, 2019. [Online]. Available: https://people.seas.harvard.edu/~apw/stress/
[83] H. Vandenbergh. *Vdbench Users Guide*. Accessed: Dec. 4, 2019. [Online]. Available: https://www.oracle.com/technetwork/server-storage/vdbench-1901683.pdf
[84] M. Mortimer. *iperf3 Documentation*. Accessed: Dec. 4, 2019. [Online]. Available: https://buildmedia.readthedocs.org/media/pdf/iperf3-python/latest/iperf3-python.pdf
[85] B. D. Wedlock and J. K. Roberge, *Electronic Components and Measurements*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1969.
[86] *R: What is R?* Accessed: Dec. 4, 2019. [Online]. Available: https://www.r-project.org/about.html
[87] *Kopytov A Sysbench*. Accessed: Aug. 14, 2018. [Online]. Available: https://github.com/akopytov/sysbench#sysbench
[88] *MPlayer—The Movie Player*. Accessed: Aug. 14, 2018. [Online]. Available: http://www.mplayerhq.hu/design7/dload.html
[89] *The PARSEC Benchmark Suite*. Accessed: Aug. 14, 2018. [Online]. Available: http://parsec.cs.princeton.edu/parsec3-doc.htm
[90] *Weka 3—Data Mining With Open Source Machine Learning Software in Java*. Accessed: Aug. 25, 2018. [Online]. Available: https://www.cs.waikato.ac.nz/ml/weka/downloading.html
[91] *H264 Video Format*. Accessed: Aug. 14, 2018. [Online]. Available: http://www.h264info.com/h264.htmlis
[92] *UCI Machine Learning Repository*. Accessed: Aug. 25, 2018. [Online]. Available: https://archive.ics.uci.edu/ml/index.php
[93] *AVI Format | AVI Player | AVI Codec*. Accessed: Dec. 4, 2019. [Online]. Available: https://www.divx.com/
[94] A. Beloglazov, "Energy-efficient management of virtual machines in data centers for cloud computing," Ph.D. dissertation, Univ. Melbourne, Melbourne, VIC, Australia, 2013.

LEILA ISMAIL received the DEA degree from the Joseph Fourier University (Grenoble I)/ENSIMAG Engineering School, France, and the Ph.D. degree (Hons.) from the Department of Computer Engineering/Distributed Systems, National Polytechnic Institute of Grenoble, France, in 2000, all in distributed systems.

She has vast industrial and academic experience with Sun Microsystems Research and Development involved in the design and implementation of highly available distributed systems and participated in the deposit of a US patent in the domain. She served as a Teacher with Grenoble I, France, and as an Assistant Professor with the American University of Beirut. She has been an Adjunct Professor with the Digital Ecosystems and Business Intelligence Institute, Curtin University, Australia. She has been an Associate Professor with the College of Information Technology (CIT), United Arab Emirates University (UAE), since 2005, where she is currently the Founder and the Head of the Distributed Computing and Distributed Systems Research Laboratory. Her current research interests include cloud computing, energy efficiency, green computing, resource management, and scheduling problems in distributed systems with an emphasis on clouds, middleware, HPC, and software security in distributed systems.

Dr. Ismail has international collaborations and is publishing the research results in prestigious journals and international conferences. She received the IBM Shared University Research (SURA) and the IBM Faculty Award, very competitive worldwide, funding for major projects as a PI/Co-PI and the funded project by UAE/NRF was top-ranked by external anonymous reviewers. She served as a Chair, Co-Chair, and Track Chair for many IEEE international conferences, including being a General Chair for IEEE DEST 2009 and a General Chair, Technical Program Chair, and Organizing Committee Chair for the 11th International Conference on Innovations in Information Technology 2015 (IIT'15) for which she received the support of the IEEE Computer Society (HQs) Technical Sponsorship. She served as an Associate Editor for the *International Journal of Parallel, Emergent and Distributed Systems* for several years. She is an Editor of the *Information Innovation Technology in Smart Cities*, (Nature Springer, 2018).

EYAD H. ABED received the S.B. degree from the Massachusetts Institute of Technology, in 1979, and the M.S. and Ph.D. degrees from the University of California at Berkeley, in 1981 and 1982, respectively, all in electrical engineering. From 2002 to 2008, he was the Director of the Institute for Systems Research, University of Maryland, College Park. From 2009 to 2012, he was the Dean of the College of Information Technology, United Arab Emirates University. From 2014 to 2017, he was a Program Director in the Energy, Power, Control and Networks Program with the Electrical, Communications and Cyber Systems Division, National Science Foundation. He has been on the Faculty of the Department of Electrical and Computer Engineering, University of Maryland, since 1983, where he is currently a Professor. His current research interests include system and control theory, especially, nonlinear dynamics and control, applications in electric power systems, communication networks, power electronics, aerospace systems, and social networks.