VM Scheduling for Efficient Dynamically Migrated Virtual Machines (VMS-EDMVM) in Cloud Computing Environment

S. Supreeth[1*], and Kirankumari Patil[2]
[1]Research Scholar, VTU, Belagavi and Assistant Professor, School of CSE, REVA University, India
[E-mail: supreeth1588@gmail.com]
[2] Professor, Department of CSE, REVA ITM, India
[E-mail: kirankumari@reva.edu.in]
[*]Corresponding author: S. Supreeth

Received September 12, 2021; revised April 27, 2022; accepted June 7, 2022; published June 30, 2022

Abstract

With the massive demand and growth of cloud computing, virtualization plays an important role in providing services to end-users efficiently. However, with the increase in services over Cloud Computing, it is becoming more challenging to manage and run multiple Virtual Machines (VMs) in Cloud Computing because of excessive power consumption. It is thus important to overcome these challenges by adopting an efficient technique to manage and monitor the status of VMs in a cloud environment. Reduction of power/energy consumption can be done by managing VMs more effectively in the datacenters of the cloud environment by switching between the active and inactive states of a VM. As a result, energy consumption reduces carbon emissions, leading to green cloud computing. The proposed Efficient Dynamic VM Scheduling approach minimizes Service Level Agreement (SLA) violations and manages VM migration by lowering the energy consumption effectively along with the balanced load. In the proposed work, VM Scheduling for Efficient Dynamically Migrated VM (VMS-EDMVM) approach first detects the over-utilized host using the Modified Weighted Linear Regression (MWLR) algorithm and along with the dynamic utilization model for an under-utilized host. Maximum Power Reduction and Reduced Time (MPRRT) approach has been developed for the VM selection followed by a two-phase Best-Fit CPU, BW (BFCB) VM Scheduling mechanism which is simulated in CloudSim based on the adaptive utilization threshold base. The proposed work achieved a Power consumption of 108.45 kWh, and the total SLA violation was 0.1%. The VM migration count was reduced to 2,202 times, revealing better performance as compared to other methods mentioned in this paper.

Keywords: Dynamic VM Migration, VM Scheduling, Energy Efficient, VM Selection, VM Placement

1. Introduction

The cloud environment is an internet-based technology that is overtaking onsite server-based technology. Cloud offers a variety of services to small, medium, and large-scale enterprises to run their business by offering a pay-as-you-go service at a low cost. The cloud comprises a pool of various IT resources that are made available over the internet, allowing a large number of people to access them [1]. Many enterprises purchase virtual servers from cloud vendors to host their applications for their business model. These applications might turn out to be massive traffic over the network, leading to large consumption of cloud resources. Running these enterprises with enormous demands on cloud computing leads to a rapid increase in operating cost and power consumption by causing an adverse effect on the environment [2].

Minimizing energy consumption [3] in cloud computing is a critical task, as there is a massive inflow of data that needs to be processed more quickly by large servers in a certain amount of time. To manage such massive requests, cloud computing incorporates large data centers with raw physical machines that produce a large number of carbon emissions, adversely impacting the environment. It is a critical issue to minimize energy consumption, and to do so effectively to address this issue; otherwise, data centers may continue consuming massive amounts of energy. The aforementioned objectives can be achieved by managing the backbone of cloud computing, i.e., VMs are installed on better computing resources (storage units with high-end dedicated networks). But all these devices need good cooling facilities, 24/7. Thus, if we manage these devices intelligently, the requirement for these cooling devices can be reduced. The main objectives of this work are to enhance an environment-friendly system by reducing overall power consumption by focusing on Quality of Service (QoS) to minimize SLA violations which can be achieved by developing effective policies and algorithms on VMs [4]. SLA decides the substance of administration, level of execution, costs, and penalty for interrupted services. Any deviation in the QoS results in an SLA violation. Subsequently, a penalty should be paid by the vendor. To avoid such big penalties, the provider should install a framework to deal with virtual resources efficiently [5].

VM consolidation [6] is also referred to as VM migration, and this is one the most valuable methods in energy/power consumption which assists the executives in a cloud environment. This procedure improves resource usage and results in effective utilization. Consolidation alludes to the live placement of VMs from one host to another, with minimal interruptions faced during the execution. The method involves moving VMs to a lower utilized host and changing the inactive hosts to the power-saving mode [7].

To achieve this objective, in this work, a series of steps are carried out on VMs in a datacenter environment. First, the resource usage data of all VMs is collected for each physical machine or host, and a Modified Weighted Linear Regression (MWLR) algorithm has been developed to determine the over-utilized host [8]. Then a dynamic and adaptive energy-efficient VM migration and Best-Fit CPU, BW (BFCB) mechanism for VM Scheduling[9], has been incorporated by considering the violation of SLA, power consumption, and VM migrations count in the cloud platform [10]. In this work, an efficient Dynamic VM Scheduling has been considered for the migrated VM in Cloud Computing Environment.

2. Related Work

If the data growth increases, then the processing time will increase leading to the over utilization of the power consumption. This is because users do not appreciate server downtime, so servers must run 24 hours a day by consuming more power for data centers' for backend operations. By managing resources intelligently, it is possible to lower the overall power consumption by all the servers in data centers with SLA violations to a minimum by minimizing VM Migrations. For this, in this section, some of the literary works of the researchers are referred[11].

The scheduling of the applications onto the virtual servers takes into consideration of the power and migration costs as performance parameters. Two contributions of this work include cost-aware placement of the applications and one more contribution towards minimization of the power consumption. This work having a power minimization framework is realistic and considered on two server platforms [12]. Proposed algorithms for Provisioning and Allocation algorithm improves the power efficiency of the Cloud by negotiating QoS and proposed architecture for energy-efficient management, policies on allocation and scheduling finally are addressed with future directions for the researchers [13].

The huge data centers consume more electrical energy and emit more CO_2. To reduce CO_2 emission, dynamic VM migrations and by shutting down the idle VMs, optimizes the resource usage and energy consumption. A novel heuristic dynamic VM adaptive algorithm [14] has been proposed. The proposed algorithm reduced energy consumption significantly by considering SLA.

To save energy consumption, the VM Scheduling mechanism [15] has been proposed and implemented. It targeted load balancing and the balanced temperature to ensure that none of the physical nodes suffer from overutilization of the temperature. The proposed algorithm worked for power consumption but was satisfatory. The reduced energy consumption [3] can be obtained by revising the scheduling method of VM’s by keeping SLA violation and Migrations as a parameter.VM Scheduling optimizes fairly well despite the user’s behavior and history of the SLA violations.

The applications of Virtualization in cloud datacenters allow a large number of VM’s hosted into fewer Physical Machines (PMs). This fashion is considered as bins. A Vector bin packing algorithm has been developed [16] for VM consolidation by powering off unused PM’s for power consumption and taking migration also into account.

In [17], deals with the problem of stress situations, when the host's capacity is exceeded by the demand for virtual machines. In this relocation problem, determining VM and migration to the host has been implemented and evaluated with First Fit-based relocation policy and finally, the research directions towards fuzzy controllers for VM selection.

Virtual Machine migration helps in energy saving, increasing energy efficiency, and other QoS parameters like VM migration time and downtime. For this, serial migration and post copy of VMs are introduced and also M/M/C/C queuing based model [18] was applied to improve the blocking ratio and average request waiting time of the VM by conducting mathematical analysis.

In a data center, energy consumption can be evaluated by comprehensive scheduling of VM migrations. This suggests a new set of evaluation methods [19] that were constructed according to some new metrics for different VM Migration Scheduling for various viewpoints of data center scale and with different workload types.

Network Traffic is created at Virtual Machine source and destination while migrating the Virtual Machine using Migration of pre and post-copies. This causes migration traffic to become congested, lengthens the migration time and degrading the VM's performance. To overcome this drawback, a traffic-sensitive live migration [20] is used for both post and pre-copy migration instead of a single predefined technique. On the KVM/QEMU framework, a prototype of traffic-aware migration is being developed and the same is compared with traditional techniques.

Energy efficiency, fault tolerance, and availability can be accomplished using a virtual machine Migration. In this, the combined forecasting technique has been introduced to predict the requirements of the Virtual machine for migration by forecasting the load. Experiments were conducted to determine the efficiency of this work by lowering the number of migrations and energy usage [21].

