IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 22, 2019, accepted August 28, 2019, date of publication September 12, 2019,
date of current version September 30, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2941145

SLA-Aware Best Fit Decreasing Techniques for Workload Consolidation in Clouds

SAAD MUSTAFA[1], KINZA SATTAR[2], JUNAID SHUJA[1], SHAHZAD SARWAR[3], TAHIR MAQSOOD[1], SAJJAD A. MADANI[4], AND SGHAIER GUIZANI[5]

[1]Department of Computer Science, COMSATS University Islamabad, Abbottabad Campus, Abbottabad 22060, Pakistan
[2]Knowledge Unit of Science and Technology, University of Management and Technology, Sialkot Campus, Sialkot 51310, Pakistan
[3]Punjab University College of Information Technology, University of Punjab, Lahore 54000, Pakistan
[4]COMSATS University Islamabad, Wah Campus, Wah 47040, Pakistan
[5]College of Engineering, Alfaisal University, Riyadh 11533, Saudi Arabia

Corresponding author: Saad Mustafa (saadmustafa@cuiatd.edu.pk)

ABSTRACT Cloud computing emerged as one of the leading computational paradigms due to elastic resource provisioning and pay-as-you-go model. Large data centers are used by the service providers to host the various services. These data centers consume enormous energy, which leads to increase in operating costs and carbon footprints. Therefore, green cloud computing is a necessity, which not only reduces energy consumption, but also affects the environment positively. In order to reduce the energy consumption, workload consolidation approach is used that consolidates the tasks in minimum possible servers. However, workload consolidation may lead to service level agreement (SLA) violations due to non-availability of resources on the server. Therefore, workload consolidation techniques should consider the aforementioned problem. In this paper, we present two consolidation based energy-efficient techniques that reduce energy consumption along with resultant SLA violations. In addition to that, we also enhanced the existing Enhanced-Conscious Task Consolidation (ECTC) and Maximum Utilization (MaxUtil) techniques that attempt to reduce energy consumption and SLA violations. Experimental results show that the proposed techniques perform better than the selected heuristic based techniques in terms of energy, SLA, and migrations.

INDEX TERMS Energy efficiency, workload consolidation, SLA violation, resource management, cloud computing.

I. INTRODUCTION

Cloud computing provides on-demand elastic resources with the help of virtualization [1]–[3]. Virtualization enables cloud service providers to share physical resources of a single machine, such as CPU, RAM, and storage among multiple users. Virtualization increases the utilization of resources and helps in minimization of active resources through resource consolidation. Moreover, resource management techniques are used to acquire and manage virtualized resources. On the arrival of a new task, either a new virtual machine (VM) is created or the task is hosted on an existing VM. Furthermore, resource management techniques also assist in maximizing the performance of cloud systems with respect to various research challenges. Resource management techniques are incorporated to enhance load balancing, energy efficiency, network load management, profit maximization, and fault tolerance [4], [5].

One of the critical research challenges related to resource management and cloud computing is energy efficiency [4], [6]. Energy efficiency enables service providers to achieve/significant reduction in operational expenditure and carbon-dioxide (CO_2) emission [6]–[8]. According to recent studies, information and communication technology (ICT) industry consumed 271.8*109 kWh electricity in 2010 [9], [10], and it is expected to increase up to 1963*109 kWh by year 2020 [11]. Moreover, in the year 2007, ICT industry produced about 2% of the total CO_2 emissions, and it is expected to increase up to 12% by the year 2020 [12]–[15]. Similarly, data centers in the U.S. consumed 91*109 kWh electricity in 2013 that is expected to increase to 140*109 kWh by 2020, and carbon emission will reach 150 *106 metric tons [16]. On the other hand, usual server utilization remained at 12% to 18% during the time between 2006 and 2012 while almost 20% to 30% of servers in large data centers remained

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

unused [16]. According to [17], against 1W energy consumption at the processor level, almost 27W of overall data center energy is consumed. Therefore, in order to mitigate the detrimental effects of CO_2 emissions and minimization of the operational cost, researchers are exploring alternate techniques to enhance energy efficiency.

Over the past decade, researchers have presented various resource allocation techniques to optimize energy efficiency of cloud systems. Techniques, such as dynamic voltage and frequency scaling (DVFS), workload consolidation, and dynamic power management (DPM) are used to provide energy-efficient solutions [18]. DVFS scales voltage of the server according to the computational load on the central processing unit (CPU). Similarly, workload consolidation techniques aim to reduce energy consumption by aggregating workload on the minimum possible servers. Both techniques use DPM to manage available servers by dynamically switching them ON and OFF. Resource management techniques proposed based on the aforementioned methods are applicable to data centers that use homogeneous servers. However, in case of heterogeneous servers, the above-mentioned solutions fail to select the best server due to resources' variation. Heterogeneous servers may differ in terms of CPU, RAM, and storage. Therefore, researchers are recently focusing to provide solutions for the heterogeneous data centers [19]–[31].

The focus of this work is on workload consolidation techniques, for heterogeneous cloud data centers. Workload consolidation techniques increase resource utilization by considering the fact that for most of the time, the resources assigned to a VM are not fully utilized [32]. The service provider and user agree on a service level agreement (SLA) that ensures the quality of service (QoS). Usually, the level of QoS provided to the user is based upon the peak amount of resources required to run the task. However, existing energy-efficient techniques only consider the current usage of resources and do not consider the future resource demands of a VM. Therefore, in case of fluctuating workloads and aggressive consolidation, the chances of SLA violations increase due to non-availability of resources and extra load on a limited number of servers [33], [39]. Moreover, due to workload consolidated on fewer servers, the load on the network between those servers increases that may lead to network delays and SLA violations. Therefore, there is a need to develop solutions that not only focus on energy efficiency but also avoid possible chances of SLA violations.

To avoid SLA violations, various techniques are used by researchers. A simple method to avoid SLA violations is to allocate the amount of resources that are agreed at the agreement level [39]. However, such method leads to low server utilization and higher energy consumption. Another set of techniques use the prediction models to forecast the future needs of VMs [31], [44], [45]. Prediction based methods perform better than simple method as they reduce energy consumption while limiting SLA violations. However, prediction based methods may suffer when workloads are fluctuating and non-periodic. To handle the issues faced in non-periodic fluctuating workloads, researchers proposed the threshold mechanisms [39]. Threshold mechanisms keep a proportion of resources free on each server to handle the future demand of hosted VMs. Though the threshold mechanisms handle the issue of SLA violations, but it is at the cost of increase in energy consumption. Threshold mechanisms consume more energy because some of the resources are kept free on each server which leads to increase in number of active servers [46], [47]. In this work, we are using threshold mechanism as most of the time workloads are fluctuating and non-periodic.

It is noticed that energy efficiency can be further enhanced by introducing workload consolidation techniques that are more aggressive in consolidating the workloads on fewer numbers of physical resources. Moreover, there is also an intention to decrease SLA violations. Therefore, we propose workload consolidation techniques which intend to provide energy efficiency while ensuring the agreed QoS. Best fit decreasing (BFD) bin packing technique is used as a base for the proposed techniques [34], [35]. BFD is considered good for VM placement due to efficient bin packing by placing the VMs with higher resource requirements first. After placing the large VMs, VMs with low resource requirements can be easily accommodated on the remaining resources. Our first technique Minimum Power Best Fit Decreasing (MPBFD) selects a server whose peak power consumption is the lowest compared to other servers. Therefore, it will select those servers that have the lowest energy consumption. The second proposed technique Maximum Capacity Best Fit Decreasing (MCBFD) selects a server with maximum CPU capacity, such that the maximum VMs can be accommodated on a minimum number of servers. In addition, we use a threshold mechanism to avoid excessive SLA violations due to higher CPU utilization. Furthermore, we have enhanced two existing techniques, namely, Enhanced-Conscious Task Consolidation (ECTC) and Maximum Utilization (MaxUtil) [40], by further improving the energy efficiency and SLA-awareness. The major contributions of this paper can be summarized as follows:

- Two SLA-aware workload consolidation techniques MPBFD and MCBFD are proposed to handle the issue of energy consumption and SLA violations.
- A detailed quantitative analysis of ECTC and MaxUtil is presented using the same environment, assumptions, and system models.
- The selected existing techniques are enhanced to further improve the energy and SLA-awareness.
- A detail complexity analysis of all the techniques is also provided.

