International Journal of

INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Efficient Load Balancing and Optimal Resource Allocation Using Max-Min Heuristic Approach and Enhanced Ant Colony Optimization Algorithm over Cloud Computing

[1*]M. R. Banupriya, [2]D. Francis Xavier Christopher

Submitted: 27/06/2023 **Revised**: 06/08/2023 **Accepted**: 28/08/2023

Abstract: The paradigm of virtualization technology, which underpins cloud computing, has lately become one of the most prominent concepts in the information technology (IT) sector. Virtualization is a technology which helps the users to access the cloud services. In the existing system, the resource allocation is not ensured and in few cases, speed of the process is reduced due to convergence issues. Hence, the performance of cloud computing as a whole has greatly declined. The Max-Min Heuristic (MMH) and Enhanced Ant Colony Optimisation (EACO) algorithms are introduced in this study to enhance load balancing and optimum resource allocation on the cloud to address the aforementioned difficulties. The suggested system comprises four primary stages, including cost-effective Virtual Machine (VM) migration, load balancing, and resource allocation. First, think about how many resources, tasks, virtual machines, and cloud users there are in cloud computing. This study uses the MMH method for load balancing, which equalizes the overall workloads throughout the cloud. By moving tasks from overloaded nodes to under loaded nodes, load balancing is accomplished. Following that, the EACO algorithm is utilized to allocate resources in a way that effectively chooses more optimum resources. In order to effectively fulfil Quality of Service (QoS) standards, it is utilized to choose the best resources for the relevant cloud needs. Increasing throughput and VM performance in the cloud, as well as lowering costs, are other key objectives. Finally, a cost-effective VM migration technique is used, which is based on the Weighted Support Vector Machine (WSVM) algorithm. With the use of SVM weight values, it is designed to identify the pattern of overload and underload. It also finds VM migration strategies that consume the least amount of energy while maintaining high service standards. The simulation results show that, compared to the current approaches, the proposed MMH+EACO algorithm performs better thanks to increased throughput and reduced computational complexity, cost complexity, Mean Square Error (MSE) rate, and energy usage.

Keywords*: Cloud computing, Max-Min Heuristic (MMH) and Enhanced Ant Colony Optimization (EACO) algorithm, load balancing, resource allocation*

1. Introduction

A significant architecture for carrying out complicated and large-scale computation is emerging: cloud computing. It gives customers "Pay-as-you-go" access to services on demand. In the field of distributed computing, it is flourishing as an emerging technology. It describes programs and services that operate across a dispersed network and are accessible via the use of widely used internet protocols and standards [1]. With minimum to no contact between the cloud service provider and end-users through the internet, cloud computing offers on-demand access to a pool of programmable resources, including software and infrastructure. By satisfying the standards for Quality-of-Service (QoS), it provides services to users in accordance with Service-Level Agreements (SLA). Resources may be supplied and released easily and with little administration.

Without possessing and controlling the complexity of the highlighted technology, it has grown to be a popular approach for carrying out large-scale complicated computation and service delivery in a distributed context [2]. In a cloud computing context, task scheduling is one of the most important technologies. It is a schedule for projects where

- Tasks are items of labor that must be completed within a certain time frame
- When work is divided into numerous tasks, scheduling is the act of assigning the proper resources to each task.

For minimal execution time and maximum resource utilization while ensuring QoS, task scheduling in cloud computing is posing a difficulty [3] [4]. Task scheduling is a crucial technique in cloud computing environments as it facilitates the allocation of tasks to suitable resources, thereby ensuring efficient resource utilization and optimizing the overall performance of the system.

[1]Associate Professor, Department of Computer Applications, Kongunadu Arts and Science College, Coimbatore.*
E-mail: banupriyakongu@gmail.com
[2]Principal & Professor in Computer Science, SRM Trichy Arts and Science College, Trichy
E-mail: principal@asc.srmtrichy.edu.in

Due to the immense number of diversely arriving tasks with diverse resource needs, task scheduling includes the load balancing of jobs on virtual machines as a key component. In order to balance the load across all the nodes (hosts or virtual machines), a technique known as load balancing must first identify overloaded and under loaded nodes [5]. In order to achieve high user satisfaction, accelerate job execution, and increase system stability, load balancing's primary goals are to optimize the system's lifespan, completion time, and resource utilization rate. Therefore, the aforementioned goals are often attained by utilizing meta-heuristic and heuristic approaches that may converge to optimum or very-optimal solutions [6]. In order to minimize or maximize at least one QoS measure, such as makespan time, execution cost, etc., the distribution of tasks across many heterogeneous VMs is the ideal approach.

Within the context of cloud computing, service providers are responsible for the management of cloud resources based on a pricing model that operates on a per-demand basis. In order to maintain profitability, service providers must effectively balance the provision of high-quality service and maximum user satisfaction. As a result, resource allocation is crucial in cloud computing [7] and has an impact on the Service Level Agreement (SLA), which measures the amount of customer satisfaction, Quality of Service (QoS), and overall system performance. The server may experience an increase in the number of simultaneous users, depending on the user's demand at any given moment. Unexpected loads may result in poor QoS that breaches SLAs if the supplier does not offer enough resources to provide QoS. According to a reaction time increase of 100 ms, Amazon claimed a loss of $245 million. An efficient resource allocation plan is necessary to stay away from such circumstances and guarantee QoS. An example of resource allocation is shown in Fig. 1.

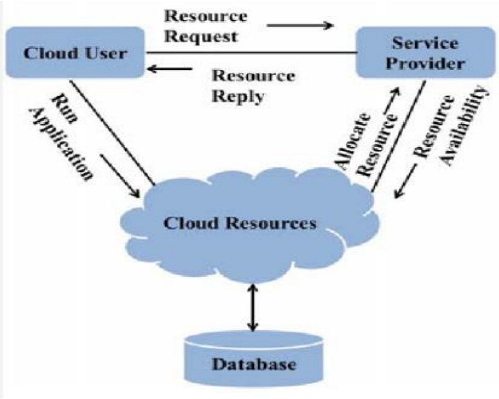


Fig 1 Example of resource allocation

In a cloud computing context, load balancing and resource allocation are the key objectives of this research project. Although several studies and approaches have been developed, little progress has been made in terms of resource allocation. With regard to process speed, the current methods have limitations. The Max-Min Heuristic (MMH) and Enhanced Ant Colony Optimization (EACO) algorithms are suggested in this study to address the aforementioned problems and enhance the overall performance of VM migration in the cloud. The creation of a system model, virtualization, load balancing, resource allocation, and cost-effective VM migration are the primary contributions of this study. Utilizing efficient algorithms for the cloud environment, the suggested solution provides superior load balancing and resource allocation.

The remaining portions of the paper are structured as follows: In Section 2, there is a short summary of some of the academic papers on VM migration-based load balancing and resource allocation. Section 3 of the proposal provides details on the MMH+EACO algorithm technique. Section 4 presents the experimental findings and a discussion of the performance analysis. In Section 5, the findings are compiled.