Time consumed to evacuate one or many states of VMs from the source Physical Machine is known as eviction time. In traditional approaches, the complete migration time is calculated during pre and post-copy of the VM Migration received at the destination. Live migration with Scatter-Gather [22] decouples source and destination by reducing eviction time during migration. This also allows eviction of multiple VMs at the same time by the use of deduplication to reduce network traffic. It runs on the KVM/QEMU environment and cuts eviction time by 6 times when compared to the traditional approach.

Advanced research techniques are one of the challenges faced to cut down on energy consumption in cloud computing. Most of the proposed efficient solutions to save energy consumption techniques have a reduced performance. This paper proposed a method for calculating the working utilization of the host using “PPRGear” [23] based on the utilization sample with power ratio and in addition, presented a framework for allocating and migrating virtual machines across multiple hosts. Compared with existing approaches, the same achieved less energy consumption of 69.31% compared to previous work.

VM Allocation and VM Migration in Cloud datacenter depends on 3 factors i.e., when which, where to move and it is very difficult to decide on these factors. The greedy approach leads to high variance and poor convergence in Heuristic-Based algorithms. An online Megh algorithm [24] was proposed and it works on a dynamic basis and does not need any pre-knowledge of the workload. This algorithm is free from execution overhead compared to migration time of the Virtual Machine by using real-time execution overhead. Also, it has been implemented using CloudSim toolkit with the PlanetLab and Google Cluster workloads and are more efficient as compared to traditional approaches.

Key strategies to increase the data center's efficiency through better resource allocation can be achieved significantly by using the Migration of virtual machines (VMs). In Live VM Migration, it preserves VM’s memory usage, and for reducing downtime, disk is copied at destination host using Pre, Post-Copy, and Hybrid classical methods that have distinct characteristics and ability to perform well. This presented dynamic hybrid Live Migration mathematical model [25] improves migration time and downtime.

Data Centers are consuming more power because of large-scale servers and this is leading to a problem. GM-DPSO algorithm is proposed based on the QoS by reducing power consumption. For load detection, the grey prediction algorithm was applied, and based on this underutilized, over-utilized VMs were detected. Next, the same VMs are placed using a discrete swarm intelligence algorithm that is better. In addition, it improves well in load balancing of the VMs and results in reduced SLA violation. This approach proved 34.53% & 97.53% improvement in energy consumption and reduced SLA compared to traditional approaches [26].

VM Migration helps in balancing the load and saves energy on cloud data centers, but it also leads to network overhead. This proposes a 3-Way decision VM migration [27] to save energy by considering network correlation between VMs. In this, first hosts are classified into under, normal, overloaded Cloud hosts. Under loaded VMs moved to lightly overloaded hosts, and moderately overloaded hosts into the lightly overloaded host. The experimental results show that by meeting the SLA, the suggested technique can reduce energy consumption.

The energy-efficient model allows sharing of information over the cloud data center. For working with efficient energy consumption, the intelligence parameters are important. Improved resource usage and energy consumption are addressed and considered as NP-Hard problems. A heuristic algorithm [28] was applied for securing optimal Virtual Machine migration. These techniques also minimize CO_2 emissions.

A two-stage load balance-based VM migration in a cloud environment was implemented. In many traditional approaches, this is considered as job assignment and considers only the current load without considering the load balance that leads to limitations in real-time approaches. As a result, Genetic-based techniques are integrated. Performance models of VMs are extracted by this method. The table is created for parameters and their performance is addressed by using Gene Expression Programming [29] for generating symbolic regression models for VM performance to predict current and future workload. The proposed approach's performance was assessed using real-cloud and experimental results outperformed.

3. System Model

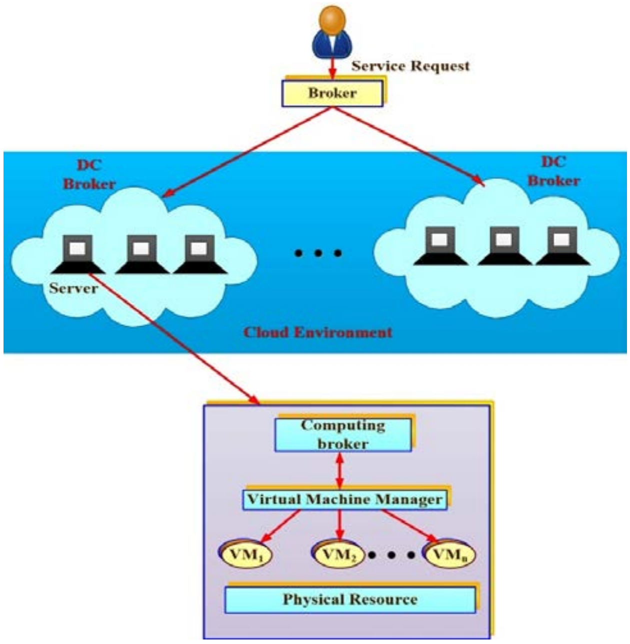

Fig. 1. System Architecture of Cloud Environment

Cloud Computing is internet-based leasing compute resource [33]. The architecture of the Cloud Computing and Cloud service requests by the Cloud Consumer requests are handled at the Backend Cloud System as depicted in **Fig. 1**. Here, we can see how the Datacenters host Virtual Machines in the Cloud Environment. Here, the Data Center (DC) broker is responsible for managing any host-related activities and it is also responsible for assigning cloudlets (user requests) to their particular virtual machine. Physical Servers are usually referred to as hosts that can accommodate the virtual machines that are used by end-users and the virtual machine is a logical computer system that is capable of performing the same functions as a physical machine[31]. To manage huge number of requests, Cloud Computing vendors maintain large Datacenters with high-end physical Machines and all these devices need to run 24/7 to avoid downtime in the Cloud services. This leads to operating all the resources up with proper mechanisms so that all the resources assigned to the cloud consumers as part of the cloud services can be utilized effectively. Cloud Vendors can not maintain the physical servers for every consumer because it is expensive and time-consuming. Instead, Cloud Vendors adapt Virtualization Technology to virtualize their IT Resources[32], so that multiple services can run simultaneously through multiple Virtual Machines that are hosted on a single physical server. But whenever it is busy executing the instruction even though the execution time is less, the resources consume a large amount of energy. Therefore if we don't manage it intelligently it may emit huge CO_2, which may impact the environment. For managing effectively, the Backbone of Cloud Computing resources, the requirement of the cooling system can also be reduced [33]-[34].

The data center broker is mimicking like an admin and manages hosts as well assigns virtual machines to specific hosts. Data center broker has to decide VM allocation on the host. This is dependent on the VM allocation policy [35]. The allocation policy determines which host is best for allocating a virtual machine for a specific task. Under the allocation policy, a host must be selected where a VM can be allocated using over or under-utilized mechanisms.

In this work, the entire cloud server is interconnected, and user requests for VM creation are processed based on user workloads on the data center via the internet on the Cloud Environment. The performance Metrics which are considered in the proposed work include:

- Total Energy/Power Consumption (E)
- VM Migrations count
- Service Level Agreement (SLA)
- The Performance Degradation due to Migration (PDM)
- SLA violation time per active host (SLATAH)
- Energy consumption and SLA Violations (ESV)

The Total Power Consumption [36],[13] of the host can be calculated using Equation (1).

$$P(u) = k \cdot Pmax + (1 - k) \cdot Pmax \cdot u \qquad (1)$$

Pmax: a host's maximum power in a running state; *k*: idle Physical Machine power consumption (in terms of %), and *u*: CPU utilization. Because of this, we define CPU utilization as a function that changes over time *u(t)*. Equation (2) gives the Total Energy Consumption.

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

In this, SLA Violation performance is measured using two values ie.

