Computer Systems Science & Engineering
DOI: 10.32604/csse.2023.024895
Article

Tech Science Press

An Efficient Framework for Utilizing Underloaded Servers in Compute Cloud

M. Hema[1,*] and S. Kanaga Suba Raja[2]

[1]Department of Information Technology, Easwari Engineering College, Chennai, 600089, India
[2]Department of Computer Science and Engineering, Easwari Engineering College, Chennai, 600089, India
*Corresponding Author: M. Hema. Email: hemsit82@gmail.com
Received: 03 November 2021; Accepted: 10 December 2021

Abstract: In cloud data centers, the consolidation of workload is one of the phases during which the available hosts are allocated tasks. This phenomenon ensures that the least possible number of hosts is used without compromise in meeting the Service Level Agreement (SLA). To consolidate the workloads, the hosts are segregated into three categories: normal hosts, under-loaded hosts, and over-loaded hosts based on their utilization. It is to be noted that the identification of an extensively used host or underloaded host is challenging to accomplish. Threshold values were proposed in the literature to detect this scenario. The current study aims to improve the existing methods that choose the underloaded hosts, get rid of Virtual Machines (VMs) from them, and finally place them in some other hosts. The researcher proposes a Host Resource Utilization Aware (HRUAA) Algorithm to detect those underloaded and place its virtual machines on different hosts in a vibrant Cloud environment. The mechanism presented in this study is contrasted with existing mechanisms empirically. The results attained from the study establish that numerous hosts can be shut down, while at the same time, the user's workload requirement can also be met. The proposed method is energy-efficient in workload consolidation, saves cost and time, and leverages active hosts.

Keywords: Workload consolidation; energy consumption; under load server; resource utilization

1 Introduction

The high penetration of the internet and the drastic developments that occurred in computing and storage technologies cost-effectively empowered the universally-accessible computing resources [1]. This technological shift gave rise to new horizons in which Cloud computing, the latest computing paradigm is realized. The cloud computing model gained much attention in recent years. Cloud is a ‘‘Model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.’’ According to NIST, there are five mandatory features present in cloud computing, namely, (i) on-demand self-service, (ii) access to broad network (iii) resource pooling, (iv) rapid elasticity or expansion, and (v) measured service. Further to the above, the cloud community has leveraged different kinds of service models to differentiate cloud

service offerings such as (a) Software as a Service (SaaS), (b) Platform as a Service (PaaS) and (c) Infrastructure as a Service (IaaS). The deployment models present in this area are as follows; private, public, community and hybrid. In spite of ideas through brainstorming and also created with the help of cloud services and the above-mentioned deployment models, there is lack of huge capital investment for programming specialists, especially in cloud computing infrastructures. This hinders the development of computing products in the real market.

According [2] to Fox et al. (2009), cloud computing functions as the topmost virtualization technology. Virtual resources are created above physical machines through the virtualization process. These virtual resources are inclusive of storage devices, internet bandwidth, main memory, operating platforms, computing resources, etc. An emulated machine that enables the utility to offer resources via network or computer, storage devices, or a platform is otherwise termed as Virtual Machine (VM). In the case of inquiring about a task with Cloud computation, the result is yielded from the virtual machine(s) are created at Cloud Service Providers (CSP) premise. So, irrespective of the type of job submitted in the cloud, it should be executed under one or more than one VM. The host can be characterized as the parallel functioning of several logical virtual machines that function under a common server. Under one datacenter, multiple hosts exist, while a CSP tends to possess different datacenters on the whole.

The cloud service providers like Google, Microsoft and Amazon started installing a huge number of data centers across the globe to meet the drastically-growing demand from customers in terms of processing power and storage. According to [3] Luo et al. (2014), various well-established and a mesh of computing resources are installed in such cloud datacenters. These machines consume a heavy electricity load for their functioning. This heavy usage of power by cloud computing data centers and the servers that are present in it resulted in a tremendous increase in the demand for electrical energy. In [4] Kaplan et al. (2008) mentioned that the electrical energy consumed by an average data center is equal to that of 25,000 domestic users. This heavy energy usage with ever-increasing demand started gaining attention in recent years. Researchers believe that energy consumption levels can be reduced by transferring the idle physical servers to a lower power state or turning them off. This needs to be accomplished without compromising the customer service requirements. Workload/server consolidation followed by task scheduling is the notable technique applied to resolve the issue discussed above. Server consolidation is performed based on a reduction in the number of active servers in the data center while at the same time, meeting customer requirements and also accomplishing the required tasks. As per the research findings of [5] Brienza et al. (2015), Sleep/Wakeup is a top classification method in which few servers can be switched off when it is idle and can be switched on when there is demand. This can save energy as well as operations costs. However, as per Fan et al. (2007), about 70% of peak power is consumed by idle servers too. So, it becomes inevitable to properly distribute the existing tasks among the available servers. This positively reduces the number of active servers, while at the same time it also meets SLA requirements for the cloud users. The data center hosts are categorized into three groups namely based on their usage: normal hosts, overloaded hosts (the hosts which are extensively used more than its capacity) and under-loaded hosts (hosts which are not used to its capacity). This classification is generally done based on how hosts are utilized; whether above the threshold value i.e., upper threshold or lesser than that i.e., lower threshold. These are named as overloaded and underloaded hosts respectively. Normal hosts are hosts which do not meet these criteria. A study by [6] Barroso et al. (2007) mentions that the data center hosts function only up to 10%–50% of its peak capacity while the hosts that are underloaded remain the primary reason for electricity getting wasted. So, these hosts should be properly leveraged to mitigate the consumption of electricity in cloud data centers via workload consolidation. This concept is illustrated in a simple form in Fig. 1

