

A Quantitative Approach to Minimize Energy Consumption in Cloud Data Centres using VM Consolidation Algorithm

M. Hema¹ and S. KanagaSubaRaja^{2*}

¹Easwari Engineering College, Chennai-89
Tamilnadu, INDIA

[E-mail : hemsit82@gmail.com]

²Easwari Engineering College, Chennai-89
Tamilnadu, INDIA

[E-mail : skanagasubaraja@gmail.com]

*Corresponding author : S. KanagaSubaRaja

*Received August 15, 2021; revised October 19, 2022; accepted January 30, 2023;
published February 28, 2023*

Abstract

In large-scale computing, cloud computing plays an important role by sharing globally-distributed resources. The evolution of cloud has taken place in the development of data centers and numerous servers across the globe. But the cloud information centers incur huge operational costs, consume high electricity and emit tons of dioxides. It is possible for the cloud suppliers to leverage their resources and decrease the consumption of energy through various methods such as dynamic consolidation of Virtual Machines (VMs), by keeping idle nodes in sleep mode and mistreatment of live migration. But the performance may get affected in case of harsh consolidation of VMs. So, it is a desired trait to have associate degree energy-performance exchange without compromising the quality of service while at the same time reducing the power consumption. This research article details a number of novel algorithms that dynamically consolidate the VMs in cloud information centers. The primary objective of the study is to leverage the computing resources to its best and reduce the energy consumption way behind the Service Level Agreement (SLA)drawbacks relevant to CPU load, RAM capacity and information measure. The proposed VM consolidation Algorithm (PVMCA) is contained of four algorithms: over loaded host detection algorithm, VM selection algorithm, VM placement algorithm, and under loading host detection algorithm. PVMCA is dynamic because it uses dynamic thresholds instead of static thresholds values, which makes it suggestion for real, unpredictable workloads common in cloud data centers. Also, the Algorithms are adaptive because it inevitably adjusts its behavior based on the studies of historical data of host resource utilization for any application with diverse workload patterns. Finally, the proposed algorithm is online because the algorithms are achieved run time and make an action in response to each request. The proposed algorithms' efficiency was validated through different simulations of extensive nature. The output analysis depicts the projected algorithms scaled back the energy consumption up to some considerable level besides ensuring proper SLA. On the basis of the project algorithms, the energy consumption got

reduced by 22% while there was an improvement observed in SLA up to 80% compared to other benchmark algorithms.

Keywords: Energy consumption, Data Centres, Virtual Machine, Physical Machine, SLA, Regression, VM migration, VM selection, VM Placement, Host Utilization.

1. Introduction

Over the past decade, cloud computing gained superior attention from various facets, thanks to its capability to provide utility-based Information Technology services to users across the globe via the internet. According to review of literature [1], cloud computing paradigm can be defined as a service model to produce measurable on-demand virtualized resources that functions on the basis of pay-as-you-go model. Various applications that start from scientific to business leverage the services of cloud in different forms like data, hardware and software. Some of the global MNCs such as IBM, Microsoft, Google and Amazon have their own cloud data centres that are located globally to render cloud services. The resources are allocated by the cloud data centres to consumers, through a mechanism that meets the promised Quality of Service (QoS), chosen by the subscribers when they enter into a Service Level Agreement (SLA) with the provider. In case of cloud environment, SLA is observed as a dual-mode agreement between the users and the cloud players. This SLA defines the services that fall under the service requirement, level of performance, price and penalty in case of non-supply of services [2].

Cloud infrastructures have become complex and complicated to handle owing to the quick development experienced in cloud services and their associated technologies. So, managing the resources remains the predominant challenge that needs to be addressed in modern cloud environments as it has a direct impact on the appropriate placement of cloud services. Though the data centres of present day yield excellent rewards to the stakeholders, energy consumption continues to be the biggest concerns faced in this field. Close to 1.5% of the electricity generated across the globe is consumed by data centres in the year 2010, which rose to 3% in the year 2016 [3]. The data centres maintained by Google consumed approximately 258 million Watts of electricity in the year 2013 which is equal to the power required for supplying 200,000 homes. On an average, 30% of the cloud data centre servers leverage close to 8-10% of the resource facility during functioning time. So, power -efficient resource management is the need of the hour to cut the operational costs as well as to reduce the environmental impact.

It is opined that when it comes to energy-efficient resource management, the VM consolidation is the most efficient technique for cloud computing since this technique increases the utilization of resources besides decreasing the energy consumption. The consolidation of VMs aims at its shifting to less number of hosts so that the passive hosts can be switched to sleeping modes [4]. In this scenario, the performance may get reduced if the VMs are consolidated aggressively with a sole aim of minimizing the energy consumption. This may result in unexpected and low performance from the system, thus affecting the quality of service which in turn may result in increase in SLA violations [5]. So it is a must

for the consolidation mechanism to quickly reduce the energy consumption without compromising on the performance and also by ensuring minimum SLA violations.

This research article proposes a vibrant, and adaptive energy-saving VM consolidation mechanism after a close observation of the SLA drawbacks for data centres in cloud. The present article focuses on the following aspects in this regard:

- Development of a host overload detection algorithm using regression model in the process of mitigating performance degradation risks and identifying two utilization thresholds
- Development of a power as well as SLA-aware VM selection algorithm with the help of three different policies to choose sufficient number of VMs that are integrated to another hosts
- Development of a Virtual Machine placement algorithm which can be leveraged for effective incorporation of novel Virtual Machines and the VMs chosen for the purposes of consolidation
- Development of a host under load detection algorithm with the help of a vector magnitude squared of multi resources in order to combine the non-passive hosts and change them to energy saving mode.
- Development of a energy-efficient Virtual Machine consolidation mechanism through the deployment of the modified algorithms based on how three host resources are utilized
- Run simulation for multiple numbers of times and perform the analyses for the proposed mechanism.

The present paper focuses on recent studies in the area in section two; section three, the system model for the present study is focused upon. In section four a detailed explanation of the proposed VM consolidation mechanism is provided; while section five illustrates the performance assessment of the mechanism proposed by the present study. Section six concludes by making summary as well as future scope based on the study.

2. Review of Literature

pMapper was designed by the researchers [6] as a power-aware workload placement controller to be utilized in case of heterogeneous virtualized server clusters. Various aspects are considered by this controller such as power consumption, migration costs and SLA requirements. pMapper comprises a power level manager, arbitrator, migration level manager and a performance manager. After considering the information supplied by above mentioned managers, the arbitrator decides the size essential for the VM's. DVFS is used by pMapper in addition to server power switching mode and consolidation of virtual machines as its power management policies. In a study conducted by Beloglazov et al. [7], the researchers projected an integration method which defined two fixed thresholds with regard to utilization of CPU. The CPU utilization increments above the upper limit threshold or else it gets reduced than the lower limit threshold. While there occurs the selection of some of the VMs that get migrated to another hosts. Having this reason being the basis, the study has been conducted to identify the finest set of lower and upper thresholds so as to mitigate Energy consumption coupled with low SLA violations. The authors also recommended the usage of three VM selection policies of which Minimization of Migration policy (MMP) is the first policy. In this policy, the lowest number of VMs gets consolidated to another host which ensures the CPU utilization is way ahead of the upper threshold. The VMs are selected randomly by following the Random Choice Policy (RC).

In case of Highest Potential Growth policy (HPG), the VMs are selected based on their least CPU utilization with regard to the whole lot of CPU capacity needed. Based on this data, it is inferred that the best results were achieved by the MMP with a least threshold level of 30%; while it was 70% for the upper threshold set.

This study shows that the results depict excellent results for both SLAV rate as well as energy consumption. But there are two disadvantages in this method - while the first being that the power consumption model considers the power usage of CPU only. The second disadvantage is the static consolidation mechanism since the thresholds are defined as fixed value. This issues drastically decrements the scalability of the approach in different workloads. But the mechanism proposed in this study considers the power consumed by whole components in the server into account.