1) PDM of VM Migration can be calculated using Equation (3) and
2) SLATAH using Equation (4).

$$PDM = \frac{1}{N}\sum_{i=1}^{N}\frac{(Pr-Pa)}{Pr} \qquad (3)$$

In Equation (3), *N*: denotes number of VMs; Pr: denotes Hosts VMs Requested performance, and *Pa*: denotes allocated performance to the VMs.

$$SLATAH = \frac{1}{N}\sum_{i=1}^{N}\frac{T_{fi}}{T_{ai}} \qquad (4)$$

In Equation (4), N corresponds to the number of hosts; T_{fi} represents the total time for full utilization by the host; and T_{ai} denotes the total time that hosts are in the active mode.The SLA Violation [37] metric is formed by joining the two previous metrics in Equation (5), as follows:

$$SLAV = SLATAH \cdot PDM \quad (5)$$

The resource management system aims to reduce both energy and SLA violations. As a result, ESV [14] is defined in Equation (6).

$$ESV = Energy\ Consumption \cdot SLAV \quad (6)$$

4. Proposed Work

In the proposed work, it has been considered if the request from the end user's for VM Scheduling on the Cloud datacenter must be scheduled properly as otherwise it may consume a large amount of power and it may impact the environment. VM Scheduling determines which host is best for Scheduling a VM on the Cloud Datacenter. Under the scheduling policy, a host must be selected where a VM can be scheduled based on over or under-utilized host mechanisms. Using VM Selection policy[38] the VM is decided for migration from the host device. When the host is under-utilized, all the running VMs are migrated and shut down. In the case of an over-utilized host, migration of VMs will take place until the host satisfies the load balance. The VM Selection policy selects an over-utilized VM from a list of VMs running in the host placed on the efficient host based on CPU, memory and bandwidth usage by using the VM Scheduling strategy. **Fig. 2** depicts the entire process of VM Scheduling for the migrated VM. The VM Migration and Scheduling includes the following steps:

1. **Detection of Over-Utilized Host:** For Migrating the VM to another host.
2. **Detection of Under-Utilized Host:** If it is least utilized, then it can be moved to sleep mode.
3. **VM Selection:** Identification of the Over-utilized or underutilized VM for migration.
4. **VM Scheduling:** Reserves the VM on a particular Host.

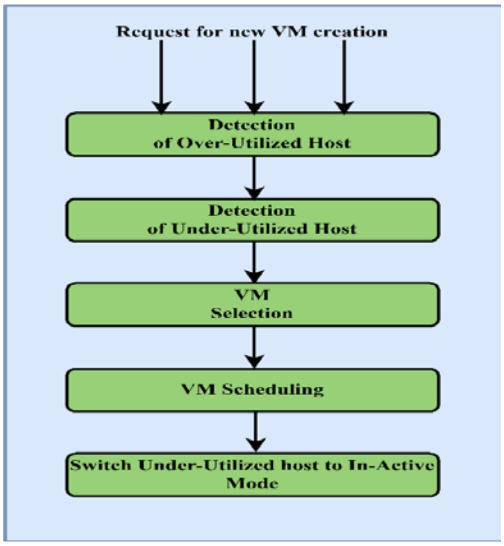

Fig. 2. The overall process of the VM Migration

4.1 Modified Weighted Local Regression (MWLR) Over-utilized host detection Algorithm

The proposed Modified Weighted Local Regression (MWLR) algorithm predicts the overutilization host by considering CPU utilization, Memory, Bandwidth. VMs running on that host can be moved to a lesser loaded physical host before an SLA violation occurs, or a physical host can be prevented as a target for VM migration entirely. The proposed MWLR algorithm-2 is based on a technique called Local Regression to forecast future CPU and Memory usage, Bandwidth (BW) utilization. It makes a proper prediction for future CPU, Memory, and BW usage based on a host's recorded resources.

Algorithm 1 mentioned below, is used to detect any over-utilized host. There are two types of approaches being used to determine whether the host is over-utilized or not. They are

1. Non-threshold based algorithm
2. Adaptive utilization threshold base

The Adaptive utilization threshold base determines whether it is overloaded based on the adaptive threshold values of the host resources. In the non-threshold-based algorithm, there is no fixed upper limit, but the host predicts the utilization in the next time frame based on recorded data. In this work, linear algebra-based regression method decides the relationship between two variables[39]. For each time interval, the two variables, ie. time and the weight of resource (CPU, Memory, BW) are used by the VM. The MWLR algorithm needs the utilization record to predict how that host will be used in the future. The future resource usage of a host is predicted based on the host's record, if the future resource usage of the host is more than or close to the calculated threshold value, then the host is considered over-utilized, and this has to be considered for migration to avoid any further SLA violations. Equation (7) is used to calculate the regression line.

$$Y = \beta_0 + \beta_1 \tag{7}$$

The dependent variable is Y, and β_0 & β_1 are the coefficients, which will be determined using the least-squares method [40] as in Equation (8) and Equation (9):

$$\widehat{\beta_0} = \bar{Y} - \widehat{\beta_1}\bar{X} \tag{8}$$

$$\widehat{\beta_1} = \frac{\sum_{i=1}^{n}(Xi-\bar{X})(Yi-\bar{Y})}{\sum_{i=1}^{n}(Xi-\bar{X})^2} \tag{9}$$

Where, $\bar{X}$ and $\bar{Y}$ are the means of the observations X and Y, $\widehat{\beta_0}$ and $\widehat{\beta_1}$ provide an estimate of β_0 and β_1. There is one observation(X_i, Y_i) for each, To assign a neighborhood weight, the square weight function [41] - [42] is defined as in Equation (10):

$$B(x) = \begin{cases} (1-x^2)^2 & For\ |x| < 1 \\ 0 & For\ |x| > 1 \end{cases} \tag{10}$$

Based on Equation (10), the neighboring weight is calculated as in Equation (11):

$$W_i(x) = B\left(\frac{(X_n-X_i)}{(X_n-X_i)}\right) = \left(1-\left(\frac{(X_n-X_i)}{(X_n-X_i)}\right)^2\right)^2 \tag{11}$$

Where, X_i and X_n are i^{th} and, as well as the final observations. K future host values are predicted using MWLR K iterations. Equation (12) defines the regression line for n data values.

$$\left.\begin{aligned} \overline{Y_1} &= \beta_0 + \beta_1 X_n \\ \overline{Y_2} &= \beta_0 + \beta_1 \overline{Y_1} \\ &\ldots\ldots \\ \overline{Y_n} &= \beta_0 + \beta_1 \, \overline{Y}_{(k-1)} \end{aligned}\right\} \tag{12}$$

Weighted Modified Linear Regression Algorithm defines two thresholds: Upper and Lower. The value of $i=1$, if the host's future utilization is expected to be higher than the threshold. However, when Weighted Modified Linear Regression identifies in future values ($i = 2$ *to* k). MWLR predicts two future values when we set $k = 2$ as in Equation (13).

$$\left.\begin{aligned} \overline{Y_1} &= \beta_0 + \beta_1 X_n \\ \overline{Y_2} &= \beta_0 + \beta_1 \overline{Y_1} \end{aligned}\right\} \tag{13}$$

In this case, there are the following constraints defined in Equation (14):

$$\begin{cases} x_n is\ upper - threshold;\ if\ c.\overline{Y_1} \geq\ 1 \\ x_n\ is\ pre - threshold;\ if\ c.\overline{Y_2} \geq\ 1 \end{cases} \tag{14}$$

Here c is the constant and Algorithm 1 shows how to find the Over Utilized host by using Algorithm 2.

