Energy-Aware Prediction-Based Load Balancing Approach with VM Migration for the Cloud Environment

Durga Patel, Rajeev Kumar Gupta and R. K. Pateriya

1 Introduction

In the IT field, a new technology is emerging regularly and adoption of these technology demands several investments on infrastructure that is no't possible for all the enterprises. Cloud computing has resolved these issues by providing services on-demand basis that may be accessed through web on pay-per-use basis. The cloud is one of the most demanding technologies in IT industries [1, 2]. The cloud has created a new look to align IT and business visions. It is providing Software-as-Service (Saas), Platform-as-Service (Paas) and Infrastructure-as-Service (Iaas) in a very virtualized cloud atmosphere. The cloud computing power is created through distributed computing and the advanced communication networks. Cloud works on the principle of virtualization of resources with on-demand and pay-as-you go model policy. It is originated from the large-scale Grid atmosphere. It is a technology of computing that is extensively utilized in today's business moreover as a society. Applications run within the cloud, the user will access it anywhere at any time through an associate Internet-enabled mobile device or a connected PC.

Due to some attractive features, demand for the cloud resources are increasing gradually. In order to fulfill this demand huge number of servers are used which runs several VMs. These servers consume large amount of energy for their operational and cooling function, which leads to emitting large amount of CO_2. This energy consumption can be minimized by reducing the number of running server in the data center [3]. The key thought of the cloud computing is Virtualization. Virtualization

D. Patel (✉) · R. K. Gupta · R. K. Pateriya
Department of CSE, MANIT, Bhopal, India
e-mail: durgapatel28@gmail.com

R. K. Gupta
e-mail: rajeevmanit12276@gmail.com

R. K. Pateriya
e-mail: pateriyark@gmail.com

R. K. Shukla et al. (eds.), *Data, Engineering and Applications*,
https://doi.org/10.1007/978-981-13-6351-1_6

has become a very popular valuable and standard technology for cloud computing environment. Virtualization is the concept that abstracts the coupling between the hardware and the operating system and enhances the computing power without installing additional hardware infrastructure. Virtual Machine Monitor (VMM) is a great tool for administrator which is responsible for the separation between software and hardware and add the flexibility to migrate the VM from one PM to another. Virtual Machine Migration allows load balancing which leads to increase the resource utilization and energy saving [4]. VM migrations techniques are classified mainly into three categories:

i. Energy-Efficient Migration Techniques
ii. Fault-Tolerant Migration Techniques
iii. Load Balancing Migration Techniques.

i. Energy-Efficient Migration Techniques
The energy consumption of data center is mostly estimated by the utilization of the servers and their systems. The servers usually went up to 70% of their highest power consumption, even at their low utilization level. Hence, there is a necessity for migration methods that preserves the energy of servers by ideal resource utilization [5].

ii. Load Balancing Migration Techniques
The main goal of load balancing migration technique is to distribute the load across the physical server to enhance the scalability of servers in a cloud environment. Load balancing helps in evading bottlenecks, minimizing resource consumption, enhancing scalability, implementation of fail-over, over-provisioning of resources, etc. [4].

iii. Fault-Tolerant Migration Techniques
Fault tolerance gives the authority to the virtual machines to do its job continuously if any part of system failure occurs. In this technique, virtual machine migrates from one physical server to another physical server and the migration is done based upon the prediction of failure. The advantages of the fault tolerance migration technique are to avoid performance degradation of applications, improve the availability of the physical server [5, 6].

This paper presents an energy-aware VM migration-based load balancing approach for the cloud, which will reduce the energy consumption and minimizes number of VM migration while keeping efficient utilization of resources. In order to minimize the number of migration due to the temporary peak load forecasting approach is used.

2 Literature Survey

Shivani Gupta et al. (2015) proposed a load balancing approach based on Dynamic Threshold [7] in cloud computing. Depending on the systems, current state load balancing problem can be categorised into two approaches namely: Static approach and Dynamic approach. In the static approach, value of the lower and upper thresholds are fixed based on the server configuration whereas in dynamic threshold approach value of the threshold changes with time. Static threshold is more suitable for the cloud due to dynamic nature of the cloud services.

