K-Means Clustering Based VM Placement Using MAD and IQR

Akanksha Tandon[1], Aditya Jena[1], Sanjeev Patel[1*]

[1]Department of Computer Science and Engineering, National Institute of Technology Rourkela , Rourkela, 769008, Odisha, India.

*Corresponding author(s). E-mail(s): patels@nitrkl.ac.in; Contributing authors: akankshatandon97@gmail.com; kingapj@gmail.com;

Abstract

Cloud Data Centers (CDCs) have been developed into a virtual computing platform for businesses. Nevertheless, CDCs require significant power, which is essential for processor speed, particularly for Internet of Things (IoT) activities. Despite the existence of a significant amount of research in the green allocation of resource methodologies, it has been carried out to minimize the usage of the CDCs. Traditional systems mainly seek to minimize the number of physical machines (PMs) and rarely tackle the problems of overload and energy efficiency of the virtual machines (VMs) regulations concurrently. Furthermore, present systems cannot evaluate and redirect traffic from relevant sources to maximize the quality of service (QoS) supplied by CDCs. To improve energy saving, we attempt to enhance the adaptive four thresholds energy-aware framework for VM deployment energy efficiency (AFED-EF) scheme to improve energy savings. That is a unique adaptive energy-aware VM allocation and deployment technique for different applications to address these issues. We conducted a comprehensive exploratory program utilizing an authentic workload of over a million Planet Lab VMs. The research results demonstrate that our modified approach outperforms the AFED-EF and other existing traditional approaches, such as median absolute deviation (MAD), interquartile range (IQR), and overload detection using exponentially weighted moving average.

Keywords: SLA violation, Energy Consumption, Cloud data centers, Virtual Machines, Energy Efficiency, virtual machine allocation (VMA), and Median absolute deviation.

1 Introduction

Cloud computing solutions allow the client to access a centralized repository with programmable resources anytime. Generally, cloud computing systems are classified into infrastructure as a Service (IaaS), Software as a Service (SaaS), and Platform as a Service (PaaS)[1]. PaaS primarily offers cloud consumers infrastructure enabling development and implementation, leaving individuals to install and manage the necessary Linux kernel and applications. In case of IaaS, each licensed system contains a Linux kernel and the essential frameworks. However, SaaS delivers technology upon request, implying that customers do not need to pay for the product's unique licensing. Through the very last couple of short decades, Kaur et al.[2] have proposed that cloud technology has already become increasingly popular as a reduced functionality information technology, enabling local firms to sublet decentralized configurable assets (super-computing and connectivity) made available as a large-scale service model without making capital stakes in maintenance and management. Among all cloud platform types, IaaS is now the most extensively used. Throughout this prototype, a network operator could indeed rent agreement virtual computing technical guidance assets (computation, flash memory, internet connectivity, and packet forwarding assistance) through one or even more virtual servers, cloud service providers (CPs) wrapped into intertwined VMs and constructed as just a network virtualization recommendation, to construct nonhomogenous Virtualization, offering personalized customer application forms to its end customers.

This paper deals with how VM allocation policies can be changed to reduce energy consumption in a data center while maintaining a reasonable QoS for its applications. This study aims to investigate the relationship between VM allocation policies, energy consumption, and QoS and identify optimal VM allocation policies that reduce energy consumption while maintaining a reasonable QoS. The study will use simulation-based approaches to evaluate the impact of VM allocation policies on energy consumption and QoS in a data center. The results of this study will provide guidelines for data center operators to make appropriate decisions regarding VM allocation policies, resulting in a more sustainable and efficient data center. This paper proposes a VM allocation method and a VM selection policy method to satisfy the above-mentioned objectives. VM consolidation consists of the following four sub-processes: VM Placement, host overloading monitoring under-load discovery, VM Selection for migrating from the overloaded server identification, and VM placement for supplying chosen VMs on such a fresh set of servers. This work precises four VM consolidation sub-components: VM placement, host overloading, VM selection, and migration. However, a few writers have concentrated exclusively on VM consolidation in their studies. Powering data center resources may be accomplished in two phases. The first stage is to position VMs efficiently, and step two is to optimize the resources allocated to them in the initial phase by employing VMs migration as resource consumption increases.

