International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Adaptive Red Deer Optimization (ADRO) Technique for Energy Efficient VM Migration in Cloud Computing

[1]D. Komalavalli, [2]Dr. T. Padma

Submitted: 20/12/2023 **Revised**: 31/01/2024 **Accepted**: 10/02/2024

Abstract: Several migration techniques are available for migrating Virtual machine (VM) from one host to another. But they fail to consider the migration cost while determining the energy consumption during migration. The migration cost includes the migration time and distance. Hence the objective of this work is to design an optimized VM migration technique which simultaneously reduces the energy consumption and cost while avoiding (QoS) degradation. For this, Adaptive Red Deer Optimization algorithm for energy efficient VM migration (ARDO-EEM) in cloud computing is proposed. In ARDO-EEM, the overloading probability of each host is determined based on the total resource utilization of the host. Then the overloaded hosts are categorized into heavy, medium and light depending on two threshold values. VMs to be migrated are selected from the heavy and medium overloaded hosts with energy consumption higher than the available energy. The target VMs are selected using the ARDO algorithm based on the migration energy and resource utilization. Then each VM in the migration list is relocated to the selected target VM. Experimental results show that the proposed ARDO-EEM attains increased resource utilization with lesser power consumption and response delay.

Keywords: *Cloud computing; Virtual Machine (VM); Migration; Adaptive Red Deer Optimization (ARDO); Energy efficient*

1. Introduction

Cloud computing, such as the use of gas, electricity, water, and telephone utilities, is made possible by cloud computing's "pay-as-you-go" model for computing services. It can be known as the fifth utility. Three well-known cloud computing services are platform as a service (PaaS), infrastructure as a service (IaaS), and software as a service (SaaS). Different deployment models are used to deploy these services. Public, private, commodity, and hybrid are the four main deployment methods for clouds, depending on the needs. Users can reduce infrastructure and maintenance costs with the use of cloud computing. Cloud computing services offer scalability, dependability, and mobile accessible features to users. Various businesses neglect the low-level hardware and software infrastructure configuration in favour of innovation and delivering economic value. All computing services are being moved in the process [1].

Virtual machines can be moved from one physical device to another through virtual machine migration. It is a component of the software that controls hardware virtualization. The different types of VM migration include cold migration, which stops the VM on the source machine and restarts it on the destination, warm migration, which suspends the VM on machine 1, copies the RAM and CPU registers, and then restarts it on machine 2 (a few seconds later), and live migration, which transfers the VM's execution environment and stops it on the origin machine before restarting it on the destination machine without completely replacing the memory pages. This method's drawback is that iteratively resending updated pages causes inefficiency by using more network capacity [2].

Modern large-scale computer systems, like data centres, face three major challenges: a reduction in energy usage, run faster and cooler, and occupy less space. Multicore processor technology is addressing these issues simultaneously. High-performing computing infrastructures have emerged as a key issue in the new global economy in order to meet the need for contemporary resource-intensive businesses and research applications. Due to the current massive electrical power consumption in large-scale computer data centres, the issue has gained in prominence. However, because of the world's high energy usage, these quick changes are seriously affecting the environment. Both the size of data centres and their operating costs are expanding. As of 20142, the estimated cost of energy usage will be 75% of the overall expenditure. There will be a demand for performance- and energy-efficient VM migration techniques [3].

2. Related Works

A reassessment of the dynamic VM consolidation problem and the best online deterministic migration techniques in an experimental setting. By simulating the

[1]*Research Scholar, Bharathiar University, Coimbatore.*
**komalavalliphd@outlook.com*
[2]*Professor and Head, Department of Master of Computer Applications, Sona College of Technology, Salem*
padmatheagarajan@gmail.com

current energy-efficient techniques, Akhter et al. [4] have discovered new information. We might utilise some additional statistical techniques, according to the reevaluation, to reduce the amount of energy used in cloud-based data centres. Environmental issues are becoming more and more important to cloud users. A significant source of CO_2 emissions, the energy consumption of data centres is expanding quickly and has led to increasing energy consumption throughout the economy. Energy regulations are a major concern for many nations worldwide in an effort to cut greenhouse gas emissions.

In order to address crucial elements that have an impact on the datacenter servers while moving VMs, Alshayeji et al. [5] have introduced the Energy Efficient Virtual Machine Migration (EVM) technique. In order to choose the victim and target servers, the EVM approach uses Energy Based Server Selection (ESS). Comparative results demonstrate that EVM outperforms existing techniques like Arbitrary Server Selection (ASS) and First Fit Strategy by reducing the amount of server state changes, VM migration, and oscillations (FFS).