The remaining structure of the paper is as follows: Section 2 presents the state-of-the-art related work of the topic. In Section 3, system model used for the study is described and Section 4, presents the complete working of the techniques which are selected for the comparison purposes. Section 5 and 6, provide the detail discussion on the proposed techniques and performance evaluation, respectively. Section 7 presents the conclusions of the study.

II. RELATED WORK

Researchers face numerous research challenges such as energy efficiency, SLA violation, profit maximization, network and server load management, and fault tolerance in the field of cloud environment. Among the aforementioned challenges, energy efficiency is considered critical due to cost and carbon dioxide (CO_2) emission. In recent times, various hardware and software solutions are proposed that attempt to improve energy efficiency of cloud data centers [18]. Primarily, DVFS techniques are considered closer to hardware due to voltage and frequency scaling [19]–[21]. DVFS techniques scale voltage of the server based on the computational load on the central processing unit (CPU). In [19], Wu *et al.* propose a server level energy-efficient scheduling algorithm for the cloud data center. Further, energy efficiency is enhanced by increasing utilization of servers and consolidating workloads. Authors in [20] propose enhanced weighted round robin (EWRR) scheduling for resource management. The proposed technique monitors the resources used by the VMs and consolidates the VMs to achieve higher energy efficiency. Authors in [21] propose a game theoretic technique for VM placement in DVFS-enabled clouds that ensures energy efficiency, performance balancing and resource efficiency. Ali *et al.* [22] propose a DVFS based framework that attempts to provide energy efficiency while guaranteeing the agreed QoS. Moreover, a hybrid technique addresses the issues like load balancing, service placement, and resource allocation. The DVFS technique provide energy efficiency but resource utilization is not improved and there is still scope to improve energy efficiency.

On the other hand, workload consolidation techniques are considered as software-based solutions as they reduce energy consumption by placing incoming workloads on minimum possible servers [23]–[28]. This technique maximizes the resource utilization by benefiting from the fact that the allocated resources to the VM remain idle for most of the time. However, if the VM's resource demand increases and enough resources are unavailable, then it would lead to service level agreements (SLA) violations [33]. A novel energy management technique named as soft-resource scaling is proposed in [24], for the large-scale virtualized environments. Moreover, resources are managed at two levels by local and global managers. A local manager deals with the energy consumption of a VM residing on a single server, whereas, a global manager deals with multiple VMs residing on various servers. In [25], the authors propose several techniques for coordinated energy management at the server and group levels. The allocation of VM is addressed as an integer programming problem to reduce energy consumption and performance loss. Notably, the authors do not consider the SLA violations under fluctuating workloads in the above mentioned techniques.

Kusic *et al.* propose a resource management technique for maximizing the service provider's profit by reducing energy consumption and SLA violations, via look ahead control [27]. This control is based upon sequential optimization model and Kalman's filter, which predicts future resource demands of the already hosted VMs. A major drawback of the technique is higher complexity and computation time. This clearly indicates, simple techniques should be used such as dynamic resource allocation that is less computation intensive. In [28], the authors propose a workload consolidation technique that dynamically allocates resources to each VM on fluctuating user demand. A prediction mechanism is used to predict the future needs of the VMs and subsequently to manage the resources accordingly. Moreover, static threshold mechanisms are used to identify over-utilized and under-utilized resources. Static thresholds are not suitable for dynamically fluctuating workloads. Therefore, mechanisms should be devised that should be able to handle ever-changing behaviors of workloads.

In [29], Addis *et al.* present a resource management technique that is based on the hierarchal framework presented in [30]. The proposed technique uses a nonlinear optimization approach to reduce energy and response time. Moreover, resource management activities are divided between central and application managers. Central manager is responsible of class and server partitioning, whereas, application manager deals with activities such as load balancing, frequency scaling and capacity. However, SLA violations are not considered by the authors. In [40], the authors propose two consolidation techniques to reduce energy consumption. The first technique uses a cost function that subtracts the minimum energy required to host a VM if other VMs are also hosted on the server, from the actual energy required to host the VM. Whereas, the second technique checks the utilization of each server and places the VM on the server that maximum utilization. Both the techniques reduce the energy consumption without considering the resultant SLA violations. In [31], the authors present a technique that dynamically allocates resources to running applications based on workload prediction. A controller forecasts the workload and manages resources on an hourly basis to reduce the system overhead.

Horri *et al.* propose a workload consolidation algorithm that provides energy efficiency and SLA-awareness [26]. The VM placement is done based on correlation between resources utilization and VMs. Authors in [23] propose a technique that uses the information related to topology and racks to place the VMs. In order to achieve workload consolidation, incoming tasks are accommodated on minimum servers and racks. To conserve the energy, idle routers and respective cooling units are turned OFF. Moreover, service level agreement (SLA) violations are avoided as all the servers are turned ON in all the used racks. In [33], authors use the algorithm proposed in [39] to handle the issue of energy consumption along with SLA violations. The consolidation algorithm is used to reduce the energy consumption, whereas, dynamic thresholds are used to avoid SL violations.

In the literature, many studies have focused on minimizing either energy consumption or SLA violations. However, there are few studies that have considered both parameters [23], [26], [31], [33]. The techniques proposed by the authors of [26] and [31] handle the issue of SLA violations, but suffers in case of fluctuating and non-periodic workloads. Whereas, the solution proposed in [23] keeps all the servers of the rack active which results in increased energy consumption. However, the technique used in [33] and [39] handles both the issues efficiently. Therefore, we will compare the proposed algorithms with the SLA-aware energy efficient solution proposed in the aforementioned work. In addition to that, the techniques proposed in [40] are selected for comparison and enhancement due to their aggressive workload consolidation and energy efficiency. Moreover, to handle the issue of energy efficiency and SLA violations, we also propose two SLA-aware energy-efficient techniques.

III. SYSTEM MODEL

The system model considered throughout this work is the one proposed by Anton and Buyya [33] which is based on a large data center with heterogeneous servers that can host multiple applications. Based on the resource requirements of running applications, heterogeneous VMs are created and placed on available servers. Heterogeneous VMs and fluctuating resource demands of applications result in dynamic workloads. Moreover, servers and VMs hosting the workloads are characterized by computing power specified in million instructions per second (MIPS), storage, memory, and bandwidth. Further, details of the system model are illustrated in Figure 1, there is a central manager named a "global manager" that manages the overall resources of the whole data center. Next, there is a local VM manager at each server that performs VM resizing, migration, and change of power mode. VM managers are also responsible for sharing local resource information with the global manager. Furthermore, live migrations are done using network attached storage (NAS). NAS provides the central storage which keeps the duplicated information of VM and facilitates the migration process [36], [48]. New server gets the current state of the VM from the NAS instead of old hosting server of VM [37], [38].

According to a recent study [39], energy consumption and CPU utilization has a linear relationship, and a major proportion of the energy is consumed by the CPU. Therefore, we have used the power model proposed by Anton *et al.* [33] which is based on the utilization of CPU and is defined by equation 1.

$$P(u) = k * P_{max} + (1 - \mathrm{k}) * P_{max} * u \tag{1}$$

where P_{max} is the maximum power, k is expressed in fraction (i.e., 70%) and is the power consumed by the idle server, and the CPU utilization is represented by u. However, the CPU utilization may change due to varying workloads. Therefore, we take the integral of the consumed power by the server between times t_o and t_1, to calculate the total energy consumption of the server, as shown in the equation 2:

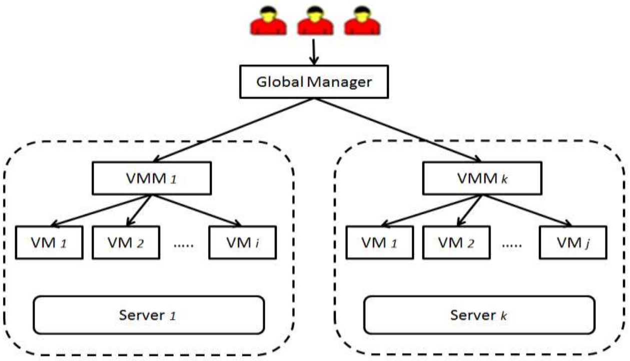


FIGURE 1. The system model.

