Cost Effective and Energy Efficient VM Migration for Scheduling in Cloud Computing

T. Ramesh (ramesh09phd@gmail.com)
R. Rajagopal
Narsimha Reddy Engineering College (Autonomous)
M. Umaselvi
P. John Paul

Research Article

Keywords: Cloud computing, CEEM-VM-MS, Utilization, Cloud-sim, Energy

Posted Date: May 10th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1600709/v1

Cost Effective and Energy Efficient VM Migration for Scheduling in Cloud Computing

*Dr.T.Ramesh,

Associate Professor,RMK Engg College,Chennai.

Corresponding Author: *ramesh09phd@gmail.com

Dr.R.Rajagopal

Associate Professor, Dept. of CSE,Narsimha Reddy Engineering College (Autonomous), Hyderabad, India.

rajagopalrmail@gmail.com

Dr. M. Umaselvi

Associate Professor,Dept. of CSE, P.A College of Engineering and Technology, Coimbatore.

mumaselvi@gmail.com

Dr. P. John Paul

Principal and Professor in Department of ECE, Ellenki College of Engineering and Technology,

jppulipati@yahoo.com

Abstract: In cloud computing, 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. Several migration techniques are available for migrating VM from one host to another. But they fail to consider the migration cost while determining the energy consumption during migration. 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 this paper, a cost effective and energy efficient VM migration for scheduling in cloud computing is developed. In this technique, VMs having minimum cost and minimized energy consumption are selected for migration. During load balancing, the tasks are migrated to the underutilized server if the needed current server is overloaded. Similarly, the tasks are migrated from the under-loaded server to the current server. In energy efficient scheduling, given the arrival of user requests, the VMs are scheduled such that the total energy consumption of the data center is minimized. The proposed CEEM-VM-MS technique is implemented in Cloudusim tool. In experiments, the upper threshold value for CPU size is increased and the performance metrics power consumption, number of VM migrations, response delay, and CPU utilization are measured. Results indicate that the proposed CEEM-VM-MS technique minimizes the power consumption, reduces the number of VM migrations, reduced the response delay and increases the CPU utilization.

Keywords: Cloud computing; CEEM-VM-MS; Utilization; Cloud-sim; Energy

1. Introduction

The "pay-as-you-go" paradigm of cloud computing allows for utility computing, such as gas, electricity, water, and telephone. It might be referred to as the fifth function. A cloud computing service is defined as a service that provides access to the software, hardware, and other resources needed to run the service. Several different deployment strategies are used to make these services available to users. The four primary cloud deployment methods are public, private, commodity, and hybrid. Infrastructure and maintenance costs are reduced by cloud computing. Cloud computing services provide scalability, dependability, and mobility to its customers. When it comes to innovation and business value, many firms disregard the low-level hardware and software infrastructure configuration. Several unsolved problems are delaying the migration of all computer services to the cloud. Many businesses are unable to move to the cloud because of concerns about data security, privacy, and efficiency, as well as a lack of common standards. Energy conservation is one of the most heavily debated topics in today's society. For data centre operators, there is a considerable deal of financial incentive and environmental assistance. [1].

Server virtualization in the datacenter allows for better resource efficiency, easier resource management, and less time spent on server maintenance. Moreover, virtualization improves the stability and availability of the system. This programme creates virtual machines (VMs) that share the underlying hardware resources and operate independently of each other. The ability to move VMs across servers in a datacenter may increase the usage of resources and reduce energy consumption. Most current CPUs include a function called DVFS, which allows them to adjust their voltage and frequency settings dynamically in response to the workload. With no load, a server consumes around 66 percent of its maximum energy consumption while in standby mode, The server's hardware and software both need a certain amount of electricity to function properly. [2].

An Internet or Intranet connection is often used to connect the front end of a cloud computing system to its back end. The primary goal of a cloud computing environment is to make best use of the computer resources that are readily accessible. An essential part of the optimization process is scheduling algorithms. As a result, scheduling user tasks using an efficient algorithm is a need. Scheduling algorithms typically aim to disperse the workload among the available processors and maximise their utilisation while reducing the overall execution time to the minimum possible. Task scheduling is one of the most well-known combinatorial NP complete issue challenges. The primary goal of task scheduling is to arrange tasks in such a way that they may be completed within the limits of a given situation. Over the years, a slew of heuristic optimization techniques have been created to help with cloud-based job scheduling. One kind of Artificial Immune System is the Clonal Selection Algorithm (CSA), which is based on the clonal selection element of the AI systems. [3].

1.1 Problem Identification and Objectives

Migration of a virtual machine (VM) from one physical host to another is known as VM migration. For appropriate load balancing in cloud computing, cloud data centres automatically relocate and install virtual machines (VM) to suit the user's needs without affecting user experience.