Raj Kumar Buyya et al. (2010) proposed Modified Best-Fit Decreasing (MBFD) algorithm for the efficient utilization of the cloud resource [3]. This paper introduced server consolidation approach for reducing the energy consumption. Fixed lower and upper thresholds are used to find the status of the PM. If the status of the PM is underloaded, all VMs are shifted or migrated to the other PM whereas if the status of the PM is overloaded then of the Best-Fit Decreasing (BFD) algorithm finds the VM where power difference between after and before allocation of VM is minimum. This approach seems good not mitigate the migration due to the temporary peak load.

Akshay Jain et al. (2013) proposed a load balancing approach, which is based on the threshold band [8]. Instead of using single upper threshold, this approach uses the threshold band to define the overloaded status of the PM. The decision of VM selection for the VM migration is depending on two factors: threshold value and mean deviation of all hosts. The value of the threshold band is different for each host depending on the type of application running on it. This approach had not used any prediction approach, which leads to increase in the migration due to the temporary peak load. Iniya Nehru et al. (2013) proposed prediction-based load balancing approach for cloud. In this technique, Neural Network based prediction approach is used to predict the load on PM, which uses past historical load data of each PM [9]. Nada M. Al Sallami et al. (2013) proposed technique for load prediction based on the artificial neural network. For Workload distribution across the nodes back-propagation learning algorithm is used and trained, feed forward Artificial Neural Network(ANN). This proposed technique is very efficient and simple when effective training or learning data sets are used. Artificial Neural Network uses prediction policy and in advance it predicts the demand of resources than start allocating the resources according to that demand [9].

S.K. Mandal et al. (2013) proposed an energy-efficient Virtual Machine placement algorithm for on-demand access to infrastructure resource in cloud computing which reduces the VM allocation time and optimize the resource utilization. In order to improve the resource utilization instead of directly sending the VM requests to the host machine, this algorithm creates the binary search tree of requested VM and then send it to the VM scheduler. To implement the best-fit strategy, the PM's are also organized in binary search tree. Now VM scheduler selects the maximum requirement VM from the VM tree and searches for the host that would be best fit in requirement. By using the binary search tree data structure, the host searching time will take average O(log N) where N is the number of physical machines [10]. Main limitation of this approach is that it is a single-dimensional best-fit approach, which is suitable for the single-dimensional cloud computing environment, but in cloud computing user can requests for multiple resources simultaneously. Moreover, in this algorithm hosts are organized in binary search tree if binary search tree is left skewed or right skewed then it will take O(n) search time in worst case to find the best-fit host because best-fit host will be in the leaf node, i.e., the last node of the binary search tree [10].

3 Proposed Methodology

Cloud is the commercial model where proper utilization of physical resources is the prime requirement of the provider. This paper proposed a load balancing approach, which optimizes the cloud services by increasing the resource utilization. The main components of cloud are data center, server (PM), and VM where each data center can have multiple PM and each PM can run a number of VMs. When user demands for the resources hypervisor (VMM) creates a VM and assigns to the user. Since in cloud, resource requirement of the VM change dynamically so it may lead to a situation where PM is unbalanced. In order to balance these PM and increase the resource utilization an efficient load balancing approach is required that takes the decision according to the situation.

This paper proposed double threshold based dynamic load balancing approach, where thresholds are calculated based on the host utilization. VMs are migrated from one PM to another based on these threshold values. Since the number of migrations affect the performance of cloud services so to minimize the number of migrations is also one of the prime requirement of the cloud provider. In order to achieve this proposed approach, we use the prediction model. Table 1 shows the symbol used in the proposed approach.

Following steps are involved in the proposed load balancing approach:

i. Calculate load on the PM and VM.
ii. Calculate the upper and lower threshold to find the overloaded and underloaded condition.
iii. Use prediction model to minimize the number of migrations
iv. Select suitable VM for the migration by using optimization technique.

Table 1 Description of the various symbols

PM^{load}	Load on the host or server
VM^{cpu}	CPU utilization of the VM
VM^{bw}	Bandwidth utilization of the VM
VM^{ram}	RAM utilization of the VM
VM^{load}	Load on the VM
n	Number of VMs running in single PM
M	Number of PM in the data center
PM_{upper}	Upper threshold of the PM
PM_{lower}	Lower threshold of the PM

v. Select suitable host to place the selected VM.