By developing a discrete-time Markov chain (DTMC) model to forecast future resource utilisation, Sayadnavard et al [6] have overcome the issue. The reliability model of PMs is used in conjunction with the DTMC model to classify PMs more accurately based on their condition. Then, using the e-dominance-based multi-objective artificial bee colony (e-MOABC) algorithm, a multi-objective VM placement approach is proposed to achieve the optimal VMs to PMs mapping. This algorithm is capable of effectively balancing the overall energy consumption, resource waste, and system reliability to meet SLA and QoS requirements. By using the CloudSim toolbox to conduct a performance evaluation study, we have confirmed the efficacy of our suggested strategy. While avoiding the ineffective VM migrations, energy consumption is greatly improved.

The resource allocation and live migration of virtual machines have been studied by Dad et al [7]. It suggests a Double Threshold Migration (DTM) technique that takes both an upper and a lower threshold of CPU utilization into account. One can choose a number of VMs to perform the transfer using these Thresholds. The live transfer of the VMs lowers the high server utilization and turns off the idle physical machines (PMs). The approach employs a variation of the Best Fit Decreasing (MBFD) method to address the issue of VM placement. The results of the experiments demonstrate that the suggested strategy increases resource utilisation, decreases energy consumption, and maintains SLA (Service Level Agreement) violations while under an energy limitation.

An effective and economical VM consolidation strategy termed EQ-VMC, put out by Li et al. [8], aims to maximize service quality and energy efficiency. To find the overall ideal solution for VM placement, a discrete differential evolution algorithm is created. EQ-VMC effectively lowers energy usage and enhances quality of service by combining this solution with a series of algorithms proposed for efficient host overload detection, VM selection, and under-loaded host identification).

Thiam and Thiam [9] have researched the issue of the best VM migration and allocation strategy to reduce energy usage in a data centre while maintaining QoS. A cloud environment is created using the CloudSim simulator. It offers the user interface for interacting with both virtual and physical machines. In order to identify the method that optimizes VM placement and migration, we evaluate and contrast the algorithms corresponding to various techniques. The simulation's outcome demonstrates how virtual machine placement and migration strategies reduce energy use, migrations, and overall simulation duration.

According to the needs of the clients, Shahapurea and Jayarekha [10] have presented a new paradigm for virtual machine migration. A customer asks for a quicker response time. A technique based Virtual Machine Migration Approach based on Distance and Traffic is created in order to do this. The methodology is based on the data centres' physical locations and traffic patterns. The programme is conducted on a regular basis to monitor network traffic. The distance between the data centres from which the client request must be sent is also checked. The nearest data centre with less traffic receives the request. This decreases the round trip time, which enhances performance. Since there is less traffic, the resources are used more efficiently. By offering clients speedier services, the reduced round trip time and preserving the round trip time with little variation even in the event of a physical machine failure contribute to performance improvement.

The modified best fit decreasing (MBFD) method and the load aware three-gear threshold (LATHR) algorithm have both been suggested by Vaneet and Balkrishan [11] for reducing overall energy usage while enhancing SLA-related quality of service. Under dynamic workload conditions and a range of VMs (1-290) distributed to each host, it shows encouraging results. SLA, energy use, instruction energy ratio (IER), and the number of migrations against the various numbers of VMs are used to evaluate the effectiveness of the proposed effort. In comparison to median absolute deviation (MAD), interquartile range (IQR), and double threshold (THR) overload detection strategies, the proposed technique decreased SLA violations and energy consumption.

By focusing on system structure analyses, Dhaya et al. [12] have developed a paradigm for the distribution of resources in private cloud data centres that is energy-efficient. On the other side, we wish to give private cloud service providers with the most up-to-date design and performance analysis for resource allocation that is energy-efficient. The methodology should be flexible enough to handle a variety of computing platforms, as well as approaches for delivering extensive and on-demand resources, scheduling cloud environments, and bridging the gap between private cloud users and a full picture of offers.

A virtual machine migration technique based on the three-way decision (VMM-3WD) has been proposed by Jiang and Shi [13] to reduce the energy consumption of cloud hosts while taking network correlation between virtual machines into account. According to their load state, the initial step of the method is to categorize hosts into overloaded, ordinary load, and underloaded hosts. Following that, various migration procedures are designed specifically for these three categories of cloud hosts. The method specifically migrates virtual machines from under-loaded servers to typical load hosts. The strategy then produces two thresholds to classify different levels of host overload into massively overloaded, moderately overloaded, and lightly overloaded hosts. The choice to migrate VMs is made with the intention of lowering network energy usage throughout the migration process.