Migrating VMs from overloaded servers to prevent performance degradation and from underloaded servers to optimize resource usage and cut down on energy consumption are the two primary components of VM migration.

The main issues of VM migration are:

(1) **Selection of VMs for migration**. Choosing which virtual machines to move from one server to another is an essential step after the choice to transfer VMs has been taken. The challenge is to figure out which VMs should be moved to which servers in order to get the most out of the system reconfiguration.

(2) **Time and position for migration:**Another important component of VM consolidation and energy usage is determining the optimum area to install new VMs or the VMs chosen for transfer to other servers.

VMs may be moved from one host to another using a variety of migration methods. Migration costs aren't included when assessing how much energy is used during migration.

One way to lower the power consumption of data centres is to reduce the number of idle servers or move idle servers into low-power sleep modes. When a server enters an active state, it may begin processing requests instantly.[14]

Achieving lower power usage without sacrificing QoS is the major objective here. Therefore, it is necessary to carry out an energy efficient VM migration and schedule.

In this paper, a cost effective and energy efficient VM migration for scheduling in cloud computing is developed.

2. Related Works

2.1 Energy efficient VM Migration

Using computer simulation, Nasrin Akhter et al [1] were able to uncover fresh insights into current energy-efficient techniques. Using a different statistical technique might help us save energy in cloud-based data centres, according to their study. Environmental concerns are increasingly being taken into account by cloud users. The fast growth of data centres' energy use has resulted in an increase in CO2 emissions in the economy, which is a key source of energy consumption. Energy regulations aimed at reducing greenhouse gas emissions are a major issue for many governments across the globe.

In order to address critical issues affecting the datacenter servers when migrating VMs, Mohammad H. Alshayeji et al [2] have introduced an Energy Efficient Virtual Machine Migration (EVM) approach. In order to pick the victim and target servers, they propose to use an EVM approach based on Energy Based Server Selection (ESS). Compared to other techniques like arbitrary server selection (ASS) and first fit strategy (FFS), their findings reveal that EVM reduces the cost by reducing the amount of server state changes, virtual machine migration, and oscillations (FFS).

In order to anticipate future resource utilisation, Monireh H. Sayadnavard et al [4] developed a discrete-time Markov chain (DTMC) model. It is more accurate to categorise PMs based on their state when the DTMC model is combined with the dependability model of PMs. An algorithm that balances energy usage, resource waste, and the system's resilience to fulfil service level agreements (SLAs) and quality of service (QoS) needs is then presented to achieve the best possible VMs to PMs mapping using the e-MOABC algorithm. They used the CloudSim toolbox to conduct a performance assessment of their suggested solution and found it to be beneficial.

For the VMP issue, Elsedimy et al. [5] have suggested VMPMOPSO, a PSO-based multi-objective algorithm that works well. VMPMOPSO uses the crowding entropy approach to increase VMP diversity and expedite convergence to the optimum solution while also improving the variety of the solutions produced. There were three algorithms tested: First Fit-Decreasing (FFD), two multiobjective ant colony and genetic algorithms, and one basic algorithm dubbed First Fit-Decreasing (FFD). It was determined that the suggested VMPMOPSO was both effective and efficient by means of two simulation trials.

Double Threshold Migration (DTM) is a new technique suggested by Djouhra Dad et al. [6] that considers both an upper and a lower threshold of CPU use. These Thresholds enable the migration of a specific subset of VMs. In order to lower the high server usage and turn off the unused physical machines, live migration of VMs was implemented (PMs). The work uses a modified version of the Best Fit Decreasing (MBFD) method to fix the VM placement issue.

When it comes to reducing data centre power usage while maintaining quality of service, researchers Cheikhou Thiam et al [7] have looked at the issue. Creating a cloud environment using the CloudSim simulator is possible. Interfaces with both real and virtual machines are provided. For the purpose of finding the best algorithm for VM placement and migration, they examined and compared their algorithms based on several methodologies.

Decentralized virtual machine migration (EDVMM) has been suggested by Jayamala et al [8] to minimise energy consumption, eliminate service-level agreement violations, and increase resource usage in cloud data centres. Virtual machines are selected using an advanced decentralised technique, and prediction is utilised to identify which VMs are suited for a host. The EDVM algorithm uses virtualization technologies to move virtual machines between hosts that are overloaded and underloaded (PMs).

2.2 Energy efficient Scheduling

Jennifer [3] has concentrated on energy and processing time optimization with the application of the TSCSA (Task Scheduling with Clonal Selection Algorithm). An open source cloud platform was used to imitate the results of TSCSA (CloudSim). It was observed that TSCSA gave the best balance of outcomes for numerous goals when it was compared with an existing scheduling algorithm.

The authors of Guangyu Du et al [9] have developed a new scheduling method for virtualized settings with heterogeneous virtual machines in order to save energy and ensure that all jobs are completed on time. In order to test the new scheduling technique, we've used the CloudSim toolkit. In terms of energy usage, this solution exceeds earlier scheduling systems by a wide margin.

