IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 26, 2019, accepted December 13, 2019, date of publication December 25, 2019,
date of current version January 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2962272

Utilization Driven Model for Server Consolidation in Cloud Data Centers

HAMMAD UR-REHMAN QAISER[1], GAO SHU[1], AND ASAD WAQAR MALIK[2]
[1]School of Computer Science and Engineering, Wuhan University of Technology, Wuhan 430070, China
[2]School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad 44000, Pakistan

Corresponding author: Hammad Ur-Rehman Qaiser (hrqaiser@hotmail.com)

ABSTRACT The application of cloud computing has diversified with the adoption of Internet of Things (IoTs) and edge computing. However, it has increased the uncertainty of workload demand; thus, the efficient utilization of cloud computing resources become more challenging. Traditionally, dynamic consolidation of workload inside cloud data centers relies on identifying overload and under-load hosts using either static or dynamic threshold value. In this paper, we propose a Utilization Driven Model (UDM) model to estimate the number of under-utilized and over-utilized processing machines through percentile ranks of low and high utilization from mean value of resource utilization of hosts and the value of mean absolute deviation of resource demand. UDM swiftly reacts to any change in workload demand and adapts the system to the current demand of resource utilization. The UDM approach not only impacts the energy consumption and quality of service but also increases the elastic nature of cloud by robustly managing the sudden changes in workload. Experiment results show that UDM is an efficient server consolidation technique, improving 30% energy, 40% quality of service compared to contemporary techniques. Thus, the UDM is more robust to support stochastic resource demand compared to traditional techniques.

INDEX TERMS Server consolidation, cloud computing, efficient resource utilization, cloud data centers, dynamic workload consolidation.

I. INTRODUCTION

Cloud Computing manages a large diversity of heterogeneous services and applications, particularly in the era of 5G. It provides computing resources and services to the organizations, based on on-demand and pay as you go model. The enormous size of modern cloud computing infrastructure consumes a huge amount of electricity for cooling, and illumination, and data center related operations. Estimates reveal that the energy cost contributes almost 40% to the total cost of the data center [1]. Apart from cost, carbon emission associated with modern cloud data centers amounts to almost 2% of the total global emission of CO_2 [2], [3]. A significant amount of energy consumption and carbon emission can be reduced by utilizing computing resources, efficiently [4]. In cloud data centers, computing resources are allocated through Virtual Machines (VMs) or containers, so that the underlying cloud infrastructure may be shared among the users. Only a small but varying fraction of allocated resources are utilized by organizations at a certain time [5]. Therefore, to enhance the utilization rate of active computing resources, more resources are allocated to processing machines than their total capacity. Resource utilization rate being unpredictably variable, the allocation of workload is dynamically consolidated on the optimal number of processing machines by migrating VMs from underutilized processing machines to other suitable machines [6]. However, it may lead to aggressive consolidation in case resource utilization requests to Processing Machines (PMs) exceed their capacity, too often resulting in the Quality of Service (QoS) issues. In cloud computing, QoS is often defined via Service Level Agreements (SLAs) [7]. A good server consolidation policy must ensure reliable QoS along with reducing energy consumption for the execution of the workload. Thus, to bring balance between the conservation of energy and quality of service, it is equally important to manage both, under-utilized and over-utilized hosts.

Studies have also shown that the pervasive adoption of cloud in the IoT framework has increased the uncertainty of resource utilization. As a result, it has exacerbated the problems of inefficiencies in resource provisioning causing

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

degradation in the provision of quality of service as defined by Service Level Agreement (SLA) [8] and an increase in energy consumption. This has increased the significance of elasticity of resource provisioning in the cloud data center. Resource elasticity is an important feature of clouds that refers to the ability of a cloud to adjust its active resources as per resource demand. The adjustment is made to accommodate changes in the resource utilization rate of a service as well as to facilitate acquiring or releasing resources on demand. In a highly distributed cloud environment based on IoTs and edge computing with stochastic workload demand, it has become ever more challenging to maintain elasticity as required. Resource management decisions based on current load demand may lead to achieving this elasticity [9]. By doing so, resource provisioning can be managed robustly. Traditionally, resource optimization in a cloud data center corresponds to an energy-quality trade-off problem and policies mainly focus on bringing the best scenario out of it. Uncertainty of workload demand is catered, to an extent, by monitoring resource utilization and dynamically adapting the resource provisioning. To forestall the repercussion of sudden change in workload demand, prediction based approaches have also been implied, as discussed in the next section in detail. However, increased stochastic resource demand due to changing environment, after the integration of IoT and edge computing, requires an extremely robust adaption of resource provisioning with more focus on elasticity. There is an ample need to improve the underlying infrastructure of service delivery as well as the robust mechanism and policies for the execution of workload that are tolerant to uncertainty and stochastic behavior of resource demands to meet the recent challenges. Kratzke [10] observes that the evolution of cloud computing architecture is a steady process of resource utilization optimization.

Keeping in view, the proposed Utilization Driven Model for server Consolidation helps in tackling the NP-hard problem of workload consolidation efficiently along with catering to the new challenges of increased uncertainty in workload demand. Utilization Driven Model (UDM) for server consolidation in the cloud data center emphasize on current resource demand. UDM optimizes resource utilization in a stochastic environment along with increased elasticity of resource provision that is best suited for IoT focused environment. UDM focus is not limited to optimizing energy consumption and quality of service but it also manages active resources by keeping utilization rate high and variance of utilization low. By robustly adapting the resource provision in a stochastic environment, UDM adds up the elastic nature of the data center. Thus, UDM contributes to solving traditional as well as new challenges faced by cloud data center due to the increased reliance of IoTs on it. It improves energy efficiency by increasing the average resource utilization and maintains the quality of service by lowering the variance in utilization on active hosts.

The remaining paper has been arranged in the following sections. Section II presents the literature review and discusses the work in the field of dynamic consolidation of workload. Section III explains the utilization driven model for consolidation, along with its algorithm. Section IV provides experiment setup and results of UDM based algorithm along with six other server consolidation techniques. Section V explains the result with comparative analysis on the basis of performance and final section concludes the work.

II. RELATED WORK

Dynamic consolidation of workload on the optimal number of servers under time-varying resource demand is an active research area and several strategies have been proposed in this regard. Best Fit Decreasing (BFD) algorithm has been utilized for allocation of virtual machines (VMs) to suitable host processing machines in a data center. In BFD, VMs are sorted in decreasing order of their resource demand. Then Best Fit algorithm is used to allocate the VMs to the processing machines, turn by turn. To allocate VMs to suitable PMs, Beloglazov *et al.* [11] proposed a modified version of BFD algorithm known as Power-Aware Best Fit Decreasing [PABFD] heuristic. Similar to BFD, PABFD, also sorts the VMs in the decreasing order of resource demand. PABFD uses CPU as a resource demand to sort the VMs. It then, turn by turn, allocates the VM to the processing machine based on the least increase in power consumption.

