

# An Optimal Energy Preserved In Dynamic Load Balancing Using Monarchy Butterfly Optimization

R. Sowmini @ Mahalakshmi <sup>1</sup>, Mr.D.Vijayakumar B.E.,M.S., <sup>2</sup>

<sup>1</sup> PG Student, Department of Computer Science and Engineering, National Engineering College, K.R Nagar Kovilpatti, Tuticorin, Tamil Nadu, India. [2151007@nec.edu.in](mailto:2151007@nec.edu.in)

<sup>2</sup> Assistant Professor (Sr. Grade), Department of Computer Science and Engineering, National Engineering College, K.R Nagar, Kovilpatti, Tuticorin, Tamil Nadu, India. [dvk@nec.edu.in](mailto:dvk@nec.edu.in)

DOI: 10.47750/pnr.2023.14.03.318

## Abstract

Cloud computing is the computation of the future in the present situation. Cloud computing makes resources available to clients on demand. Uneven task assignment to processors causes imbalance, i.e., some processors are loaded while others are loading control. Load balancing's purpose is to transparently shift the load from an overloaded process to a process that is loaded. The Critical issue in cloud computing is load balancing. Workloads must be transferred between nodes in the cloud network to achieve high performance, a short response time, and a resource utilization ratio. In the following parts, we will discuss load-balancing approaches and the suggested work of our load-balancing system. Utilizing the Cloudsim simulator, performance is examined. Simulation results show that the suggested method evenly balances the workload among all accessible virtual machines without under or overusing them.

**Keywords-** Load balancing, Monarchy Butterfly optimization and energy consumption.

## I. INTRODUCTION

It was challenging to assume that happened a short time ago. Today's industry may seem obvious and definite in its speculations. The genesis of the Cloud has been puzzling due to different definitions. It sounds like a baby giving birth, much like "Christ." In contrast to using personal drives of private systems, cloud computing stores and acquires data and logical instructions via the web. Said the Cloud represents the Internet. The term "cloud computing" refers to a type of computer technology that mimics the architecture of a system while utilizing the processing capacity of multiple interconnected computers. Using multi-tenant setups made possible by virtualization, the Cloud is a new generation of IT service delivery from a remote location, either through the Internet or an intranet.

SaaS, PaaS, IaaS, DBaaS, and many more service models exist in the Cloud computing industry. Cloud computing includes three main service model types: SaaS, PaaS, and IaaS, each of which has prominence. Through an online service approach, virtualized computing resources are made available for IAAS. The tools are provided through a cloud computing approach to create applications only for PAAS. The hosting of software so that users can access it is a type of cloud computing service model for SAAS. Virtualization is the key to maximizing server resources, and it is frequently used software formerly known as VMware. Cloud computing architecture is constructed with several cloud components. The Cloud is divided into two categories to manage the enormous protocol: Front - Back. Front-End: A network, typically the Internet, connects the functional ends. Let's look deeper to comprehend it more: The front end side end users can connect to the backend via user interfaces like a web browser, this combines the client's network and apps needed to access the Cloud. Back-End - The "cloud," personal devices, centralized servers, and data backup devices, literally handle as the main backend architecture. Importantly, it is the back end's responsibility. To overcome some issues in cloud computing, we are going for load balancing with virtual machines. Through the implementation, we obtained energy efficiency.

It is a common practice to divide computational workloads between two or more computers, known as load balancing. To distribute network traffic among numerous servers, I know that this is where it all on the Internet, load balancing is frequently used. As a result, each server has less stress and becomes more effective, which speeds up performance and lowers latency. The majority of Internet applications cannot run successfully without load balancing. The research describes the load balancing issue to prevent task execution delays and Consume energy. The relevant research on load balancing and energy consumption issues is shown in Section II. The system design and load-balancing strategy are presented in Section III. The simulation tool and the outcomes of the implementation are covered in Section IV. In Section V, we wrap things up and discuss our next steps.

## II. RELATED WORKS

The study is to attain optimal energy preservation while addressing various requests. Load balancing is widely used to divide network traffic among multiple servers. Migrating and adjusting operators are detected using the Monarchy Butterfly Optimization Algorithm (MBO) to reduce burden. As a result, performance metrics, comparisons of different optimization techniques, and the operation of MBO are surveyed in this segment.