In a study by [8]Beloglazov and Buyya, a dynamic integration of virtual machines was proposed in order to mitigate the power consumption and also for observing SLA violations in cloud data centres since the static thresholds seem to be inadequate for workloads of dynamic and unpredictable nature. The proposed mechanism leverages historical data on how resources are utilized in identifying the adaptive thresholds for every server. In determination of dynamic upper thresholds, three policies were introduced in this study. The LRP is developed on the basis of Loess method with an aim to calculate the upper limit threshold by identifying the regression curve that approximates the future data. There were two VM selection policies proposed in this research by the authors. The VMs are selected by the Minimum Migration Time (MMT) policy with the least possible time required for migration. The second policy i.e., Maximum Correlation policy (MCP) identifies the virtual machine with the highest correlation of utilization of CPU in addition to other virtual machines. The power consumed by all the server components is taken into account in this mechanism. But the drawback of this method is the presence of CPU being the only considered factor VM consolidation.

In a research conducted by Mhedheb et al. [9], a mechanism for consolidating the virtual machines on the basis of load and thermal-aware method to be used in cloud data centres was proposed. The primary motive of that study was to ensure equilibrium between the load and the temperature, thereby ensuring that the system does not undergo any hyper utilization or high temperature. The study also focuses on server energy consumption reduction. Thermal-aware Scheduler (ThaS) was proposed in the study with the help of DVFS as its power management technique. The virtual machines are scheduled in this method on the basis of temperature of the CPU. The study [10] proposed a Dynamic contract generation for service level agreement between Cloud service provider and consumer. The study [11] proposed a dual-phase mechanism for the consolidation of virtual machines in order to overcome incomplete migration challenges. Perplex VMs are nothing but VMs that need to be integrated since it is void of place in another hosts. So, the migration is terminated by the system which then moves the VMs to proper place. This issue resulted in the loss of capacity of CPU and power when the network experienced heavy overhead. As per the proposed framework, at the time of first phase, the virtual machines that were present in the over utilized hosts were migrated to less-loaded hosts. During the second phase, the virtual machines that were present in the Lower utilized hosts were diverted to another host.

Few researchers utilized the optimization problem of bin-packing mechanism when it comes to VM placement algorithms. In the study conducted by Shi et al [12], the researchers came up with online as well as offline VM placement algorithms through the modification of first fit bin-packing algorithms. The study conducted by Yen Wen et al [13], a proposal for various VM placement policies which make use of various combinations of server sorting methods and VM was made. The study results show that sorting in VM in increasing order can

enhance the energy usage when the servers are arranged in decreasing order to utilize VPU, leaving behind the other policies. According to the research [14], a technique was proposed to consolidate the VMs on the basis of dynamic machine learning which can lower the count of active hosts while at the same time, it can also optimize the utilization of resources at data centres. The proposed technique made use of a reinforcement learning method in order to acquire knowledge on optimal power mode to an agent. This was completed based on the previously available data and disconnecting the idle nodes.

In an earlier study [15], an energy-saving scheduling was proposed in order to attain high performance computing tasks in virtualized clusters. Both DVFS as well as server consolidation techniques were used in this research to mitigate the energy usage while at the same time optimize the acceptance ratio job. A CPU reallocation algorithm was proposed by the researchers [16] by integrating DVFS as well as Virtual Machine migration process to be energy-efficient in real time. The virtual machines are selected by this algorithm, based on which one needs to be transmitted in terms of heavy utilization of CPU, long time for completion and low utilization of CPU.

A Work flow scheduling was proposed by daeyong et al [17] aiming at decline in the consumption of energy with the help of live VM migrations, by taking SLA expectations at data centres into account. In order to achieve this aim, the WFS made use of two buffering levels such as local and global buffers to meet the workload differences. Close to 10% of the CPU capacity is denoted by the local buffer reserved in every server, while the CPU capacity with a reserved pool makes note of global buffer across all the servers. With the help of these servers, the number of migrations is minimized by WFS. Further the SLA violations are avoided through the allocation of the reserved CPU capacity to any increment in the demands, whenever a quick difference is experienced in the incoming workload. In case of a lesser local buffer than the threshold, few VMs get migrated to other servers. But for CPU capacity, this solution defines both local as well as global buffers, while overlooking the rest of the resources.

A distributed live VM migration mechanism was proposed by Wang et al [18] in cloud data centres. The load vectors were used in the mechanism proposed by this study by enabling every server to gather the data with regards to the incoming workload from the rest of the servers. The load vector takes into account various information such as destination index, source index and CPU utilization based on the amount of source. This An adaptive Three-threshold Energy-aware (ATEA) VM placement algorithm was proposed by Zhou et al. [19] aiming at reduction in both consumption of energy and SLAV in cloud data centres. The servers were classified by this algorithm into hosts with low, light, moderate and high loads. Three thresholds form the basis for this kind of classification; Thr_{low} , Thr_{mid} and Thr_{High} ($0 \leq Thr_{low} \leq Thr_{mid} \leq Thr_{High} \leq 1$), where, Thr_{low} , Thr_{mid} and Thr_{High} , denote the lower threshold, median threshold, higher threshold, and CPU utilization, respectively. The VMs are collected from different hosts of varying loads i.e., little load ($0 < Util < (Thr_{low})$) and high loads ($Thr_{High} \leq Util$) only. The results inferred that better results were achieved by KAM compared to KAI in terms of SLA metrics and energy reduction. Appendix I compares the results of other related works.

3. The System Model

The cloud is touted to be an IaaS environment that is globally distributed. Every server is characterized by network bandwidth, CPU performance and the RAM capacity. The performance of the CPU is denoted by Millions of Instructions per Second (MIPS). Network

Attached Storage (NAS), the common system storage in clouds, enables the dispersal of VM live migration. Otherwise, with the system being knowledge-free, the mechanism that is presented here is dependent on the workload. When multiple users submit the application required to the system, one or more heterogeneous VMs meet their needs.. There is a prism of workload types present in cloud applications starting from High-Performance Computing (HPC) to web-applications.

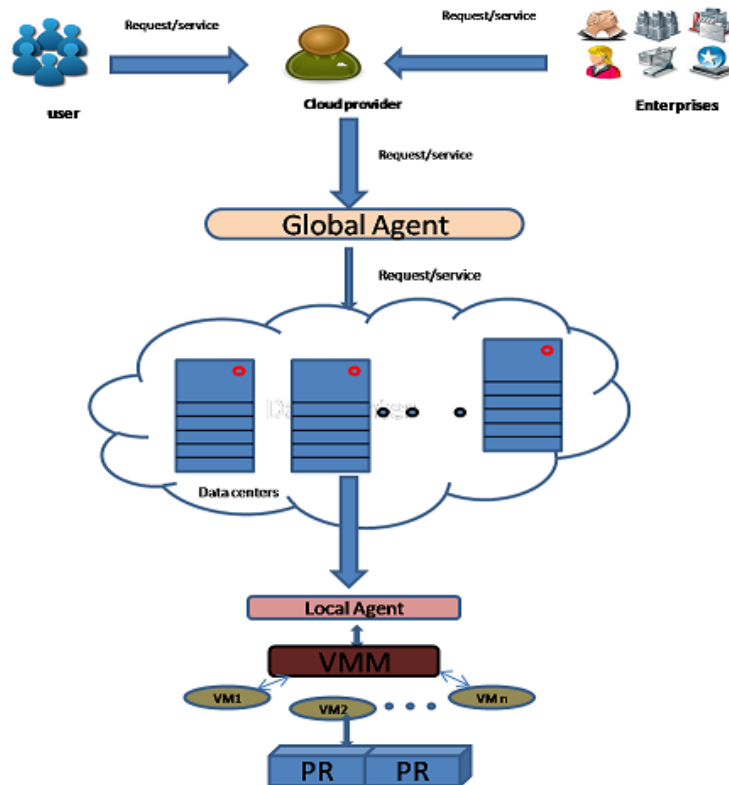


Fig. 1. System Architecture Diagram

There is an SLA contract signed between consumers and the Cloud Service Provider (CSP) upon the required QoS. Whenever an SLA violation occurs, then CSP needs to pay the penalty accordingly. **Fig. 1** shows the requests made by consumers to the global manager who is in-charge of meeting new demands and managing the transfer of VMs to available hosts. Every host has a local manager who is responsible for monitoring as well as management of host resources. The algorithms that consolidate the virtual machines are incorporated in this module. The host resources are monitored by the local broker who performs the decision-making according to the sources available. The Virtual Machine Manager (VMM) is the sole responsible person to decide which virtual machines need to be switched on or off.

