Load Aware Hotspot Selection for SLA Improvement in Cloud Computing and Protect Environment by Reduction In CO₂ Emissions

Vaneet Kumar A.P.¹,a) and Balkrishan Jindal ²,b)

¹University Institute of Engineering, Department of Computer Science
Chandigarh University, Gharuan,
Punjab, India.

²Computer Engineering Section,
Yadvindra College of Engineering, Punjabi University, Guru Kashi Campus, Talwandi Sabo,
Punjab, India

a)aggarwal.vkg@gmail.com
b)balkrishan_76@rediffmail.com

Abstract: Internet of Things (IoT) is a leading concept that envisions everyday objects around us as a part of internet. In order to accomplish this attribution, cloud computing provides a pathway to deliver all the promises with IoT enabled devices. The outbreak of COVID-19 coronavirus, namely SARS-CoV-2, acts as feather to the cap for the growth of Cloud users. With the increasing traffic of applications on cloud computing infrastructure and the explosion in data center sizes, QoS along with energy efficiency to protect environment, reducing CO₂ emissions is need of the hour. This strategy is typically achieved using Three Layer upper Threshold (TLTHR) policy to analyze and perform VM consolidation. The proposed model controls number of migrations by placement of virtual machines, based on VMs and their utilization capacity on host. The efficacy of the proposed technique is exhibited by comparing it with other baseline algorithms using computer based simulation. Hence better QoS and energy efficiency has been obtained than other classical models.

Keywords: Cloud Computing, IoT, Threshold, CO₂ emissions, Service Level Agreement.
1. INTRODUCTION

Due to recently emerged large scale data centers, energy consumption in cloud computing also has increased at an unprecedented rate. These data centers consume around 18% of the total global energy used by information and communication technology (ICT) [1]. The ICT industry is predicted to consume about 20% of electricity by 2025 worldwide. Moreover, day by day escalating electricity consumption in cloud data centers has increased the operation expenditure (OPEX). Besides the running cost of ownership (RCO) energy consumption in virtualized CDCs put an unseen effect on environment in terms of CO2 emissions with more that 43 million ton every year. Carbon dioxide (CO$_2$) emissions by electricity production is a global problem that badly effects the environment and becomes a big threat for the human health. Even, researches from entire world is working to produce electricity using natural resource though maximum utilization of energy is also downright important. To deal with abovementioned challenges virtualization technology is a prevailing strategy [2]–[4].

As a consequence of COVID-19 pandemic worldwide, cloud market also will see a paramount jump with a compound annual growth rate (CAGR) of 12.5% from USD 233 billion. In 2019 to USD 295 billion by the end of 2021. Carbon Although the important factors controlling the market growth includes the support of enterprises to the remote workforce in form of investment in IT infrastructure for hyper scalar & Cloud Service Providers (CSPs). With the popularity of Internet of things the burden on cloud data centers is also increased accordingly. Internet of things (IOT) is a concept that envisages all objects in our vicinity as an indispensable part of Internet. All the commonly used objects in day to day life like automobiles, physical devices and many more use embedded electronics, software, & network connectivity enabling the two way communication of data are considered to be an integral part of IOT. The enormous amount [5]–[7]of Big Data generated by IOT puts the Internet Infrastructure under huge strain leading the system to a situation of fatigue. The Cloud computing model provides on-demand ingress to a common pool of various configurable IT resources, provisioned as Software (S), Applications (A) and Infrastructure (I) as a Service (XaaS). Cloud based platform allows the user to connect to the infrastructure (IaaS) in their proximity, using built in and Customizable applications irrespective of time and location of the user, thus making it more user friendly. This enables Cloud to act as a front end to access the Internet of things.

To meet the future demands of worldwide clients from CDCs, the major cloud service providers such as Microsoft Azure, Yahoo, IBM, Google, Alibaba, and other similar large enterprises, etc are expanding their data center’s support which makes cloud computing QoS and energy constrained system. Therefore to reduce energy expenditure along with satisfying user’s requirements virtualization technology is used [8][9]. This technology creates an intelligent layer of abstraction called Virtual Machine Monitor (VMM) to serve on demand resources in pay-as-you-use model [10][11]. Energy consumption can be controlled by consolidating partial loaded by migrating VMs to put idle or little loaded servers on sleep mode. The primarily aim of this study is to improve QoS and fulfill promises provided to users. Even though the VM consolidation can improve energy efficiency at the cost of throughput and response time, reducing energy consumption is a significant challenge while maintaining Service Level Agreement.

In [12], authors presented the VM consolidation process using hotspot selection, VM selection and VM-to-PM mapping. To solve this issue, violation of threshold plays a key role [13][14][15]. Thus, a strong statistical approach is proposed to obtain workload scenario based threshold to improve response time. In this scheme firstly, average CPU utilization of VMs on each host is calculated, secondly, robust statistical method is applied for obtaining load aware threshold. The following are the main contributions of this paper:

- To determine when a host is considered heavily loaded, proper upper threshold value is required.
- We have introduced TLTHR algorithm to estimate upper threshold value based on recent resource
utilization for overloaded host detection and VM placements. This algorithm mainly aims to minimize the total power consumption of data center.

