Work Load Prediction and Energy Efficient Task Scheduling using ECSO Algorithm in Cloud Computing

Mrs.A.Nagaswathy, M.Sc.,M.Phil.,[SET][1], Dr.M.Suganya[2]

[1]Research Scholar, Rathinavel Subramaniam College of Arts and Science, Coimbatore, Tamilnadu, India

[2]Associate Professor & HoD-B.Sc(IT), School of Computer Studies(UG), Rathinavel Subramaniam College of Arts and Science, Coimbatore, Tamilnadu, India.

Abstract

In cloud computing, scheduling algorithms play an important role to schedule the user tasks. In cloud computing environment, as the number of data centers is escalating, the server and other component's energy consumption has augmented radically. The main objective of this work is to perform energy efficient task scheduling and prediction of work loads of servers. This paper proposes an work load prediction and energy efficient task scheduling (WLP-EETS) using Enhanced Crow Search Optimization (ECSO) algorithm in cloud computing. In this technique, given the arrival of user requests, the Virtual Machines (VMs) are scheduled using ECSO algorithm such that the virtual machine (VM) migration time and task completion time are minimized. The total energy consumption of the data center is minimized by changing some inactive servers into sleep state. For predicting the workloads of all tasks, the similar tasks are clustered into a group using K-means clustering and then one model is built for each task cluster. By simulation results, it will be shown that the proposed algorithm minimizes the power consumption, reduces the response delay and increases the CPU utilization.

Keywords: Energy; Cloud; Task Scheduling; CPU; Delay

1. Introduction

Cloud computing offers computing services as a “pay-as-you-go” model, which enables utility computing, such as gas, electricity, water and telephone utilities. It could be referred to as the 5th utility. Software as a Service (SaaS), Infrastructure as a Service(IaaS) and Platform as a Service (PaaS) are three commonly known cloud computing services. These services are deployed by following various deployment models. Depending on the requirements public, private, commodity and hybrid are four major cloud deployment models. Cloud computing helps users to save infrastructure and maintenance cost. Users receive scalability, reliability and mobile accessibility facilities from cloud computing services. Various companies focus on their innovation and creating business values by overlooking the low-level hardware and software infrastructure setup. The process of moving all computing services into the cloud is slowed down by various open challenges. Security, privacy, energy efficiency and common standards restrict many companies from migrating their computing services into the cloud. One of the most significant current discussions is energy saving issues. There is exceptional economic encouragement for data center operators, and it also supports greater environmental sustainability. [1][2].

Cloud computing architecture typically consists of a front end and a back end connected by Internet or Intranet. The main objective of cloud computing environment is to optimally use the available computing resources. Scheduling algorithms play an important role in optimization process. Therefore user tasks are required to schedule using efficient scheduling algorithm. The scheduling algorithms usually have the goals of spreading the load on available processors and maximizing their utilization while minimizing the total execution time. Task scheduling is one of the most famous combinatorial NP complete problems. The main purpose of scheduling is to schedule the tasks in a proper sequence in which tasks can be executed under problem specific constraints. Many heuristic optimization algorithms have been developed and solved the task scheduling in cloud environment over the years. Clonal Selection Algorithm (CSA) is a special class of Artificial Immune System which uses the clonal selection part of the Artificial Immune Systems as a main mechanism. [3].

1.1 Problem Identification and Objectives

In cloud computing environment, as the number of data centers is escalating, the server and other substructure's energy consumption has augmented radically. In CC, to achieve proper load balancing, the cloud data center dynamically migrates and deploys VM to meet user's requirements without degrading the service being delivered to the user [4].

One of the options to reduce the power consumption of data centers is to reduce the number of idle servers, or to switch idle servers into low-power sleep states. However, the servers cannot process the requests immediately when transiting to an active state.

In most of the VM allocation approaches, the resource utilization and energy consumption parameters are mainly considered for VM allocation. But they fail to consider the work load of future resource requirements. Moreover, the deadline of each task was not considered.

The main objective is to perform efficient scheduling of user tasks which predicts the future traffic loads such that the total energy consumption of the data center is minimized.

In this paper, a work load prediction and energy efficient task scheduling using ECSO algorithm in cloud computing is developed.

2. Related Works

Samah Alshathri et al [5] have introduced a novel algorithm that could allocate resources in a cloud-computing environment based on an energy optimization method called Sharing with Live Migration (SLM). In this scheduler, they used the Cloud-Sim toolkit to manage the usage of virtual machines (VMs) based on a novel algorithm that learns and predicts the similarity between the tasks, and then allocates each of them to a suitable VM. On the other hand, SLM satisfies the Quality of Services (QoS) constraints of the hosted applications by adopting a migration process.