By consolidating the VMs, the cloud service providers are able to optimize their resource utilization patterns besides being able to

o reduce the power consumption of data centres. There are four algorithms present in the proposed VM consolidation mechanism.

- Over loaded host detection: It differentiates at the time when hosts should be regarded as overloaded when either one or several VMs need live migration to another hosts to avoid over utilization of the host.
- Under loading host detection: It differentiates at the time which should be regarded as under loaded, when all the virtual machines are integrated with other hosts and the host is moved back to sleep mode.
- Selection of virtual machines: The utmost appropriate Virtual Machines that need migration are identified from the overloaded hosts. This is to avoid the reduction in the performance.
- VM placement: It finds the most optimum destination host for the chosen VMs.

Fig. 2. depict about overall flow diagram of proposed virtual machine consolidation algorithm.

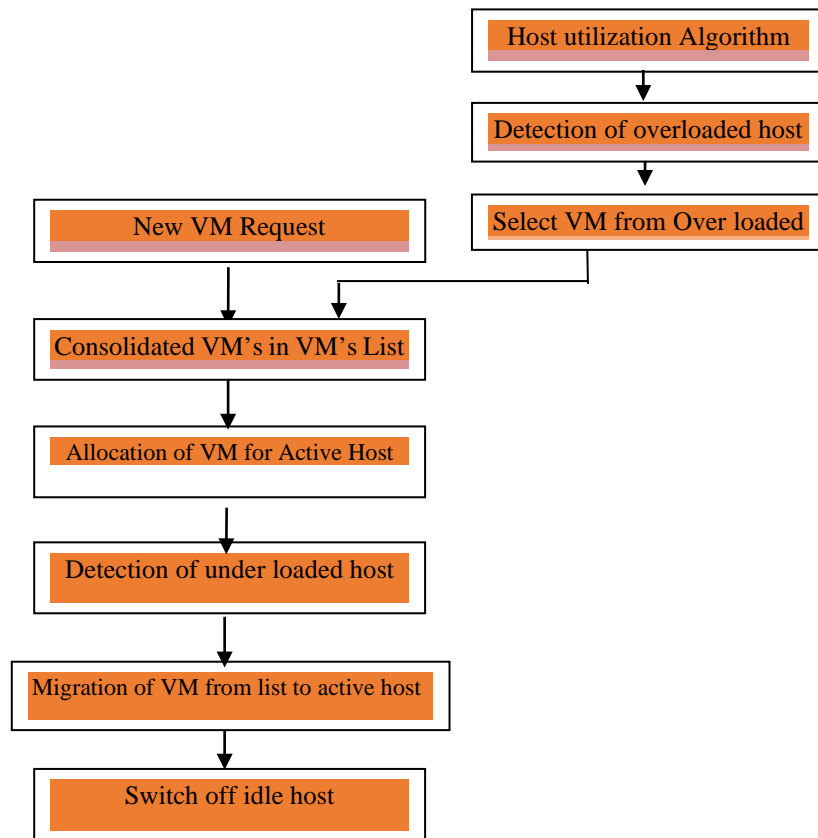


Fig. 2. Overall flow diagram of Proposed VM consolidation Algorithm

3.1 Host Utilization algorithm using Linear Regression

Regression is one of the statistical methods used for analysing the measurable data to forecast the forthcoming values of data. The regression analysis generally used for predictions in different areas of the study [21].

This statistical method classified as simple regression or multiple regressions, in which the former has only one input while the latter has more than one input. The purpose for conducting regression analysis is to approximate a regression function (linear or nonlinear) to determine the relationship that exists between input Z variable and output X variable by the regression line. A simple weighted linear regression is used by the proposed algorithm for forecasting future host utilization. Equation (1) shows the simple regression line.

$$Z = \alpha + \beta X \quad (1)$$

In the equation given above, X denotes the independent variable while Z denotes the dependent variable. The regression coefficients are α and β which are retrieved from least squares technique [20] as given herewith

$$\hat{A} = \hat{z} - \hat{\beta}\hat{x} \quad (2)$$

Where

$$\hat{\beta} = \frac{\sum_i^n (x_i - \bar{X})(z_i - \bar{Z})}{\sum_i^n (x_i - \bar{X})^2} \quad (3)$$

In the above equation, \hat{X} and \hat{Z} denote the means of X and Z, α observations, and are $\hat{\alpha}$ estimations of α and $\hat{\beta}$ respectively. In case of every observation (x_i, z_i) , a neighborhood weight is assigned with the help of tri cube function presented in the equations [22] and [23] as follows:

$$f_w(u) = \begin{cases} (1 - |u|)^3, & \text{if } |u| < 1 \\ 0, & \text{if } |u| > 1 \end{cases} \quad (4)$$

According to the formula given above, the neighbourhood weight can be defined as follows:

$$W_i = f_w(u) = \left(1 - \left(\frac{x_n - x_i}{x_n - x_1}\right)^3\right)^3 \quad (5)$$

Where x_1 and x_n are the first and the last observations, respectively. HULR uses N iterations to detect N future values of the host utilization. For n data values (previous host utilization), the linear regression line is defined as follows:

$$\begin{aligned} \hat{Z}_1 &= \alpha_0 + \beta_1 \hat{x}_n \\ \hat{Z}_2 &= \alpha_0 + \beta_1 \hat{Z}_1 \\ &\dots\dots\dots \\ \hat{Z}_n &= \alpha_0 + \beta_1 \hat{Z}_{n-1} \end{aligned} \quad (6)$$

The HULR calculates two utilization thresholds: the upper utilization threshold and pre-utilization threshold. Given an expected estimation of n, if the future host usage is anticipated to be higher than the absolute limit (100%) then that host is considered as overloaded host. When the host resource utilization is more than the absolute capacity (100%), HULR detects that the host is pre utilization limit threshold using the values $(i=2, i=k)$ the host is set as under tension; for this situation the host doesn't acknowledge any new VMs.

In the proposed host utilization algorithm, The value k is set as 2($k=2$) to predict two values.

$$\hat{Z}_1 = \alpha_0 + \beta_1 x_n$$

$$\hat{Z}_2 = \alpha_0 + \beta_1 \hat{Z}_1 \quad (7)$$

Algorithm 1 and 2 depict host utilization and host overload detection based on utilization.

Algorithm 1: Host Utilization algorithm	
Input:	(PM/VM) host utilization
Output:	Upper Utilization Limit, Pre Utilization Limit
step 1:	for every $i=1$ to n do Assign $x_i \leftarrow i$; Assign $z_i \leftarrow \text{History}(i)$; Calculate w_i using equation(5) Calculate $\bar{x}_i \leftarrow x_i * w_i$ Calculate $\bar{z}_i \leftarrow z_i * w_i$ End for
Step 2:	calculate the value of α , using equation (2)
Step 3:	calculate the value of β , using equation (3)
Step 4:	Pre Utilization Limit = $\alpha + \beta * \text{Currutil}(\text{host})$
Step 5:	Upper Utilization Limit = Pre Utilization Limit
Step 6:	Update x, z, w
Step 7:	update α and β
Step 8:	for every $i=2$ to k do KpredictUtil (i) = $\alpha + \beta * \text{Pre Utilization Limit}$ Pre utilization limit \leftarrow KpredictUtil (i) End for
Step 9:	return Upper Utilization Limit
Step 10:	return Pre Utilization Limit

Algorithm 2: Host overloaded Detection Algorithm	
Input:	host list
Output:	Overloaded host list, host pressure list
Step1:	UUC \leftarrow HULR(CPU).Upper Utilization Limit;
Step 2:	PULC \leftarrow HULR (CPU).Pre Utilization Limit;
Step 3:	UUM \leftarrow HULR (Memory).Upper Utilization Limit;
Step 4:	ULM \leftarrow HULR (Memory). Pre Utilization Limit;
Step 5:	UUB \leftarrow HULR (BW).Upper Utilization Limit;
Step 6 :	PULB \leftarrow IWLR(BW). Pre Utilization Limit
Step 7:	if ((PULC or PLUM or PULB) ≥ 1) then host Pressure List \leftarrow host; else if ((UUC or UUM or UUB) ≥ 1) then Overloaded host List \leftarrow host; end if
Step 8:	return host Pressure List;
Step 9:	return Overloaded host List

3.2. VM Selection Algorithm

The complete sets of overloaded hosts are first detected in the first part of the consolidation mechanism as described earlier. Following this, one or more than one VM is chosen from the detected host with the help of VM selection algorithm so that the utilization of the host gets reduced below the threshold level. Being an iterative algorithm, after each VM gets selected, the host resources' utilization is rechecked. When the host still remains overloaded, high numbers of VMs are chosen. This section proposes three policies for this algorithm as shown in algorithm 3.