- To determine lightly loaded host, we used constant lower CPU utilization threshold proposed in [12]. All the virtual machines are selected for reallocation and put them on sleep mode in order to reduce energy consumption.
- Find a near optimum host where selected VM would be reallocated. We used three layer modification of best fit decrease (TLMBFD) VM allocation approach that reduces number of migrations and hence improves Service Level Agreement and instruction energy ratio.
- Matlab tool kit is used to evaluate the performance and effectiveness of the TLTHR algorithm with MAD, IQR, THR in the literature

Cloud computing is used by almost all the electronic gadgets like cellphones, laptops, wireless sensor networks, home appliances etc. thus, the growing dependency of cloud processing and data consumes huge energy. The proposed work helps the corporate by cutting down the power bills through efficient utilization of resources. Besides the monetary benefits, this work also protects the environment indirectly by lessons in CO2 emissions by reducing overall energy consumption.

The rest of the paper is organized as follows: Section 2 discusses related work along the gap in study, Section 3 elaborates the proposed model with detailed description of our approach, Section 4 and 5 introduce system models and results respectively. At the end section 6 presents the conclusion.

2. RELATED WORK

Much research work was tried to improve energy efficiency, which, slightly similar to our approach, were based on metaheuristic algorithms. In this section, we present different threshold analysis policies and their effects on several VM allocation and reallocation approaches in VM consolidation process.

Beloglazov and Buyya[12] presented the Random Selection (RS), Maximum Correlation (MC), and Minimum Migration Time (MMT) algorithms to select virtual machines. Authors were used Double threshold (THR), Median Absolute Deviation (MAD), and Inter Quartile Range (IQR) for threshold measures. These policies were predicate the mechanism to find single upper threshold value for the entire data center.

Alboaneen et al [16] discussed the host load categorization by classified the host into different states like normal, critical, under load and overloaded. Additionally, Zhou et al [17] investigated an Adaptive three Threshold Energy Aware (ATEA) VM placement algorithm that reduce simultaneous energy consumption and SLA violations. They used K-means clustering to find values of these thresholds and
generate fewer migrations than IQR, MAD and THR policies. Kansal and Chana [18] presented Mean of mid and highest host utilization has been used to fix upper threshold but could not able to clear about the underload host detection. Wood et al [19][20] have suggested two approaches to diminish the overloaded machines by monitoring target hotspots. Marzolla et al. [21] applied local VM consolidation based on coarse grained gossip to migrate VMs from smallest to largest laden node. Mehta and Neogi [22] have presented a ReCon tool to dynamically consolidate servers in data centers. The work cited in [23][24][25][26] does not principally aims to achieve high response time. All the above approaches improved the energy efficiency but failed unnecessary migrations with fixed upper threshold results in more SLA violations. Thereby, these techniques differ from our proposed approach, which technically aims to improve QoS with the help of reducing number of migrations.

Cioara et al [27] presented the reinforcement learning based algorithm to improve energy efficiency. For evaluating greenness level, Entropy (E) is used and compared with fixed predefined threshold value to trigger consolidation process. They used Euclidian distance to select a server for VM migration. Simultaneously, Mosa and Sakellariou [28] developed a new parameter-based VM consolidation solution that can give more flexibility in choosing between different trade-offs. The hypothesis used here could not be accepted by greedy algorithms as it can lead to under loaded hosts most of the time. Mazrekaj et al. [29] proposed migration based on the information provided by host’s age. For this upper threshold is calculated using SLA violations and lower threshold is set to 10% of host utilization capacity. The results reveal high energy saving as compared to centralize approach. It, however, achieve noteworthy low SLA violations.

Monil and Rehman [30] explored energy saving consolidation using Mean Mode Standard deviation (MMMSD) adaptive threshold method. They used correlation coefficient and steady state criteria to predict future needs. Hence, it improves energy efficiency by 8.5% and SLA by 84% due to fewer migrations. Further, learning automata[31] overload detection is proposed by predicting host utilization based on VMs requirement history. Probability of ASC and DESC is calculated by standard deviation and average of all VM to decide migration. Support Vector Regression [32] for auto scaling to predict the processing load of a web server, based on historical observations has been applied by computing the MAE and RMSE error measurements for the 24 hour test intervals. Although future needs is very hard to predict in cloud computing thus, our algorithm provides current load based threshold for each host. Kusic et al. [33] solve this dynamic resource provisioning problem using look ahead control policy and kalman filter has been used to improve QoS. However, the time complexity is so high that it is obviously not suitable for large scale data centres.
To control VM consolidation\cite{34}\cite{35}\cite{36}, various threshold finding algorithms were proposed under two categories, Static and Adaptive. The static CPU threshold is not suitable for unpredictable workload. To make the system workload dependent, heuristic algorithms for auto adjustment of the upper threshold is the best strategy. The number of VMs on host and its utilization can be very dynamic in real world. So, in this study, instead of single upper threshold we propose three layer upper threshold (TLTHR) scheme. Here load dependent upper threshold value is calculated for each host. In the present study energy consumption, SLA violation and number of VM migrations are considered at the same time.

3. SYSTEM MODELS AND PROBLEM DEFINITION

The aim of the study is to minimize the overall energy consumption of data center while optimizing the resources utilization. This section configures the cloud data center and cloudletsto define the problem.