We have tried to implement K-Means-MAD-IQR and medium fit power efficient algorithm to determine the threshold for over-utilization and choice of host for migration, which are paramount in deciding energy consumption.

1.1 Contribution

We have followed the paper presented by author Zhou et al. [3]. In which authors have determined four thresholds based on the K-Means-MAD-IQR. Though the authors have developed a new VM deployment mechanism based on the four thresholds determined, our contribution is to study these two approaches for determining the threshold values and VM placement mechanism. We have observed that the identification of new threshold values will play a significant role during the VM placement. It improves the performance of adaptive four threshold energy aware for deploying VM allocation (AFED-EF). We have modified the VM placement scheme by introducing new energy efficiency parameters. We have implemented K-Means-MAD-IQR and Medium fit power efficient algorithms to determine the threshold for over-utilization and choice of host for migration.

This paper has been presented in the following order. Section 2 presents related work. Section 3 highlights the system model of the complete VM placement structure, and the results are presented in section 4. In the last section, we have concluded and stated the future direction of the result.

2 Related Work

The process of consolidating virtual machines, which entails three crucial steps: (i) identifying overloaded hosts, (ii) choosing which virtual machines to pick for consolidation, and (iii) deciding where to deploy these virtual machines, has been thoroughly studied by many academics. A thorough summary of related work in VM consolidation is given in Table 1. In this Table, Beloglazov et al. [4] and Xu et al. [5] have looked at the goals, methods, standards for measuring performance, and simulators used by different writers in this field. The Best Fit Decreasing (BFD) method for deploying virtual machines was introduced by Beloglazov et al. The four VM selection strategies they proposed were migration minimization (MM), static threshold (ST), high-performance growth (HPG), and random choice (RC). Researchers and engineers are working to increase the efficiency and advantages of cloud computing in response to its rising popularity. When cloud infrastructure is used efficiently and economically, cloud computing becomes viable, allowing organizations of all sizes to become stable. Cloud computing allows users to provide resources on demand and execute programs in a way that suits their demands by picking virtual resources that meet their application's resource requirements. It is therefore up to cloud resource providers to transform virtual resources into physical resources in order to accommodate these virtual resources on physical resources. Despite considering the providers' optimization aims and resource suppliers, the author discovers particular inherent challenges in cloud computing [6].

Kulshrestha et al. [12] have proposed that only one variable exponential weight moving average (EWMA) uses the mean of time-proportionally weighted information as its foundation. Hussain et al. [13] have proposed a technique to conserve resources using an audiovisual public cloud. Its primary goal is identifying appropriate hosts to close down to conserve energy. Li et al. [14] created a novel VMP approach called

Table 1: Comparative review of VM placement

Author	Objective	Methodology	Performance Evaluation Metrics	Simulator
Beloglazov et al. [4]	1) VMP 2) VM selection	1) Modified Best Fit Decreasing (BFD) 2) Utilization thr related dynamic policies - MM, ST, HPG, RC	a) Energy consumption b) SLAV c) VMM count	CloudSim
Wang et al. [7]	1). Host overload detection 2). Based on Host susceptibility	1). Based on Resource utilization correlation (RUC) for allocated VMs.	a). VM migration count b). Hot/cold spot count c). SLA violation d). Performance degradation	CloudSim
Moges et al. [8]	1) VMP	1) MBFD 2) PABFD 3) PEFFD 4) PEBFD 5) MFPED	a) Energy consumption b) overload time fraction c). VMM count	CloudSim
Kulkarni et al. [9]	1) VMP 2) VM Placement Optimization 3) Host Overload Detection 4) VM Selection	1) Power Aware Best Fit Decreasing (PABFD) 2) Global Workload Scheduler 3) THR	a) Energy Consumption b) Overall SLA Violation c) VM Migration d) Total Host (PM) Shutdown e) Mean VM Allocation Time Mean Host Selection policy	CloudSim
Fu et al. [10]	1) VM Placement 2) VM Selection	1) PABFD, minimum correlation coefficient (MCC) 2) Utilization threshold-based dynamic policies - MMT, MU, MC, RC	a) Energy Consumption b) Overall SLA Violation c) VM Migration	CloudSim
Melhem et al. [11]	1) VM Placement 2) Host overload detection.	1). Markov chain model	a) Energy consumption b) PlanetLab, c) Random	CloudSim

Modified Particle Swarm Optimization (MPSO) that depends on several resources. The MPSO method foregoes local optimization.