An energy-efficient and QoS-aware VM consolidation strategy has been put forth by Wang et al [14]. To detect host state, a mixed prediction model based on grey model and ARIMA is used. A new technique is offered, using a VM selection strategy known as AUMT to choose VMs with low average CPU utilization and short migration times, and a VM placement policy based on resource use and variable energy consumption to choose the best host. When compared to benchmark approaches, this approach enables the reduction of energy consumption, the number of migrations, SLAV, and ESV objectives. The AUMT can also reduce energy consumption, the number of migrations, and ESV.

In our previous works [16][17], swarm intelligence based task scheduling algorithm and optimal server selection model for task allocation were proposed. Apart from these works, a Neuro-Fuzzy Inference QoS aware genetic algorithm [18] is proposed for cloud resource optimization.

2.1 **Research gaps**

Nowadays more attention focuses on VM management strategies in a variety of scenarios. The problem of virtual machine placement and migration is an optimization problem aiming for multiple goals. The efficiency of a data center, therefore, depends a lot on how virtual machines are provisioned and where they are located. An efficient VM allocation policy will improve energy efficiency while limiting the degradation of the QoS and alleviate hotspots, but will also reduce the operating costs of the data center. Migrating VMs into a fewer number of Physical Machines (PMs) can maximize the utilization of Cloud servers and reduce the energy consumption of the Cloud data center.

3. Proposed Methodology

3.1 Overview

In this paper, ARDO-EEM algorithm is proposed. In this work, the overloading probability of each host is determined based on the total resource utilization of the host. Then the overloaded hosts are categorized into heavy, medium and light depending on two threshold values. VMs to be migrated are selected from the heavy and medium overloaded hosts with energy consumption higher than the available energy. The target VMs are selected using the ARDO algorithm based on the migration energy and resource utilization. Then each VM in the migration list is relocated to the selected target VM.

3.2 Energy Consumption for VM migration

The utilization rate R_i of the CPU in physical host i is defined as below:

$$R_i = \frac{w_i^{workload}}{w_i^{max}} \tag{1}$$

where $w_i^{workload}$ is the overall workload on i and w_i^{max} is the CPU capacity of full workload.

Then, the energy consumption E_i of host i is represented by:

$$E_i = E_{idle} + (E_{max} - E_{idle})R_i(t) \tag{2}$$

where E_{idle} and E_{max} are the power consumption of the physical host in idle and full workloads, respectively. The value of R_i ranges between 0 and 1.

The energy that physical host j consumes in period [$t0$, $t1$] is defined as below:

$$E_{E_i} = \int_{t_0}^{t_1} E_i(R_i(t))dt \tag{3}$$

The energy consumption of VM migration is calculated using the physical host's and communications' energy consumptions.

Let l(k) represent the volume of data that will be moved during the VM migration over communication k. The formula below can be used to determine how much energy is used to transport l(k) units of data.

$$E(k) = e_k \cdot l(k) \tag{4}$$

where e_k is explained as the amount of energy used for a unit of data transfer when using various communication methods during VM migration.

Calculations for the total energy used by VM migration are shown below.

$$E_M = \sum_{E_i \in E_\Pi} E_{E_i} + \sum_{k \in K_\Gamma} E(k) \quad (5)$$

where E_Π indicates the group of physical hosts in the VM migration; K_Γ indicates the group of different communications.

3.3 Overloading Probability

Let RP = {BW, MeM, CPU} be the resource pool on each host h_j, where BW, MEM and CPU corresponds to the available bandwidth, memory and CPU capacities.

Let CP_j denote the resource capacity of h_j , for each resource $res \in$ RP. Then the resource utilization of VM_i on h_j is computed as

$$RU_{ij} = AU_i / CP_j \quad (6)$$

where AU_i indicates the actual resource utilization of VM_i in h_j.

Then the total resource utilization of the host h_j is computed as

$$RU_j = \sum_{vm_i \in V} AU_i / CP_j \quad (7)$$

where *V* denotes the group of VMs installed on h_j, .