Chonglin Gu et al [6] have considered using stateof- the-art servers with multi-sleep modes. The sleep modes with smaller transition delays usually consume more power when sleeping. Given the arrival of incoming requests, their goal is to minimize the energy consumption of a cloud data center by the scheduling of servers with multi-sleep modes. They formulated this problem as an integer linear programming (ILP) problem during the whole period of time with millions of decision variables. To solve this problem, they divided it into sub-problems with smaller periods while ensuring the feasibility and transition continuity for each sub-problem through a Backtrack-and-Update technique. They also considered using DVFS to adjust the frequency of active servers, so that the requests can be processed with the least power.

Cheikhou Thiam et al [7] have studied the problem of optimal VM allocation policy and migration to minimize power consumption in a data center while preserving QoS. CloudSim simulator is used to create a cloud environment. It provides the interface to deal with the physical and virtual machines. They evaluated and compare their algorithms corresponding to different approaches in order to find the one that optimizes VM placement and migration.

Kepi Zhang et al [8] have focused on the energy saving issue for virtual machine (VM) selections on an overloaded host in a cloud computing environment. They analyzed the energy influencing factors during a VM migration, then design energy efficient VM selection algorithms based on greedy algorithm and dynamic programming method.

Prasad Babu et al [9] have proposed an energy efficient scheduling algorithm based on prediction model used for scheduling the resources in time. Mainly, proposed algorithm has taken the leverage of prediction methods which is working based iterative fraction method. The CPU utilization is the primary factor in the part of power consumption of cloud resources. In experiments, simulation model used real traces of Google clusters. The proposed approach results presents, proposed algorithm has utilized more CPU memory than remaining algorithms which are round robin, minimum migration time and first fit.

Sanjaya K. Panda et al [10] have proposed an energy-efficient task scheduling algorithm (ETSA) to address the demerits associated with task consolidation and scheduling. The proposed algorithm ETSA takes into account the completion time and total utilization of a task on the resources, and follows a normalization procedure to make a scheduling decision. They evaluated the proposed algorithm ETSA to measure energy efficiency and makespan in the heterogeneous environment. The experimental results are compared with recent algorithms, namely random, round robin, dynamic cloud list scheduling, energy-aware task consolidation, energy-conscious task consolidation and MaxUtil. The proposed algorithm ETSA provides an elegant trade-off between energy efficiency and makespan than the existing algorithms.

Mehboob Hussain et al [11] have proposed an Energy and Performance- Efficient Task Scheduling Algorithm (EPETS) in a heterogeneous virtualized cloud to resolve the issue of energy consumption. There are two stages in the proposed algorithm: initial scheduling helps to reduce execution time and satisfy task deadlines without considering energy consumption, and the second stage task reassignment scheduling to find the best execution location within the deadline limit with less energy consumption. Moreover, to make a reasonable balance between task scheduling and energy saving, they suggested an energy-efficient task priority system.

Leila Ismail et al [12] have proposed an Energy-Aware Task Scheduling algorithm on cloud VMs (EATSVM) that assigns a task to the VM where the increase in energy consumption is the least, considering both active and idle VMs. The algorithm also takes into consideration the increase in the energy consumption of the already running tasks on the VM due to increase in their execution time, while assigning a new task to that VM.

3. Proposed Methodology

3.1 Overview

In this paper, WLP-EETS using ECSO algorithm in cloud computing is proposed. In this technique, given the arrival of user requests, the Virtual Machines (VMs) are scheduled using ECSO algorithm such that the virtual machine (VM) migration time and task completion time are minimized. Given the arrival of user requests, schedule the servers such that the total energy consumption of the data center can be minimized. Initially, among the total N servers, m servers are put to sleep mode. At a given time slot, if the number of active servers are less than the or equal to the required maximum servers, some of the servers at sleep state are put to wake up stage, otherwise, the inactive servers are continued to be in sleep state. For predicting the workloads of all tasks, the similar tasks are clustered into a group using K-means clustering and then one model is built for each task cluster.

3.2 Energy Efficient Scheduling

3.2.1 Scheduling Constraints and optimization problem

Then VM's migration time T_m is given by:

$$T_m = \frac{Size(VM_j)}{ABW_j} \quad (1)$$

The task completion time (T_c) is given by

$$T_c = T_{start} + T_{trans} + T_{exec} \quad (2)$$

Where T_{start}, T_{trans} and T_{exec} are the task starting, transmission and execution times , respectively.

Then the constraint for task completion time is given by

$$T_c < D \quad (3)$$

As stated in (6), the task completion time of a n should be less than the user specified task deadline D.

Where ABW is available bandwidth (ABW) of VM_j

Based on these constraints, the optimization problem can be formulated as below:

The objective of this optimization is to minimize the migration time and task completion time for a given task

(ie) (i) Minimize T_m

(ii) Minimize T_c

Hence a fitness function is defined as

$$F() = 1 / (w_1.T_m + w_2.T_c + w_3.\ PC^t) \quad (4)$$

Where w_1,w_2 and w_3 are weighting constants in the range of (0,1).