2. Related Work

In [8], Zhang et al (2014) proposes a method for migrating virtual machines (VMs) based on metadata (Mvmotion) that makes use of memory de-redundant technology between two physical hosts to minimize the amount of data sent during migration. To detect duplicate memory of VMs across two hosts, Mvmotion generates Metadata of RAM using hash-based fingerprints. The transmission of duplicate memory data during migration may be avoided based on the metadata. An experiment shows that Mvmotion may lower the total amount of data sent by 29–97% and the migration time by 16–53% when compared to Xen's default migration strategy.

In [9], Li et al (2015) provided a methodology for placing virtual machines under multi-objective constraint optimization that considers resource and energy waste as the primary objectives. Using the discrete firefly approach, the model is solved. It uses brightness as the objective value and the position of the firefly as the placement result. Darker fireflies in the solution space migrate to brighter fireflies as a result of its movement strategy. Using the discrete method, the continuous position after movement is discretized. The local search method for the best solution is introduced in order to expedite the search process. Compared to previous algorithms, the method uses less energy and wastes fewer resources, according to experimental data from the OpenStack cloud platform.

In [10], Zhixue et al (2017) a new computer paradigm that focuses on the capacity of data and its processing, called cloud computing. Virtualization, distributed data storage, distributed parallel programming, big data management, and distributed resource management are just some of the information and communication technologies that are integrated. Cloud computing has reached a phase of fast expansion after more than ten years of development, and more and more businesses are using its services. Cloud computing's foundational technologies have progressed concurrently. The technologies of the next generation are improving and even replacing those of the previous generation. A new sort of virtualization technology is called a container. Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) design and implementation will alter as a result of container technology's benefits of being lightweight, elastic, and fast. This paper provides a comprehensive examination of container virtualization technology, including a detailed description of its features and functionality. The advantages and disadvantages of container technology are thoroughly discussed, followed by a comparative analysis of its suitable use cases in comparison to virtual machine technology. Furthermore, the paper offers insights into future research directions and development trends in cloud computing virtualization technology.

In [11], Gong et al (2019) discussed a resource allocation technique with adaptive control that responds to changing workloads and resource needs. Because the workloads and resource needs associated with service requests fluctuate over time, service-based systems resource allocation in cloud computing is a crucial strategy for satisfying service requests. Adaptive resource allocation to achieve the QoS with the lowest resource consumption is tough when dealing with constantly variable service requests and resource needs. In cloud computing, services compete for limited resources like CPU and memory while sharing the same resource pool. Focusing on a single resource might result in excessive or inadequate resource allocations, failed service requests, or other problems since services need arbitrary resource combinations. Interference from co-hosted services may lower QoS in cloud computing. Multivariable control allocates resources for different services based on dynamic fluctuating demands and analyzes interference across co-hosted services, providing QoS even if the resource pool is inadequate. The comparison studies demonstrate that, regardless of whether the resource pool is enough, the strategy may satisfy service demands and can increase resource usage.

In [12], Lin et al (2019) scheduling issue is resolved using the ACO method and a well-established multi-objective optimization model for container-based micro service scheduling. The algorithm takes into account the number of requests for micro services as well as the failure rate of the physical nodes in addition to how well the computational and storage resources of the physical nodes are being used. The technique integrates multi-objective heuristic data to increase the selection probability of the optimum route and employs the quality assessment function of the possible solutions to assure the accuracy of pheromone update. The experimental findings demonstrate that the optimization method outperforms competing similar techniques in terms of network transmission overhead, cluster load balancing, and cluster service dependability.

In [13], Su et al (2021) presented the cloud computing and the Ant Colony Optimization Algorithm (ACOA) for work scheduling and resource allocation. The limits of ACOA are first examined, along with the issues with cloud computing's resource allocation and work scheduling. Second, the Q-ACOA, a representation of the ACOA that is optimized to satisfy the anticipated time and expense. Also established are the parameters for the anticipated and pheromone heuristic factors. The Round-Robin scheduling (RR), Min Min (MM), and Time, Cost, and Load Balance-Ant Colony Optimization (TCLB-ACO) algorithms are contrasted with Q-ACOA in the final analysis. The task completion time, the overall data transfer time, the cost of the work, and the happiness of participating users are the accepted assessment indicators in cloud computing task scheduling.

3. Proposed Methodology

In this work, Enhanced Simulated Annealing and Weighted Support Vector Machine algorithm based Cost effective-VM migration (ESA+WSVMCVM) approach is proposed for VM migration based load balancing over cloud environment. The system model, virtualization, load balancing, cost-effective VM migration, and outcomes assessment are all included in the proposed work.

3.1 System model

In this work, system model includes no of VMs, cloud user, CPU, memory usage, no of tasks and no of resources. Cloud infrastructure is specifically engineered to execute a collection of programs and applications in order to accomplish specific tasks. The execution of these programs necessitates the utilization of certain resources within the cloud environment [14]. For the data center's resources to be used as effectively as possible, VM placement is essential. Both the active state and the sleep state are options for each host in the data center. While the host is in the active state, which indicates that it is assigned for execution purposes, it is not assigned to any machines in the sleep state. Based on variables like processor power, memory, and storage capacity, it is divided into several types of finite VMs.

3.2 Virtualization

A hypervisor software layer is to be set up on a physical hardware platform using virtualization technologies. An operating system called a hypervisor controls requests from and responds to VMs while safeguarding a VMH's resources. Type-1 hypervisors directly operating on VMH hardware and type-2 hypervisors running on the VMH OS are defined depending on the application environments. Please see for additional information about hypervisors [16]. Virtual memory (vRAM), virtual hard drives (vHDs), and virtual processors (vCPUs) are the three components that make up a VM, which is an abstract computer. There is no direct access to the actual hardware possible for a user application operating on a VM. The VMH's hypervisor, however, wraps all resources instead. The majority of the time, a VMH runs and oversees many VMs.

A VM is conceptually made up of a disk image file that contains the user data (vHD) and a "configuration file" that describes the parameters of the vCPU, vRAM, and vHD. When the VM is executing, a volatile "memory page" is created in the main memory of the VMH. The aforementioned files are produced when a VM is built on the storage of the VMH. A VM must be deleted, copied, or moved in order to avoid losing its data. The simplicity with which VMs may be moved from one virtual machine to another virtual machine is one of the benefits of virtualization. Migration enables virtual machines to run on many VMH platforms. If the VM is running while the migration is taking place, it must first be halted. The VM memory pages must be transferred from the present VMH's main memory to the target VMH in addition to relocating the VM files. The VM may be reactivated after all the movements are finished.

3.3 Load balancing using Max-Min Heuristic (MMH) algorithm

The Max-Min Heuristic (MMH) technique is used in this study to efficiently balance the load. To prevent any resource from being over- or under-utilized, load balancing includes dividing tasks among the available resources. Data centres, real computers, virtual machines, and any application software are the key resources used for load balancing in cloud computing [15].

Need for load balancing in cloud computing

Tasks and other projects involving the VM are referred to as loads in the world of cloud computing. There are three groups into which these loads are divided:

- Under-loaded
- Over-loaded
- Balanced

In the context of cloud computing, load-balancing algorithms are in charge of attempting to balance out the overall workload. The system's throughput is increased by moving tasks from overloaded nodes to under loaded nodes in order to do this.