Algorithm 1:- Detecting Over Utilized Host

Input:- Hosts from datacenter

Result:- Overloaded Host Detection (True or False)

for each PM in PM_List do
    Threshold <- PM(CPU, Memory, BW)
    Predicted_Threshold<-MWLR(CPU, Memory, BW)
    if (Upper_Threshold>=Predicted_Threshold) then
        PM_OverLoaded <- True
        PM_OverLoad_List<-PM
    end if
    else
        PM_Overloaded=False
return PM_OverLoad_List

Algorithm 2:- MWLR Algorithm

Input:- Physical Host/Machine Utilization

Result:- Upper Threshold

for each j=1 to n do
    X_i <-j
    Y_i <- Utilization(j)
    w_i <- Find_Weight(using Equation (11))
    X_i <- X_j * w_i
    Y_i <- Y_j * w_i
end for
Calculate β_0 and β_1 using the Equation (8) and (3)
Upper_Threshold <- $\beta_0 + \beta_1 * Cuttent\_Util$
for j=1 to 2 do

Find $\overline{\overline{Y}_1}$ and $\bar{Y}_2$
end for
return Upper_Threshold($\bar{\bar{Y}}_1$ and $\bar{Y}_2$)

4.2 Detecting Under-Utilized Host Algorithm

Algorithm 3 is used to detect an underutilized host, as described below. When an underutilized host is discovered, VMs which are running on those hosts are migrated to another by managing the power. If the utilization threshold calculated by considering CPU, Memory, and Bandwidth is less than the lower threshold, then the host is said to be under-Utilized. The lowest limit for utilization is updated as the new host encounters the least utilization. Algorithm 3 is used to find the Underutilized host. Equation (15) is used to find the utilization of the host.

$$Z = \sqrt{Util(CPU)^2 + Util(Memory)^2 + Util(BW)^2} \tag{15}$$

Previous lower host utilization is the median value Q1. So to find the lower threshold value, i.e. Q1 (median) lower half of the data set is used. So the lower threshold of the hosts in Datacenter can be calculated using the Equation (16).

$$T_{low} = u(\frac{1}{4}(n+1)) \tag{16}$$

Where, u: Host utilization; n: data values of data set

Algorithm 3:- Detecting an Under-Utilized Host/Physical Machine

Input:- Datacenter host_list and VMList
Result:- List of Migrating VMs
for-each host in hostList do
  if ($\mathrm{host.util}_{\mathrm{CPU}} < \mathrm{T}_{\mathrm{low}}$(CPU) && $\mathrm{host.util}_{\mathrm{Memory}} < \mathrm{T}_{\mathrm{low}}$(Memory) &&
    $\mathrm{host.util}_{\mathrm{BW}} < \mathrm{T}_{\mathrm{low}}$(BW)) do
    underloadedList <- h
  end if
end for
for-each host in hostList do
  $\mathrm{host.Util}_{CPU}$ <- (allocatedMIPS/totalMIPS)^2
  $\mathrm{host.Util}_{RAM}$ <- (allocatedRAM/totalRAM)^2
  $\mathrm{host.Util}_{BW}$ <- (allocatedBW/totalBW)^2
  $Z = \sqrt{host.Util_{CPU} + host.Util_{RAM} + host.Util_{BW}}$
end for
for each host in underloadedList do
  for each VM in hostVMList do
    for each host in hostList do
      if (host != PM_Overloaded) then
        The VM Migrates if the host has sufficient CPU, BW and RAM.
        VMMigrationList <- host.VM
        hostVMList<- hostVMList – host.VM
      end if
    end for
  end for
  if (count(VMMigrationList)>=1) do
    return VMMigrationList
  end if
end for

After detecting the over utilized VM, then it is selected using the MPRRT algorithm for the migration from the pool of Virtual Machines. The Virtual Machine selection approach selects VMs from a host when considered as over or under-utilized. When the host is over-utilized, the load will be balanced across multiple hosts, and if host is underutilized, all the VMs in the host are migrated to another host by saving power.

4.3 VM Selection Approach

When a host is overloaded, the VM selection approach is shown in Equation (17) & Equation (18) begins in choosing a VM to be migrated which would be the best choice for reducing SLA violation and power consumption with lower VM Migrations count. Using the VM selection approach, one or more overloaded VMs have been selected by minimizing host utilization under the threshold. This approach is repeated for each VM, and the host resource utilization is inspected once more after each selection. The Maximum Power Reduction and Reduced Time (MPRRT) policy is used to select a VM(v), and decreases the power consumption of the host that has a better trade-off with less migration time. *VMj* is a collection of VMs with host *i*, the MPRRT policy looks for a set $V \in VMj$ defined in Equation (17) and a migration time of *t(v)* defined in Equation (18), shown below:

$$V = \left\{ \begin{array}{cc} \left\{ \begin{array}{c} \{L|L \in VMj,\ u_i - \sum_{v \in L} u(v) < Tup,\ \ |L| \rightarrow min\} \\ \{\mathrm{P}\ |\mathrm{u(v)}| \rightarrow \mathrm{max}\ \&\ \mathrm{t(v)} \rightarrow \mathrm{min}\} \end{array} \right\} & \text{if } u_i > T_{up} \\ \emptyset & \text{Otherwise} \end{array} \right\} \tag{17}$$

$$t(v) = \frac{RAM(v)}{BWi} \tag{18}$$

Where u_i: host i's utilization; T_{up}: the upper threshold; *u(v)*:- CPU usage assigned to v; *P |u(v)|*:- the power utilized by VMs(v) in host *i*, and *t(v)*:- migration time.

4.4 Best Fit CPU, BW (BFCB) algorithm for VM Scheduling

BFCB algorithm is applied in two phases shown in **Fig. 3**. Over utilized host is scheduled on the available host in the first phase, and VMs are organized in decreasing order of the CPU utilization. The over-utilized VM in the host is checked against the normal host such that after the migration it must run in normal mode. To identify whether the host has enough CPU, Memory, and BW after the migration is calculated using the Mark Value of the Physical Machine List using the Equation (19).

$$Mark = \frac{CPU}{Allocated\ BW} \tag{19}$$

If none of the normal hosts satisfies the above condition, then a migration VM is checked against the under-utilized host. Even if an under-utilized host does not satisfy the requirements, a new host will be created to move the Virtual Machine.

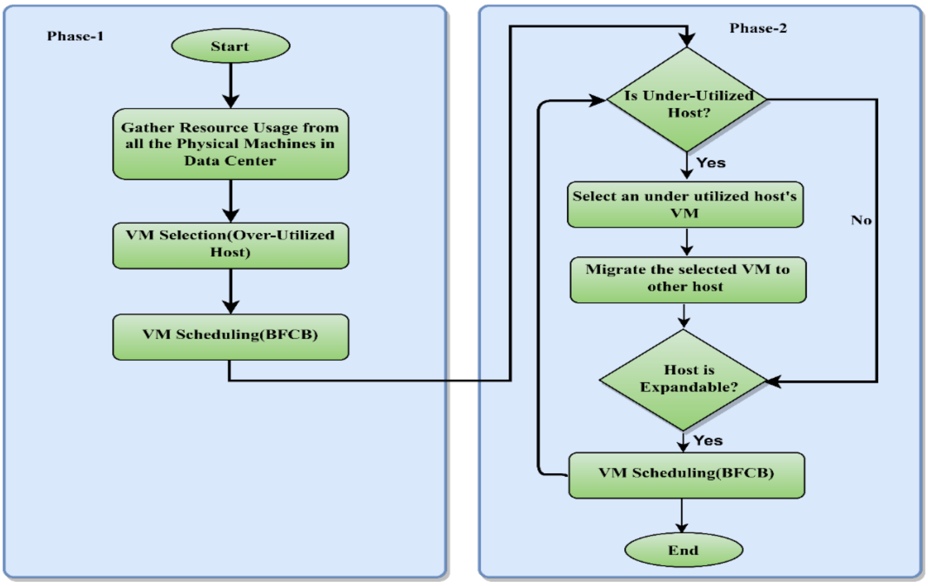

Fig. 3. Shows the Two-Phase BFCB Method

In the second Phase, the under-utilized host has to be moved to another host so that the host can be turned off and this helps in saving energy. This phase also uses the phase-1 strategy ie. Under-utilized VMs are cross verified against the normal host. If it is not satisfactory then it uses the second phase. In this, the BFCB algorithm first checks the VMList which consists of hosts which are raised for VM Migration requests. Finally, still if it is unable to find the existing host, then VM is been created on the new host using Algorithm 4.