3.1 Load Calculation for PM and VM

One of the challenging tasks in the cloud is to calculate the load on physical and virtual machine because of its dynamic nature. Load on the PM mainly depends on the three parameters namely CPU, memory, and bandwidth. CPU utilization, RAM utilization and Bandwidth utilization can be calculated by using the following formulae.

$$VM^{cpu} = \frac{\text{Total Requested Mips}}{\text{Total Mips of PM}}$$

$$VM^{bw} = \frac{\text{Bandwidth used by VM}}{\text{Total Bandwidth of host}}$$

$$VM^{ram} = \frac{\text{Ram used by VM}}{\text{Total RAM of host}}$$

Since each PM can have a number of VMs and each VM running some applications, so total load on the PM is the summation of load of all VM running in the server. This paper considers only CPU utilization of the VM for calculating load on the PM. Hence,

$$VM_{load} = VM^{cpu}$$

Total load on the host is given by

$$PM^{load} = \frac{\sum_{i=1}^{n} VM_i^{cpu}}{Total\ MIPS\ of\ the\ PM}$$

Where n is number of running VM in the *i*th PM.

3.2 Lower and Upper Threshold

The performance of the PM depends on its load, which decreases with increase in load. So to define the amount of resources that can be used lower and upper thresholds are used. These thresholds define the overloaded and under loaded situation. A PM is underloaded if PM load is lower than its lower threshold and if the load is more than upper threshold than PM is called overloaded. Thresholds can be Static as well as dynamic. In case of static threshold, lower and upper thresholds are fixed thus, these value does not change with time, while in the dynamic threshold, value of the lower and upper thresholds keep changing (usually lower threshold value is kept constant because it is pre-decided according to system configuration. Due to this reason, the proposed approach uses fixed value of lower threshold, i.e., 20. Appropriate value of upper threshold plays a vital role in the load balancing and highly influence the number of migration.

The lower value of upper threshold will increase the resource wastage whereas higher value will lead to higher number of migrations. Hence, to balance the tradeoff between resource wastage and number of migration is a challenging task. In order to find upper threshold, the proposed approach takes the average of all PM load available in the data center. The main component of the PM is the CPU, RAM and bandwidth, but the previous studies say that CPU is the core element of the cloud resources and performance of the cloud services mainly depends on the CPU. Due to this, proposed approach considers only CPU load for evaluating the performance. In this approach, upper load on the PM is propositional to the average load of the data center. To ensure the high performance, upper threshold should not be less than 70% and not exceed 95% of the total capacity of the PM.

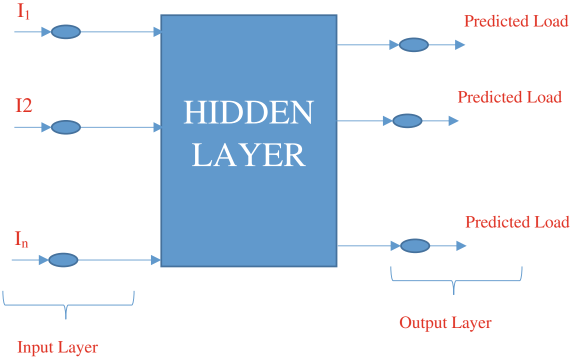
$$
\begin{aligned}
PM_{upper} &= \frac{\sum_{i=1}^{m} PM^{load}}{m}, \text{ Hence}\\
PM_{lower} &= 0.2\\
PM_{upper} &= 1 - x * T_{avg}\\
PM_{upper} &= 1 - \frac{5 * T_{avg}}{100}\\
70 &< PM_{upper} < 90
\end{aligned}
$$

3.3 Load Prediction Using Artificial Neural Network

Due to the dynamic behavior of the cloud environment, if the VM is migrated on the basic of lower and upper threshold then it will lead to the unnecessary migration due to the temporary peak load. This migration can be avoided by using some forecasting method. Our proposed approach uses a prediction approach, which is based on the ANN for prediction the future load on the PM. In ANN backpropagation algorithm is used with Multi-layer perceptron model for finding the predicted load. In neural network, a basic model of three layers with input nodes are used for implementation of ANN. The load prediction of each PM is done by training the perceptrons, for training these perceptrons backpropagation algorithm is used. In order to make effective prediction of load the ANN is trained with a data set. The training is finished by implementing the set input and target data (Fig. 1).

