IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 23 April 2023, accepted 5 May 2023, date of publication 15 May 2023, date of current version 22 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3276292

RESEARCH ARTICLE

Genetic-Based Virtual Machines Consolidation Strategy With Efficient Energy Consumption in Cloud Environment

MOHAMMED RADI [1], **ALI A. ALWAN** [2], **AND YONIS GULZAR** [3]
[1]Department of Computer and Information Sciences, Faculty of Computers and Information Technology, Al-Aqsa University, Gaza 4051, Palestine
[2]School of Theoretical and Applied Science, Ramapo College of New Jersey, Mahwah, NJ 07430, USA
[3]Department of Management Information Systems, College of Business Administration, King Faisal University, Al-Ahsa 31982, Saudi Arabia

Corresponding authors: Mohammed Radi (moh_radi@alaqsa.edu.ps) and Yonis Gulzar (ygulzar@kfu.edu.sa)

This work was supported by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia, under Project GRANT3374.

ABSTRACT In cloud computing environments, virtualization is used to share physical machine (PM) resources among multiple users by creating virtual machines (VMs). Running the PM consumes a large amount of energy. Additionally, the PM will be overloaded when the demand for resources exceeds the PM capacity. This overload on the PM leads to violations of Service Level Agreements (SLAs). Dynamic VM consolidation techniques use live migration of VMs to optimize resource utilization and minimize energy consumption. However, excessive migration of VMs impacts negatively the application performance due to the incurred overhead at the runtime. This paper presents a modified genetic-based VM consolidation (MGVMC) strategy that aims to replace VMs in an online manner taking into account energy consumption, SLA violations, and the number of VM migrations. The MGVMC strategy utilizes the genetic algorithm to migrate VMs to the appropriate PM in a way that minimizes the number of over-utilized and under-utilized physical machines (PMs) as low as possible. The performance of the MGVMC strategy was evaluated using the CloudSim Plus framework with a large number of VMs and workload traces from the PlanetLab platform. The experimental results revealed that the MGVMC strategy achieved a significant improvement in energy consumption, SLA violations, and the number of VM migrations compared to other recent approaches. These results demonstrate the effectiveness of the MGVMC strategy in optimizing VM consolidation in the cloud environment.

INDEX TERMS Cloud computing, dynamic VM consolidation, energy consumption, genetic algorithm, resource utilization, service level agreements.

I. INTRODUCTION

The cloud computing model provides end users with on-demand services based on virtualized resources. Virtualization is used to create and assign various virtual machines (VMs) to a physical machine (PM), thus minimizing the costs for the acquisition of servers. User applications and/or workloads are highly non-linear in clouds [1]. Thus, it results in fluctuating the overall performance of the system, dramatic raise in power consumption, and deterioration in the quality of services (QoS) [2]. For example on Azure and Alibaba cloud platforms, the average CPU utilization is quite low (less than 20%) [3], while on the IBM cloud platform, the average CPU utilization and memory usage vary between 17.76% and 77.99%, respectively. Meanwhile, the average CPU Utilization and memory utilization on Amazon Web Services (AWS (cloud platform are 25% and 28.63%, respectively. Apparently, the Google Cloud Platform (GCP) has recorded on average over 57% CPU Utilization, and on average 20% memory utilization [4]. The CPU and memory consumption in most cloud service providers does not exceed 60% while great resources in cloud data centers are not efficiently utilized. Cloud service providers aim to ensure QoS and deliver efficient and reliable services to end-users [5], [6]. Therefore,

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

CSPs should avoid any violations of the performance requirements while providing these services to their clients [7]. Cloud computing data centers are equipped with many expensive and high-performance servers to provide on-demand services to end users [8]. High-performance servers and the attached cooling systems consume a tremendous amount of power that also needs electricity. Moreover, even applications with limited workloads consume a significant amount of energy on high-performance servers. On other hand, in cloud systems with limited servers and high workloads, the resource demand for virtual machines may result in the over-utilization of physical machines [9].

Most of the recent studies found in the literature argued that energy consumption in data centers is one of the crucial challenges that should be addressed. A number of these studies suggested efficient resource management could be a solution to reducing the overall energy consumption of these data centers [2], [9], [10], [11]. Several attempts based on VM consolidation and VM placement techniques have been made aiming at reducing power consumption by exploiting efficient resource utilization through consolidating the active virtual machines to the minimum possible physical machines [10], [11]. The idea of VM consolidation relies on dynamically migrating the virtual machine from one physical machine to a more suitable physical machine [11]. This means physical machines that have no active VMs are turned to sleep mode. Any VM consolidation technique should consider resource utilization, and energy consumption while ensuring the QoS and minimizing the number of VM migrations [9]. While the VM placements techniques attempt to place the VMs of different applications on a set of PMs. Efficient virtual machine placement (VMP) minimizes energy consumption, reducing the resource wastage of PMs, and maximizing the QoS offered to cloud users [12]. A set of strategies based on constraint programming, linear programming, and evolutionary approaches have been introduced in the paper to address the issues related to VM consolidation.

This paper presents a modified Genetic-based VM consolidation (MGVMC) strategy for optimizing the consolidation of virtual machines dynamically. The main purpose of the proposed approach is to ensure minimum energy consumption and prevent SLA violations. Besides, the proposed solution also attempts to minimize the total number of VM migrations by using a modified genetic algorithm to reduce the number of under-utilized and over-utilized physical machines (PMs) as low as possible. The idea of migrating VMs from under-utilized PMs is very beneficial as it generates PMs with no VMs and switches those PMs to sleep mode, which results in a significant improvement in energy consumption. Additionally, by migrating VMs from over-utilized PMs, until they reach a state of fair utilization, the SLA violations can be voided. The experimental results show that the proposed approach MGVMC is effective in lowering energy consumption, avoiding SLA violations, and reducing the number of VM migrations compared to other approaches.

This paper's main contributions can be summarized as follows:

1) Propose a novel approach to solving the VM Consolidation problem known as the Modified Genetic-based VM Consolidation (MGVMC) strategy, considering various critical parameters such as energy consumption, SLA violations, and the number of VM migrations.
2) Develop a novel technique to accurately identify over- and under-utilized PMs in real-time based on Moving Range (MR), which is used to identify upper and lower thresholds for current workloads based on a continuous analysis of utilization values over time.
3) Propose a Modified Genetic Algorithm (MGA) that places VM in the optimal location, including a modified Initial Population Generation Strategy, implements a problem-specific crossover strategy, develops a problem-specific mutation strategy, and raises new fitness functions based on a weighted sum of multiple objective functions.
4) We implement the MGVMC strategy and validate its efficiency compared to other recent approaches. We conduct extensive experiments using the CloudSim Plus framework with a large number of VMs and workload traces from the PlanetLab platform.

The remainder of the paper is organized as follows. Section II reviews and examines the related works relevant to the VM consolidation problem. The system architecture and the problem formulation are described in Section III. Section IV illustrates the detailed steps of the proposed VM consolidation algorithm. The experimental setup and results obtained from the designed scenarios are given in Section. 5. Section VI concludes the paper and draws some interesting future research directions.

