International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 9 Issue 1 ǁ 2021 ǁ PP. 29-39

Energy-aware Reinforcement Learning Based Dynamic VM Placement Approach for Cloud Data Centers

Anand Jumnal [1] and Dilip Kumar S M [2]
[1,2] *University Visvesvaraya College of Engineering Bangalore, India.*
Corresponding Author: anandsbj1989@gmail.com

Abstract
Nowadays, almost all the applications are being migrated to the cloud and a further increase in demand for cloud services leads to an increase in data centers. Data centers consume a huge amount of energy; hence energy efficiency has become one of the major focuses of research in Cloud computing. However, minimizing energy consumption without any increase in SLA violation or maintaining the model performance is quite challenging. Furthermost existing approach of VM consolidation approach considers system performance as constraints which cause the scheduling overhead and fails to minimize the energy consumption without degrading the cloud service quality. In this research work, we have proposed an efficient framework of three blocks; the first block designs the Resource usage prediction model, the second block focuses on designing the optimized learning based controller through temporal difference learning and stores the Q-values. Further, these values are induced in the optimal VM placement framework where optimal VM placement is designed; moreover, this designed model is evaluated by considering the PlanetLab data and Cloudsim simulator by considering the energy consumption, VM migration, and SLA-violation. Moreover, comparative analysis with the existing model indicates that the proposed mechanism outperforms the existing model.
Keywords: *VM placement; Temporal Difference learning; Energy consumption; cloud computing.*

Date of Submission: 22-12-2020 Date of acceptance: 04-01-2021

I. INTRODUCTION

Cloud computing has become a huge computational paradigm that is a feature through the capabilities of providing the computation service over the internet for an extensive number of users globally [1]. Moreover with the daily rise in demand and rapid growth; cloud infrastructure, cloud services, and data centers are becoming expensive, complex, and high energy-consuming. Moreover, resource management and energy management are considered to be the major concern considering the various vulnerabilities of cloud computing; furthermore, through extensive survey and research, it was found that there would be a nearly 66% increase in electricity demand by 2035[2].Although Cloud computing has several advantages and looks very impressive in implementation, it is facing energy consumption and cost as a major hindrance. Cloud Computing environment comprises thousands of virtual machines which are performing for facilitating the client; thus server consumes 80% of energy and provides only 20 % of utilization. Hence various researcher and professional have focused on developing the efficient mechanism which can reduce energy consumption [3][4]. Moreover, virtualization is one of the key mechanisms for consolidating the virtual machine number on a physical server to minimize energy consumption. For instance, there are two physical servers and a single VM runs on each server; further, it is efficient to run both VM on a single server since the power consumed by two servers becomes half when consolidates into a single. Virtual machine consolidation is a defined approach for the efficient management of the resource in cloud computing; VM consolidation comprises three distinctive approaches i.e. physical machine selection for source, VM selection for migration, and PM selection for destination[5][7]. VM-consolidation faces several issues as it has responsibilities to avoid any kind of performance degradation through an optimal resource. Moreover in VM-consolidation and its optimization, several parameters are considered such as energy consumption, memory, host CPU, data centers and VM-placement, and SLA (Service Level Agreement). VM placement is one of the primary issues in cloud computing; VM placement algorithms should be designed in such a way that they can handle the heterogeneous environment and also they must be capable of catering to the traffic congestion cost and energy consumption. Further Service level Agreements help in identifying the virtual machine capacity; hence considering the large scale data center SLA should not be violated. In general, it needs to achieve a balance among the networks, memory, and CPU.

1.1 Motivation and contribution of this research work

Cloud computing can provide affordable resources for computation and data-intensive applications such as ML-deployed computation, RRR (Rapid Response Request) processing, and big data processing. Although cloud services improve the computation and revenue model through providing the scalable virtual machine to users, they have the biggest drawback of providing the QoS(Quality of Service) such as cost, SLA violation, and environmental factor. In the past several mechanisms have been developed to achieve efficient modeling; they are categorized as VM scheduling and VM consolidation. Hence in this research work, we develop an efficient framework to optimize the VM Consolidation procedure; further, the contribution of this research work is highlighted through the below points:

- In this research work, a dynamic framework is designed for achieving dynamic VM consolidation.
- Proposed VM consolidation comprises three major parts namely Resource usage prediction model, Reinforcement learning-based controller, and optimal VM placement.
- The resource usage prediction model framework is designed for managing resource allocation and management. The prediction model obtains the resource usage status for rational allocation.
- The reinforcement learning-based control framework is designed for achieving the balance between energy and performance.
- Optimal placement is designed for optimal VM placement and VM migration.
- Further integrating all these parts; the proposed framework achieves the minimization in terms of energy consumption, SLA violation, and VM migration.
- The proposed framework is evaluated on PlanetLab data and outperforms the existing model.

This particular research is organized as the first section discusses the background of cloud computing and the significance of VM consolidation, further, the same section highlights the motivation and contribution of particular research work. The second section discusses the various existing VM consolidation mechanism and their drawbacks; the third section focuses on the proposed mechanism along with the algorithm and process. The fourth section evaluates the proposed mechanism by considering the various parameter.