When the total resource utilization of a host becomes higher than its resource capacity, then the chances of overloading will be higher for that host. The overloading probability of h_j , denoted as OPr_j is derived as

$$OPr_j = Pr_{res}(RU_j > CP_j), \ \forall res \in RP \quad (8)$$

where Pr_{res} is the probability distribution function of the utilization of various resources on a host.

3.4 Threshold determination

Let Th_{up} and Th_{down} be the two thresholds for the upper and lower bounds of the overload probability, respectively. The hosts can be categorized as shown in Table 1, depending on these two thresholds:

Category	Definition	Condition
HO	Heavy overloaded	$OPr_j > Th_{up}$
MO	Medium overloaded	$Th_{down} \leq OPr_j < Th_{up}$
LO	Lightly overloaded	$OPr_j < Th_{down}$

Table 1 Overloading Category of Hosts

The classification of hosts as overloaded, usually loaded, and underloaded hosts occurs during the first stage of threshold determination in VM migration, which is covered in more detail in this section.

3.5 Selection of Target VMs for Migration

3.4.1 Deriving the fitness function

A Fitness function is derived to select the target VMs for migration, based on the total VM migration energy consumption E_M and the actual resource utilization AU.

$$F(VM_i) = 1 / (w_1.E_{Mi} + w_2.AU_i) \quad (9)$$

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

The objective is to select the VMs having maximum fitness function. For achieving this solution, we apply ARDO algorithm, as described in the following section.

3.4.2 ARDO Algorithm

Like other optimization algorithms, the Red Deer Optimization (RDO) [15] algorithm begins with initial populations RD. The best RD is the male RD and remaining ones are known as hinds. Based on the roaring mechanism, the RD is classified as leader and stags. The mathematical formulation of the ARDO is explained below:

a. Initialization

The process begins with M_{pop} (population) and the best RD to M_{male} and remaining are M_h . M_{male} and M_h are used for managing intensification and diversification characteristics.

Roaring characteristics of male RDs: In this stage, male RDs tries to enhance the grace using roar. Based on the solution space, neighbors of the male RD's are identified and when the solution of neighbors are better than male RD, the previous solution is replaced. Each male RD vary the position and it is given as:

$$male_n = \begin{cases} male_o + b_1 \times ((UL - LL) \times b_2) + LL) & when\ b_3 \geq 0.5 \\ male_o - b_1 \times ((UL - LL) \times b_2) + LL) & when\ b_3 < 0.5 \end{cases} \quad (10)$$

where $male_n$ and $male_o$ are the updated and present positions of male RD. b_1, b_2 and b_3 are the random numbers, UL and LL are the upper and lower limits of the search space.

b. Selecting β percentage of the best male RD

The male RD's are divided into leaders and stags. Total of male leaders is represented as:

$$M_{lead} = round\{\beta \times M_{male}\} \quad (11)$$

where β is the initial value and the total of stags are computed by:

$$M_{stag} = M_{male} - M_{lead} \quad (12)$$

c. Collision between the leader and stags

Let the leader and stags collide randomly. The fitness solution of collision process is given as:

$$N1 = \frac{leader + stag}{2} + b_1 \times ((UL - LL) \times b_2) + LL) \quad (13)$$

$$N2 = \frac{leader + stag}{2} - b_1 \times ((UL - LL) \times b_2) + LL) \quad (14)$$

where $N1$ and $N2$ are the new solution obtained during the collision.

d. Generation of harems

To generate the harems, the hinds between the leaders are divided and it is given as:

$$U_m = u_m - \max(u_m) \quad (15)$$

where u_m is the energy of the m^{th} leader and U_m is a normalization power of the leader.

Mate leader of harem: RD mate with each other and it is performed by the leader and β percentage of hinds in the harem is given as:

$$M.harem_m^{mate} = round(\beta \times M \times harem_m) \quad (16)$$

where $M \times harem_m$ is the total hinds of the m^{th} harem.

Mate leader of harem: In the harem, total of hinds that mate with the leader is given as:

$$M.harem_k^{mate} = round(\beta \times M \times harem_k) \quad (17)$$

where $M.harem_k^{mate}$ is the hinds in the k^{th} harem.

e. Mating of stag and nearby hind

Here, every stag mate with the nearby one. For finding the nearby hind, the distance of stage and hinds in the l^{th} dimension is given as:

$$D_l = \left(\left(Stag_l - hind_l^j \right)^2 \right)^{1/2} \quad (18)$$

where D_l is the distance of stage and hinds in the l^{th} dimension.