3 System Model

3.1 Working of K-Means-MAD-IQR

We have used two methodologies in this work by combining power usage, SLA breaches, and the host's energy efficiency and naming them K-Means-MAD-IQR Algorithm and Medium Fit Power Efficient Decreasing (MFPED). Our approach involves two parts: finding a threshold to determine the overutilization of the host and finding a suitable host for VMs that need to be migrated from an overutilized or an underutilized host. To determine the overutilization threshold, we use the K-Means-MAD-IQR Algorithm.

Algorithm 1 K-Mean-MAD-IQR [3]

Require: The past utilization set, $T = T_1, T_2, \ldots, T_n$.
Ensure: IQR value of Host Utilization.

pastCpuUtilization = host.getUtilizationHistory()
k = 7
clusters = KMeansClusters(pastCpuUtillization, k)
for $i = 1$ to k **do**
$MC[i] \leftarrow MAD(C[i])$ /* Using eq.(5) and(6)*/

end for
IQR value of Host Utilization

we calculate medium absolute deviation and IQR to determine the overutilization threshold. We use the K-Means-MAD-IQR Algorithm. The K-Means-MAD-IQR Algorithm takes in the utilization history of all the host VMs, T = T1, T2, . . . , Tn, and formed clusters using the K-Means clustering algorithm. Thereafter, the median absolute deviation (MAD) is calculated for each cluster, and the results are put in the form of an array, MC. Median absolute deviation of a cluster C as shown in Eq. (1).

$$MAD(C) = Median(|C_i - median(C)|) \tag{1}$$

after that, the array MC's interquartile range value, IQR, is found. Using this IQR value, the host is checked for overutilization. The IQR range of array MC is determined using Eq. (2):

$$IQR(MC) = Q_3(MC) - Q_1(MC) \tag{2}$$

The threshold for over-utilization P is determined as Eq. (3),

$$P = (1 - s * IQR), \tag{3}$$

where s is the safety parameter. Thereafter, we calculate the host utilization threshold, U_{Thr} using Eq (4),

$$U_{Thr} = \frac{\sum_{j=1}^{M} R_i}{Host_{MIPS}}, \tag{4}$$

where, R_j is the requested MIPS for j^{th} VMs. M is the total number of VMs. $Host_{MIPS}$ is the total MIPS the cost currently provides.

3.2 Working of Medium Fit Power Efficient Decreasing

VM placement problem can be transformed into a bin-packing problem. A bin-packing problem is an allocation problem where user requests will be mapped to available resources subject to minimize wastage. This process is known as the fitting approach. There are many versions of the fit model that are available in the literature [8]. Researchers have widely used bin packing techniques and their variants for VM placement problems. The interested one may refer [8] for detailed discussion.

The following is the medium-fit bin-packing rule: Letting L_h be a host's ideal resource utilization rate, which is provided as [8].

$$L_h = \frac{overload_{thr} + underload_{thr}}{2} \tag{5}$$

where L_h denotes the required levels of resource utilization in a host.
Moges et al. [8] proposed the Medium Fit Power Efficient Decreasing (MFPED) algorithm, incorporating a novel bin-packing heuristic called MFrule. The operational process of the MFPED algorithm is illustrated in Fig 1.

