2024

EEVMC: An Energy Efficient Virtual Machine Consolidation Approach for Cloud Data Centers

Attique Ur Rehman

Songfeng Lu

Mubashir Ali

Florentin Smarandache

Sultan S. Alshamrani

See next page for additional authors

Authors
Attique Ur Rehman, Songfeng Lu, Mubashir Ali, Florentin Smarandache, Sultan S. Alshamrani, Abdullah Alshehri, and Farrukh Arslan

IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 12 June 2024, accepted 9 July 2024, date of publication 16 July 2024, date of current version 8 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3429424

RESEARCH ARTICLE

EEVMC: An Energy Efficient Virtual Machine Consolidation Approach for Cloud Data Centers

ATTIQUE UR REHMAN[1], SONGFENG LU[1,2], MUBASHIR ALI[3], FLORENTIN SMARANDACHE[4], SULTAN S. ALSHAMRANI[5], ABDULLAH ALSHEHRI[6], AND FARRUKH ARSLAN[7]

[1]School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
[2]Shenzhen Huazhong University of Science and Technology Research Institute, Shenzhen 518057, China
[3]Department of Software Engineering, Lahore Garrison University, Lahore 54810, Pakistan
[4]Mathematics, Physics, and Natural Science Division, The University of New Mexico, Gallup, MN 87301, USA
[5]Department of Information Technology, College of Computer and Information Technology, Taif University, Taif 21944, Saudi Arabia
[6]Department of Information Technology, Faculty of Computing and Information Technology, Al-Baha University, Al Bahah 65779, Saudi Arabia
[7]Department of Electrical Engineering, University of Engineering and Technology, Lahore 54500, Pakistan

Corresponding author: Farrukh Arslan (farrukh_arslan@uet.edu.pk)

This work was supported by Taif University, Saudi Arabia, under Project TU-DSPP-2024-52.

ABSTRACT The dynamic landscape of cloud computing design presents significant challenges regarding power consumption and quality of service (QoS). Virtual machine (VM) consolidation is essential for reducing power usage and enhancing QoS by relocating VMs between hosts. OpenStack Neat, a leading framework for VM consolidation, employs the Modified Best-Fit Decreasing (MBFD) VM placement technique, which faces issues related to energy consumption and QoS. To address these issues, we propose an Energy Efficient VM Consolidation (EEVMC) approach. Our method introduces a novel host selection criterion based on the incurred loss during VM placement to identify the most efficient host. For validation, we conducted simulations using real-time workload traces from Planet-Lab and Materna over ten days, leveraging the latest CloudSim toolkit to compare our approach with state-of-the-art techniques. For Planet-Lab's workload, our EEVMC approach shows a reduction in energy consumption by 80.35%, 59.76%, 21.59%, and 7.40%, and fewer system-level agreement (SLA) violations by 94.51%, 94.85%, 47.17%, and 17.78% when compared to Modified Best-Fit Decreasing (MBFD), Power-Aware Best Fit Decreasing (PABFD), Medium Fit Power Efficient Decreasing (MFPED), and Power-Efficient Best-Fit Decreasing (PEBFD), respectively. Similarly, for Materna, EEVMC achieves a reduction in energy consumption by 16.10%, 61.0%, 4.94%, and 4.82%, and fewer SLA violations by 76.99%, 88.88%, 12.50%, and 48.65% against the same benchmarks. Additionally, Loss-Aware Performance Efficient Decreasing (LAPED) significantly reduces the total number of VM migrations and SLA time per active host, indicating a substantial improvement in cloud computing efficiency.

INDEX TERMS Virtual machine consolidation, quality of service, energy efficient, VM migration, placement algorithm, OpenStack cloud.

I. INTRODUCTION

Cloud computing has emerged as a revolutionary computational technique, offering access to on-demand computational resources [1]. The concept of virtualization provides access to physical resources in a virtual manner. Virtual machines (VMs) run on these physical resources, and each VM has its own operating system facilitating the scheduling of various user requests [2]. The significant challenge in computational resource handling is the increasing power consumption and poor quality of service (QoS). Advancements in the design and functionality of cloud applications have significantly increased the power consumption of data centers [3]. The power consumption in data centers is projected to rise from 286 TWh to 752 TWh by 2030 which is 2.13% of global power availability [4]. On the other hand, maintaining

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

QoS while providing access to physical resources is also a significant challenge. The data centers need to fulfill the demand from cloud users. This demand is referred to as a service level agreement (SLA) [5]. One way to maintain the SLA is to provide enough resources to fulfill each demand. In this way, more physical resources will be used, and power consumption will increase. If more requests are being accommodated for one or more physical resources, then there are more chances that the physical resource(s)/host(s) will be overutilized and SLA will be violated [5]. To resolve this problem, the concept of virtual machine consolidation plays an important role which migrating the virtual machines from one physical resource to another if the host is overutilized or underutilized without interruption in services [6]. If the utilization of a host falls below a specific threshold, then the host is considered to be underutilized. VMs running on the underutilized host have to migrate to another host by efficiently using the physical resources [7]. On the contrary, VMs are migrated to another host to reduce the workload if the host is overutilized. If these VM(s) migrations are taken online, then this migration is called the Online or live migration approach [8]. In this regard, the efficient algorithm minimizes energy consumption, reduces the SLA violations and the total number of VM migrations [9]. It also reduced the time in which an active host faces 100% of its utilization [10].

The VM consolidation can be accomplished in multiple ways depending on the needs, objectives, and targets [11]. It also depends on which type of computational technique is used to solve the problem [12]. The utilization of physical resources in the normal data centers is an average of 12% to 18% [13], which is low utilization compared to cloud data centers [14]. This low utilization causes more power consumption [15]. While in the cloud data centers, the average utilization is found near about 40% to 70%. So, VM consolidation plays a vital role in reducing energy consumption in cloud data centers [16]. Multiple factors affect the VMC technique such as the Band Width of the Network [17], SLA [18], switching of the power cycles [19], performance loss overheads due to migrations [20], load balancing, virtual machine affinity [21], reliability, and resource utilization [22]. Generally, to improve the energy efficiency, these factors are taken into consideration [23].

We developed an open-source VM consolidation approach for Open-Stack clouds. This framework mainly addresses the four significant issues. Firstly, the framework identifies overutilized and underutilized hosts by comparing current utilization with a fixed threshold value. It performs robust statistical analyses, including Median Absolute Deviation (MAD), Interquartile Range (IQR), Local Regression (LR), and Local Regression Robust (LRR), based on the historical utilization data of the hosts [24]. Secondly, if a host is detected as overutilized or underutilized, the framework selects VMs for migration using strategies such as Minimum Migration Time (MMT), Maximum Correlation (MC), Minimum Utilization (MU), and Random Selection (RS). The combination of these strategies helps achieve a balance between energy efficiency, performance, and SLA compliance [25]. Thirdly, the VM placement process follows the Power-Aware Best-Fit Decreasing (PABFD) technique by default. This method is enhanced with additional efficient heuristics like Power-Efficient First-Fit Decreasing (PEFFD), Power-Efficient Best-Fit Decreasing (PEBFD), and Medium-Fit Power-Efficient Decreasing (MFPED), which outperform the default heuristics in the CloudSim toolkit [26]. Finally, the framework integrates the unique logic and benefits of each VM selection strategy and heuristic to address various aspects of VM consolidation challenges in cloud data centers. This comprehensive approach ensures effective resource management and optimized performance of cloud environments [27]. Moreover, we employed efficient heuristics (i.e., PEFFD, PEBFD, MFPED), which outperforms the default heuristics for this framework [28].

The novelty of this paper is the introduction of an Energy Efficient Virtual Machine Consolidation (EEVMC) approach that employs a unique host selection criterion based on the loss incurred during VM placement. This new criterion focuses on minimizing both energy consumption and SLA violations, setting it apart from existing methods. The EEVMC approach was validated using real-time workload traces from Planet-Lab and Materna, with simulations conducted via the latest CloudSim toolkit. Results show significant improvements over state-of-the-art techniques like MBFD, PABFD, MFPED, and PEBFD, in both energy consumption and SLA compliance. By decomposing the system into multiple losses and selecting hosts based on their fitness value, the EEVMC approach optimizes VM placement and reduces energy consumption and SLA violations. This method is particularly useful for cloud computing applications that require efficient resource management and energy conservation, enhancing the performance and sustainability of large-scale cloud data centers.