II. LITERATURE SURVEY

In recent years, various researches has been carried out considering the energy performance optimization and VM consolidation; moreover, several methods were developed such as meta-heuristic and greedy heuristic did show promising results. In this section, we have performed an extensive review of the existing methodology. [8] developed a consolidation mechanism through two fixed values computed based on the utilization rates of processors; in [9], the author developed an algorithm based on local regression by integrating the local regression with a particular VM selection policy which is based on minimum migration time. In [10], the author proposed the M-convex mechanism which was based on the semi-quasi framework based on M-convex optimization. Further in [11], section framework was developed and none-or-all migration strategy was considered where all the VMs in given in one active physical machine and it is designed tentatively such that it can be migrated from one migrated PM to the other. Moreover, an iterative approach was adopted till the improvement is observed. Similarly, in paper [12], a paper placement mechanism was introduced which aims at determining the target scheme through an energy-aware algorithm. Moreover, this mechanism classifies the physical machine into the donor group and the receiver group by comparing the target of the previous scheme and their target scheme. In [13], the author tested ACO (Ant Colony Optimization) method which was fully in a decentralized environment and was based on the unstructured P2P network; it promises the minimizing the physical machine and migration, however they suffer from complexity. It achieves better migrations, however, there is complexity. [14] developed an online optimization method based on the metaheuristic algorithm to find the optimized solution in case of dynamic consolidation. Moreover, it promises to achieve better performance while meeting the QoS; further, they developed a multi-objective function that considers the number of migrations and the number of the physical machine. [15] Developed an improvised group genetic algorithm for VMconsolidation to achieve tradeoff among the migration cost and energy consumption in heterogeneous clouds. [16] developed a model named PESOA(Penguin) Search optimization for generating the better VM consolidation mechanism which further helps in planning and going up with various concurrent VMs considering the different applications. [17] developed an algorithm that was mainly based on the co-operative game; it allows cloud providers to set up the federation in such a way that individual profit is improvised concerning the isolation.

Moreover, throughout the literature review we have made few observations through the below points:

- Most existing works considered migration cost and energy consumption as objective, however, the tradeoff between the heterogeneous cloud and energy consumption was ignored.

- Several existing methods considered the scenarios of closing PMs for energy consumption; however, this leads to the problematic and misleading since physical machine in heterogeneous clouds varies with energy consumption characteristics.
- Other existing work focuses on considering the heterogeneous VMs and PMs; however, they ignore the heterogeneity of the given workload.

Hence considering all the above drawbacks of the existing mechanism we address the problem of efficient VM consolidation mechanism and develop a novel methodology in the next section.

III. PROPOSED METHODOLOGY

In this section, the VM consolidation framework is proposed for reducing the energy, reducing VM migration, and also minimizing the SLA Violation. The proposed mechanism comprises three major parts; at first, we develop a particular prediction model to obtain the information regarding the resource storage in advance for decision support for rational allocation of resources. The later controller is developed using the temporal difference learning approach for balance between the application performance and energy consumption. At last optimal VM, placement is designed; moreover, these three-part helps in achieving the dynamic VM consolidation.

3.1 Resource usage prediction model

Since the cloud resources are in a heterogeneous environment; the workload keeps changing dynamically over time. Hence it is essential to develop the absolute prediction model for managing resource allocation and management. The prediction model obtains the resource usage status for rational allocation. Moreover in the proposed mechanism optimized prediction model is introduced; the main motive behind this model is to balance the upper utilization depending on the deviation strength of CPU utilization; a higher deviation indicates the lower upper utilization threshold. Optimized prediction model parts the time series into low value and high value through mean value.

Let's assume that resource utilization is $V_u^s = \{v_{u-o+1}^s \dots \dots .. v_{u-1}^s, \dots .., v_u^s\}$ at given time, where denotes the length, indicates the resource utilization. Hence utilization predicted can be defined as:

$$\bar{v}_{u+1}^{s} = l\left(\sum_{u-0+1}^{u}\left\{v_u^s / v_u^s \geq \bar{\bar{N}}\right\}(|I|)^{-1} + (1-l)(\sum_{u-0+1}^{u}\left\{v_u^s / v_u^s \langle \bar{\bar{N}}\right\}\right)(|M|)^{-1} \quad (1)$$

$$l = \sigma\left(\max\left\{v_u^s \geq \bar{\bar{N}}\right\}\right)^{-1} \left(u-0+1 \leq j \leq t\right) \quad (2)$$

In the above equation, $\bar{\bar{N}}$ indicates medium value of v_u^s, M indicates the lower value and H indicate the higher values of v_u^s,$\bar{v}_{u+1}^s$indicates the prediction utilization at time $u+1$with resource s. Similarly l represents the coefficient which further indicates the higher value weight. Moreover, the main aim here is to predict the host workload. Higher value of σ indicates more utilization of resources. σ is computed as:

$$\sigma = median_j\left(\left|Y_j - median_j\left(Y_j\right)\right|\right) \quad (3)$$

Where Y is univariate dataset.

3.2 Reinforcement learning based controller

In this section, optimized control is design for achieving the balance between performance and energy consumption; moreover, reinforcement learning can achieve management without any prior knowledge. Further reinforcement learning model enables for control model of resource allocation.