VM Consolidation - Selection Algorithm
Input: Overloaded host List, VM List Output: SVM List Step 1: for every host in overloaded Host List do Step 2: for every VM in VMList do Step 3: SVM←NULL; Step 4: SVMList← SVM; Step 5: end for Step 6: currCPUutil←currCPUutil - SVMCPUutil; Step7: currRAMutil←currRAMutil - SVMRAMutil; Step 8: currBWutil←currBWutil - SVMBWutil; Step 9: if((currCPUutil< Upper Utilization Limit) && (currRAMutil< Upper Utilization Limit) && (currBWutil< Upper Utilization Limit)) then Step10: break; Step11: Else Step 12: VMList← VMList- SVM; Step13 : go to step 2; Step 14: end if Step15: End for Step 16: return SVMList;

3.2.1. Maximum Power Reduction Policy (MRP)

The MPR policy chooses and makes a VM migration that minimize the physical machine power consumption after migrating more than other VMs allocated to the PM. Let be a set of VMs allocated to the host , then this policy defines a set as in (8).

$$s = \begin{cases} M \setminus M \in VM_i, util_j - \sum_{v \in M} util(v) > T_{upl}, if |M| \rightarrow min, if util_j > T_{upl} \\ p|util(v)| \rightarrow max, & if util_j > T_{upl} \\ \emptyset, & otherwise \end{cases} \quad (8)$$

Where $util_i$ = resource utilization of VM.

$Thr_{upper\ limit}$ = upper limit threshold

$util(VM)$ = fractional load of CPU allotted for VM

$p(util(vm))$ = power consumption of VM

3.2.2 Time and Power Trade Off Policy

The (TPT) policy chooses and makes a VM migration which has the best trade-off between the smallest amount of migration time and the power drop after Virtual Machine

migration compared to the other Virtual Machines allocated to the PM (host). Let a set of VMs be allocated to the PM (host), then the TPT policy defines a set as in (9).

$$s = \begin{cases} M \setminus M \in VM_i, util_j - \sum_{v \in M} util(v) > T_{upl}, if |M| \rightarrow min, if util_j > T_{upl} \\ p|util(v)| \rightarrow max, t(v) \rightarrow min if util_j > T_{upl} \\ \emptyset, otherwise \end{cases} \quad (9)$$

Where $util_i$ = resource utilization of VM.

T_{upl} = upper limit threshold

$util(VM)$ = fractional load of CPU allotted for VM

$p(util(vm))$ = power consumption of VM

$t(vm)$ = vm migration time

$$VMmigration = \frac{RAM(VM)_i}{Active BW_{host}} \quad (10)$$

3.2.3 The Proposed VM Selection Policy

In this policy uses a various mechanisms to choose the virtual machines to be migrated. PVMSP choose all the virtual machines which have a CPU violation in the host. A Virtual machine is allocated with less CPR MIPS than it requested will be chosen and migrated to another host. . Let be a set of VMs M allocated to the host, then the PVMSP policy defines a set as in (11).

$$s = \begin{cases} M \setminus M \in VM_i, util_j - \sum_{v \in M} util(v) > T_{upl}, if |M| \rightarrow min, if util_j > T_{upl} \\ \frac{u_{al}(v)}{u_{req}(v)} < 1, if util_j > T_{upl} \\ \emptyset, otherwise \end{cases} \quad (11)$$

3.3 Virtual Machine(VM) Placement Algorithm

Once the hosts are detected as overloaded then the sufficient numbers of VMs are selected, this step requires the VM to be placed in good destination hosts. The aim is to find out the best hosts so that the chosen VMs can perform incurring minimum energy while committing very few SLA violations. The Virtual Machines-Proposed Placement Algorithm (VMPPA) is proposed to achieve this purpose.

3.3.1 VM Proposed Placement Algorithm (VMPPA)

There are two stages in the VMPPA process - the first stage is to facilitate migration of the chosen VMs from hosts overloaded list; and the second stage involves placing of the migrated VMs from under-loaded hosts. All the chosen virtual machines are placed in the order of decreasing fashion in the first stage, as per RAM capacity. From the list, the virtual machines are cross verified against the hosts to arrive at the best match. **Fig. 3**, shows that one can characterize four types of hosts on the basis of how they are utilized such as overloaded, under loaded, under pressure and normal. The last one i.e., normal hosts are those hosts in which the resource utilization is higher than low limit threshold while lower than the upper limit threshold.

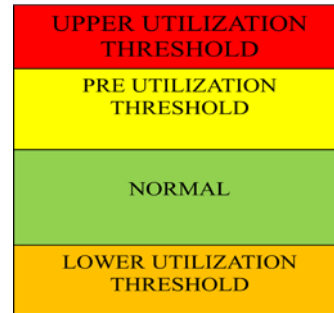


Fig. 3. Various Host Utilization

The VMPPA cross verifies the hosts in the normal list. When any host lacks sufficient resources for VM and in the event of not being heavily loaded once the placement is over, then the load of host ($Hostload_i$) is determined on the basis of equation (12) for those hosts; At the end, the host with less Load is selected as the target

$$Hostload_i = \frac{VMUtil(CPU)_i + VMUtil(RAM)_i + VMUtil(BW)_i}{HostUtil_i} \quad (12)$$

In the situation discussed above, no host has the ability to be chosen from the ordinary rundown. The VMPPA investigates the hosts on under-loaded rundown and then rehashes the strategy. When there is no competent host present on the under loaded into another host is projected. This methodology is followed in case of all the selected. Calculation 5 shows the VMPPA calculation in its principal stage.

In order to ensure that all the selected VMs are present in the main stage and are recognized as under loaded hosts, the selected VMs are moved from the under loaded host in the next stages. The first segment of VMPPA in the next stage is equivalent to the main stage. To start with, the VMPPA verifies ordinary hosts as discussed earlier in the primary stage.

In any kind of scenario, the important thing is the point when the calculation is unable to find the goal in the middle of ordinary hosts. The under loaded host gets separated into two records such as received VM list and hosts' list. While the former refers to the hosts that received any VM during the main stage, the latter did not receive VM during the primary stage.. The VMPPA explanation takes an initiative to switch off the under loaded host. But as predicted, when the VM migrations are increased, it may bring SLA infringement due to interference in the performance. So, the VMPPA tries to reduce the migration of VMs in the next stages. For this point, the VMPPA first analyses the hosts on the received VM list. In this scenario, not even a single satisfactory host can be selected when the hosts' lists are evaluated. Another host will be projected when chosen VMs not present. Calculation 6 shows how VMPPA is calculated in the next stages. This calculation can be used for VM union and asset portion so as to conduct independent calculation.

3.4 Under loading Host Detection Algorithm

During the identification of over-burdened host, the under-loaded host need to be resolved by selection and migration of VMs to different hosts. The combination part proposed herewith, is a strong tool to different sorts of remaining tasks at hand, there is an expectation for a true versatile technique that can decide the lower edge and identify the under loaded host. The MRULHDA calculation is thus proposed.

3.4.1 Multi Different Resources Under load Host Detection Algorithm (MRULHDA)

The lower quartile (Q1) of the past host is decided by MRULHDA as the lower edge is used. Q1 denotes the middle section in the lower half of the informational calculation. In line with this, the lower limit is shown as $\text{Thr}_{\text{Low}} = \text{util}(1/4(n+1))$, where util denotes the host usage and n denotes the quantity of information valued by the information collection.