For dynamic server consolidation, under-load and overload host are detected using upper and lower resource utilization thresholds. After that, the workload is re-allocated to suitable hosts. In another work [12], the authors proposed three heuristics for dynamic consolidation of servers based on adaptive thresholds of resource utilization. Historical traces of workload are used to adjust the utilization thresholds. Three adaptively adjusted host overload detection methods were proposed: Inter-Quartile Range (IQR), Median Absolute Deviation (MAD), and Local Regression (LR). Inter-Quartile Range (IQR) uses first and third quartile as the lower and upper threshold of utilization. The first quartile is the measure of 0.25 percentile while the third quartile is the 0.75 percentile. Local Regression uses a linear equation to map the relationship between input variables and a dependent output variable. MAD is a median of absolute deviation from the median value of the observed data. As compared to standard deviation, MAD is more resilient to outlier data. For the reallocation of VMs, they used the PABFD algorithm. Farahnakian and Pahikkala [13] proposed a K-nearest neighbor (DC-KNN) algorithm for dynamic consolidation of VMs on the optimal number of servers. K-value is predicted using cross-validation technique and K-nearest neighbor regression is employed to predict the workload on PMs. Based on cluster results, server consolidation is executed. Li [14] identified the overload threshold of resource utilization based on Markov decision process. He utilizes a modified version of the Bellman optimality equation to adaptively adjust the overload threshold value. He considered resource utilization, energy consumption, and quality of service. A prediction based algorithm, utilization

prediction-aware best fit decreasing (UPBFD) algorithm was proposed in [15]. UPBFD is based on best fit decreasing which also predict workload demand and performance for consolidation of workload.

Salimian *et al.* [16] proposed a fuzzy threshold-based approach for the detection of under-utilized and over-utilized processing machines. They applied Sugeno fuzzy rule set based fuzzy inference engine to detect over utilized hosts for energy and performance efficient consolidation of virtual machines. In [17], a self-adaptive heuristic for the detection of under-utilized and over-utilized processing machines was presented. It used queuing theory to propose a probabilistic model of the data center. Consolidation related decisions were taken from the assessment results obtained from the probabilistic model. In their work [18], authors propose a slightly different approach for reducing the workload execution time. They proposed that the mapping of tasks and servers should be based on the nature of tasks and task relative resource availability of servers. They are of the view that task scheduling time can be reduced if careful mapping of servers and tasks is carried out. In their work [19], authors utilized neural network model along with a factor model to forecast the change in resource utilization using historical data or resource utilization. Abbasi and Jin [20] proposed algorithms that apply Fuzzy AHP on a graphed-theoretic model for efficient workload placement and server consolidation. This approach is easy to implement and efficient for a limited number of processing machines and tasks, but for larger networks of machines and tasks, the combinatorial approach increases the complexities in resource allocation. To efficiently consolidate workload on optimal hosts, Han *et al.* [21] proposed two heuristics, remaining utilization-aware (RUA) algorithm and power-aware (PA) algorithm. RUA initially places the VMs to the suitable hosts while PA finds proper hosts for VM replacement during the consolidation phase.

Another class of algorithms for efficient resource utilization is known as Meta-heuristic based algorithms. A meta-heuristic based algorithm, Simulated Annealing, was proposed in [22] that used the optimizing technique for finding an approximation to the global optimum of a function in a very large search space. Annealing is a technique used in metallurgy in which metal is heated and then cooled in a controlled manner to improve its crystalline shape and remove defects. In this process, perturbation phase is analogous to VM consolidation. First, a host with the least utilization rate is selected as a source host. Then workload is transferred from this host to some host having medium utilization rate. In this way, the workload is consolidated on fewer processing machines. Occasionally a less optimized solution is selected for adopting the processing to escape stagnation of solution in local minima and local maxima. Joshi and Kaur [23] proposed a Cuckoo Algorithm based server consolidation method. It deals with consolidation as a multi-dimensional packing problem for optimizing utilization of computing resources. Joshi did not consider the dynamic consolidation and did not cater to the variable workload demand and resource utilization. In their work [24], the authors modified the Grey Wolf Optimization algorithm for server consolidation, named as levy based multi-objective gray wolf optimization (LMOGWO) algorithm. The idea is based on the behavior of grey wolf when they hunt for food. Zheng *et al.* [25] proposed a meta-heuristic based algorithm for optimizing resource utilization in a cloud data center based on biogeography combining optimization algorithm. He focuses on current resource demand to optimize the energy consumption and quality of service but ignores the stochastic needs of workload demand. Ant Colony Optimization (ACO) based algorithm was proposed by Farahnakian *et al.* [26] for consolidating the workload. The method LiRCUP is utilized to identify the outlier hosts a global agent was utilized to dynamically consolidate the workload on the optimum number of servers. LiRCUP uses a static value of upper and lower thresholds to identify over-utilized and under-utilized hosts. This can effectively reduce the search space to a limited number of VMs. Similar to [26], Particle Swarm Optimization (PSO) based algorithm was proposed by Li *et al.* [27] for VMs consolidation that also utilizes static thresholds for detecting under-utilization and over-utilization hosts. Another PSO based VM consolidation model is proposed by Li *et al.* [28]. The proposed model incorporates Euclidean distance of resource and degree of user satisfaction along with traditional energy consumption and quality of service related traits. PSO, in combination with the proposed model, is utilized to optimize the objective function based on power consumption per QoS value.

Dynamic consolidation helps in optimizing resource utilization in the cloud data center [29]. Meta-heuristic based methods [22]–[28] have occasionally shown better results, but they tend to slow down the optimization processes with the exceeding number of VMs involved as the search space grows quite significantly. Some works [25]–[27] tried to limit the number of VMs for reducing search space by taking a static value of threshold of utilization. This solves the problem of growing search space to some extent but it impacts the optimization process adversely. It has been shown [12] that adaptive threshold based techniques yield far better results than static threshold based techniques. Some methods [12]–[17] utilize adaptive thresholds that can adjust their values based on prediction. Historical data is used to predict resource utilization and to adjust the threshold value accordingly. However, due to increasing uncertainty of workload demand in modern cloud infrastructures based on IoTs and edge, these techniques become lethargic to the change in resource demand that is not predictable from past data. This impacts the robustness and elasticity of the resource provisioning of the cloud system.