II. RELATED WORK

The consolidation of virtual machines is a crucial research topic in cloud platforms, which aims to optimize energy consumption, avoid service level agreements (SLAs) violations, and reduce the number of VM migrations [9], [10], [11]. The process of VM consolidation involves four main phases, namely: detection of over-utilized hosts, detection of under-utilized hosts, VM migration, and VM placement. VM consolidation is an NP-hard optimization problem that can be approached in several ways, including linear programming [13], heuristics [14], constraint programming [15], and genetic algorithms [16]. The work introduced in [13]addresses the issue of VM consolidation in cloud computing. They use a linear programming-based heuristic approach to solve the VM consolidation problem in Virtualized Data Centers. The idea of the proposed solution exploits the data pre-processing method to characterize the workload traces and derive parameters for the decision model. The work presented in [14] also tackled the problem of VM consolidation by proposing a heuristic-based mechanism, called

(EEHVMC). The proposed approach aims at reducing power consumption and avoiding the violation of the SLA. The first step of EEHVMC is classifying the host into three main classes based on the adaptive utilization threshold. These classes are Host Over-Loaded (HOL), Host Medium-Loaded (HML), and Host Under-Loaded (HUL) machines. Then, reallocates the virtual machines from one physical host to another to minimize the consumption of energy. The work contributed by [15]highlights the issue of VM consolidation in federated cloud data centers with the intention of balancing between power consumption and performance requirements. The idea of the proposed approach epmloys the constraints expressed through SLA to reallocate the VM using Constraint Programming (CP) that leads to identify the best possible placement of these VMs. Furthermore, the issue of VM consolidation on cloud platforms has been addressed by the work in [16]. They introduce an approach that utilizes the constraint programming concept to resolve the virtual resources allocation problem by exploiting the users' QoS requirements, and the cost of virtual cloud resources. However, the experimental results showed that the proposed model which is built based on linear programming and Constraint Programming is ineffective when the number of variables such as the number of PM and number of VM increases.

Various approximation and heuristic-based algorithms have been introduced in the literature aiming at resolving the issue of VM consolidation in a cloud computing environment. The work in [17]proposes a novel adaptive heuristic-based algorithm that aims to maintain the requirements of the SLA violations while minimizing the energy usage at the data centers. The suggested method makes use of past data to analyze the resource utilization of the virtual machines and create a dynamic consolidation model for them. The proposed approach found the overloaded and underloaded hosts using different statistical methods of historical data collected during the lifetime of VMs. While the Minimum Migration Time, Random Choice, and Maximum Correlation Policies are suggested to address VM placement and selection. The Power Aware Best Fit Decreasing (PABFD) algorithm, a variation of the Best Fit Decreasing (BFD) algorithm, is then used to allocate a host with the lowest required power consumption. Moreover, a heuristic approach called the energy efficiency heuristic using virtual machine consolidation is proposed in [18]. The proposed approach attempts to migrate the VM from one physical host to another based on two setting thresholds that lead to reducing the energy consumption in the cloud. The threshold is defined using inter-quartile range (IQR) based on current CPU and memory utilization, after that and based on the threshold host machines are classified into Host Over-Loaded (HOL), Host Medium-Loaded (HML), and Host Under-Loaded (HUL) machines. The VMs are migrated from the HOL and HUL to HML and put the inactive host into power-saving mode. The authors claim that the results of this approach minimize power consumption and reduce SLAV but this approach may overload the HML machines and introduce A performance bottleneck problem.

The issue of VM consolidation in a heterogeneous cloud environment has been highlighted by the work in [19]. A VM consolidation strategy has been proposed to concentrate on minimizing energy consumption and reducing migration costs via dynamic VM consolidation. The proposed strategy uses a score function based on a migration cost estimation method and an upper bound estimation method for maximal saved power. Moreover, it improved the grouping genetic algorithm (IGGA) to optimize the consolidation score. The work presented in [20]proposes an improved KnEA (Knee Point-Driven Evolutionary Algorithm), named EEKnEA (Energy-Efficient KnEA), to solve the virtual machine placement optimization problem. Other works consider the energy consumption in both the servers and the communication network in the data center. For instance, the work in [21] suggests using a genetic algorithm for a virtual machine placement problem, while the work in [22]proposes using a Decrease and Conquer Genetic Algorithm (DCGA) to keep the solution quality while decreasing the problem size and the number of VM migrations.

A recent approach based on normalization-based VM consolidation (NVMC) strategy is proposed in [4]. The approach strives to optimize the consolidation process of virtual machines (VMs) in cloud platforms. The main purpose of the NVMC approach is to conserve energy consumption, sustain service level agreement (SLA) violations, and reduce the number of VM migrations. To achieve this, the NVMC strategy utilizes resource parameters of VMs and hosts to identify over-utilized hosts and the cumulative available-to-total ratio to identify under-utilized hosts. These values are then normalized to determine the appropriate placement of VMs. In addition, the NVMC approach includes the use of threshold values to detect over-utilized hosts and optimize the consolidation process. A utilization-aware VM placement (UAVMP) technique to optimize the placement of virtual machines (VMs) during the consolidation process in cloud platforms is proposed in [6]. The UAVMP technique selects the target host for VMs based on the host's utilization and resource skewness. This approach aims to effectively balance the workload of hosts and improve the efficiency of the consolidation process. Last but not least, the work contributed by [5] has proposed the Hybrid VM allocation and placement policy (HVMAP) for VM consolidation in cloud data centers. The HVMAP strategy combines an improved host overload detection algorithm with migration control to identify overloaded hosts and the PEBFD placement algorithm to determine the appropriate destination host for migrating VMs. The proposed approach attempts to ensure low energy consumption while maintaining the service level agreement (SLA) violations during the consolidation process.

III. PROBLEM DEFINITION

A key element of cloud computing systems is the consolidation of virtual machines (VMs), which are hosted by physical machines (PMs) that employ virtualization technology. Various tools are provided in the cloud system to facilitate the process of accessing the cloud resources and services by the end users and submitting their applications to the cloud service provider (CSP) through the internet. Typically, each application requires a certain number of VMs, each of which has specific desirable resource requirements. The efficient utilization of PM resources is essential to ensure minimizing both running costs and power consumption when these PMs are running.

Therefore, the VM consolidation technique is a crucial process that attempts to assign all VMs in a way that guarantees involving a minimum number of PMs using live migration. The live migration process involves moving VMs from under-utilized PMs to other PMs and switching unutilized PMs to sleep mode. However, excessive VM migration and the over-utilization of some PM resources can violate the service level agreements (SLAs) and negatively impact the performance of the system, and delay the response time for the user application.

To address these issues, live migration can also be used to migrate some VMs from overloaded PMs to other PMs. Based on the resource requirements and capacities of the VMs, and the current utilization of the PMs, the decision of which VMs to migrate and which PMs to use as destination hosts must be carefully considered. This process, known as the placement phase, is a critical aspect of VM consolidation techniques.