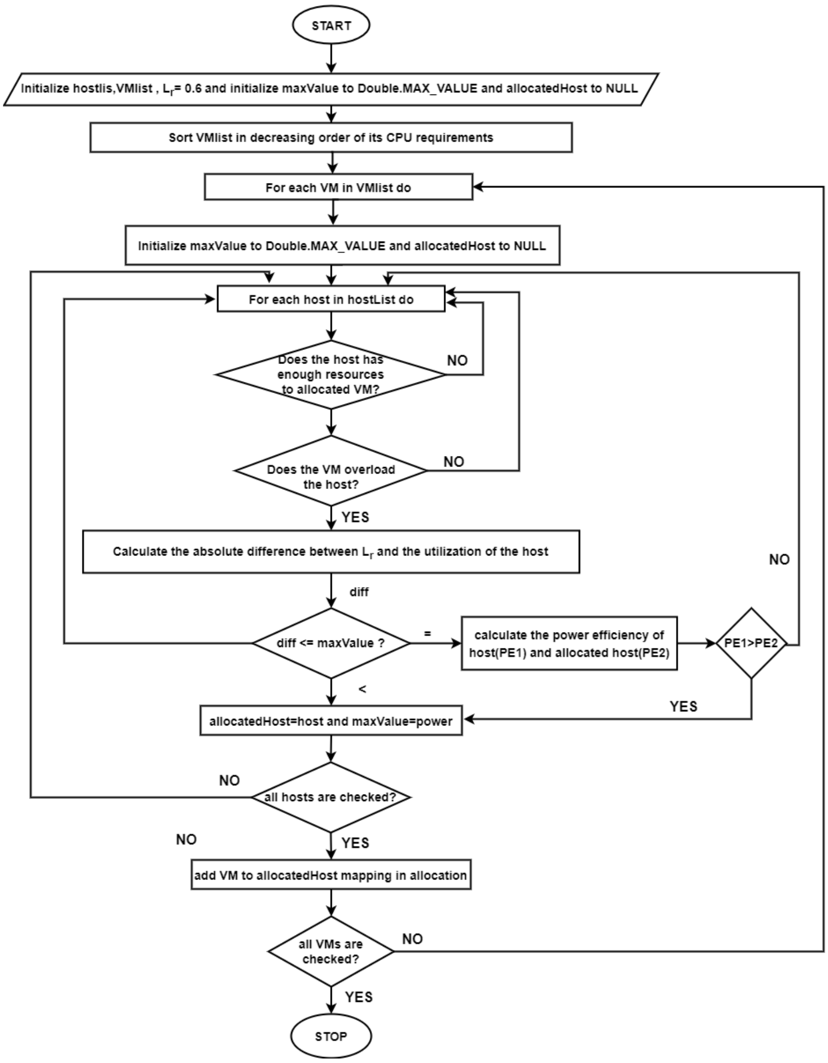

Fig. 1: Flowchart of medium fit power efficient decreasing (MFPED) [15]

In this approach, a physical machine is considered more suitable for VM placement than others if it exhibits a minimal difference in CPU utilization from the desired level, which is set at 0.6 in this paper. In cases where two physical machines have equivalent utilization levels, the selection is based on the superior power efficiency of the respective machines. Fig 2 represents the system model of the proposed algorithm.