According to the volume of work, a load might be low, moderate, or heavy. It is the total number of tasks in the queue. A procedure called load balancing is used to optimize the performance of computational cloud systems so that all of the cloud's processing nodes are used as equitably as possible, increasing throughput and reducing execution times. When a project has to be scheduled for further processing, dynamic load balancing makes scheduling selections. It is known that load balancing may maximize the availability and usage of the whole cloud system while still delivering users with a suitable level of service performance.

All of the shared resources on cloud servers' virtual machines will have their current load calculated. Once each resource's load index has been calculated, a load balancing operation will be started to utilize the resources efficiently and dynamically in order to lower the load value. As a result, the optimum distribution issue that arises from allocating resources to the correct nodes is what led to the development of the Max-Min heuristic method. It is important to balance the loads on the nodes when designing a schedule, and scheduling queue optimization is a key consideration. When deploying tasks, the best physical host must be selected carefully in order to provide load balancing of cloud data centres.

In order to schedule the nodes, the suggested methodology sets off a process for establishing efficient load balancing in the cloud system. The collection of unmapped tasks used by the Max-Min heuristic method is where it starts. The set of nodes with the least amount of load possession should be used for scheduling. The next available VM is assigned the task with the greatest (maximum) makespan by the algorithm. This work is removed from the list of tasks after it has been assigned to the best machine. Until every task has been completed, the procedure iterates. To guarantee that a job may be completed simultaneously with reduced execution durations, the task's maximum execution time should be mapped to the high-capacity machine. The makespan is made better with the use of the Max-Min task scheduling method. On the other side, increased make span guarantees that the burden on each machine in the cloud computing environment is distributed more evenly.

Node Selection with Least Load: The MMH algorithm serves as inspiration for the node selection procedures. The scheduling queues' distance measurements between the nodes. It locates the node with the lowest load values before starting the computation. For the queue's calculation of the least distinct nodes, the node with the least load is taken into account.

The Cartesian distance, which is represented by the following formula, is used to determine the node's distance values.

$$dist = \sqrt{\sum_{j=1}^{k}(n_i - n_j)^2} \qquad (1)$$

The equation Dist, where n_i stands for the chosen node and n_j for the comparing node, displays the distance. The nodes are ordered from the least distinct node to the pivot node after all distance values between the node values have been determined. It is generally accepted that the most relevant scheduling queue is the one at the top of the schedule list. A service resource will be allocated to the best server node in accordance with cloud load balancing, to make it. They may use it to decide intelligently whether to allocate a task to a virtual machine with limited capacity or to simply wait for a large-capacity machine to start up. Additionally, MMH has a more effective scheduling system for all tasks that are waiting.

Utilizing these heuristic methods for task assignment to resources primarily guarantees that all tasks are completed rapidly, hence reducing the overall makespan of the active virtual machine. The fundamental objective of the MMH algorithm is to decrease the waiting time for larger tasks and allocate them to virtual machines with more capacity. The smaller tasks are then given to either idle or underused machines once all the larger tasks have been assigned to their corresponding devices. This practice guarantees equitable distribution of workload among machines, thereby minimizing the occurrence of both task overload and underutilization of machines.

Algorithm 1: MMH algorithm for load balancing

```
1.  Start
2.  For all the tasks in the set G; Ti
3.  For all the resources; Rj
4.  Cij=Eij+rj
5.  While the tasks set G has tasks
6.  Select a task Tk with the highest completion time.
7.  Find the distance between nodes using (1)
8.  sort the resources in the order of completion time
9.  Allocate that task to a resource (Rj), which will
    give the minimum execution time
10. Empty the task Tk from set U.
11. Repeat steps 1 to 7 until the set U is empty
12. Equalize the loads in cloud
13. End
```

The resource R_j that is prepared to carry out a task is represented by r_j in the aforementioned pseudocode. E_{ij} is the execution time, while C_{ij} is the anticipated completion time. It optimizes the utilization of the VM and the CPU. The MMH algorithm efficiently allocates equalization loads to the cloud system, resulting in a more balanced distribution of workload across all machines in the cloud computing environment. Considerably.

3.4 Resource allocation Enhanced Ant Colony Optimization (EACO) algorithm

EACO is the algorithm used in this paper to allocate resources. EACO is a better version of ACO that executes tasks on available resources with the least amount of time required to complete them all while retaining cost and QoS. Data centers serve as resource providers and VMs are real resources in the cloud computing ecosystem. The technique of correctly arranging submitted tasks on available VMs is known as task scheduling in cloud computing. Reduced time complexity and increased resource utilization while retaining cost is the major aim of the proposed EACO. Figure 2 illustrates how the EACO algorithm allocates resources.