Utilization driven model for server consolidation, as proposed in this work focuses on robustness in resource provision and mainly relies on current load demand. It abruptly reacts to a change in workload by adjusting utilization threshold for efficient resource utilization. It focuses on five significant aspects of dynamic workload consolidation

simultaneously, in addition to improve the energy consumption and quality of service; it considers high utilization of processing machines with low variation of load on them. UDM uses percentile ranks to estimate under-utilized and over-utilized hosts based on current workload in such a way that the consolidation process is robust. Thus, it also adds to the elasticity of the cloud environment. In order to focus on degraded electric consumption percentile ranks are based on the mean utilization rate of active processing machines, whereas to maintain the quality of service, variation in resource demands is also given significant importance. This increases the utilization rate when workload-demand is uniform and changes it according to the level of variations in workload demand. The process is done rather robustly, by incorporating percentile ranks to provide the number of over-utilized and under-utilized hosts.

III. UTILIZATION DRIVEN MODEL FOR SERVER CONSOLIDATION BASED ON CURRENT DEMAND

For dynamic consolidation of servers in data centers, under-utilized and over-utilized machines are identified using static or dynamic thresholds. Then workload is migrated accordingly. Utilization Driven Model (UDM) for server consolidation, as proposed in the work, estimates the number of under-utilized and over-utilized hosts using current resource demand by VMs and utilization rate of PMs. Estimation is made after determining the upper and lower percentile rank. First, we calculate the 'mean' value of resource utilization of active hosts and the value of 'mean absolute deviation' of the current resource demand of active VMs. Then the values of the upper and lower threshold are obtained using mean value and the value of mean absolute deviation, as discussed. Finally, upper and lower percentile ranks are determined using thresholds. These percentile ranks estimates the number of over-utilized hosts and under-utilized hosts in time-varying scenario.

Suppose i is a number of active hosts and h is a host with u representing corresponding current resource utilization of h then the set of all hosts with corresponding resource utilization is as follows $\{(\mathrm{h,u})\} = \{(h_1, u_1), (h_2, u_2)\,(h_3, u_3), \ldots\ldots, (h_i, u_i)\}$. Likewise, for j number of virtual machines with v as a virtual machine and d as a corresponding resource demand of v, set of all VMs with corresponding resource demand is as follows $\{(\mathrm{v,d})\} = \{(v_1, \mathrm{d}_1), (v_2, \mathrm{d}_2)\,(v_3, \mathrm{d}_3), \ldots., (v_j, \mathrm{d}_j)\}$. Mean value of utilization μ_u of active hosts and the value of mean absolute deviation D_d for resource demand by VMs can be determined using the following equations

$$\mu_u = \frac{1}{i}\sum_{n=1}^{i} u_n \tag{1}$$

$$D_d = \frac{1}{j}\sum_{n=1}^{j} |d_n - m(d)| \tag{2}$$

Here μ_u is mean resource utilization of active hosts, u_n is the utilization of n^{th} host, D_d is the mean absolute deviation from the median for resource requests, $m(d)$ is the median of resource requests by VMs, d_n is the resource demand by n^{th} VM. Here it is pertinent to mention that out of different types of mean absolute deviation we used mean absolute deviation from median because it yields the least value as compared to all other mean absolute deviations and is helpful for the purpose of best convergence.

We determined upper threshold of utilization T_u as "complement 1 of the product of μ_u and D_d" as follows

$$T_u(u, d) = 1 - \mu_u \times D_d \tag{3}$$

Whereas lower threshold of utilization T_l is a split function with value as "complement 1 of sum of μ_u and D_d" when $\mu_u + D_d < 1$ and with value as "product of μ_u and D_d" when $\mu_u + D_d \geq 1$.

$$T_l(u, d) = \begin{cases} 1 - (\mu_u + D_d) & \text{if } \mu_u + D_d < 1 \\ \mu_u \times D_d & \text{if } \mu_u + D_d \geq 1 \end{cases} \tag{4}$$

As indicated in (3) and (4), upper threshold of utilization T_u and lower threshold of utilization T_l are based on resource utilization of hosts μ_u and resource demand of virtual machines D_r. It means that we incorporate both current utilization of hosts and current resource demand of VMs to determine the utilization thresholds.

In order to see the effectiveness of the technique and to formulate the optimization function for resource consolidation, acceptance range R is calculated by subtracting the lower threshold T_l from the upper threshold T_u of utilization of resources. Acceptance range is the range of resource utilization where hosts are considered to be well utilized i.e. neither the resources are being under-utilized nor there is a significant QoS related issue and it can be determined as follows

When $\mu_u + D_d < 1$

$$\begin{aligned} R &= T_u - T_l \\ &= 1 - \mu_u \times D_d - (1 - (\mu_u + D_d)) \\ &= 1 - \mu_u \times D_d - (1 - \mu_u - D_d) \\ &= 1 - \mu_u \times D_d - 1 + \mu_u + D_d) \\ &= \mu_u + D_d - \mu_u \times D_d \end{aligned}$$

When $\mu_u + D_d \geq 1$

$$\begin{aligned} R &= T_u - T_l \\ &= 1 - \mu_u \times D_r - \mu_u \times D_r \\ &= 1 - 2 \times \mu_u \times D_r \end{aligned}$$

From the above values of R, optimization function is determined as

$$R(u, \mathrm{d}) = \begin{cases} \mu_u + D_d - \mu_u \times D_d & \text{if } \mu_u + D_d < 1 \\ 1 - 2 \times \mu_u \times D_d & \text{if } \mu_u + D_d \geq 1 \end{cases} \tag{5}$$

This acceptance range R gets itself fine-tunes depending upon the current resource demand, utilization rate of resources and nature of stochastic behavior. This range gets wider and lower when resource demand is highly uncertain and it gets

narrow and high when the resource demand is uniform in nature. This optimizes resource utilization without impacting the quality of service. It is pertinent to see how R converge the consolidation process towards optimal resource utilization. As μ_u is the mean resource utilization of the active hosts and D_d is the mean absolute deviation from the median of resource demand. The high values of μ_u and D_d will result into an increase in range R and a decrease in the values of T_u and T_l. On the other hand, the low values of μ_u and D_d will result in a decrease in range R and an increase in the values of T_u and T_l. When only one'' of the values of μ_u and D_d are high, it will result in a moderate value of range R, T_u and T_l. So, (5) will always converge the solution towards optimal utilization.