3.1 Modeling of Physical Machines and VM Requests

Consider $N$ is the number of VM's and $M$ is the number of physical machines in cloud data center. A set \{$V_1,V_2,\ldots,V_N$\} of Virtual Machines where $V_v \mid 1 \leq v \leq N$ is consider as cluster of $N$ VMs. A set \{job$_1$,job$_2,\ldots,job_K$\} of task running on each VM where \{job$_j \mid 1 \leq j \leq K$\} is a group of $k$ jobs. Each virtual machine $v$ is characterized by its CPU utilization $Uv^CPU$ in MIPS as we only take CPU utilization of VMs in our modeling. Upon allocation of all the Virtual Machine requests, the resource reservation broker will calculate whether enough hosts are available to accommodate all their requests.

| Notation | Description |
|----------|-------------|
| $Uv^CPU$ | CPU utilization of Virtual Machine $v$ |
| $v_j$    | $j^{th}$ Virtual Machine |
| $h_i$    | $i^{th}$ Host |
| $M(i,j)$ | VM request $v_j$ is allocated to the host $h_i$ |
| $T(M)$   | Virtual Machine requests served by cloud data center |
| $RAR(M)$ | Resource Acceptance Ratio |
| $Uh^CPU$ | host utilization |
| $UTHR_{h1}$ | Upper threshold of host $h$ for scenario $s1$ |

$$M(v,h)=\begin{cases} 1, \text{ virtual machine } vis \text{ allocated to host } h \\ 0, \text{ otherwise } \end{cases} \quad (1)$$
In equation (1) let $\mathcal{M}(i,j)$ indicate whether the VM request $v_j$ is allocated to the host $h_i$ or not

$$T(\mathcal{M}) = \sum_{h=1}^{M} \sum_{v=1}^{N} \mathcal{M}(v,h)$$  \ \ \ (2)$$

In equation (2), we use $T(\mathcal{M})$ to denote the virtual machine requests served by cloud data center

$$RAR(\mathcal{M}) = \frac{T(\mathcal{M})}{N}$$  \ \ \ (3)$$

At the same time requests may be rejected if sufficient host resources are not available to serve the request. $RAR(\mathcal{M})$ in equation (3), formulate Resource Acceptance Ratio to denote virtual proportion of machines accepted by the system.

### 3.2 Modeling of Energy Consumption

The data center consumes energy by various IT devices except from the cooling system and power distribution units. The most easily optimizable IT devices are CPU, Memory, Bandwidth and Disk because these can improve energy efficiency of data center without upgrading any hardware. It has been observed that the energy saving to a certain degree can be adjusted only by CPU voltage frequency [37][33][38]. Further equation (4) shows that the power consumption $PC_h$ of host $h$ has linear relation with host utilization $U_h^{\text{cpu}}$ [39][40]. It has also been observed from table 2, that servers in the idle state consume approximately 70 percent of their peak utilization power [41].

**TABLE 2. Power Consumption by the two servers at different load levels in Watts.**

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

$$PC_h = \begin{cases} (PC_h^{\text{busy}} - PC_h^{\text{idle}}) \times U_h^{\text{cpu}} + PC_h^{\text{idle}}, & U_h^{\text{cpu}} > 0 \\ 0, & \text{Otherwise} \end{cases}$$  \ \ \ (4)$$

From equation (4) for $N$ virtual machines and $M$ physical machines, total power consumption of the data center is calculated as:

$$TPC = \sum_{h=1}^{M} b_h \times \left( (PC_h^{\text{busy}} - PC_h^{\text{idle}}) \times \sum_{v=1}^{N} (a_{vp}, C_v) \right) + PC_h^{\text{idle}}$$  \ \ \ (5)$$

To check energy efficiency of VM to PM mapping, equation (6) is used to calculate the average number of instructions executed for a given amount of energy in Million Instructions Per Joule (MIPJ).
IER(ℳ) = \sum_{h=1}^{M} \sum_{v=1}^{N} \frac{a_{vh} \cdot C_v}{TPC} \quad (6)

Note that if the variables \( b_h \) and \( a_{vp} \) are 1, it means virtual machine \( v \) is assigned to host machine \( h \). The objective this study is to improve.

1. Minimize \( \sum_{h=1}^{M} PC_h \)

2. Maximize IER(ℳ)

3. Improve SLA = SLATH \times PDM

Subject to constraints:

\[ \sum_{h=1}^{M} a_{vh} = 1, \quad \forall h \in H \quad (a) \]

\[ \sum_{v=1}^{N} a_{vh} \cdot C_v \leq \text{associated threshold to } h \forall h \in H \quad (b) \]

Constraint (a) defines that VM \( v \) is allowed to allocate on host \( h \) whereas Constraint (b) defines total MIPS allocated by virtual machines of particular host below associated upper threshold value \( (UTHR_{h}^{S1}, UTHR_{h}^{S2}, UTHR_{h}^{S3}) \). Here \( UTHR_{h}^{Sx} \) is scenario based upper CPU utilization threshold associated to each host \( h \).