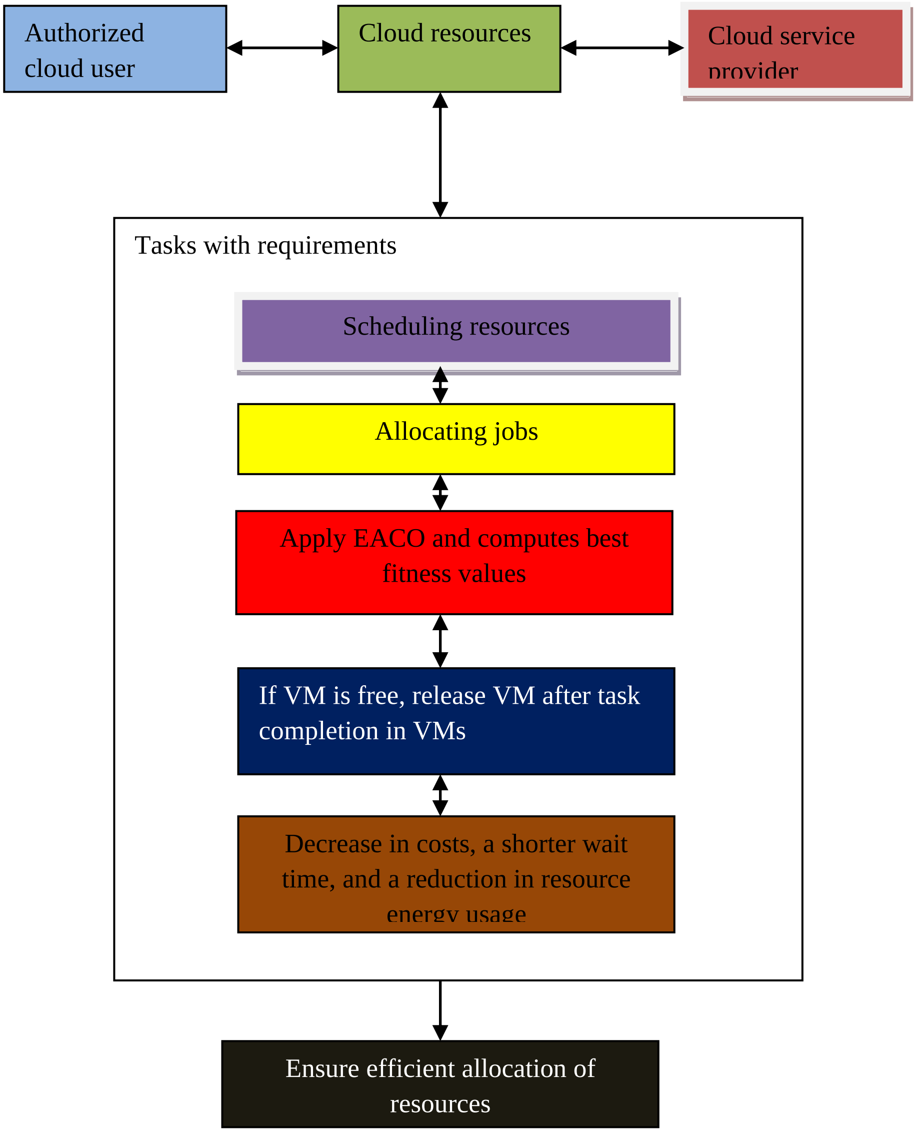

Authorized cloud user
Cloud resources
Cloud service provider
Tasks with requirements
Scheduling resources
Allocating jobs
Apply EACO and computes best fitness values
If VM is free, release VM after task completion in VMs
Decrease in costs, a shorter wait time, and a reduction in resource energy usage
Ensure efficient allocation of resources

Fig 2 EACO algorithm for resource allocation

Finding the best answers to challenging optimization issues is done using the population-based metaheuristic method known as ACO. ACO uses the fundamental principle of ant colonies' foraging behavior in its simulations. Ants are social insects that live in colonies where the survival of the colony is prioritized above the life of the individual members. Ants search for food the furthest from their colony in order to survive. Ants employ a substance known as pheromone in this strategy to locate food with the lowest distance. When ants go from their colony to a food source, they drop pheromones along the route. Ants forage and idly investigate their surroundings. In order to collect food as close to their nest as possible [16] [17]. In comparison to other food sources, ants may move fast when a food supply is located at a short distance. Ants can detect pheromone, which is strongly concentrated on the shortest route. Ants discover the best method for acquiring food for their colony in this manner. When scheduling a lot of tasks on the cloud, ACO is utilized to reduce execution time. Time complexity is still a problem with the typical ACO algorithm. EACO algorithm is presented as a solution to this issue.

A number of data centers are taken into account by the EACO algorithm, and each data center holds a number of hosts. In cloud computing, each host holds a number of VMs. Each VM on a host may have m mips, ram, vmm, and other components. The following is how the suggested algorithm describes each resource using VMi:

$$VM = VM_i \quad where\ i = 1,2,3, \dots .n \qquad (2)$$

The m separate tasks, each with a distinct size, are taken into account by the EACO algorithm. Task j, which includes length, file size, and other information, is how EACO describes these tasks.

$$Task = task_j \text{ where j=1, 2, 3, …….m;} \qquad (3)$$

The process of distributing these m tasks to the n available VMs is known as scheduling in the cloud. An allocation plans that details which task is assigned to which VM is created by the scheduler as part of the scheduling process.

The allocation matrix indicates which tasks are planned on which VM and it is assumed that each VM is given just one task. A binary variable is set to 1 if a task is assigned to a virtual machine; otherwise, it is set to 0. That is shown below.

$$ap_i = \begin{cases} 1 & if\ vm_i\ assigned\ to\ task_j \\ 0 & otherwise \end{cases} \qquad (4)$$

To perform task scheduling using the Optimization) algorithm, the initial step involves the arrival of tasks. Subsequently, a sorting technique is employed to arrange the submitted tasks in a specific order, based on their respective lengths. The sorting algorithm compares and arranges the lengths of tasks available in the task list in ascending order.

This list of ordered tasks is divided into groups at the second stage, where they are then given to an allocation procedure that assigns them to the proper resources. It is verified that the bunch length must be smaller than the whole resource capacity, or resource usage status indication, before forming bunches to obtain maximum utilization. It may be computed using the formula below.

$$uc = \sum_{i=1}^{n} VM_i(MIPS)/n \qquad (5)$$

Therefore, tasks are planned depending on the resources that are available in order to maximize resource usage and minimize time complexity. Tasks are grouped into bunches, and the length of each bunch is decided using the formula below.

$$bunch_{length} = \sum_{l=1}^{bunch\ size} task(length) \qquad (6)$$

Task collection in bunches is when the bunch length violates the total mean of the resources that are accessible, which is determined by:

$$bunch\ length \leq uc \qquad (7)$$

Now that bunches with all the criteria have been created, they are delivered to the allocation task function, which applies EACO to assign tasks to the appropriate VMs. All ants are randomly allocated with beginning VMs for task scheduling in the cloud. Once all tasks were assigned to VMs, they migrated from one VM to the next in search of the best solution. The maximum number of tasks, or tmax, are allocated to iteration with index 1. The fundamental tasks carried out by ACO are pheromone initialization, task selection via virtual machines, and pheromone updating.

The EACO algorithm's primary objective is to decrease task scheduling's overall execution time while preserving costs and maximizing resource usage. Every task's execution time is calculated using ACO. The following is how each task's execution time is determined.

$$ET = (task_i(length)/VM_i(no.ofpes) * VM_i(MIPS) + (task(filesize)/VM_i(BW)) \qquad (8)$$

Where $i = 1,2,3 \dots., n$ and $j = 1,2,3, \dots m$

These tasks and VMs serve as the initialization data for both ACO and pheromone. On VMs, these duties are rotated, and the best course of action is determined. Pheromone is updated each time a better solution is discovered, which implies that each time the scheduling strategy is examined to see whether a new allocation plan provides a better execution time. EACO follows these stages repeatedly to get the best possible outcome. For best resource allocation, the EACO method is shown below:

Algorithm 2: EACO for resource allocation

Input: no.of tasks and no. of resources

Output: Optimal resource selection

```
start
sort all the tasks
collect tasks into bunches
for all task (taskj)∈ $task\ do$
    $if\ \left(task_j < total\ no..of\ \frac{task}{total} no.of\ VM - 1\right) then$
        For all tasks$(task_j) \in task\ do$
            For all tasks$(vm_j) \in VM\ do$
                Compute execution time
            End for
        End for
        Initialize pheromone
        While (not reached to maximum no.of task) do
            Position each task in beginning VM
            For every task $(task_j) \in task\ do$
                Select VM index for next task based on the execution time, pheromone and tasks
            End for
            Update the pheromone
            Update the global pheromone
        End while
    End if
End for
End
Return optimal resources
```

3.5 Cost effective VM migration using Weighted Support Vector Machine (WSVM) algorithm

In this work, cost and migration time of VMs migration is done by using WSVM algorithm. Another crucial factor that has to be taken into consideration is the expense of moving. The time and resources required to store and move VM data, as well as the revenue lost as a result of service interruption, make migration expensive. Decreased migrating VMs are hence preferable for a new VM-VMH assignment.

Flexible resource setup is one of the benefits of cloud computing. From the perspective of the service provider, this feature allows dynamic resource demand expansion and contraction, optimally adapting the cost of leased resources to the projected workload. Similarly, by turning off unused or inactive resources, the infrastructure provider may prevent the inefficient use of resources.