$$E = \int_{t_0}^{t_1} P(u(t))\, dt \tag{2}$$

IV. COMPARISON TECHNIQUES

In this work, we have analyzed some of the selected energy-efficient algorithms for VM placement in clouds. The analysis of algorithms is done on different metrics that include: (a) energy consumption, (b) Average SLA violation, and (c) SLA violations due to virtual machine migrations. In addition, we propose extensions of ECTC and MaxUtil techniques to improve their performance in terms of SLA violations. Following are the techniques selected for the comparison purposes:

- SLA-aware Best-Fit Decreasing Algorithm (SBFD)
- Modified Best-Fit Decreasing (MBFD)
- Enhanced-Conscious Task Consolidation (ECTC)
- Maximum Utilization (MaxUtil)

A. SLA-AWARE BEST-FIT DECREASING ALGORITHM (SBFD)

The traditional BFD algorithm performs sorting of incoming tasks based on their CPU requirements [34]. Once the sorting is complete, the task placement process starts. A task is selected from the top of the list and placed on the server that is already in use and has the maximum CPU capacity compared to other servers. In such a case, there is no used server that can host the task, then the server with minimum CPU capacity is selected. The traditional BFD algorithm does not consider the SLA violations. However, authors in [35] proposed an SLA-aware version of the algorithm that uses an upper threshold to keep some of the resources free. The free CPU capacity is used to handle the future resource demands of the tasks.

B. MODIFIED BEST-FIT DECREASING (MBFD)

Authors in [39] propose Best-Fit decreasing based workload consolidation technique to achieve energy efficiency. The Modified Best-Fit decreasing (MBFD) algorithm sorts the incoming tasks similar to the BFD algorithm. Tasks from the top of the list are placed on the server that shows a minimum change in energy consumption after hosting the VM, as calculated by equation 3. The energy consumption of the server at any instance can be calculated by using equation 1.

$$P_{diff} = P_{AP} - P_{BP} \tag{3}$$

where P_{BP} is the power consumption before VM placement, and P_{AP} is the power after VM placement.

C. ENHANCED-CONSCIOUS TASK CONSOLIDATION (ECTC)

In ECTC, every resource is checked, and the most energy-efficient resource is picked [40]. The ECTC uses a cost function that subtracts the energy consumption of the VM when it is placed on an already used server, from the energy consumption of the same VM when it is running alone. The value of the cost function $f_{i,j}$ of a virtual machine (VM_i) when placed on a resource r_j is computed using equation 4.

$$f_{i,j} = \left(\left(P\Delta * U_j + P_{min}\right) * \tau_0\right) - \left(\left(P\Delta * U_j + P_{min}\right) * \tau_1 + P\Delta * U_j * \tau_2 \right. \tag{4}$$

where $P\Delta$ is the difference between P_{max} and P_{min}, P_{max} is the power when server is fully utilized, and P_{min} is the power consumed by the server when it is idle. U_j is the utilization rate of the VM_i, and τ_0, τ_1 and τ_2 are the processing time when VM_i is executed alone, along with one VM, and with multiple VMs, respectively.

D. MAXIMUM UTILIZATION (MAXUTIL)

According to Lee and Zomaya [40], MaxUtil and ECTC follow the same steps for picking an energy-efficient server. The only difference is their cost function. MaxUtil uses a cost function that selects a server based on an average utilization. The aim is to maximize the server utilization and workload consolidation. This aids in the reduction of energy as it reduces the number of active servers. The cost function $f_{i,j}$ of MaxUtil when a virtual machine VM_i is placed on a resource r_j can be defined by equation 5.

$$f_{i,j} = \frac{\sum_{\tau=1}^{\tau_0} Ui}{\tau_0} \tag{5}$$

where Ui is the resource utilization of the server under consideration during the processing time of VM_i represented by τ_0.

E. ENHANCED TECHNIQUES

Enhanced ECTC and MaxUtil techniques are proposed that use thresholds to identify underutilized and overutilized servers. A lower threshold identifies a server if the resources in use are below the given utilization threshold. In such a case, VMs hosted on the server are transferred to the other server(s), and the offloaded server is switched to the power saving mode. On the other hand, the upper threshold is used to keep the check on the server's utilization so that it should not be overutilized, and SLA violations could be avoided. In case of an SLA violation, a VM or set of VMs is transferred to servers that can easily accommodate incoming VM(s) without violating the threshold. Moreover, an efficient migration technique is used to reduce the network's performance degradation. Our selected technique picks a VM that needs the least time for transfer. By selecting such a VM, the network load incurred due to migrations is reduced. Details of used threshold mechanism and migration technique are discussed below.

Algorithm 1 Pseudo-Code for Proposed MPBFD Algorithm

Minimum Energy (MPBFD) Algorithm

```
Input: $sList(S), vList(V), threshold$;
Output: VM allocation
$V_s' \leftarrow$ sort all $v \in V$ in descending order based on CPU requirement
$S_i' \leftarrow$ sort all $s \in S$ in ascending order based on peak power consumption
for all $v_s \in V_s'$ do
    $Minimum\_power \leftarrow$ maximum value
    $Hosting\_server \leftarrow$ Null
    $res\_req_s \leftarrow$ computational resources required by VM $v_s$
    for all $p_i \in S_i'$ where utilized resources of $p_i + res\_req_s < threshold$ do
        $Peak\_Power_i \leftarrow$ Maximum power $p_i$ can consume
        $Overall\_CPU\_cap_i \leftarrow$ Overall CPU capacity of $p_i$
        $CPU\_avail\_cap_i \leftarrow$ CPU's available capacity of $p_i$
        if $CPU\_avail\_cap_i = Overall\_CPU\_cap_i$ and $Hosting\_server$ != Null
            continue
        end if
        if $Peak\_Power_i > Minimum\_power$
            $Hosting\_server \leftarrow p_i$
            $Minimum\_power \leftarrow Peak\_Power_i$
        end if
    end for
    if $Hosting\_server \neq$ Null
        place $v_s$ on $Hosting\_server$
    end if
end for
$Allocation$ = get_allocation( )
return $Allocation$
```

1) DYNAMIC THRESHOLD MECHANISM

Dynamic threshold mechanisms are based on the previous threshold values. A certain function is applied to the set of previous threshold values to get new threshold values. We used a median absolute deviation (MAD) mechanism to calculate the new threshold values. MAD is selected due to its robustness and resilience against outliers [33]. Equation 6 is used to calculate the dataset shown by $X_1, X_2, \ldots, X_n$.

$$MAD = median_i(\left|X_i - median_j\left(X_j\right)\right|) \tag{6}$$

where X_i is the univariate data. After calculating MAD, equation 7 is used to calculate the threshold (T_u) value.

$$T_u = 1 - s.MAD \tag{7}$$

where $s\epsilon\mathbb{R}^+$ is a safety parameter. The higher the value of s, the higher will be the SLA violations and vice versa.

2) MIGRATION POLICY

To handle improve the SLA at the network level, a minimum migration time (MMT) policy is used [33]. MMT selects a VM that will take the least time for migration. For example, the MMT will select a VM v if it satisfies equation 8.

$$v\epsilon V_j|\forall a\epsilon V_j, \quad \frac{RAM_u\,(v)}{BW_j} \leq \frac{RAM_u\,(a)}{BW_j} \tag{8}$$

where, V_j is the VMs placed on server j, $RAM_u\,(v)$ represents the RAM utilized by VM v, and BW_j is the unutilized bandwidth of server j.

V. PROPOSED TECHNIQUES

Keeping in view the shortcomings of ECTC and MaxUtil, we propose two new techniques: Maximum Capacity Best-FIT Decreasing (MCBFD) and Minimum Power Best-Fit Decreasing (MPBFD). To achieve a better energy efficiency, the proposed techniques use workload consolidation and lower threshold. Moreover, the upper threshold mechanism and MMT migration policy are used to reduce the SLA violations at the server and network levels, respectively. The MCBFD and MPBFD work in a similar manner; however, the basic difference between the two algorithms is of the server selection criteria.