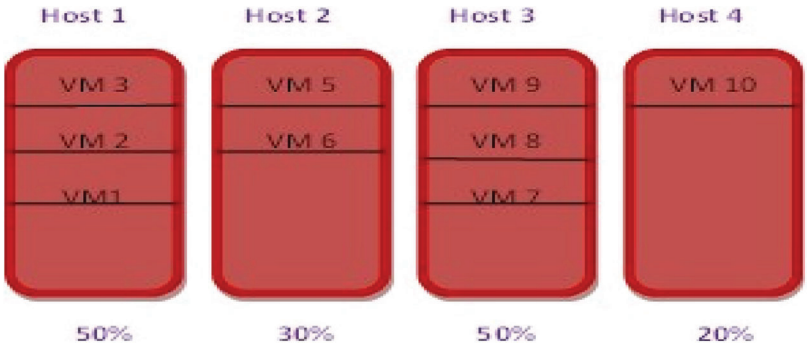

Figure 1: Before workload consolidation of host

Fig. 1 shows the condition of workload before consolidation. In this figure, the author made use of eight hosts of different capacities from 20%–50%. During workload consolidation, the workloads are shifted from one to the another and are managed with equilibrium so that the target host remains evenly loaded. The present research analyzes the overall workload of the datacenters and calculates the lowest threshold to forecast the largest number of hosts that can be evacuated.

Fig. 2 illustrates the workload status after its consolidation. It can be inferred from the figure that four hosts (2 and 4) can be switched to power saving mode while the rest of the hosts (1 and 3) can be left to remain active and with their utilization in the range of 70% to 80%.

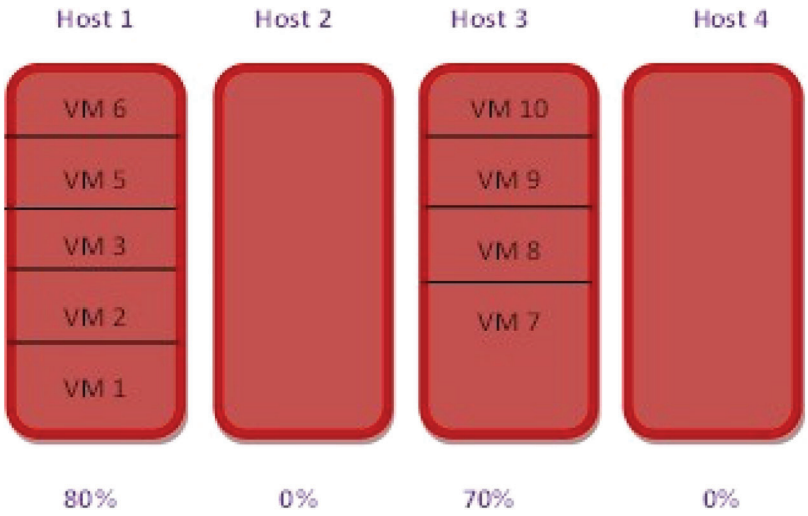

Figure 2: After workload consolidation of host

On the whole, the workload consolidation process is inclusive of the following functions (i) selection of a few virtual machines from hosts that are overloaded and transferring them to the rest of hosts in such a way that the source host attains normality while equivalent loading is found in target hosts. (ii) Selection of all the VMs from the underloaded hosts and switching them to the rest of the hosts so that the target hosts remain evenly loaded. When all the VMs are successfully migrated, then the power supplied for underloaded hosts is cut off to save energy. In the current research work, the key aim is to achieve the second type i.e., selection of underloaded hosts and migration of virtual machines from such underloaded hosts.

The researcher extensively reviewed the existing research studies that proposed techniques to detect underloaded hosts and then vacate them. Based on the review of literature, the researcher found that there is still room for exploration in this research domain i.e., deciding which hosts are underloaded and how VMs in these hosts can be migrated to other hosts. Host Resource Utilization Aware (HRUAA) algorithm is proposed in the current study to detect underloaded hosts and migrate the VMs to other hosts in a vibrant cloud environment. In contrary to the previous mechanisms, HRUAA calculates the least threshold and considers overall utilization of data center and simultaneous utilization of active hosts in the data center. HUA can predict the highest count of hosts to be vacated based on the overall data center workload. The current study compared HRUAA with an existing method (Beloglazov et al., 2012) through experimentation. The study results established that the proposed HUA was able to switch off more hosts while at the same time; it also did not compromise the workload requirements of the users [7,8]. Thus, it resulted in energy-efficient workload consolidation by efficiency utilization of active hosts and minimal migration costs.

The article is organized into the following sections. In the second section, a summary of literature review outcomes is provided. In the third section, the researchers propose the algorithm. Section 4 discusses the implementation of the algorithm, dataset, experimentation procedure, results and the discussion. Session 5 concludes the research paper and then the references used in the work are listed.