After training the network, load of each host is given as input to ANN, which gives the predicted load in the output layers. ANN is divided into three layers first, the input layer, second, the hidden layer, third is the output layer. Each layer has their own calculations and in combination produces an optimized result which is near to

Fig. 1 ANN architecture



the real result. The input layer takes the CPU utilization value in terms of MIPS and number of VM available at the host. These two values are given and we make an activation function in order to activate the ANN for functioning.

$$\text{input layer: i} = \text{g}\left(\sum_{i=0}^{n} x_i w_i + b\right)$$

Hidden layer of ANN implements an exponential function in order to get trained and circulates the data from input layer to hidden layer and vice versa.

Output layer contains the neuron, which produces the output using following formula:

$$\text{output layer} = \frac{1}{1 + e^{-i}}$$

The output layers give the predicted load, which is used to trigger the VM migration polices.

3.4 VM Selection Algorithm

When the status of the PM is overloaded or underloaded some/all VM need to be migrated to balance the PM. Since each host runs a number of VM and these VM change their resource requirement dynamically, so VM selection is the critical job in the cloud. A wrong VM selection may lead to increase the total migration time and downtime. When the status of PM is underloaded all VM running on that PM has to be migrated to save energy whereas if the status of PM is overloaded proposed approach migrate the VM till the PM is balanced. In the overloaded case, we migrate the VM whose size is greater than or equal to the difference between upper threshold and host utilization. VM with higher VM_{size} will be chosen first.

$$\mathrm{VM}_{\mathrm{size}} = PM_{upper} - PM^{load}$$

VM Selection Algorithm when the PM is Overloaded

```
Algorithm 1 VM Selection Algorithm
Input - VmList,HostList
Output - MigrationList
Arrange host into decreasing order according to utilization
for each h in hostList do
hostUt ←host.util()
bestVmUt←Utilization of first VM
whilehostUt>host.upThresh do
for each vm in vmList do
diff ← hostUt − host.upThresh
if vm.util() > diff then
$T_{avg}$ ← vm.util()
if $T_{avg}$<bestVmUt then
bestVmUt ← $T_{avg}$
bestVm ←vm
else
if bestVmUt= First VM then
bestVm ← vm
break
hostUt ← hostUt − bestVm.util()
migrationList.add(bestVm)
vmList.remove(vm)
```

VM Selection Algorithm when the PM is Underloaded (VM Consolidation)

```
if hotUtil < lowThresh() then
Migrate all VM from the host
```

3.5 Select the Host to Place the Selected VM

Final PM Selection is the most critical step in the VM migration, because it affects the overall performance of the system. Wrong selection of the PM may increase the number of VM migration as well as resource wastage. A VM needs to be placed in two different situations either for the new VM or migrated VM. In both cases, the proposed approach chooses VM in such a way it will minimize energy consumption and resource wastage. In the proposed algorithm, first we find the overloaded host and underloaded host by the use of upper threshold and lower threshold after that selection of VM and migration is done after this load balancing by using DDT algorithm and then data sent to the ANN, it gives predicted host CPU utilization list and this CPU utilization list is used for calculating upper threshold and finding the overloaded host. So, in second iteration VM migration is done according to the predicted load given by ANN and this process is continued.