A. MINIMUM POWER BFD (MPBFD)

Both the proposed techniques work like the existing MBFD technique. However, there are two server selection criteria are used by both of them. First, they use the server selection criteria of MBFD algorithm and along with that, they use their own criteria. The MPBFD sorts the VMs and selects the VM from the top of the list. The selected VM is placed on the server that shows the minimum change in power consumption and consumes minimum peak power. The complete working of MPBFD technique is demonstrated in Figure 2. Let's assume that the minimum power a server can consume is zero and maximum power is shown in the Figure 2. MPBFD selects *server 3* for the hosting of *V2* because it shows minimum change in power consumption after the hosting of VM and also has minimum peak power consumption compared to the other two servers. Moreover, after that *server 2* is selected because it has the second-lowest power consumption. After the complete VM placement, *server 1* is powered off.

The pseudo code of the MPBFD is shown in Algorithm 1. Initially, lists of VMs and servers are sorted based on the CPU requirements and peak power consumption, respectively. After sorting, VMs are selected from the top of list and placed on the servers one by one. It is preferred that the hosting server is already being utilized and can host the VM. The server is selected on the basis of peak power that a server can consume. Server with least power consumption is selected and VM is placed on it.

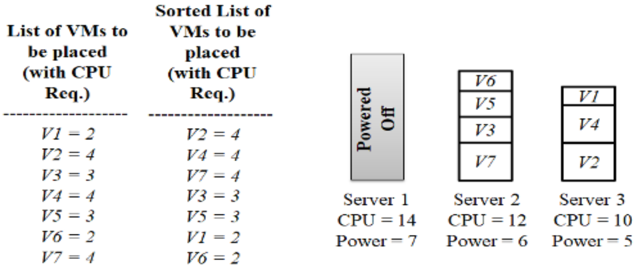


FIGURE 2. Working of MPBFD.

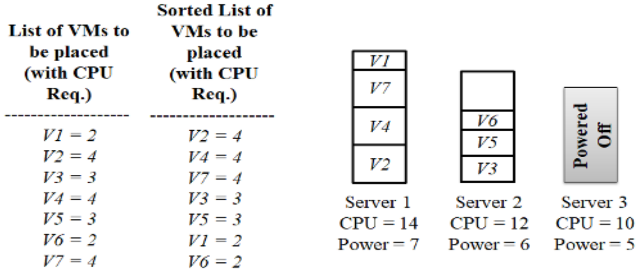


FIGURE 3. Working of MCBFD.

B. MAXIMUM CPU CAPACITY BFD (MCBFD)

MCBFD is a CPU capacity based technique that selects a server that has the maximum CPU capacity and shows the minimum change in power consumption. By choosing servers with high CPU capacity, more VMs can be accommodated on fewer servers. It reduces energy consumption and expenditures incurred on it. The complete working of the MCBFD is demonstrated in Figure 3. Consider the example, in Figure 3, *V2* is selected from the top of list and placed on *server 1* because *server 1* has the maximum CPU capacity. After *V2*, *V4* is selected from the list of VMs and placed on *server 1* again. This process will continue until *server 1* does not have a CPU capacity to satisfy the demands of next VM. Afterwards, the next server with highest CPU capacity is selected, for instance, *server 2* in the above example, and the same process is repeated for all the VMs.

The pseudo code of the MCBFD is shown in Algorithm 2. Step by step working of the MCBFD is similar to the MPBFD except in the server selection criteria part. Instead of using the minimum power consumption, the MCBFD selects a server based on the CPU capacity.

C. COMPLEXITY ANALYSIS OF PROPOSED MPBFD AND MCBFD ALGORITHMS

In this section, the complexity analysis of the selected and the proposed techniques is presented.

1) TIME COMPLEXITY

Time complexity of the proposed MPBFD and MCBFD algorithms is discussed in this section. In the first step, both algorithms sort the list of x VMs in descending order with the complexity of $O(x.log(x))$. Second, the server sorting is done

Algorithm 2 Pseudo-Code for Proposed MCBFD Algorithm

Maximum Capacity (MCBFD) Algorithm

```
Input: $sList(S), vList(V), threshold$;
Output: VM allocation
$V'_s \leftarrow$ sort all $v \in V$ in descending order based on CPU requirement
$S'_i \leftarrow$ sort all $s \in S$ in descending order based on CPU capacity
for all $v_s \in V'_s$ do
    $Maximum\_cap \leftarrow 0$
    $Hosting\_server \leftarrow$ Null
    $res\_req_s \leftarrow$ computational resources required by VM $v_s$
    for all $p_i \in S'_i$ where utilized resources of $p_i + res\_req_s < threshold$ do
        $Overall\_CPU\_cap_i \leftarrow$ Overall CPU capacity of $p_i$
        $CPU\_avail\_cap_i \leftarrow$ CPU's available capacity of $p_i$
        if $CPU\_avail\_cap_i = Overall\_CPU\_cap_i$ and $Hosting\_server$ != Null

            continue
        end if
        if $Overall\_CPU\_cap_i > Maximum\_cap$
            $Hosting\_server \leftarrow p_i$
            $Maximum\_cap \leftarrow Overall\_CPU\_cap_i$
        end if
    end for
    if $Hosting\_server \neq$ Null
        place $v_s$ on $Hosting\_server$
    end if
end for
$Allocation =$ get_allocation( )
return $Allocation$
```

based on a set of criteria. The time complexity of sorting y servers is $O(y.log(y))$. Moreover, the outer loop will run x times to place VMs on servers, and the inner loop will be executed y times in a worst-case scenario and z (number of used servers) times in the best-case scenario. In the worst-case scenario, a used server will not be available for VM hosting, and a new server will be selected from unused servers. Therefore, the best-case and worst-case time complexities can be calculated with the help of equation 9 and equation 10, respectively

$$O(\left[x\,(\log(x) + z)\right] + y \times \log(y)) \tag{9}$$

$$O(\left[x\,(\log(x) + y)\right] + y \times \log(y)) \tag{10}$$

2) SPACE COMPLEXITY

In the following equation, we present the space complexity of the MCBFD and MPBFD algorithms.

$$O(2x + y) \tag{11}$$

TABLE 1. Time and Space Complexities of selected and proposed techniques.

Resource Management Technique	Time Complexity		Space Complexity
	Best Case Scenario	Worst Case Scenario	
MCBFD	$O([x(log(x)+z)]+y\times log(y))$	$O([x(log(x)+y)]+y\times log(y))$	$O(2x+y)$
MPBFD	$O([x(log(x)+z)]+y\times log(y))$	$O([x(log(x)+y)]+y\times log(y))$	$O(2x+y)$
MBFD	$O([x(log(x)+z)]+y\times log(y))$	$O([x(log(x)+y)]+y\times log(y))$	$O(2x+y)$
MaxUtil	$O([x(log(x)+z)]+y\times log(y))$	$O([x(log(x)+y)]+y\times log(y))$	$O(2x+y)$
ECTC	$O([x(log(x)+z)]+y\times log(y))$	$O([x(log(x)+y)]+y\times log(y))$	$O(2x+y)$

where x represents the number of VMs, and y shows the number of servers. The space complexity of both algorithm is quite modest as the space complexity of the MCBFD and MPBFD algorithms is very modest because two lists of sizes x and y are used to serve VMs and servers, respectively. Moreover, to save the VM placement on a server, a list of size x is used that stores the server id against each VM. Table 1 shows the complexity analysis of all the techniques.

VI. EXPERIMENTAL EVALUATION

This section provides a detailed discussion of the performance evaluation environments, parameters and achieved results.

A. EXPERIMENTAL SETUP

For the performance analysis, we used a simulation tool known as CloudSim [41]. It is a java based open-source tool that provides a cloud data center environment to implement and evaluate energy-efficient techniques. Moreover, various parameters are provided to evaluate the techniques, such as, energy consumption, SLA violations, host shutdowns, and performance degradation due to migrations. The aforesaid parameters are used to check the performance of resource management techniques.

For evaluation purposes, 800 heterogeneous servers with configuration shown in Table 2 and VMs instances similar to Amazon EC2 [42] are used. We have also used real-world workloads of 10 days provided by PlanetLab [43]. The data presented in the aforesaid workloads is related to the CPU utilization of 500 servers when various number of VMs hosted on those servers.