The cost of moving is influenced by a variety of variables, like as (1) the size of each virtual machine's memory and its rate of memory updates, (2) how many virtual machines there are to move overall, (3) the migration-ready network bandwidth, and (4) in the time of transfer, the load of the source and destination servers [18] [19].

Consequently, setting the available network bandwidth to a constant number and doing the following are adequate to estimate the cost of service relocation.

- In terms of (1) and (2) (above), one may calculate the cost of migrating; and,
- Only the migration time should be determined since it influences both the energy overhead and the migration delay.

When doing data analysis and pattern recognition in data mining, the SVM is a supervised machine learning technique. Based on structural risk minimization analysis and statistical learning theory, the SVM determines a maximum margin hyperplane to divide the two classes, a class of positive samples and a class of negative samples, in binary classification.

It is impossible to predict in advance how many virtual machines should be transferred. In this way, the entire migration time may be expressed:

$$t = \sum_{i=1}^{n} t_i$$

where n and t_i are statistically independent for all i, and t_i is the time required to migrate the i-th virtual machine.

The technique being referred to is a robust machine-learning method utilized for the purpose of data classification. Its objective is to identify a linear separating hyperplane that maximizes the margin, thereby effectively segregating data points within a higher-dimensional space. The conventional SVM algorithm is characterized by a lengthy training duration. In order to address the aforementioned concern, the present study introduces a weight-based SVM approach.

In separating the various classes, the WSVM achieves a separation level which is near optimum. Data that can only be separated by nonlinear rules using linear algebra and geometry are implicitly embedded into a high-dimensional feature space using WSVM. It increases the distance of either class

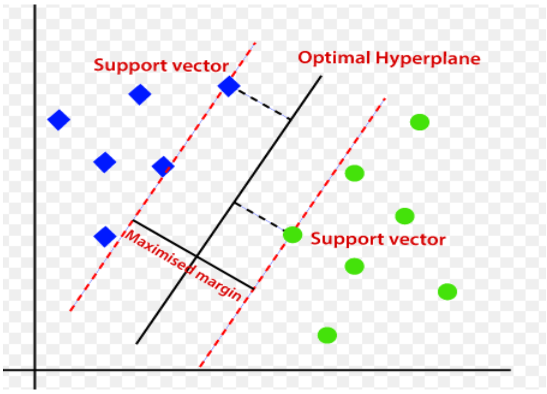


Fig 3 structure of SVM algorithm

In order to make sure that various data points contribute differently to the decision surface's learning, the primary principle behind WSVM is to give each data point a distinct weight based on its relative relevance in the class. If the weights are determined, the training data set changes$\{(x_i, y_i, W_i)\}_{i=1}^{l}$ $x_i \epsilon R^N$, $y_i \epsilon \{-1, 1)$, $W_i \epsilon R$ (9)

where the scalar $0 \leq Wi \leq 1$ is a data point given a weight x_i

The WSVM aims to optimize the margin of separation and reduce the classification error, cost, and time so that strong generalization ability may be attained. This process begins with the building of a cost function. The penalty term is weighted by WSVM in order to lessen the impact of less significant data points even if C is constant and all training data points are handled identically throughout the training phase. As follows is how the restricted optimization issue is stated:

$$Minimize\ \Phi(w) = \frac{1}{2} w^T w + CTE \sum_{i=1}^{l} W_i \xi_i \qquad (10)$$

where C is cost, T is time and E is error rate

Subject to

$$y_i(\langle w, \phi(x_i) \rangle + b \geq 1 - \xi_i, \quad i = 1, \ldots l \qquad (11)$$

$$\xi_i \geq 0 \qquad i = 1, \ldots l$$

To the data point x_i in the formulation above, it applies the weight Wi. The dual formulation emerges as a result

$$W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} (c_i c_j t_i t_j e_i e_j) \qquad (12)$$

subject to

In SVM, the upper bounds of α_i are constrained by a constant while being constrained by weight value-based dynamical bounds $cteWi$ in WSVM. It focuses on leveraging SVM weight values to identify overload and underload patterns and finds VM migration strategies that consume the least amount of energy while maintaining QoS. Fig. 4 displays the proposed system's overall block diagram.