For achieving this solution, we apply ECSO algorithm, as described in the following section.

3.2.2 Basic principles of ECSO

Crow search algorithm (CSA) is one of the newest population based evolutionary algorithms introduced based on crows behavior in finding their food. A crow checks the places while others birds hide their foods and the crow tries to steal them when the owner leaves its place.

In CSA, the position of each crow (its hiding places) represents a possible solution of the optimization problem. It is assumed that Nc is the number of crows in the flock (population size) and the itrmax is the maximum number of iteration of the algorithm. For a d-dimensional search space, the position vector of ith crow is specified by

$$X^{i,ier} = [X_1^{i,iter}, X_2^{i,iter}, ... X_d^{i,iter}]$$

where itr = 1,2, … , $itermax$ and i = 1,2, … , Nc and d is the number of decision variables.

The memory of the crow i which is shown by $X^{i,ier} = [X_1^{i,iter}, X_2^{i,iter}, ... X_d^{i,iter}]$ represents the best experience of the crow i.

Crows move in the environment and search for the better food sources (hiding places). In this method, at each iteration for the crow *i, K* of the best crows in the population are selected, then the crow i selects one of these top crows j randomly to following it.

For this purpose, it is assumed that in the iteration itr, the crow i decides to follow the crow j (a random member in the flock) to access the best hiding place of crow j ($M^{j,itr}$). In this case, there are two possibilities.

Iin the beginning of iterations, the value of K starts from large number and its value is reduced according to (5) and at the final iterations the K has the small value.

$$K^{itr} = round\left(k_{\max} - \frac{K_{\max} - K_{\min}}{itr\max} Xitr \right) \qquad (5)$$

Then the next position of the crow i is expressed by (8).

$$X^{i,iter+1} = \begin{cases} X^{i,iter} + r_i x\, fl^{i,itr} x(M^{j,itr} - X^{j,itr}) & r_j \geq AP^{j,itr} \\ arandomposition & otherwise \end{cases} \qquad (6)$$

Where, r_i and r_j are the uniform distributed random numbers in the interval (0,1) and; $fl^{i,itr}$ and $AP^{i,itr}$ are the flight length (fl) of the crow i and awareness probability (AP) of the crow j at the iteration itr, respectively.

We need to define a threshold for the distance among $X^{i,}$ and $M^{j,}$ in order to choose best value of *fl.* In this regard, Eq.(7) is defined to the determination of flight length in leads to select the suitable value of *fl* with respect to the situations of the crows.

$$fl^{i,itr} = \begin{cases} 2 & if\ D^{i,j} > D_{thr} \\ fl_{thr} & if\ D^{i,j} \leq D_{thr} \end{cases} \qquad (7)$$

Where, $D^{i,}$ is the vector which contains the distances between the crow i and crow j ($M^{j,} - X^{i,r}$), D_{thr} is the distance threshold value and fl_{thr} is a number greater than 2.

The crow i uses (11) to update its memory.

$$M^{i,iter+1} = \begin{cases} X^{i,iter} & f(X^{i,iter+1})\ isbetterthanf(M^{i,iter}) \\ M^{i,itr} & otherwise \end{cases} \qquad (8)$$

In (8), f(.) represents the amount fitness function, as defined by Eq.(4)

3.2.3 ECSO Algorithm

The procedure of the CSA optimizing is as the following steps.

1. Initialize the current position of each crow in the flock and setting the initial memory of crows to their initial position

2. For each crow, evaluate the position quality by inserting of the crow in the objective function.

3. Generate the new position of crows.

4. Check the feasibility of new positions. If the new position of each crow is not feasible, then the crow stays at its current position and does not move to the new position.

5. Evaluate the fitness function of the new positions for each crow.

6. Update the memory of crows using (7)

7. For each iteration, steps 3-6 are repeated until the number of iteration reaches to the *itrmax,* then the best memory position is selected as the solution.

3.3 Constraints for Server Power Optimization

The power consumption of a server is the sum of the idle power and the current frequency level, which can be represented as follows:

$$PC_{server} = PC_{idle} + PC_f(l) \quad (9)$$

where PC_{idle} denotes the idle power, $PC_f(l)$ denotes the dynamic power at frequency level l, respectively.

Let $N_0^t(l)$ denote the number of servers at frequency level *l* in timeslot *t*, the total power consumption of the active servers in timeslot t can be represented as follows:

$$PC_{active}^t = \sum_{l=1}^{L}[N_0^t(l).(PC_{idle} + PC_f(l))] \quad (10)$$

Let N_k^t be the number of servers at sleep state. The total power consumption of the sleeping servers in timeslot *t* can be calculated as follows:

$$PC_{sleep}^t = \sum_{k=1}^{K}(PC_k \cdot N_k^t) \quad (11)$$

Where PC_k denotes the sleep power

Thus, the total power consumption of the servers in different states in timeslot t can be calculated as:

$$PC^t = PC_{active}^t + PC_{sleep}^t \quad (12)$$

Let $R_j(t)$, j=1,2.... be the user requests arrived at time t.