Algorithm 4:- VM Scheduling

Input:- List of VMs from the host
Result:- Allocates VMs on the host

```
1.  Sort all the Over and Under-Utilized VMs are listed in decreasing order of CPU usage.
    //For Over-Utlized go to line-2 and for Under-Utilized go to line 21
2.  for each host in hostList do
3.      if ( lowerThreshold < currentUtil < preThreshold )
4.          normalHost <- host
5.      else if(curentUtil < lowerThreshold )
6.          underLoadedUtil <- host
7.      end if
8.  end for
9.  for  each VM in selectedVMList do
10.     find mark using Eqn. 11
11.     for each host in normalHost do
12.         Estimate the host if normal after VM Scheduling
13.         if(utilAfterScheduling < upperThreshold) do
14.             selectedHost <- host
15.         else
16.             Create new host
17.         end if
18.     end for
19. end for
20. for each host in hostList do
```

```
21.     if ( curThreshold < preThreshold )
22.        normalHost <- host
23.     else if(curentUtil < lowerThreshold )
24.        underLoadedUtil <- host
25.     end if
26.  end for
27.  for-each VM in selectedVMList do
28.     find mark using Eqn. 11
29.     for each host in normalHost do
30.        Estimate the host if normal after VM Scheduling
31.        if (utilAfterScheduling < upperThreshold) do
32.           selectedHost <- host
33.        else
34.           Create new host
35.        end if
36.     end for
37.  end for
38.  if (selectedHost==null) then
39.     for each h in receivedVMList do
40.       Go to line 30 to 37;
41.     end for
42.  end if
43.  return selectedHostList
```

4.5 The Proposed VM Scheduling for Efficient Dynamically Migrated VM (VMS-EDMVM) Approach

The state-of-the-art proposed VMS-EDMVM approach is a combination of 4 mechanisms: Host Over-Utilized detection, Host Under-Utilized detection, VM selection, 2-Phase VM Scheduling algorithm. VMS-EDMVM is a dynamic approach, because adaptive thresholds are used rather than fixed-value, making it applicable to real-world scenarios, since Workloads in Cloud datacenters are unpredictable. But this method is also adaptive because it automatically adjusts its behaviour based on the resource utilization record to predict varying workloads.

5. Results and Discussions

5.1 Experiment Setup

CloudSim [43] simulation tool has been used for VM Migration problems. Using this toolkit a software-based Cloud data center can be created for simulation. For the current simulation, datacenter with 500 hosts upon that 1050 VM was considered. The HP ProLiant G4 and G5 servers' power consumption model sets a PMs power consumption. The physical machine consumes power up to 86W at 0% CPU usage, and the maximum power consumption is 135W at 100% CPU usage. The dual-core processors for the PMs were chosen as it is easy to over-utilize a PM with a lesser workload. The data collected on power usage from the SPECpower [14] benchmark is shown in **Table 1** from both G4 and G5 servers.

Table 1. Data collected on power usage from SPECpower

Server	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
HP ProLiant G4	86	89.4	92.6	96	99.5	102	106	108	112	114	117
HP ProLiant G5	93.7	97	101	105	110	116	121	125	129	133	135

5.2 Workload Characterization

The workload used in our simulation consists of data generated from real systems which are the source to run the simulation. The workloads are based on Planet Lab's[44] real-world system data. Planet Lab contains real servers like HP and IBM servers used as benchmarks for the simulation. CoMon [45] collects workload data for every 5 minutes from 1000 VMs with 500 servers around the world that are used in this workload. **Table 2** shows the data collected between March-2011 to April-2011 with 200-250 experiments workload data of the VMs. In the workload data used for simulation, data considered is with CPU usage below 50%, and the VM assignments have been random during the simulation run.

Table 2. Represents the VMs count for different PlanetLab trace workloads

SI. No	Workloads	Number of VMs	Mean
1	20110303	1052	12.31%
2	20110306	898	11.44%
3	20110309	1061	10.70%
4	20110322	1516	9.26%
5	20110325	1078	10.56%
6	20110403	1463	12.39%
7	20110409	1358	11.12%
8	20110411	1233	11.56%
9	20110412	1054	11.54%
10	20110420	1033	10.43%

5.3 Proposed VMS-EDMVM Simulation

The proposed VM Scheduling for Efficient Dynamically Migrated VM (VMS-EDMVM) approach is simulated using a Java-based CloudSim simulation tool with the Planet Lab data for different trials. **Table 3** shows the proposed work results, which show the performance of different policies with the parameters Power Consumption, Total SLA Violation, and Total VM Migration, SLATAH, PDM, and ESV. **Table 4** shows the summary of the proposed mechanism's improvement percentages in comparison to the benchmark mechanisms.

Table 3. The Performance comparison of different policies

Methodology	Power Consumption (KWH)	Total SLA Violation (%)	Number of VM Migration	SLATAH (%)	PDM (%)	ESV ($X10^{-2}$) (%)
VMS-EDMVM	104.45	0.1	2202	1.82	0.05	0.22
ε-MOABC	105.24	0.17	6717	1.9	0.054	1.044
GM-DPSO	110.2	0.19	2303	3.41	0.055	0.375
PCM	117.33	0.2	4462	4.245	0.049	0.498
LR-RS	150.09	1.6	22791	6.41	0.086	0.967
LR-MC	150.33	1.6	23004	6.21	0.085	0.933
LR-MMT	163.15	1.5	27632	6.65	0.08	1.085

IQR-MC	177.1	1.6	23035	6.879	0.099	1.218
IQR-MMT	188.86	0.7	26476	5.023	0.065	0.948
LRR-MC	150.33	1.7	23004	7.64	0.103	1.148
LRR-MMT	163.15	1.4	27632	4.96	0.080	0.809
MAD-MC	176.13	1.6	23691	7.019	0.101	1.236
MAD-MMT	184.88	1.3	26292	5.052	0.65	0.934
THR-MC	174.19	1.7	22208	7.092	0.100	1.235
THR-MMT	207.32	1.8	29398	3.442	0.064	0.713

Table 4. Summary of proposed (VMS-EDMVM) mechanism's improvement percentages in comparison to the benchmark mechanisms

Methodology	Power Consumption (KWH)(%)	Total SLA Violation (%)	Number of VM Migration (%)	SLATAH (%)	PDM (%)	ESV ($X10^{-2}$) (%)
ε-MOABC	0.750665	41.17647	67.21751	1.733746	7.407407	78.9272
GM-DPSO	5.217786	47.36842	4.385584	22.08674	9.090909	41.33333
PCM	10.97758	50	50.64993	28.56302	15.25424	61.26761
LR-RS	30.40842	93.75	90.33829	67.13144	41.86047	77.24922
LR-MC	30.51952	93.75	90.42775	66.27415	41.17647	76.42015
LR-MMT	35.97916	93.33333	92.03098	67.78947	37.5	79.7235
IQR-MC	41.02202	93.75	90.44063	68.94625	49.49495	81.9376
IQR-MMT	44.69448	85.71429	91.68303	54.65715	23.07692	76.79325
LRR-MC	30.51952	94.11765	90.42775	71.69695	51.45631	80.83624
LRR-MMT	35.97916	92.85714	92.03098	58.78456	37.5	72.80593
MAD-MC	40.69721	93.75	90.70533	69.44098	50.49505	82.20065
MAD-MMT	43.50389	92.30769	91.62483	59.05354	92.30769	76.4454
THR-MC	40.03674	94.11765	90.08465	69.9645	50	82.18623
THR-MMT	49.61895	94.44444	92.50969	44.50578	21.875	69.14446

The proposed VMS-EDMVM approach is more efficient by comparing the policies of CloudSim such as LR-RS, LR-MC, LR-MMT, IQR-MC, IQR-MMT, LRR-MC, LRR-MMT, MAD-MC, MAD-MMT, THR-MC, THR-MMT, GM-DPSO[20], PCM[3], ε-MOABC[46]. As per the results, the proposed approach outperforms better as compared to the other policies against power consumption, total SLA violation rate, total VM Migration count, SLATAH, PDM, and ESV.

Different policies of Power Consumption are shown in **Fig. 4**. In this, the proposed VMS-EDMVM approach results better compared to other policies and this consumes a power of 104.45KWh which is the least power consumption compared with existing algorithms and mechanisms. THR-MMT has the highest power consumption, 207.32KWh and compared to this algorithm the proposed VMS-EDMVM approach consumes 49.62% less power. Hence VMS-EDMVM has been proven to be energy efficient based on the results arrived through the CloudSim simulation tool. The results of Total SLA Violation of different policies are shown in **Fig. 5**. The proposed algorithm is compared with the other methods. As per the results of the proposed VMS-EDMVM approach, SLA violation results in 0.1% which is less compared to all other methods and the proposed approach consumes 41.18% to 94.44% less. Based on the results the proposed VMS-EDMVM approach is the most efficient in managing SLA violations.