B. PERFORMANCE EVALUATION PARAMETERS

The proposed and selected techniques are evaluated based on the following parameters.

1) ENERGY CONSUMPTION

Energy consumption of a data center is the amount of energy consumed by the active servers for a time under

TABLE 2. Server configurations.

Server	CPU Model	No. of Cores	Clock Rate (MHz)	RAM (GB)	Min Power (W)	Max Power (W)
IBM Server x3550	2 x Intel Xeon 5675	2 x 6	3067	16	58.4	222
IBM Server x3250	Intel Xeon 3470	4	2933	8	41.6	113
HP ProLiant ML110G5	Intel Xeon 3075	2	2660	4	93.7	135
HP ProLiant ML110 G4	Intel Xeon 3040	2	1860	4	86	117

consideration. The energy model used for this study is already discussed in Section 3.

2) PERFORMANCE DEGRADATION DUE TO MIGRATIONS (PDM)

In this study, PDM is used to check the impact of migrations that are encountered due to SLA violations. This can help reduce migrations and performance degradation by selecting a better migration technique. Equation 12 is used to calculate the value of the PDM.

$$PDM = \frac{1}{N}\sum_{i=1}^{N}\frac{C_{d_i}}{C_{r_i}} \tag{12}$$

where N are the total VMs, C_{d_i} is the expected performance degradation of a certain VM i and C_{r_i} is the requested capacity by that VM.

3) SLA VIOLATION (SLAV)

The proposed techniques are further evaluated on the basis of SLA violations. *SLAV* is a metric proposed by authors in [20] to keep the check on the SLA violations that are encountered due to non-availability of resources. Equation 13 is used to calculate the *SLAV*.

$$SLAV = SLATAH * PDM \tag{13}$$

where, *SLATAH* is the total violation time of the active server, and *PDM* is already discussed in the previous section. The *PDM* and *SLATAH* can be calculated using equations 12 and 14, respectively.

$$SLATAH = \frac{1}{N}\sum_{i=1}^{N}\frac{T_{s_i}}{T_{a_i}} \tag{14}$$

where N are the total servers, T_{s_i} represents the total time when the server i remains fully utilized and T_{a_i} is the total active time of the server.

C. RESULTS

All the techniques discussed in this study attempt to reduce energy consumption. However, the ECTC and MaxUtil do not take SLA violations into account. Consequently, we modified these techniques to introduce the SLA-awareness and further improve the energy consumption. Therefore, in the first subsection, we discuss the performance of different variants of these techniques based on the above-mentioned performance metrics. In the second subsection, we compare our proposed techniques, the MCBFD and MPBFD with the selected techniques.

TABLE 3. Details of ECTC and MaxUtil variants.

Technique Name	VM Allocation Technique	Migration Technique	Lower Threshold	Upper Threshold
ECTC	ECTC	RS	No	No
MaxUtil	MaxUtil	RS	No	No
BFD	BFD	RS	No	No
ECTC/ MMT	ECTC	MMT	No	No
MaxUtil/ MMT	MaxUtil	MMT	No	No
BFD/MMT	BFD	MMT	No	No
EECTC	ECTC	MMT	Yes	No
EMaxUtil	MaxUtil	MMT	Yes	No
EBFD	BFD	MMT	Yes	No
SEECTC	ECTC	MMT	Yes	Yes
SEMaxUtil	MaxUtil	MMT	Yes	Yes
SEBFD	BFD	MMT	Yes	Yes

D. PERFORMANCE EVALUATION OF ECTC AND MAXUTIL VARIANTS

In this section, the original ECTC, MaxUtil and BFD are compared with their enhanced versions. The ECTC considers energy consumption during the server selection process, whereas, the selection criteria of the MaxUtil and BFD use the CPU utilization information. Table 3 shows the details of each technique. ECTC, MaxUtil and BFD represent the original techniques that use random select (RS) migration policy. The RS migration policy randomly selects VMs from the overloaded servers for migration. The ECTC/MMT, MaxUtil/MMT and BFD/MMT use the MMT migration policy for the selection of VMs. Whereas, EECTC, EMaxUtil and EBFD use lower threshold mechanism along with the MMT to further improve the energy consumption. Moreover, SEECTC, SEMaxUtil and SEBFD are SLA-aware versions that use the upper threshold mechanism to reduce SLA violations. The comparison of these variants along with original techniques is presented in Figures 4, 5, and 6.

1) ENERGY CONSUMPTION

Figure 4 shows the energy consumption of the different variants of ECTC, MaxUtil and BFD techniques. Results show that the enhanced variants perform better than the existing ones in terms of energy consumption. The reason behind the better energy consumption is the aggressive consolidation and efficient resource utilization. The aggressive workload consolidation results in less active servers and low energy

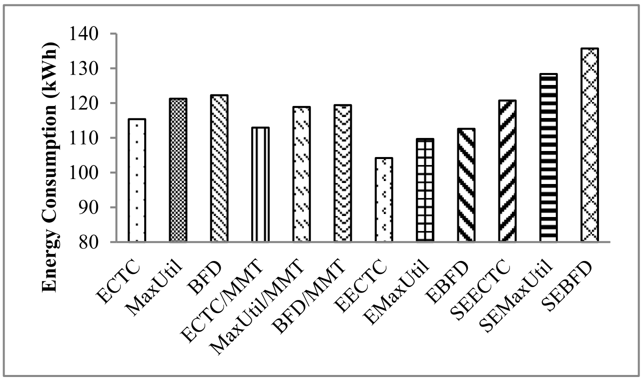


FIGURE 4. Energy consumption of ECTC, MaxUtil and BFD variants.

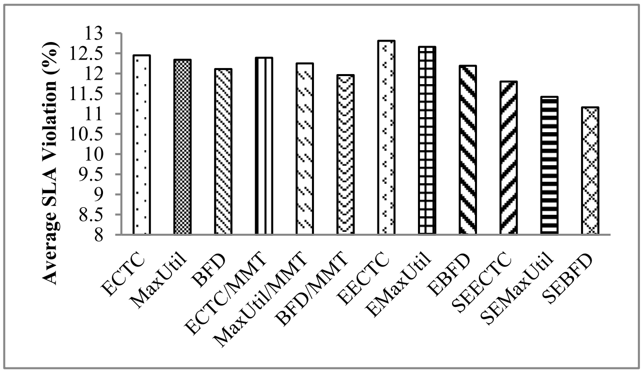


FIGURE 5. Average SLA violations of ECTC, MaxUtil and BFD variants.

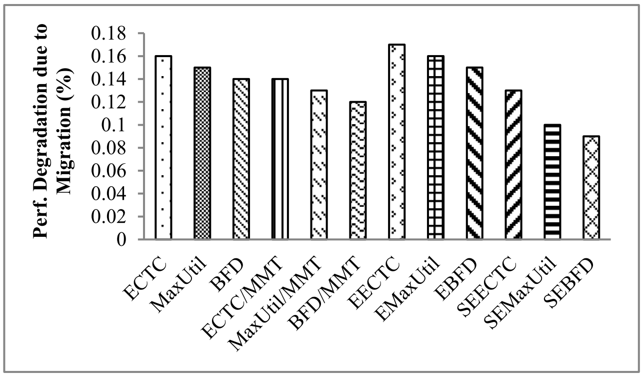


FIGURE 6. Performance degradation due to migrations of BFD, ECTC and MaxUtil variants.

consumption. Therefore, results show that the EECTC is the most energy-efficient technique, whereas, the SEBFD is the least energy-efficient resource management technique. The EECTC consumes 10% less energy compared to the original ECTC, and 8% less energy compared to ECTC/MMT. Moreover, the energy consumption of EECTC is 14% less than SLA-aware counterpart SEECTC. If most energy-efficient and least energy-efficient techniques are compared, then it can be noticed that EECTC uses 23% less energy compared to SEBFD. Whereas, if the best two techniques are compared, then the EECTC consumes 5% less energy than the EMaxUtil and 7.5% less compared to EBFD.

TABLE 4. Details of selected and proposed techniques.