Then the requests have to be scheduled to servers which are at sleep state at various intervals such that the total power consumption PC^t is minimized.

It involves the following phases:

1) Changing V servers from active mode to sleep mode so that {No – V} becomes minimum

2) Adjusting the frequency of intervals (f) of the servers so that all requests are scheduled with PC_t minimized.

Algorithm: Minimizing Total Server Power

Notations	Definition
T	Length of time Interval
τ	Number of time slots
$R_j(t)$,j=1,2....	User request arrival at slot t, $t \in (1,2...\tau)$
i	Current slot
(i+1)	Next slot
N_server	Number of servers
$Server_{max}$	Maximum no. of servers
$Server_{active}$	Servers at active mode
$Server_{sleep}$	Servers at sleep mode
$Server_{min\_sleep}$	Minimum number servers at sleep mode
PC_{max}	Maximum limit for power consumption

Initially, $Server_{max} = Server_{active}$
For each $t \in (1,2...\tau)$
For each $R_j(t)$
  If $Server_{active} >= Server_{max}$ then
      $Server_{sleep} = N\_server - Server_{active}$
  Else
      $Server_{sleep} = Server_{sleep} - Server_{min\_sleep}$
      For each $N_k \in Server_{min\_sleep}$
            Change N_K to active mode
            If ($PC^t < PC_{max}$)
                $Server_{active} = Server_{active} + 1$
            Else
                Keep N_k in sleep mode
            End if

15. End For
16. End if
17. End For

The total time interval T is divided into τ number of slots. Let Rj(t) be the user request arrived at time slot t and $Server_{max}$ be the maximum number of servers required to fulfil all the user requests arrived at time slot t.

In this algorithm, initially the number of active servers are kept at the maximum level.

At time slot t, if the number of active servers are greater than or equal to the required maximum servers, the algorithm does nothing . (ie) all the remaining servers are continued to be in sleep state.

On the other hand, if the number of active servers are less than the or equal to the required maximum servers, some of the servers at sleep state should be wake up. Let $Server_{min\_sleep}$ be the number of sleep servers randomly selected for waking up. After changing each sleep server to active mode, the total power consumption (estimated in Eq. (8)) is checked. If it does not reaches the maximum power level PC_{max}, then the number of active servers is incremented and the number of sleep servers is decremented. If the total power consumption reaches the maximum level PC_{max}, then again the server is changed to sleep mode or continue to be in sleep mode. This process continues until the number of active servers reaches the maximum level $Server_{max}$ required or all the servers in $Server_{min\_sleep}$ are checked.

3.4 Work Load prediction using ECSO

Different tasks have different workload features. We build one model that is used for predicting the workloads of all tasks, based on the similarity of tasks. For this, the similar tasks are clustered into a group, and then one model is built for each task cluster.

For this the distance based clustering algorithm is used where the shortest distance between each task to the center is determined. K-means is a distance based iterative algorithm. It clusters the whole data points instances into N clusters so that each observation instance is smaller than the center point of the current cluster, compared to other cluster's center point.

The tasks defined by the pair <CPU size, memory size> will be divided into N clusters such that the total distance between each task to the center is shortest, in each cluster.

4. Experimental Results

The proposed WLP-EETSA is implemented in Cloudsim and compared with the task scheduling using Clonal Selection Algorithm (TSCSA) [2]. The NASA workload has been used as the emulator of Web users requests to the Access Point (AP). This workload represents realistic load deviations over a period time. It comprises 100960 user requests sent to the Web servers during a day.

4.1 Based on Active servers

In the first experiment we vary the number of active servers from 2 to 12.

Active servers	WLP-EETSA (KW/h)	TSCSA (KW/h)
2	0.71	0.82
4	0.67	0.8
6	0.65	0.77
8	0.62	0.76
10	0.61	0.72
12	0.55	0.68