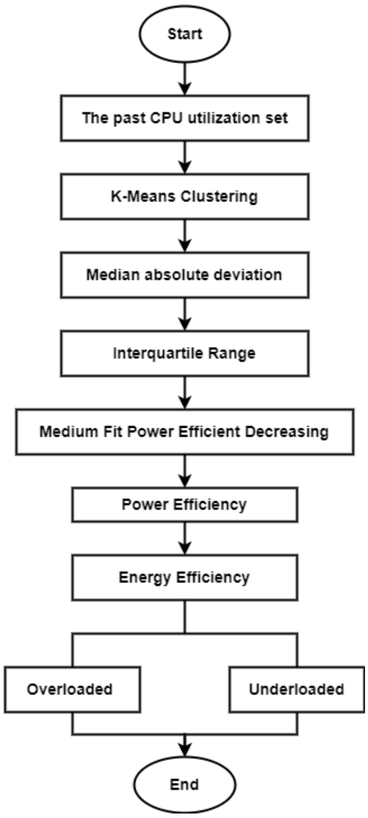

Fig. 2: Working flow of system model

4 Results and Discussions

In the following sections, we will carry out several exercises to assess the effectiveness of the AFED-EF algorithm. The suggested computation benefits are embodied in the MEEVMP [15], EWMA [12], THR-0.8 [16], MAD 2.5 [16], and IQR-1.5 [16]. These methods are chosen to compare in regards to the energy usage, SLA violation time per active host (SVTH), performance degradation due to migration (PDCM), and a variety of VM migrations. The MMT and MU are used to compare the VM selection policy. We examine the proposed algorithm findings for different measures during the initial set of experiments by modifying the values of criterion c. We adjust the parameter c between 0.5 to 3.0 with a difference of 0.5. The primary outcome of this investigation assists us in determining the ideal amount of parameter estimate c for the proposed algorithm. In the following research, we evaluate the effectiveness of the recommended method with an energy-efficient VM selection and without an energy-efficient VM selection.

4.1 Performance Metrics

This paper uses three performance metrics: Energy Efficiency, SLA violations, and several VM migrations.

4.1.1 Energy Consumption

We calculated energy consumption in order to calculate energy efficiency. The data center holds the power values for every host, using which we can calculate the total energy consumption. The total energy consumption (EC) is thus given by:

$$EC = \sum_{h=0}^{H_{count}} \sum_{t_1}^{t_2} E_h(v(t)) \tag{6}$$

where $E_h(v(t))$ is the power of a host at any given time interval, t and EC may be understood as energy consumed between time t_1 to t_2 (Energy(t_1, t_2)); h represents the host number in the range [0, H_{count}]; t_1 is the start time taken as 0; t_2 is end time taken as τ.

4.1.2 Service Level Agreement

The concept of SLA violation as defined by [3] indicates the situations in which a host is incapable of providing a specific VM with the requested number of instructions (MIPS) at a given moment. The measurement of SLA violation can be conducted either from the perspective of hosts or VMs, with both approaches yielding identical results. SLA violations are calculated as:

$$SLA = \text{PDCM} \ \times \text{Time per active host} \tag{7}$$

where PDCM denotes the performance degradation caused by VM migration.

4.1.3 Average Service Level Agreement

Average SLA violations are calculated as [12]:

$$\text{Average SLA violations} = \sum_{w=1}^{V_{count}} \left(\frac{\sum_{t=0}^{t=T} req(w,t) - alloc(w,t)}{V_{count}}\right) \tag{8}$$

Where req(w, t) is the MIPS requested by VM v at time t, and alloc(w, t) represents the MIPS allocated to the VM at that time.

4.1.4 VM Migration

The associated metrics to VM migration is calculated as:

$$VMM = \text{The number of VM migrations in the data center each day.} \quad (9)$$

4.2 Simulation Set-up

We have used the setup provided by Beloglazov et al. [16] described in their research, depicted in Table 2. In this, configuration are 800 physical computers with two distinct power models, and 1024 physical machines. Four distinct kinds of virtual servers. The workload samples were created randomly and derived using existing cloud records (PlanetLab). Research has proven that computer processors seriously influence server power consumption in data centers. Server processors, storage, disk, and throughput all impact how much power they use. We use the significant energy consumption data from the SPECpower benchmarking [17]. We choose two servers with two cores each. Table 3 shows CPU utilization.

Table 2: Parameters and configurations for simulation

Parameters	**Configuration**
Host type	HP ProLiantML110G4(2*1800MIPS)
Host type	HP ProLiantML110G5(2*2600MIPS)
Number of host	800,400 of each host type
VM type	2500MIPS, 2000MIPS, 1500MIPS, 1000MIPS
Workloads	PlanetLab(10 days of traces)

Table 3: CPU utilization

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

4.3 Workloads

We used a windows computer with an Intel i7-6700 3.4 GHz processor and 8 GB of memory for the tests, as the computer's high processing power and sufficient memory allowed for the smooth running of the experiments. The experiment group's composition of 10 instances of intake workload traces collected via PlanetLab on 10 separate days provided a diverse sample that accounted for any variations in workload over time. Gathering input workload information through 800 physiological nodes and 1,000 virtual machines is significant as it indicates the actual usage of PlanetLab cloud servers. This large sample size accurately represents the workload more accurately than a smaller sample size. Configuration of VMs is shown in Table 4.