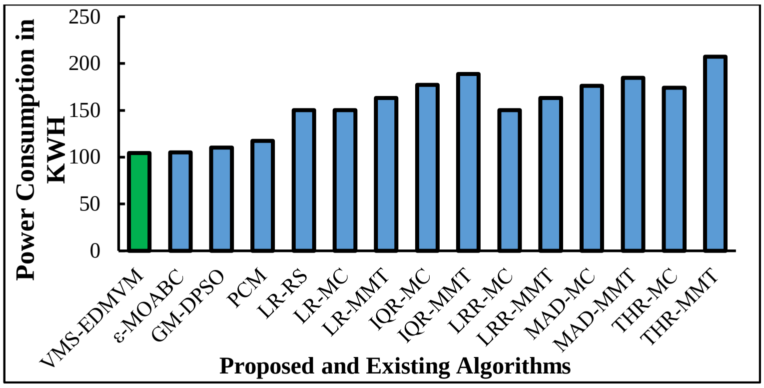

Fig. 4. Comparison of power consumption for proposed and existing Algorithms

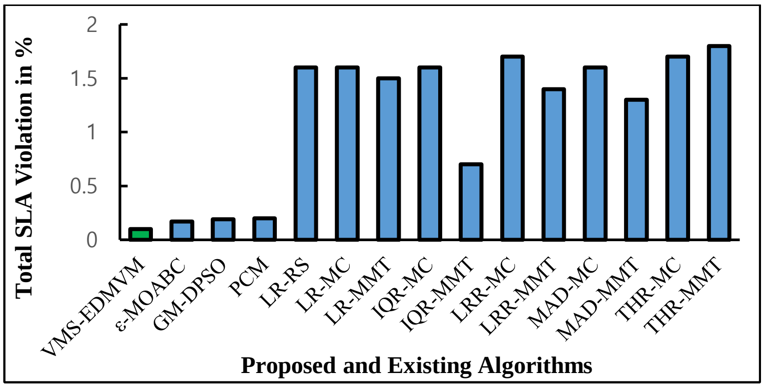

Fig. 5. Comparison of Total SLA violation for proposed and different algorithms.

The results of VM Migrations of different policies are shown in **Fig. 6**. The proposed VMS-EDMVM approach is compared with the other methods and has 2202 VM migrations count during the simulation. Based on the simulation, the VMS-EDMVM approach has a lower number of VM migrations. The proposed approach takes 4.38% to 92.5% less. Based on the results the proposed algorithm is efficient in managing VM Migrations. The results of SLATAH of different policies are shown in **Fig. 7**. The proposed approach is compared with the other algorithms and the proposed algorithm has 1.82% SLATAH during the simulation. Based on the simulation, the VMS-EDMVM approach has a lower SLATAH. The proposed algorithm takes 1.73% to 68.25% less. Based on the results the proposed approach is efficient in managing the SLATAH metric.

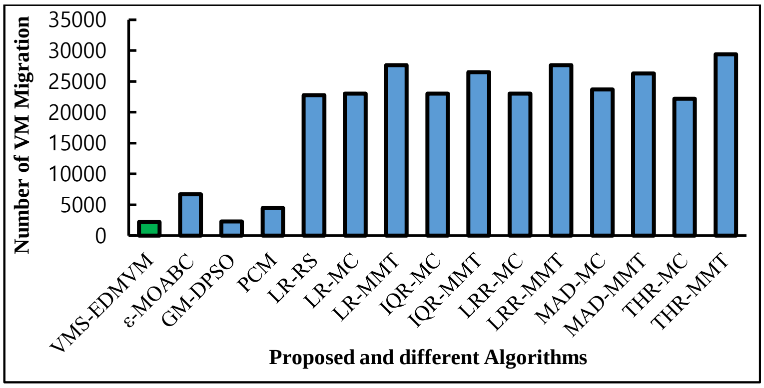

Fig. 6. Comparison of the number of VM migrations for proposed and different Algorithms.

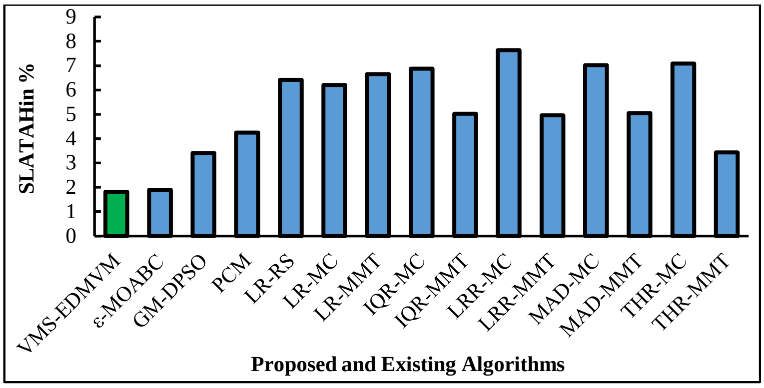

Fig. 7. Comparison of SLATAH for proposed and different algorithms

The results of PDM of different policies are shown in **Fig. 8**. The proposed approach is compared with the other algorithms and the proposed approach has 0.05% during the simulation. Based on the simulation, the VMS-EDMVM approach has a lower number of PDMs. The proposed approach takes 7.41% to 92.3% less. Based on the results the proposed approach is efficient in managing PDM. The results of ESV of different policies are shown in **Fig. 9**. The proposed approach is compared with the other algorithms and has 0.22% ESV during the simulation. Based on the simulation, the VMS-EDMVM approach has a lower number of ESV. The proposed approach takes 41.33% to 82.3% less. Based on the results the proposed approach is efficient in managing ESV.

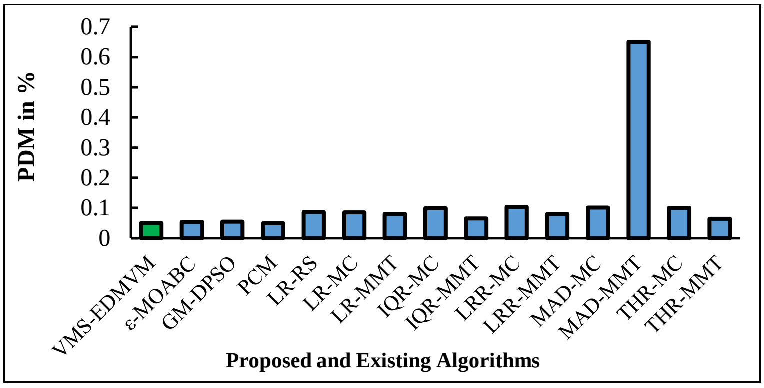

Fig. 8. Comparison of PDM for proposed and different algorithms

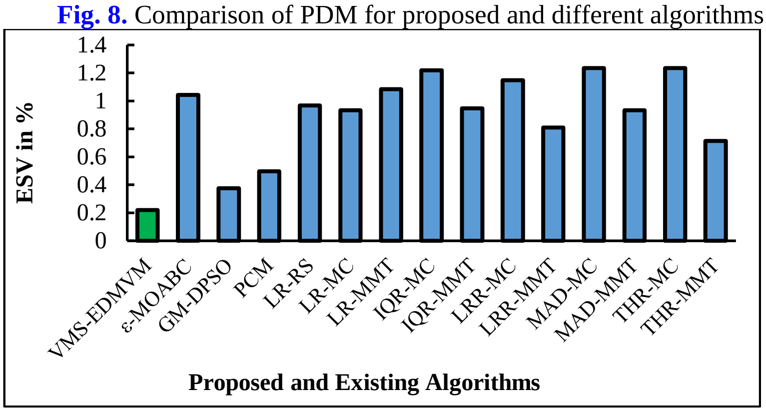

Fig. 9. Comparison of ESV for proposed and different algorithms

The results of all the performance metrics of different policies are shown in **Fig. 10**. The proposed VMS-EDMVM approach is compared with the other algorithms and yields improvement (%) as compared to the existing approaches.