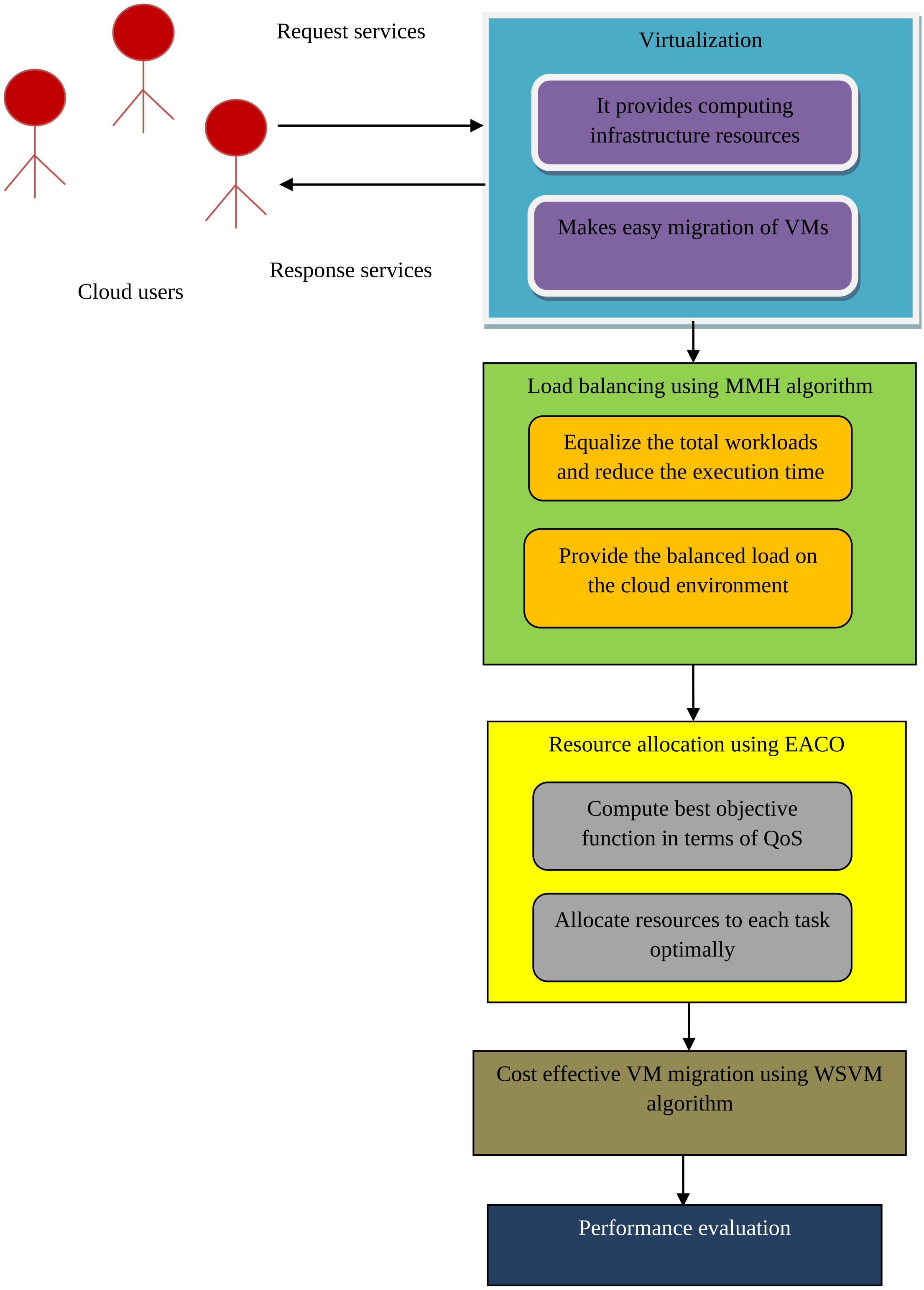

Request services
Virtualization
It provides computing infrastructure resources
Makes easy migration of VMs
Response services
Cloud users
Load balancing using MMH algorithm
Equalize the total workloads and reduce the execution time
Provide the balanced load on the cloud environment
Resource allocation using EACO
Compute best objective function in terms of QoS
Allocate resources to each task optimally
Cost effective VM migration using WSVM algorithm
Performance evaluation

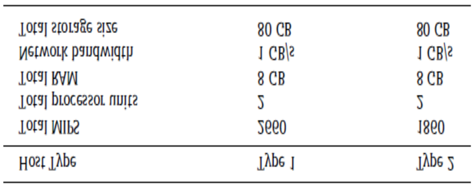
Fig 4 The proposed system's overall block diagram

4. Experimental Result

In this study, a genuine cloud environment, Jnet, is used to accomplish the suggested solutions. Jnet is a private intranet with about 20 different kinds of servers that roughly 200 individuals may access. Jnet was connected to the VMHs and the load balance monitor. Four VMH servers, including three HP ProLiant, made up the experimental setup. CentOS is used by two HP ProLiant ML350 servers and two BL460c blade servers. Using FreeNAS, an HP ProLiant ML110 served as the server for the shared storage. 1 GB of Ethernet connects all of the devices.

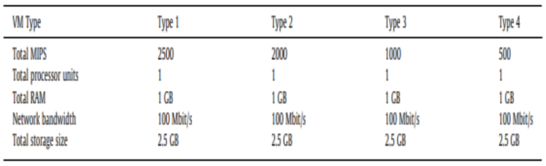
Tables 1 and 2 show, respectively, the physical hosts' and VM specification characteristics [21]. On migration energy and QoS for VMH-VM, the effect of the solution is investigated. Following are the metrics applied to each parameter:

Table 1 Physical Hosts Specification



Host Type	Type 1	Type 2
Total MIPS	2660	1860
Total processor units	2	2
Total RAM	8 GB	8 GB
Network bandwidth	1 GB/s	1 GB/s
Total storage size	80 GB	80 GB

Table 2



VM Type	Type 1	Type 2	Type 3	Type 4
Total MIPS	2500	2000	1000	500
Total processor units	1	1	1	1
Total RAM	1 GB	1 GB	1 GB	1 GB
Network bandwidth	100 Mbit/s	100 Mbit/s	100 Mbit/s	100 Mbit/s
Total storage size	2.5 GB	2.5 GB	2.5 GB	2.5 GB

The existing ACO, GA-GEP, ESA+WSVMCVM algorithms and proposed MMH+EACO algorithm are evaluated to compare the performance metrics. The metrics are considered such as time complexity, cost complexity, throughput, energy consumption and Mean Square Error (MSE) rate.

4.1 Time complexity

When the suggested method runs with reduced time consumption, the system performs better.

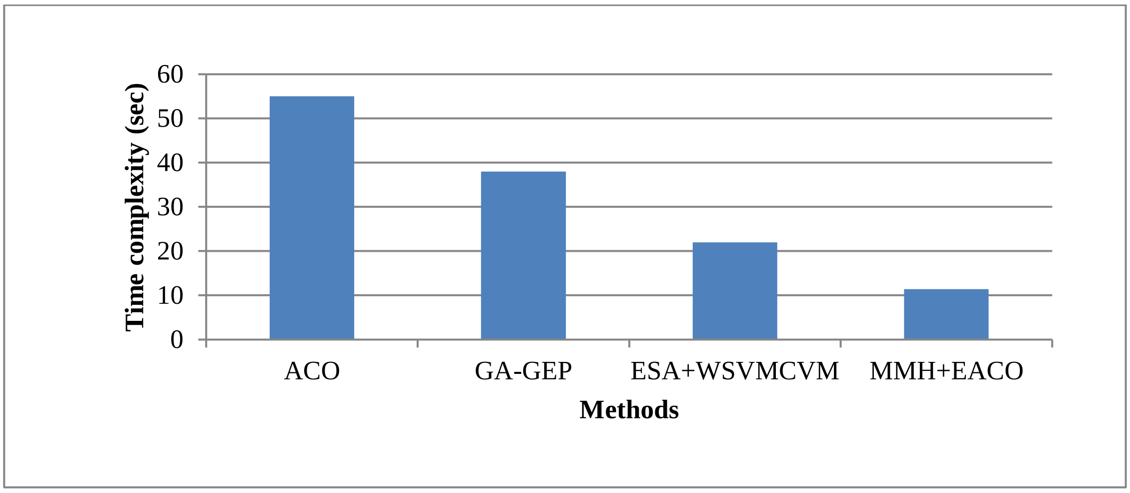


Fig 5 Time complexity

It is clear from the aforementioned Fig. 5 that the comparison measure is assessed in terms of time complexity using both the current and suggested methods. The techniques are displayed on the y-axis, and the value for time complexity is obtained on the x-axis. The suggested MMH+EACO algorithm has reduced time complexity compared to current techniques like ACO, GA-GEP, and ESA+WSVMCVM algorithms. In this proposed research work, load balancing is done by using MMH algorithm and resource allocation is performed via EACO algorithm. The suggested approach accelerates reaction times, which enhances the overall performance of VMH migration. The conclusion drawn from the results is that the suggested MMH+EACO algorithm improves load balancing effectiveness in cloud computing.

4.2 Cost complexity

When the suggested plan offers reduced cost complexity, the cloud is better.