Table 1: Result table for Power Consumption

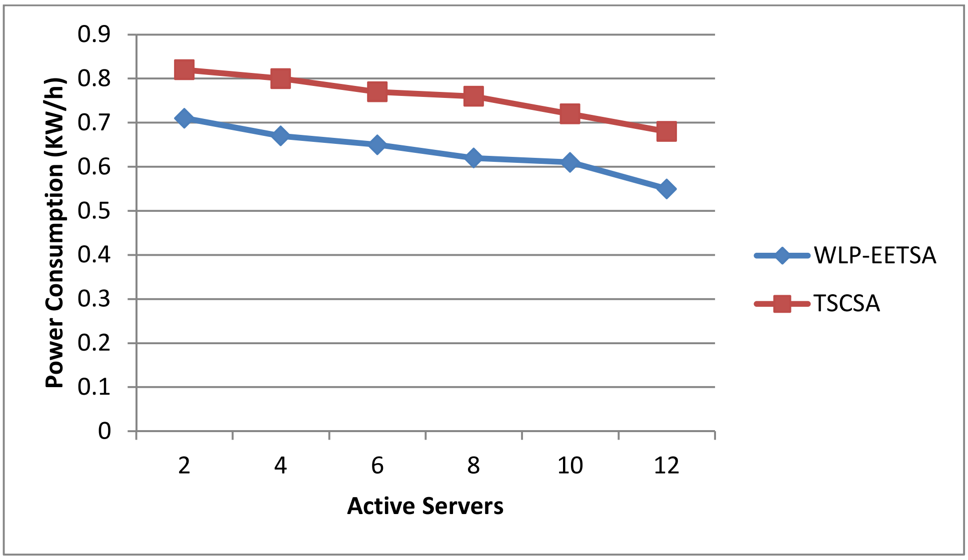

Figure 1 Power Consumption Vs Active servers

From the Figure 1, we can observe that the power consumption of our proposed algorithm EETSA achieves 4% lesser power consumption than WLP-TSCSA for different number of VMs scenario.

Active servers	WLP-EETSA	TSCSA
2	640	734
4	481	578

6	331	421
8	248	363
10	181	287
12	125	235

Table 2: Result table for No of VM Migrations

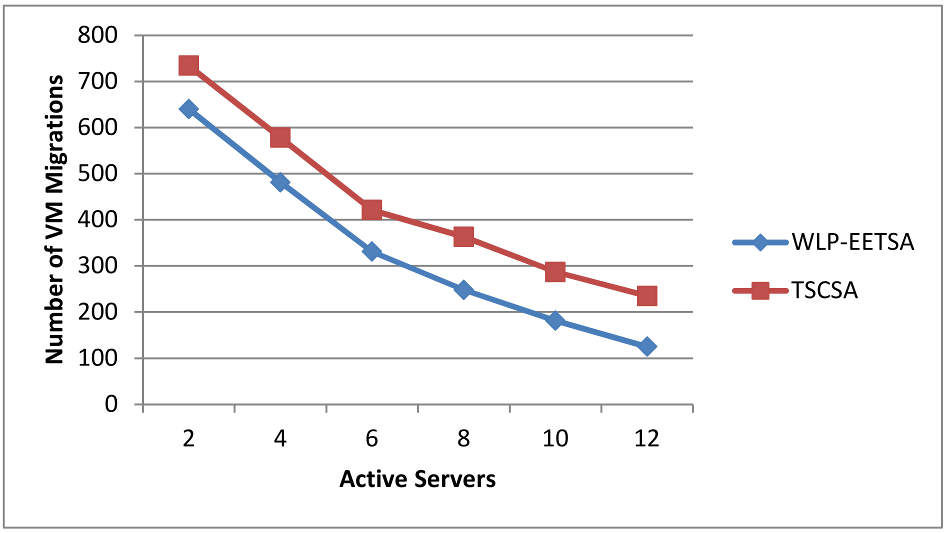

Figure 2 No of VM Migrations Vs Active servers

From the figure 2, we can observe that the No of VM Migrations of our proposed algorithm WLP-EETSA achieves 28% lesser No of VM Migrations than TSCSA for different number of VMs scenario.

Active servers	WLP-EETSA (%)	TSCSA (%)
2	70.55	65.28
4	71.33	67.73
6	72.52	70.36
8	73.11	71.66
10	76.50	73.11
12	78.00	75.25

Table 3: Result table for CPU Utilization

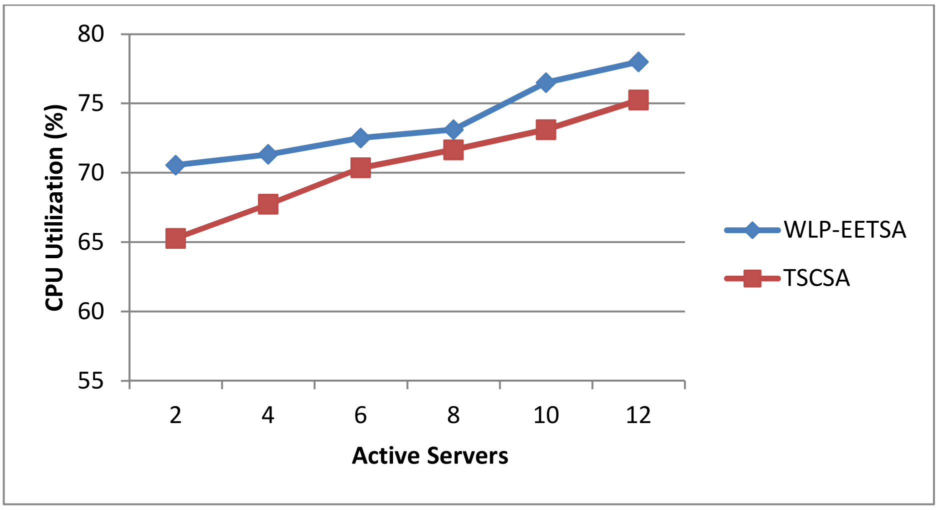

Figure 3 CPU Utilization Vs Active servers

From the figure 3, we can observe that the CPU Utilization of our proposed algorithm WLP-EETSA achieves 4% higher CPU Utilization than TSCSA for different number of VMs scenario.