Muhammad Ziad Nayyer et al.[1] proposed that IoT-based solutions considering future ICT infrastructure are becoming increasingly popular. Due to this, the demand for edge devices and digital equipment has increased dramatically. To reduce resource congestion, an edge load-balancing approach is necessary.[2]. L.-H. Hung suggested a two-stage genetic load-balancing approach for virtual machine hosts (VMHs) in cloud computing. The experiment results reveal that technique outperforms traditional strategies such as heuristics and statistical regression. More and more automobiles are sending and receiving data over the Internet. Data acquired via radio waves is only occasionally correct. Edge servers are assigned in the same way as base stations get data from cars. It allows the edge server to aggregate vehicle data without being affected by radio traffic circumstances[4].[5] To solve the load balancing issue, provide Deep Reinforcement Learning with Parallel Particle Swarm Optimization (DR LPPSA), an effective scheduling technique in this work. According to the results of the experiments, the suggested scheduling technique raises when the task set is smaller.

Load balancing is still a challenging subject. It should create a suitable access schedule to enable the system to distribute work across all resources. A multi-objective task scheduling optimization combining a Q-learning algorithm and the Artificial Bee Colony Algorithm is suggested as a standalone job scheduling technique.[6]. DIAL is an interference-aware load-balancing framework that cloud users can employ independently of provider assistance. The load is then dynamically transferred away from vulnerable VMs while remaining within the application's tail latency SLOs using estimates of colocated burden. DIAL may reduce tail latencies by up to 70% compared to current techniques.[7].

A dynamic resource allocation system is critical in today's cloud computing environment. In this research, we suggest a resource allocation technique that, while respecting performance constraints, optimizes profit under a specific combination of costs and revenues. We conducted numerous MATLAB simulations and AWS-based tests that revealed insights and benefits over conventional guidelines[8]. Although scattered computing's challenges have mostly been solved by cloud computing, several issues could still arise. Significant problems with cloud computing include security and load balancing. The mechanism for ensuring that each cloud node is properly load-balanced is proposed in this work as EMAMBO. It is reasonable to conclude that the suggested system performs better than several benchmarked current systems based on the metrics assessed[10]. A set of requests is split up into a set of resources using the load-balancing technique to spread the burden. It is a significant obstacle in cloud computing and is acknowledged as an NP-hard problem. Despite the significance of nature-inspired meta-heuristic strategies for solving the load-balancing issue in the cloud environment, more comprehensive and in-depth research needs to review and study the critical, crucial subjects in this field.

Additionally, to address the load-balancing issue, Nature-inspired meta-heuristic algorithms have features and drawbacks that have been researched, and their primary problems have been taken into account to suggest future solutions that are more successful[11]. The fundamental goal of these strategies was to balance the extra burden that occurs during the execution phase[12]. The time required for execution will be directly impacted by the determination of whether additional virtual machines are needed based on the incoming requests. To avoid delays and the creation of any faults, the load placed on the system should also be efficiently balanced. Additionally, by idling the less-used and idle virtual machines during execution, it should carry out all of the tasks mentioned above in an energy-consuming manner.

Cloud service providers construct data centers (DC) worldwide to accommodate the rising demand for computing and storage resources. These DCs store tens of thousands of servers for information technology. This work created a load-balancing system based on the adaptive cat swarm optimization (ACSO) method to deal with optimization problems. The suggested plan's effectiveness is assessed using various value measures, and its performance is contrasted with competing approaches[13]

Numerous load balancing techniques were succinctly discussed in[15], which provides a clear concept of how to pick the approximation approach for the issue.

The Internet of Things (IoT), where everything is connected to it, is one of the most significant technologies to have been deployed recently. It can enhance the performance of the healthcare system through IoT and cloud computing integration. IoT-based healthcare system design and implementation and load-balancing issues in the Cloud are the two main components of this effort[16].

Velde et al. demonstrates the scheduling framework and optimization challenges in cloud architecture, primarily focused on QoS. The Service Level Agreement (SLA) at Level 1 forms the basis of the "planning algorithm," which contains detailed details about the distribution and assignments to the cluster[17]. This research introduces a novel adaptive Pbest discrete PSO static load balancing method. The optimum positions for each particle are updated using reasonable solutions from an external archive. Simulation studies using arbitrary synthetic jobs are carried out on the MATLAB and CloudSim platforms. Our proposed approach significantly increased the swarm's convergence and variety.[20].