A. VIRTUAL MACHINE

Virtualization in cloud computing involves submitting a set of user applications to the cloud provider, and each application requires a set of virtual machines (VMs). Each VM, represented as VM_i is characterized by its CPU and memory requirements, denoted as $\mathrm{VM}_i = \{VM_i^C, VM_i^m\}$ for CPU requirements and memory requirements respectively. A VM_i is considered active if it is currently executing an application on any PM_j, and inactive otherwise. This activity status of VM_i on a PM_j can be mathematically represented using the following equation:

$$VMS_{ij} = \begin{cases} 1, & if\, VM_i hosted\ at\ PM_j \\ 0, & Otherwise \end{cases} \tag{1}$$

B. PHYSICAL MACHINE

In the cloud computing environment, there are N physical machines (PMs) with varying resource capacities and power consumption characteristics. Each PM, denoted as PM_i, has a CPU capacity represented by PM_i^C and a memory capacity represented by PM_i^m . A PM is considered active if there is at least one actively functioning virtual machine (VM) placed on it, and inactive if there are no active VMs present. This activity status of PM_i can be mathematically represented using the following equation:

$$PMS_i = \begin{cases} 1 & if\ VM_j\ are \text{ hosted to } \mathrm{PM}_i \\ 0 & \text{Otherwise} \end{cases} \tag{2}$$

Algorithm 1 Modified Genetic based VM consolidation (MGVMC)

Input:the set of N physical machines, a set of m virtual machines, the activity status of VM, CPU capacity of VM, and CPU capacity of PM.
Output: Upper and lower thresholds
1. Upper and lower thresholds
2. Physical machine classification
3. The Modified Genetic Algorithm (MGA)
4. **Do the migration based on the**best individual
5. For i=1 to N do
6. If $PMS_i = 0$
7. switching those PMs to sleep mode

IV. MODIFIED GENETIC-BASED VM CONSOLIDATION (MGVMC) STRATEGY

Optimization of resource utilization and reduced energy consumption are some of the main challenges faced by cloud service providers. This is accomplished by migrating VMs from underutilized and overutilized PMs by using VM consolidation. It is important to note, however, that a large number of migrations may negatively impact the system's performance. This requires balancing the migration between performance and energy in the consolidation process. We propose a method called Modified Genetic-based VM consolidation (MGVMC) for merging VMs that involves migrating them to another PM and minimizing energy consumption, resource waste, and the number of migrations. The proposed MGVMC strategy comprises of the main consolidation algorithm (Algorithm 1), which calls three sub-algorithms. Algorithm 2 is responsible to identify the upper and lower thresholds regarding CPU utilization. Algorithm 3 attempts to classify the PM based on the upper and lower thresholds into three classes (over-utilized, under-utilized, and fair-utilized). While Algorithm 4 is responsible to find the best placement of VMs based on the proposed modified genetic algorithm. The detailed steps of the proposed Modified Genetic-based VM consolidation algorithm are depicted in Algorithm 1. The algorithm starts in step 1 by setting the upper and the lower thresholds of CPU utilization by calling Algorithm 2. Then in step 2, the physical machine classification is performed by calling Algorithm 3. This is followed by performing the Visual machine placement using the Modified Genetic Algorithm as stated in step 4. The MGA is responsible to find an optimal solution for the VM placement problem. In step 4, the migration process is performed based on the best individual. After that, for each physical machine, if the physical machine allocated no VM then the activity status of PMj, =0, then switch those inactive PMs to sleep mode (steps 5-7).

A. IDENTIFYING UNDER-UTILIZED PM AND OVERUTILIZED PM

The utilization of physical machines (PMs) at any given time can be challenging in systems with dynamic workloads and unpredictable workload patterns. In these cases, manually setting upper and lower thresholds for PM utilization and

Algorithm 2 Upper and lower thresholds

Input:the set of N physical machines, a set of m virtual machines, the activity status of VM, CPU capacity of VM, and CPU capacity of PM.
Output: Upper and lower thresholds
1. For i=1 to N do
2. For j=1 to m do
3. $\rightarrow PMU_i^C \sum_{1}^{k} \frac{VMS_{ji} \times VM_i^C}{PM_i^C}$
4. **AvgPMU**$\rightarrow$ [[**space**]] $\sum_{\mathbf{i=1}}^{\mathbf{n}} \frac{PMU_i^C}{\mathbf{N}} \forall PM_j \in PM$
5. $\rightarrow \sigma \sqrt{\frac{\sum_{i=1}^{n} \left(PMU_j^C - \textbf{AvgPMU}\right)^2}{n}}$
6. UCL$\rightarrow$ **AvgPMU+3**
7. LCL$\rightarrow$ **AvgPMU+3**
8. Uth$\rightarrow$ (1 − £) UCL
9. Lth$\rightarrow$ (£ − 1) LCL
10. Return Uth, Lth

classifying PMs with usage above or below these thresholds as over- or underutilized is not a suitable approach. This is due to the fact that a PM's workload can vary significantly over time, fixed thresholds may not be accurate indicators of the PM's current use. To address this challenge, we propose a new technique called Moving Range (MR) that automatically determines the appropriate utilization threshold for PMs in a dynamic environment. MR is a statistical method with a focus on identifying upper and lower thresholds for current workloads based on a continuous analysis of utilization values over time. By doing so, we can more accurately identify over and under-utilized PMs in real-time, and adjust to changes in workload on the fly. MR is an effective approach for managing the PMs in systems with dynamic workloads by dynamically identifying utilization thresholds. Algorithm 2 is able to dynamically computes the upper and lower thresholds regarding CPU utilization. The algorithm input comprises of the set of N physical machines, the set of m virtual machines, the activity status of each VM, the CPU capacity of VMs, and the CPU capacity of PMs. Initially, the algorithm computes the PMU_j^C as CPU utilization for each PM as described in steps (1-3). This is followed by calculating the average of PMU_j^C and the standard deviation as shown in steps (4 - 5). In steps 6-7, the Upper Control Limit (UCL) and Lower Control Limit (LCL) are calculated. It is important to note that the value of £directly influences energy consumption and SLA violation. For that reason, we have carried out many experiments to examine the effect of £on both energy consumption and SLA violation and identify a value of s that balances both energy consumption and SLA violation. Finally, in steps (8-9) we attempt to find the calculated Upper Threshold (Uth) and Lower Threshold (Lth). After completing all steps of the algorithm, the upper bound Uth and the lower bound Lth thresholds are returned (step 10).

Algorithm 3 is responsible to classifies the physical machines (PMs) into three main categories based on their CPU utilization: Over-utilized, Under-utilized, and

Algorithm 3 Physical machine classification

Input:Inputs: the set of N physical machines, This activity status of PM, CPU utilization.
Output:Outputs: Number of Over-utilized, number of Under-utilized,
1. NUU=0, NOU=0, NFU=0;
2. For i=1 to N do If$PMSi = 1$
If (PMU_j^C >= Uth)
3. PM_i.Class= "Over-utilized", NOU++
4. Else if (PMU_j^C <= Lth)
5. PM_i.Class =" Under-utilized", NUU++
6. Else
7. PM_i.Class = "Fair-utilized", NFU++

Fair-utilized. The idea of classification in Algorithm 3 relies on exploiting the upper and lower thresholds of CPU utilization. To do this, a comparison is made between the PM's current CPU utilization and upper and lower thresholds. Therefore, PMs with a CPU utilization above the upper threshold are classified as Over-utilized, while PMs with a CPU utilization below the lower threshold are classified as Under-utilized. The PMs that have a CPU utilization between these two thresholds are considered fair-utilized. Based on these categories, the algorithm then counts how many PMs are in each category. The steps of Algorithm 3 are as follows.