The main contributions of this work are listed below:

- To introduce an energy-efficient VM consolidation (VMC) technique that reduces power consumption and SLA violations within the OpenStack Neat framework by efficiently placing VMs on available hosts.
- The proposed technique efficiently performed VM placement in the existing framework with flexibility and dynamism to adapt to system requirements, ensuring implementation without incurring additional costs.
- The proposed approach minimized the VM migrations, reduced the mean-time and standard deviation leading up to a host shutdown, and decreased the duration during which a single host operates at 100% of its utilization capacity.

The rest of the paper is organized as follows; section II discusses the related studies. The proposed approach is elaborated in section III and results are presented in section IV. Finally, section V concludes the paper with future directions.

II. LITERATURE WORK

VM consolidation is an NP-hard problem, and different solutions have been proposed such as dynamic programming, Constraint satisfaction, and linear programming [28]. M. Rezaei-Mayahi et al. proposed a solution to the VM placement as Integer linear programming (ILP), which locates the most suitable host based on its power consumption [29]. They have considered the number of active hosts, shelves, and the relationship between the active hosts to minimize the rotational airflow in cloud corridors. Medara, et al. presented an approach related to the efficiency constraints of hosts and VMs [5]. If the constraint is satisfied, then it may enhance the host's efficiency.

Integrating deep learning into the CloudSim toolkit can significantly enhance the optimization of simulation results for energy-efficient VM consolidation in cloud data centers. Deep learning models, trained on historical data, can accurately predict resource usage patterns and optimize VM placement decisions, thereby reducing energy consumption and SLA violations. By considering various factors such as CPU and memory usage, network bandwidth, and power consumption, these models can dynamically adjust resource allocation policies in real-time. This integration enables CloudSim to simulate more realistic scenarios, ultimately improving the overall energy efficiency and QoS of cloud data centers.

Authors [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42] on the strategy for splitting resource assignment, Virtual Machine Consolidation (VMC) techniques are divided into two categories: static and dynamic. The taxonomy considers several parameters. Table 1 provides the metrics that are believed to be successful in achieving the goals, such as lowering power consumption. The data set and assessment methodologies are two more important aspects to classify. Synthetic and actual data sets are included in the data sets. Simulation, implementation, hybrid (simulation plus implementation), and formal approaches are among the evaluation methods.

The EEVMC approach builds upon the foundational concepts of Sandpiper and incorporates advanced techniques from recent literature to enhance energy efficiency and performance in cloud data centers [33], [34]. By addressing power-performance trade-offs, interference-aware migration, reliability, heterogeneity, and performance overhead, EEVMC aims to provide a comprehensive solution for efficient VM consolidation. Integrating these insights can lead to improved resource management and sustainability in large-scale cloud environments [32], [42].

This technique determines the magnitude of the pessimistic effect of VMs on each other following the specific VMs allocated to the same host. Affinitive VMs and those VMs that made each other performance as least coherent may be placed on the host. Medara, et al. develop the inter and intra-cloud data center traffic by making a suitable choice of VM placement [5]. This method reduces the energy consumption but increasing the distances causes inter VMs to significantly reduce the available bandwidth, which further causes the delay in the application execution and reduces the QoS. In another work, the authors proposed the VM placement problem as Particle swarm optimization [7]. This algorithm was efficient in terms of energy efficiency and reducing the *SLA* violations. They have provided a trade-off between reducing power consumption and SLA violations. Specifically related to the underlying framework, the authors proposed the Power-Aware best-fit decreasing (PABFD) heuristic [3]. This VM placement problem was a simple best-fit bin-packing heuristic in which the bin represents the physical nodes. To apply this algorithm, they first sorted the VMs according to decreasing utilization. Afterward, they allocated the VMs to the hosts so that the increase in power would be mini-mum. PABFD chooses a host with a minimum increase in power, but such selection may increase the SLA violations. This technique does not consider the idle power of the hosts, which decreases the system's energy efficiency overall. It can be possible that a host gives a minuscule increase in power after VM placement, but such allocation, in turn, increases the total VM migrations and SLA violations. Another efficient heuristic, modified best-fit decreasing (MBFD), was proposed by the [45]. MBFD reduces energy consumption and SLA violations. The motivation behind this technique is to utilize the resources efficiently. Some VMs have over-provisioned the applications running on them. On the contrary, some VMs are running with very low utilization. This algorithm handles the available CPU MIPS resources such that the allocated host will have the least available CPU MIPS at the current time. If the situation is a tie between two hosts, the host with minimum RAM available will be allocated. MBFD is considered the baseline algorithm for many energy-efficient approaches. A. Aryania, et al. proposed VM placement algorithms Modified Worst fit decreasing VM placement (MWFDVP) in which worst-fit bin-packing heuristic is followed. In this, algorithm VMs are sorted according to decreasing utilization [46]. The host with a maximum increase in power is selected as the allocated host. They have presented another version of their approach called Second Worst Fit Decreasing (SWFDVP). This heuristic was the same as the MWFDVP. The only difference is that it selects the second host, which gives a maximum increase in power. They have further provided another approach of the Modified K-means (MK) algorithm, which modifies the sorting of VMs in K-means clustering and makes the clusters of VMs that further apply in Modified Worst-fit Placement clustering (MWFCP-C). In this algorithm, the VMs from dense clusters are allocated first to the hosts. This technique improves power efficiency compared to baseline algorithms, but this solution was computationally expensive as sorting takes $O(n \log n)$, and K-means clustering takes $O(n2)$ while n is the number of migrating VMs. So, this solution can be considered adequate when the number of migrating VMs is not very large.

TABLE 1. Static and dynamic consolidation of virtual machine.

Articles	Dynamic Resource Assignment Policy	Performance	Reliability	Cooling	Thermal Heating	Dataset/ Evaluation
F. Xu, et al. [34]	✓			✓	✓	Synthetic data set/ Cloud Sim
Wei Deng, et al. [33]				✓	✓	Synthetic data set/ Cloud Sim
F. Xu, et al. [32]	✓	✓				Co, Mon project/Cloud Sim
Y. Saadi, et al. [42]			✓		✓	Real time data/ Cloud Sim
Khizer Abbas, et al. [41]		✓	✓		✓	Real time data/ Cloud Sim
H. Ali , et al. [40]	✓		✓			Synthetic data set/ Cloud Sim
M. Alam, et al. [39]	✓	✓	✓		✓	Synthetic data set/ Cloud Sim
M. Alam, et al. [38]	✓	✓	✓			google Data Center/ execution
Z. Li et al. [37]	✓	✓				google/ Cloud Sim
SM. Rozehkhani et al. [36]	✓			✓	✓	SPE Crower/ Cloud Sim
JY. Luo et al. [35]	✓	✓				Synthetic data set/ DC Sim
F. Moges et al. [25]	✓	✓	✓			SPEC/ (MATLAB) Simulation
M. Kumar et al.[24]				✓	✓	Co. Mon project: Planet lab/ Cloud Sim
J. Witanto et al. [33]	✓	✓	✓		✓	EC2 Amazon/ Cloud Sim
P. Bodik et al. [20]		✓				Co. Mon project: Planet lab/ Cloud Sim
W. Yao et al. [44]			✓			Google DC, TU-Berlin)/ execution
U. Arshad et al. [33]		✓				Google/ Cloud Sim
J. Cao et al. [43]		✓				Real data/ Cloud Sim
Purposed work	✓	✓	✓	✓	✓	Planet lab/Cloud Sim toolkit

In recent years more efficient heuristics methods have been proposed by [47]. These heuristics were Power-efficient first fit, decreasing (PEFFD) power-efficient best fit decreasing (PEBFD), and Medium-fit power-efficient decreasing (MFPED). In these algorithms, authors have declared the criteria of a host being an efficient host by dividing the total MIPS (Number of instructions a host can execute in one second) by the maximum power of the host. Based on these criteria, they have applied the First-fit heuristic rather than the best-fit heuristic for the PEFFD and PEBFD. For MFPED, they have taken the absolute value of the utilization of CPU from a fixed desired value considered as fitness value. Afterward, they applied the BFD algorithm for the host offering the best fitness value as the least absolute distance. The authors have shown the improvements of their techniques by comparing them with the baseline algorithms MBFD and PABFD.