2 Related Work

In the study conducted by [9] Robert et al. (2012), different sets of power consumption models were attempted in storage devices, servers and network equipment. In their study, the researchers proposed a three-step model for optimization followed by reconfiguration and then monitoring for saving electricity. The study confirmed that 20% of the energy consumption can be saved if power consumption prediction models guide the energy optimization policy. In a study conducted by [10] Nathuji et al. (2007), different issues were discussed which include power management issues in e-data centers at an early cloud emergence phase. The researchers proposed a solution in which all power management technologies, as well as policies framed, are to be conjoined with virtualization technologies to attain an active deployment in data centers. According to the proposed mechanism, there are two parts present in a resource manager such as local manager and global manager while the former is deemed to be the guest OS' power management strategy. Further, the global manager collects data from local managers for the placement of Virtual Machines. However, the research article fails to strategize the policies for focused resource management for global managers. In the study conducted by [11] Verma et al. (2008), the challenges involved in the power-aware vibrant deployment of applications were deliberated upon from the viewpoint of the bin-packing problem. In general, binds vary in terms of costs and sizes. Live migration is generally applied for the migration of VMs from one host to another one during the usually-scheduled interval. However, the study did not disclose information regarding SLA. In a study conducted [12] by Song et al. (2014), the authors used virtualization to vibrantly allocate the resources according to workload requirements. The study also optimized the number of active hosts to attain energy efficiency in the cloud data center. To achieve this objective, the authors proposed a Variable Item Size Bin Packing (VISBP) algorithm in the resource allocation model based on a relaxed on-line bin packing problem. When rules regarding VM and PM categorization are followed in-tact, then VISBP can be operated with acceptable size differences. The results inferred better performance by the proposed algorithm in the migration of hot spots as well as load balance in comparison to previous algorithms. However, in the study mentioned, all the PMs are deemed to be homogeneous with unit capacity, which remains a drawback for its real-time application. A sub-optimal dynamic SLA-aware resource allocation strategy was proposed [13] by Huang et al. (2013) to attain energy efficiency in cloud computing.

A prediction mechanism was proposed by the authors in the first phase with the help of Support Vector Regressions (SVR). This mechanism was leveraged to determine resource utilization as per SLA requirements. With the help of a genetic algorithm, the authors applied the resource reallocation mechanism to find out the VM requirements of the user. According to the authors, the proposed method met the SLA and satisfied QoS while the profit gains of the cloud provider also got improved significantly. Despite these, GA algorithm provided no convergence to the local optimal solutions at a remarkable execution time.

In the research work conducted [14] by Beloglazov et al. (2010), the researchers proposed to set up different utilization thresholds such as upper and lower utilization to find out the servers that are over-and under-loaded. If host utilization is above the upper threshold, then some of the VMs can be migrated from the specific host to others to mitigate SLA violation (SLAV). On the other end, if there is less utilization than the lower threshold, the whole set of VMs present in underloaded hosts can be transferred, while the host can be switched off to save energy. In a different study conducted [15], by the same authors (Beloglazov et al., 2010) a method based on statistical analysis conducted upon VM utilization theory that adjusts the utilization thresholds automatically was proposed. The study devised an equation to determine the lower threshold based on a few factors such as standard deviation of utilization and probability intervals, host utilization and the number of hosts. In this scenario, a simple approach was proposed [16], by the same authors in a different study (Beloglazov et al., 2012) which compared the relative host utilization to determine the underloaded hosts. In this approach, a minimally-utilized host is first identified. Then the VMs are shifted from this host to the others while at the same time, it is also ensured not to overload other hosts. After accomplishing this task, the original host is then transferred back to power-saving mode. The above-mentioned process is iterated until no VM placement is possible anymore.

A study conducted by [17] Horri et al. (2014), followed the path of existing researches and continued the same by proposing a new technique to execute host utilization. Their research work considered two factors such as host's utilization of the CPU and the count of virtual machines in a host. These factors are important to detect underloaded hosts. The authors also proposed to allocate dynamic weight to both the factors considered above. VM-based Dynamic Threshold (VDT) was applied in this research work at regular intervals. The authors confirmed that the number of VMs getting migrated got significantly reduced which eventually resulted in fewer SLA violations and less energy consumption. Peak load-based static resource allocation schemes mostly under-utilize the computing resources, as per the study findings [18] of Lin et al. (2011). To overcome this challenge, the authors proposed a novel technique to determine the lower threshold. In their work, they considered (1) normal and maximum workloads of a VM (2) maximum and normal workload rates (3) present count of VMs followed by (4) threshold rate in the range of 0 to 1. This simple implementation schema enhanced the resource utilization, as per the study results and it further reduced the cost incurred upon usage by the user. In a study conducted [19] by Yang et al. (2014), the authors proposed a scheme based on load ratios to find the count of PMs to be run or switched off. In this study, the gross-occupied resource weight ratio was calculated by the author based on the ratio of workload to available physical capacity. This ratio was then contrasted against two different factors such as minimum critical occupied resource weight ratio and maximum tolerant occupied resource weight ratio. The authors emphasized that a standby physical machine to be wake up so that it joins with other running physical machines. This should occur only in the case of two instances; when a gross-occupied resource weight ratio is higher than the maximum tolerant-occupied resource weight ratio; and the number of running physical machines is less compared to the total number of physical machines. On the other hand, one running physical machine that has the least load should be selected as the migratory machine from which all the VMs should be moved out to other running PMs and then it should be shut down. This scenario should be followed only under two instances: when the gross-occupied resource weight ratio is lesser than the minimum critical occupied resource weight ratio; and the

number of running physical machines remains more than 1. This paper [20] demonstrates a cost-effective energy solution for the problem by properly placing Virtual Machines (VMs) onto respective servers. A method for effectively deploying virtual machines has been presented to fulfill user requirements.