Active servers	WLP-EETSA (sec)	TSCSA (sec)
2	1.25	2.58
4	1.78	2.94
6	2.25	3.65
8	2.66	4.15
10	4.34	5.25
12	4.92	5.82

Table 4: Result table for Response Delay

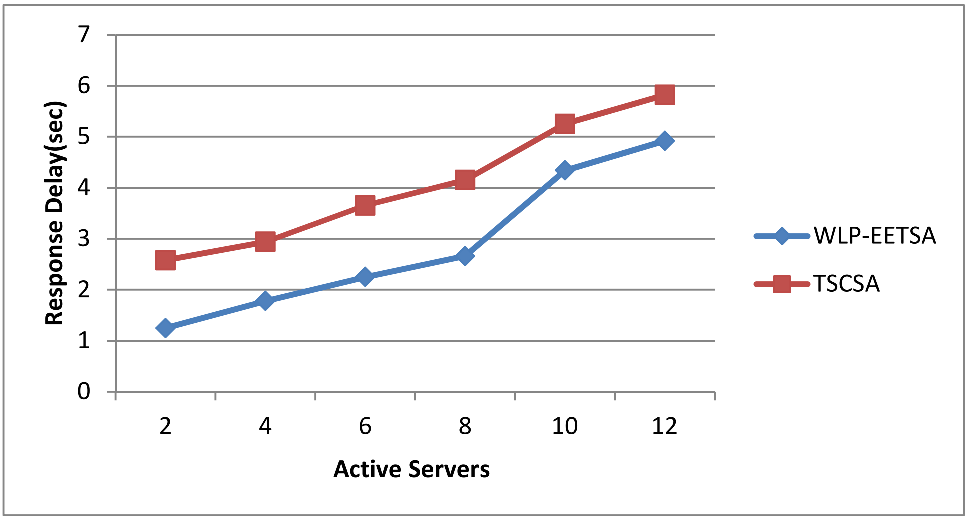

Figure 4 Response Delay Vs Active Servers

From the figure 4, we can observe that the Response Delay of our proposed algorithm WLP-EETSA achieves 33% lesser Response Delay than TSCSA for different number of VMs scenario.

4.2 Based on VMs

In the second experiment we vary the number of VMs from 10 to 50

No of VMs	WLP-EETSA (KW/h)	TSCSA (KW/h)
10	0.71	0.82
20	0.73	0.84
30	0.75	0.87
40	0.79	0.89
50	0.81	0.92

Table 5: Result table for Power Consumption

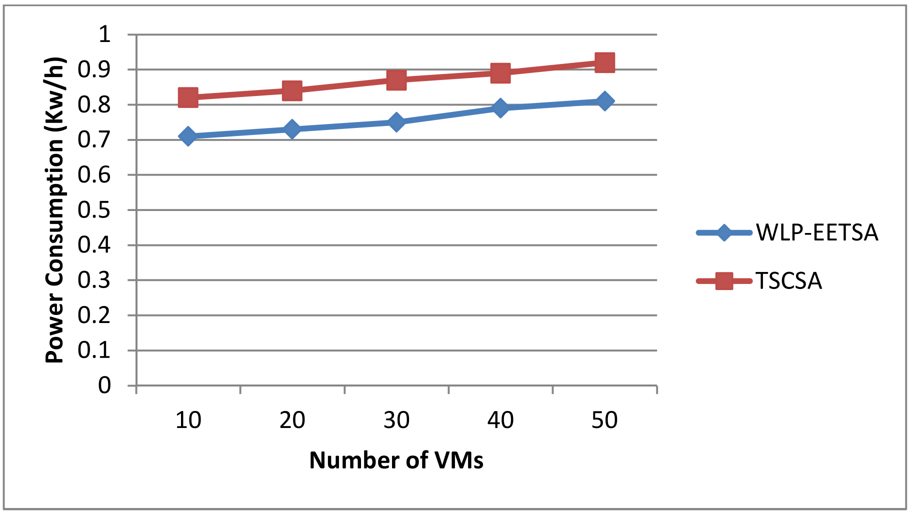

Figure 5 Power Consumption Vs No of VMs

From the figure 5, we can observe that the power consumption of our proposed algorithm WLP-EETSA achieves 13% lesser power consumption than TSCSA for different number of VMs scenario.