According to the efficient heuristic criteria discussed earlier, an efficient heuristic should place the VMs such that the total energy consumption will be minimal, and there will be fewer SLA violations. There are many existing standard bin-packing heuristics, i.e., First Fit (FF), Best Fit (BF), Worst-Fit (WF), Next-Fit (NF), and Any-Fit (AF), but these techniques are suitable for the homogeneous hosts. In cloud data centers, we have heterogeneous hosts, so there is a need to modify these standard heuristics to define the fitness criteria for VM placement. An algorithm is said to be more efficient than baseline algorithms if it minimizes the energy consumption on the same SLA or minimizes the SLA violations by keeping the same energy consumption or minimizes the SLA and energy consumption. Our objective is to present an efficient algorithm and test the algorithm on real-time workload traces and host configurations to show the efficiency over baseline algorithms. In our work, we have presented a new approach to declare the fitness of available hosts. Our approach decomposes the system into multiple losses. The criteria for selecting a suitable host for allocation is based on the fitness value, and the host with the maximum fitness value will be selected for allocation. We have compared our technique with four baseline algorithms MBFD, PABFD, PEBFD, and MFPED,and the experiment results show our approach outperforms the baseline approaches by maximizing the energy efficiency and reducing the SLA violations.

III. PROPOSED APPROACH

Before discussing the proposed work, it is necessary to discuss the worst-case asymptotic performance ratio (WCAPR). This ratio determines the proposed solution's distance compared to the optimal solution in the worst-case performance ratio. Let ALG(m) be the number of hosts when m numbers of VMs are going to be allocated to the hosts by any Algorithm and OPT (m) be the number of hosts allocated by an optimal solution for m number of VMs. Then the following equation explains the number of hosts taken by algorithm ALG to the optimal algorithm [27].

However, the paper could benefit from a more detailed discussion of how WCAPR is used to guide the design and implementation of the EEVMC approach. Specifically, the authors could clarify:

Comparative Analysis: How does the WCAPR of the EEVMC algorithm compare to existing algorithms like MBFD, PABFD, MFPED, and PEBFD?

Algorithm Design: How insights from WCAPR influenced the design choices made in the EEVMC algorithm, particularly in selecting hosts and VMs for migration.

Performance Guarantees: Explicit references to WCAPR in the results section to highlight the worst-case performance guarantees of the EEVMC approach.

The flowchart of the Python-controlled Energy-efficient Genetic Algorithm (EGA) optimization design begins with initializing parameters and settings, such as data size, crossover rate, mutation rate, and selection criteria, followed by generating a random initial data size of solutions. Each solution's effectiveness is then evaluated using a fitness function that quantifies energy efficiency and Quality of Service (QoS), with VM ranked accordingly. Selection processes of VMs like selection prioritize higher best-fit scores for VMC, leading to crossover operations that combine parts of VMC to create diverse offspring. Mutation further introduces diversity by randomly altering VM, maintaining a balance between exploration. The new VM is then transmitted to the Control and Simulation Tool (CST) module for simulation in a cloud data center environment, where performance data on energy consumption and QoS is collected. This data is used to update the fitness function, guiding the EGA toward better solutions over multiple generations. Iterations continue until predefined convergence criteria are met. Finally, the best solution is selected based on fitness score and undergoes further validation to ensure robustness and effectiveness. This comprehensive description ensures the clarity and reproducibility of the optimization design, ultimately enhancing the efficiency of virtual machine consolidation in cloud data centers.

$$\left\{\begin{array}{l} R_{ALG}\left(m\right)=\frac{ALG(m)}{OPT\left(m\right)} \\ APR\left(ALG\right)\equiv inf\left\{r\geq 1: ForsomeN>0,\right. \\ \left. R_{ALG}\left(m\right)\leq r\forall LwithOPT\left(m\right)\geq N\right\} \end{array}\right\} \tag{1}$$

In above equation $(r \in \mathbf{R}|r \geq R_{ALG}(m))$. WCAPR for 17/10 for FF and BF heuristics. For FFD and BFD the WCAPR is 11/9.

A. LOSS-AWARE PERFORMANCE EFFICIENT DECREASING

We have divided the available hosts into two categories. The first category is related to the active hosts at time t, which at least have one VM allocated to them, and the other category is related to the active hosts that are not active at time t. The time t denotes the time at which the decision of VM allocation is taken. For each category, we compute the current loss such that the total maximum loss is one and the minimum is zero.

Suppose we have the host list HostList = {h1, h2, …, hn} while n is the total number of available hosts and VmList= {vm1, vm2, …, vmm} is the VM list having a total number of m VMs to be allocated. If Pi(t) is the power taken by the hi at time t and P_i^{max} is the maximum power taken by the host hi define the current power ratio of hi as $CPR_i(t)$.

$$\left\{\begin{array}{l} CPR_i\left(t\right)=\frac{P_i(t)}{P_i^{max}} \\ CMR_i\left(t\right)=\frac{CPU_i(t)}{CPU_i^{max}} \end{array}\right\} \tag{2}$$

This shows $max\left(CPR_i(t)\right) = 1.IfCPU_i\left(t\right)$ denotes the current available frequency of hi at time t and CPUmax is the maximum frequency of hi define the mips ratio of hi as $CMR_i(t)$.

1) FITNESS OF ACTIVE HOSTS

The fitness metrics for active and inactive hosts are designed to balance power efficiency and performance, ensuring that VMs are allocated to the most suitable hosts based on their current power consumption and CPU utilization. This detailed mechanism provides a structured way to optimize VM allocation in cloud data centers, enhancing overall energy efficiency and service quality. Among all the hosts, we will first select the hosts that will not be over-utilized after VM allocation. We call these hosts as available hosts and the hosts declared in the HostList are all available hosts. The host overutilization detection is out of the scope of this paper, and we will use any of the already discussed overutilization detection policies in section I. A host hi is considered an active host at time t if the host is available such that at least one VM has already been assigned to it, and it is currently active. Whenever a decision of VM allocation is made, this host is not required to be activated, for which host activation loss will not incur. As we have already discussed, the maximum loss for the active hosts is one, and the minimum loss is zero; therefore, we assign the active loss factor $\{\alpha|0 \leq \alpha \leq 1\}$ as the weight of importance to each loss. The active host loss AHLi(t) of hi at time t can be defined as follows:

$$AHL_i\left(t\right)=\alpha\times CPR_i\left(t\right)+\left(1-\alpha\right)\times CMR_i\left(t\right) \tag{3}$$

The Fit active host FAH(t) can be determined as $\left\{hi|min\left\{AHLi\left(t\right)\right\}_{i=1}^{n};\forall i\in activehost\right\}$ the host giving the minimum loss at time t. This can be computed as while n is the total number of available hosts. This method is useful for selecting the efficient host among active hosts because of the heterogeneity of the hosts in data centers. Hosts have different power consumption at different utilization levels. Compared to PEBFD and PEFFD, we have defined active host fitness by giving importance to the increase in power and frequency. Such selection gives a suitable choice of the host at the time of selection by deciding the importance of power consumption and available CPU frequency as shown in Table 2.

TABLE 2. Hosts characteristics at time (t).

Host	$P_i(t)$	P^{max}	$CPU_i(t)$	CPU_i^{max}
1	52.3	113	1205	2933
2	61.8	113	2081	3067

We compute for *h1* *CPR1(t)* = 0.4628, *CMR1(t)* = 0.4108 and the same way we can compute *CPR2(t)* = 0.5469, *CMR2(t)* = 0.6785 for *h2*. We set $\alpha = 0.6$ for this problem and compute *AHL1(t)* = 0.442 and *AHL2(t)* = 0.599. As *h1* is giving the minimum loss which leads to *FAH(t)* = *hi*then it will be suitable choice for Vm allocation at time *t*.