Instead of using directly as a threshold of utilization, we used the values of T_u and T_l to calculate the percentile ranks P_u and P_l for estimating over-utilized hosts and under-utilized hosts using (6) and (7), respectively

$$P_u = 100 \times T_u \tag{6}$$

$$P_l = 100 \times T_l \tag{7}$$

Percentile utilization P_i of host i can be determined using (8), where n represents the number of hosts with resource utilization less than the resource utilization of i and N represents the total number of active hosts.

$$P_i = \frac{n}{N} \times 100 \tag{8}$$

Hosts, with resource utilization percentile P_i more than P_u, are marked as over-utilized hosts and hosts with resource utilization percentile P_i less than P_l are marked as under-utilized hosts.

As percentile represents the percentage of the number of observations instead of the value of observed phenomenon, the method directly estimates the number of over-utilized hosts and under-utilized hosts instead of linking with the threshold value of resource utilization. Taking percentile ranks instead of thresholds for identifying outlier hosts makes the cloud system more elastic. Incorporating the mean utilization rate of active hosts in optimizing function helps in increasing the utilization of active resources. Hence it helps to reduce energy consumption. On the other hand, incorporating the mean absolute deviation of resource demand helps to ensure the quality of service as it caters for the sudden changes in workload demand. We, further analyze the effectiveness of optimization function by taking four different cases.

1) Let suppose μ_u and D_d are low and have values, 0.50 and 0.05, respectively, then P_u will be 97.5 and P_l will be 45 and this will result in the selection of 2.5% of machines as over-utilized and 45% of machines as under-utilized.
2) If μ_u and D_d are relatively high and have values, 0.90 and 0.15, respectively, then P_u will be 86.5 and P_l will be 13.5 and this will result in the selection of 13.5% of high utilized machines as over-utilized and about 13.5% will be selected as under-utilized hosts.
3) If μ_u has relatively low and D_d has high value, 0.50 and 0.15, respectively then P_u will be 92.5 and P_l will be 35 and this will result in the selection of 7.5% of high utilized machines as over-utilized and about 35% will be selected as under-utilized hosts.
4) If μ_u has relatively high and D_d has low value, 0.90 and 0.05, respectively then P_u will be 95.5 and P_l will be 5 and this will result in the selection of 4.5% of high utilized machines as over-utilized and about 5% will be selected as under-utilized hosts.

Thus the model identifies hotspots based on dispersion in resource demand and mean value of resource utilization. The Algorithm 1 presents a utilization driven model for server consolidation based on current resource demand. List of hosts and virtual machines along with their available resources and requested resources, respectively, have been provided as an input to the UDM algorithm and list of over-utilized hosts and under-utilized hosts are obtained as an output of the algorithm. UDM algorithm, calculates the upper threshold T_u and lower threshold T_l of utilization, using (1), (2), (3), and (4), after determining μ_u and D_d.

The values of T_u and T_l are used to calculate the percentile ranks P_u and P_l for estimating over-utilized hosts and under-utilized hosts. The use of percentile ranks P_u and P_l instead of the values of T_u and T_l, makes consolidation process robust to cater to the uncertainty of workload demand. On the basis of the values of P_i, P_u and P_l, over-utilized and under-utilized hosts are identified. As indicated in step 9 of the algorithm, any host having P_i more than P_u is marked as over-utilized host. On the other hand, step 12 segregates under-utilized hosts on the basis of host having P_i less than P_l. Thus, UDM algorithm first determines the upper and lower threshold values based on current resource requests and current resource utilization of active hosts. After than the percentile ranks of over-utilization are determined using threshold values to identify the over-utilized and under-utilized hosts. Finally, the lists of under-utilized and-over utilized hosts are returned for further consolidation process. We used a slightly modified PABFD algorithm for replacement of VM from the list of over-utilized and under-utilized hosts obtained using UDM. While selecting a host with the least power increase as is the case with PABFD, we assigned additional value of power increase to the hosts that are in shutdown state. This increases the priority to active hosts for replacement of workload and increased the resource utilization of active hosts.

IV. EXPERIMENT AND RESULTS

A. EXPERIMENTAL SET UP

In order to perform experiments to evaluate our proposed Utilization Driven Model (UDM) based algorithms with six other server consolidation policies, i.e. Static Threshold (STR), Inter-Quartile Range (IRQ), Local Regression (LR), Median Absolute Deviation (MAD), K-nearest neighbor (DC-KNN), and Utilization Prediction Best Fit

TABLE 1. Specification of servers.

Server	HP ProLiant ML110 G4	HP ProLiant ML110 G4
Quantity	400	400
Processor(MIPS)MHz	1880	2660
Number of cores	2	2
Memory (MB)	4096	4096
Bandwidth (Gb/s)	1	1

TABLE 2. Features of virtual machines.

Virtual Machine	Large	Medium	Small	Micro
Processor(MIPS)MHz	2500	2000	1000	500
Number of cores	1	1	1	1
Memory (MB)	870	1740	1740	613
Bandwidth (Mb/s)	100	100	100	100

TABLE 3. Features of planet lab traces.

Trace	20110303	20110322
VM Utilized	1052	1516
Mean	12.31	9.26
Standard Deviation	17.09	12.78

Decreasing (UPBFD, we used CloudSim toolkit. The CloudSim toolkit is a well known platform that is used to simulate experiments for Cloud computing environments [30]. The data center used in this study comprised of 800 heterogeneous servers with characteristics given in Table 1.

Relatively large memory size of VM types is to enable over-subscription.

Real-time trace of workload demand, taken from Planet lab, has been used in Cloudsim Toolkit [31]. In the work, we are comparing seven policies based on five performance indices. However, for evaluation purpose, two traces from Planet Lab are used i.e. 20110303 and 20110322 having the highest and the lowest standard deviation of the resource request. Features of these two traces are provided in Table 3. This selection is made to study the effect of burst on the performance of consolidation technique. Therefore, one of the trace (20110303) having relatively more bursty traffic compared to other trace (20110322) used in this study. This is identified with the help of standard deviation of the resource request.

As the workload is dynamic, with 288 instances of new workload every 300 ms, the total time to execute workload is 86,400 ms. The consolidation of VMs is performed on receiving a new workload. Furthermore, to execute UDM, IRQ, MAD, LR, and ST, we used PABFD VM replacement technique along with the Minimum Migration Time (MMT) VM selection algorithm. For selecting VMs from over-utilized hosts to migrate, MMT selects VM that requires the least time to migrate until the over-utilized host becomes normal again.