PM Selection Algorithm to Place New VM

```
Algorithm 3 HostSelection for VM
Input - HostList, vmList
Output -  Allocation of VMs
Sort all PM according their utilization
for eachvm in vmList do
for each host in hostList do
if Host_Load<=H_UTD &&Host_Load>=H_LTD
Assign VM to the host where less increment in the power
else
Activate new host and assign VM to that host
```

Energy-Aware Optimized PM Selection Algorithm to Place Migrated VM

```
   Input: Host CPU Utilization List., Output: Minimizing VM Migration
1.  Host_Utl_List in MIPS
2.  if (Host_Utlgreater thanH_LT &&Host_Utl  less than H_UT) then
3.    Wait for time t sec
4.  elseif(Host_Utl  > H_UT) then
5.  Select VM from VM Migration List-1 &migrate VM with Host_Utl between UT and LT and
     (UtliAfterAlloc greater than PrvUtli this Host ) then
6.       Migration successful
7.        if
8.            successful then
9.             wait for time t
10.       else
11.            repeat
12. elseif (Host_Utl <  H_LT) then
13.      Select VM from VM Migration List-2 & migrate VM with Host_Utl > LT and
         (UtliAfterAlloc > PrvUtli this Host ) then
14.      Migration successful
15.      if
16.        successful then
17. send the data to ANN
18.     else
19.           repeat
20.     ANN gives predicted Host CPU Utilization List
21.   repeat
22. form step-1
```

4 Performance Evaluation

4.1 Experiment Setup

For performing experiments physical machine has a 2.40 GHz Intel(R) core(TM) i5 processor, 8 GB RAM, 1 TB hard disk, and 64-bit Operating System Windows 10 Professional. In our experiment latest version of CloudSim toolkit, CloudSim 3.03 [10–12] is used to for simulation of Cloud computing scenarios and MATLAB 2015a is used for implementing ANN.

4.2 Experimental Results

We have analyzed different scenarios by taking different numbers of hosts, different numbers of virtual machines and various number of cloudlets (tasks), i.e., load to evaluate the performance of proposed algorithm. It is essential to use workload traces from real system. In this simulation, we have taken different work load. We have plotted different graphs based on different workloads between two strategies, proposed method DDT with ANN and the existing method DDT (Dynamic Double Threshold). The performance and accuracy of the proposed approach is calculated for two parameters (i.e., Energy Consumption(Energy_Con), Number of VM migrations(VM_Mig)). These results can vary according to the different time slots (Table 2).

Figure 2 represents a comparison between VM migration and energy consumption for dynamic double threshold and double threshold with ANN. It is evident from the graph that as the number of VM migration increases energy consumption increases

Table 2 Performance of the various approaches for different number of PM

No. of hosts	No. of VMs	No. of cloudlets	VM_Mig (DDT)	VM_Mig (ANN)	Energy_Con (DDT)	Energy_Con (ANN)
5	10	20	3	3	0.12	0.12
10	20	30	16	16	0.26	0.26
15	30	40	28	26	0.34	0.30
20	40	50	33	31	0.48	0.40
25	50	60	54	50	0.62	0.52
30	60	70	57	50	0.77	0.54
35	70	80	66	62	0.90	0.81
40	80	90	77	77	0.98	1.00
45	90	100	101	81	1.12	0.01
50	100	110	107	85	1.23	1.20

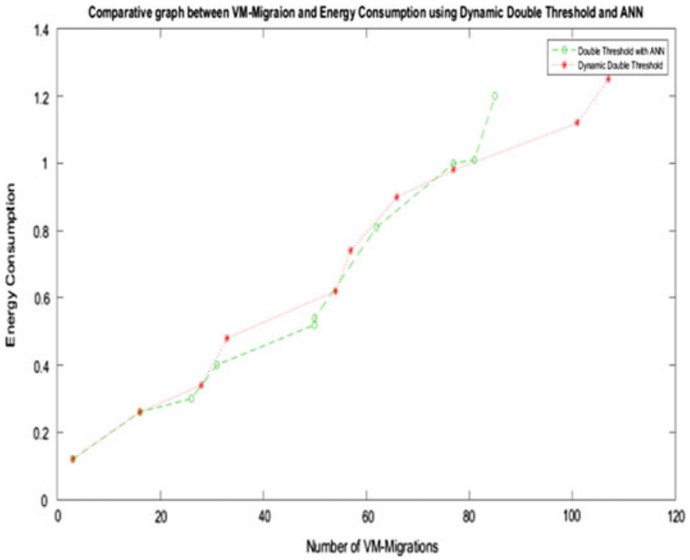


Fig. 2 Comparative graph between VM migration and energy consumption using DDT and DDT with ANN

in both of the load balancing algorithms. The graph between VM migration versus energy consumption is plotted by continuously increasing the number of hosts, VM and cloudlets. Analysis shows that number of VM migration is less for double threshold with ANN, consequently energy consumption will also be less.