4. PROPOSED WORK

In [42], it is clearly shown that CPU consumed less than 30% of the full load power while server is in low active mode and in high active mode, power consumption may be more than 70%. In our proposed model CPU utilization parameter is consider as a threshold measure. Thus a robust statistical method is embedded with threshold (THR) policy which is more effective than classical method for data containing outliers or coming from non-normal distributions. The normalization curve is examined based on different threshold combinations of host utilization in data center. Mean utilization of data center is close to 70% of peak capacity. Major existing method assume data center utilization lower than 30% is consider for virtual machine migration and data center utilization higher than 70% consider as over utilization with chances of occurrence of over utilization is 7%. Area under normal curve in 1 theta is 86.4%. So consider area outside side curve from one side is 7.6%. So if, it increases overutilization factor from 70 to 80% it
become equal to 1.5 theta whose area under curve is approximately 96.4% so chances of over utilization will decrease from 7% to 2.4%. Utilization curve shown in Figure 1 depicts major hosts fall under the area of 60-70% utilization curve thus in our proposed work we focus on number of virtual machines to find values and gap between three layers of upper threshold discussed in experiment and discussion section.

Figure 1 Host Utilization Curve

4.1 Scenario based Three Layer upper THReshold (TLTHR)

Two key techniques are generally used to optimize VM-PM mapping in the cloud environment” i.e. VM’s placement and VM’s consolidation. VM consolidation under scenario based threshold is productive method to improve QoS and to control VM migrations in order to reduce energy consumption. To support this statement TLTHR (Three Layer upper THReshold) policy is proposed. As shown in flowchart if the average VM load \( (AVL_h) \) of particular host \( h \) is greater than threshold parameter \( Uthr_3 \), then TLTHR associate \( UTHR_{h^3} \) to host \( h \) in order to maximize utilization of resources as it considered a small number of heavily loaded VMs on it; when \( (AVL_h) \) of a host is between \( Uthr_2 \) and \( Uthr_3 \), the host is considered to be in scenario2 and associated \( UTHR_{h^2} \), and when \( (AVL_h) \) of a host is between \( Uthr_1 \) and \( Uthr_2 \), then the host will work in at \( UTHR_{h^1} \) to restrict the number of migrations in coming iterations, see figure 2.

\[
AVL_h = \frac{\text{MIPS requested by VMs allocated to host } h}{\text{Number of VMs allocated on host } h}
\]  

(8)
Algorithm 1 provides flexibility in VM consolidation based on the average CPU utilization capacity of VMs served by each host. The basic idea here is the provision of load based threshold that facilitates the CDC to control unwanted VM migrations thus reducing the overall energy consumption while improving SLA. Our proposed work mainly performs two functions. In the first stage, algorithm inspects mean of virtual machines load \( (AVL_h) \) as shown in equation (8) served by host \( h \) (corresponding to Line 1-b in the Algorithm 1). Then values of three layers of upper threshold \( (Uthr_{ht}, Uthr_{ml}, Uthr_{ul}) \) are depicted from the average \( \mu \) and standard \( \rho \) of \( VML \) on each host as shown in equation (9). In later stage all the active hosts are associated with upper threshold values \( UTHR^x_h \) according to their workload scenarios. This association depends on the conditions corresponding to Line 6 in the Algorithm 1 which is further derived from Eq. (9).

\[
Uth = \begin{cases} 
Uthr_{ht} = (\mu - \delta) \\
Uthr_{ml} = \mu \\
Uthr_{ul} = (\mu + \delta)
\end{cases}
\]
Algorithm 1: Three Layer upper Threshold (TLTHR)

Input: host_list, VM list, VM_host allocations

Output: upperthresholdforeachhost

1. For each host h
   a. \( V_h = \text{vms}(h) \) // \( V_h \) list of virtual machines allocated to host h
   b. \( \text{AVL}_h = \frac{\text{vcpu}}{|V_h|} \)
2. End
3. Get Mean(\( \mu \)) of Host’s AVL\( _h \)
4. Get Stddeviation(\( \delta \)) of Host’s AVL\( _h \)
5. Find upper threshold parameters by equation 9
6. For each host
   a. \( \text{if host_util} \geq \text{Uth}_h \)
      Senerio-3 with threshold \( \text{UTHR}^3_h \)
   b. \( \text{else if host_utilization} \geq \text{Uth}_ml \)
      Senerio-2 with threshold \( \text{UTHR}^2_h \)
   c. \( \text{else if host_utilization} \geq \text{Uth}_ul \)
      Senerio-1 with threshold \( \text{UTHR}^1_h \)
   d. \( \text{else} \)
      Senerio-2 with threshold \( \text{UTHR}^2_h \)
7. End

4.2 VM Placement

A review of the traditional and widely used threshold policies revealed that THR in a static class and MAD, IQR in adaptive class are the existing methods for solving the second sub problem whereas, modified best fit decrease algorithm (MBFD) can be applied to solve VM allocation problem. Generally speaking the process of VM deployment can be divided into two parts: initially VMs are allocated based on requested resources by assuming 100% CPU utilization of the host. As discussed in algorithm-1 hosts are associated with particular upper threshold i.e. \( \text{UTHR}^x_h \). Thus after determining the overloaded hosts, double threshold minimum migration policy is used in which one or more VMs are selected based on the lowest required time for migration and all VMs from an underutilized host are also added in migration list. Further, the list of migrable virtual machines are placed on one of the normal loaded host that provides the least increase in power consumption to improve the energy efficiency and restrict the SLA
violations. During this work, a Modification of Best Fit Decrease (MBFD) algorithm is developed and incorporated to optimize the migrations of VMs along with suitable threshold value associated with each host machine. This three layer aware MBFD is responsible to search a host with the nearest optimal VM consolidation with reduction of unnecessary migrations. The pseudo-code of TLMBFD algorithm is as follows:

Algorithm 2: Three Layers Modified Best Fit Decreasing (TLMBFD)

1. Sort vmlist in descending order
2. For every vm in vmlist
   a. MinimumEnergy ← maximum
   b. Allocated host ← null
   c. for each host
      i. if hostutilization > 0 then
      ii. allocate(host, vm)
      iii. if hostutilization < UTHR\textsubscript{h} by TLTHR then
      iv. calculate energy after allocation
      v. if energy ≤ MinimumEnergy then
      vi. Curr_Alocate_host ← host
      vii. MinimumEnergy ← energy
     viii. else
      d. allocate vm to idle host
   e. end
3. Return: VM Consolidation Schedule
4. End

5. PERFORMANCE EVALUATION

5.1 Simulation Environment Setting

In this section, the experimental results of the proposed method are presented and discussed. Computer-based simulation is designed in Matlab to perform the process of VM consolidation that includes data centers, host machines, and virtual machines [43].

For justification and accurate evaluation, the traditional and widely used threshold policies (THR, MAD and IQR) are taken for comparisons. The data center comprising of 800 host machines, each node is modeled to have maximum performance equivalent to \(2660 \times 10^6 \text{ s}^{-1}\), 4096 MB of RAM and two CPU core. Requested tasks are served by 1175 VM on datacenter. Each VM can run any kind of application with dynamic workload. The experimental parameters from existing studies [8]. Discrete event simulation has been chosen to evaluate the performance of the proposed algorithm. Initially, VMs are allocated based on requested resources by assuming 100% CPU utilization of host. Since workload on each VM is provided randomly, we performed each experiment 10 times and noted that the variability in results was negligible, between the different runs [44].
5.2 Performance Metrics

A number of performance metrics are discussed in literature; two main metrics to measure this problem are Service Level Agreement (SLA) and energy consumption, with customers and service providers as stakeholders respectively. High energy efficiency reduces Total Cost of Ownership while customers seek high quality of service. However, these metrics are inversely correlated as SLA violations can be decreased by the increased level of energy consumption. Formulation of energy consumption and instruction energy ratio (IER) is defined in section 3.2. SLA violation metrics caused by SLATH and PDM are referenced from [12]. Another metric is number of migrations also considered.

5.3 Performance Evaluation

Considering the trade-off between energy efficiency and SLA violations, it is important to determine the optimal interval among values of three layers of upper threshold. A series of experiments have been conducted on TLTHR policy to select optimal interval between different threshold layers and their optimal threshold values. Error! Reference source not found.2 and Figure 3 show the energy consumption and SLA violations of all combinations \((U_{th_l}, U_{th_{ml}}, U_{th_{ul}})\) with gap, an integer multiple of 0.02. The results in Error! Reference source not found.2 shows that energy consumption is declining proportionally the interval increases. As SLA violations is also very important for VM deployment so Figure 4 depicts SLA violation is minimum with interval value 12 when \((\text{middle layer }) = 70\).

![Figure 3: Finding gap based on energy consumption](image1)

![Figure 4: Finding gap based on SLA](image2)

After finding the gap between different layers and deciding their threshold values. The process of reorganizing resource allocations is divided into three sub problems. The proposed algorithm is compared
with THR, MAD and IQR in the first phase, Minimum Migration [45] in the second phase and finally allocation is done using three layer Modification of BFD policy. Comparison between the referenced algorithms and TLTHR are presented separately under both scenarios using different metrics, see figure 5 and 6.

![Figure 5: Energy consumption v/s virtual machines]

In Error! Reference source not found., the four overload detection policies are compared with each other in terms of energy consumption. Based on the results, the proposed method consumes 4.09%, 7.6% and 2.18% less energy as compared with the MAD/SM/MM, THR/SM/MM and IQR/SM/MM algorithms respectively.

![Figure 6: Instruction energy ratio v/s virtual machines]

Error! Reference source not found. depict the results obtained by the four algorithms, in terms of IER (Instruction Energy Ratio). Based on the results it can be observed that the proposed algorithm is able to make the system to execute 27.12%, 36.33% and 7.16% more instructions compared with the MAD/SM/MM, THR/SM/MM and IQR/SM/MM algorithms respectively for the same energy consumed. The reason is that the host with CPU utilization below the minimum prescribed condition turns shut down and its load migrated to appropriate host which makes the machines working on near optimal load, see figure 7 and 8.
Figure 7: Number of migration v/s virtual machines

Figure 8: The SLA violation v/s virtual machines

Figure illustrate proposed algorithm has 7.05% less migrations compared with other overload detection policies. Moreover, Figure illustrates that SLA violations also improved. The reason is being less number of migrations due to nearly optimal threshold selection based on the current resource requests.

TABLE:- 3 Simulation Results of Host overload detection algorithms

| POLICY      | ENERGY (kW/S) | SLEEP NODES | MIGRATIONS (# of VMs) | IER in MIPj | SLA× 100 |
|-------------|---------------|-------------|------------------------|-------------|----------|
| THR/SM/MM   | 13990.95      | 664         | 121                    | 31.19833    | 1.869264 |
| IQR/SM/MM   | 13215.61      | 674         | 109                    | 32.87258    | 1.281983 |
| MAD/SM/MM   | 13478.8       | 670         | 116                    | 32.46272    | 1.633064 |
| TLTHR_MM_0.7| 12928.05      | 679         | 107                    | 34.28253    | 1.190218 |

Table 3 denotes the comparisons of TLTHR over the MAD/SM/MM , THR/SM/MM and IQR/SM/MM algorithms respectively indicating improvement in energy consumption, number of idle nodes, number of migrations, SLA and instruction energy ratio.