B. VM PLACEMENT USING A MODIFIED GENETIC ALGORITHM

The VM placement problem is an optimization problem that can be solved using Genetic Algorithm (GA). In order to improve the efficiency of the GA that has been used in this work, we develop a modified genetic algorithm (MGA) that places the VM in the optimal location. In this section, we introduce the details of the proposed Modified Genetic Algorithm (MGA). We present first the encoding and decoding method, then the proposed initial population generation and parent selection strategies, followed by the proposed fitness functions, crossover strategy, and mutation strategy. A detailed description of the proposed MGA is given in algorithm 4.

1) ENCODING AND DECODING METHOD

The VM placement solution is encoded in the form of chromosomes. Each chromosome is represented as a gene order, with each gene representing a PM where the VM will be placed.

Figure 1 shows an example of Virtual machines placed to physical machines and their equivalent chromosome when the number of visual machines is 10 and the number of physical machines is 4. The V_1, V_2, V_3 are placed in M_1, while V_4, V_5 are placed in PM_2, and V_6, V_7, V_8, V_9 are placed in PM_3 finally V_{10} is placed in PM_4.

2) INITIAL POPULATION GENERATION STRATEGY

Genetic Algorithms (GAs) are well regarded in optimization for their ability to greatly depend on the characteristics of

Virtual machines	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	V_{10}
Physical machines	PM_1			PM_2		PM_3				PM_4

VM	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
Chromosome	PM_1	PM_1	PM_1	PM_2	PM_2	PM_3	PM_3	PM_3	PM_3	PM_4

FIGURE 1. Example of VM placed to PM and its equivalent chromosome.

their initial populations. The GA will likely be able to find an optimal or near-optimal solution more efficiently if the fitness of the individuals in the initial population is high. For this reason, we have created a modified Initial Population Generation Strategy for use in a Modified Genetic Algorithm (MGA). Using this strategy, the VMs are divided into two groups. The first group contains the VMs hosted by underutilized or overutilized PMs, and the second group contains the other VMs. The VMs in the second group remain in the same place in the initial population to generate fit individuals. For VMs in the second group, we employ a random search technique to maintain diversity within the initial population In this case, we do not get the initial population, so we use a random search for all VMs. This approach has greatly improved MGA efficiency in certain cases.

3) PARENT SELECTION STRATEGY

In the field of optimization, parent selection is a crucial component of Genetic Algorithms (GAs). It involves selecting a subset of individuals from the current population, known as parents, to produce offspring for the next generation through mating and recombination. The efficiency and convergence rate of the GA can be significantly impacted by the method used to select parents. In the Modified Genetic Algorithm (MGA), we employ a roulette wheel selection approach, in which the probability of choosing an individual as a parent is directly proportional to their fitness. This means that individuals with higher fitness values have a higher probability of being selected as parents. This approach seems to be very effective in guiding the MGA toward optimal or near-optimal solutions.

4) VM PLACEMENT CROSSOVER STRATEGY

The Modified Genetic Algorithm (MGA) uses crossover to generate new offspring solutions by combining elements from two selected parent solutions. The MGA implements a problem-specific crossover strategy called "VM placement" which involves selecting the parent with the lowest fitness and choosing the PMs from the underutilized group and PMs from the overutilized group. The VMs (virtual machines) located on these PMs are then placed on the PMs of the second parent, with the goal of creating offspring solutions that exhibit good VM placements. This process is executed in two main steps. First, select the parent with the lowest fitness and identify the VMs located on the chosen PM. Second, placing those VMs on the PMs of the second parent. Apparently, the MGA uses crossover as a means of generating new offspring solutions that incorporate elements from multiple parent solutions, in an effort to improve the quality and effectiveness of the solutions being generated.

5) VM PLACEMENT MUTATION STRATEGY

The next step in GA is the use of a mutation strategy aiming at identifying certain genes from parents and changing them to keep diversity in the generation. In this work, the proposed MGA strategy develops a problem-specific mutation strategy (VM placement), in which the PM from the underutilized group and PMs from the overutilized group are selected and their VMs are placed to the highest available in the fair group that satisfied the constraints of that VM.

6) FITNESS FUNCTION

Our main goal in this work is to reduce energy and optimize resource utilization, which is a multi-objective problem. The first objective is to reduce energy consumption, which is achieved by minimizing under-utilized PMs. The second objective is to optimize resource utilization, which is accomplished by minimizing over-utilized VM. GA is customized to address multi-objective problems by using specialized fitness functions. For that reason, we develop new fitness functions based on a weighted sum of multiple objective functions. The proposed fitness function determines how the solution fits the addressed challenge. This is reflected in the fitness function by considering two objective functions the number of over-utilized (NOU) PMs and the number of under-utilized (NUU) PMs. So that the problem is converted to a single objective problem with a scalar objective function by assigning a weight Wi to each objective function. In this case, we have chosen equal weight for both NUU and NOU. The following equation computes the fitness value of each individual, x, in the population.

$$MinF(x) = W1NUU(x) + W2NOU(x) \tag{3}$$

Where,

NUU(x): number of under-utilized PMs.

NoU(x): number of over-utilized PMs.

And $\sum$ wi =1.

The steps of Algorithm 4 are explained as follows. The first step is the initial population is generated using the initial population strategy described in section IV-B2. then compute the fitness function of individuals using equation 3. In the loop part the parent selection, Crossover Strategy, and Mutation Strategy are applied using the strategies described in subsections IV-B3,IV-B4,IV-B5 respectively. After that compute the fitness of each individual and check whether the fitness of the fittest individual is enough. If so, return the fittest individual, otherwise, go to step 3.

Figure 2 Depicts the Process Flow of the Proposed MGA Algorithm

V. EXPERIMENT SETUP AND RESULTS

This section presents the details of the experimental setup and highlights the results of the experiments performed to

Algorithm 4 The Modified Genetic Algorithm (MGA)

Input: the set of N physical machines, a set of m virtual machines, the activity status of VM, the activity status of PM, the CPU capacity of VM, CPU capacity of PM.
Output: best VMs placements.
1. Generate initial population
2. compute the fitness of individuals
3. do
4. Parent selection
5. VM placement Crossover Strategy
6. VM placement Mutation Strategy
7. Compute the fitness of individuals
8. while the fitness of the fittest individual is not high enough
9. return the best individual

assess and evaluate the modified Genetic-based VM consolidation (MGVMC) strategy for improving energy efficiency and quality-of-service (QoS) for cloud service providers. In this section, we attempt to investigate the performance of the proposed strategy with respect to the average energy consumption in kWh by physical machines across different VM consolidation strategies, the average number of VM migrations across different VM consolidation strategies, and the average SLA violation across different VM consolidation strategies. We argue that these are the most crucial performance metrics that have been used by many relevant studies in the literature [9], [10], [11], [23]. The experimental setup of the simulation has been explained in subsection V-A. The metrics considered for the performance evaluation have been described in subsection V-B. Lastly, the results of the experiment have been presented and discussed in subsection V-C.