Sharing with Live Migration, developed by Samah Alshathri et al. [10], is a new technique for allocating resources in a cloud computing environment based on energy optimization (SLM). A unique algorithm that learns and predicts the similarity between jobs is utilised in this scheduler, which uses the Cloud-Sim toolkit to govern the utilisation of virtual machines (VMs). SLM, on the other hand, uses a migration procedure to meet the hosted applications' Quality of Services (QoS) criteria.

Multi-sleep mode servers have been explored by Chonglin Gu et al [11]. The sleep modes with the shortest transition times are more power-hungry while you're asleep. Cloud data centres' energy usage is reduced by arranging servers in multi-sleep modes when there are many incoming requests. Integer linear programming (ILP) was used to solve this issue with millions of decision variables across the whole time span. The solution they came up with was a Backtrack-and-Update approach, which broke the issue down into smaller sub-problems and ensured that each sub-problem was feasible and transitioned smoothly. Consider utilising DVFS to modify the frequency of active servers, so that requests may be handled at a lower power consumption.

A study by Cheikhou Thiam et al. [12] investigated how to optimise data centre power usage while maintaining quality of service (QoS) by using an efficient VM allocation strategy and migration. Creating a cloud environment using the CloudSim simulator is possible. Interfaces with both real and virtual machines are provided. Their methods were analysed and compared in order to discover the most efficient method for VM placement and migration.

In a cloud computing setting, Kepi Zhang et al [13] have concentrated on the energy-saving problem of VM choices on an overloaded host. Based on this analysis, they devised an energy-efficient VM selection process using a greedy algorithm and dynamic programming approach to identify the most energy-efficient VMs.

Collaborative execution in mobile cloud computing has been studied by Weiwen Zhang et al [14]. An app for mobile devices is made up of a series of small tasks that create a linear topology. These tasks may be completed locally on the mobile device or transferred to the cloud for processing. The goal of the design is to keep the mobile device's power consumption to a minimum while yet achieving the deadline. "LARAC" technique was used to approximate this issue's solution, which is a constrained-shortest route problem on a directed acyclic network.

3. Cost Effective and Energy Efficient VM Migration for Scheduling (CCEM-VM-MS)

3.1 Overview

In this paper, a cost effective and energy efficient VM migration for scheduling in cloud computing developed. Figure 1 shows the block diagram of the CCEM-VM-MS technique.

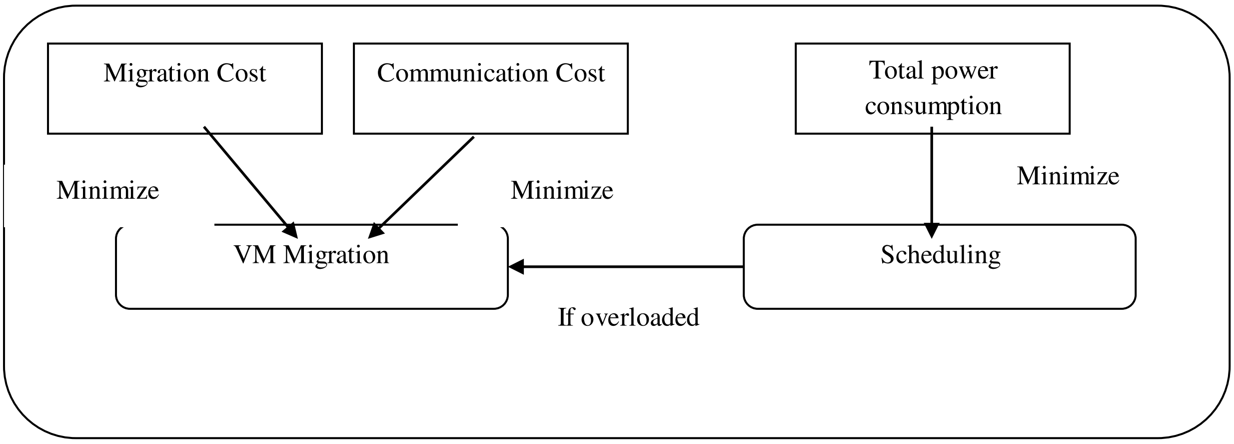

Figure 1 Block diagram of CCEM-VM-MS

In this technique, VMs selected for migration should have minimum cost and minimized energy consumption. The total cost of a data file F is given by the sum of migration cost and communication cost of each data block. The migration cost is derived in terms of migration distance and size of data file. The communication cost is derived in terms of distance between source server and host server.

Migrate the tasks to the replica underutilized server if the needed current server is overloaded. Similarly, migrate tasks from the under-loaded replica server to the current server, then set the replica back to sleep mode.

In energy efficient scheduling, given the arrival of user requests, schedule the servers such that the total energy consumption of the data center can be minimized.