2) FITNESS OF INACTIVE HOSTS

A host *hi* is considered an inactive host if no VM is allocated to it at time *t* and it is currently inactive. Whenever a decision of VM allocation is made to this host, it is first required to be activated, which will incur host activation loss. If there are any conditions in which active host selection is unsuitable or active hosts will be overutilized after VM placement, we have to select the most suitable host among inactive hosts. In this section, we will discuss the selection criteria for the inactive host.

If *hi* is the current inactive host then we compute the maximum CPU frequency of *hi* and compare it. The maximum CPU frequency among all the inactive hosts we denote it as

$$MAXCPU = \left\{CPU_i^{max} | max \left\{CPU_i^{max}\right\}_{i=1}^{n}; \forall i \in inactivehosts\right\} \quad (4)$$

If the difference between *MAXCPU* and CPU_i^{max} is minimum then the loss will be minimum. We further compute the current power ratio by considering if the inactive host is bringing to an idle state from an inactive state. The host which will consume minimum power will have the minimum loss. For this purpose, we will compute the current power ratio of the host *hi*. We further assign inactive loss factor $\{\beta | 0 \leq \beta \leq 1\}$ as the weight of importance to each loss in the inactive host. We compute inactive host loss *IHLi(t)* of *hi* at time *t* as follows. *IHLi(t)* of *hi* at time *t* as follows:

$$IHL_i(t) = \beta \times CPR_i(t) + (1-\beta) \times \left(\frac{MAXCPU - CPU_i^{max(t)}}{MAXCPU}\right) \quad (5)$$

The *Fit* inactive host *FIH(t)* can be determined as the host giving the minimum loss at time *t*. This can be computed as $hi | min \{IHLi(t)\}_{i=1}^{n}; \forall i \in inactivehost$ while *n* is the total number of available hosts.

B. ALGORITHM

The algorithm 1 describes the LA-PED. It takes HostList, VmList, and MAXCPU as input and returns VmPlacement as output. We initialize the α, β and set the values according to requirements in the initial phases. Then we will assign null values to FIH, FAH, and the allocated host in the initial phase. At line 7, we will sort the VmList according to decreasing the CPU utilization. Then at line 10, we check the feasibility of the current host for VM allocation. This feasibility check will contain the availability of resources and the host overutilization detection. If the host is not feasible, it will be skipped, and we check the next host. If the host is feasible, then for each VM in VmList, we will check the FAH or FIH depending on the host's current state. We will select the allocated host and add it to the VM placement list. The run-time complexity of LA-PED is O (n log(n) + (n × m)) while n is the total number of migrating VMs and m is the total number of available hosts.

This research should explore the integration steps and technical requirements for implementing the proposed VM consolidation approach, offering detailed guidelines for a seamless process. It should evaluate the impact on management complexity, comparing it with existing methods to assess administrative overhead and operational practicality. Additionally, the influence on user experience, including service performance and satisfaction, should be investigated through user feedback and case studies. Addressing these aspects will provide a comprehensive view, ensuring the approach meets the needs of both administrators and end-users.

Algorithm 1 Loss-aware performance efficient decreasing LA-PED

```
   Input: HosList,VmList,MAXCPU
   Output: VmPlacement
1  initialize ·α, β
2  FAH← null
3  FIH← null
4  allocatedHost← null
5  Max_AH_efficiency← MaxValue
6  Max_IH_efficiency← MaxValue
7  Sort VmList according to decreasing CPU utilization
8  foreach Vm in VmList do
9      foreach host in HostList do
10         if host Is Feasible (host,vm)then
11             if host is inactive then
12                 Compute IHL(t)
13                 if IHL_i ( t)<Max_IH_efficiency then
14                     Max_IH_efficiency← IHL_i (t)
15                     FIH← host
16             ;
17             if host is active then
18                 Compute AHL_i (t)
19                 if AHL_i (t)<Max_AH_efficiency then
20                     Max_AH_efficiency← AHL_i (t)
21                     FAH ← host
22             if FAH≠ null then
23                 allocatedHost ← FAH
24             else
25                 allocatedHost ← FIH
26     add (allocatedHost, Vm) to VmPlacement
```

IV. EXPERIMENTATION AND RESULTS

Differences between simulations and real experimental results in cloud data centers stem from several key factors. Simulations using CloudSim offer a controlled environment with idealized hardware variability, leading to consistent results, while real data centers have diverse hardware with varying performance, affected by wear, failures, and inconsistencies, potentially reducing observed efficiency gains. Network conditions in simulations may not capture real-world complexities like congestion and packet loss, impacting VM migration and efficiency in actual environments.

Additionally, simulations use predefined workloads that lack the dynamic variability of real user behavior, affecting consolidation and migration patterns. Power consumption models in simulators, based on theoretical data, may not accurately reflect real power usage influenced by hardware efficiency, cooling needs, and environmental conditions, leading to discrepancies. Furthermore, environmental factors such as temperature and humidity, often simplified in simulations, are crucial in real data centers for maintaining hardware efficiency and preventing thermal throttling. Recognizing these differences is essential for translating simulation benefits into real-world applications. To enhance the quality of service and energy responsiveness we have to perform the experiments in an environment that can easily support IaaS. This setup should consist of many physical servers supporting multicore architecture, high performance, high bandwidth, and storage area network (SAN).

This setup should also include in-depth information about the power utilization of the servers. However, building such an environment is very hard for practical purposes. Researchers have developed a simulation tool CloudSim [46] to simulate large and small infrastructures. Instead of using the analytical power model, they have used the Realtime power model, which gives a more generic view of power consumption. These experiments were performed on the latest version of the CloudSim. CloudSim is a Java-based simulator for testing and simulation of cloud data centers, including IaaS. We have considered four state-of-the-art base-line algorithms PEBFD, MFPED, PABFD, and MBFD, and compared the efficiency of LA-PED to the baseline algorithms on ten days of real-time work-load traces of PlanetLab and Statical Data of experimental evaluation of PlantLab work-load (Mean-Values) of Materna trace-3.

A. PERFORMANCE MATRIX

Along with the power consumption, we explain some essential performance matrices for measuring SLA violations. We use the same performance matrices. SLA is the service level agreement between the consumer and service provider, and service providers are bound to provide services to consumers to meet the QoS criteria. According to authors in [46], there are two groups in which SLA matrices can be distinguished.

- The percentage of time in which an active host faces 100% utilization. this is referred to as an SLA violation per active host SLATAH.
- The overall performance degradation due to Vm migration which is referred to as PDM.

These matrices can be expressed as follows:

$$\left\{ \begin{aligned} SLATAH &= \frac{1}{m}\sum_{i=1}^{m}\frac{T_i^{mak}}{T_i} \\ PDM &= \frac{1}{n}\sum_{j=1}^{n}\frac{EPDM_j}{RCPU_j} \end{aligned} \right\} \tag{6}$$

While m is the total number of hosts and T_i^{mak} is the time in which a host hi experiences 100% utilization and Ti is the time in which host hi remains active. On the other hand, n is the total number of migrating VMs and EPDMj denotes the estimate of degradation of performance of Vmj due to migration and RCPUj is the total requested CPU capacity by Vmj during its lifetime. A general performance metric SLAv is defined by multiplying the SLATAH and PDM.

$$SLA_v = SLATAH \times PDM \tag{7}$$

B. EXPERIMENTS ON PLANETLAB WORKLOAD

For PlanetLab ten-day workload, there are four types of hosts with heterogeneous architecture. The total number of hosts is 800, with 200 hosts of each type. The information about servers is given in Table 4. The power consumption model of the servers used in this experiment is obtained from the SPEC. For this experiment, we set $\alpha = 0.6$ and $\beta = 0.5$. Moreover, to know the performance of the underlying algorithm, we have fixed the host overutilization detection policy as Local regression (LR) and VM selection policy as Minimum Utilization (MU) for these experiments.

TABLE 3. PlanetLab workload characteristics.