3.2.1 Reinforcement learning

Reinforcing Learning enables the agent for learning optimal behavior through trial and error for mapping the situations into actions; further learning process comprises two-element i.e. agent and environment. An agent is one who is responsible for executing actions and analyze the generated result. Here, the agent is considered as the auto-scaler; auto scaler communicates with environment considering the scaling options; further, wait and receive a response this response is known as a reward; Further, each action is taken based on the recent state. Moreover, following reward, auto scaler learns the efficient scaling actin through the trial and error approach.

Moreover, any learning process contains three major steps:
State-space: This is considered as a set of the environment; at each given time step auto scalar occupies the state.
Action space: Auto-scalar chooses the best possible action from a set of all possible action.
Reward: once the action is chosen and execution takes place.

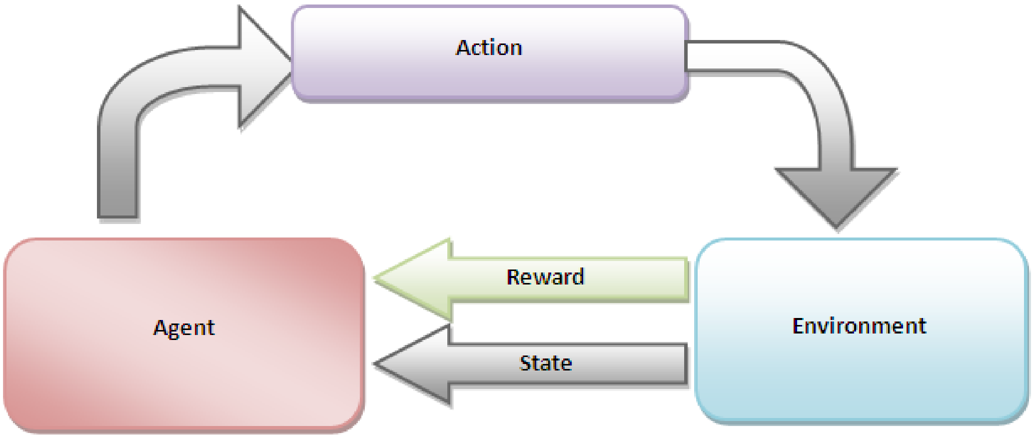

Figure 1: Typical Reinforcement learning.

The **Figure 1** shows the typical diagram of reinforcement learning; here agent perceives the environment and selects an action to select the optimal reward through interacting with the environment. Furthermore, each time the RL model interacts with the environment at first it accepts input as the environment state then the output of the action.

3.2.2 Fuzzy logic based reinforcement learning

Fuzzy logic is used for modeling the human knowledge that helps in converting the knowledge of the expert into the rules and it is applied to the given situation further optimal action is taken following expert knowledge. In general fuzzy rules comprise rules for humans to take a decision.

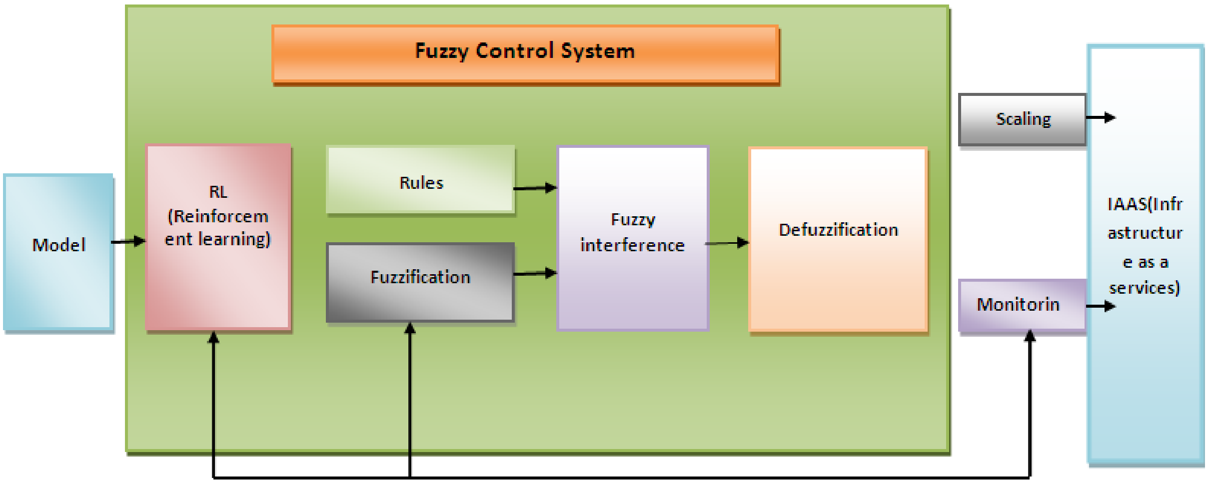

Figure 2 : Fuzzy based reinforcement learning architecture block diagram.

Moreover, we design a fuzzy controller; the fuzzy controller comprises three stages of execution; the first stage is the transformation of a crisp set into a fuzzy set, the second stage is approximate reasoning and the third stage includes the transformation of fuzzy result into the Crisp logic.