## III. PROBLEM FORMULATION

The model is mainly made up of the following elements: There are three sections: The Data Center Controller performs the Node Detection Phase. (ii) The Optimization phase collects more information about the server's load. It makes decisions on task distribution among servers based on parameters. (iii) The Load Allocation Phase allocates, reallocates, or removes any task from the server based on priority, thereby reducing communication delay and execution time

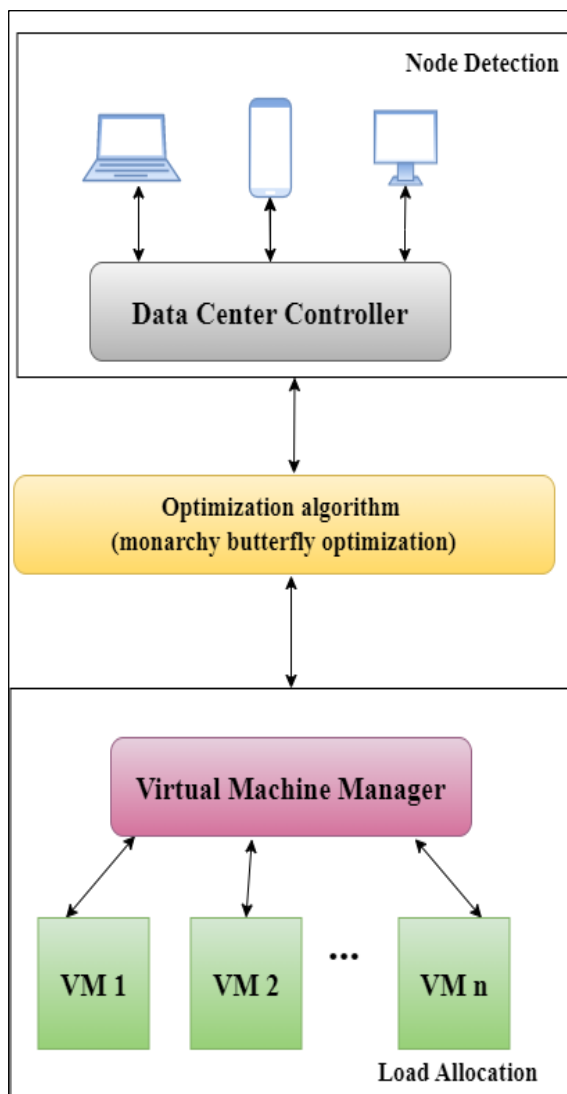


FIG 3.1 Load Balancing optimization model

## A .NODE DETECTION PHASE

The distributed operating system generates tokens when it is idle. When the system first boots up, A token manager is a distributed operating system component in charge of token generation. The Token identifies the state of the machine. The token manager also decides how many tokens will be released. When this Token comes into contact with an idle computer, the Token is accepted. In other words, if the number of jobs in the ready queue equals the threshold, the node is said to be idle. Suppose the number of available positions is. Suppose the line is more significant than the node is considered overcrowded. Every node periodically checks its state and will be designated.

### THRESHOLD TRIGGER:

- Active client rebalances threshold:

50% is the established threshold. On a cluster member, the current active load.

- Rebalance threshold for standby clients:

A cluster member's standby workload. The cutoff point is set at 75%.

- Inequivalent threshold:

The gap between the loads on the cluster nodes with the highest and lowest loads. The threshold is set at 5%, meaning that the managed devices' loads must differ by at least 5%.

## B. OPTIMIZATION PHASE

Reducing production costs or increasing production efficiency may be the design goal while optimizing a design. An optimization algorithm compares several solutions iteratively until an ideal or suitable answer is identified.

### MONARCHY BUTTERFLY OPTIMIZATION

Monarch butterfly migration behavior is included into the following rules to address various optimization difficulties.

1. Monarch butterflies can only be found in Areas 1 or 2. That is, the monarch butterfly population is made up entirely of monarch butterflies from Areas 1 and 2.