3.2 System Model

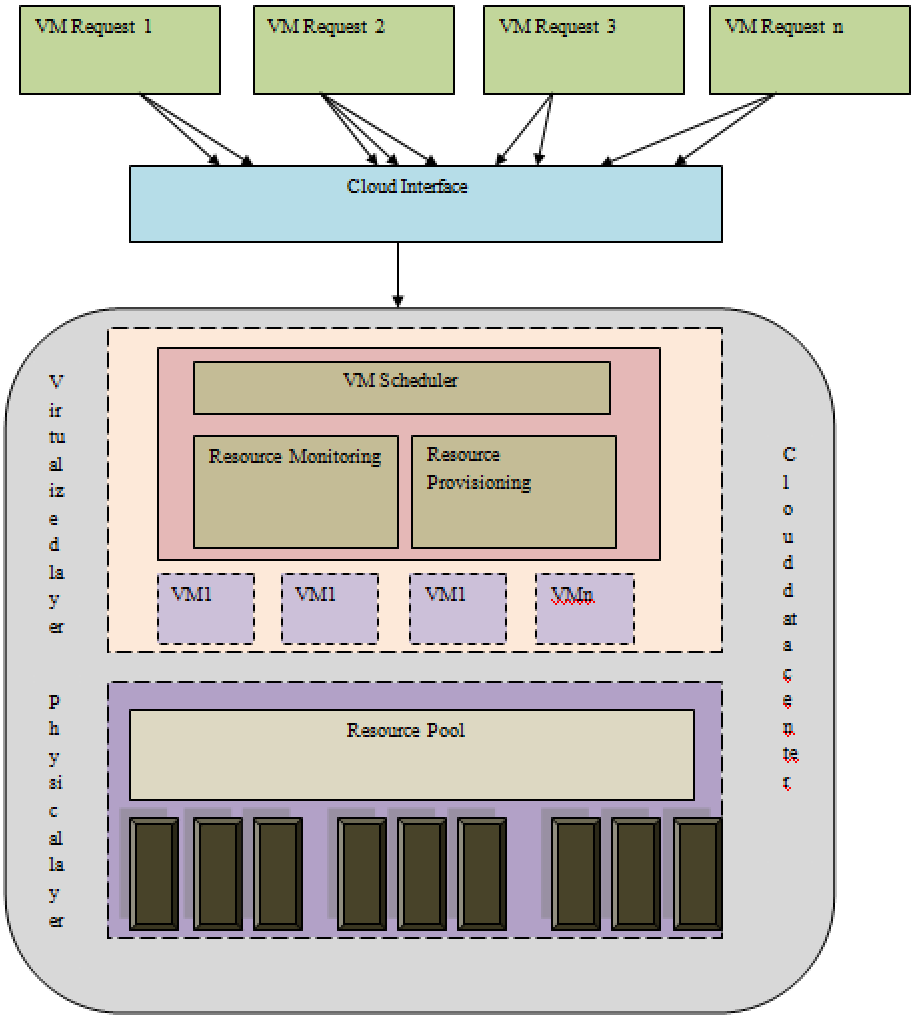

Figure 2 The system architecture for VM management in the cloud environment.

3.3 Categorization of Servers

This step of VM consolidation process is performed by local agents distributed on the PMs. Each local agent determines PM status using historical data and two distinct Markov chains. In this regard, a DTMC model as an efficient tool for analysis of complex systems in uncertain environments is used to forecast the load status of the PMs. Furthermore, a CTMC model is used to predict PMs reliability. Formal definitions of DTMC and CTMC are as follows:

3.3.1 DTMC-Based host load status prediction

The workload of a project manager is determined by the number of VMs given to him or her over a period of time. PM workloads may be determined by a variety of parameters, including CPU consumption, memory use, and storage. CPU use, on the other hand, is often the most important factor. In this case, we have assumed that the workload is fixed and constant. Using a discrete time Markov chain, the data gathered on CPU usage of the hosts may be simulated. Indeed, the probability of transitioning from one state to another is time-independent, therefore it may be represented as a DTMC that is time-independent. There's also a general assumption in the literature that an underload PM does not immediately transition into overload mode. Model state transition diagram for each PM's probabilistic load status in the data centre. Underload, Overload and Well-utilized are the three states illustrated in the model, which is shown in the figure. So the discrete state space is defined as $S = \{U, O, W\}$. The matrix P determines, at each time step, the chance of moving from one state to another. When a transition begins, the rows represent the initial state and the columns represent the final state, respectively. Hence, pij is the probability of transitioning from current condition I to state j at time *n+1*.

$$P = \begin{bmatrix} p_{uu} & p_{uw} & p_{uo} \\ p_{wu} & p_{ww} & p_{wo} \\ p_{ou} & p_{ow} & P_{oo} \end{bmatrix} \tag{1}$$

To construct the matrix of transition probabilities the maximum likelihood estimation method is applied. Using the state definition and historical data the probability of transition can be estimated as:

$$\hat{p}_{ij} = \frac{C_{ij}}{\sum_{k \in S} C_{ik}} \tag{2}$$

Where C_{ij} is the transition counts and defines the number of times state i is followed by j.