The **Figure 2** shows the block diagram of fuzzy-based reinforcement learning architecture; this starts as monitoring various applications such as response time and workload, further this satisfy the resource allocation adoption and system goals. The monitoring component gathers various metrics such as the number of VM which is denoted by, workload denoted through and response time denoted as. This is fed to the learning component and logic controller (LC); Logic controller monitors the data and further computes the VMS is Virtual machine scaling which indicates the increase or decrease in the virtual machine.

In general fuzzy controller is involved in all the function which takes place such as rules of, logic operations and functions; it involves the various step, at first we divide state-space input into different sets through the member function. A member function is denoted as. It indicates the input signal to the given fuzzy set. Further, we adopt the elasticity policies as it allows the cluster for dynamic resource allocation based on the user's demand. Moreover considering the advantage of fuzzy logic and on-policy temporal difference advantage, we integrate both models for achieving the VM consolidation. Moreover, the on-policy temporal difference is carried out in three-stage discussed earlier in the same section

3.2.3 Fuzzy based On-policy Temporal Difference learning for decision process

The earlier discussion on reinforcement learning gives an advantage for capturing the action instead of relying on the static; in this section, we integrate on-policy temporal difference learning which is part of reinforcement learning with the fuzzy-based logic controller. Moreover, here the given state comprises three distinctive characteristics namely and i.e. total number of VM involved response time, and Workload. The **Table 1** presents the fuzzy-based on-policy reinforcement algorithm.

Table 1: Fuzzy based on-policy temporal difference learning algorithm

Input: Learning Rate φ and discount rate Ί

1. Initialization of Q values
2. state monitoring i.e. $state$
3. selecting partial action $action_j$ from state $state$
4. Computing $action$ from $action_j$ and it quality $Qlty$
5. Imply action and monitor updated $state'$
6. Receiving the acknowledgement as rwd
7. Selecting partial action i.e. $action_j$ from $state'$
8. Computing error signal δQ_{FTDL} $(state, action)$ from $action_j'$ and $Qlty\left(state', action'\right)$
9. Updation of $q\left[j, action_j\right]$
10. $state \Leftarrow state'$
11. $action \Leftarrow action_j$
12. Repeat step5 to step 11 until the convergence is obtained

Furthermore, in the above algorithm, the action is taken through a temporal difference Learning mechanism and it is divided into 8 stages.

Stage 1: Initializing the value: In this stage, we assign each and every member to assign some value which describes the particular pair which is known as (state, action) pair; further this pair gets updating while the learning process. In general, all the q-values are set to null.

Stage 2: Choosing an appropriate action: In order to learn from the system, knowledge is explored; this is also known as exploitation strategy. Although the action that has the best reward is chosen else random selection is carried out. In this stage, the main aim is to encourage exploitation till no further exploitation is needed.

Stage 3: Computation of control action: In this stage control action is calculated in accordance through fuzzy controller; the outcome of this stage is a weighted average that can be written as:

$$action = \sum_{j=1}^{0} action_j \times mf_j(y) \tag{4}$$

The above equation, which indicates the number of rules, is a degree of truth for rule and input signal; consequent function is denoted through.

Stage 4: Approximation: In this stage, we approximate the Quality-function from levels of rules and current Q-function; in traditional reinforcement learning single state-action pair is executed at once whereas in the proposed methodology multiple rules can be adopted. Thus Q value for the current state for action is computed as below equation.

$$Qlty(state, action) = \sum_{j=1}^{0} q\left[j, action_j\right] \times mf_j(state) \tag{5}$$

Stage 5: Compute reward value: In this stage, the controller receives the recent values of and which corresponds to the model; the further reward is computed considering the two criteria i.e. SLA violations and resources.

Stage 6: Compute the updated state: moreover in this stage, we compute the updated status considering the action a; further updated status is computed through the below equations and updated equations are denoted as

$$new(state^{'}) = \sum_{j=1} mf_j(state^{'})^{\max(q[j,action_l])} \tag{6}$$

$\max(q[j, action_l])$is maximum q values for status $state'$

Stage 7: Error computation: In this stage error is computed; since this approach is on policy, the proposed method estimates the action value in a given state considering, and the error signal is given in the below equation. is the discount rate;

$$\delta Qlty_{FTDL}(state, action) = rwd + Qlty(state^{'}, action^{'}) \times \Gamma - Qlty(state, action) \tag{7}$$

Stage 8: q-values updating: In this stage we update the q values is updated and given in the below equation.

$$q\lfloor j, action_j \rfloor = \Theta \delta Qlty.mf_j(state(time)) + q\lfloor j, action_j \rfloor \tag{8}$$

In the above equation, is the learning rate, values of learning rate lie between 0 and 1; further lower learning rate means more preference for the old values hence each value are updated this gives more impact. Once the Temporal Difference algorithm converges, values from these are stored in the database and provide the decision support for allocating the resource.