Figure 3 represents number of VM migrations and energy consumption with the condition of increasing number of cloudlets at constant number of hosts and VMs. In this case, number of VM migration and energy consumption for dynamic double threshold is greater than double threshold with ANN. There is one outlier with the trend, with number of migration equivalent to 16. This condition arises due to external factors existing in cloud (Table 3).

Figure 4 represents comparison between number of VM migration and energy consumption for dynamic double threshold and double threshold with ANN, keeping number of hosts and cloudlets constant. Both algorithms perform equivalently in this situation (Table 4).

Figure 5 represents a comparison between VM migration and energy consumption for dynamic double threshold and double threshold with ANN. The above case is considered for different time intervals keeping number of hosts, VMs and cloudlets constant. Dynamic double threshold's performance is poorer than double threshold with ANN.

The workspace area shows in Fig. 6 the variables and parameters related to ANN implementation. The values of different parameters are also known to us. From here, we can plot graphs for different parameters available like value of input and output, Root Mean Square Error rate, etc. (Fig. 7).

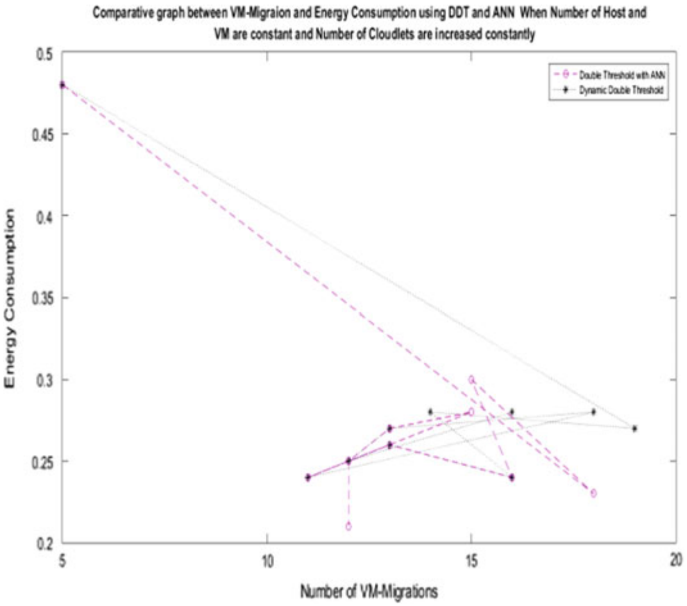


Fig. 3 Comparative graph between VM migration and energy consumption using DDT and ANN (Number of host, VMs are constant and cloudlets are increased constantly)

Table 3 Performance of the various approaches for 10 number of PM

No. of hosts	No. of VMs	No. of cloudlets	VM_Mig (DDT)	VM_Mig (ANN)	Energy_C (DDT)	Energy_C (ANN)
10	5	50	0	0	0.06	0.06
10	10	50	8	8	0.13	0.13
10	15	50	12	12	0.18	0.18
10	20	50	13	13	0.25	0.25
10	25	50	21	21	0.36	0.36
10	30	50	20	20	0.42	0.42
10	35	50	28	25	0.45	0.43
10	40	50	30	26	0.51	0.47
10	45	50	28	28	0.46	0.46
10	50	50	30	29	0.48	0.47

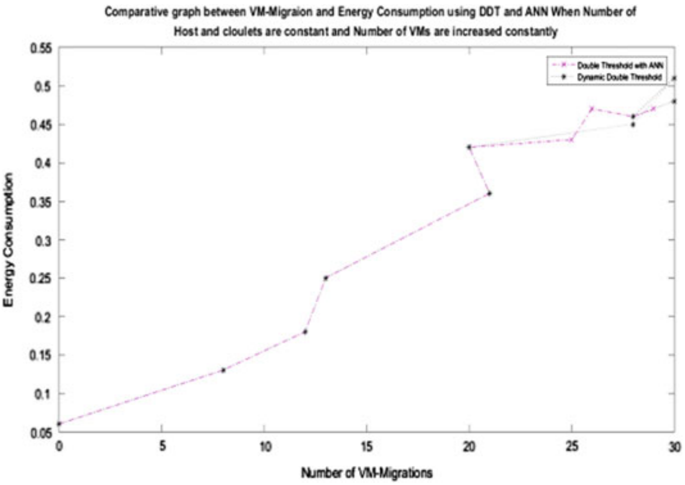


Fig. 4 Comparative graph between VM migration and energy consumption using DDT and DDT with ANN (Number of host, cloudlets are constant and VMs are increased constantly)