In case of a low usage for CPU, RAM and BW then, the host can be said as under loaded. The vector extent squared (13) is calculated out the value of different measurements which are provided to sort out the under loaded has in expanding request.

$$VMG = \sqrt{X^2 + Y^2 + Z^2} \quad (13)$$

Where X=Host util(CPU), Y=Host Util (RAM), Z=Host util(BW)

At present, there is a rundown of under-loaded host in the framework though it is not possible for a whole number of hosts to be combined. To begin with, the framework must ensure that the whole set of virtual machines of every host can be shifted to other such hosts. In line with this, the framework keeps on testing the probabilities of solidifying the whole set of VMs to other vibrant host prior to the induction of live relocations. To acknowledge any VM, there must be three conditions, a host must fall into such categories: (1) it should feel the squeeze, (2) VM should have sufficient resources in it, (3) while conceding VM, no over-burdening should occur. When all the other dynamic hosts concede the whole set of VMs, there is a change for the host to be in the rest mode. when the VMs get remembered for the movement list; the host remains dynamic in any case. This procedure is iteratively rehashed in case of all under loaded host.

4. The proposed Virtual Machine Consolidation Mechanism (PVMCA)

There are four calculations involved in the proposed VM union component such as under loading host identification calculation, VM determination calculation, over-burdening host discovery calculation and VM arrangement calculation. PVMCA remains a mix of best-displayed calculations among the four subsystems that are discussed earlier. PVMCA can be considered to be dynamic since it leverages dynamic limits instead of fixed-esteem edges that make it without any doubt a flighty outstanding burden that is normal in cloud situations. Besides, this is a versatile system on grounds and it generally modifies its conduct on the basis of examinations of recorded asset use information for any application with various outstanding task at hand designs. In the long run, the proposed system is online due to the fact that calculations are performed for run time and activity is conducted due to each solicitation.

VM Placement Algorithm.

Input: Host List, SVM list

Output: Resalloc of VMs

Step 1 : SVM list sort Dec ();

Step 2 : For every host in host List do

Step 3 : if (lower utilization limit <CurrUtil< pre utilization limit) then

Step 4 : Normal_HostList ← host;

Step 5 : else if (CurrUtil< lower Utilization limit)

```

Step 6:           Under_loadedhostList ← host;
Step 7:           end if ;end for;
Step 8:           For every VM in SVMlist do
Step 9:             miniLoad ← MAX;
Step10:            Selected_Host ← null;
Step 11:           For every h in Normal_Host do
Step 12:             calculate utilAfterPlacement;
Step 13:            if (utilAfterPlacement<upperUtilization limit) then
Step 14:              Calculate Load by Equ. (13);
Step 15:              if (Load <minLoad) then
Step 16:                Selected_Host ← host;
Step 17:                Selected Hostlist ← selected_Host;
Step 18:                Mini Load ← Load;
Step 19:            end if ;end if; end for;
Step 20:            if (selected Host= null) then
Step 21:              for each host in underloaded_HostList do
Step 22:                line 13 to 19;
Step 23:              end for;
Step 24:            end if;
Step 24:            if (selected Host= null) then
Step 25:              Selected Hostlist ← new host;
Step 26:            end if;
Step 27:           end for;
Step 28:           return selected_hostlist ;

```

5. Execution Evaluation

At present, the reproduction in the after impact of the proposed VM solidification instrument is discussed herewith. The proposed system is exposed to usual cloud conditions, (IaaS for instance). So, the virtualized server farm foundation needs to be evaluated for huge scope. As the VM solidification component needs to be surveyed on a repetitive assessment mode, it is challenging to lead the trail on genuine cloud condition. So, it is attractive to recreate the same so as to assess the proposed component. CloudSim toolbox [24] is chosen as the re-enactment stage as it strengthens the vital effective techniques in cloud asset provisioning. In addition to that, the capacity is also underpinned so as to recreate the applications with dynamic outstanding tasks in hand.

5.1 Test Setting

In order to evaluate the proposed VM union system, the researchers mimicked a server farm which had a total of 800 physical servers in it. Being heterogeneous server, there are two server designs present in which the first half is composed of “HP ProLiant ML110 G4 (Intel Xeon 3040, dual core 1860 MHz, 4 GB, 1 Gbps)” while the rest is made up to “HP ProLiant ML110 G5 (Intel Xeon 3075, dual core 2660, 4 GB, 1 Gbps)”. The aim of this experimentation is to understand the impact of the proposed VM combination instrument. The servers that are limited with less assets are increasingly advantageous based on the fact that the servers may get heavily burdened quickly through lighter remaining tasks in hand. It is possible to run the VMs in any centre as it is not connected with any specific centre. The current research used

four kinds of VMs, compared to Amazon EC2 occurrence types, such as “Micro occasion (500 MIPS, 613 MB), Small Instance (1000 MIPS, 1.7 GB), Extra enormous Instance (2500 MIPS, 3.75 GB), High-CPU Medium Instance (2500 MIPS, 0.85 GB)”. With the help of NAS, the live migration of VM occurs in the framework without any justification to utilize direct-appended capacity. This kind of capacity further reduces the relocation overhead since there exist no solid reason to duplicate the circle content.

As the recreation process with the help of information gained from genuine frameworks can be progressively appropriate, the rest of the genuine burden is collected by the CoMon framework and is utilized in the reproduction. The CoMon venture [25] produces a framework for PlanetLab [26] and waits from it to provide data with regards to facts and information to two clients and heads. The remaining tasks are collected by CoMon in regular intervals, about the hand information for 400-450 dynamic PlanetLab hubs, and 201-250 dynamic tests running on PlanetLab. The remaining task at hand ensures the appropriate usage of assets using more than 900 VMs, in addition to 510 physical machines which are positioned somewhere else. The outstanding task of every VM is randomly committed to a VM at the time of re-enactment. The physical servers generally determine the asset utilization using VMs for every five minutes. The proposed system must be run like a clock down on the basis of data attained by outstanding tasks as discussed earlier. There occur multiple runs of trials for every calculation whereas the middle worth is found out. This appears in one of the presentation measurements.

5.2. Performance Metrics

The researchers made use of some exhibition measurements in order to estimate the impact of the proposed Virtual Machine consolidation system as given herewith: Vitality utilization, Energy and SLA Violation (ECSLAV), SLA infringement Time per Active Host (SLATAH), the quantity of VM movement, Performance Degradation due to Migration (PDM) and SLA violation. The PDM determines the corrupted framework execution performed by VM relocations using the following process.

$$PDM = \frac{1}{N} \sum_{i=1}^N \frac{R_{VM-AVM}}{R_{VM}} \quad (14)$$

Here N denotes the no.of VMs whereas R VM denotes the Request given by VMs in the host list. The allocated VM is denoted by AVM. SLATAH is characterized as the threshold upto which the dynamic hosts underwent whole i.e., 100% usage. It is important to know that the SLA is met when the overall exhibition, as sought by the applications within Virtual Machines, is practiced. In this situation, a host time is completely used i.e., 100% whereas the host may not be able to provide service to requested host. This alerts that there exists a SLA infringement. Following is the step to determine SLATAH.

$$SLATAH = \frac{1}{N} \sum_{i=1}^N \frac{F_i}{A_{host_i}} \quad (15)$$

Here N denotes the number of hosts and shows the absolute time when the whole set of hosts got completely utilized. Denotes the all-out time while the ‘has’ was maintained under dynamic mode. The integration of two earlier measurements offers the primary SLA infringement metric which can be characterized follows.

$$SLAV = PDM * SLATAH \quad (16)$$

This measurement determines both the execution corruptions that occurred due to VM movements and over-burdening of the hosts. As per the study conducted earlier [10], SLA and

e-contract and most of the vitality utilization of the servers is kept under control by CPU, RAM, power supplies cooling frameworks and circle stockpiling. Further, there were also few research studies [27] which mention that the power utilization of physical machines can be precisely characterized through a straight connection between force utilization and the usage of CPU. At present, the real information is collected from SPEC power benchmark [28] and is utilized as the information on power utilization for two different hosts used in the re-enactment. In Table 1, the gathered data on power utilization at the re-enactment is shown. ECSLAV, the final measurement, evaluates the proposed VM union system on the basis of SLV infringement rate and vitality utilization. The ECSLAV is determined as given herewith.

$$\text{ECSLAV} = \text{Energyconsumption} * \text{SLAV} \quad (17)$$

It is not possible for the VMPPA to identify the final point in the set of normal hosts. In such a case, there will be two lists created for the under-loaded hosts such as the received VM list and the hosts' list. While the former refers to the hosts which have retrieved any VM during the first phase, the latter denotes the non-receipt of VMs in the first phase. This might be attributed to the reason that VMPPA attempts to call off the under loaded hosts to the maximum extent. However, when there is an increase in the VM migrations, the SLA violations are also enhanced due to interferences. So, the VMPPA attempts to reduce the number of Virtual Machine migrations in the next phase. To achieve this aim, the VMPPA, in the beginning, analyses the hosts that are received in the VM list. When there is no more adequate host to be chosen, the other hosts list is interrogated. Towards the end, in the absence of any more selected host, the launch of a new host occurs. In algorithm 6, the VMPPA algorithm in the second phase is shown. This algorithm is used for VM consolidation, resource allocation as well as for provisioning algorithm individually.