A. EXPERIMENTAL SETUP

To evaluate the performance of the proposed approach we extended the CloudSim Plus framework [24] by adding VM consolidation. The simulations are carried out on a Lenovo laptop with a core i7 processor, 8 GB of RAM, a Windows 10 Pro operating system, and 250 GB SSD. All experiments have been implemented at a Datacenter having 800 heterogeneous PMs, each host has 4 G RAM and 1T storage. Half of the PMs are HP Proliant-ML110-G4 with Intel Xeon 3040 and 1860(MHz) Core frequency and the other half are HP Proliant-ML110-G5 with Intel Xeon 3075 and 2660 (MHz) Core frequency. The characteristics of the VMs workloads are obtained from real system traces obtained from the CoMon [25]project, a monitoring system for the PlanetLab testbed [10]. In which the used VMs includes four categories, the first category with 500 MIPS CPU and 613 MB RAM, the second category with 1000 MIPS CPU and 1740 MB RAM, the third with 2000 MIPS CPU and 1740, MB RAM, and the fourth category with 2500 MIPS CPU and 870 MB RAM. In our MGA usage, we have adopted standard parameter configurations commonly used in the field. We have specifically selected a population size of 64 individuals to ensure a rapid convergence rate and mitigate the risk of becoming trapped in local optima. Moreover, the MGA stops running after 50 generations with no further improvement

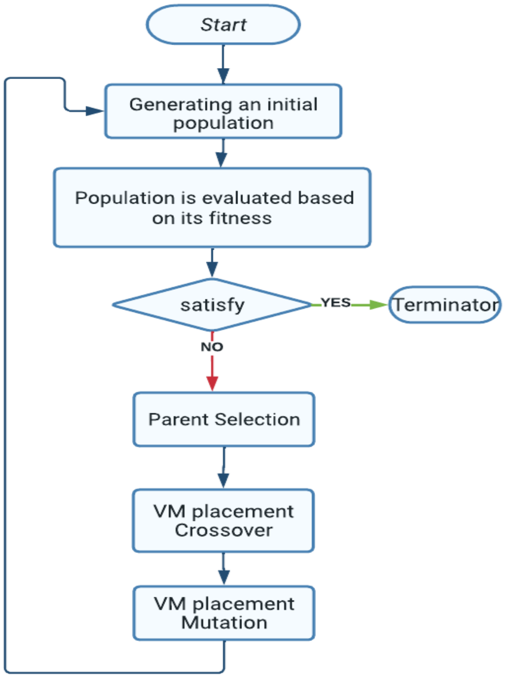

FIGURE 2. The process flow of the MGA algorithm.

and sets a maximum iteration limit of 999. Moreover, the MGA implements a problem-specific crossover strategy and problem-specific mutation strategy described in section IV.

For evaluation purposes, we used the PlanetLab workloads which are widely used for evaluating the performance of dynamic VM consolidation strategies reference. The (W1–W6) contain date-wise traces of execution (representing utilization of resources at several instances) on a cloud platform, as obtained for different days of March and April (2011). Table 1 displays the used workload and number of VMS.

In order to evaluate the efficiency and the performance of our proposed approach to replace VMs in an online manner taking into account energy consumption, SLA violations, and the number of VM migrations, we used the CloudSim Plus framework [24]. We have extended the CloudSim Plus framework by adding Algorithms 1,2,3 and 4. The simulations are carried out on a Lenovo laptop with a core i7 processor, 8 GB of RAM, a Windows 10 Pro operating system, and 250 GB SSD. All experiments have been implemented at a Datacenter having 800 heterogeneous PMs, each host has 4 G RAM and 1T storage. Half of the PMs are HP Proliant-ML110-G4 with Intel Xeon 3040 and 1860(MHz) Core frequency and the other half are HP Proliant-ML110-G5 with Intel Xeon 3075 and 2660 (MHz) Core frequency. The characteristics of the VMs workloads are obtained from real system traces obtained from the CoMon [25] project, a monitoring system for the PlanetLab testbed [26], in which the used VMs include four categories, the first category with 500 MIPS CPU and 613 MB RAM, the second category with 1000 MIPS CPU and 1740 MB RAM, the third with 2000 MIPS CPU and 1740, MB RAM, and the fourth category with 2500 MIPS CPU and 870 MB RAM. For

TABLE 1. workload and number of VMS.

Workload	Date	No of VMs
W1	2011/03/03	1052
W2	2011/03/06	898
W3	2011/03/22	1516
W4	2011/04/03	463
W5	2011/04/09	1358
W6	2011/04/20	1033

evaluation, we used the PlanetLab workloads which are widely used for evaluating the performance of dynamic VM consolidation strategies reference [9], [10], [11], [23]. The PlanetLab workloads comprises of 6 workloads (W1–W6) contain date-wise traces of execution (representing utilization of resources at several instances) on a cloud platform, as obtained for different days of March and April (2011). Table 1 outlies the details of the workloads and the number of VMS used in this work.

B. METRICS FOR PERFORMANCE EVALUATION

The parameters to evaluate our approach are the number of energy consumption (EC), the number of VM migrations (VMM), and the SLA violations (SLAV). Most of the previous studies conducted to improve the energy efficiency and quality-of-service (QoS) for cloud service providers report their results using these metrics literature [9], [10], [11], [23]. These performance metrics are explained below.

- **Energy consumption (EC)**: represents the energy consumed by physical machines while executing user workloads. The energy consumption of the PM is computed as shown in Equation 4.

$$EC\,(PM_i) = EPM_i^{idle} \times EPM_i^{full} + \left(1-EPM_i^{idle}\right) \times EPM_i^{full} \times PMU_i^{C} \quad (4)$$

Where EPM_i^{idle} is the fraction of energy consumed when PMj is idle;EPM_i^{full} is the energy consumption of a physical machine PM_i when it is fully utilized; and PMUi is the CPU utilization of PM_i.

Based on the above formula, the total energy consumption is computed as illustrated in Equation 5:

$$EC = \sum\nolimits_{i=1}^{m} E(PM_i) \quad (5)$$

- **The number of VM migrations (VMM)**: the VM migration affects the performance of the application which has a negative impact on the system performance. The VM consolidation strategy should avoid the VMM as possible and produce a minimum number of VM migrations performed from one physical machine to another during consolidation.
- **The SLA violations (SLAV)**

The quality of service (QoS) is a crucial factor for both cloud service providers and users. Cloud providers should provide a high level of service to their users, which can be outlined in the Service Level Agreement (SLA) form. The SLA can specify the minimum throughput and maximum response time requirements for the QoS. In our experiment, we evaluated SLA violations as a product of Overload Time Fraction (OTF) and Performance Degradation due to Migrations (PDM).