The standard RDO has seven input parameters like a number of iterations, population, male, hind, α, β and γ. These parameters make the tuning process as complex. To tune these parameters, a number of leaders must be set. Every leader makes a harem. The following expression is used for updating the percentage of leaders between the males.

$$\gamma = \left(0.1 + 0.9 \times \frac{iter}{\max\_iter} \right) \quad (19)$$

$$\alpha = \left(0.5 + 0.5 \times \frac{iter}{\max\_iter} \right) \quad (20)$$

$$\beta = 1 - \alpha \quad (21)$$

In ADRO, only the parameters number of iteration, population, and male are considered. The pseudocode of ADRO is given in Algorithm 1.

Algorithm 1: Pseudocode of ADRO

Initialized the population of RD

Initialize the parameters number of iteration, population, and male

Compute fitness using Equation (2)

$iter = 1$

while $(iter < \max\_iter)$

for every male RD

Position is updated when the obtained solution is better than the previous one

end for

Arrange the male and leader as γ percent of every males and γ is updated by Equation (12)

for every male leader

Collision among the leader and stag

Update the leader and stag position

end for

Create harems

for every male leader

Randomly choose harem k

if the fitness is less than harem k

Male leader mate with the chosen hinds using Equation (13) of the harem

Male leader mate with the chosen hinds using Equation (14) of the harem

else

Male leader mate with the chosen hinds using Equation (14) of the harem

Male leader mate with the chosen hinds using Equation (13) of the harem

end if

end for

for every stag

Compute the distance of stage and hinds and choose the nearby hind

end for

Update the better solution is obtained

$iter = iter + 1$

end while

Return optimal solution

3.6 VM Migration Algorithm

The steps involved in ADRO based VM migration are summarized in the below algorithm.

Notations	**Definition**
========	========
NoVMs	List of VMs to be migrated.
h_t	Target host
E_{Mj}	total energy consumption of host h_j.

AE_j Available battery energy of h_j

ME_{max} Maximum threshold for migration energy

bestVM best VM returned from ARDO

Algorithm-2: ADRO-EEVM

Let NoVMs = *NULL*

For each host h_j

Estimate VM_j using Eq.(5)

Estimate OPr_j using Eq.(7)

If $E_{Mj} > AE_j$ or (h_j = HO or h_j=MO), then

For each vm_i of h_j

If $(E_i(k)_r) < ME_{max}$, then

Add vm_i to {NoVMs}

Else

Skip vm_i

End if

End for

If (NoVMs > 0)

For each vm_i of {NoVMs}

Select target host h_t

For each vm_r of h_t

Return bestVM using ARDO()

If vm_r == bestVM, then

Relocate vm_i to vm_r

End if

End For

End For

End if

End For

In this algorithm, the total energy consumption and overload priority of each host are determined. If the host falls in the HO or MO category or the energy consumption becomes greater than its available battery energy, then it will be added in the list of migration VMs. If the migration energy of the added VM is more than the maximum level ME_{max}, then it will be removed from the list. If multiple VMs are selected from the host, then the target VMs are selected from the target host ht, based on the best returned VM from ARDO algorithm. Then each VM in the migration list is relocated to the selected target VM.

4. Experimental Results

The proposed ADRO-EEVM algorithm is implemented in Cloudsim and compared with the Energy-aware dynamic VM consolidation (DVMC) [3] and Energy-efficient and Quality-aware VM Consolidation (EQ-VMC) [8] techniques. 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 Results

In each experiment, the number of tasks is varied from 10 to 60 and the performance is evaluated in terms of power consumption, number of VM migrations, CPU utilization (%) and response delay.

No of Tasks	ADRO-EEVM (KW/h)	DVMC (KW/h)	EQ-VMC (KW/h)
10	0.75	0.81	0.84
20	0.72	0.77	0.81
30	0.7	0.75	0.8
40	0.67	0.73	0.78
50	0.65	0.7	0.75
60	0.64	0.68	0.73