3.3 Optimal VM placement

Table 2: step by step designed for optimal VM allocation

Input: host list and vm list **Output**: vm allocation	
Step1:	Classification of cluster
Step2:	Sorting the cluster (descending order)
Step3:	For each VM in VM list do
Step4:	Min = maximum reward
Step5:	Host allocated is null
Step6:	For each cluster in cluster do
Step7:	For each host in host do
Step8:	IF(host = enough resource)
Step9:	Get reward from Qvalue
Step10:	IF reward is greater than max reward
Step11:	Host = allocated host
Step12:	Reward = maximum reward
Step13:	If allocated host = host
Step14:	Power = manpower
Step15:	Else if allocated host is not null
Step16:	Add allocation(host , allocated VM)
Step17:	Return migraation map

Moreover, VM placement is considered as the eminent for energy minimization. The proposeVM allocation mechanism is given below; at first in cluster VMs are sorted in descending order, further host in the given cluster with higher performance ratio is selected for VM. . The main idea to choose the highest performance ratio is for resource capacity and minimal energy consumption Further current state of each host is found through the Q-Value obtained through the controller part and later it allocates VM to host which gives the higher Q value through allocation. The **Table 2** shows the step by step designed for optimal VM allocation.
Further once a three-way framework is implemented, three eminent parameters are formulated i.e. Energy consumption, VM Migration, and SLA-violation.

3.4 Energy Consumption

Energy consumption can be described through the linear relationship with the CPU utilization, energy consumption can be computed through the below equation

$$eng(u) = \int_{u}^{\inf} P(u)du \tag{9}$$

3.5 VM Migration

VM migration allows the transferring of virtual machines among the physical node within a short time and without any suspension; however, VM migration leaves a highly negative impact on the application performance. Moreover, each migration causes some SLA violation, hence it is necessary to reduce the VM migration; migration length depends on the memory used and the network bandwidth available. Further in the case of the proposed mechanism migration time and any performance degradation is given as in the below equation.

$$U_{nk} = N_k (C_k)^{-1} \tag{10}$$

The above equation presents the time taken to complete the migration; here indicates the memory used whereas indicates the network bandwidth available. Similarly below equation indicates the performance degradation; here indicates the CPU utilization and indicates the time migration started.

$$V_{ek} = 0.1 \int_{u_0}^{u_0+U_{nk}} v_k(u)du \tag{11}$$

3.6 SLA-violation

In a cloud computing environment, meeting QoS is very important. In general QoS requirements are formalized in SLA-violation form; SLA is determined through maximum response time or minimum throughput delivered. SLA-violation is computed through the below equation.

$$SV = \left(\frac{1}{M} \sum_{j=1}^{M} U_{tj} (U_{bj})^{-1} \right) \left(\frac{1}{N} \sum_{i=1}^{N} D_{ek} (D_{sk})^{-1} \right) \tag{12}$$

The above equation contains two terms; the first term indicates SLA violation timer per active host whereas the second term indicates performance degradation occurred due to the migrations. Further in the above equation, U_{t_j} indicates the total time where host j experiences 100% utilization and causes the SLA violation. U_{b_j} indicates total host in the active host indicates total host in the active host, M indicates the number of virtual machine, indicates performance degradation, U_{b_j} indicates total host in the active host indicates total CPU capacity requested and N indicates the number of hosts.

IV. PERFORMANCE EVALUATION

Cloud computing demand has been increasing day by day due to the extensive use of portable gadgets, network appliances, digital instruments, and various devices. The VM consolidation method is a well-known technique which can be utilized in these cloud computing devices. Therefore, the performance of these computing devices must be superior due to the extensive demand for these computing devices in day-to-day life. However, high energy consumption in these computing devices can disturb their performance. In this section, we evaluate the proposed methodology considering the three important parameters i.e. Energy consumption, SLA violation, and VM migration by varying the number of hosts and workload. Moreover, we have used the data available from Planet lab [18] and data is chosen randomly for 10 days. Furthermore, CloudSim [19] is used as a simulation toolkit for modeling as well as simulating cloud computing; it provides an important class to describe management policies, cloud users, and computational resources. CloudSim ensures the reproducibility and reusability of the experiment. Moreover, simulation is carried out on windows 10 platform with eclipse as an editor and java is used as the programming language; further i7 Intel processor packed with 2GB NVidia graphics and 16 GB RAM. Here FFD is First Fit Decreasing.

4.1 Energy Consumption

The energy consumption metric is depicted in **Figure 3** and **Figure 4**, **Figure 3** shows the energy consumption on various workloads whereas figure 4 shows the energy consumption by varying the number of the host.

Moreover in **Figure 3**, 10 distinctive workloads are considered and it is observed that for workload 1, the existing model takes 180 kWh whereas the proposed model takes 150.5. Similarly for workload 2, workload 3, and workload 4 energy consumed by the existing model is 153.02, 180.17, and 219.04 respectively. Whereas the proposed model takes 144.05, 165. 71 and 201.86 respectively. Further w5, w6, w7, and w8 consume 188.44, 275.84, and 222.36, and the proposed model requires 173.95, 256.75 and 206.8 respectively.

Further, proposed model is evaluated by considering the various number of host, a comparison has been carried out and depicted in **Figure 4**. Moreover, in the case of 800 hosts the energy required by the existing model is 204.22 kWh, whereas the proposed model requires the only 188.86 kWh. Similarly increasing the number of the host as 1600, 2400, and 3200 existing model requires 203.46, 206.62, and 206.62 whereas the proposed model requires 190.6, 191.6, and 191.6 respectively.