2. The migration operator creates each young monarch butterfly individual from a monarch butterfly in Area 1 or Area 2. An overview of the migratory and butterfly adjustment operators is given in the following subcategories.
3. An elderly monarch butterfly will die once a young is born to maintain the population. In the MBO approach, it can be performed by replacing its parent with a newly produced one if it is more suited than its parent. However, if the newly formed do not outperform its parent in terms of fitness, it is likely to be eliminated. In this case, the parent is safe and uninjured.
4. Individual monarch butterfly adults with the highest fitness move on to the next generation automatically and cannot be changed by any operators. This ensures that the monarch butterfly population's quality or efficacy stays unchanged as the population grows.

### MIGRATION OPERATOR:

The movement of individual I in subpopulation one is affected by the location of others in subpopulation 1 or 2, It started adjusting the ratio p can counter. The migration operator aims to improve information exchange inside the subpopulation and between the two subpopulations1.

The following numerical expression describes how subpopulation 1's individual I moves on the kth dimension.

$$x_{i,k}^{t+1} = x_{r1,k}^t \quad \text{if } r \leq p \quad \dots\dots\dots (i)$$

$$x_{i,k}^{t+1} = x_{r2,k}^t \quad \text{else } \dots\dots\dots(ii)$$

where  $x_{i,k}^{t+1}$  denotes  $x_i$  kth dimension at generation  $t+1$ . From subpopulations 1 and 2, respectively, the parameters  $r1$  and  $r2$  are integer indexes that were selected at random. The value of the parameter  $r = \text{rand} * \text{peri}$  is a real number with a range of  $[0, 1]$  called rand, and the term peri stands for the migration period.

### BUTTERFLY ADJUSTING OPERATOR :

The movement is defined for an individual I in subpopulation2 in terms of the current global best individual, a randomly chosen person in subpopulation2, and Levy flight, which may be calculated using the butterfly adjustment rate BAR and modifying ratio p. Three elements are mainly considered while employing a butterfly adjusting operator: (1) The global optimum's influence on the social model. (2) The influences of other people on a random person's cognition. (3). introduced the Levy flight to broaden the search area and boost population variety.

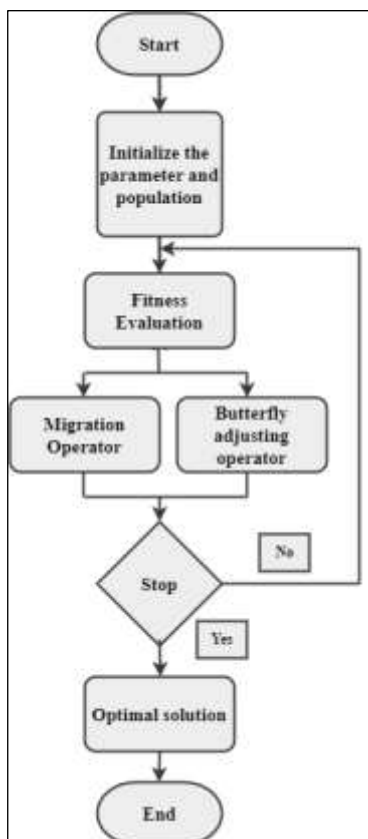


FIG 32 Flow diagram of MBO

### C. LOAD ALLOCATION PHASE : VIRTUAL MACHINE:

A virtual environment that serves as a virtual computing device with its CPU, cache, network interface, and storage (located off- or on-premises) is called Virtual Machine (VM). The resources are separated from the hardware and furnished correctly so that the virtual machine (VM) can use

them. It is termed Virtualization Process. These resources are necessary for virtual environments, and the hypervisor software divides the existing resources. Most businesses use hypervisors, which can be installed directly on a server or on top of an operating system like a laptop, to virtualize their infrastructure. The physical resources are evenly divided by hypervisors so it can use virtual environments. Resources are divided between the many virtual and natural worlds as necessary. Users can perform calculations and engage in virtual world interaction (typically called a guest machine or virtual machine). Operating as a single data file, the virtual machine. Like any digital file, it may be opened on either machine and is supposed to work the same way when transferred between them.