Date	Mean-load (%)	St.dev (%)	VMs
2023/03/03	12.31	17.09	1052
2023/03/06	11.44	16.83	898
2023/03/09	15.57	1061	10.70
2023/03/22	9.26	12.78	1516
2023/03/25	10.56	14.14	1078
2023/04/03	12.39	16.55	1463
2023/04/09	11.12	15.09	1358
2023/04/11	11.56	15.07	1233
2023/04/12	11.54	15.15	1054
2023/04/20	10.43	15.21	1033

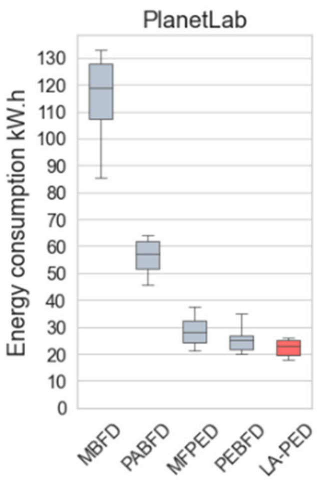
The workload characteristics for ten days of PlanetLab workload are given in Table 3. These characteristics include the mean and standard deviation of workload for each day, including No. of VMs for each day and the utilization for each VM recorded after five minutes. There are four types of VMs used in this experiment. Requirements of each VMs including frequency, core, and RAM is shown in Table 4.

TABLE 4. VM types.

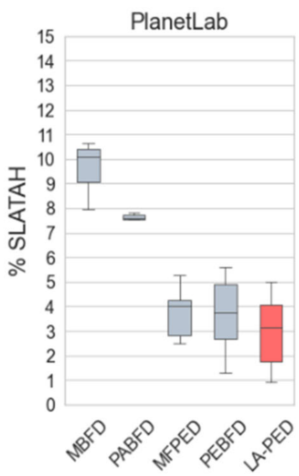
VM Type	Frequency (MHZ)	Core	RAM (GB)
Large Instance	2500	1	0.87
Medium-Instance	2000	1	1.74
Small-Instance	1000	1	1.74
Micro-Instance	500	1	0.61

In the PlanetLab workload, EEVMC outperformed traditional methods such as MBFD and PABFD. The results showed that EEVMC reduced energy consumption by 80.35% compared to MBFD, 59.76% compared to PABFD, and 21.59% compared to MFPED. Additionally, EEVMC achieved a reduction in SLA violations by 94.51% compared to MBFD, 94.85% compared to PABFD, and 47.17% compared to MFPED.

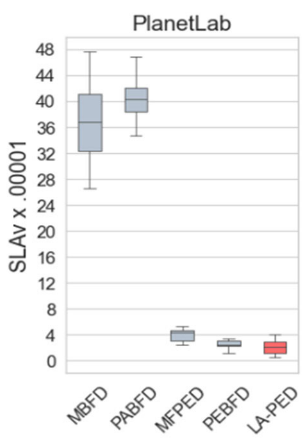
It can be seen in Fig. 1a that the mean-energy consumption of LA-PED for ten days workload of PlanetLab is less than other baseline algorithms. For this experiment, LA-PED performs relatively better on average 80.35%, 59.76%, 21.59%, and 7.40% in energy efficiency compared to MBFD, PABFD, MFPED, and PEBFD, respectively. For SLATAH LA-PED performs relatively better on average 70.77%, 61.57%, 20.16%, 25.84% and in SLAv reduction LA-PED performs relatively better on average 94.83%, 94.85%, 47.20%, 17.78% for MBFD, PABFD, MFPED, PEBFD respectively (See Fig. 1).



(a) Energy Consumption kW.h.



(b) % SLATAH.



(c) SLAv x 0.00001.

FIGURE 1. (a) Energy Consumption, (b) %SLATAH, (c) SLAv for ten days workload of Planet Lab.

The performance efficiency of LA-PED is obvious in the baseline algorithms. It also satisfies the definition of the efficient algorithm presented earlier. If we observe the results, we can see LA-PED improves 7.40% in energy efficiency compared to PEBFD, but it improves 25.84% in reducing %SLATAH and 17.78% in reducing SLAv. Although the relative improvement of LA-PED compared to MFPED is 20.16% for SLATAH and 21.59% for energy efficiency, it improves 47.20% for SLAv, which is a considerable amount. If we analyze LAPED compared to MBFD and MFPED, the improvement in energy efficiency, SLATAH, and SLAv is significantly large, as discussed earlier.

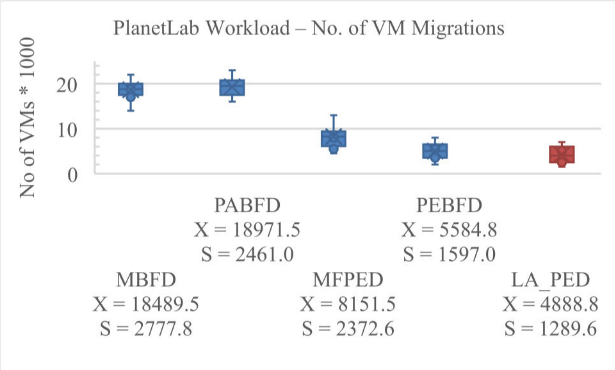


FIGURE 2. Vm migrations of LA-PED as compared to baseline algorithms.

For further evaluation of LA-PED, we have computed the total number of VM migrations which can be seen in Fig. 2. We can see that LA-PED significantly reduces the number of VM migrations as compared to the baseline algorithms.

This improvement is 73.77%, 74.23%,40.02%, and 12.46% for MBFD, PABFD, MFPED, and PEBFD respectively. We have computed the Mean and Standard deviation of the time before a host shutdown for each day, and the performance of LA-PED can be seen in Fig. 3 where the performance of LA-PED is relatively better than the baseline algorithms as discussed in Table 6.

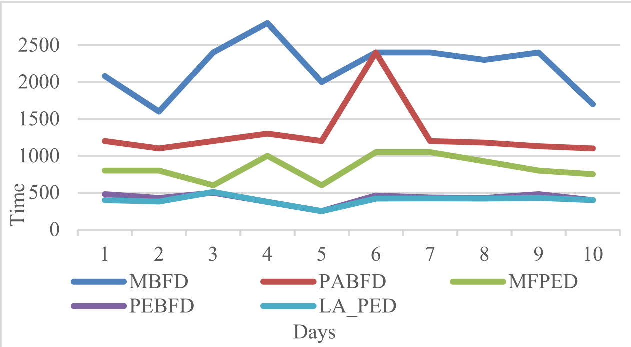
C. EXPERIMENTS ON MATERNA WORKLOAD

Experiments on a unique workload can give a pessimistic overview of any algorithm. Therefore, we have performed the experiments on fast Materna trace-3. The characteristics of Materna's workload for ten days are given in Table 7. The server configuration is given in Table 5 and the VM types are given in Table 4. We have kept the host overutilization detection policy as Local regression (LR), and the VM selection policy as Minimum Utilization (MU) for this experiment also to exactly determine the improvement due to Vm Placement only. Further, we have kept the same values of $\alpha = 0.6$ and $\beta = 0.5$. Statistical data of the experimental evaluation of Materna workload is given in Table 8.

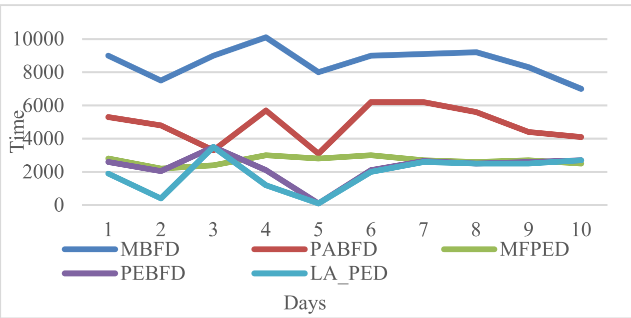
It can be seen that the mean energy consumption of LA-PED for the ten-day workload of Materna is less than other baseline algorithms. For this experiment, LA-PED performs relatively better on average 16.403%, 61.009%, 4.94%, and 4.827% in energy efficiency compared to MBFD, PABFD, MFPED, and PEBFD, respectively. For %SLATAH LA-PED performs relatively better on average 66.175%, 68.806%, 13.538%, 39.839%, and in SLAv LA-PED performs relatively better on average 76.995%, 88.881%, 12.500%, 48.65% for MBFD, PABFD, MFPED, PEBFD respectively.