Table 4 Performance of the various approaches for 20 number of PM

No. of hosts	No. of VMs	No. of cloudlets	VM_Mig (DDT)	VM_Mig (ANN)	Energy_Con (DDT)	Energy_Con (ANN)
20	35	35	32	27	044	0.41
20	35	35	27	25	0.45	0.45
20	55	35	27	26	0.41	0.40
20	35	35	29	27	0.45	0.41
20	35	35	28	29	0.40	0.41
20	35	35	22	24	0.45	0.40
20	35	35	33	28	0.43	0.41
20	35	35	28	27	0.41	0.41
20	35	35	33	26	0.41	0.40
20	35	35	30	28	0.42	0.41

By using ANN, the percentage of accuracy is “94.5555%” which is better prediction and this accuracy level will increase more and more to nearly about 100%, i.e., 99.9999%. This much of accuracy will be achieved as many input values are passed through ANN.

The result shows that proposed method Energy-Aware VM Migration-based Load Balancing Approach for the Cloud Computing give the result better than existing method in all scenarios. It gives less number of migrations as compared to existing method in all conditions. It is achieved by future load prediction by using ANN. ANN shows 94.5555% accuracy of load prediction.

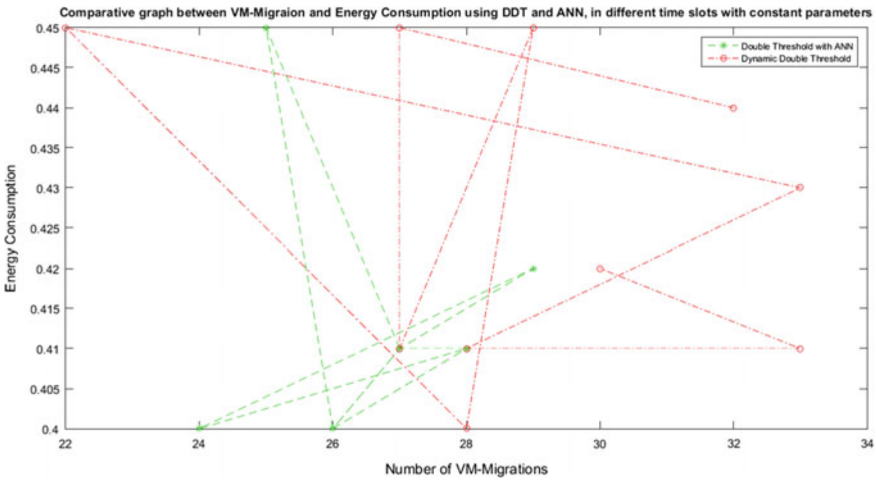
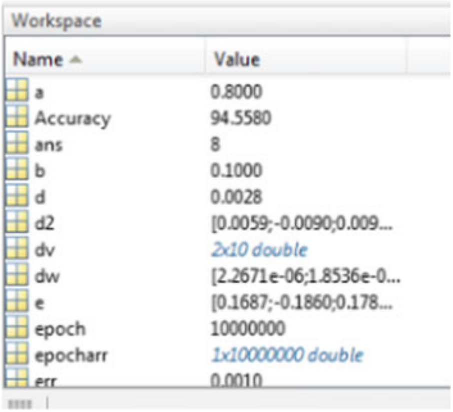