Technique Name	VM Allocation Technique	Migration Technique	Lower Threshold	Upper Threshold
EECTC	ECTC	MMT	Yes	No
EMaxUtil	MaxUtil	MMT	Yes	No
MCBFD	MCBFD	MMT	Yes	No
MPBFD	MPBFD	MMT	Yes	No
SEECTC	ECTC	MMT	Yes	Yes
SEMaxUtil	MaxUtil	MMT	Yes	Yes
SMCBFD	MCBFD	MMT	Yes	Yes
SMPBFD	MPBFD	MMT	Yes	Yes
MBFD	MBFD	MMT	Yes	Yes

2) AVERAGE SLA VIOLATIONS

Figure 5 shows that the proposed SLA-aware extensions of ECTC, MaxUtil and BFD outperform other variants in terms of SLA violations. As the energy efficient versions of the techniques perform aggressive workload consolidation, which leads to better resource utilization. However, in case of increase in resource demand of a VM, hosting server does not have enough resources, which leads to non-availability of resources and SLA violations. To handle this issue, SLA versions use the upper threshold. If the resource demand of any VM changes, free resources are assigned to it and SLA violation is avoided. Therefore, it can be seen that the SLA-aware version of BFD (SEBFD) performs best and EECTC performs worst. The SEBFD has 13% fewer SLA violations compared to EECTC, 12% less than EMaxUtil and 8.5% fewer violations than the energy efficient variant, i.e., EBFD. Moreover, SEBFD has 8% less SLA violations than the original BFD and 6.5% fewer violations compared to BFD/MMT. If the best techniques in terms of SLA violations are compared, then the SEBFD has 2.5% and 4.5% less violations than the SEMaxUtil and SEECTC, respectively.

3) PERFORMANCE DEGRADATION DUE TO MIGRATION

Figure 6 shows that the SEBFD has 47% less performance degradation compared to EECTC, which shows the highest performance degradation. On the comparison of the SLA-aware BFD (SEBFD) with its most energy-efficient variants EBFD, it shows 40% less performance degradation. Moreover, performance degradation of the SLA-aware BFD is 36% less compared to the original BFD. Furthermore, the comparison of the best two techniques shows that the SEMaxUtil has 10% higher performance degradation compared to the SEBFD.

E. pERFORMANCE eVALUATION OF mcbfd AND mpbfd

This section presents the performance analysis conducted among the proposed and elected techniques. Details of the techniques discussed in this section are shown in Table 4. The energy efficient and SLA-aware versions of the ECTC and MaxUtil are selected for the comparison with variants of the proposed MCBFD and MPBFD techniques. Moreover, well

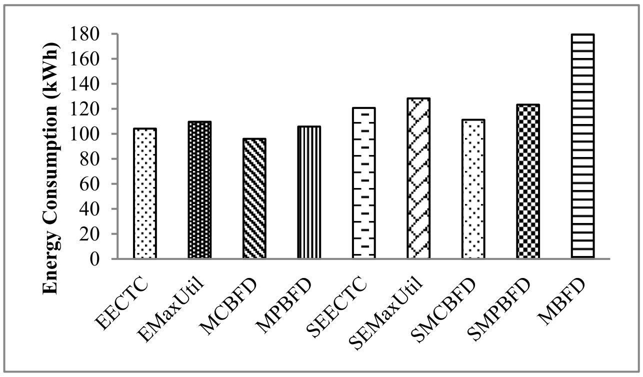


FIGURE 7. Energy consumption of proposed MPBFD and MCBFD algorithms along with selected techniques.

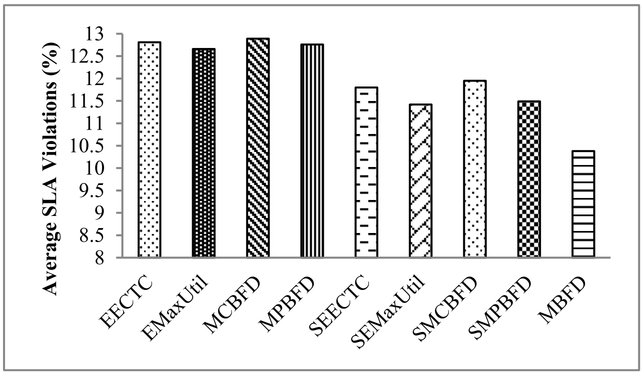


FIGURE 8. Average SLA violations of proposed MPBFD and MCBFD algorithms along with selected techniques.

known SLA-aware energy efficient BFD algorithm (MBFD) is also included for comparison.

1) ENERGY CONSUMPTION

In Figure 7, the energy consumed by the techniques under consideration is shown. The proposed MCBFD and MPBFD techniques perform better than other techniques. The reason behind the superior performance of the proposed techniques is the improved selection criteria of servers that help achieve better workload consolidation. The MCBFD slightly outperforms the MPBFD because it selects a server with higher CPU capacities that enables the MCBFD to consolidate more workloads on fewer number of servers. However, the MPBFD outperforms the MaxUtil and MBFD because it selects the servers that consume the minimum power. Whereas, the MaxUtil selects a server with higher CPU utilization, and the MBFD places the VM on a server that depicts a minimum hike in the energy consumption after the VM placement. Both techniques do not consider the overall CPU capacity and energy consumption of the servers. Consequently, the energy-aware MCBFD (MCBFD) consumes 46% lower energy compared to the MBFD algorithm. Whereas, the MCBFD consumes 14% less energy in comparison to SMCBFD. Moreover, if we compare the energy-aware MCBFD with the energy-aware ECTC, which is second-best, then it is observed that MCBFD consumes 8% less energy compared to ECTC.

2) AVERAGE SLA VIOLATIONS

Figure 8 highlights the fact that the SMCBFD and SMPBFD beat energy-efficient techniques in terms of the average SLA violations. The results show that the MBFD algorithm outperforms the rest of the algorithms in terms of the average SLA violations. The MBFD performs well compared to other techniques because of non-aggressive workload consolidation and use of thresholds. This helps the algorithm to keep enough free resources on each server and accommodate the fluctuating workload. The MBFD has 9% fewer violations compared to the next-best SLA-aware MaxUtil (SEMaxUtil) and 10% less violations, compared to third-best SLA-aware MPBFD (SMPBFD). When the MBFD is compared to the worst technique, it exhibited 19% fewer violations compared to the MCBFD.

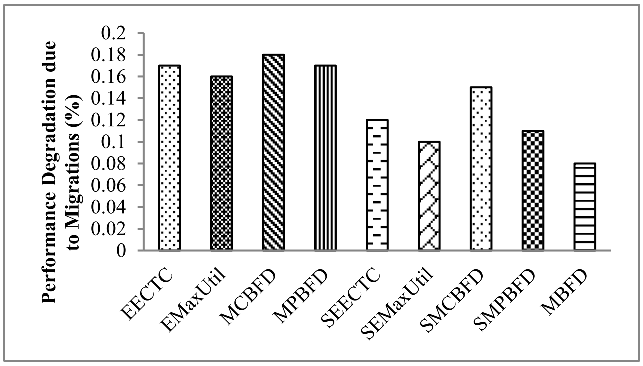


FIGURE 9. Performance degradation due to migrations of proposed MPBFD and MCBFD algorithms along with selected techniques.

3) PERFORMANCE DEGRADATION DUE TO MIGRATION

In Figure 9, the performance degradation of the proposed and selected techniques is shown. The MBFD is the best technique with respect to SLA violations and performance degradation. The MBFD has 20% lower performance degradation in comparison to the SMaxUtil, which is 2nd best. Whereas, the MBFD has 27% less degradation in comparison to 3rd best SLA-aware MPBFD (SMPBFD). Moreover, the MBFD has 55% lower performance degradation compared to the energy-aware MCBFD, which is worst in terms of performance metric under consideration.

VII. CONCLUSION

In this work, we conducted a comprehensive analysis of the selected resource allocation techniques for cloud environments. Major findings of this paper are that our proposed techniques, MCBFD and MPBFD, outperform their previous counterparts with respect to energy consumption, SLA violations, and performance degradation. The proposed algorithms achieve the aforesaid goals by selecting the best servers with regards to energy consumption and CPU capacity. Moreover, the use of lower threshold identifies the underutilized server that leads to decrease in energy consumption, whereas, the upper threshold reduces the SLA violations by keeping some of the resources free to accommodate the ever-changing demands of VMs. Furthermore, the minimum migration time policy along with an upper threshold reduces the performance degradation due to migration by choosing a VM that requires a least migration time.