Table 2: Results of Power Consumption

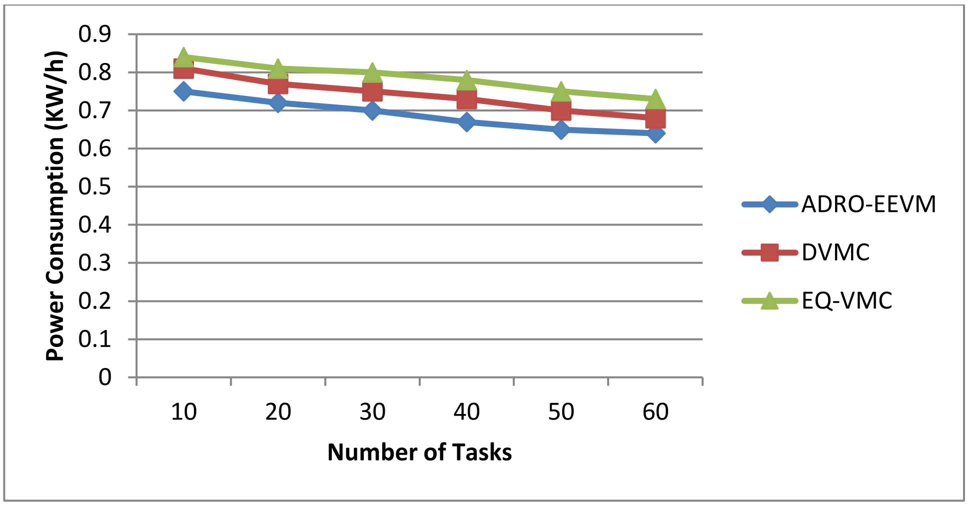

Fig 1 Number of tasks Vs Power Consumption

Table 2 and Figure 1 show the results of power consumption for varied tasks. From figure 1, we can observe that the power consumption of our proposed ADRO-EEVM is 7% lesser than DVMC and 12% lesser than EQ-VMC.

No of Tasks	ADRO-EEVM	DVMC	EQ-VMC
10	2400	3345	3785
20	2210	3181	3554
30	1980	2810	3241
40	1750	2630	2921
50	1630	2475	2752
60	1450	2350	2545

Table 3: Results of Number of VM Migrations

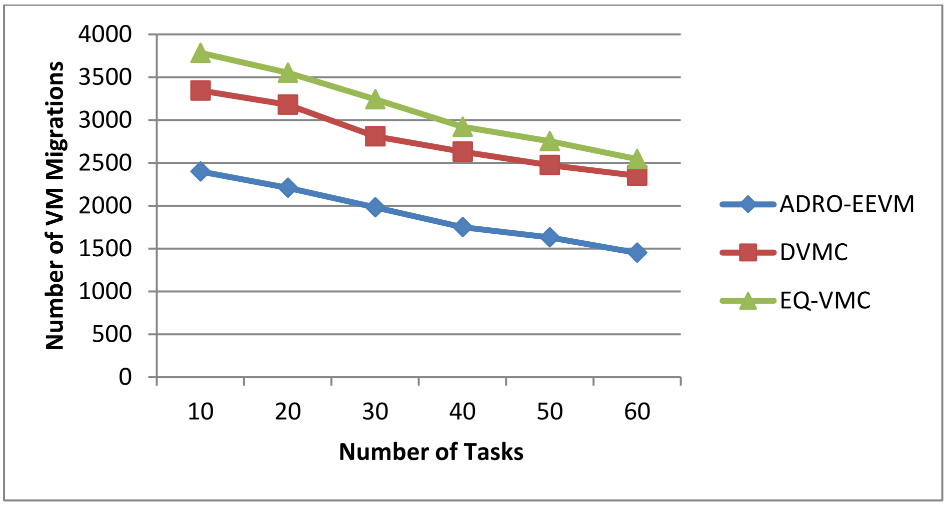

Fig 2: Number of Tasks Vs VM Migration

Table 3 and Figure 2 show the results of number of VM migrations for varied tasks. From figure 2, we can observe that the number of VM Migration of our proposed ADRO-EEVM is 32% lesser than DVMC and 40% lesser than EQ-VMC.

No of Tasks	ADRO-EEVM (%)	DVMC 9%)	EQ-VMC (%)
10	86.55	82.28	75.65
20	88.33	83.73	77.48
30	92.52	85.36	81.98
40	93.11	87.66	83.65
50	95.50	90.11	86.84
60	97.00	92.25	88.76