The OTF is the time percentage of a host being active during full utilization of the CPU and is calculated as shown in Equation 6:

$$OTF = \frac{1}{q}\sum\nolimits_{i=1}^{m} \frac{T_i^f}{T_a^i} \quad (6)$$

Where, T^f is the time for full CPU utilization,T^a is the time during which the host is active.

The PDM is the CPU utilization incurred due to migration and is computed as demonstrated in Equation 7:

$$PDM = \frac{1}{n}\sum\nolimits_{i=1}^{n} \frac{U_i^{mf}}{U\ _i} \quad (7)$$

Where, U^m is the CPU utilization during migration, and, U^f is the total CPU utilization requested by a virtual machine. Finally, the SLAV is calculated as described in Equation 8:

$$SLAV = OTF \times PDM \quad (8)$$

C. EXPERIMENTAL RESULTS

The experimental results are applied to evaluate the performance of (the MGVMC) strategy in a cloud environment. In all experiments, all parameter settings are similar except the data set. Furthermore, all the reported results are the average of a total of 10 simulation runs. The aim of the first set of experiments evaluates the efficiency of the proposed MGVMC strategy and to identify the best value of £which is able to balance energy consumption, the number of VM migrations, and SLA violations. The performance of the MGVMC strategy using different values of £is compared to different scenarios of the EAC strategy [17] based on Energy consumption (EC), Number of VM migrations (VMM), and The SLA violations (SLAV). A benchmark (EAC) was evaluated using three host overload detection strategies namely (interquartile range (IQR), Median Absolute Deviation (MAD), and Local Regression Robust (LRR)). The parameters for the three strategies are (1, 2, and 3) for IQR and MAD, and (1.0, 1.2, 1.4) for LRR.

The proposed MGVMC strategy appears to be more energy efficient than other techniques when analyzing the average energy consumption of six workloads as demonstrated in Figure 3. From the figure, it seems that the MGVMC_0.3 strategy consumes the least amount of energy. The performance of MGVMC variants is ranked in a specific order, MGVMC_0.3 performs the best in terms of energy efficiency, followed by MGVMC_0.4, MGVMC_0.5, and MGVMC_0.6. The performance of MGVMC variants is followed by EAC_LRR_1.0 and EAC_MAD_1. On the other hand, EAC_IQR_3 has the highest energy consumption, followed by EAC_IQR_2 and EAC_LRR_1. 4.

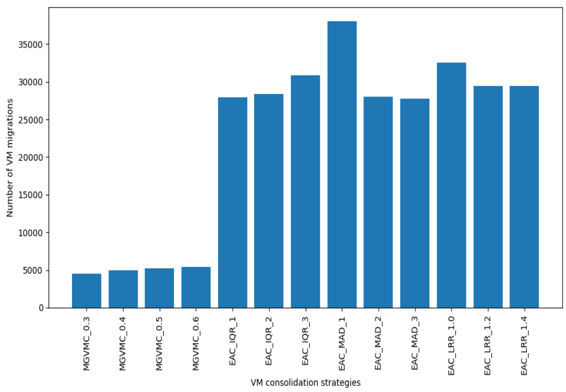

FIGURE 3. **The average energy consumption in kWh by physical machines across different VM consolidation strategies.**

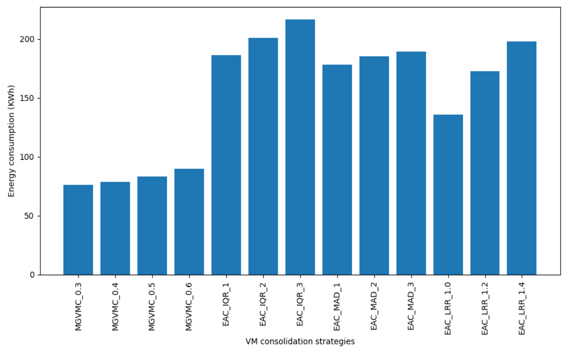

FIGURE 4. **The average number of VM migrations across different VM consolidation strategies.**

MGVMC_0.3 also performs 1.77 times better in terms of energy consumption compared to the EAC_LRR_1.0 strategy. The minimum energy consumption of the MGVMC comes as a result of minimizing the number of active PMs during the implementation of the workload.

The average number of VM migrations for the six workloads is depicted in Figure 4. As shown in the figure, the MGVMC strategy minimizes the number of VM migrations compared to other techniques. The MGVMC_0.3 strategy has the minimum number of VM migrations followed by MGVMC_0.4, MGVMC_0.5, and MGVMC_0.6. the performance of MGVMC variants outperforms all other EAC variants followed by EAC_MAD_3, EAC_IQR_1, and EAC_MAD_2. The worst performance is registered by EAC_MAD_1 followed by EAC_LRR_1.2. On average MGVMC_0.3 performs 6.12 times better than the best-performing EAC strategy, EAC_MAD_3. MGVMS strategy registered these outperforming results in terms of the average number of VM migrations since MGA takes into account the number of VM migrations in the initial population generation strategy, VM placement Crossover Strategy and VM placement Mutation Strategy.

In Figure 5, it is shown that the MGVMC variants perform better than other strategies in terms of the average Service Level Agreement (SLA) violations for the six workloads. Specifically, the MGVMC_0.3 strategy generates the smallest number of SLA violations followed by MGVMC_0.4,

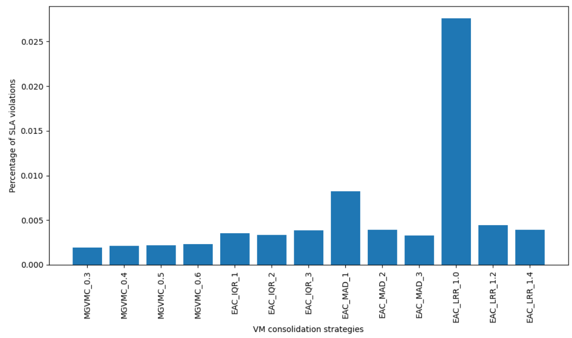

FIGURE 5. **The average SLA violation across different VM consolidation strategies.**

MGVMC_0.5, and MGVMC_0.6. The MGVMC variants are followed by the EAC_MAD_3 and EAC_IQR_2.0 strategies. EAC_LRR_1.0 registered the worst SLA violations followed by EAC_MAD_1. This suggests that the proposed MGVMC strategy is able to effectively meet the required service levels while minimizing the number of SLA violations. This is because the fitness function of the MGA minimizes the number of overloaded PMs, which directly minimize both (OTF) and (PDM).

It is important to consider energy efficiency and the ability to meet SLA requirements when selecting an appropriate strategy for a given workload while minimizing the number of VM migrations. Overall, the MGVMC_0.3 strategy outperforms other strategies by significantly enhancing results of both energy consumption, number of VM migrations, and SLA violations.

To assess the effectiveness of the MGVMC_0.3 strategy, it was compared with recent VM consolidation strategies including PABFD [23], NVMC [9], UAVMP [11], and HVMAP [10]. The comparison was based on three metrics: energy consumption (EC), number of VM migrations (VMM), and service level agreement violations (SLAV).