B. EVALUATION METRICS

In order to evaluate the performance of UDM algorithm, as proposed in this work, we selected five matrices: Energy Consumed (EC), Service Level Agreement Violations (SLAV), SLA violation Time per Active Host (SLATAH), Performance Degradation due to Migrations (PDM), and number of migrations required for the execution of workload.

1. Energy Consumption (EC) is the total amount of energy consumed by processing machines for the execution of workload. Energy consumption is the primary consideration for efficient allocation policies; therefore, low energy consumption is desirous. In order to calculate the energy consumption, CloudSim uses the SPECpower [32] benchmark which is based on the current CPU utilization of processing machines.
2. The number of migrations counts the total number of VM migrations required to perform the execution of workload. It reflects the network congestion contributed through various policies.
3. SLA violation Time per Active Host (SLATAH) represents the degradation of quality of services due to over-utilization of hosts. As long as a host remains overloaded, it cannot fulfill resource demands to all the VMs assigned to it, thus results in a poor performance. SLATAH reflects the fraction of duration an active host remains over-utilized during workload performance and is calculated as follows

$$SLATAH = \frac{1}{L}\sum_{i=1}^{L}\frac{T_{si}}{T_{ai}} \tag{9}$$

where L represents the number of hosts in a datacenter, T_{si} corresponds to the total time, during which the CPU utilization of host i has experienced 100% utilization resulting in the SLA violations, T_{ai} corresponds to the total time of host being in an active state.
4. Performance Degradation due to Migrations (PMD) reflects the QoS degradation due to VM migrations. PDM is given by as follows

$$PDM = \frac{1}{M}\sum_{j=1}^{M}\frac{C_{mj}}{C_{rj}} \tag{10}$$

Here M represents the total number of migrations, C_{mj} represents the cost of migration due to VM j, and C_{rj} is total resource requested by VM j during its life.
5. SLA violation (SLAV) is the fraction of time SLA violations occur when the workload is executed. It reflects the quality of service maintained by consolidation policy. Workload consolidation policies having lower values in SLA violations are desirous. SLAV is the composite of SLATAH and PDM and calculated as following

$$SLAV = SLATAH \times PDM \tag{11}$$

TABLE 4. Results for Policies using trace 20110322.

Method	EC (kWh)	Migrations	SLATAH	PDM	SLAV
UDM	117.33	15776	3.74	0.06	0.00210
UPBFD	140.11	19379	4.07	0.06	0.00241
DCKNN	160.73	24072	4.68	0.06	0.00439
IQR	201.03	28635	4.94	0.06	0.00280
STR	201.78	28769	4.91	0.06	0.00286
LR	177.04	32368	6.22	0.07	0.00461
MAD	197.75	28389	4.96	0.06	0.00288

TABLE 5. Results for Policies using trace 20110303.

Method	EC (kWh)	Migrations	SLATAH	PDM	SLAV
UDM	110.47	12111	3.63	0.05	0.00174
UPBFD	133.10	16242	3.01	0.07	0.00234
DCKNN	152.57	20427	4.01	0.06	0.00453
IQR	189.60	26068	4.80	0.06	0.00290
STR	192.31	26866	4.93	0.06	0.00318
LR	164.00	28103	5.94	0.08	0.00451
MAD	185.98	26275	4.96	0.07	0.00332

C. RESULTS

The performance of seven workload consolidation strategies performed on Cloudsim toolkit is covered in this section. Table 4 and Table 5 provide results of consolidation algorithms along with Utilization Driven Model for consolidation, in each row of the table.

The results of all five indices, Energy Consumption (EC) measured in kWh, number of VM migrations, SLA time per active host, Performance Degradation due to Migrations and SLA violations, as discussed in previous sub-section, have been provided in the respective columns of the tables.

Table 4 indicates the results obtained using planet lab workload trace 20110322 for seven policies in terms of energy consumption, migrations and quality of service (SLATAH, PDM, and overall SAL violation). UDM has executed the workload with the consumption of energy as low as 117.33 kWh whereas power consumption for other policies ranges from 140.11 kWh (UPBFD) to 200 kWh (STR). The migration count for UDM stands at 15,775 whereas other policies cause migrations ranging from 19,379 for UPBFD to 32,368 for LR. SLA stands as low as at 0.00210 for UDM, while other policies contribute more SLA violations ranging from 0.0041 to 0.00461. Table 5 provides results of experiments using Planet lab trace of 20110303. UDM has again shown improvement in results as compared to other policies. Energy required to perform the workload demand remained as low as 110.47 kWh by UDM. UPBFD required 133.10 kWh and STR required 192.31 kWh of energy for the execution of same workload. Migration count for UDM was lowest and stands at 12,111. It was the second best for UPBFD with value 16,242 and the highest for LR with 26,866 migrations. SLA violations stand at 0.00174, whereas it ranges from 0.00234 to 0.00451 for other policies.

D. DISCUSSION

Results of experiments show that UDM improved the energy consumption from 16% to 42% as compared to the second best policy UPBFD to worst policy STR, in terms of energy consumption for trace 20110322. Likewise, UDM improved the quality of service in terms of SLA violations (percent time when SLAV occurs) 13% as compared to UPBFD and 53% to LR. For this trace, utilization driven model proposed in the work showed, on an average showed 35 % improvement in energy consumption, 31% in the number of VM migrations, and 37% improvement in SLA violation.

For trace 20110303 UDM consumed 17% less energy than UPBFD and 42% less energy than STR, UPBFD is the second best and STR is the worst policy in terms of energy consumption for the second workload trace. It showed 26% improvement in quality of service as compared to UPBFD. On average, UDM consumed 36% less energy with 43% improvement in the number of migrations, and 45% of improvement in SLA violations as compared to other six policies. Thus, the UDM showed improved results in all the metrics we described above with low energy consumption, the number of migrations and quality of service related indicators. As for as other policies are concerned, they yield less efficient results for all the indicators. UPBFD showed the second-best results and then DCKNN. Other polices gave mixed results for different parameters. Further analyses for better performance of UDM as compared to other policies have been carried out in the next section.