Table 4: Results of CPU Utilization

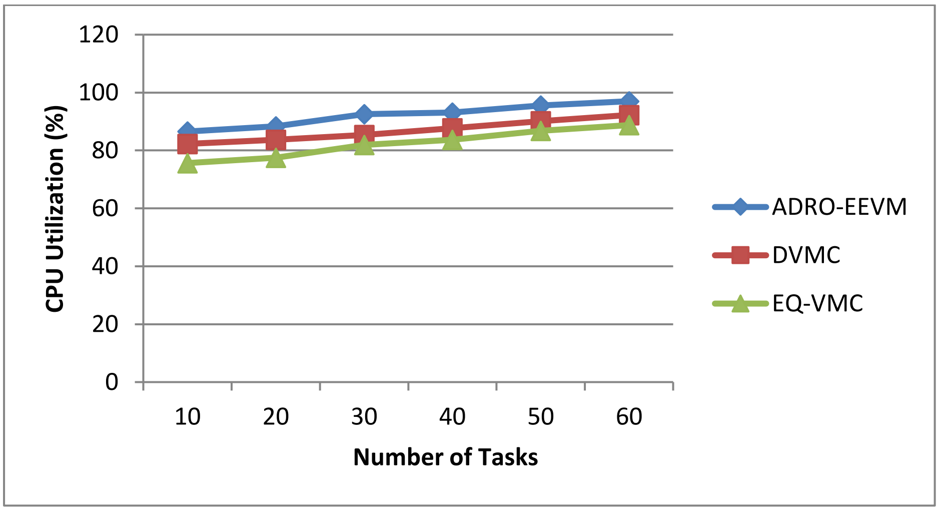

Fig 3: Number of Tasks Vs CPU Utilization

Table 4 and Figure 3 show the results of CPU utilization for varied tasks. From figure 3, we can observe that the CPU Utilization of our proposed ADRO-EEVM is 6% lesser than DVMC and 11% lesser than EQ-VMC.

No of Tasks	ADRO-EEVM (sec)	DVMC (sec)	EQ-VMC (sec)
10	1.25	2.58	3.12
20	1.78	2.94	3.45
30	2.25	3.65	4.12
40	2.66	4.15	4.95
50	4.34	5.25	5.01
60	4.92	5.82	5.52

Table 5: Results of Response Delay

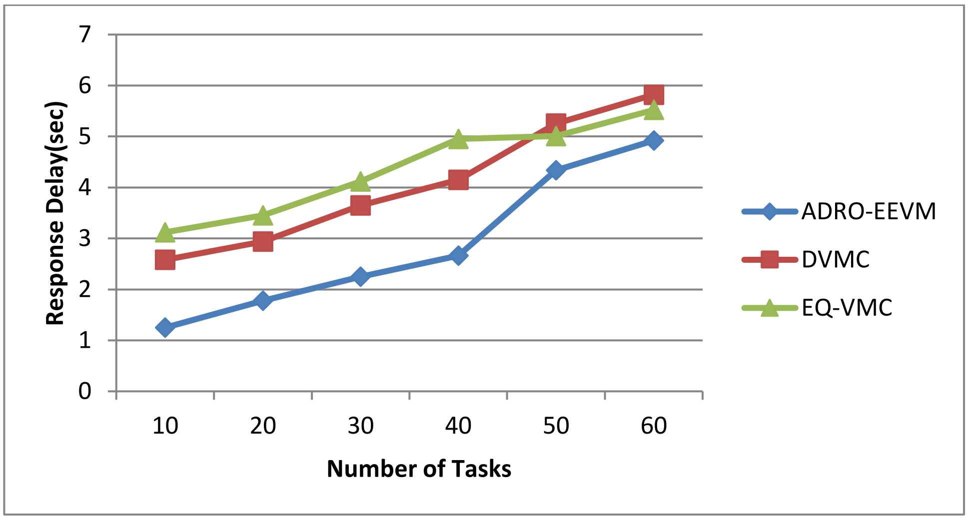

Fig 4: Number of Tasks Vs Response Delay

Table 5 and Figure 4 show the results of response delay for varied tasks. From figure 4, we can observe that the response delay of our proposed ADRO-EEVM is 33% lesser than DVMC and 37% lesser than EQ-VMC.

5. Conclusion

In this paper, ARDO-EEM technique for cloud computing is proposed. In this work, the overloading probability of each host is determined based on the total resource utilization of the host. Then the overloaded hosts are categorized into heavy, medium and light depending on two threshold values. VMs to be migrated are selected from the heavy and medium overloaded hosts with energy consumption higher than the available energy. The target VMs are selected using the ARDO algorithm based on the migration energy and resource utilization. Then each VM in the migration list is relocated to the selected target VM. Experimental results show that the proposed ARDO-EEM attains increased resource utilization with lesser power consumption and response delay.

Ethical Approval: The authors have ethical approval to publish this research manuscript.