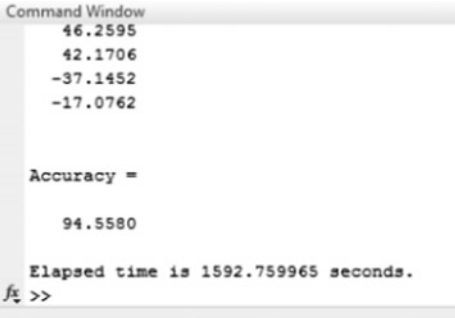
Fig. 5 Comparative graph between VM migration and energy consumption using DDT and DDT with ANN in different time slots with constant parameter

Fig. 6 Parameters related to ANN



Name	Value
a	0.8000
Accuracy	94.5580
ans	8
b	0.1000
d	0.0028
d2	[0.0059;-0.0090;0.009...
dv	2x10 double
dw	[2.2671e-06;1.8536e-0...
e	[0.1687;-0.1860;0.178...
epoch	10000000
epocharr	1x10000000 double
err	0.0010

Fig. 7 Accuracy of ANN



```
   46.2595
   42.1706
  -37.1452
  -17.0762


Accuracy =

   94.5580

Elapsed time is 1592.759965 seconds.
>>
```

5 Conclusion and Future Work

In this research work, we proposed an Energy-Aware VM Migration-based Load Balancing Approach for the Cloud Computing. Our work implements ANN with Dynamic Double Threshold algorithm for calculating VM migration and energy consumption with CPU utilization as a parameter. It not only minimizes number of VM migrations but also reduces energy consumption. The comparative graphs between VM migration and energy consumption explains that, if we integrate ANN with existing load balancing algorithms then the number of VM migrations and energy consumption reduces marginally but it saves lots of power consumption of cloud service providers, as the time pass on. The RMS value for calculating the output by ANN shows maximum accuracy rate.

In the future, we can extend our implementation from CPU utilization as a parameter to CPU, bandwidth, RAM altogether as a parameter. This will improve efficiency of load calculation which minimizes VM migration and energy consumption. Since, ANN is a learning machine and gives optimized result after each iteration, we can integrate ANN with Cloud server for continuous monitoring of load during different times slots. Consequently, by using real cloud data into ANN, we can improve the efficiency of load balancing approach in future. This approach can be integrated with other existing load balancing algorithms for the best result which any algorithm can get.

References

1. Buyya, R., et al.: Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. In: Future Generation Computer Systems, pp 599–616. Elsevier Science (2009)
2. Qaisar, E.J., et al.: Introduction to cloud computing for developers key concepts, the players and their offerings. In: Proceeding of IEEE Conference on Information Technology Professional, pp. 1–6 (2012)
3. Beloglazov, A., et al.: Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Elsevier J. Future Gener. Comput. Syst. 755–768 (2012)
4. Kapil, D., et al.: Live virtual machine migration techniques: survey and research challenges. In: IEEE 3rd International Advance Computing Conference (IACC), pp. 963–969 (2013)
5. Maziku, H., Shetty, S.: Network aware VM migration in cloud data centers. In: Third GENI Research and Educational Experiment Workshop, pp. 25–28. IEEE (2014)
6. Cerroni, W., Esposito, F.: Optimizing live migration of multiple virtual machines. IEEE Trans. Cloud Comput. (2016)
7. Gupta, Shivani, Tiwari, Damodar: Energy efficient dynamic threshold based load balancing technique in cloud computing environment. Int. J. Comput. Sci. Inf. Technol. **6**(2), 1023–1026 (2015)
8. Jain, A., et al.: A threshold band based model for automatic load balancing in cloud environment. In: IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), pp. 1–7 (2013)
9. Nehru, E.I., Venkatalakshmi, B.: Neural load prediction technique for power optimization in cloud management system. In: Proceedings of IEEE Conference on Information and Communication Technologies (2013), 541–544 pp.

10. Mandal, S.K., Khilar, P.M.: Efficient virtual machine placement for on-demand access to infrastructure resource in cloud computing. Int. J. Comput. Appl. 25–32 (2013)
11. Calheiros, R., Ranjan, R., De Rose, C.A.F., Buyya, R.: CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services, pp. 23–50 (2011)
12. Wickremasingh, B., et al.: A CloudSim-based visual modeller for analyzing cloud computing environments and applications. In: Proceedings 24th International Conference on Advanced Information Networking and Applications (AI NA) (2010)