5.3 Simulation Results

The PVMCA should have only the topmost performing algorithms that are discussed in the previous section. So, to identify the most efficient policy, the three VM selection policies i.e., MPR, TPT and PVMSP were compared. In the Fig. 4, the authors showcase the energy consumed by the data centre, whereas the Fig. 5 shows SLAV in case of the presented Virtual Machine selection policies. The results inferred that the PVMSP attained the least SLA violation as well as consumed minimum energy when compared to the other policies. The PVMSP reduces the usage of energy by 5% and 18% in comparison with MPR and TPT respectively. Further, the SLAV got reduced by PVMSP up to 47% and 29% in comparison with MPR and TPT respectively. These results infer the fact that PVMSP algorithm is able to efficiently manage the utilization of the host so that the physical hosts achieve the highest capacity of their resource utilization. This is in parallel with the guarantee for meeting SLAs. This is possible only through the selection of sufficient VMs which need to be migrated.

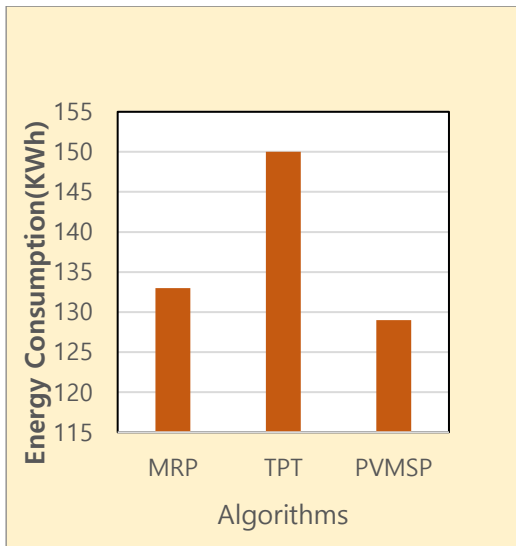


Fig. 4. Energy Consumption of PVMSP

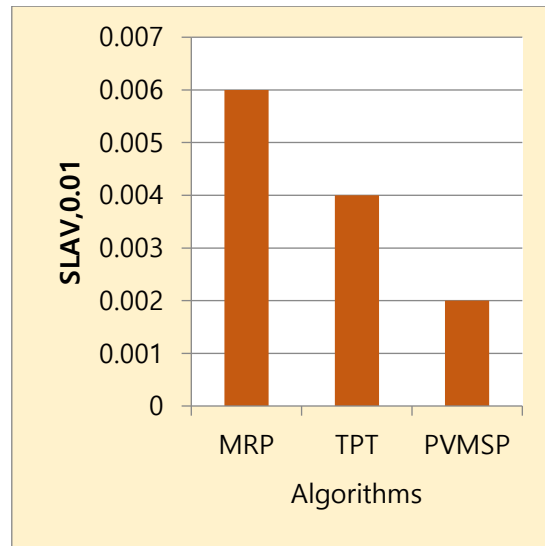


Fig. 5. SLA violation of PVMSP

According to the results attained above, it can be said that PVMCA has four algorithms in total such as HULP, PVMSP, VMPPA, and MRULHDA. In order to assess the PVMCA with three traditional algorithms [8]. **Fig. 6** shows the number of VM that got migrated by PVMCA in comparison with the benchmark algorithms. The results infer the fact that PVMCA migrate only 4,459 VMs to other hosts when simulated, when compared to the other Traditional Algorithms. There was a drastic decrease in the number of VM migrations when using PVMCA

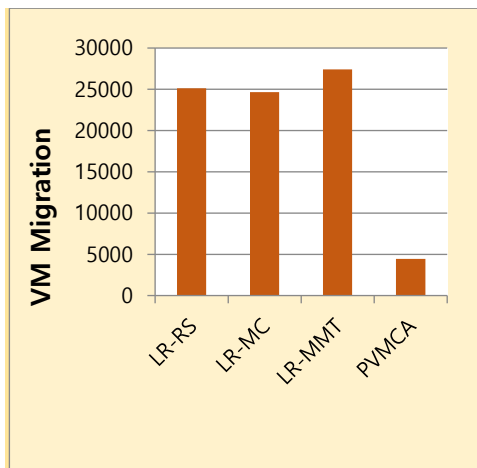


Fig. 6. VM Migrations using PVMCA Vs Traditional Algorithm

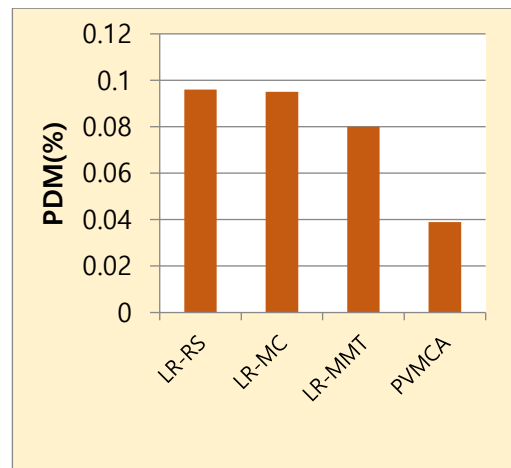


Fig. 7. PDM using PVMCM Vs. Traditional Algorithm

The live VM migration incurred heavy overheads for the system. The cloud administrators confined the number of VM migrations on the basis of acceptable Virtual Machine migration overhead. At that time, a preference for a mechanism which needs only less number of migrations to consolidate the VMs has arisen. The PVMCA outperformed other Traditional

algorithms such as LR-RS, LR-MC and LR-MMT by 81%, 82 and 84% in terms of number of shifting of virtual machines.

Table 1. Power consumption data from the SPEC power benchmark [8].

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

The experimental results for PDM are shown in **Fig. 7**. It can be observed that PVMCA has notably reduced the PDM in comparison with other mechanisms since the PVMCA considers the advantages of PVMSP and VMPPA algorithms. The PVMSP chooses the mandatorily-required VMs alone encounter with SLA violation needs to be immediately migrated. This problem is able to prevent the rate of more SLA violations for those Virtual Machines and can decrease the number of unwanted migrations too. Further, the PVMCA is supported by VMPPA as the VM placement algorithm, in terms of choice of destination hosts for not only CPU, but for RAM and BW as well. This way, an accurate placement is conducted and the replacement of VMs due to unsuccessful migrations is avoided. it is observed that the Traditional Algorithms got improved by PVMCA upto 59%, 57% and 51% in terms of PDM.