Funding: The authors didn't get any funding support for this research

Data Availability: The authors didn't use any third party data materials in their research.

Conflict of Interest: The authors don't have any conflict of Interest

Authors Contribution: In this manuscript preparation author 1 prepared the concept and author 2 prepared the implementation part.

References

[1] Zhen Xiao, Senior Member, IEEE, Weijia Song, and Qi Chen, "Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment," IEEE transactions on parallel and distributed systems, vol. 24, no. 6, pp 1107-1117 IEEE, JUNE 2013

[2] Priyanka H, Mary Cherian, 2020, The Challenges in Virtual Machine Live Migration and Resource Management, International Journal Of Engineering Research & Technology (IJERT) IETE – 2020 (Volume 8 – Issue 11).

[3] Huiwang and huaglorytian field, "Energy-Aware Dynamic Virtual Machine Consolidation for Cloud Datacenters", IEEE transactions, volume 6, 2018.

[4] Nasrin Akhter1, Mohamed Othman1, Ranesh Kumar Naha, "Evaluation of Energy-efficient VM Consolidation for Cloud Based Data Center - Revisited", 2020.

[5] Mohammad H. Alshayeji*, Sa'ed Abed and Mault D. Sam rajesh, "Energy efficient virtual machine migration algorithm", 2017.

[6] Monireh H. Sayadnavard a , Abolfazl Toroghi Haghighat b,⇑ , Amir Masoud Rahmani, "A multi-objective approach for energy-effificient and reliable dynamic VM consolidation in cloud data centers", Engineering Science and Technology, an International Journal, 2021.

[7] Djouhra Dad, et al., "Energy Efficient VM Live Migration and Allocation at Cloud Data Centers", International Journal of Cloud Applications and Computing, September 2014.

[8] Zhihua Li a,c,∗, Xinrong Yu a, Lei Yu b, Shujie Guo a,c, Victor Chang, "Energy-efficient and quality-aware VM consolidation method", Future Generation Computer Systems 102 (2020) 789–809.

[9] Cheikhou Thiam and Fatoumata Thiam, "An Energy-Effificient VM migrations optimization in Cloud Data Centers", 2020 IEEE.

[10] Nagamani H Shahapurea and P Jayarekha, "Distance and Traffic Based Virtual Machine Migration for Scalability in Cloud Computing", Procedia Computer Science 132 (2018) 728–737.

[11] GARG Vaneet and JINDAL Balkrishan, "Energy efficient virtual machine migration approach with SLA conservation in cloud computing", J. Cent. South Univ. (2021) 28: 760−770.

[12] Dhaya R. ,1 Ujwal U. J.,2 Tripti Sharma ,3 Mr. Prabhdeep Singh ,4 Kanthavel R., Senthamil Selvan,6 and Daniel Krah, "Energy-Efficient Resource Allocation and Migration in Private Cloud Data Centre", Wireless Communications and Mobile Computing, Volume 2022, Article ID 3174716, 13 pages.

[13] Chunmao Jiang ∗, Ling Yang, Rui Shi, "An energy-aware virtual machine migration strategy based on three-way decisions", Energy reports, 2021.

[14] Jinjiang Wang, Hangyu Gu, Junyang Yu* , Yixin Song, Xin He and Yalin Song, "Research on virtual machine consolidation strategy based on combined prediction and energy-aware in cloud computing platform", Journal of Cloud Computing (2022) 11:50.

[15] Fathollahi-Fard, A.M., Hajiaghaei-Keshteli, M. and Tavakkoli-Moghaddam, R., "Red deer algorithm (RDA): a new nature-inspired meta-heuristic", Soft Computing, 2020, 24(19), pp.14637-14665.

[16] D Komalavalli, T Padma, Swarm intelligence-based task scheduling algorithm for load balancing in cloud system, International Journal of Enterprise Network Management 12 (1), 1-16, 2021. https://doi.org/10.1504/IJENM.2021.112669

[17] D Komalavalli, T Padma, An Optimal Server Selection Model for Dynamic Task Allocation in Cloud, International Conference on Communication and Computational Technologies, 879-890, 2023. https://doi.org/10.1007/978-981-99-3485-0_69.

[18] D Komalavalli, T Padma, An Optimal Server Selection Model for Dynamic Task Allocation in Cloud, International Conference on Communication and Computational Technologies, 879-890, 2023. https://doi.org/10.1007/978-981-99-3485-0_69.