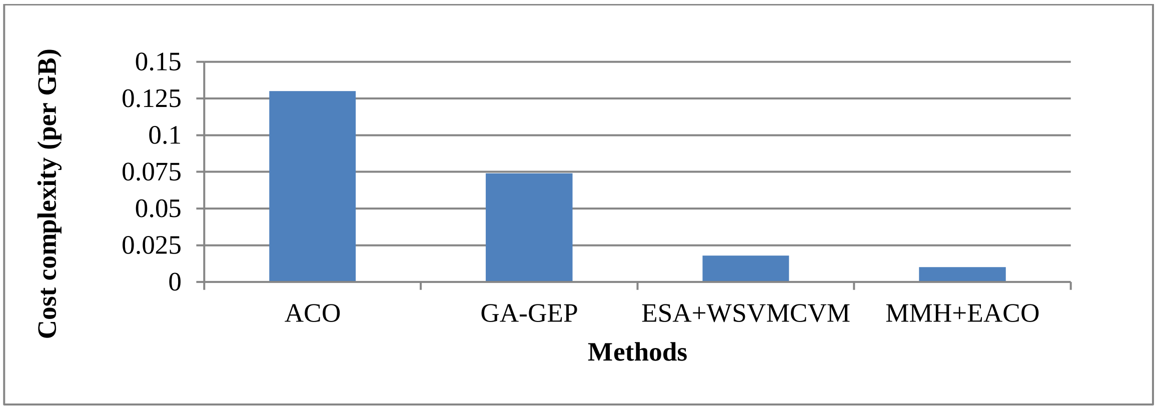


Fig 6 cost complexity

The comparative measure is assessed using both the present and suggested technique in terms of cost complexity, as can be seen in the aforementioned Fig. 6. The techniques are taken as the x-axis, and the cost complexity value is represented as the y-axis. The suggested MMH+EACO algorithm offers reduced cost complexity than the current approaches, such as ACO, GA-GEP, and ESA+WSVMCVM algorithms. In this proposed research work, resource allocation is performed by using EACO algorithm via best pheromone values. The conclusion derived from the results is that the suggested MMH+EACO algorithm improves the performance of VM-VMH in cloud environments.

4.3 *Throughput:*

The throughput of a network or communication channel is the pace at which data packets are successfully transferred through it.

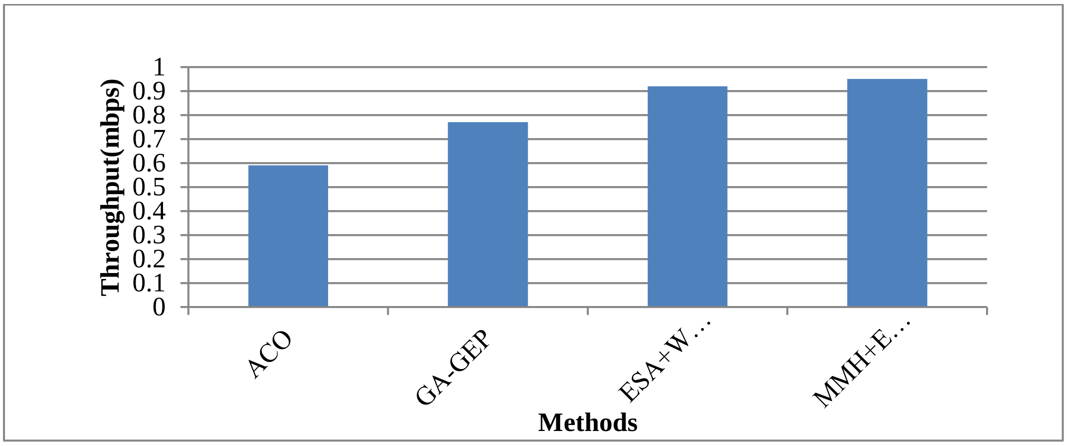


Fig 7 Throughput comparison

Fig 7 illustrates the comparison between the ACO, GA-GEP, ESA+WSVMCVM and MMH+EACO techniques for the throughput metric. It demonstrates that the suggested MMH+EACO scheme has a greater throughput than the current ACO, GA-GEP, and ESA+WSVMCVM techniques. By balancing effective loads across a cloud environment, the suggested solution accelerates the massive volume of data transfer. EACO algorithm is used to dramatically decrease makespan time and enhance resource usage.

4.4 *Energy consumption*

The average amount of energy required throughout time to send, receive, or forward a packet to a network node is referred to as energy consumption.

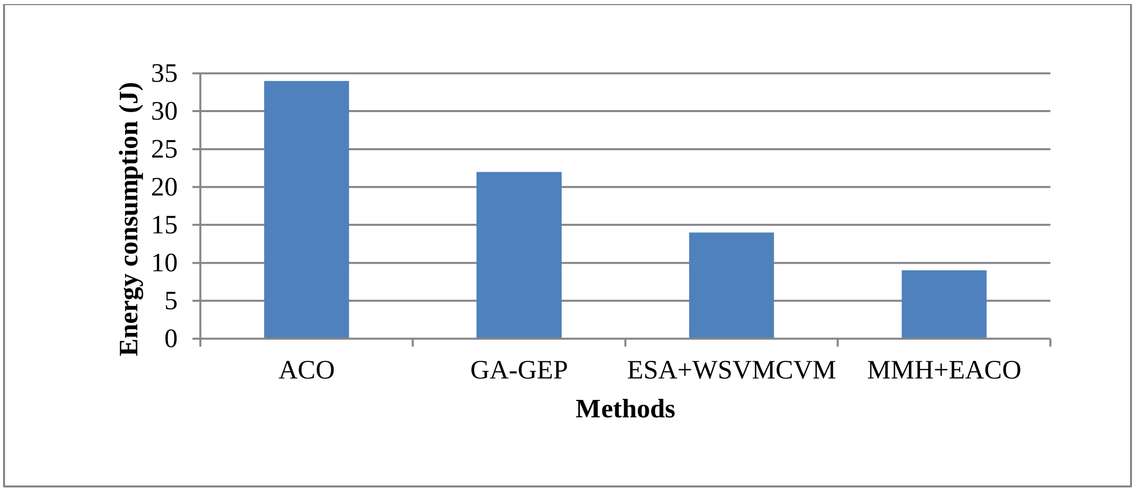


Fig 8 Energy consumption comparison

Using the current ACO, GA-GEP, ESA+WSVMCVM, and suggested MMH+EACO methods, energy usage may be compared in Fig. 8. It demonstrates that although the suggested MMH+EACO technique uses less energy than the current approaches, they both produce increased energy usage. The optimum energy model design in the suggested strategy resulted in decreased energy consumption for massive data transfer.

4.5 *Mean Squared Error (MSE)*

In statistics, the average squared error, or the difference between the estimated values and the actual value, is what an estimator's MSE measures.

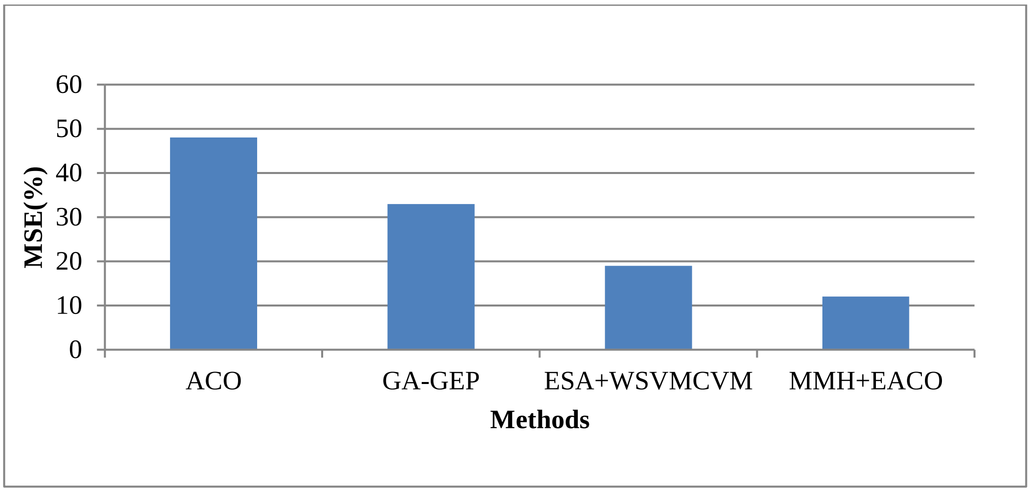


Fig 9 MSE

As shown in Fig. 9, the comparison measure is assessed in terms of MSE using current methodologies. The x-axis is considered to represent the techniques, while the y-axis displays the MSE value. The proposed MMH+EACO method provides lower MSE whereas existing ACO, GA-GEP and ESA+WSVMCVM methods provide higher MSE rate. The conclusion drawn from the results is that the suggested MMH+EACO technique enhances the performance of VM-VMH.