TABLE:- 4 Percentage improvement in proposed algorithm compare to others in various performance metrics

| REFERENCE ALGORITHM | ENERGY (kW/S) | SLEEP NODES | MIGRATIONS (# of VMs) | IER in MIPj | SLA× 100 | AVG   |
|---------------------|--------------|-------------|------------------------|-------------|----------|-------|
| THR_MM_0.7          | 7.60         | 2.26        | 11.57                  | 36.33       | 9.89     | 13.84 |
| IQR_MM_1.5          | 2.18         | 0.74        | 1.83                   | 7.16        | 4.29     | 5.26  |
From Table 4 and Error! Reference source not found., it is evident that the individual and average percentage has been improved over the reference methods. As the host with huge remaining resource utilization of allocated VMs is more prone to overload hence, TLTHR sets threshold at lower level \((U_{thr_l} = 0.58)\) which prevents the host to allocate more VMs on it. Conversely the host with less remaining resource utilization of allocated VMs is less prone to overload hence TLTHR sets threshold at higher layer \((U_{thr_u} = 0.82)\) which prevents the host to migrate its borderline VMs.

The study in [46] reveals that the higher threshold permits energy efficient VM consolidation at the cost of declining Quality of Service. TLTHR provides optimal Upper Threshold that controls the excessive migrations and prevents hosts to experience 100% utilization of CPU. Hence, better results in cooperation with the MBFD approach are achieved.

6. CONCLUSION

In this study, energy efficiency of large-scale data centers is improved using the proposed TLTHR technique. Based on the TLTHR, scenario based placement policies is presented. The proposed algorithm considers changes as per the dynamic resources requirement to prevent host overloading as well as to improve shutting down of idle PMs. By considering three layers of upper threshold, numbers of migrations are reduced providing better Quality of service. Improved instruction energy ratio is also achieved. Experimental results have shown that the proposed method has better performance than Median Absolute Deviation (MAD), Inter Quartile Range (IQR) and Double Threshold (THR) methods in terms of SLA improvement and number of migrations.

The proposed work outperforms in reducing energy consumption as well as providing Quality of Service but the NP-hard complexity of bin packing problem for multi objective scenario is still a challenging issue for researchers. Our work will help the research upto some extent as we provide three upper thresholds based on the nature of incoming workload. In future, researchers can use workload characteristics based consolidation approaches to solve energy-SLA trade-off. Despite of saving million dollars in energy costs, efficient utilization of resources in cloud computing cut down the carbon footprints to protect environment and shift to a greener and smarter future.

References:-

[1] E. Gelenbe and R. Lent, “Optimising Server Energy Consumption and Response Time,” *Theor.*
16

**Appl. Informatics**, vol. 24, no. 4, pp. 257–270, Jul. 2013, doi: 10.2478/v10179-012-0016-1.

[2] U. Singh and M. Rattan, “Design of linear and circular antenna arrays using cuckoo optimization algorithm,” *Prog. Electromagn. Res. C*, vol. 46, pp. 1–11, 2014, doi: 10.2528/PIERC13110902.

[3] A. T. Abbas *et al.*, “Sustainability assessment associated with surface roughness and power consumption characteristics in nanofluid MQL-assisted turning of AISI 1045 steel,” *Int. J. Adv. Manuf. Technol.*, vol. 105, no. 1–4, pp. 1311–1327, 2019, doi: 10.1007/s00170-019-04325-6.

[4] S. Aggarwal, A. Jindal, R. Chaudhary, A. Dua, G. S. Aujla, and N. Kumar, “EnergyChain: Enabling energy trading for smart homes using blockchains in smart grid ecosystem,” 2018, doi: 10.1145/3214701.3214704.

[5] A. Kumar *et al.*, “Multi-Objective Optimization of WEDM of Aluminum Hybrid Composites Using AHP and Genetic Algorithm,” *Arab. J. Sci. Eng.*, Jul. 2021, doi: 10.1007/s13369-021-05865-4.

[6] B. S. Sidhu, R. Sharda, and S. Singh, “An assessment of water footprint for irrigated rice in Punjab,” *J. Agrometeorol.*, vol. 23, no. 1, pp. 21–29, 2021, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85103944855&partnerID=40&md5=655a1f8b3d23612ca8d9f7059247de7b.

[7] B. S. Sidhu, R. Sharda, and S. Singh, “Water footprint of crop production: A review,” *Indian J. Ecol.*, vol. 48, no. 2, pp. 358–366, 2021, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85105496108&partnerID=40&md5=d09f93f7b95268817d6d0130d114cde9.

[8] M. Soltanshahi, R. Asemi, and N. Shafiei, “Energy-aware virtual machines allocation by krill herd algorithm in cloud data centers,” *Heliyon*, vol. 5, no. 7, pp. 1–9, Jul. 2019, doi: 10.1016/j.heliyon.2019.e02066.