This article [21] presents a unique method for optimizing resource use and lowering resource costs in the Cloud. The research focuses on the server consolidation process in data centers, where power and energy usage are key considerations. Linear regression, Minimum Migration Time (MMT), and a new node categorization system are used to provide an enhanced consolidation technique on the Open stack platform.

3 Host Resource Utilization Aware (HRUAA) Algorithm

An inference from the second section i.e., review of literature, is that majority of the studies conducted earlier followed a common practice to find out the underloaded host i.e., selection of host with least utilization. Once it is identified, the authors attempted to shift the whole number of VMs from that specific host and proceed with the next host if one can migrate all VMs. This practice is inclusive of sorting the hosts in descending order based on their utilization. Further, the hosts with the least utilization are deemed to be underloaded host which is then processed i.e., migration of virtual machines from this host to different hosts. This task is performed without overloading the target hosts and was repeated. All the VMs are shifted from the least utilized hosts that are least utilized to an appropriate host. These conventional techniques missed taking the data center's overall utilization (by the whole set of hosts) into account when identifying the underloaded hosts. This results in the transfer of all virtual machines from underloaded hosts to different hosts. In addition to the above, the author opines that the knowledge about how hosts are utilized, for instance, the entire workload processed by the data center, is important to optimally choose the host(s) to be vacated followed by placement of such VMs in other hosts. Further, the existing mechanisms simply migrate the chosen virtual machines from under-utilized hosts to different ones haphazardly and do not consider how the target host is being utilized. This may eventually lead to VM placement on the target host that was earlier planned to be vacated shortly. This indirect over-utilization of the hosts may result in a high count of active hosts in every data center which eventually results in more count of virtual machine migrations in the future.

As a solution for the challenges discussed above, the author hereby proposes a new technique to find out the lower threshold. This mechanism calculates how the entire data center is utilized. This is achieved by calculating the utilization of the whole set of active hosts present in the data center. This mechanism also makes use of information on the total workload of the data center and predicts the whole count of hosts that can be vacated. The predicted value of the highest count of hosts to be vacated is then leveraged to determine the lower threshold value. Followed by the segregation of the whole set of available hosts occurs into two categories such as (i) under-utilized hosts below lower threshold and (ii) equal or heavily-utilized hosts than the lower threshold. After this, the virtual machines are selected from the host list (i) and attempted to transfer to the host list of (ii) with minimal alterations in the consumption of electricity.

More restricted decision-making in our suggested algorithm inhibits aggressive consolidation. In general, the suggested method looks for VMs that are better suitable for migration. Because limiting migration of VMs from overloaded servers avoids MT, which may greatly increase the number of migrations, the number of migrations in our method is decreased due to improved selection of VMs to be transferred. The proposed algorithm will reduce the power consumption in cloud data centers and also improve resource utilization in the cloud.

These steps are presented in Algorithm 1.

Algorithm 1: Host Resource Utilization Aware (HRUAA) Algorithm for underloaded Host Detection

```
Step 1:   Input: Phost list, vmList
          Output: vmAllocation
Step 2:   Move all overloaded Hosts and switchOffHost in
          EPhost list
Step 3:   Phost list. sortIncreasingUtilization ()
  Calculate Workload of Datacenter from Eq. (1)
    Calculate Hostmax t from Eq. (2)
    Calculate Low_Threshold
Step 4:     for every Phost ranges in 1 to Maxhost do
              EPhost list to get a new Placement host. add (host)
Step 5:     end for
Step 6:     PhostMinUtilization ← 1
Step 7:     while (TRUE)
Step 8:       if Hosttot ← EPhost list to get underloaded
                    host. Size ()
Step 9:               break
    Endif
Step 10:      if Phost MinUtilization does not belong to EPhost list to get under loaded host then
                EPhost list to discover under loaded host.
              add (PhostMinUtilization)
Step 11:        Endif
Step 12:        If Phost MinUtilization does not belong to EPhost list to get a new Placement host then
Step 13:          EPhost list to get a new Placement host. add
        (Phost MinUtilization)
Step 14:    Endif
Step 15:  for Every VM in PhostMinUtilization do
Step 16:    VM_Util ← getVMUtil (vm, PhostMinUtilization)
Step 17:          minimumPowerDiff ← MAXIMUM
Step 18:          Hostallocated ← NULL
Step 19:  for every host in Phost list do
Step 20:      if host belongs to EPhost list to get New Placement then
Step 21:        Phost++;
Step 22:      endif
              For every VM ranges in 1 to phost do
```

(Continued)

Algorithm 1: (continued).