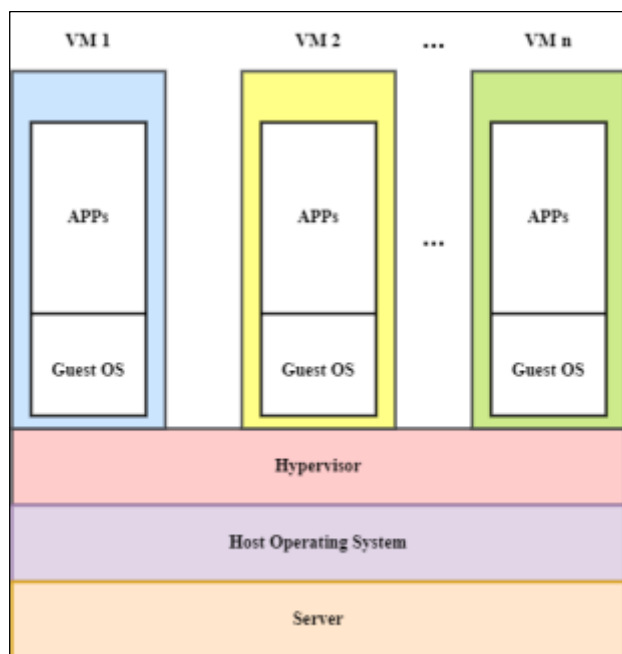


FIG 3.3 virtual machine architecture

### ALGORITHM OF MBO

**Begin**

**Step 1: Initialization.**  
 the population of NP monarch butterfly individuals, and the maximum generation MaxGen, at random.  
 peri, p, BAR, and Smax

**Step 2: While ( $G < \text{MaxGen}$ )**  
 Non-ascending sorts the population in accordance with level of fitness.  
 Create a subpopulation with higher-quality individuals and subpopulation2 on the rest.  
 Create a subpopulation's offspring solution using equation (1).  
 Create a subpopulation 2 offspring solution using equation (2).  
 The newly created subpopulations should be merged into the original population.  
 Find the specimen who is best right now.

**End while**

**Step 3: Find the ideal specimen.**

**End.**

## PROBLEM DESCRIPTION :

As an example, have a look at cloud  $c$ , which consists of  $n$  physical machines or any physical machine made up of  $m$  VMs:[15]

$$C = \{M1, M2, Mn\} \dots\dots(1)$$

Additionally, each physical machine is made up of several virtual machines in the following ways:[15]

$$Mn = \{VM1, VM2, VMm\} \dots\dots(2)$$

The first virtual machine is denoted by VM1, while the last is denoted by VMm (VM).

Similar to how users ( $U_i$ ) make up the Cloud, users can be identified by their function as follows:

$$U_i = \{f1, f2, \dots, fj\} \dots\dots(3)$$

The objectives of this process are to minimize expenses and energy use, maximize resource utilization, and ensure an even load distribution across all VMs in a cloud environment. Loads need to be balanced to achieve an effective planning process. Without it, the system uses time and effort to do its tasks. This study proposes an adequate and balanced multi-purpose procedure built on the MBO algorithm to overcome this problem.

## IV. EXPERIMENTAL SETUP:

CloudSim, an open-source framework, simulates cloud computing infrastructure and applications. The CLOUDS Lab built it, and it is written in Java. It mimics and reproduces a cloud computing environment to test a theory before developing software, allowing for regular testing and discoveries.

This toolset enables the following:

- In a controlled and repeatable environment, test the application services.
- Before deploying apps in a real cloud, optimize the system's bottleneck.
- Experiment on a simulated infrastructure with various workload combinations and resource performance situations to develop and test application provisioning techniques.

## FEATURES :

- Servers, hosts, and large-scale virtualized data centers.
- Policies for provisioning hosts to virtual machines that are customizable.
- Computing resources that are energy efficient.
- Applications for message-passing in data center networks.
- Dynamic simulation of entity addition with simulation pause and resume.
- Allocation and provisioning rules that the user sets.

## SIMULATION PARAMETERS:

Simulation Parameters	Values
Number of Physical Machines (PM)	4
Data Center	2
Number of Virtual Machine	60
Number of Task / Node	100-700
Bandwidth	100 mps

## V. RESULT:

The following measures were found to be considered for monitoring the performance of MBO with the previously indicated methodologies.

1. Response Time: The response time is the overall amount of time it takes the system to complete a task.
2. Throughput: The number of processes completed in a given time unit.
3. Migration time: The time spent transferring a task from an overloaded node to a node with fewer loads.
4. Transmission Time: The amount of time it takes for a task to reach a specific VM. It is also determined by the task's size and the VM's bandwidth.
5. MakeSpan: The makespan is the time it takes to complete the previous job.