5. Conclusion

In this work, MMH+EACO scheme is proposed to enhance the load balancing and resource allocation during VMH-VM migration over the cloud environment. The number of cloud users, the migration model, and the energy model are used to build the system model initially. After then, MMH is used for load balancing, with an emphasis on choosing the host or VM with the least amount of load. This algorithm is the source of inspiration for MMH. VMs in the scheduling queues are separated from one another using a distance calculation. It locates the VM with the least amount of load values before starting the computation. When calculating the least distinct nodes, the VM with the smallest load is taken into account. The WSVM algorithm is utilized for the purpose of achieving cost-effective VM migration. This algorithm is specifically designed to analyse the types of overload and underload by assigning weight values to support vector machines (SVM). Additionally, it identifies VM migrations that minimize energy consumption while maintaining the QoS. From the findings of the experiment, it can be inferred that the MMH+EACO algorithm, as proposed, exhibits superior cloud performance in relation to existing methods. This superiority is demonstrated through higher throughput, reduced computational complexity, cost complexity, MSE rate, and energy consumption. Also, in the future work, novel encryption algorithm can be developed for dealing with secured data transmission prominently

References

[1] Potluri, Sirisha, and Katta Subba Rao. "Quality of service based task scheduling algorithms in cloud computing." *International Journal of Electrical and Computer Engineering* 7.2 (2017): 1088.

[2] Tsai, Jinn-Tsong, Jia-Cen Fang, and Jyh-Horng Chou. "Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm." *Computers & Operations Research* 40.12 (2013): 3045-3055.’

[3] Ding, D., Fan, X., Zhao, Y., Kang, K., Yin, Q., & Zeng, J. (2020). Q-learning based dynamic task scheduling for energy-efficient cloud computing. *Future Generation Computer Systems*, *108*, 361-371.

[4] Yakubu, Ismail Zahraddeen, et al. "Service level agreement violation preventive task scheduling for quality of service delivery in cloud computing environment." *Procedia Computer Science* 178 (2020): 375-385.

[5] Gamal, Marwa, et al. "Osmotic bio-inspired load balancing algorithm in cloud computing." *IEEE Access* 7 (2019): 42735-42744.

[6] Ebadifard, Fatemeh, Seyed Morteza Babamir, and Sedighe Barani. "A dynamic task scheduling algorithm improved by load balancing in cloud computing." *2020 6th International Conference on Web Research (ICWR)*. IEEE, 2020

[7] Xiao, Zhen, Weijia Song, and Qi Chen. "Dynamic resource allocation using virtual machines for cloud computing environment." *IEEE transactions on parallel and distributed systems* 24.6 (2012): 1107-1117.

[8] Zhang, Zhenzhong, et al. "Mvmotion: a metadata based virtual machine migration in cloud." *Cluster Computing* 17.2 (2014): 441-452

[9] Li, Xiao-Ke, et al. "Virtual machine placement strategy based on discrete firefly algorithm in cloud environments." *2015 12th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*. IEEE, 2015

[10] Zhixue, W. U. "Advances on virtualization technology of cloud computing." *Journal of Computer Applications* 37.4 (2017): 915

[11] Gong, Siqian, et al. "Adaptive multivariable control for multiple resource allocation of service-based systems in cloud computing." *IEEE Access* 7 (2019): 13817-13831

[12] Lin, Miao, et al. "Ant colony algorithm for multi-objective optimization of container-based microservice scheduling in cloud." *IEEE access* 7 (2019): 83088-83100

[13] Su, Yingying, Zhichao Bai, and Dongbing Xie. "The optimizing resource allocation and task scheduling based on cloud computing and Ant Colony Optimization Algorithm." *Journal of Ambient Intelligence and Humanized Computing* (2021): 1-9.

[14] Annadanam, Chakravarthy Sudarshan, Sudhakar Chapram, and T. Ramesh. "Intermediate node selection for Scatter-Gather VM migration in cloud data center." *Engineering Science and Technology, an International Journal* 23.5 (2020): 989-997

[15] Maipan-Uku, J. Y., et al. "Max-average: An extended max-min scheduling algorithm for grid computing environtment." *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)* 8.6 (2016): 43-47

[16] Peng, Huijun, et al. "An improved feature selection algorithm based on ant colony optimization." *IEEE Access* 6 (2018): 69203-69209.

[17] Paniri, Mohsen, Mohammad Bagher Dowlatshahi, and Hossein Nezamabadi-Pour. "MLACO: A multi-label feature selection algorithm

[18] D'Agostino, Daniele, et al. "Combining edge and cloud computing for low-power, cost-effective metagenomics analysis." *Future Generation Computer Systems* 90 (2019): 79-85.

[19] Singh, Yashaswi, Farah Kandah, and Weiyi Zhang. "A secured cost-effective multi-cloud storage in cloud computing." *2011 IEEE conference on computer communications workshops (INFOCOM WKSHPS)*. IEEE, 2011.

[20] Rustam, Zuherman, Jacub Pandelaki, and Arga Siahaan. "Kernel spherical k-means and support vector machine for acute sinusitis classification." *IOP Conference Series: Materials Science and Engineering*. Vol. 546. No. 5. IOP Publishing, 2019

[21] Elshabka, Mohamed A., et al. "Security-aware dynamic VM consolidation." *Egyptian Informatics Journal* (2020)

[22]

[23] Arularasan, A. N. ., Aarthi, E. ., Hemanth, S. V. ., Rajkumar, N. ., & Kalaichelvi, T. . (2023). Secure Digital Information Forward Using Highly Developed AES Techniques in Cloud Computing. International Journal on Recent and Innovation Trends in Computing and Communication, 11(4s), 122–128. https://doi.org/10.17762/ijritcc.v11i4s.6315

[24] Sherje, D. N. . (2021). Content Based Image Retrieval Based on Feature Extraction and Classification Using Deep Learning Techniques. Research Journal of Computer Systems and Engineering, 2(1), 16:22. Retrieved from https://technicaljournals.org/RJCSE/index.php/journal/article/view/14

[25] Kshirsagar, P. R., Reddy, D. H., Dhingra, M., Dhabliya, D., & Gupta, A. (2022). A review on application of deep learning in natural language processing. Paper presented at the Proceedings of 5th International Conference on Contemporary Computing and Informatics, IC3I 2022, 1834-1840. doi:10.1109/IC3I56241.2022.10073309 Retrieved from www.scopus.com