$Utilhost \leftarrow \frac{VMutil(cpu)+VMutil(RAM)+VMutil(BW)}{Utilhost}$

Endfor

Step 23: **if** Phost has enough resources for vm which need to migrate then
Step 24: **If** Utilhost (AF placement) > UppThreshold then
Step 25: Phost++
Step 26: **endif**
Step 27: **if** Phost (AF placement) - Phost (BF placement) < minimum PowerDiff then
Step 28: minimum PowerDiff Phost (AF placement) - Phost (BF placement)
Step 29: allocatedHost ← host
Step 30: **endif**
Step 31: **endif**
Step 32: **end for**
Step 33: **if** Hostallocated ≠ NULL then
Step 34: **if** Hostallocated not belong to EPHost List to get underUtilized then
Step 35: EPhost list to get UnderUtilized.add (allocatedHost)
Step 35: **endif**
Step 36: migMap.add (vm, Hostallocated)
Step 37: **else**
Step 38: break
Step 39: **endif**
Step 40: **end for**
Step 41: PhostMinUtilization ← Phost Min Utilization + 1
Step 42: **end while**

At first, in Host Resource Utilization Aware (HRUAA) algorithm, the researcher added all the over-utilized hosts as well as switched-off hosts in the lists such as 1) exclude Host List for Finding under Utilized and 2) exclude Host List for Finding New Placement. This is done to skip them for further search that results in efficient cost-saving upon computation. After this, the highest count of to-be-vacated hosts ($Host_{max}$) is calculated based on total data center workload,

$$WDC_{tot} \leftarrow \sum_{i=1}^{i=host_{tot}} Util_i \tag{1}$$

where $Util_i$ is the utilization of I^{th} host. while this can be determined based on how active hosts are utilized. Subsequently, using $Host_{max}$, the lower threshold value ($Low_{thershold}$) is brought up.

$$Host_{max} \leftarrow host_{tot} \frac{WDC_{tot}}{Upperthershold} \tag{2}$$

This value is used to search for an under-utilized host. Then, those hosts which are utilized lower than $Low_{thershold}$ are excluded for VM placement.

$$Low_{Threshold} = Host_{sort}\ [WDC_{tot}] \tag{3}$$

In the next step, the least–utilized host is selected and all the VMs in it are attempted for migration to another host. At the time of placing VMs, it is ensured that the target host contains sufficient resources to accommodate the newly-migrated VM.. In this way, there occurs no overutilization of the target host once the new virtual machines are placed. When the above-discussed terms are met, such hosts are selected for the new placement, resulting in a minimal increment in the consumption of electricity after the placement. In addition to the above, the author also verified the availability of suitable hosts for the shifted VMs (from under-utilized hosts) for placement. Only based on availability, the migration of VMs occurs and the host is switched off. This is a repetitive process that occurs for successive underutilized hosts too till all the placements are possibly done. The HRUAA takes only a few milliseconds to migrate the VM to another host.

4 Evaluation of the Performance

Performance evaluation of the current research includes Experimentation test bed and simulation outcomes are as follows.

4.1 Experimentation Test Bed

To assess the technique presented in this study, the model should be implemented in a large-scale cloud data center under a dynamic workload. However, due to the cloud infrastructure's intricate requirements, the CloudSim toolkit [22] (Calheiros et al., 2014) was used in the research work as a simulation environment. The most significant and efficient simulation environment, CloudSim was used since it has a few promising factors namely, energy consumption and accounting modeling, virtualized resource management and modeling, VM migration, workload dynamism and SLA computation [23] (Patel et al., 2016). The current study experiment testbed configurations are provided in Tab. 1 for 800 heterogeneous hosts. Almost 50% of these hosts belong to HP ProLiant ML110 G4 servers (Type 1), and the balance 50% belongs to HP ProLiant ML110 G5 servers (Type 2). The researcher mapped the server's CPU frequency in the range of MIPS ratings: 1860 MIPS while each core HP ProLiant ML110 G5 server and 2660 MIPS each core of the HP ProLiant ML110 G5 server. Every server is created with a capacity of 1 GB/s network bandwidth. The configurations for VM and host are tabulated see in Tab. 1. With single-core VMs, each VM type information is given herewith High-CPU Medium Instance (2500 MIPS, 0.85 GB); Extra Large Instance (2000 MIPS, 3.75 GB); Small Instance (1000 MIPS, 1.7 GB); and Micro Instance (500 MIPS, 613 MB).

In the beginning, the VMs were allocated based on requirements raised by the resource and designated by types of VM. To attain the electricity consumption data of a host, the author used real power consumption data sourced from the SPEC power benchmark (SPEC power benchmark, 2008). The author was able to arrive at the accurate value of electricity consumed by servers using a linear relationship between the energy consumed and how far the CPU is utilized. From the table, it can be observed that the hosts tend to consume power significantly even at low utilization. So, such sort of hosts should be switched off when not in use.

CoMon project was the source of real datasets based on which workload traces were collected. CoMon project is a scalable monitoring infrastructure developed for PlanetLab [24] (Park et al., 2006). Slice-centric daemon data which shows resource consumption per slice was collected. With the help of NAS, live migration of VM is facilitated in the system, by eliminating the need to use direct-attached storage. This type of storage reduces migration overhead as there is no need to copy the disk content.