TABLE 5. Configuration of servers for planetlab and materna workload.

	CPU Model	Cores	Frequency (MHZ)	RAM (GB)	PlanetLab Host Count	Materna Host Count
HP Proliant Ml110g4	Intel Xeon3040	2	1860	4	200	600
HP Proliant Ml110g5	Intel Xeon3075	2	2660	4	200	600
IBM System X3250	Intel XeonX3470	4	2933	8	200	-
Dell Power edge R520	Intel XonE52470	16	2300	24	200	-



(a) Mean-time Before Host Shutdown.



(b) Standard deviation Before a host Shutdowns.

FIGURE 3. (a) Mean and (b) standard deviations of time before a host shutdown on daily basis for PlanetLab.

TABLE 6. Statical data of experimental evaluation of plantlab workload (Mean Values).

Algorithm	Energy (kW.h)	% SLATAH	SLAv ×0.00001	Vm Migrations	PDM
MBFD	118.485	10.095	40.280	18489.5	0.037
PABFD	57.869	7.677	40.46	18971.5	0.053
MFPED	29.695	3.695	3.940	8151.5	0.01
PEBFD	25.142	3.978	2.530	5584.8	0.009
LA-PED	23.281	2.95	2.080	4888.8	0.009

For Materna's workload, the performance efficiency of LA-PED is evident. We can see that the improvement in energy efficiency is not too much compared to MFPED and PEBFD, but the improvement in reducing SLATAH is 13.53%, and 29.83% for MFPED, and PEBFD, respectively. In contrast, it is 66.17%, and 68.80% for MBFD and PABFD respectively. It is a significant amount of improvement. If we compare the results for SLAv, we can see the minimum

TABLE 7. PlanetLab workload characteristics.

Mean load (%)	Date	St. Dev (%)	VMs
2023/01/04	4.87	2.004	547
2023/01/05	4.70	1.876	547
2023/01/06	4.54	1.701	546
2023/01/07	4.97	1.987	544
2023/01/08	5.01	1.982	544
2023/01/09	4.90	2.039	544
2023/01/10	4.75	1.643	542
2023/01/11	4.83	1.859	539
2023/01/12	4.90	1.879	539
2023/01/13	4.77	2.040	536

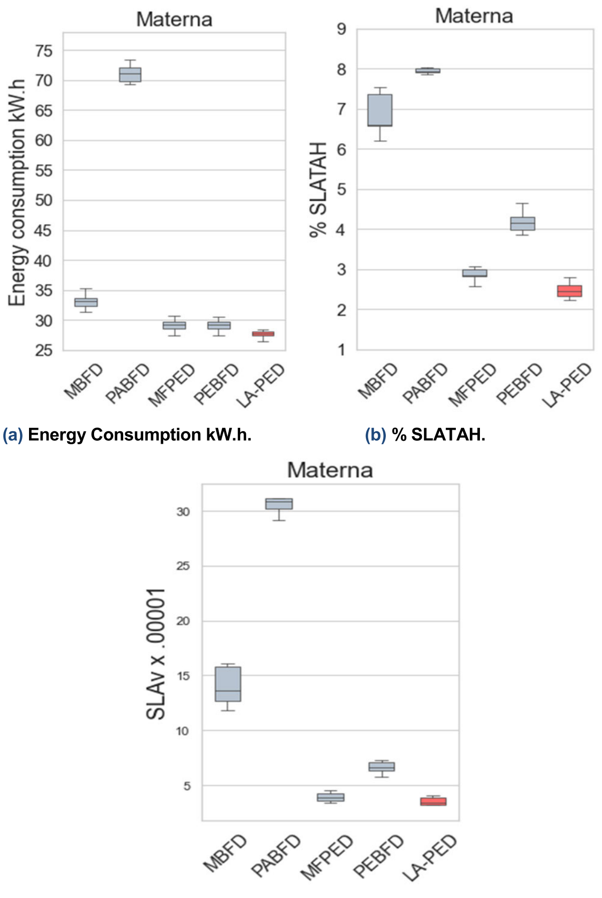
improvement is 12.50% and the maximum improvement is 88.88% for MFPED and PABFD, respectively. For Maternal workload, our technique also performs better in reducing the total number of VM migrations (See Fig.5). The average VM migrations are 32.76%, 56.472%, 4.47%, 21.58% for MBFD, PABFD, MFPED, PEBFD respectively (See Fig.4).

TABLE 8. Statical data of experimental evaluation of materna workload (Mean-Values).

Algorithm	Energy (kW.h)	%SLATAH	SLAv×0 .00001	VM Migrations	PDM
MBFD	33.182	7.326	14.910	10060.6	0.02
PABFD	71.143	7.944	30.85	15539.2	0.04
MFPED	29.183	2.866	3.920	7080.4	0.011
PEBFD	29.146	4.119	6.680	8625.2	0.019
LA-PED	27.739	2.478	3.430	6763.8	0.01

We have performed experiments on two different workload characteristics. The performance of LA-PED is evident for both workloads. We can see that MBFD performs better than PABFD in energy efficiency, %SLATAH, and SLAv, especially in Materna. In Contrast, MFPED performs better energy efficiency, %SLATAH, and SLAv than MBFD, PABFD, and PEBFD. Our presented approach outperforms the baseline techniques in both workloads. We have also observed the efficiency by reducing the hosts' heterogeneity, increasing the host count, and varying the workload. LA-PED still outperforms the baseline algorithms, which is a considerable achievement.

The authors should provide a deeper analysis of the evaluation results, explaining the reasons behind the observed performance improvements. Additionally, they should discuss the feasibility and potential challenges of applying the proposed method in real-world cloud data centers.



(a) Energy Consumption kW.h.

(b) % SLATAH.

(c) SLAv x 0.00001

FIGURE 4. (a) Energy Consumption, (b) %SLATAH, (c) SLAv for ten days workload of Materna.

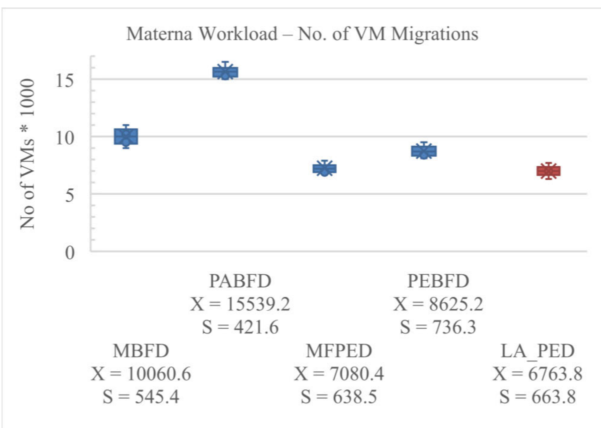


FIGURE 5. Vm migrations of LA-PED as compared to baseline algorithms.

V. CONCLUSION

We introduced an energy-efficient VM placement technique within the OpenStack Neat VM consolidation framework. Our approach categorizes hosts as either active or inactive and evaluates the loss incurred during VM placement. Active hosts with minimal loss are prioritized, and if VM placement leads to the overutilization of an active host, an inactive host with minimal post-placement loss is selected. Experiments using ten-day workloads from PlanetLab and Materna demonstrate that our technique not only enhances energy efficiency but also improves Quality of Service (QoS), making it highly suitable for cloud computing. For PlanetLab's ten-day workload, our technique achieved energy efficiency improvements ranging from 7.40% to 80.35% compared to PEBFD and MBFD, respectively. Similarly, improvements in service level agreement violations (SLAv) ranged from 17.78% to 94.83% relative to these benchmarks. Additionally, our approach significantly reduced the total number of VM migrations and the mean and standard deviation before a host shutdown. Testing with the Materna ten-day workload yielded energy efficiency improvements between 4.827% and 61.009% compared to PABFD and PEBFD, respectively, and SLAv improvements between 12.50% and 76.99% compared to MFPED and MBFD. Notably, while baseline algorithms performed well with the PlanetLab workload, they underperformed with Materna, whereas our technique effectively managed both. In the future, we plan to dynamically adjust loss criteria at runtime using reinforcement learning to further improve QoS. The future work section could be enhanced by detailing implementation plans, addressing potential challenges, and highlighting the benefits of using reinforcement learning for dynamic loss criteria adjustment in VM consolidation.