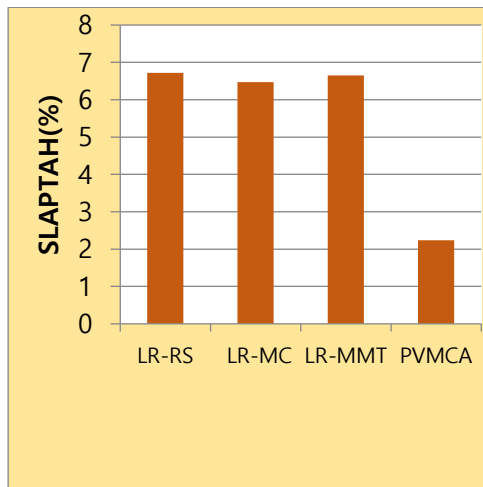


Fig. 8. SLAPTAH of PVMCA Vs Traditional Algorithms

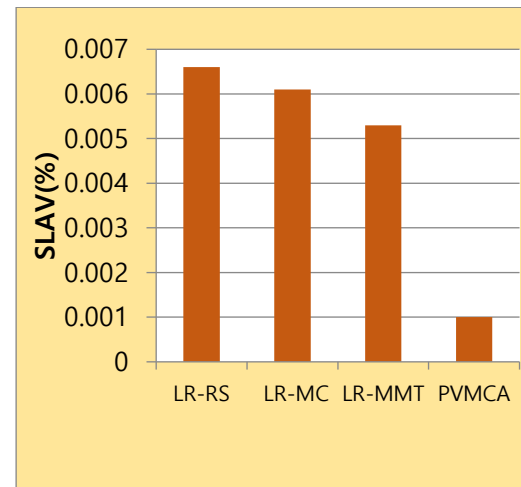


Fig. 9. SLAV of PVMCA Vs Traditional Algorithms

From **Fig. 8**, it can be observed that the performance of PVMCA is above the SLATAH than the Traditional algorithms. This phenomenon can be detailed by the fact that PVMCA is composed of IWLR algorithms. It has the ability to reduce SLATAH since its primary objective is to identify the overloaded hosts prior to the occurrence of a violation. Two thresholds are determined here in order to ensure that the host utilization increase does not create future violations. So the duration for which the hosts present with full capacity is reduced.

As per the discussions made in previous performance metrics, the PVMCA seemed to have improved other mechanisms in terms of PDM and SLATAH. The proposed mechanism is predicted to decrease the SLAV since the latter is an integration of PDM and SLATAH. In Fig. 9, the PVMCA is shown to have drastically reduced the number of SLA violations in comparison with the other mechanisms. In terms of SLAV, the PVMCA enhanced the traditional mechanisms by 86%, 85% and 84% respectively.

Energy consumption is the next performance metric and is shown in the Fig. 10. The PVMCA consumed only a meagre 117 kWh at the time of experimentation, thus providing excellent performance than the benchmarked mechanisms up to 22%. This can be attributed to the reason that MRULHDA algorithm is able to switch back the number of under loaded hosts to sleep mode. This resulted in low energy consumption when compared to idle state, which led to higher the energy savings by physical hosts.

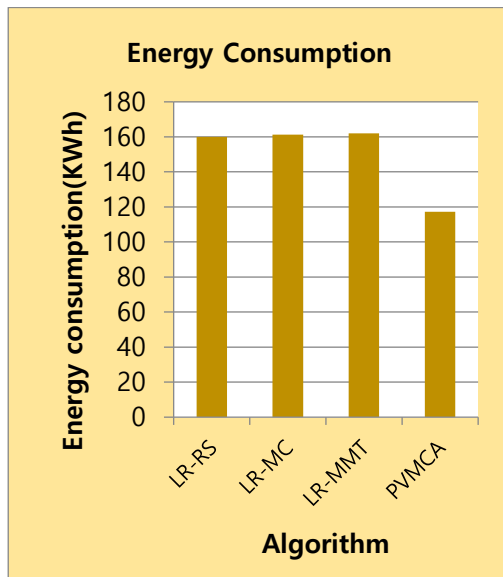


Fig. 10. Energy Consumption of PVMCA Vs Traditional Algorithm

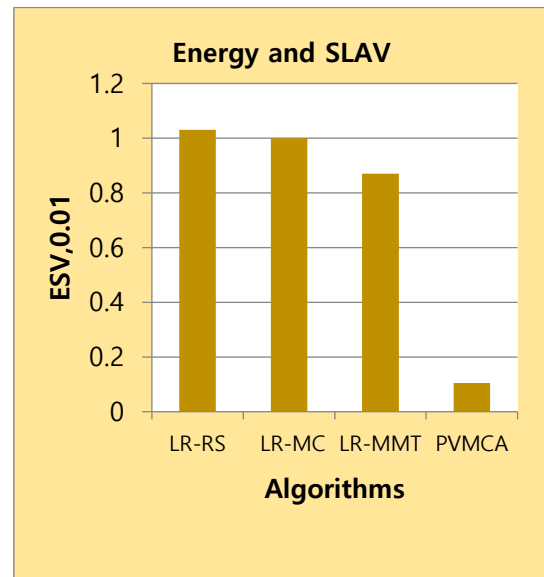


Fig. 11. Energy Consumption & SLAV of PVMCA Vs traditional Algorithm

Fig. 11 shows the ECVSLAV results attained by PVMCA and Traditional Algorithms. The performance of PVMCA was higher than other mechanisms considering SLAV (80%) and energy consumption (22%). So, the PVMCA is expected to enhance the ECVSLAV in comparison with other Traditional Algorithms. As per the results, the PVMCA outperformed Traditional Algorithms by 89%, 89% and 88% in terms of ECVSLAV. Table 2 shows simulation results of proposed Algorithm (PVMCA)

Table 2. Comparison of Simulation result of PVMCA VS Traditional Algorithm

Algorithms	LR-RC	LR-MC	LR-MMT	PVMCA
No.of.VM Migration	25108	24606	27408	4459
SLATAH	6.52	6.57	6.65	2.235
Energy consumption	159.05	160.33	160.93	116.13
ECSLAV($\times 10^{-2}$)	1.02	1	0.85	0.104
SLAV($\times 10^{-2}$)	0.0065	0.0061	0.0054	0.001
PDM	0.095	0.094	0.07	0.038

The summary of the experiment results that compared the PVMCA mechanism with other traditional Algorithms is shown in **Table 3**. The results inferred the fact that the performance of PVMCA was superior to other traditional algorithms in all the performance metrics considered for the study.

Table 3. Enhancement percentage for the PVMCA compared to the traditional Algorithm

Algorithms	LR-RC	LR-MC	LR-MMT
No.of.VM Migration(%)	81	82	84
SLATAH(%)	67	65	66
Energy consumption(%)	28	28	26
ESV(%)	89	89	88
SLAV(%)	86	85	84
PDM(%)	59	57	51

The PVMCA attained the least level of SLAV and the improvement was significant with 80%. The energy savings of PVMCA was phenomenal i.e., 28% higher than the mechanisms considered. These results prove that the proposed mechanism is excellent in terms of reducing energy consumption and maintaining low SLA violations, and justifies the primary target of this research. This improvement was achieved by the proposed mechanism by leveraging the benefits of every algorithm. Further, the PVMCA was able to consolidate the VMs on the basis of three resources such as CPU, RAM and BW, thus providing sufficient algorithms against Traditional Algorithms.

6. Conclusion and Future Work

The cloud providers can successfully optimize the utilization of their resources and reduce the consumption of energy through vibrant integration of Virtual Machines and switching back the idle servers to sleep mode. But when the consolidations of VMs are performed aggressively, it may result in reduced performance as well. There are various energy-efficient techniques proposed in the studies conducted earlier. However, the SLA violation rate still remains notably higher. Further, the current algorithm takes CPU as the sole factor into account when it comes to VM consolidation. Based on the analysis of previous studies, the current study proposed an energy-efficient Virtual Machine consolidation mechanism. The primary aim of this mechanism is to reduce the energy consumption of a data center while at

the same time, ensure the system performance is maintained without compromising SLA constraints with regards to CPU, RAM and BW. In order to assess the presented mechanism for consolidating the virtual machines, the researchers selected the simulation platform, CloudSim. The authors vigorously stimulated the model on a huge experiment setup with the help of workloads that are found from more than 1,000 Planet Lab VMs. The study results proved the mechanism proposed in this study is effective in comparison with other benchmark algorithms. Proposed algorithms (PVMCA) tries to make best potential utilization of a minimum number of the physical machine while make an attempt to allocate a maximum number of tasks to the active physical machines. As energy consumption is directly proportional to the number of active machines in a datacenter, increase in the number of host shutdown would result into reduction in energy consumption. The proposed method also considers the exclusion of hosts which are estimated to be vacated in near future during VM placement. This results into better workload balancing which evacuates more number of hosts. As can be seen from the table that even at low utilization, the host consumes a significant amount of power. Hence it is required to turn off such kind of hosts, when not in use. It can be inferred that PVMCA outperformed all the other algorithms when it comes to energy consumption (22%) and SLA (80%). To conclude, a performance-awareness strategy which has the capability to handle different workloads rendered by system-based applications can enhance the energy-saving virtual machine consolidation mechanism in cloud data centers. Though it is not a part of the current study, it can be considered as a scope for future research works.