3.3.2 CTMC-Based host reliability prediction

In this paper, we have used CTMC model that presented in our previous work to analysis PMs reliability. Since, exponential random variable is the only continues random variable with Markov property and hardware and software fault are commonly modelled as exponential distribution, it is assumed that the time to transit from a system state to another due to failures and recovery follows an exponential distribution.

The findings of the previous step are sent to the global agent by local agents. The global agent classifies the PMs according to their projected state based on the information it has received. The pseudocode for the PMs categorisation algorithm is shown in Algorithm 1. Each category is composed of six separate sets, which combine DTMC's under- and over-stressed and well-utilized states with CTMC's dependable and unreliable states. The results are summarised in the table below. Therefore, each PM will be in one of the six sets WR, OR, UR, WU, OU and UU. In the CTMC model, unreliable states are related to semi-active and fail states. As a last step, these sets are categorised into crucial, optimum, and sub-optimal groups. An unreliable or overburdened PM will be placed in the critical category for further attention. An ideal category is reserved for well-used and dependable PMS; all other PMS fall into this subcategory. VM migration source PMs are identified after categorisation.

3.4 Estimation of VM Cost

(i) Migration Cost

Since servers are located in different geographic locations, different migration distance occurs during migration of data block from one server to another. Hence, migration cost is related to migration distance and size of data file.

Therefore, the migration cost of data block f_k migrated from the VM_i to the VM_j is expressed as:

$$C_{mig}(f_{,k}, VM_i, VM_j) = C_m S(f_k, VM_i) d(VM_i, VM_j) \tag{3}$$

where, C_m is parameter of migration cost.

$S(f_k, VM_i)$ is the size of the data to be migrated.

$d(VM_i, VM_j)$ is the distance between the two VMs

Hence, the migration cost of data file F migrated from the VM_i to the VM_j is:

$$\sum_{k=1}^{M}\sum_{j} C_{mig}(f_k, VM_i, VM_j) =_m \sum_{k=1}^{M}\sum_{j} C_m S(f_k, VM_i) d(VM_i, VM_j)$$

(4)

(ii) Communication Cost

Data transmission among servers mainly embodies in communication flow due to different geographic locations of servers, hence, distance between source (VM_i) and host (VM_j) is another factor influencing communication cost, expressed as:

$$C_{com}(f_k) = \sum_{i=1, i\neq j}^{N}\sum_{j=1}^{N}[S(f_k, VM_i)W(VM_i, VM_j)d(VM_i, VM_j)]$$

(5)

where, $W(VM_i, VM_j)$ represents the communication time among the VMs.

The whole communication cost of data file F after storage is:

$$C_{com}(F) = \sum_{k=1}^{M}\sum_{i=1, 1\neq j}^{N}\sum_{j=1}^{N}[S(f_k, VM_i)W(VM_i, VM_j)d(VM_i, VM_j)]$$

(6)

(iii) Total Cost

Total cost of VM_i is the sum of migration cost and communication cost of each data block.

$$C(F, VM_i) = C_{mig}(f_{,k}, VM_i, VM_j) + C_{com}(F)$$

(7)

3.5 Energy efficient VM Migration

Algorithm-1: Energy Efficient VM Migration

Let *MigrationVMs = NULL*

Let OverloadVMs = NULL

For each vm_i of host H_j

mCost <- *vmList[i].getMigrationCost(F)*

cCost <- vmList[i].getCommunicationCost(F)

tCost = mCost + cCost

If (sizeof(H_j) >*upperThresholdXhostCPUTotal)*

add *vm_i to OverloadVMs*

If (tCost(vm_r) = Minimum)

add vm_r to *MigrationVMs*

Else

$H_j = H_j + 1$

End if

End if

End for

If (*MigrationVMs* > 0)

Sort *MigrationVMs* based on tCost

For each vm_i of *OverloadVMs*

For each vm_r of *MigrationVMs*

If Memory(vm_r) > Memory(vm_i)

Relocate vm_i to vm_r

End if

End For

End For

In this algorithm, initially the list of MigrationVMs and OverloadVMs are initialized to NULL. Then an overloaded VM is detected by checking the total size of the host. If it is higher than a threshold for CPU, then the host is considered as overloaded and the corresponding overloaded VM should be relocated. A VM is included in the MigrationVMs list if it has the minimum total cost. If no such VM can be found, then the next host will be checked for suitable VMs. Then the overloaded VMs are scheduled and migrated to each VM on the MigrationVMs list such that the available memory of migrated VM is higher than that of the overloaded VM In this way, the process continues, until all the overloaded VMs are relocated or the load of the host becomes less than the upper threshold.