In Figure 6, the energy consumption for various VM consolidation strategies is presented. The MGVMC strategy showed the lowest energy consumption in all cases, with values of 83.45, 60.33, 80.11, 105.47, 85.24, and 57.88 for W1, W2, W3, W4, W5, and W6, respectively. The MGVMC strategy was superior to the other strategies in terms of energy consumption, with the UAVMP strategy coming in second place and PABFD in last place. The MGVMC achieves its minimum energy consumption by reducing the number of active PMs while implementing the workload.

Figure 7 illustrates the number of VM migrations for various VM consolidation strategies during the execution of W1, W2, W3, W4, W5, and W6 workloads. The MGVMC strategy resulted in the fewest number of VM migrations for all workloads, followed by the NVMC strategy. The results indicate that the MGVMC strategy is the most effective at minimizing the number of VM migrations. This is attributed to the fact that the MGA algorithm incorporates the number of VM migrations as a crucial factor in the initial population generation strategy, the VM placement crossover strategy,

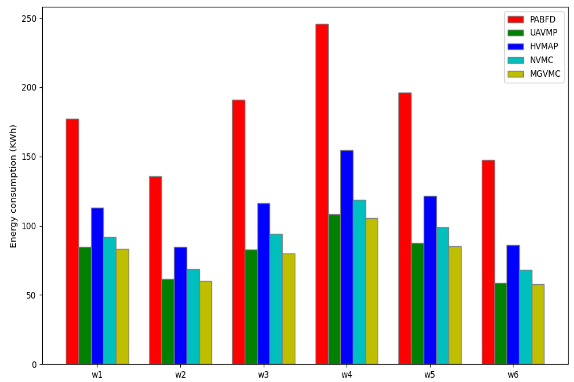

FIGURE 6. Energy consumption in kWh by physical machines across different VM consolidation strategies.

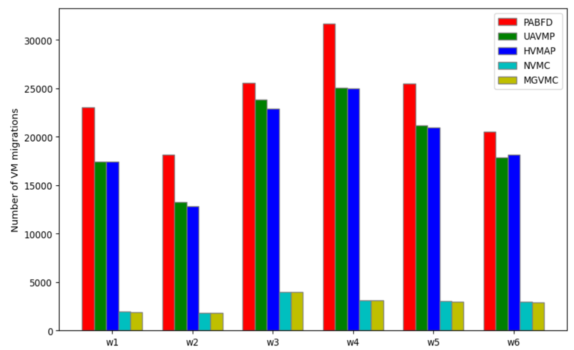

FIGURE 7. Number of VM migrations across different VM consolidation strategies.

and the VM placement mutation strategy. This comprehensive approach to considering VM migration in multiple stages of the algorithm enables MGVMS to generate more efficient and effective solutions.

Figure 8 presents the percentage of SLA violations that occurred during the execution of the six workloads using various VM consolidation strategies. The MGVMC strategy had the lowest percentage of SLA violations, outperforming the other strategies. The MGVMC strategy recorded the lowest number of SLA violations, followed by the NVMC strategy. These results suggest that the MGVMC strategy is the most effective at avoiding SLA violations. These results suggest that the proposed MGVMC strategy is capable of meeting the required service levels while avoiding SLA violations. This is achieved through the MGA algorithm's fitness function, which minimizes the number of overloaded physical machines. By doing so, both the (OTF) and (PDM) are reduced. This comprehensive approach to minimizing SLA violations allows the MGVMC strategy to be the most effective at avoiding SLA violations.

Based on the results of all workloads, the MGVMC strategy outperforms other VM consolidation strategies in terms of energy consumption, number of VM migrations, and SLA violations. The MGVMC strategy can minimize the three-performance metrics simultaneously, making it the most effective strategy overall.

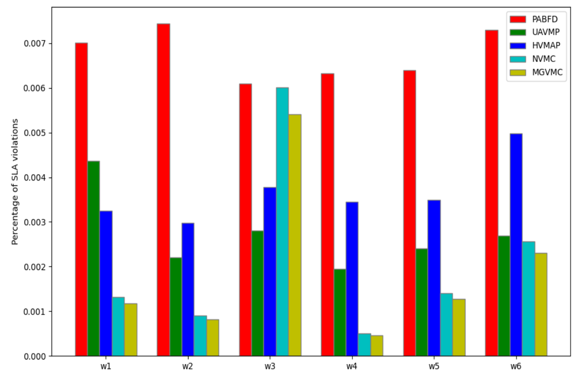

FIGURE 8. SLA violation across different VM consolidation strategies.

While MGVMC's strategy focused on energy consumption, VM migrations, and SLA, we did not investigate MGVMC's impact on efficiency, reliability, and scalability. For MGA, we have adopted standard parameter configurations commonly used in the field. We do not focus on parameter configuration impact. Despite MGVMC's superior performance in a simulated environment, the impact of their use in an actual cloud environment remains unknown.

VI. CONCLUSION

In this paper, we propose a modified Genetic-based VM consolidation (MGVMC) strategy for improving energy efficiency and quality-of-service (QoS) for cloud service providers. The goal of the MGVMC strategy is to minimize energy consumption, the number of VM migrations, and SLA violations. The proposed strategy uses a modified genetic algorithm to minimize the number of under-utilized and over-utilized physical machines (PMs) as much as possible. By migrating VMs from under-utilized PMs to PMs with no VMs and switching those PMs to sleep mode, energy consumption can be minimized. Additionally, by migrating VMs from over-utilized PMs, until they reach a state of fair utilization, the number of SLA violations can be minimized. We conducted large-scale experiments using real-world data from execution traces of PlanetLab virtual machines and found that the MGVMC strategy outperforms other VM consolidation strategies, significantly optimizing energy consumption, the number of VM migrations, and SLA violations.

In the future, we intend to extend MGVMC to consider another issue that affects cloud processing other than energy consumption. This includes efficiency, reliability, and scalability. Furthermore, we will focus on MGA parameter settings, to find the most efficient parameter settings. Finally, we plan to extend MGVMC to real-world cloud environments.

REFERENCES

[1] J. Dogani, F. Khunjush, and M. Seydali, "Host load prediction in cloud computing with discrete wavelet transformation (DWT) and bidirectional gated recurrent unit (BiGRU) network," *Comput. Commun.*, vol. 198, pp. 157–174, Jan. 2023.

[2] T. Khan, W. Tian, S. Ilager, and R. Buyya, "Workload forecasting and energy state estimation in cloud data centres: ML-centric approach," *Future Gener. Comput. Syst.*, vol. 128, pp. 320–332, Mar. 2022.

[3] S. A. Javadi, A. Suresh, M. Wajahat, and A. Gandhi, ''Scavenger: A black-box batch workload resource manager for improving utilization in cloud environments,'' in *Proc. ACM Symp. Cloud Comput.*, Nov. 2019, pp. 272–285.

[4] R. N. Avula and C. Zou, ''Performance evaluation of TPC-C benchmark on various cloud providers,'' in *Proc. 11th IEEE Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON)*, Oct. 2020, pp. 226–233.