Table 1: Host and VM specification

Virtual machine specification					Host specification					
VM type	MIPS	Core (Processing elements)	RAM (MB)	Bandwidth (Mbps)	Name	MIPS	Core (Processing elements)	RAM (MB)	Bandwidth (Gbps)	Number of hosts
High-CPU medium instance	2500	1	870	100	HP ProLiant ML110 G4-Xeon 3040	1860	2	4096	1	200
Extra-large instance	2000	1	1740	100	HP ProLiant ML110 G5-Xeon 3075	2660	2	4096	1	200
Small instance	1000	1	1740	100	HP ProLiant DL360 G7-Xeon X5675	3067	12	16,384	1	200
Micro-instance	500	1	613	100	HP ProLiant DL360 G9-Xeon E5-2699	2300	36	65,536	1	200

The type of data collection was overall CPU and memory utilization, transmission and receipt rates during the 1 and 15 mins consumed physical memory and consumed virtual memory, context number (numeric user id for the slice), number of processes, and slice name. Most of the nodes have 50 slides andon its order running at a time. The current study mostly used overall CPU utilization data collected at an interval of 5 min for >1000 virtual machines from servers that are functioning at 500 locations spread globally. The author selected the dataset collected on 03rd of March 2011 which had a total of 1052 VMs. Tab. 1 lists of types of VMs such as type 1, type 2, type 3 and type 4 among these 1052 VMs.

During simulation, each VM's workload trace is dynamically assigned to virtual machines. Every 5 min interval the resource utilization is measured by VMs in the physical server. The proposed algorithm, HRUAA must run every 5 min based on data collected by workload traces. The test is executed 10 times for an algorithm and the median value is calculated and presented in terms of each of the performance metrics.

4.2 Simulation Outcomes

The proposed HRUAA algorithm was compared with traditional approaches discussed in Beloglazov and Buyya (2012). The author used different metrics to compare and contrast the performance and efficiency of the HRUAA algorithm with that of the other approaches. The parameters used include the number of SLA performance degradation due to migration (PDM), average SLA violation, Service Level Agreement Violation (SLAV), SLA time per active host (SLATAH), energy consumption, overall SLA violation, and several host shutdowns and VM migration. Among these performance metrics, the energy consumption of the host is a critical parameter. One of the important metrics is a count of virtual machine migrations that started during the placement of virtual machines after selecting them from the least-utilized host. SLAV corresponds to SLA violations that occurred at the time of the workload consolidation phase. PDM is performance degradation of the system due to VM migration.

$$PDM = \frac{1}{N}\sum_{i=1}^{N}\frac{Req_{vm} - Alloc_{vm}}{Req_{vm}} \tag{4}$$

where N is the total number of VM, Req_{vm} is requested VM and $Alloc_{vm}$ is allocated VM.

The SLATAH is the time percentage when active hosts experience 100% CPU utilization.

$$SLATAH = \frac{1}{N}\sum_{i=1}^{N}\frac{totF_i}{Acthost_i} \tag{5}$$

where N is the number of hosts, $totF_i$ is the total amount of time the host ids utilized fully, $Acthost_i$ is the total amount of time the host active. The overall reduction in the performance, as a result of transferring virtual machines, is denoted by Performance Degradation due to Migrations (PDM). SLAV is calculated based on SLATAH and PDM.

$$SLAV = PDM * SLATAH \tag{6}$$

Different combinations of VM selection policies and overloaded host detection methods were used in this study. Some of the overloaded host detection techniques include robust local regression (LRR), local regression (LR), median absolute deviation (MAD) and Static Threshold (ST) and interquartile range (IQR). VM selection policies considered here are Minimum Migration Time (MMT, Maximum Correlation (MC) and Maximum Utilization (MU).

Safety parameter (s) value is adjusted to keep both SLA violation and consumption of energy under control. The consolidation of VMs by the system occurs based on s value. When a host has a low value, it shows less energy consumption through the high value of SLA violation is a result of consolidation and vice versa. So, one could address the tradeoff between SLA violation as well as consumption of energy. The author simulated different permutations of overloaded host detection techniques (MAD, IQR, LR, LRR and ST) and virtual machine selection techniques (MU, MC, MMT and RS) for previous and the proposed approach for under loaded host detection followed by VM placement. The results attained from the assessment are shown in Figs. 3–7. From the results, it could be inferred that superior performance is attained by the presented technique due to the increased number of host shutdowns (approximately 67%). This might be attributed to the following scenario; the proposed method took the exclusion of hosts that are possibly-to-be-vacated in the future also were taken into consideration at the time of VM placement. This technique was lacking in the existing methods which considered only the power consumption metric. The result showcased better workload balancing with the evacuation of more hosts. The method presented in the study attempts to leverage the minimum number of hosts while trying to allot the highest count of jobs to active hosts at the same time.