The work presented in this study can be further improved by introducing more research challenges, such as network load, load balancing, fault tolerance, and profit maximization. The network load consideration can further decrease the network load and performance degradation issues. Moreover, considering fault tolerance while performing the energy efficiency can be an interesting research dimension.

REFERENCES

[1] B. P. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," in *Proc. 5th Int. Joint Conf. INC, IMS IDC*, Aug. 2009, pp. 44–51.

[2] P. Mell and T. Grance, "The NIST definition of cloud computing (Draft)," NIST, Gaithersburg, MD, USA, Tech. Rep. 800-145, 2011, vol. 800, p. 145.

[3] V. Medina and J. García, "A survey of migration mechanisms of virtual machines," *ACM Comput. Surv.*, vol. 46, no. 3, p. 30, Jan. 2014.

[4] S. Mustafa, B. Nazir, A. Hayat, A. R. Khan, and S. A. Madani, "Resource management in cloud computing: Taxonomy, prospects, and challenges," *Comput. Electr. Eng.*, vol. 47, pp. 186–203, Oct. 2015.

[5] S. H. H. Madni, M. S. A. Latiff, Y. Coulibaly, and S. M. Abdulhamid, "Resource scheduling for infrastructure as a service (IaaS) in cloud computing: Challenges and opportunities," *J. Netw. Comput. Appl.*, vol. 68, pp. 173–200, Jun. 2016.

[6] J. Shuja, K. Bilal, S. A. Madani, M. Othman, R. Ranjan, P. Balaji, and S. U. Khan, "Survey of techniques and architectures for designing energy-efficient data centers," *IEEE Syst. J.*, vol. 10, no. 2, pp. 507–519, Jun. 2016.

[7] S. H. H. Madni, M. S. A. Latiff, Y. Coulibaly, and S. M. Abdulhamid, "Recent advancements in resource allocation techniques for cloud computing environment: A systematic review," *Cluster Comput.*, vol. 20, pp. 2489–2533, Sep. 2017.

[8] J. Shuja, A. Gani, S. Shamshirband, R. W. Ahmad, and K. Bilal, "Sustainable Cloud Data Centers: A survey of enabling techniques and technologies," *Renew. Sustain. Energy Rev.*, vol. 62, pp. 195–214, Sep. 2016.

[9] J. Koomey, *Growth in Data Center Electricity Use 2005 to 2010.* Oakland, CA, USA: Analytics Press, Jul. 2011. [Online]. Available: http://www.analyticspress.com/datacenters.html

[10] K. Bilal, S. U. Khan, and A. Y. Zomaya, "Green data center networks: Challenges and opportunities," in *Proc. 11th Int. Conf. Frontiers Inf. Technol.*, Dec. 2013, pp. 229–234.

[11] K. Bilal, S. U. R. Malik, S. U. Khan, and A. Y. Zomaya, "Trends and challenges in cloud datacenters," *IEEE Cloud Comput.*, vol. 1, no. 1, pp. 10–20, May 2014.

[12] R. Bianchini and R. Rajamony, "Power and energy management for server systems," *Computer*, vol. 37, no. 11, pp. 68–76, Nov. 2004.

[13] A. Khosravi, S. K. Garg, and R. Buyya, "Energy and carbon-efficient placement of virtual machines in distributed cloud data centers," in *Euro-Par 2013 Parallel Processing* (Lecture Notes in Computer Science), vol. 8097. Berlin, Germany: Springer, 2013, pp. 317–328.

[14] G. Koutitas and P. Demestichas, "Challenges for energy efficiency in local and regional data centers," *J. Green Eng.*, vol. 1, no. 1, pp. 1–32, 2010.

[15] M. Webb, "SMART 2020: Enabling the low carbon economy in the information age," Climate Group, London, U.K., Tech. Rep., 2008. [Online]. Available: https://www.theclimategroup.org/what-we-do/news-and-blogs/SMART-2020-Enabling-the-low-carbon-economy-in-the-information-age

[16] J. Whitney and P. Delforge, "Data center efficiency assessment—Scaling up energy efficiency across the data center industry: Evaluating key drivers and barriers," NRDC, New York, NY, USA, Tech. Rep. LBNL-363E, 2014.

[17] P. Scheihing, "Creating energy efficient data centers," presented at the Data Center Facilities Eng. Conf., Washington, DC, USA, May 2007. [Online]. Available: https://slideplayer.com/slide/5896044/

[18] A. Hameed, A. Khoshkbarforoushha, R. Ranjan, P. P. Jayaraman, J. Kolodziej, P. Balaji, S. Zeadally, Q. M. Malluhi, N. Tziritas, A. Vishnu, S. U. Khan, and A. Zomaya, "A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems," *Computing*, vol. 98, no. 7, pp. 751–774, Jul. 2016. doi: 10.1007/s00607-014-0407-8.

[19] C.-M. Wu, R.-S. Chang, and H.-Y. Chan, "A green energy-efficient scheduling algorithm using the DVFS technique for cloud datacenters," *Future Generat. Comput. Syst.*, vol. 37, pp. 141–147, Jul. 2014.

[20] A. Alnowiser, E. Aldhahri, A. Alahmadi, and M. M. Zhu, "Enhanced weighted round robin (EWRR) with DVFS technology in cloud energy-aware," in *Proc. Int. Conf. Comput. Sci. Comput. Intell.*, Mar. 2014, pp. 320–326.

[21] Y. Ren, J. Suzuki, C. Lee, A. V. Vasilakos, S. Omura, and K. Oba, "Balancing performance, resource efficiency and energy efficiency for virtual machine deployment in DVFS-enabled clouds: An evolutionary game theoretic approach," in *Proc. Companion Publication Annu. Conf. Genetic Evol. Comput.*, Jul. 2014, pp. 1205–1212.

[22] S. Ali, S.-Y. Jing, and S. Kun, "Profit-aware DVFS enabled RM of IaaS cloud," *Int. J. Comput. Sci. Issues*, vol. 10, no. 2, pp. 237–247, 2013.

[23] S. Esfandiarpoor, A. Pahlavan, and M. Goudarzi, "Structure-aware online virtual machine consolidation for datacenter energy improvement in cloud computing," *Comput. Elect. Eng.*, vol. 42, pp. 74–89, Feb. 2015.

[24] R. Nathuji and K. Schwan, "VirtualPower: Coordinated power management in virtualized enterprise systems," *ACM SIGOPS Oper. Syst. Rev.*, vol. 41, no. 6, pp. 265–278, 2007.

[25] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu, "No 'power' struggles: Coordinated multi-level power management for the data center," in *Proc. 13th Int. Conf. Architectural Support Program. Lang. Operating Syst.*, Mar. 2008, pp. 48–59.

[26] A. Horri, M. S. Mozafari, and G. Dastghaibyfard, "Novel resource allocation algorithms to performance and energy efficiency in cloud computing," *J. Supercomput.*, vol. 69, no. 3, pp. 1445–1461, Sep. 2014.

[27] 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 Comput.*, vol. 12, no. 1, pp. 1–15, Mar. 2009.

[28] Z. Xiao, W. Song, and Q. Chen, "Dynamic resource allocation using virtual machines for cloud computing environment," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1107–1117, Jun. 2013.

[29] B. Addis, D. Ardagna, B. Panicucci, M. S. Squillante, and L. Zhang, "A hierarchical approach for the resource management of very large cloud platforms," *IEEE Trans. Dependable Secure Comput.*, vol. 10, no. 5, pp. 253–272, Sep./Oct. 2013.

[30] T. Nowicki, M. S. Squillante, and C. W. Wu, "Fundamentals of dynamic decentralized optimization in autonomic computing systems," in *Self-Star Properties in Complex Information Systems* (Lecture Notes in Computer Science), vol. 3460. Berlin, Germany: Springer, 2005, pp. 204–218.