We first assess overall performance over a range of task counts using the total makespan acquired. Fig.4.1 compares the makespan for various nodes counts ranging from 100 to 700. The graph contrasts the outcome with the Ant Colony Optimization(ACO) and Cuckoo Search Algorithm (CSA). The suggested strategies accomplish multiple jobs in less time than the others.

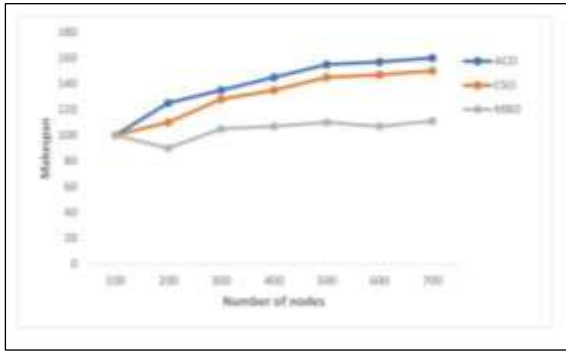


FIG 4.1 Make Span

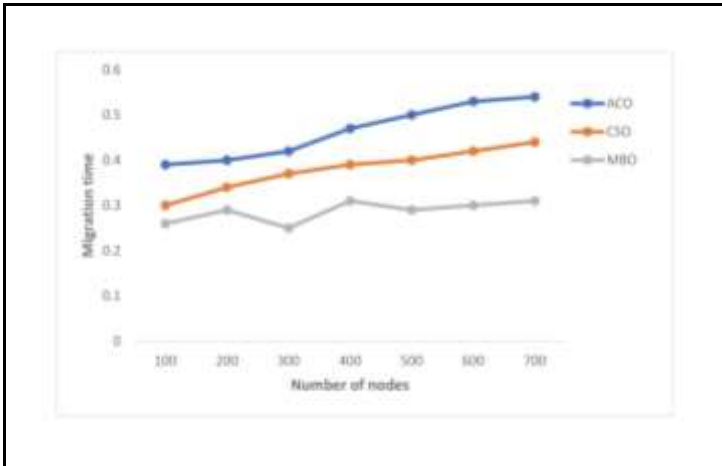


Fig4.2 Migration Time

Compared to the benchmarks CSA and ACO, the migration time of the proposed work MBO is reduced by 7% and 10%, respectively, in Fig 4.2. Compared to the current job, The proposed work is determined to work better by minimizing the time required for migrating the nodes.

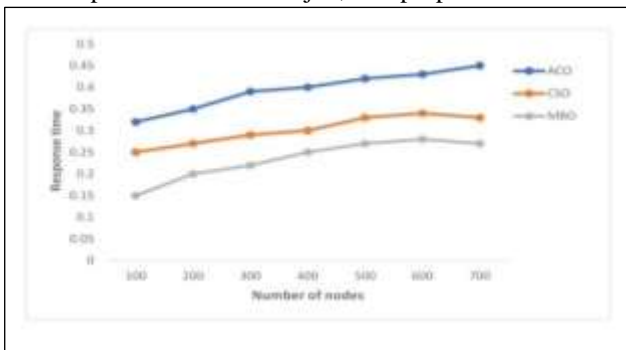


Fig 4.3 Response Time

Figure 4.3 depicts the response time by the suggested work, which has been studied under various node counts. The response time is when it takes the system to complete a task. Compared to the current approaches, the Cuckoo Search Algorithm (CSA) and Ant Colony Optimization (ACO), the suggested MBO algorithm's response time is increased to a greater extent. It varies with the number of nodes. According to Fig4.3, the suggested algorithm reduces response time by 3% and 6%, respectively.

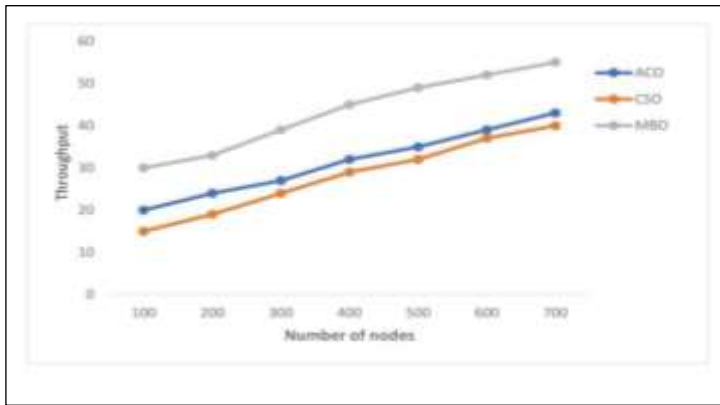


Fig 4.4 Throughput

Figure 4.4 displays the throughput achieved by the proposed work, which is further analyzed in relation to the number of nodes and the performance is compared to the chosen benchmark. The throughput is critical in determining the efficiency of the suggested design. In Fig. 4.4, the suggested system MBO's throughput is observed to be increased by 8%, 10%, against the benchmarks CSA and ACO as the number of workloads increases.