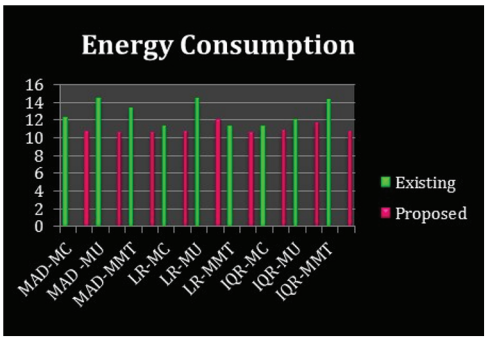

Figure 3: Energy consumption

There exists a direct correlation between the consumption of energy and the count of active hosts present in the data center. So, when the count of hosts to be shut down increases, it directly reduces the energy consumption.

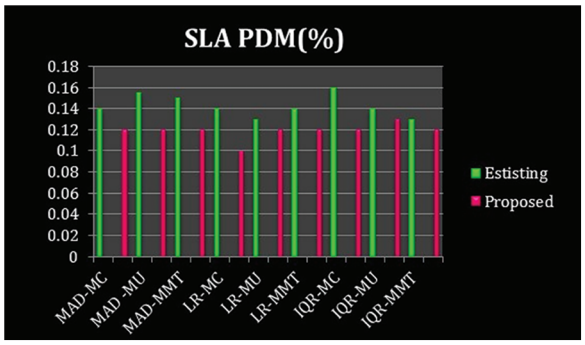

Figure 4: Comparison of SLA–PDM

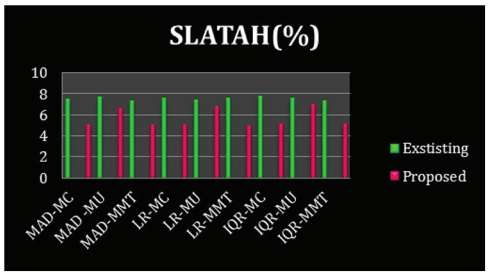

Figure 5: Comparison of SLATAH

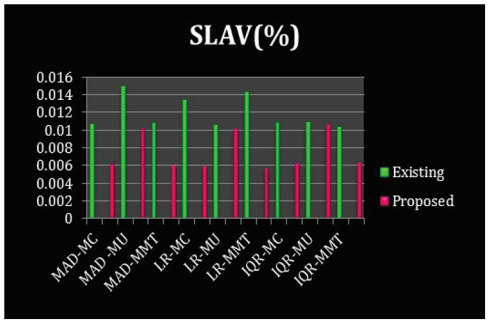

Figure 6: Comparison of SLAV

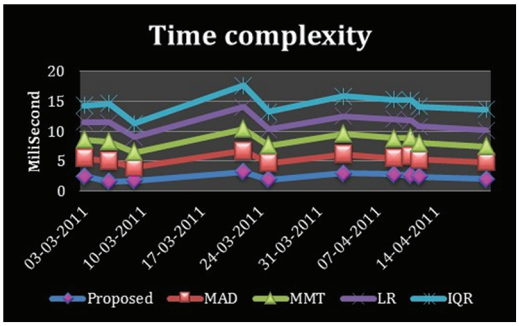

Figure 7: Time complexity

From Fig. 3, show the energy consumption it can be observed that the energy efficiency status got improved on an average-up to 14.59%. Only when more number of hosts is shut down, one can achieve this improvement in energy consumption (i.e., improvement by 69% as shown in Fig. 2). To shut down more hosts, the workload should be appropriately consolidated in a minimal number of hosts required for the number of VM migrations. SLAV metric is generally calculated based on (i) SLA performance degradation due to migration (PDM) (Fig. 4 shows the Comparison of SLA-PDM followed by (ii) SLA violation Time per Active Host (SLATAH) shown in Fig. 5. An improvement was observed in PDM as shown in Fig. 5 by 7.40%, while the SLATAH also got improved up to 13.35%. According to the results attained, HRUAA can be inferred to be efficient in deciding the under loaded host. It has the potential to vacate more hosts which eventually saves the consumption of energy. All these tasks were accomplished without compromising SLA.

Any Algorithm that needs to evaluate the important parameter needs to face time complexity. Sometimes an algorithm can perform better, but with high time complexity, an algorithm is not acceptable. The Time Complexity of HRUAA is O (m*n), Where m is the number of active hosts and n is the number of VMs migrated. Fig. 7 shows the time complexity depicts the result of the time complexity of HRUAA in a millisecond. At the end of each period and simulation of every day, and calculate the mean of all achieved times. The result found that HRUAA time complexity is very low and acceptable.

5 Conclusion

Under-loaded host detection remains the most crucial phase in the workload consolidation process. The current research reviewed the traditional methods being used till date in the calculation of lower threshold value which in turn is utilized to determine the underutilized hosts. Traditional methods determine the lower threshold value based on how far the available hosts are utilized. But these methods miss considering the holistic scenario of the total workload of the datacenters into account. Host resource Utilization Aware (HRUAA) algorithm was proposed in this study as a novel technique to overcome the challenges discussed earlier. This algorithm predicts the highest count of hosts that can be vacated by calculating the least threshold. The threshold value is calculated by considering the overall utilization of the data center. The study results established the efficiency of HUA in finding the underloaded hosts and correspondingly vacating a high count of hosts. In this way, it saves much energy from getting consumed while at the same time; it also ensures the non-occurrence of SLA violation. The proposed technique can be implemented in a real-time setup in the future under a data center environment using a vibrant set of variables.