[31] D. Ardagna, B. Panicucci, M. Trubian, and L. Zhang, "Energy-aware autonomic resource allocation in multitier virtualized environments," *IEEE Trans. Serv. Comput.*, vol. 5, no. 1, pp. 2–19, Jan./Mar. 2012.

[32] R. W. Ahmad, A. Gani, S. H. Ab Hamid, M. Shiraz, A. Yousafzai, and F. Xia, "A survey on virtual machine migration and server consolidation frameworks for cloud data centers," *J. Netw. Comput. Appl.*, vol. 52, pp. 11–25, Jun. 2015.

[33] 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, no. 13, pp. 1397–1420, Sep. 2012.

[34] S. Martello and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations*. Hoboken, NJ, USA: Wiley, 1990.

[35] S. Mustafa, K. Bilal, S. A. Madani, N. Tziritas, S. U. Khan, and L. T. Yang, "Performance evaluation of energy-aware best fit decreasing algorithms for cloud environments," in *Proc. IEEE Int. Conf. Data Sci. Data Intensive Syst.*, Dec. 2015, pp. 464–469.

[36] C. Li, D. Feng, Y. Hua, and L. Qin, "Efficient live virtual machine migration for memory write-intensive workloads," *Future Gener. Comput. Syst.*, vol. 95, pp. 126–139, Jun. 2019.

[37] J. Shuja, A. Gani, Muhammad H. ur Rehman, E. Ahmed, S. A. Madani, M. K. Khan, and K. Ko, "Towards native code offloading based MCC frameworks for multimedia applications: A survey," *J. Netw. Comput. Appl.*, vol. 75, pp. 335–354, Nov. 2016.

[38] M. H. ur Rehman, C. Sun, T. Y. Wah, A. Iqbal, and P. P. Jayaraman, "Opportunistic computation offloading in mobile edge cloud computing environments," in *Proc. 17th IEEE Int. Conf. Mobile Data Manage.*, Jun. 2016, pp. 208–213.

[39] A. Beloglazov, J. Abawajy, and R. Buyya, ‘‘Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing,’’ *Future Generat. Comput. Syst.*, vol. 28, no. 5, pp. 755–768, 2012.

[40] Y. C. Lee and A. Y. Zomaya, ‘‘Energy efficient utilization of resources in cloud computing systems,’’ *J. Supercomput.*, vol. 60, no. 2, pp. 268–280, May 2012.

[41] R. 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, no. 1, pp. 23–50, 2011.

[42] *Amazon Elastic Computing Cloud (EC2)*. Accessed: Jul. 15, 2019. [Online]. Available: http://aws.amazon.com/ec2/instance-types>

[43] K. Park and V. S. Pai, ‘‘CoMon: A mostly-scalable monitoring system for PlanetLab,’’ *ACM SIGOPS Oper. Syst. Rev.*, vol. 40, no. 1, pp. 65–74, 2006.

[44] F. Farahnakian, T. Pahikkala, P. Liljeberg, J. Plosila, N. T. Hieu, and H. Tenhunen, ‘‘Energy-aware VM consolidation in cloud data centers using utilization prediction model,’’ *IEEE Trans. Cloud Comput.*, vol. 7, no. 2, pp. 524–536, Apr./Jun. 2019. doi: 10.1109/TCC.2016.2617374.

[45] F. Farahnakian, R. Bahsoon, P. Liljeberg, and T. Pahikkala, ‘‘Self-adaptive resource management system in IaaS clouds,’’ in *Proc. IEEE 9th Int. Conf. Cloud Comput.*, Jun./Jul. 2016, pp. 553–560.

[46] S. Mustafa, K. Bilal, S. U. R. Malik, and S. A. Madani, ‘‘SLA-aware energy efficient resource management for cloud environments,’’ *IEEE Access*, vol. 6, pp. 15004–15020, 2018.

[47] N. Tziritas, S. Mustafa, M. Koziri, T. Loukopoulos, S. U. Khan, C.-Z. Xu, and A. Y. Zomaya, ‘‘Server consolidation in cloud computing,’’ in *Proc. IEEE 24th Int. Conf. Parallel Distrib. Syst.*, Dec. 2018, pp. 194–203.

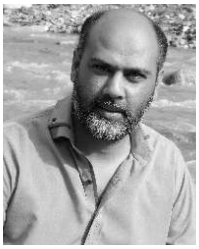
[48] B. Shi and H. Shen, ‘‘Memory/disk operation aware lightweight VM live migration across data-centers with low performance impact,’’ in *Proc. IEEE Conf. Comput. Commun.*, Apr./May 2019, pp. 334–342. doi: 10.1109/INFOCOM.2019.8737639.

SAAD MUSTAFA received the B.S., M.S., and Ph.D. degrees in computer science from COMSATS University Islamabad, in 2007, 2010, and 2018, respectively. He is currently with COMSATS University Islamabad, Abbottabad, Pakistan, since 2010. His research interests include resource management, energy-efficient systems, cloud computing, mobile edge computing, the Internet of Things, and wireless networks.

KINZA SATTAR received the bachelor’s degree in telecommunication and networking, and the master’s degree in computer science from COMSATS University Islamabad, in 2012 and 2015, respectively. She is currently serving as a Lecturer with the University of Management and Technology, Sialkot, Pakistan. Her research interests include resource management in cloud computing and energy-efficient resource allocation algorithms.

JUNAID SHUJA received the B.S. degree in computer and information science from the Pakistan Institute of Engineering and Applied Sciences (PIEAS), Islamabad, in 2009, the M.S. degree in computer science from CIIT Abbottabad, in 2012, and the Ph.D. degree from the University of Malaya, Malaysia, in 2017. His Ph.D. thesis focused on the execution of SIMD instructions on heterogeneous mobile and cloud platforms. He is currently an Assistant Professor with COMSATS University Islamabad (CUI), Abbottabad Campus, Pakistan. His research interests include energy-efficient cloud data centers, edge computing, and applications of Blockchain in finance. He has published research in more than 30 International journals/conferences. He is an Associate Editor of IEEE Access.

SHAHZAD SARWAR received the B.Sc. degree in civil engineering from the University of Engineering and Technology (UET) Taxila, Pakistan, in 1998, the M.S. degree in computer science from the Lahore University of Management Sciences (LUMS), Pakistan, in 2004, and the Ph.D. degree in electrical engineering and information technology from the Vienna University of Technology, Austria, in 2008. He is currently an Assistant Professor with the Punjab University College of Information Technology (PUCIT), University of the Punjab, Pakistan. His main areas of research are optical burst switched (OBS) networks, high-speed data centers, the Internet of Things, and cloud computing. He has participated in ePhotonONe+ and BONE projects funded by the European Union. He is a member of the Pakistan Engineering Council, and the Pakistan Engineering Congress.



TAHIR MAQSOOD received the M.Sc. degree in computer networks from Northumbria University, U.K., in 2007, and the Ph.D. degree in computer science from COMSATS University Islamabad, Pakistan, in 2017. He is currently an Assistant Professor with COMSATS University Islamabad, Abbottabad Campus, Pakistan. His research interests include resource allocation, multi/manycore systems, reliable systems, the Internet of Things, and mobile edge computing.

SAJJAD A. MADANI received the M.S. degree in computer sciences from the Lahore University of Management Sciences, and the Ph.D. degree from the Vienna University of Technology. He is currently a Professor with COMSATS University Islamabad, Pakistan. He has published more than 90 articles in peer-reviewed international conferences and journals. His areas of interests include low power wireless sensor network and green computing.

SGHAIER GUIZANI received the Ph.D. degree from the University of Quebec, Canada, in 2007. He is currently an Assistant Professor with the Electrical Engineering Department, Alfaisal University, Riyadh, Saudi Arabia. His research interests include communication networks and security (particularly wireless ad hoc, sensor networks, QoS, wireless sensor network security, and RFID/NFC application and security) and the Internet of Things. He has published a number of research articles in refereed international conferences and journals. He has been involved in a number of conferences and workshops in various capacities. He has served/is serving as an Associate Editor for the *Security and Communication Networks* (Wiley), the *International Journal of Sensor Networks (Inderscience)*, and the *Journal of Computer Systems, Networking, and Communications*.