References

- [1] Youssef Saadi, and Said El Kafhali, "Energy-efficient strategy for virtual machine consolidation in cloud environment," *Journal of soft computing*, vol. 24, pp. 14845-14859, 2020. [Article \(CrossRef Link\)](#).
- [2] B.P. Rimal, A. Jukan, D. Katsaros, and Y. Goeleven, "Architectural requirements for cloud computing systems: an enterprise cloud approach," *Journal of Grid Computing*, vol. 9, no. 1, pp. 3-26, 2011. [Article \(CrossRef Link\)](#)
- [3] J. Koomey, "Growth in data center electricity use 2005 to 2010," *A report by Analytical Press, completed at the request of The New York Times*, vol. 9, 2011.
- [4] S. Srikantaiah, A. Kansal, and F. Zhao, "Energy aware consolidation for cloud computing," in *Proc. of the Conference on Power Aware Computing and Systems*, vol. 10, pp. 1-5, 2008. [Article \(CrossRef Link\)](#)
- [5] R. Buyya, A. Beloglazov, and J. Abawajy, "Energy-efficient management of data center resources for cloud computing: a vision, architectural elements, and open challenges," in *Proc. of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA2010)*, pp.1-12, July 2010. [Article \(CrossRef Link\)](#)
- [6] A. Verma, P. Ahuja, and A. Neogi, "pMapper: power and migration cost aware application placement in virtualized systems," in *Proc. of the ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing*, pp. 243-264, December 2008. [Article \(CrossRef Link\)](#)
- [7] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future generation computer systems*, vol. 28, no. 5, pp. 755-768, 2012. [Article \(CrossRef Link\)](#)
- [8] 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 and Computation: Practice and Experience*, vol. 24, no. 13, pp. 1397-1420, 2012. [Article \(CrossRef Link\)](#)

- [9] Y. Mhedheb, F. Jrad, J. Tao et al., "Load and thermal-aware vm scheduling on the cloud," in *Proc. of the International Conference on Algorithms and Architectures for Parallel Processing*, pp. 101-114, December 2013. [Article \(CrossRef Link\)](#)
- [10] KanchanaRajaram, and UshaKirthika, "Dynamic Contract Generation and Monitoring for B2B Applications with Composite Services," in *Proc. of the International conference on Information and communication Technologies*, pp. 362-364, 2010. [Article \(CrossRef Link\)](#)
- [11] M. Taheri, and K. Zamanifar, "2-phase optimization method for energy aware scheduling of virtual machines in cloud data centers," in *Proc. of the IEEE International Conference for Internet Technology and Secured Transactions (ICITST)*, pp. 525-530, December 2011. [Article \(CrossRef Link\)](#)
- [12] L. Shi, J. Furlong, and R. Wang, "Empirical evaluation of vector bin packing algorithms for energy efficient data centers," in *Proc. of the IEEE Symposium on Computers and Communications (ISCC)*, pp. 9-15, July 2013. [Article \(CrossRef Link\)](#)
- [13] Yen-Wen Chen, Meng-Hsien Lin, and Min-Yan Wu, "Study of data placement schemes for SNS service in cloud environment," *KSII Transactions on Internet and Information Systems*, vol. 9, no. 8, August 30, 2015. [Article \(CrossRef Link\)](#)
- [14] F. Farahnakian, P. Liljeberg, and J. Plosila, "Energy-efficient virtual machines consolidation in cloud data centers using reinforcement learning," in *Proc. of the 22nd Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, pp. 500-507, February 2014. [Article \(CrossRef Link\)](#)
- [15] I. Takouna, W. Dawoud, and C. Meinel, "Energy efficient scheduling of hpc-jobs on virtualize clusters using host and vm dynamic configuration," *ACM SIGOPS Operating Systems Review*, vol. 46, no. 2, pp. 19-27, 2012. [Article \(CrossRef Link\)](#)
- [16] W. Chawarut, and L. Woraphon, "Energy-aware and real-time service management in cloud computing," in *Proc. of the 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pp. 1-5, May 2013. [Article \(CrossRef Link\)](#)
- [17] Daeyong Jung, Taeweon Suh, Heonchang Yu, and JoonMin Gil, "A Workflow Scheduling Technique Using Genetic Algorithm in Spot Instance-Based Cloud," *KSII Transactions on Internet and Information Systems*, vol. 8, no. 9, September 29, 2014. [Article \(CrossRef Link\)](#)
- [18] M.Y. Lim, F. Rawson, T. Bletsch, and V.W. Freeh, "Padd: Power aware domain distribution," in *Proc. of the 29th IEEE International Conference on Distributed Computing Systems (ICDCS'09)*, pp. 239-247, June 2009. [Article \(CrossRef Link\)](#)
- [19] X. Wang, X. Liu, L. Fan, and X. Jia, "A decentralized virtual machine migration approach of data centers for cloud computing," *Mathematical Problems in Engineering*, vol. 2013, pp. 1-10, 2013. [Article \(CrossRef Link\)](#)
- [20] Z. Zhou, Z. Hu, and K. Li, "Virtual Machine Placement Algorithm for Both Energy-Awareness and SLA Violation Reduction in Cloud Data Centers," *Scientific Programming*, vol. 2016, pp. 1-11, 2016. [Article \(CrossRef Link\)](#)
- [21] J.B. Guerard Jr, *Introduction to Financial Forecasting in Investment Analysis*, New York: Springer Science and Business Media, 2013.
- [22] S. Weisberg, *Applied linear regression*, vol. 528, John Wiley & Sons, 2005. [Article \(CrossRef Link\)](#)
- [23] W.S. Cleveland, "Robust locally weighted regression and smoothing scatterplots," *Journal of the American statistical association*, vol. 74, no. 368, pp. 829-836, 1979. [Article \(CrossRef Link\)](#)
- [24] W.S. Cleveland, and C. Loader, "Smoothing by local regression: Principles and methods," *Statistical theory and computational aspects of smoothing*, pp. 10-49, 1996. [Article \(CrossRef Link\)](#)
- [25] R. Buyya, and M. Murshed, "Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing," *Concurrency and computation: practice and experience*, vol. 14, no. 13-15, pp. 1175-1220, 2002. [Article \(CrossRef Link\)](#)
- [26] K. Park, and V.S. Pai, "CoMon: a Mostly-Scalable Monitoring System for PlanetLab," *ACM SIGOPS Operating Systems Review*, vol. 40, no.1, pp. 65-74, 2006. [Article \(CrossRef Link\)](#)

- [27] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman, "Planetlab: an Overlay Testbed for Broad-Coverage Services," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 3, pp. 3-12, 2003. [Article \(CrossRef Link\)](#)
- [28] D. Kusic, J.O. Kephart, J.E. Hanson, N. Kandasamy, and G. Jiang, "Power and Performance Management of Virtualized Computing Environments via Lookahead Control," *Cluster Computing*, vol. 12, no. 1, pp. 1-15, 2009. [Article \(CrossRef Link\)](#)



M.Hema received his M.E. degree in Embedded System technology from Anna University, Tamilnadu, India, in 2009. She is currently a Ph.D. student at Anna University. She is a Assistant Professor in Easwari Engineering college, Chennai. Her research interests include Cloud Computing, deep learning, IoT and Data Science.



Dr.S.KanagaSuba Raja is working as an Professor & Head in AIDS Department at Easwari Engineering College. He obtained his Ph.D degree in Computer Science in 2013 from Manonmaniam Sundaranar University, Tirunelveli. He has about 16 years of teaching and research experience. He has successfully guided many graduate and under-graduate students for their research projects. He has several papers published in International journals and conferences to his credit. His current research interests are Wireless Body Area Networks, AI and Data Science Deep learning, IoT and Mobile and Agricultural Engineering.