[9] U. Wajid *et al.*, “On Achieving Energy Efficiency and Reducing CO2 Footprint in Cloud Computing,” *IEEE Trans. Cloud Comput.*, vol. 4, no. 2, pp. 138–151, Apr. 2016, doi: 10.1109/TCC.2015.2453988.

[10] A. Hameed *et al.*, “A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems,” *Computing*, vol. 98, no. 7, pp. 751–774, 2016, doi: 10.1007/s00607-014-0407-8.

[11] S. S. Manvi and G. Krishna Shyam, “Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey,” *J. Netw. Comput. Appl.*, vol. 41, no. 1, pp. 424–440, 2014, doi:
10.1016/j.jnca.2013.10.004.

[12] A. Beloglazov and R. Buyya, “Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers,” *Concurr. Comput. Pract. Exp.*, vol. 24, no. 13, pp. 1397–1420, 2012, doi: 10.1002/cpe.1867.

[13] L. Wu, S. K. Garg, and R. Buyya, “SLA-based resource allocation for software as a service provider (SaaS) in cloud computing environments,” *Proc. - 11th IEEE/ACM Int. Symp. Clust. Cloud Grid Comput. CCGrid 2011*, pp. 195–204, 2011, doi: 10.1109/CCGrid.2011.51.

[14] S. Esfandiarpoor, A. Pahlavan, and M. Goudarzi, “Structure-aware online virtual machine consolidation for datacenter energy improvement in cloud computing,” *Comput. Electr. Eng.*, vol. 42, no. 6, pp. 74–89, 2015, doi: 10.1016/j.compeleceng.2014.09.005.

[15] C. J. Huang et al., “An adaptive resource management scheme in cloud computing,” *Eng. Appl. Artif. Intell.*, vol. 26, no. 1, pp. 382–389, 2013, doi: 10.1016/j.engappai.2012.10.004.

[16] D. A. Alboaneen, B. Prangono, and H. Tianfield, “Energy-aware virtual machine consolidation for cloud data centers,” in *Proceedings - 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing. London, UK*, 2014, pp. 1010–1015, doi: 10.1109/UCC.2014.166.

[17] Z. Zhou, Z. Hu, and K. Li, “Virtual Machine Placement Algorithm for Both Energy-Awareness and SLA Violation Reduction in Cloud Data Centers,” *Sci. Program.*, vol. 2016, no. 15, pp. 1–11, 2016, doi: 10.1155/2016/5612039.

[18] N. J. Kansal and I. Chana, “Energy-aware Virtual Machine Migration for Cloud Computing - A Firefly Optimization Approach,” *J. Grid Comput.*, vol. 14, no. 2, pp. 327–345, 2016, doi: 10.1007/s10723-016-9364-0.

[19] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, “Black-box and Gray-box Strategies for Virtual Machine Migration,” Amherst, MA 01003, United States, 2009.

[20] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, “Sandpiper: Black-box and gray-box resource management for virtual machines,” *Comput. Networks*, vol. 53, no. 17, pp. 2923–2938, Dec. 2009, doi: 10.1016/j.comnet.2009.04.014.

[21] M. Marzolla, O. Babaoglu, and F. Panzieri, “Server consolidation in clouds through gossiping,” in *2011 IEEE Int. Symp. a World Wireless, Mob. Multimed. Networks, WoWMoM 2011 - Digit. Proc.*, vol. 11, no. 5, pp. 1–6, 2011, doi: 10.1109/WoWMoM.2011.5986483.

[22] S. Mehta and A. Neogi, *NOMS 2008-2008 IEEE Network Operations and Management*
Symposium. Salvador, Bahia: IEEE Network Operations and Management Symposium, 2008.

[23] C. Ghribi, M. Hadji, and D. Zeghlache, “Energy efficient VM scheduling for cloud data centers: Exact allocation and migration algorithms,” in Proceedings - 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, CCGrid 2013, 2013, pp. 671–678, doi: 10.1109/CCGrid.2013.89.

[24] Y. L. Min, F. Rawson, T. Bletsch, and V. W. Freeh, “PADD: Power-aware domain distribution,” in 29th IEEE International Conference on Distributed Computing Systems, Montreal, QC, 2009, pp. 239–247, doi: 10.1109/ICDCS.2009.47.

[25] Í. Goiri et al., “Energy-efficient and multifaceted resource management for profit-driven virtualized data centers,” Futur. Gener. Comput. Syst., vol. 28, no. 5, pp. 718–731, May 2012, doi: 10.1016/j.future.2011.12.002.

[26] F. Farahnakian, T. Pahikkala, P. Liljeberg, J. Plosila, N. T. Hieu, and H. Tenhunen, “Energy-aware VM consolidation in cloud data centers using utilization prediction model,” IEEE Trans. Cloud Comput., vol. 7, no. 2, pp. 524–536, Apr. 2019, doi: 10.1109/TCC.2016.2617374.

[27] T. Cioara, I. Anghel, I. Salomie, G. Copil, D. Moldovan, and A. Kipp, “Energy aware dynamic resource consolidation algorithm for virtualized service centers based on reinforcement learning,” Proc. - 2011 10th Int. Symp. Parallel Distrib. Comput. ISPDC 2011, pp. 163–169, 2011, doi: 10.1109/ISPDC.2011.32.