Since the migration contains the VMs with minimum cost, the energy consumed during the migration process will be less.

3.6 Energy Efficient Scheduling

3.6.1 Power Consumption of the Server

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

$$PC_{server} = PC_{idle} + PC_f(l) \tag{8}$$

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:

$$PC_{active}^{t} = \sum_{l=1}^{L}[N_0^t(l).(PC_{idle} + PC_f(l))] \tag{9}$$

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:

$$PC_{sleep}^{t} = \sum_{k=1}^{K}(PC_k \cdot N_k^t) \tag{10}$$

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:

$$PC^t = PC_{active}^{t} + PC_{sleep}^{t} \tag{11}$$

3.6.2 Constraint optimization for Scheduling

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 2: 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

1. Initially, $Server_{max} = Server_{active}$

```
For each t ∈ (1,2...τ)
For each Rj(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 ∈ 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
        End For
End if
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 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.

4 Experimental Results

The proposed CCEM-VM-MS technique is implemented in Cloudsim and compared with the Energy-Efficient VM migrations optimization (EE-VM-Optimized) [12] technique. The NASA workload [14] 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.

Table 1 shows the experimental parameters assigned in this work

Parameter	**Value**

Work load	NASA traces
Resource Utilization Thresholds	$U^{low-thr} = 20\% and U^{high\_thr} = 80\%$
Response Time Thresholds	$RT^{low-thr} = 200ms and RT^{high\_thr} = 1000ms$
Scaling Intervals	$\Delta t = 10min$
Desired Response Time	DRT = 1000ms=1s
Fault rate	1 to 2
Configuration of VMs	Medium and Large
Maximum On-demand VM Limitation	*Max*VM=10VM
Task and Resources Scheduling Policy	Time-Shared

Table 1 Experimental Parameters

4.1 Results and Discussion

In this section, the upper threshold value for CPU size is increased from 0.25 to 1, since a server is considered to be overloaded if its size exceeds this threshold value (as illustrated in Algoritrm-1).

The performance metrics power consumption, number of VM migrations, response delay, and CPU utilization are considered for evaluation.

. (i) Response Delay

Table 2 and Figure 3 shows the response delay measured for both the schemes.

CPU Threshold value	CEEM-VM-MS (kW/h)	EE-VM-Optimized (kW/h)
0.25	0.71	0.82
0.40	0.67	0.8
0.55	0.65	0.77

0.70	0.62	0.76
0.85	0.61	0.72
1.00	0.55	0.68

Table 2 Energy consumption for various CPU thresholds

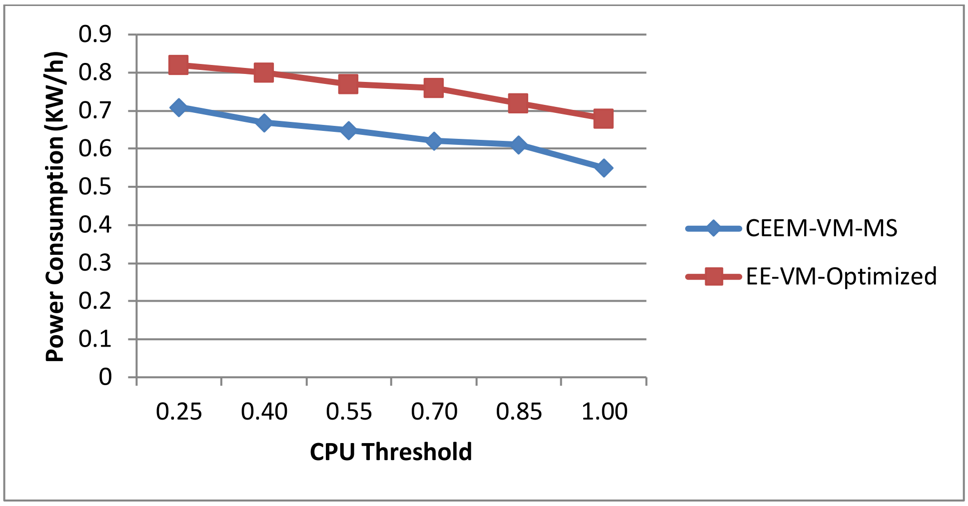

Figure 3 Power consumption for various CPU thresholds