V. PERFORMANCE ANALYSIS

A. ENERGY EFFICIENCY ANALYSIS

The results of five indices described in the section are obtained using the cloudsim. To provide a holistic view of these indices, Zhou *et al.* [33] formulated energy-efficiency metrics as a composite of energy consumption and quality of service, as follows

$$P = \frac{1}{EC \times SLAV} \tag{12}$$

Here P indicates the energy-efficiency of the policy, EC is the energy consumption and SLAV represents the overall SLA violations occurred during workload execution. The energy consumption and SLA violations are inversely proportional to the efficiency of the system and their lower values are more desirous. Thus, energy efficiency indicator presented by Zhou *et al.* takes the reciprocal of the product of power consumption and SLA violations. We used the model presented by Zhou for evaluating the energy-efficiency of seven

Algorithm 1 Utilization Driven Model Algorithm

Input:

HOST_LIST $= \{h_i \mid i = 1, 2, 3, \ldots .n\}$;
VM_LIST $= \{v_i \mid i = 1, 2, 3, \ldots .m\}$;

Output:

OVER_UTILIZED_LIST $= \{h_o | h_i \in h_o\}$
UNDER_UTILIZED_LIST $= \{h_u | h_i \in h_o\}$

1. $\mu_u = \frac{1}{i} \sum_{n=1}^{i} u_n$ as (1)
2. $D_d = \frac{1}{j} \sum_{n=1}^{j} |d_n - m(d)|$ as (2)
3. $T_u = 1 - \mu_u \times D_d$ as (3)
4. $T_l = \begin{cases} 1 - (\mu_u + D_d) & \text{if } \mu_u + D_d < 1 \\ \mu_u \times D_d & \text{if } \mu_u + D_d \geq 1 \end{cases}$ as (4)
5. $P_u = 100 \times T_u$ as (6)
6. $P_l = 100 \times T_l$ as (7)
7. **for** every h_i in HOST_LIST **do**
8. $P_i = \frac{n}{N} \times 100$ as (8)
9. **if** $P_i > P_u$ **then**
10. *OVER_UTILIZED_LIST* $\leftarrow P_i$
11. **end if**
12. **if** $P_i < P_l$ **then**
13. *UNDER_UTILIZED_LIST* $\leftarrow P_i$
14. **end if**
15. **end for**
16. **return** OVER_UTILIZED_LIST
17. **return** UNDER_UTILIZED_LIST

under consideration VM consolidation policies. Thus, to evaluate the performance of consolidation techniques, energy-efficiency graph for all the policies under consideration has been provided after calculating the P using (13). This will help in comparing the policies holistically as Efficiency graph provides the combined value of energy consumption and quality of service.

Fig. 1 provides holistic performance of policies with policy name on x-axis and energy-efficiency value on y-axis as calculated using (13). Performance for both the traces has been represented in separate columns with policy name labeled. The high value of P implies that UDM has been the most efficient VM consolidation policy, followed by UPBFD and then the others. In the case of trace 20110322, UDM is 38% more efficient than the second-best policy (UPBFD) and 230% more efficient than the worst performer (LR). For workload trace 20110303, UDM is 63% more efficient than the second-best policy (UPBFD) and 280% more efficient than the worst performer (LR). Apart from indicating the better performance of UDM, the above results show that the improvement is not uniform for both the planet lab traces. Some policies are more efficient to perform 20110322 traces while others are more efficient when executing the workload trace 20110303. As both the traces have different values of the mean and standard deviation of workload demand, therefore policies tackle the varied dispersion of workload demand differently.

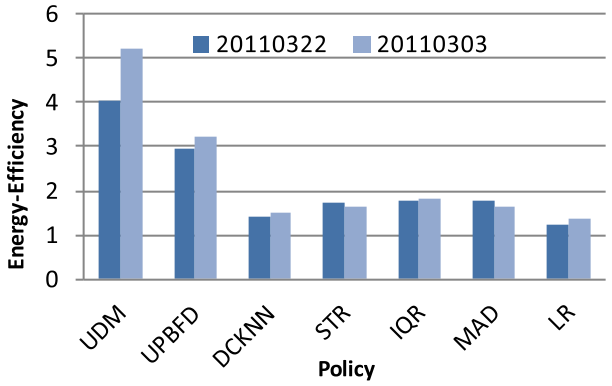


FIGURE 1. Performance measure as a composite of Energy Consumption and SLA violations for seven policies using both planet lab traces as workload.

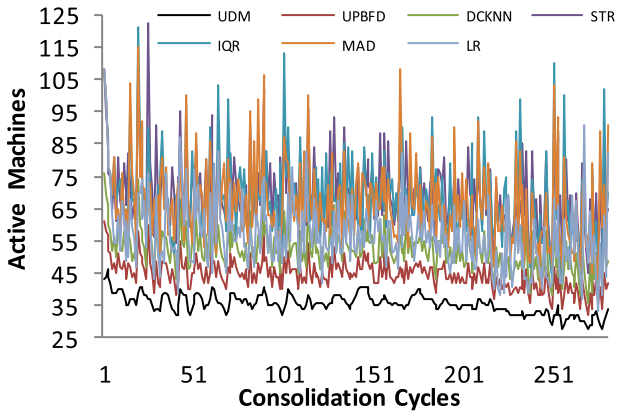


FIGURE 2. Numbers of active machines during each consolidation cycle are indicated.

Some policies react to stochastic demand more efficiently than others. It is pertinent to study which policy may act better in the increasing stochastic and uncertain environment due to the rapid integration of IoT with the cloud.

As the first trace is less stochastic with lower variance in workload demand than the second trace, the graph indicates how well a policy may react to the increased uncertainty in workload demand. The results indicate that when the resource request is highly dispersed in nature UDM's performance increased from 38% to 63% as compared to UPBFD and it increases from 230% to 280% as compared to LR. Thus, UDM is better able to handle non-uniform workload demand. It is because of extremely robust over-load and under-load detection mechanisms based on (5) complemented by the percentile value of upper and lower limits of the utilization of threshold provided in (3) and (4). This makes UDM an ideal server consolidation policy for the changing environment of cloud-IoT due to its ability to cater to the increased uncertainty and variability in workload demands.

B. WORKLOAD CONSOLIDATION ANALYSIS

As we have described in the previous section that the experimental setup to execute seven policies is based on cloudsim and uses 800 hosts for the execution of the workload. The nature of planet lab workload traces is dynamic and it

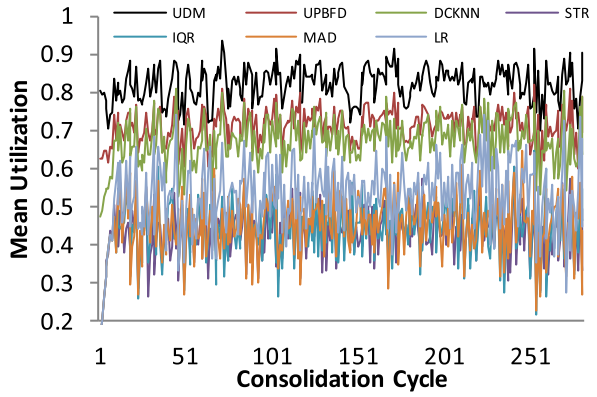


FIGURE 3. Mean utilization rate of resources of active hosts during each consolidation cycle.