[5] M. Radi, A. Alwan, A. Abualkishik, A. Marks, and Y. Gulzar, ''Efficient and cost-effective service broker policy based on user priority in VIKOR for cloud computing,'' *Basic Appl. Sci. Sci. J. King Faisal Univ.*, vol. 22, no. 2, pp. 1–8, 2021.

[6] F. A. Saif, ''Hybrid meta-heuristic genetic algorithm: Differential evolution algorithms for scientific workflow scheduling in heterogeneous cloud environment,'' in *Proc. Future Technol. Conf. (FTC)*, vol. 3. Cham, Switzerland: Springer, 2022, pp. 16–43.

[7] M. M. Alshammari, ''Data backup and recovery with a minimum replica plan in a multi-cloud environment,'' *Int. J. Grid High Perform. Comput. (IJGHPC)*, vol. 12, no. 2, pp. 102–120, 2020.

[8] F. A. Saif, ''Performance evaluation of task scheduling using hybrid meta-heuristic in heterogeneous cloud environment,'' *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 8, no. 6, pp. 3249–3257, Dec. 2019.

[9] M. A. Khan, ''An efficient energy-aware approach for dynamic VM consolidation on cloud platforms,'' *Cluster Comput.*, vol. 24, no. 4, pp. 3293–3310, Dec. 2021.

[10] D. Dabhi and D. Thakor, ''Hybrid VM allocation and placement policy for VM consolidation process in cloud data centres,'' *Int. J. Grid Utility Comput.*, vol. 13, no. 5, pp. 459–471, 2022.

[11] D. Dabhi and D. Thakor, ''Utilisation-aware VM placement policy for workload consolidation in cloud data centres,'' *Int. J. Commun. Netw. Distrib. Syst.*, vol. 28, no. 6, pp. 704–726, 2022.

[12] S. Omer, S. Azizi, M. Shojafar, and R. Tafazolli, ''A priority, power and traffic-aware virtual machine placement of IoT applications in cloud data centers,'' *J. Syst. Archit.*, vol. 115, May 2021, Art. no. 101996.

[13] B. Speitkamp and M. Bichler, ''A mathematical programming approach for server consolidation problems in virtualized data centers,'' *IEEE Trans. Services Comput.*, vol. 3, no. 4, pp. 266–278, Oct. 2010.

[14] K. Karmakar, S. Banerjee, R. K. Das, and S. Khatua, ''Utilization aware and network I/O intensive virtual machine placement policies for cloud data center,'' *J. Netw. Comput. Appl.*, vol. 205, Sep. 2022, Art. no. 103442.

[15] C. Dupont, T. Schulze, G. Giuliani, A. Somov, and F. Hermenier, ''An energy aware framework for virtual machine placement in cloud federated data centres,'' in *Proc. 3rd Int. Conf. Future Energy Syst., Where Energy, Comput. Commun. Meet*, May 2012, pp. 1–10.

[16] L. Zhang, Y. Zhuang, and W. Zhu, ''Constraint programming based virtual cloud resources allocation model,'' *Int. J. Hybrid Inf. Technol.*, vol. 6, no. 6, pp. 333–344, Nov. 2013.

[17] 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.

[18] U. Arshad, M. Aleem, G. Srivastava, and J. C.-W. Lin, ''Utilizing power consumption and SLA violations using dynamic VM consolidation in cloud data centers,'' *Renew. Sustain. Energy Rev.*, vol. 167, Oct. 2022, Art. no. 112782.

[19] Q. Wu, F. Ishikawa, Q. Zhu, and Y. Xia, ''Energy and migration cost-aware dynamic virtual machine consolidation in heterogeneous cloud datacenters,'' *IEEE Trans. Services Comput.*, vol. 12, no. 4, pp. 550–563, Jul. 2019.

[20] X. Ye, Y. Yin, and L. Lan, ''Energy-efficient many-objective virtual machine placement optimization in a cloud computing environment,'' *IEEE Access*, vol. 5, pp. 16006–16020, 2017.

[21] G. Wu, ''Energy-efficient virtual machine placement in data centers by genetic algorithm,'' in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland: Springer, 2012, pp. 315–323.

[22] C. Sonklin, M. Tang, and Y. Tian, ''A decrease-and-conquer genetic algorithm for energy efficient virtual machine placement in data centers,'' in *Proc. IEEE 15th Int. Conf. Ind. Informat. (INDIN)*, Jul. 2017, pp. 135–140.

[23] A. Beloglazov, ''Energy-efficient management of virtual machines in data centers for cloud computing,'' Ph.D. thesis, Dept. Comput. Inf. Syst., Univ. Melbourne, Australia, Melbourne, VIC, Australia, 2013.

[24] M. C. Silva Filho, R. L. Oliveira, C. C. Monteiro, P. R. M. Inacio, and M. M. Freire, ''CloudSim plus: A cloud computing simulation framework pursuing software engineering principles for improved modularity, extensibility and correctness,'' in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM)*, May 2017, pp. 400–406.

[25] H. Li and T. Z. Li Shuhua, ''Energy-performance optimisation for the dynamic consolidation of virtual machines in cloud computing,'' *Int. J. Services Oper. Inform.*, vol. 9, no. 1, pp. 62–82, 2018.

[26] Z. A. Mann and M. Szabó, ''Which is the best algorithm for virtual machine placement optimization,'' *Concurrency Comput., Pract. Exper.*, vol. 29, no. 10, 2017, Art. no. e4083.

MOHAMMED RADI received the B.A. degree in computer science from Al-Azhar University, Gaza, in 2001, the master's degree in computer science from the University of Jordan, in 2003, and the Ph.D. degree in distributed databases from the University of Putra Malaysia, in 2009. He is currently an Assistant Professor with the Department of Computer and Information Sciences, Al-Aqsa University, Gaza, Palestine. He is the Dean of planning, development, and quality assurance with Al-Aaqsa University. He published a set of research papers in international journals and conference proceedings and actively participate in a large number of workshops and local forums. His research interests include cloud computing, cloud data storage, data management, and high-performance computing.

ALI A. ALWAN received the Master of Computer Science and Ph.D. degrees in computer science from Universiti Putra Malaysia (UPM), Malaysia, in 2009 and 2013, respectively. He is currently an Assistant Professor with the School of Theoretical and Applied Science, Ramapo College of New Jersey, USA. His research interests include databases (incomplete, uncertain, and probabilistic), preference queries, query processing and optimization, data management, data integration, location-based social networks (LBSN), recommendation systems, data mining, database in the cloud, big data management, and crowd-sourced database.

YONIS GULZAR received the bachelor's degree in computer science from the University of Kashmir, in 2010, the master's degree in computer science from Bangalore University, India, in 2013, and the Ph.D. degree in computer science from International Islamic University Malaysia, Malaysia, in 2018. He was a Lecturer (PT), a Teaching Assistant, and a Research Assistant with the Department of Computer Science, International Islamic University Malaysia, until 2018. Since 2019, he has been an Assistant Professor with King Faisal University (KFU), Saudi Arabia. His research interests include preference queries, skyline queries, query processing, machine learning, deep learning, image processing, and computer vision.