Since lower values of CPU thresholds result in frequent migrations, there will be less number of migrations at higher CPU thresholds. As we can see from figure, the power consumption tend to decrease from 0.71 to 0.55 kW/h for CEEM-VM-MS and 0.82 to 0.68 kW/h for EE-VM-Optimized. However CEEM-VM-MS has 16% lesser power consumption when compared to EE-VM-Optimized.

(ii) Number of VM Migrations

Table 3 and Figure 4 shows the number of VM migrations measured for both the schemes

CPU Threshold value	CEEM-VM-MS	EE-VM-Optimized
0.25	6400	7345
0.40	4810	5781
0.55	3310	4210
0.70	2480	3630
0.85	1810	2875
1.00	1250	2350

Table 3 Number of VM migrations for various CPU thresholds

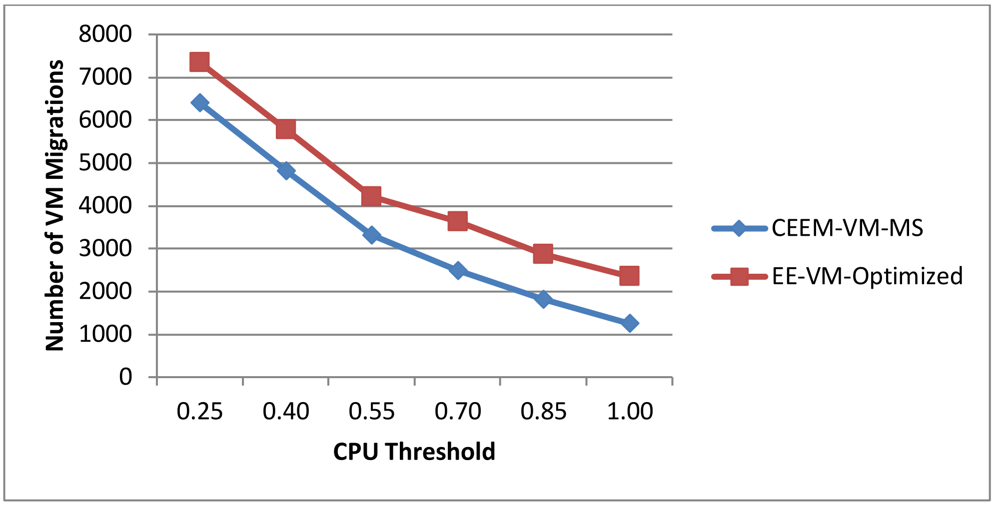

Figure 4 Number of VM migrations for various CPU thresholds

Since lower values of CPU thresholds result in frequent migrations, there will be less number of migrations at higher CPU thresholds. As we can see from figure, the number of migrations tends to decrease from 6400 to 1250 for CEEM-VM-MS and 7345 to 2350 for EE-VM-Optimized. However CEEM-VM-MS has 27% lesser VM migrations when compared to EE-VM-Optimized.

(iii) CPU Utilization

Table 4 and Figure 5 shows the CPU Utilizations measured for both the schemes

CPU Threshold value	CEEM-VM-MS (%)	EE-VM-Optimized (%)
0.25	70.55	65.28
0.40	71.33	67.73
0.55	72.52	70.36
0.70	73.11	71.66
0.85	76.50	73.11
1.00	78.00	75.25

Table 4 CPU utilization (%) for various CPU thresholds

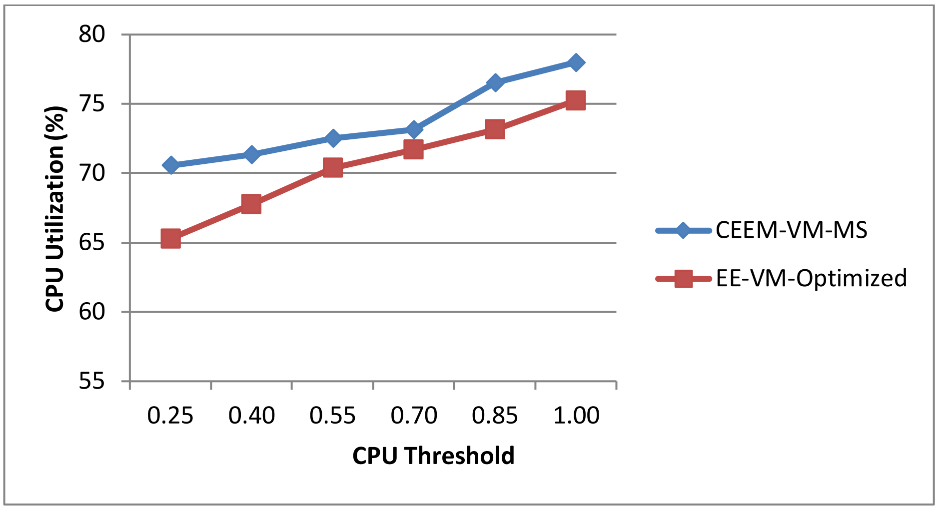

Figure 5 CPU utilization (%) for various CPU thresholds

Since lower values of CPU thresholds result in frequent migrations, the load will be balanced in overloaded and under loaded servers. Hence the CPU utilization tends to increase from 70.55% to 78% for CEEM-VM-MS and 65.28% to 75.25% for EE-VM-Optimized. However CEEM-VM-MS has 4% higher CPU utilization when compared to EE-VM-Optimized.

(iv) Response Delay

Table 5 and Figure 6 shows the response delay measured for both the schemes

CPU Threshold value	**CEEM-VM-MS (sec)**	**EE-VM-Optimized (sec)**
0.25	1.25	2.58
0.40	1.78	2.94
0.55	2.25	3.65
0.70	2.66	4.15
0.85	4.34	5.25
1.00	4.92	5.82

Table 5 Response delay for various CPU thresholds

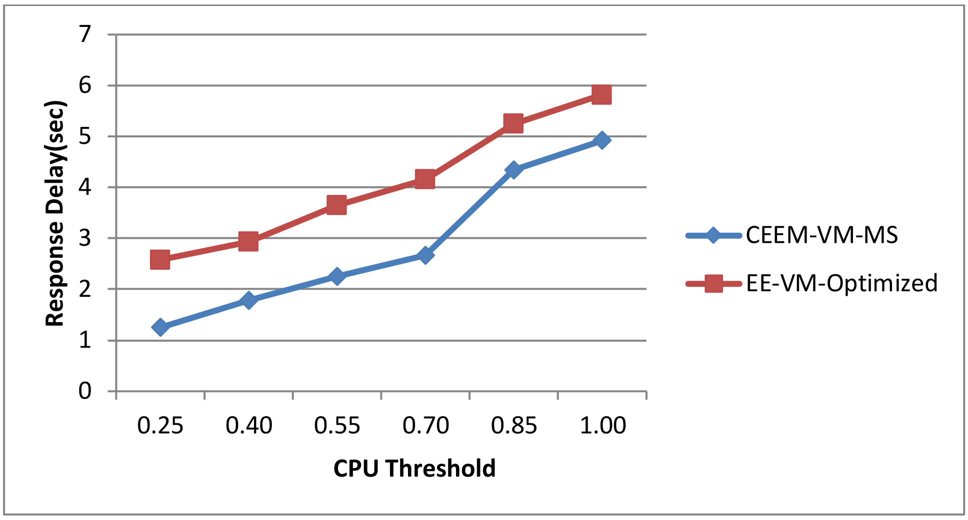

Figure 6 Response delay for various CPU thresholds

Since lower values of CPU thresholds result in frequent migrations, the load will be balanced leading to shorter response delays. Hence the response delay tends to increase from 1.92 to 4.5 seconds for CEEM-VM-MS and 2.58 to 5.82 seconds for EE-VM-Optimized. However CEEM-VM-MS has 33% lesser response delay when compared to EE-VM-Optimized.

5 Conclusion