ACKNOWLEDGMENT

The authors extend their appreciation to Taif University, Saudia Arabia, for supporting this work through project number (TU-DSPP-2024-52).

REFERENCES

[1] R. Karthikeyan, U. Srinivasarao, S. Hariharan, P. K. Sarangi, and B. S. Panigrahi, "Improving data centre power efficiency by applying the VM consolidation approach," in *Proc. Int. Conf. Adv. Commun. Technol.*, 2023, pp. 79–91.

[2] D. Dabhi and D. Thakor, "Utilisation-aware VM placement policy for workload consolidation in cloud data centres," *Int. J. Commun. Netw. Distrib. Syst.*, vol. 28, no. 6, p. 704, 2022, doi: 10.1504/IJCNDS.2022.126224.

[3] B. Magotra, D. Malhotra, and A. K. Dogra, "Adaptive computational solutions to energy efficiency in cloud computing environment using VM consolidation," *Arch. Comput. Methods Eng.*, vol. 30, no. 3, pp. 1789–1818, Apr. 2023, doi: 10.1007/S11831-022-09852-2.

[4] (2021). *Usage Impact on Data Center Electricity Needs: A System Dynamic Forecasting Model*. Accessed: Nov. 19, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261921003019

[5] R. Medara, *Dynamic Virtual Machine Consolidation in a Cloud Data Center Using Modified Water Wave Optimization*. Cham, Switzerland: Springer, 2023.

[6] A. Roytman, A. Kansal, S. Govindan, J. Liu, and S. Nath, "PACMan: Performance aware virtual machine consolidation," in *Proc. 10th Int. Conf. Auton. Comput. (ICAC)*, 2013, pp. 83–94. Accessed: Nov. 19, 2023. [Online]. Available: https://www.usenix.org/conference/icac13/technical-sessions/presentation/roytman

[7] B. Magotra and D. Malhotra, *Adaptive Computational Solutions to Energy Efficiency in Cloud Computing Environment Using VM Consolidation*. Cham, Switzerland: Springer, 2023.

[8] X. Li, Y. Li, C. Shen, A. Dick, and A. V. D. Hengel, "Contextual hypergraph modeling for salient object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3328–3335, doi: 10.1109/ICCV.2013.413.

[9] W. Yao, Z. Wang, Y. Hou, X. Zhu, X. Li, and Y. Xia, "An energy-efficient load balance strategy based on virtual machine consolidation in cloud environment," *Future Gener. Comput. Syst.*, vol. 146, pp. 222–233, Sep. 2023. Accessed: Nov. 20, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X23001498

[10] D. Dabhi and D. Thakor, ''Utilisation aware virtual machine selection policy for workload consolidation in cloud data centres,'' *Int. J. Commun. Netw. Distrib. Syst.*, vol. 29, no. 2, p. 125, 2023, doi: 10.1504/IJCNDS.2023.129226.
[11] U. Arshad, M. Aleem, G. Srivastava, and J. C.-W. Lin, ''Utilizing power consumption and SLA violations using dynamic VM consolidation in cloud data centers,'' *Renew. Sustain. Energy Rev.*, vol. 167, Oct. 2022, Art. no. 112782.
[12] R. Zolfaghari and A. M. Rahmani, ''Virtual machine consolidation in cloud computing systems: Challenges and future trends,'' *Wireless Pers. Commun.*, vol. 115, no. 3, pp. 2289–2326, Dec. 2020, doi: 10.1007/S11277-020-07682-8.
[13] A. Beloglazov, R. Buyya, and Y. Lee. (2011). *A Taxonomy and Survey of Energy-efficient Data Centers and Cloud Computing Systems*. Accessed: Mar. 1, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780123855121000037
[14] J. Cao, Z. Ma, J. Xie, X. Zhu, F. Dong, and B. Liu, ''Towards tenant demand-aware bandwidth allocation strategy in cloud datacenter,'' *Future Gener. Comput. Syst.*, vol. 105, pp. 904–915, Apr. 2020. Accessed: Nov. 19, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X17311834
[15] D. M. Vistro, A. Ur Rehman, S. Mehmood, M. Idrees, and A. Munawar, ''A literature review on security issues in cloud computing: Opportunities and challenges,'' 2020. Accessed: Nov. 20, 2023. [Online]. Available: https://www.jcreview.com/paper.php?slug=a-literature-review-on-security-issues-in-cloud-computing-opportunities-and-challenges
[16] M. Sayadnavard. (2022). *A Multi-objective Approach for Energy-efficient and Reliable Dynamic VM Consolidation in Cloud Data Centers*. Accessed: Nov. 20, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S221509862100104X
[17] M. Malekloo and N. Kara, *An Energy Efficient and SLA Compliant Approach for Resource Allocation and Consolidation in Cloud Computing Environments*. Amsterdam, The Netherlands: Elsevier, 2018.
[18] M. Alicherry, ''Network aware resource allocation in distributed clouds,'' in *Proc. IEEE INFOCOM*, Jul. 2012, pp. 1–20.
[19] M. Khelghatdoust, V. Gramoli, and D. Sun, ''GLAP: Distributed dynamic workload consolidation through gossip-based learning,'' in *Proc. IEEE Int. Conf. Cluster Comput. (CLUSTER)*, Sep. 2016, pp. 80–89, doi: 10.1109/CLUSTER.2016.24.
[20] P. Bodak, I. Menache, M. Chowdhury, P. M. Microsoft, and D. A. M. Microsoft. (2012). *Surviving Failures in Bandwidth-Constrained Datacenters*. Accessed: Nov. 19, 2023. [Online]. Available: https://www.microsoft.com/en-us/research/publication/surviving-failures-in-bandwidth-constrained-datacenters/
[21] N. El-Sayed, I. A. Stefanovici, G. Amvrosiadis, A. A. Hwang, and B. Schroeder, ''Temperature management in data centers: Why some (might) like it hot,'' *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 40, no. 1, pp. 163–174, Jun. 2012, doi: 10.1145/2318857.2254778.
[22] K. Halder, U. Bellur, and P. Kulkarni, ''Risk aware provisioning and resource aggregation based consolidation of virtual machines,'' in *Proc. IEEE 5th Int. Conf. Cloud Comput.*, Jun. 2012, pp. 598–605. Accessed: Nov. 19, 2023.
[23] S. Mazumdar, *Power Efficient Server Consolidation for Cloud Data Center*. Amsterdam, The Netherlands: Elsevier, 2017.
[24] K. Dinesh Kumar and E. Umamaheswari, ''An efficient proactive VM consolidation technique with improved LSTM network in a cloud environment,'' *Computing*, vol. 106, no. 1, pp. 1–28, Jan. 2024, doi: 10.1007/S00607-023-01214-5.
[25] (2020). *A Four Green TM/Red TE Demultiplexer Based on Multi Slot-waveguide Structures*. Accessed: Jun. 5, 2024. [Online]. Available: https://www.mdpi.com/1996-1944/13/14/3219
[26] J. Menahem. (2022). *14 Wavelength Demultiplexer C-band Using Cascaded Multimode Interference on SiN Buried Waveguide Structure*. Accessed: Jun. 5, 2024. [Online]. Available: https://www.mdpi.com/1996-1944/15/14/5067
[27] F. F. Moges and S. L. Abebe, ''Energy-aware VM placement algorithms for the OpenStack neat consolidation framework,'' *J. Cloud Comput.*, vol. 8, no. 1, pp. 1–23, Dec. 2019, doi: 10.1186/S13677-019-0126-Y.
[28] J. Witanto. (2018). *Adaptive Selection of Dynamic VM Consolidation Algorithm Using Neural Network for Cloud Resource Management*. Accessed: Dec. 08, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X17323336
[29] M. Rezaei-Mayahi, *Temperature-Aware Power Consumption Modeling in Hyperscale Cloud Data Centers*. Amsterdam, The Netherlands: Elsevier, 2019.
[30] A. Ullah, H. Elahi, Z. Sun, A. Khatoon, and I. Khatoon, ''Comparative analysis of AlexNet, ResNet18 and SqueezeNet with diverse modification and arduous implementation,'' *Arabian Journal Science Engineering*, vol. 47, no. 2, pp. 2397–2417, 2022.
[31] S. M. Rozehkhani, F. Mahan, and W. Pedrycz, ''Efficient cloud data center: An adaptive framework for dynamic virtual machine consolidation,'' *J. Netw. Comput. Appl.*, vol. 226, Jun. 2024, Art. no. 103885.
[32] F. Xu, F. Liu, and H. Jin, ''Heterogeneity and interference-aware virtual machine provisioning for predictable performance in the cloud,'' *IEEE Trans. Comput.*, vol. 65, no. 8, pp. 2470–2483, Aug. 2016.
[33] W. Deng, F. Liu, H. Jin, X. Liao, and H. Liu, ''Reliability-aware server consolidation for balancing energy-lifetime tradeoff in virtualized cloud datacenters,'' *Int. J. Commun. Syst.*, vol. 27, no. 4, pp. 623–642, Apr. 2014.
[34] A. Hussain, A. Ullah, A. Aslam, and A. Khatoon, ''A modified Siamese network for facial assimilation,'' *WSEAS Trans. Signal Process.*, vol. 16, pp. 60–66, 2023.
[35] Z. Li, X. Yu, L. Yu, S. Guo, and V. Chang, ''Energy-efficient and quality-aware VM consolidation method,'' *Future Gener. Comput. Syst.*, vol. 102, pp. 789–809, Jan. 2020.
[36] A. Ullah, S. S. Rizvi, A. Khatoon, and S. Jin Kwon, ''The empirical analysis, mathematical modeling, and advanced control strategies for buck converter,'' *IEEE Access*, vol. 12, pp. 19924–19941, 2024.
[37] M. Alam, S. Mustajab, M. Shahid, and F. Ahmad, ''Cloud computing: Architecture, vision, challenges, opportunities, and emerging trends,'' in *Proc. Int. Conf. Comput., Commun., Intell. Syst. (ICCCIS)*, Nov. 2023, pp. 1–26.
[38] M. Alam, M. Mahak, R. A. Haidri, and D. K. Yadav, ''Efficient task scheduling on virtual machine in cloud computing environment,'' *Int. J. Pervasive Comput. Commun.*, vol. 17, no. 3, pp. 271–287, Jun. 2020.
[39] H. Ali, S. Abidin, and M. Alam, ''Auditing of outsourced data in cloud computing: An overview,'' in *Proc. 11th Int. Conf. Comput. Sustain. Global Develop. (INDIACom)*, Feb. 2024, pp. 1–6.
[40] F. Smarandache and V. Christianto. (2007). *Hadron Models and Related New Energy Issues*. [Online]. Available: https://fs.unm.edu/NewEnergy.pdf
[41] Y. Saadi and S. El Kafhali, ''Energy-efficient strategy for virtual machine consolidation in cloud environment,'' *Soft Comput.*, vol. 24, no. 19, pp. 14845–14859, Oct. 2020, doi: 10.1007/S00500-020-04839-2.
[42] F. Xu, F. Liu, H. Jin, and A. V. Vasilakos, ''Managing performance overhead of virtual machines in cloud computing: A survey, state of the art, and future directions,'' *Proc. IEEE*, vol. 102, no. 1, pp. 11–31, Jan. 2014.
[43] S. Garg and A. Toosi, *SLA-Based Virtual Machine Management for Heterogeneous Workloads in a Cloud Datacenter*. Amsterdam, The Netherlands: Elsevier, 2014.
[44] M. Kumar, *Heterogeneity and Thermal Aware Adaptive Heuristics for Energy Efficient Consolidation of Virtual Machines in Infrastructure Clouds*. Amsterdam, The Netherlands: Elsevier, 2016.
[45] A. Ur Rehman, S. Mahmood, G. Mustafa, and S. Mehta, ''A comparative analysis on handling big data using cloud services,'' *Lahore Garrison Univ. Res. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 4, pp. 78–86, Dec. 2021.
[46] F. Smarandache and V. Christianto, ''Observation of anomalous potential electric energy in distilled water under solar heating,'' in *Proc. Bulletin American Physical Society, Joint Spring Meeting*, 2011, pp. 1–23.
[47] A. Tchana, N. De Palma, I. Safieddine, and D. Hagimont, ''Software consolidation as an efficient energy and cost saving solution,'' *Future Gener. Comput. Syst.*, vol. 58, pp. 1–12, May 2016, doi: 10.1016/J.FUTURE.2015.11.027.