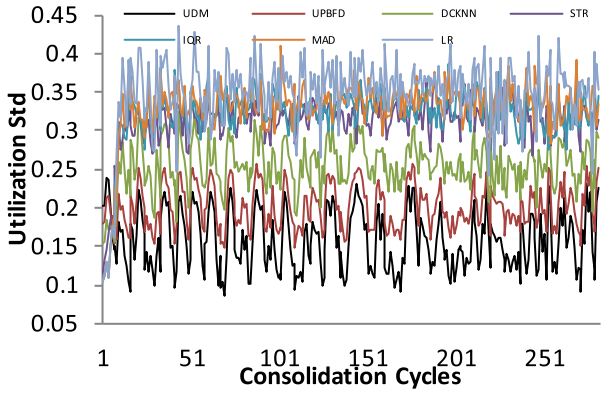


FIGURE 4. Standard deviation of utilization rate of resources of active hosts during each consolidation cycle.

introduces fresh workload 288 times and every 300 mS. This makes total execution time to 86400 mS. There are 288 total consolidation cycles during the simulation of each policy. To further evaluate the effectiveness of the policies under consideration, it is pertinent to show the number of machines being used, mean resource utilization on these machines and standard deviation of utilization at every cycle of consolidation. This will indicate how well each policy is utilizing the available resource for the execution of workload. For this purpose, for all the policies under consideration, the number of active hosts at each cycle of consolidation as obtained during the experiments has been graphed in Fig 2-4.

Lower the number of machines being used is better, as energy consumption increases with the increasing number of active machines. As indicated in Fig. 2, on an average, UDM utilizes 35.5 machines for the workload performance; UPBFD utilizes 44.5 machines whereas all other policies except UDM utilize about 60 machines for workload performance. Thus UDM utilizes 20% lesser machines than UPBFD's and 40% less machine than average of all the other policies.

Fig. 3 and 4 represents the mean and standard deviation of the utilization of resources of hosts by all the VM consolidation policies at each cycle of VM consolidation. The graph provided in Fig. 3 shows the average resource utilization of active processing machines at each cycle of VM consolidation as a ratio to the total capacity of machines. It is quite clear that average resource utilization by processing machines is much higher for UDM as compared to all other policies and it varies from as low as 0.70 to as high as 0.93. Average of this, for complete workload performance, remains at 0.82 for UDM, 0.71 for UPBFD, 0.67 for DCKNN, and around 0.50 for rest of the policies. Thus utilization driven model has 15% more utilization of active resources than UPBFD and about 50% more than all the policies. One more thing that can be noticed from the graph in Fig. 3 is that UDM achieved its saturation level in just a couple of cycles of consolidation whereas other consolidation policies took 8 to 10 cycles to achieve saturation. This shows the robustness of the consolidation by UDM.

Fig. 4 indicates the graph of the standard deviation of resource utilization on active hosts as a ratio of machines' total capacity. It indicates that when the workload was consolidated using UDM, the average of the standard deviation of utilization for all the cycles of consolidation stands at 0.15.

All other policies show the standard deviation of utilization about double of that of UDM. The value ranges from 0.08 to 0.23 for UDM with average at about 0.15 which is about 23% less than UPBFD. The above results showed that our proposed policy has utilized resources much better than the others. It uses less number of machines with a high utilization ratio. The high utilization ratio of UDM is due to the R function provided in (5), robustly converges the acceptable range according to current resource request making it possible to efficiently utilization of available resources of active hosts. This increase in mean utilization and decrease in the standard deviation of utilization makes UDM more energy and SLA violation savvy server consolidation policy.

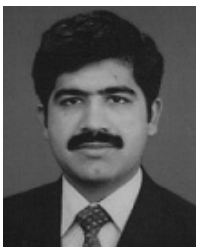
VI. CONCLUSION

Diverse applications of cloud computing has intensified the challenge of efficient utilization of computing resources. With the increased uncertainty of workload demand, maintaining quality of service along with degraded power consumption has become more challenging than ever. Prediction based approaches mostly rely on historical data for forecasting workload and adapting the system. However, with increased stochastic nature of workload demand, more emphasis needs to be given on robust adaption to the change in utilization. Utilization Driven Model focuses on varying utilization rates and adapts the utilization threshold accordingly. This approach enhances the elasticity of cloud resource provisioning, making the cloud more robust to change than traditional approaches. Experimental results have shown that UDM approach is more elastic; hence, it increases the utilization rate of active hosts to the optimal. The approach helps in maintaining quality of service with reduced energy consumption.

REFERENCES

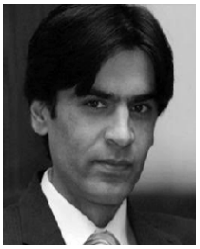
[1] J. Pan and J. McElhannon, "Future edge cloud and edge computing for Internet of Things applications," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 439–449, Feb. 2018.