No of VMs	WLP-EETSA (%)	TSCSA 9%)
10	92.1	88.1
20	88.7	77.4
30	76.2	72.7
40	74.4	66.5
50	72.1	65.8

Table 6: Result table for CPU Utilization

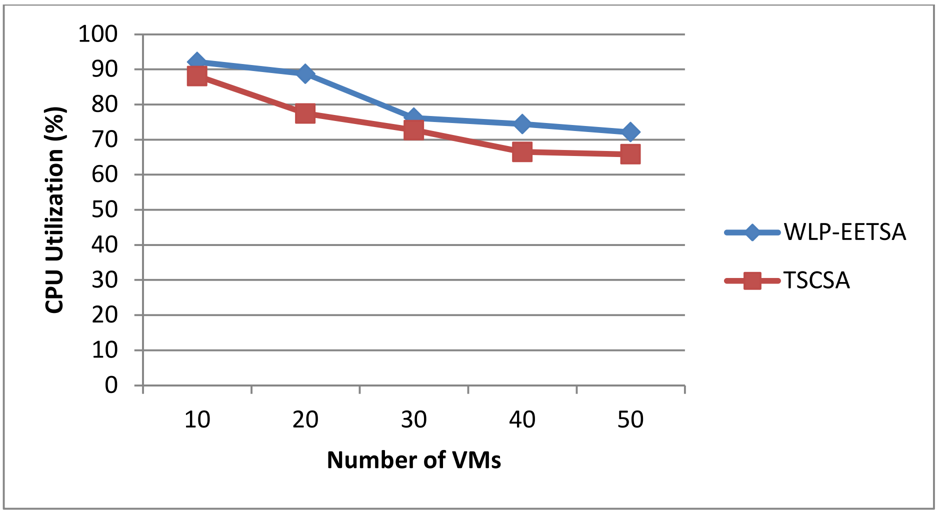

Figure 6 CPU Utilization Vs No of VMs

From the figure 6, we can observe that the cpu utilization of our proposed algorithm WLP-EETSA achieves 8% higher cpu utilization than TSCSA for different number of VMs scenario.

No of VMs	WLP-EETSA (sec)	TSCSA (sec)
10	1.32	0.79
20	1.73	1.17
30	2.04	1.45
40	2.31	1.81
50	2.79	2.55

Table 7: Result table for Response Delay

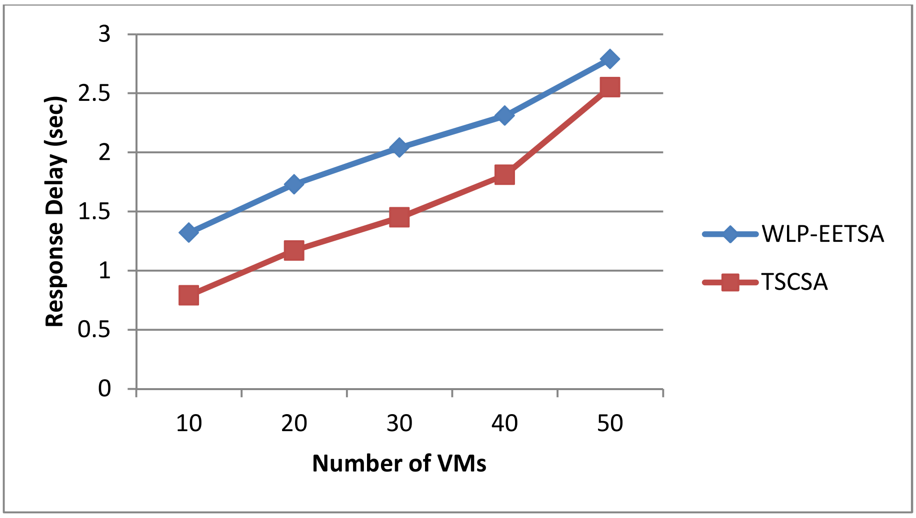

Figure 7 Response Delay Vs No of VMs

From the figure 7, we can observe that the response delay of our proposed algorithm WLP-EETSA achieves 26% lesser response delay than TSCSA for different number of VMs scenario.