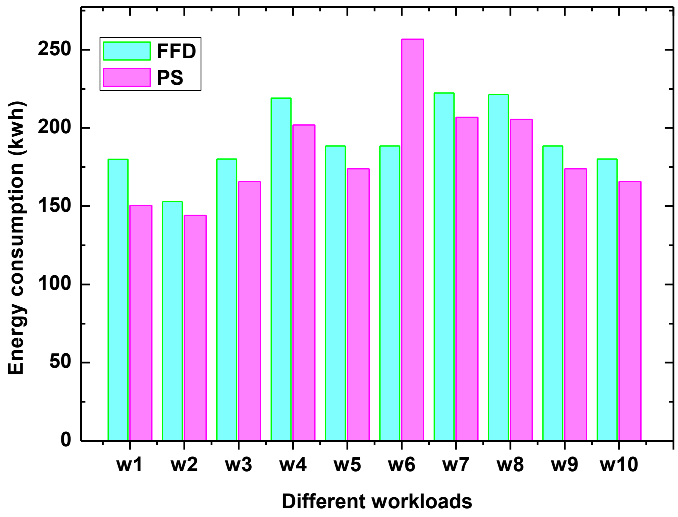

Figure 3: Energy consumption for different workloads.

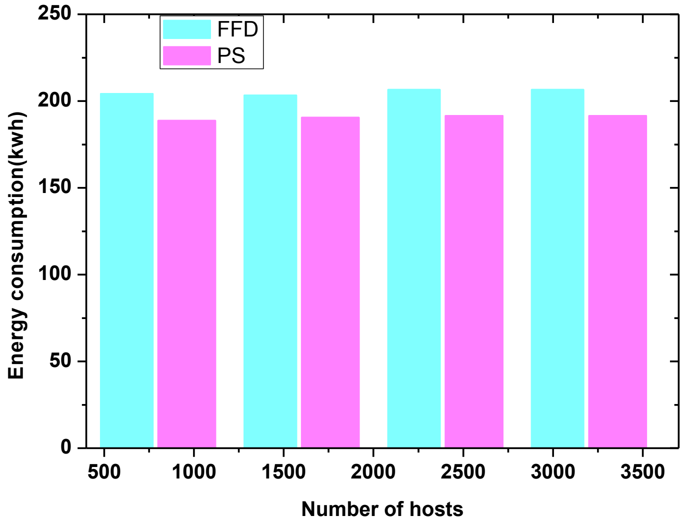

Figure 4: Energy consumption for different number of hosts.

4.2 SLA violation

QoS (Quality of service) is a key issue in cloud computing; as different users apply different applications. Further, we compare the average SLA violation considering the workload and number of hosts, **Figure 5**shows the comparison on workloads, and **Figure 6** shows the comparison on the number of the host. Further in figure 3, for w1, w2 and w3 average SLA violations are 10.17, 10.21, and 10.21 % whereas the proposed methodology average SLA violation is 9.98, 10.17, and 10.14 %. Similarly for w4, w5, and w6, average SLA violations are 10.18, 10.18, and 10.15 whereas the proposed model takes average SLA violations are 10.19, 10.09, and 10.07 respectively. Further in W7, W8, W9 and W9 average SLA violations by the existing model are 10.14, 10.21, 10.25, and 10.40 % whereas the proposed model SLA violation is 10.07, 10.23, 10.11 and 10.39% respectively. Further evaluation is carried out varying the number of the host; in figure 4, for 800 hosts, the average SLA-violation by the existing model is 10.17 % whereas the proposed model SLA violation is 9.98 %. Increasing the number of a host as 1600, 2400 and 3200, average SLA violation by the existing model is 10.09, 10.16 and 10.16 % whereas SLA violation by proposed model is 9.95, 9.98 and 9.98 % respectively.

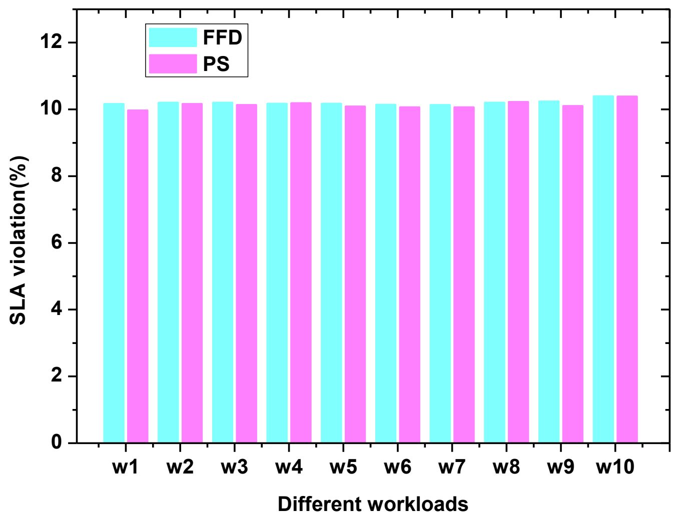

Figure 5: SLA Violation for different workloads.