[28] A. Mosa and R. Sakellariou, “Virtual machine consolidation for cloud data centers using parameter-based adaptive allocation,” ACM Int. Conf. Proceeding Ser., vol. Part F1305, no. 16, pp. 1–10, 2017, doi: 10.1145/3123779.3123807.

[29] A. Mazrekaj, D. Minaroli, and B. Freisleben, “Distributed resource allocation in cloud computing using multi-agent systems,” Telfor J., vol. 9, no. 2, pp. 110–115, 2017, doi: 10.5937/telfor1702110m.

[30] M. A. H. Monil and R. M. Rahman, “VM consolidation approach based on heuristics fuzzy logic, and migration control,” J. Cloud Comput., vol. 5, no. 1, 2016, doi: 10.1186/s13677-016-0059-7.

[31] M. Ranjbari and J. Akbari Torkestani, “A learning automata-based algorithm for energy and SLA efficient consolidation of virtual machines in cloud data centers,” J. Parallel Distrib. Comput., vol. 113, no. 1, pp. 55–62, 2018, doi: 10.1016/j.jpdc.2017.10.009.

[32] R. Moreno-Vozmediano, R. S. Montero, E. Huedo, and I. M. Llorente, “Efficient resource provisioning for elastic Cloud services based on machine learning techniques,” J. Cloud Comput.,
vol. 8, no. 1, Dec. 2019, doi: 10.1186/s13677-019-0128-9.

[33] 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 Comput.*, vol. 12, no. 1, pp. 1–15, 2009, doi: 10.1007/s10586-008-0070-y.

[34] R. Yadav, W. Zhang, K. Li, C. Liu, M. Shafiq, and N. K. Karn, “An adaptive heuristic for managing energy consumption and overloaded hosts in a cloud data center,” *Wirel. Networks*, vol. 26, no. 3, pp. 1905–1919, 2020, doi: 10.1007/s11276-018-1874-1.

[35] R. Yadav, W. Zhang, O. Kaiwartya, P. R. Singh, I. A. Elgendy, and Y. C. Tian, “Adaptive Energy-Aware Algorithms for Minimizing Energy Consumption and SLA Violation in Cloud Computing,” *IEEE Access*, vol. 6, pp. 55923–55936, 2018, doi: 10.1109/ACCESS.2018.2872750.

[36] S. Y. Hsieh, C. S. Liu, R. Buyya, and A. Y. Zomaya, “Utilization-prediction-aware virtual machine consolidation approach for energy-efficient cloud data centers,” *J. Parallel Distrib. Comput.*, vol. 139, pp. 99–109, 2020, doi: 10.1016/j.jpdc.2019.12.014.

[37] A. Aryania, H. S. Aghdasi, and L. M. Khanli, “Energy-Aware Virtual Machine Consolidation Algorithm Based on Ant Colony System,” *J. Grid Comput.*, vol. 16, no. 3, pp. 477–491, Sep. 2018, doi: 10.1007/s10723-018-9428-4.

[38] A. N. A. Verma, P. Ahuja, “pMapper: Power and Migration Cost Aware Application Placement in Virtualized Systems,” in *Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware*, IIT Delhi, 2008, pp. 243–264, doi: 10.1108/IJAIM-04-2014-0025.

[39] Q. Zheng et al., “Virtual machine consolidated placement based on multi-objective biogeography-based optimization,” *Futur. Gener. Comput. Syst.*, vol. 54, no. 1, pp. 95–122, Jan. 2016, doi: 10.1016/j.future.2015.02.010.

[40] X. Zhang et al., “Energy-aware virtual machine allocation for cloud with resource reservation,” *J. Syst. Softw.*, vol. 147, no. 1, pp. 147–161, 2019, doi: 10.1016/j.jss.2018.09.084.

[41] M. K. Gupta and T. Amgoth, “Resource-aware virtual machine placement algorithm for IaaS cloud,” *J. Supercomput.*, vol. 74, no. 1, pp. 122–140, 2018, doi: 10.1007/s11227-017-2112-9.

[42] M. A. Khan, A. Paplinski, A. M. Khan, M. Murshed, and R. Buyya, “Dynamic virtual machine consolidation algorithms for energy-efficient cloud resource management: A review,” in *Sustainable Cloud and Energy Services: Principles and Practice*, Springer International Publishing, 2017, pp. 135–165.

[43] C. Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, esar A. F. De Rose3, and and R.
Buyya, “CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,” *Softw. - Pract. Exp.*, vol. 41, no. 7, pp. 23–50, 2011, doi: 10.1002/spe.

[44] Yongqiang Wu, “Energy Efficient Virtual Machine Placement in Data Centers A thesis submitted as a requirement for the degree of Master of Information Technology Queensland University of Technology,” 2013.

[45] A. Beloglazov, J. Abawajy, and R. Buyya, “Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing,” *Futur. Gener. Comput. Syst.*, vol. 28, no. 5, pp. 755–768, 2012, doi: 10.1016/j.future.2011.04.017.

[46] Z. Zhou, Z. G. Hu, T. Song, and J. Y. Yu, “A novel virtual machine deployment algorithm with energy efficiency in cloud computing,” *J. Cent. South Univ.*, vol. 22, no. 3, pp. 974–983, 2015, doi: 10.1007/s11771-015-2608-5.