No of VMs	WLP-EETSA	TSCSA
10	275	367
20	332	528
30	443	646
40	523	788
50	648	828

Table 8: Result table for Number of VM Migrations

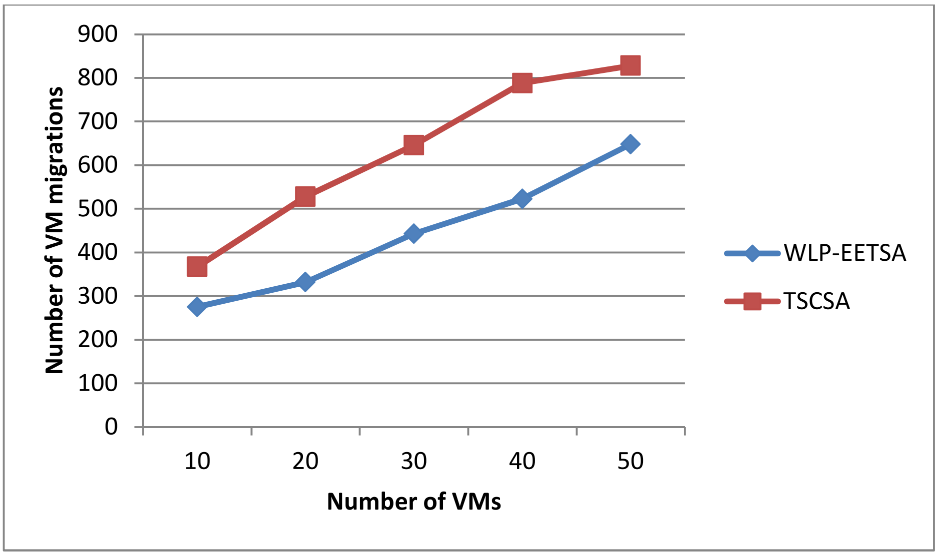

Figure 8 Number of VM migrations Vs No of VMs

From the figure 8, we can observe that the number of VM Migrations of our proposed algorithm WLP-EETSA achieves 13% lesser number of VM migrations than TSCSA for different number of VMs scenario.

5. Conclusion

This paper proposes WLP-EETS using ECSO algorithm in cloud computing. In this technique, given the arrival of user requests, the Virtual Machines (VMs) are scheduled using ECSO algorithm such that the virtual machine (VM) migration time and task completion time are minimized. The total energy consumption of the data center is minimized by changing some inactive servers into sleep state. For predicting the workloads of all tasks, the similar tasks are clustered into a group using K-means clustering and then one model is built for each task cluster. The proposed WLP-EETSA is implemented in Cloudsim and compared with the TSCSA. By simulation results, it has been shown that the proposed WLP-EETSA minimizes the power consumption, minimizes the number of VM migrations, reduces the response delay and increases the CPU utilization, when compared to TSCSA.

References

1.Nasrin Akhter, Mohamed Othman and Ranesh Kumar Naha, "Evaluation of Energy-efficient VM Consolidation for Cloud Based Data Center - Revisited", ArXiv, 2018.

2.R.K Jena, "Energy Efficient Task Scheduling in Cloud Environment", Elsevier, Energy Procedia, Vol-141, 2017.

3.Weiwen Zhang, Yonggang Wen, and Dapeng Oliver Wu, "Energy-efficient Scheduling Policy for Collaborative Execution in Mobile Cloud Computing", Proceedings IEEE INFOCOM, 2013.

4.Guangyu Du, Hong He and Qinggang Meng, "Energy-Efficient Scheduling for Tasks with Deadline in Virtualized Environments", Hindawi Publishing Corporation Mathematical Problems in Engineering, Volume 2014, Article ID 496843, 7 pages, 2014.

5.Samah Alshathri, Bogdan Ghita and Nathan Clarke, "Sharing with Live Migration Energy Optimization Scheduler for Cloud Computing Data Centers", Future Internet, Vol-10, 2018.

6.Chonglin Gu, Zhenlong Li, Hejiao Huang and Xiaohua Jia, "Energy Efficient Scheduling of Servers with Multi-Sleep Modes for Cloud Data Center", IEEE, 2017.

7.Cheikhou Thiam and Fatoumata Thiam, "An Energy-Efficient VM migrations optimization in Cloud Data Centers", IEEE, 2020.

8.Kepi Zhang, Tong Wu, Siyuan Chen, Linsen Cai, Chao Peng, "A New Energy Efficient VM Scheduling Algorithm for Cloud Computing based on Dynamic Programming", IEEE 4th International Conference on Cyber Security and Cloud Computing, 2017.

9.G. Prasad Babu and A. K. Tiwari, "Energy Efficient Scheduling Algorithm for Cloud Computing Systems Based on Prediction Model", Int. J. Advanced Networking and Applications, Volume: 10 Issue: 05 Pages: 4013-4018, 2019.

10.Sanjaya K. Panda and Prasanta K. Jana, "An energy-efficient task scheduling algorithm for heterogeneous cloud computing systems", Springer,Cluster Computing, 2018.

11.Mehboob Hussain, Lian-Fu Wei, Abdullah Lakhan, Samad Wali, Soragga Ali and Abid Hussain, "Energy and performance-efficient task scheduling in heterogeneous virtualized cloud computing", Elsevier, Sustainable Computing: Informatics and Systems, Vol-30, 2021.

12. Leila Ismail and Huned Materwala, "EATSVM: Energy-Aware Task Scheduling on Cloud Virtual Machines" Elsevier, 3rd International Conference on Computer Science and Computational Intelligence, 2018.