Acknowledgement: The author with a deep sense of gratitude would thank the supervisor for his guidance and constant support rendered during this research.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] C. J. Huang, C. T. Guan, H. M. Chen, Y. W. Wang, S. C. Chang *et al.,* "An adaptive resource management scheme in cloud computing," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 382–389, 2013.

[2] A. Fox, R. Griffith, A. Joseph, R. Katz and A. Konwinski, "Above the clouds: A berkeley view of cloud computing," in *Department of Electrical Engineering and Computer Sciences*, vol. 17, Berkeley, CA, USA: Technical Report, pp. 1–23, 2009.

[3] J. Luo, X. Li and M. Chen, "Hybrid shuffled frog leaping algorithm for energy-efficient dynamic consolidation of virtual machines in cloud data centers," *Expert System with Applications*, vol. 41, no. 13, pp. 5804–5816, 2014.

[4] J. M. Kaplan, W. Forrest and N. Kindler, "Revolutionizing data center energy efficiency," in *Proc. McKinsey & Company*, New York, NY, USA, Technical Report, pp. 1–13, 2008.

[5] S. Brienza, S. E. Cebeci, S. S. Masoumzadeh, H. Hlavacs, Ö. Özkasap *et al.,* "A survey on energy efficiency in P2P systems: File distribution, content streaming, and epidemics," *ACM Computation Surveys*, vol. 48, no. 3, pp. 1–37, 2015.

[6] L. A. Barroso and U. Holzle, "The case for energy-proportional computing," *Computers*, vol. 40, no. 12, pp. 33–37, 2007.

[7] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.

[8] X. Fan, W. D. Weber and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *Proc. Annual Int. Symp. on Computer Architecture*, San Diego, California, USA, pp. 13–23, 2007.

[9] B. Robert, M. D. Hermann, L. Ricardo and G. Giovanni, "Cloud computing and its interest in saving energy: The use case of a private cloud," *Journal of Cloud Computing Advances, System and Application*, vol. 1, no. 1, pp. 1–25, 2012.

[10] R. Nathuji and K. Schwan, "Virtual power: Coordinated power management in virtualized enterprise systems," *ACM SIGOPS Operating Systems Review*, vol. 41, no. 6, pp. 265–278, 2007.

[11] A. Verma, P. Ahuja and A. Neogi, "pMapper: Power and migration cost aware application placement in virtualized systems," in *Proc. ACM/IFIP/USENIX 9th Int. Middleware Conf.*, Leuven, Belgium, pp. 243–264, 2008.

[12] W. Song, Z. Xiao, Q. Chen and H. Luo, "Adaptive resource provisioning for the cloud using online bin packing," *IEEE Transactions on Computers*, vol. 63, no. 11, pp. 2647–2660, 2014.

[13] C. J. Huang, C. T. Guan, H. M. Chen, Y. W. Wang, S. C. Chang *et al.,* "An adaptive resource management scheme in cloud computing," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 382–389, 2013.

[14] A. Beloglazov and R. Buyya, "Energy efficient resource management in virtualized cloud data centers," in *Proc. IEEE/ACM Int. Conf. on Cluster, Cloud and Grid Computing*, Melbourne, Victoria, Australia, pp. 826–831, 2010.

[15] A. Beloglazov and R. Buyya, "Adaptive threshold-based approach for energy efficient consolidation of virtual machines in cloud data centers," in *Proc. Int. Workshop on Middleware for Grids, Clouds and e-Science*, Bangalore, India, pp. 1–6, 2010.

[16] 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.

[17] A. Horri, M. S. Mozafari and G. Dastghaibyfard, "Novel resource allocation algorithms to performance and energy efficiency in cloud computing," *Journal of Supercomputing*, vol. 63, no. 3, pp. 1445–1461, 2014.

[18] W. Lin, J. Z. Wang, C. Liang and D. Qi, "A threshold-based dynamic resource allocation scheme for cloud computing," *Procedia Engineering*, vol. 23, pp. 695–703, 2011.

[19] C. T. Yang, J. C. Liu, K. L. Huang and F. C. Jiang, "A method for managing green power of a virtual machine cluster in cloud," *Future Generation Computer Systems*, vol. 37, pp. 26–36, 2014.

[20] P. Pete, K. Patange, M. Wankhade, A. Chatterjee, M. Kurhekar *et al.,* "3E-VMC: An experimental energy efficient model for VMs scheduling over cloud," in *Proc. Int. Conf. on Secure Cyber Computing and Communication (ICSCCC)*, Jalandhar, India, pp. 322–327, 2018.

[21] S. Qamai, R. K. Koppanati and K. Kumar, "VM-MMT: A novel approach for VM consolidation over openstack cloud using linear regression and minimum migration time," in *Proc. Int. Conf. on Intelligent Computing and Control Systems (ICICCS)*, Madurai, India, pp. 1814–1819, 2018.

[22] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. D. Rose and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.

[23] N. Patel and H. Patel, "A comprehensive assessment and comparative analysis of simulations tools for cloud computing," *International Journal of Engineering and Computer Science*, vol. 5, no. 11, pp. 18972–18978, 2016.

[24] K. S. 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.