[2] S. Wibowo and M. Wells, "Green cloud computing and economics of the cloud: Moving towards sustainable future," *GSTF J. Comput.*, vol. 5, no. 1, pp. 15–20, 2016.
[3] A. Khosravi and B. Rajkumar, "Energy and carbon footprint-aware management of Geo-distributed cloud data centers: A taxonomy, state of the art," in *Advancing Cloud Database Systems and Capacity Planning With Dynamic Applications*. Hershey, PA, USA: IGI Global, 2018, pp. 1456–1475.
[4] S. Kaushal and D. Gogia, B. Kumar, "Recent trends in green cloud computing," in *Proc. Int. Conf. Commun. Comput. Netw.*, 2019, pp. 947–956.
[5] L. A. Barroso and U. Holzle, "The case for energy-proportional computing," *Computer*, vol. 40, no. 12, pp. 33–37, Dec. 2007.
[6] M. Xu, W. Tian, and R. Buyya, "A survey on load balancing algorithms for virtual machines placement in cloud computing," *Concurrency Comput., Pract. Exper.*, vol. 29, no. 12, 2017, Art. no. e4123.
[7] S. Wu, K. Garg, and R. Buyya, "Service level agreement (SLA) based SaaS cloud management system," in *Proc. Int. Conf. Parallel Distrib. Syst. (ICPADS)*, 2015, pp. 440–447.
[8] L. Ismail and H. Materwala, "Energy-aware VM placement and task scheduling in cloud-IoT computing: Classification and performance evaluation," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 5166–5176, Dec. 2018.
[9] M. D. De-Assuncao, V. A. da-Silva, and R. Buyya, "Distributed data stream processing and edge computing: A survey on resource elasticity and future directions," *J. Netw. Comput. Appl.*, vol. 103, pp. 1–17, Feb. 2018.
[10] N. Kratzke, "A brief history of cloud application architectures," *Appl. Sci.*, vol. 8, no. 8, p. 1368, 2018.
[11] 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.
[12] A. Beloglazov and R. Buyya, "Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers," in *Proc. 8th Int. Workshop Middleware Grids, Clouds e-Sci.*, 2016, pp. 1–6.
[13] 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, Jun. 2019.
[14] Z. Li, "An adaptive overload threshold selection process using Markov decision processes of virtual machine in cloud data center," *Cluster Comput.*, vol. 22, no. 2, pp. 1–13, Mar. 2019.
[15] F. Farahnakian, T. Pahikkala, P. Liljeberg, J. Plosila, and H. Tenhunen, "Utilization prediction aware VM consolidation approach for green cloud computing," in *Proc. IEEE 8th Int. Conf. Cloud Comput.*, Jun./Jul. 2015, pp. 381–388.
[16] L. Salimian, F. S. Esfahani, and M.-H. Nadimi-Shahraki, "An adaptive fuzzy threshold-based approach for energy and performance efficient consolidation of virtual machines," *Computing*, vol. 98, no. 6, pp. 641–660, Jun. 2016.
[17] R. M. Abadi, A. M. Rahmani, and S. H. Alizadeh, "Self-adaptive architecture for virtual machines consolidation based on probabilistic model evaluation of data centers in Cloud computing," *Cluster Comput.*, vol. 21, no. 3, pp. 1711–1733, 2018.
[18] M. L. Chiang, Y. F. Huang, H. C. Hsieh, and W. C. Tsai, "Highly reliable and efficient three-layer cloud dispatching architecture in the heterogeneous cloud computing environment," *Appl. Sci.*, vol. 8, no. 8, p. 1385, 2018.
[19] T. Chen, Y. Zhu, X. Gao, L. Kong, G. Chen, and Y. Wang, "Improving resource utilization via virtual machine placement in data center networks," *Mobile Netw. Appl.*, vol. 23, no. 2, pp. 227–238, 2018.
[20] A. Abbasi and H. Jin, "v-Mapper: An application-aware resource consolidation scheme for cloud data centers," *Future Internet*, vol. 10, no. 9, p. 90, 2018.
[21] G. Han, W. Que, G. Jia, L. Shu, and A. Jara, "An efficient virtual machine consolidation scheme for multimedia cloud computing," *Sensors*, vol. 16, no. 2, p. 246, 2016.
[22] A. Marotta and S. Avallone, "A simulated annealing based approach for power efficient virtual machines consolidation," in *Proc. IEEE 8th Int. Conf. Cloud Comput.*, Jun./Jul. 2015, pp. 445–452.
[23] S. Joshi and S. Kaur, "Cuckoo search approach for virtual machine consolidationin cloud data centre," in *Proc. ICCCA*, 2015, pp. 683–686.
[24] A. Fatima, N. Javaid, A. A. Butt, T. Sultana, W. Hussain, M. Bilal, M. A. R. Hashmi, M. Akbar, and M. Ilahi, "An enhanced multi-objective gray wolf optimization for virtual machine placement in cloud data centers," *Electronics*, vol. 8, no. 2, p. 218, 2019.
[25] Q. Zheng, J. Li, B. Dong, R. Li, N. Shah, and F. Tian, "Multi-objective optimization algorithm based on bbo for virtual machine consolidation problem," in *Proc. Int. Conf. Parallel Distrib. Syst.*, 2015, pp. 414–421.
[26] 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.
[27] H. Li, G. Zhu, C. Cui, H. Tang, Y. Dou, and C. He, "Energy-efficient migration and consolidation algorithm of virtual machines in data centers for cloud computing," *Computing*, vol. 98, no. 3, pp. 303–317, Mar. 2016.
[28] H. Li, G. Zhu, Y. Zhao, Y. Dai, and W. Tian, "Energy-efficient and QoS-aware model based resource consolidation in cloud data centers," *Cluster Comput.*, vol. 20, no. 3, pp. 2793–2803, 2017.
[29] M. A. Khan, A. Paplinski, A. M. Khan, M. Murshed, and R. Buyya, "Dynamic virtual machine consolidation algorithms for energy-efficient cloud resource management: A review," in *Sustainable Cloud and Energy Services*. Cham, Switzerland: Springer, 2018, pp. 135–165.
[30] 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.
[31] 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, 2012.
[32] *All Published SPECpower_ssj2008 Results*. Accessed: Nov. 23, 2019. [Online]. Available: https://www.spec.org/power_ssj2008/results/power_ssj2008.html
[33] Z. Zhou, H. Zhigang, and L. Keqin, "Virtual machine placement algorithm for both energy-awareness and SLA violation reduction in cloud data centers," *Sci. Program.*, vol. 2016, p. 15, Mar. 2016.



HAMMAD UR-REHMAN QAISER received the B.S. degree in computer science from the Institute of Management Sciences, Lahore, Pakistan, in 2006, and the M.S. degree in computer science from the Lahore University of Management Sciences, in 2010. He is currently pursuing the Ph.D. degree in computer science with the Wuhan University of Technology, Wuhan, China. From 2006 to 2008, he worked as a Research Assistant. Since 2011, he has been working with the Government of Pakistan, Department of Information and Broadcasting. He has also been a visiting Faculty Member with Bahria University, Islamabad. His primary area of interest includes parallel and distributed, mobile edge computing, and the Internet of Things.

GAO SHU received the Ph.D. degree from the Wuhan University of Technology, Wuhan, China, in 2007. She is currently a Professor with the School of Computer Science, Wuhan University of Technology. Her research interests include distributed computing, data analysis, and their application in intelligent transportation system.



ASAD WAQAR MALIK the Ph.D. degree in parallel and distributed simulation/systems from the National University of Sciences and Technology (NUST), Islamabad, Pakistan, in 2012. He is currently an Assistant Professor with the School of Electrical Engineering and Computer Science, NUST. Besides, he is also working as a Senior Lecturer with the Department of Information Systems, Faculty of Computer Science and Information Technology, University of Malaya, Malaysia. His primary areas of interest include distributed simulation, cloud/fog computing, and the Internet of Things.