Table 4: Configurations of VM

Virtual Machine	core	Capacity(MIPS)	RAM(GB)
High Instance	1	2500	0.85
Large Instance	1	2000	3.75
Small Instance	1	1000	1.7
Micro Instance	1	500	0.61

4.4 Results Analysis

We ran the analysis using datasets taken from PlanetLab, a real-world cloud. The PlanetLab is a powerful research platform supported by CoMon [18], a monitoring and controlling system that collects footprints. The data contains the CPU usage in percentage of over 100 virtual machines running on global hosts. These statistics were collected on various days in March and April of 2010. Table 5 and Table 6 show ten days of planate-lab workload. Table 7 shows the comparison of proposed algorithm power consumption, the average SLA breach, and the VMs migration. Results show that the proposed algorithm achieves the best results in terms of energy consumption, and the following techniques perform better in the sequence MEEVMP, EWMA, THR-0.8, MAD-2.5, and IQR-1.5. In terms of average SLA, EWMA outperforms the other compared techniques, while MFPED performs best in terms of VM migration. It has been observed that there is a tradeoff between various output performance measures such as energy consumption, ASLA, and VM migrations.

Table 5: QoS parameters of medium fit decreasing on various workloads using MMT

Date	Energy Con.(KWH)	SLA(%)	VM Mig
03/03/2011	100.70	0.01581	20447
06/03/2011	74.53	0.01426	15252
09/03/2011	84.41	0.02122	19317
22/03/2011	101.59	0.02034	25375
25/03/2011	88.69	0.01841	20286
03/04/2011	136.66	0.01662	26523
09/04/2011	106.83	0.01768	22979
11/04/2011	103.97	0.01812	21852
12/04/2011	90.57	0.01929	19809
20/04/2011	73.21	0.02851	19821

Table 6: QoS parameters of medium fit decreasing on various workloads using MU

Date	Energy Con.(KWH)	SLA(%)	VM Mig
03/03/2011	99.21	0.03392	37117
06/03/2011	73.17	0.03398	30302
09/03/2011	82.67	0.04902	37963
22/03/2011	100.03	0.04837	48686
25/03/2011	87.07	0.03995	37510
03/03/2011	134.8	0.04434	53721
09/03/2011	105.1	0.04165	43732
11/04/2011	102.42	0.04303	43515
12/04/2011	90.57	0.01929	19809
20/04/2011	89.02	0.042187	37414

Table 7: Comparison of various algorithms

Algorithim	Energy Con.(KWH)	ASLA(%)	VM Mig
MEEVMP	111.77	10.98	17639
EWMA	129.7	8.87	14684
IQR1.5	188.86	9.98	26476
MFPED	110.93	3.62	10975
Mad2.5	184.88	10.18	26292
THr0.8	163.33	9.10	45517
EEVMP	112.56	11.07	18860
Proposed Algo	100.70	9.52	20447

5 Conclusion

This study offers AFED-EF, an individual VM allocation and positioning technique, in order to properly manage changes and achieve optimized energy effectiveness for IoT systems in a CDC. Our paper gives more weightage to energy efficiency than SLA violation. The proposed algorithm, when compared with the VM placement algorithm MEEVMP, reduces the energy consumption by 9.85%, and the average SLA violation by 13.2%. As a result, researchers want to use AFED-EF to expand VM optimizations and assessment methods inside this OpenStack public cloud in the future. This effort will also encompass micro services IoT requirements that operate on virtualized back-end platforms and gateways to describe various heterogeneity in temporal and spatial IoT requirements. This medium enables the proposed technique to continuously regulate the hosts and activate transducers to move some parts of the requests to neighboring hosts to conserve resources and power to prevent violating the customers' SLA.