In this paper, a cost effective and energy efficient VM migration for scheduling in cloud computing is developed. In this technique, VMs having minimum cost and minimized energy consumption are selected for migration. During load balancing, the tasks are migrated to the underutilized server if the needed current server is overloaded. Similarly, the tasks are migrated from the under-loaded server to the current server. In energy efficient scheduling, given the arrival of user requests, the VMs are scheduled such that the total energy consumption of the data center is minimized. The proposed CCEM-VM-MS technique is implemented in Cloudsim and compared with the Energy-Efficient VM migrations optimization (EE-VM-Optimized) technique. In experiments, the upper threshold value for CPU size is increased and the performance metrics power consumption, number of VM migrations, response delay, and CPU utilization are measured. Results indicate that the proposed CEEM-VM-MS technique minimizes the power consumption, reduces the number of VM migrations, reduced the response delay and increases the CPU utilization.

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.Mohammad H. Alshayeji, Sa'ed Abed and Mault D. Samrajesh, "Energy efficient virtual machine migration algorithm"Journal of Engineering Research, Vol-5, No-2, 2017.

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

4.Monireh H. Sayadnavard , Abolfazl Toroghi Haghighat, Amir Masoud Rahmani , "A multi-objective approach for energy-efficient and reliable dynamic VM consolidation in cloud data centers", Elsevier,Engineering Science and Technology, an International Journal, 2021.

5.E. I. Elsedimy and Fahad Algarni, "Toward Enhancing the Energy Efficiency and Minimizing the SLA Violations in Cloud Data Centers", Hindawi Applied Computational Intelligence and Soft Computing,Volume 2021, Article ID 8892734, 14 pages, 2021.

6.Djouhra Dad,Djamel Eddine Yagoubi and Ghalem Belalem, "Energy Efficient VM Live Migration and Allocation at Cloud Data Centers", International Journal of Cloud Applications and Computing, 2014.

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

8.R. Jayamala and A. Valarmathi, "An Enhanced Decentralized Virtual Machine Migration Approach for Energy-Aware Cloud Data Centers", Intelligent Automation & Soft Computing, Vol-27, No-2, 2021.

9.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.

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

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

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

13.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.

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