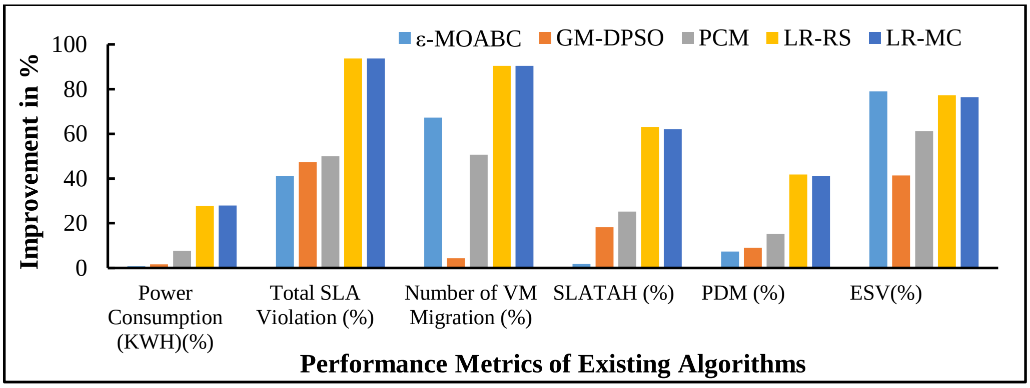

Fig. 10. Shows the comparison of improvement of VMS-EDMVM (in %) approach against some traditional approaches

Conclusion

In Cloud Computing, Cloud providers must intelligently manage the Data Center resources and it should not be overburdened to get the maximum utilization of the Cloud resources, to avoid Overburdening that impacts effective utilization of power consumption and SLA violation. Several algorithms are proposed on Power-Aware VM Consolidation by considering only CPU traces. But, in this research work, Cloud resources like CPU, RAM, and Bandwidth Utilization have been considered for calculating power consumption, SLA violations, and VM migrations count. To assess this, a series of operations have been performed and implemented using CloudSim by considering Dynamic VM consolidation calculations to improve the efficiency. First, a Modified Weighted Linear Regression Algorithm has been proposed to calculate the over-utilized host by calculating weights of the neighbor host using the square weight function and by implementing an under-utilized host detection algorithm for selecting the VM for Migration. A Two-Phase Best Fit CPU, BW (BFCB) Algorithm has been developed by marking CPU against the Bandwidth and reducing VM Migration considering the SLA Violations. Along with SLATAH, PDM, ESV are computed against most previous approaches. As per the obtained results, the proposed VMS-EDMVM approach simulates the SLA violation rate of 0.1%, Power Consumption of 104.45 KWH, the VM migrations being only 2202 times, SLATAH resulting in 1.82%, PDM to 0.05%, and ESV to 0.22%. Also, VMS-EDMVM results in 0.75%(ε-MOABC), 5.22% (GM-DPSO), 10.98%(PCM), 30.41%(LR-RS), 30.52%(LR-MC) better in terms of power consumption, 41.18%(ε-MOABC), 47.37%(GM-DPSO), 50%(PCM), 93.75%(LR-RS), 93.75%(LR-MC) better in terms of SLA Violation, 67.22%(ε-MOABC), 4.36%(GM-DPSO), 50.65%(PCM), 90.34%(LR-RS), 90.43%(LR-MC) better in terms of VM Migration count resulting in the most efficient solution for all the criteria. In the future, VM Scheduling on migrated VMs can be extended to Federated Cloud Computing.

References

[1] N. Dimitri, "Pricing cloud IaaS computing services," *J. Cloud Comput.*, vol. 9, no. 1, 2020. Article (CrossRef Link).

[2] T. Ciesielczyk et al., "An approach to reduce energy consumption and performance losses on heterogeneous servers using power capping," *J. Sched.*, vol. 24, pp. 489-505, 2021. Article (CrossRef Link).

[3] M. M. Taheri and K. Zamanifar, "2-Phase optimization method for energy aware scheduling of virtual machines in cloud data centers," in *Proc. of 2011 Int. Conf. Internet Technol. Secur. Trans. ICITST 2011*, no. December, pp. 525–530, 2011, Article (CrossRef Link).

[4] S. Supreeth and K. K. Patil, "Virtual machine scheduling strategies in cloud computing- A review," *Int. J. Emerg. Technol.*, vol. 10, no. 3, pp. 181–188, 2019. Article (CrossRef Link).

[5] M. Aldossary, "A Review of Dynamic Resource Management in Cloud Computing Environments," *Comput. Syst. Sci. Eng.*, vol. 36, no. 3, pp. 461–476, 2021. Article (CrossRef Link).

[6] M. A. L. I. Khoshkholghi, M. N. Derahman, and A. Abdullah, "Energy-Efficient Algorithms for Dynamic Virtual Machine Consolidation in Cloud Data Centers," *IEEE Access*, vol. 5, pp. 10709–10722, 2017. Article (CrossRef Link).

[7] M. Tighe, G. Keller, M. Bauer, and H. Lutfiyya, "A distributed approach to dynamic VM management," in *Proc. of 2013 9th Int. Conf. Netw. Serv. Manag*., pp. 166–170, 2013. Article (CrossRef Link).

[8] M. Tarahomi and M. Izadi, "A prediction-based and power-aware virtual machine allocation algorithm in three-tier cloud data centers," *Int. J. Commun. Syst.*, vol. 32, no. 3, pp. 1–20, 2019. Article (CrossRef Link).

[9] S. Supreeth & Kirankumari Patil, "Hybrid Genetic Algorithm and Modified-Particle Swarm Optimization Algorithm (GA-MPSO) for Predicting Scheduling Virtual Machines in Educational Cloud Platforms," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 17(07), pp. 208–225, 2022. Article (CrossRef Link).

[10] F. F. Moges and S. L. Abebe, "Energy-aware VM placement algorithms for the OpenStack Neat consolidation framework," *J. Cloud Comput.*, vol. 8, no. 1, 2019. Article (CrossRef Link).

[11] A. Verma, P. Ahuja, and A. Neogi, "pMapper: Power and migration cost aware application placement in virtualized systems," in *Proc. of Middleware 2008*, pp. 243–264, ACM/IFIP/USENIX 9th International Middleware Conference Leuven, Belgium, December 1-5, 2008. Article (CrossRef Link).

[12] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," *Futur. Gener. Comput. Syst.*, vol. 28, no. 5, pp. 755–768, 2012. Article (CrossRef Link).

[13] M. M. Hassan et al., "Efficient virtual machine resource management for media cloud computing," *KSII Trans. Internet Inf. Syst.*, vol. 8, no. 5, pp. 1567–1587, 2014. Article (CrossRef Link).

[14] 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," *Concurr. Comput. Pract. Exp.*, vol. 24, no. 13, pp. 1397–1420, 2012. Article (CrossRef Link).

[15] Y. Mhedheb, F. Jrad, J. Tao, J. Zhao, J. Kołodziej, and A. Streit, "Load and thermal-aware VM scheduling on the Cloud," in *Proc. of Algorithms and Architectures for Parallel Processing*, pp. 101-114, 13th International Conference, ICA3PP 2013, Vietri sul Mare, Italy, December 18-20, 2013. Article (CrossRef Link).

[16] L. Shi, J. Furlong, and R. Wang, "Empirical evaluation of vector bin packing algorithms for energy efficient data centers," in *Proc. of Int. Symp. Comput. Commun.*, no. July, pp. 9–15, 2013. Article (CrossRef Link).

[17] G. Keller, M. Tighe, H. Lutfiyya, and M. Bauer, "An analysis of first fit heuristics for the virtual machine relocation problem," in *Proc. of 2012 8th Int. Conf. Netw. Serv. Manag. CNSM 2012*, pp. 406–413, 2012. Article (CrossRef Link).