Further, our attempt would be made to select appropriate clustering techniques to improve the quality of services. We will use different clustering algorithms in order to determine appropriate thresholds for the identification overloaded and underloaded VMs that leads to better performance.

References

[1] Liu, X., Cheng, B., Wang, S.: Availability-aware and energy-efficient virtual cluster allocation based on multi-objective optimization in cloud datacenters. IEEE Transactions on Network and Service Management **17**(2), 972–985 (2020)

[2] Kaur, P.D., Chana, I.: A resource elasticity framework for qos-aware execution of cloud applications. Future Generation Computer Systems **37**, 14–25 (2014)

[3] Zhou, Z., Shojafar, M., Alazab, M., Abawajy, J., Li, F.: Afed-ef: An energy-efficient vm allocation algorithm for iot applications in a cloud data center. IEEE Transactions on Green Communications and Networking **5**(2), 658–669 (2021)

[4] Beloglazov, A., Buyya, R.: Energy efficient allocation of virtual machines in cloud data centers. In: 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, pp. 577–578 (2010). IEEE

[5] Zhou, Z., Abawajy, J.H., Li, F., Hu, Z., Chowdhury, M.U., Alelaiwi, A., Li, K.: Fine-grained energy consumption model of servers based on task characteristics in cloud data center. IEEE Access **6**, 27080–27090 (2018) https://doi.org/10.1109/ACCESS.2017.2732458

[6] Tandon, A., Kulshresha, S., Patel, S.: A review on vm placement scheme using optimization algorithms. Advances in Distributed Computing and Machine Learning: Proceedings of ICADCML 2023, 27–38 (2023)

[7] Wang, J.V., Ganganath, N., Cheng, C.-T., Chi, K.T.: Bio-inspired heuristics for vm consolidation in cloud data centers. IEEE Systems Journal **14**(1), 152–163 (2019)

[8] Moges, F.F., Abebe, S.L.: Energy-aware vm placement algorithms for the openstack neat consolidation framework. Journal of Cloud Computing **8**(1), 2 (2019)

[9] Kulkarni, A.K., Annappa, B.: Context aware vm placement optimization technique for heterogeneous iaas cloud. IEEE access **7**, 89702–89713 (2019)

[10] Fu, X., Zhou, C.: Virtual machine selection and placement for dynamic consolidation in cloud computing environment. Frontiers of Computer Science **9**, 322–330 (2015)

[11] Melhem, S.B., Agarwal, A., Goel, N., Zaman, M.: Markov prediction model for host load detection and vm placement in live migration. IEEE Access **6**, 7190–7205 (2017)

[12] Kulshrestha, S., Patel, S.: An efficient host overload detection algorithm for cloud data center based on exponential weighted moving average. International Journal

of Communication Systems **34**(4), 4708 (2021)

[13] Hussain, M., Wei, L.-F., Lakhan, A., Wali, S., Ali, S., Hussain, A.: Energy and performance-efficient task scheduling in heterogeneous virtualized cloud computing. Sustainable Computing: Informatics and Systems **30**, 100517 (2021)

[14] Liu, X., Li, W., Zhang, X.: Strategy-proof mechanism for provisioning and allocation virtual machines in heterogeneous clouds. IEEE Transactions on Parallel and Distributed Systems **29**(7), 1650–1663 (2017)

[15] Sunil, S., Patel, S.: Energy-efficient virtual machine placement algorithm based on power usage. Computing, 1–25 (2023)

[16] Beloglazov, A., Buyya, R.: Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. Concurrency and Computation: Practice and Experience **24**(13), 1397–1420 (2012)

[17] Zhou, Z., Hu, Z.-g., Song, T., Yu, J.-y.: A novel virtual machine deployment algorithm with energy efficiency in cloud computing. Journal of Central South University **22**(3), 974–983 (2015)

[18] Park, K., Pai, V.S.: Comon: a mostly-scalable monitoring system for planetlab. ACM SIGOPS Operating Systems Review **40**(1), 65–74 (2006)