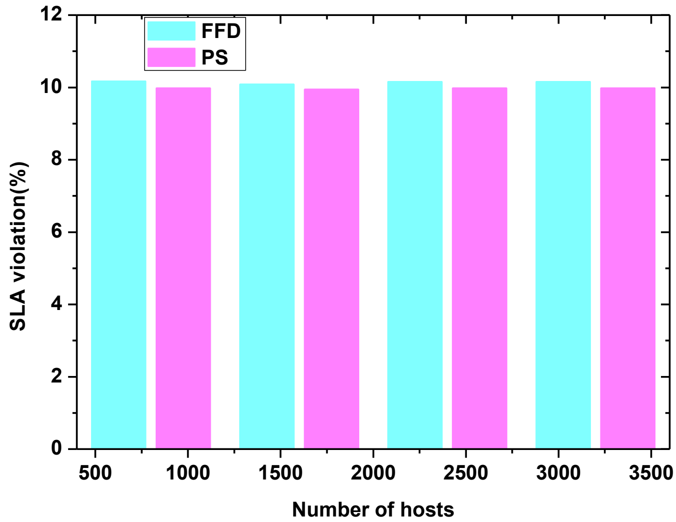

Figure 6: SLA Violation for different number of hosts.

4.3 VM migration

VM migration is defined as the method of moving VM between the physical machines without any interruption; here we compare the VM migration; the lower migration value indicates the better and efficient model. **Figure 7** and **Figure 8** shows the graphical comparison by varying the number of host and number of workload; in figure 5, for 800 number of the host; existing methodology achieves the value of 29901 whereas proposed mechanism achieves the value of 26476. Further increasing the number of host 1600, 2400 and 3200; existing model takes a value of 29837, 30138 and 30138 whereas proposed mechanism achieves 26481, 26523 and 26523 respectively.

Furthermore in the figure proposed mechanism is evaluated by considering the different workloads; as w1, w2, w3, and w4 take the 29901, 23256, 27177, and 33084 whereas the proposed model takes C, 20879, 23579, and 28948 respectively. Similarly for W5, W6, W7, and W8 existing model migration are 27754,39910, 32502, and 32064 whereas proposed model migration is 24778, 35245 respectively; at last for Workload 9 and workload 10, existing model migration is 28026 and 26511 whereas proposed model migration is 24973 and 22903 respectively.

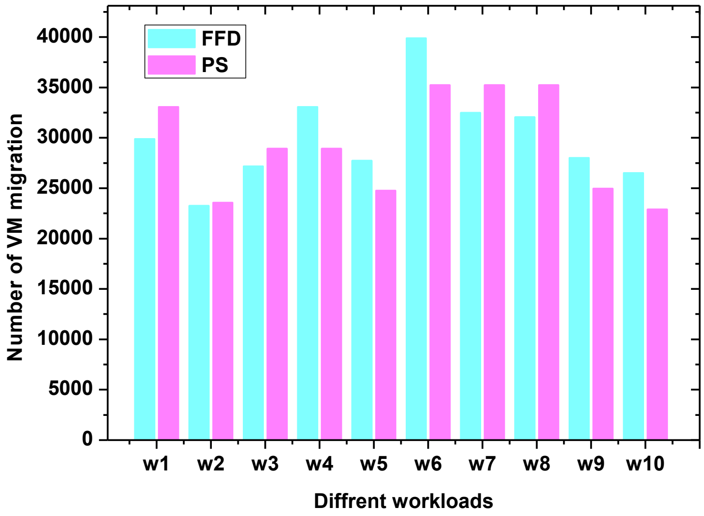

Figure 7: Number of VM migrations for different workloads.

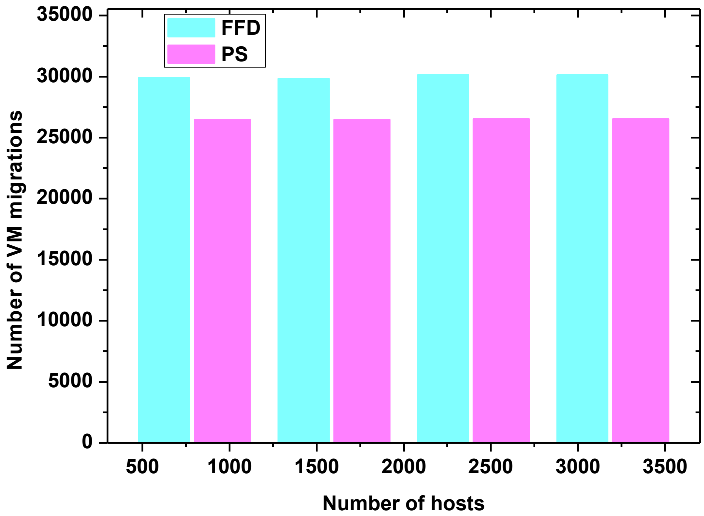

Figure 8: Number of VM migrations for different number of hosts.

V. CONCLUSION

VM consolidation is considered one of the emerging solutions for reducing energy consumption in the cloud data center. In this research work, an efficient mechanism is developed to achieve the trade-off among the energy and performance; moreover, three important metrics VM migration, energy consumption, and VM migration are considered as the evaluation parameter. In this research we have developed an energy-aware VM consolidation framework which comprises three sub-framework; the first framework is designed for the model prediction, the second sub-framework is for the decision process and the third sub-framework is for VM migration. Moreover, the proposed mechanism is evaluated on the PlanetLab dataset using the cloud sim simulator; further comparative analysis is carried out on the three-parameter discussed above, and the proposed model simply excels in comparison with the existing model.Although the proposed model performs better than the existing model; considering the VM consolidation area as a novice research area in cloud computing, there are several still open research area which needs to be focused.