[18] G. Sun, D. Liao, V. Anand, D. Zhao, and H. Yu, “A new technique for efficient live migration of multiple virtual machines,” *Futur. Gener. Comput. Syst.*, vol. 55, pp. 74–86, 2016. Article (CrossRef Link).
[19] D. Li, W. Wang, Q. Li, and J. Cheng, “A Comprehensive Evaluation of Scheduling Methods of Virtual Machine Migration for Energy Conservation,” *IEEE Syst. J.*, vol. 11, no. 2, pp. 898–909, 2017. Article (CrossRef Link).
[20] U. Deshpande and K. Keahey, “Traffic-sensitive Live Migration of Virtual Machines,” *Futur. Gener. Comput. Syst.*, vol. 72, pp. 118–128, 2017. Article (CrossRef Link).
[21] G. J. L. Paulraj, S. A. J. Francis, J. D. Peter, and I. J. Jebadurai, “A combined forecast-based virtual machine migration in cloud data centers,” *Comput. Electr. Eng.*, vol. 69, pp. 287–300, 2018. Article (CrossRef Link).
[22] U. Deshpande, D. Chan, S. Chan, K. Gopalan, and N. Bila, “Scatter-Gather Live Migration of Virtual Machines,” *IEEE Trans. Cloud Comput.*, vol. 6, no. 1, pp. 196–208, 2018. Article (CrossRef Link).
[23] X. Ruan, H. Chen, Y. Tian, and S. Yin, “Virtual machine allocation and migration based on performance-to-power ratio in energy-efficient clouds,” *Futur. Gener. Comput. Syst.*, vol. 100, pp. 380–394, 2019. Article (CrossRef Link).
[24] D. Basu, X. Wang, Y. Hong, H. Chen, and S. Bressan, “Learn-as-you-go with Megh: Efficient Live Migration of Virtual Machines,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 8, pp. 1786–1801, 2019. Article (CrossRef Link).
[25] M. A. Altahat, A. Agarwal, N. Goel, and J. Kozlowski, “Dynamic Hybrid-copy Live Virtual Machine Migration: Analysis and Comparison,” *Procedia Comput. Sci.*, vol. 171, no. 2019, pp. 1459–1468, 2020. Article (CrossRef Link).
[26] Y. Shao, Q. Yang, Y. Gu, Y. Pan, Y. Zhou, and Z. Zhou, “A Dynamic Virtual Machine Resource Consolidation Strategy Based on a Gray Model and Improved Discrete Particle Swarm Optimization,” *IEEE Access*, no. Vm, pp. 228639–228654, 2020. Article (CrossRef Link).
[27] C. Jiang, L. Yang, and R. Shi, “An energy-aware virtual machine migration strategy based on three-way decisions,” *Energy Reports*, vol. 7, pp. 8597–8607, Nov. 2021. Article (CrossRef Link).
[28] S. Vatsal and S. Agarwal, “Energy-Efficient Virtual Machine Migration Approach For Optimization Of Cloud Data Centres,” in *Proc. of 2021 2nd International Conference for Emerging Technology (INCET)*, pp. 1–7, 2021. Article (CrossRef Link).
[29] L. H. Hung, C. H. Wu, C. H. Tsai, and H. C. Huang, “Migration-based load balance of virtual machine servers in cloud computing by load prediction using genetic-based methods,” *IEEE Access*, vol. 9, pp. 49760–49773, 2021. Article (CrossRef Link).
[30] Gonzalez, N., Carvalho, T. & Miers, C., “Cloud resource management: towards efficient execution of large-scale scientific applications and workflows on complex infrastructures,” *J Cloud Comp*, vol. 6, 2017, Article no. 13. Article (CrossRef Link).
[31] Supreeth S, & Shobha Biradar, “Scheduling Virtual Machines for Load balancing in Cloud Computing Platform,” *International Journal of Science and Research (IJSR)*, vol. 2, pp. 437–441, 2013. Article (CrossRef Link).
[32] N. Gupta et al., “Enhanced Virtualization-Based Dynamic Bin-Packing Optimized Energy Management Solution for Heterogeneous Clouds,” *Math. Probl. Eng.*, vol. 2022, 2022. Article (CrossRef Link).
[33] A. K. Bhoi, M. R. Kabat, S. C. Nayak, and G. Palai, “Renewable energy source based quality of service (QoS)-aware routing mechanism in cloud network,” *Wirel. Networks*, vol. 28, no. 4, pp. 1703–1718, 2022. Article (CrossRef Link).
[34] Z. Zhou, Z. Hu, and K. Li, “Virtual Machine Placement Algorithm for Both Energy-Awareness and SLA Violation Reduction in Cloud Data Centers,” *Sci. Program.*, vol. 2016, no. i, 2016. Article (CrossRef Link).
[35] B. Bouterse and H. Perros, “Performance analysis of the reserve capacity policy for dynamic VM allocation in a SaaS environment,” *Simul. Model. Pract. Theory*, vol. 93, pp. 293-304, 2019. Article (CrossRef Link).

[36] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," *ACM SIGARCH Comput. Archit. News*, vol. 35, no. 2, pp. 13–23, 2007. Article (CrossRef Link).
[37] X. Fu and C. Zhou, "Virtual machine selection and placement for dynamic consolidation in Cloud computing environment," *Front. Comput. Sci.*, vol. 9, no. 2, pp. 322–330, 2015. Article (CrossRef Link).
[38] J. Zhao, Y. Ding, G. Xu, L. Hu, Y. Dong, and X. Fu, "A Location Selection Policy of Live Virtual Machine Migration for Power Saving and Load Balancing," *The Scientific World Journal*, vol. 2013, pp. 1-17, 2013. Article (CrossRef Link).
[39] T. Deepika and N. M. Dhanya, "Multi-objective Prediction-Based Optimization of Power Consumption for Cloud Data Centers," *Arab. J. Sci. Eng.*, 2022. Article (CrossRef Link).
[40] S. Weisberg, *Applied Linear Regression*, vol. 528, Hoboken, NJ, USA: Wiley, 2005. Article (CrossRef Link).
[41] William S. Cleveland, "Robust Locally Weighted Regression and Smoothing Scatterplots," *Journal of the American Statistical Association*, Vol. 74, no. 368, pp. 829-836, 1979. Article (CrossRef Link).
[42] W. S. Cleveland and C. Loader, "Smoothing by Local Regression: Principles and Methods," *Statistical Theory and Computational Aspects of Smoothing*, pp. 10–49, the COMPSTAT '94 Satellite Meeting held in Semmering, Austria, 27–28 August 1994. Article (CrossRef Link).
[43] Calheiros, R.N., Ranjan, R., Beloglazov, A., De Rose, C.A.F. and Buyya, R, "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. Article (CrossRef Link).
[44] B. Chun et al., "PlanetLab: An overlay testbed for broad-coverage services," *Comput. Commun. Rev.*, vol. 33, no. 3, pp. 3–12, 2003. Article (CrossRef Link).
[45] K. Park and V. S. Pai, "CoMon: a mostly-scalable monitoring system for PlanetLab," *ACM SIGOPS Oper. Syst. Rev.*, vol. 40, no. 1, pp. 65–74, 2006. Article (CrossRef Link).
[46] M. H. Sayadnavard, A. Toroghi Haghighat, and A. M. Rahmani, "A multi-objective approach for energy-efficient and reliable dynamic VM consolidation in cloud data centers," *Eng. Sci. Technol. an Int. J.*, vol. 26, p. 100995, 2022. Article (CrossRef Link).

Mr. Supreeth S holds M. Tech. degree in Computer Science and Engineering from Visvesvaraya Technological University (VTU), Belgaum, and currently he is pursuing his Doctoral program (Ph. D.) in VTU and an Assistant Professor with the School of Computer Science and Engineering, REVA University, Bengaluru, India. He has been a technology educator for 10 years, teaching various subjects for UG and also PG programs. His research area includes Cloud Computing, Image processing, and Big Data. He has published several papers in international journals and presented many papers at International/National conferences in the domain of Cloud Computing, Fog Computing, Big Data and Image processing. He is a member of IEEE and the Indian Society for Technical Education (ISTE), Computer Society of India (CSI).

Dr. Kirankumari Patil is currently working as a professor in the School of CSE and Director, University Industry Interaction Center (UIIC) at REVA University, with 20 years of experience in mentoring and managing Software Product Design and development projects, Business Development and Business Incubation activities. Experienced in management and administration of Innovation and Entrepreneurship Development Activities. She has built & led highly effective teams to face research challenges and has been instrumental in Operations, Planning, Flawless Execution and Customer Relations. Project management, project planning, and resource management skills. Have successfully led teams in the past and managed deliverables to perfection. Organized and executed various Industry bond sponsored projects funded by IEEE Engineering Project in Community Services. Research Lead for Nokia Research Labs at REVA ITM, sponsored under the Nokia-VTU collaboration Project. She has managed 20+ research and skill development national and international projects.