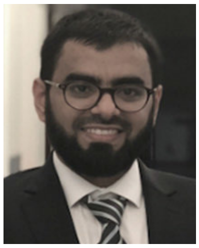
ATTIQUE UR REHMAN received the M.S. degree in CS from Bahauddin Zakariya University, Multan, in 2018. He is currently pursuing the Ph.D. degree in computer science with Huazhong University of Science and Technology, Wuhan, Hubei, China. He is a Senior Lecturer with the Department of Computer Science, Lahore Garrison University, Lahore, Pakistan. His research interests include networks, information security, cloud computing, and the Internet of Things (IoT).

SONGFENG LU received the Ph.D. degree from Huazhong University of Science and Technology, Wuhan, China, in 2001. He is currently a Professor with the School of Cyber Science and Engineering, Huazhong University of Science and Technology. He has authored more than 200 papers in international conference proceedings, and journals and holds more than 50 patents. His current research interests include information security, quantum computing, and artificial intelligence.

MUBASHIR ALI received the M.C.S. and M.S.C.S. degrees in computer science from Bahauddin Zakariya University (BZU), Multan, Pakistan, in 2016 and 2018, respectively. He is currently a Senior Lecturer with the Department of Software Engineering, Lahore Garrison University (LGU), Pakistan. He has published more than 25 international journals and conference papers in IEEE and Elsevier. His research interests include the Internet of Things, wearable healthcare, edge-loud, and predictive modeling.

FLORENTIN SMARANDACHE received the M.Sc. degree in mathematics and computer science from the University of Craiova, Romania, and the Ph.D. degree in mathematics from the State University of Kishinev. He completed the postdoctoral research in applied mathematics from Okayama University of Sciences, Japan, and The Guangdong University of Technology, Guangzhou, China. He is currently an Emeritus Professor of mathematics with The University of New Mexico, United States. He has been a Founder of neutrosophy (generalization of dialectics), neutrosophic set, logic, probability, and statistics, since 1995. He has published more than 100 articles and books with reputable publishers.

SULTAN S. ALSHAMRANI received the bachelor's degree (Hons.) in computer science from Taif University, in 2007, the master's degree in information technology (computer networks) from The University of Sydney, Sydney, Australia, and the Ph.D. degree from The University of Liverpool, U.K. He is currently a Professor with Taif University, Saudi Arabia, and the Head of the Department of Information Technology. He was a General Grade of ''Excellent'' with first honor and an accumulative GPA of (4,85) out of (5,00) where considered the highest GPA in the college for the bachelor's degree. He is the Chairperson of the Committee of Faculty Members Affairs.



ABDULLAH ALSHEHRI received the B.S. degree in computer science from Taif University, Saudi Arabia, the M.S. degree in software engineering from Wollongong University, Australia, and the Ph.D. degree in computer science from The University of Liverpool, U.K., in 2018, majored in data mining and machine learning. He is currently an Assistant Professor with the Faculty of Computer Science and Information Technology, Al-Baha University, Saudi Arabia. He has published a number of refereed publications on AI-related research, and he has chaired several sessions at international conferences in the AI field. His research interests include artificial intelligence (AI), machine learning, deep learning, time series data mining (TSDM), and applied AI in cybersecurity, biometrics, and authentication.

FARRUKH ARSLAN received the B.Sc. degree in electrical engineering from the University of Engineering and Technology (UET), Lahore, Pakistan, and the M.Sc. and Ph.D. degrees from Purdue University, USA. Currently, he is an Assistant Professor with UET. He is experienced with working in information technology and data science in academia. His research interests include networks, machine learning, data mining, and data science.