REFERENCE

[1]. Make IT Green (2010) Cloud computing and its contribution to climate change. Greenpeace International, p 83

[2]. Wang J, Gong B, Liu H, Li S (2019) Intelligent scheduling with deep fusion of hardware-software energy-saving principles for greening stochastic nonlinear heterogeneous super-systems. ApplIntell, 1–14

[3]. El Motaki S, Yahyaouy A, Gualous H, Sabor J (2019) Comparative study between exact and metaheuristic approaches for virtual machine placement process as knapsack problem. J Supercomput, 1–21
[4]. Shaw R, Howley E, Barrett E (2019) An energy efficient anti correlated virtual machine placement algorithm using resource usage predictions. Simul Model Pract Theory 93:322–342.
[5]. Xu J, Fortes JAB (2010) Multi-objective virtual machine placement in virtualized data center environments. In: 2010 IEEE/ACM int'l conference on & int'l conference on cyber, physical and social computing (CPSCom) green computing and communications (GreenCom). IEEE, pp 179–188
[6]. Gao Y, Guan H, Qi Z, Hou Y, Liu L (2013) A multi-objective ant colony system algorithm for virtual machine placement in cloud computing. J Comput SystSci 79(8):1230–1242
[7]. Zheng Q, Li R, Li X, Shah N, Zhang J, Tian F, Chao K-M, Li J (2016) Virtual machine consolidated placement based on multiobjective biogeography-based optimization. FuturGener Comput, Syst 54:95–122
[8]. Jamali S, Malektaji S, Analoui M (2017) An imperialist competitive algorithm for virtual machine placement in cloud computing. J ExperTheorArtifIntell 29(3):575–596
[9]. Li Z, Li Y, Yuan T, Chen S, Jiang S (2019) Chemical reaction optimization for virtual machine placement in cloud computing.ApplIntell 49(1):220–232.
[10]. A. Beloglazov and R. Buyya, ``Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers," in *Proc. 8th Int.Workshop middleware Grids, Clouds e-Sci.*, 2010, pp. 1_6.
[11]. A. Beloglazov and R. Buyya, ``Optimal online deterministic algorithms and adaptive heuristics for energy and performance ef_cient dynamic consolidation of virtual machines in cloud data centers," *ConcurrencyComput., Pract. Exper.*, vol. 24, no. 13, pp. 1397_1420, 2012.
[12]. Z. Huang and D. H. K. Tsang, ``M-convex VM consolidation: Towards a better VM workload consolidation," *IEEE Trans. Cloud Comput.*, vol. 4,no. 4, pp. 415_428, Dec. 2016.
[13]. A. Murtazaev and S. Oh, ``Sercon: Server consolidation algorithm using live migration of virtual machines for green computing," *IETE Tech. Rev.*, vol. 28, no. 3, pp. 212_231, 2011.
[14]. A. Verma, P. Ahuja, and A. Neogi, ``pMapper: Power and migration cost aware application placement in virtualized systems," in *Proc. Middleware*. Berlin, Germany: Springer, 2008, pp. 243_264.
[15]. E. Feller, C. Morin, and A. Esnault, ``A case for fully decentralized dynamic VM consolidation in clouds," in *Proc. 4th IEEE Int. Conf. CloudComput. Technol. Sci. (CloudCom)*, 2012, pp. 26_33.
[16]. F. Farahnakian, A. Ashraf, T. Pahikkala, P. Liljeberg, J. Plosila, I. Porres, and H. Tenhunen, ``Using ant colony system to consolidate VMs for green cloud computing," *IEEE Trans. Services Comput.*, vol. 8, no. 2, pp. 187_198, Mar./Apr. 2015.
[17]. Q.Wu, F. Ishikawa, Q. Zhu, and Y. Xia, ``Energy and migration cost-aware dynamic virtual machine consolidation in heterogeneous cloud datacenters," *IEEE Trans. Services Comput.*, to be published.
[18]. N. Divya, S. Suprakash, and S. P. Balakannan, ``An ef_cient VM consolidation using penguin search optimization algorithm (PeSOA) for making cloud greener," in *Proc. Int. Conf. Elect., Electron.,Optim. Techn.(ICEEOT)*, 2016, pp. 2793_2796.
[19]. M. Guazzone, C. Anglano, and M. Sereno, ``A game-theoretic approach to coalition formation in green cloud federations," in *Proc. IEEE/ACM Int.Symp. Cluster, Cloud Grid Comput.*, May 2014, pp. 618_625.
[20]. K. Park and V. S. Pai, ''CoMon: A mostly-scalable monitoring system for PlanetLab,'' ACM SIGOPS Oper. Syst. Rev., vol. 40, no. 1, pp. 65–74, 2006.
[21]. R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, ''CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,'' Softw., Pract. Exper., vol. 41, no. 1, pp. 23–50, Jan. 2011.