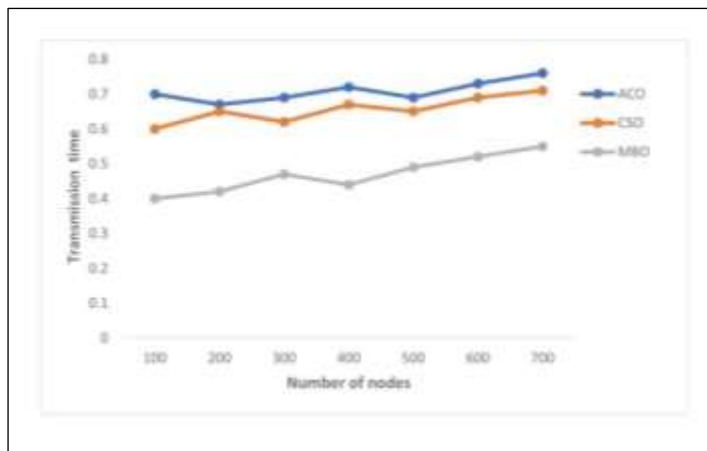


Fig 4.5 Transmission Time

Compared to the benchmarks CSA and ACO, the migration time of the proposed work MBO is reduced by 6% and 9%, respectively, in Fig 4.5. Compared to the current job, the proposed work is determined to work better by minimizing the time required for transmitting the number of nodes.

## V. CONCLUSION

The monarchy butterfly optimization method in cloud computing is proposed in this article employing dynamic load balancing. By estimating each VM's fitness, this strategy considered the selection of appropriate VMs. The proposed method aids in the load-balancing of tasks using existing system resources.

We have put forth a method to assign the best resources to incoming requests from users, regardless of their number, to address one of the problems in the cloud environment. We can also use the monarchy butterfly optimization to reduce the burden. The simulation output demonstrates that the suggested strategy performs better than comparing existing algorithms.

## References

- [1]. M. Z. Nayyer et al., "LBRO: Load Balancing for Resource Optimization in Edge Computing," in *IEEE Access*, vol. 10, pp. 97439-97449, 2022, doi: 10.1109/ACCESS.2022.3205741.
- [2]. L. -H. Hung, C. -H. Wu, C. -H. Tsai and H. -C. Huang, "Migration-Based Load Balance of Virtual Machine Servers in Cloud Computing by Load Prediction Using Genetic-Based Methods," in *IEEE Access*, vol. 9, pp. 49760-49773, 2021, doi: 10.1109/ACCESS.2021.3065170.
- [3]. S. Souravlas, S. D. Anastasiadou, N. Tantalaki and S. Katsavounis, "A Fair, Dynamic Load Balanced Task Distribution Strategy for Heterogeneous Cloud Platforms Based on Markov Process Modeling," in *IEEE Access*, vol. 10, pp. 26149-26162, 2022, doi: 10.1109/ACCESS.2022.3157435.
- [4]. K. Hosono, A. Maki, Y. Watanabe, H. Takada and K. Sato, "Implementation and Evaluation of Load Balancing Mechanism With Multiple Edge Server Cooperation for Dynamic Map System," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7270-7280, July 2022, doi: 10.1109/TITS.2021.3067909.
- [5]. A. Pradhan, S. K. Bisoy, S. Kautish, M. B. Jasser and A. W. Mohamed, "Intelligent Decision-Making of Load Balancing Using Deep Reinforcement Learning and Parallel PSO in Cloud Environment," in *IEEE Access*, vol. 10, pp. 76939-76952, 2022, doi: 10.1109/ACCESS.2022.3192628.

- [6].B. Kruekaew and W. Kimpan, "Multi-Objective Task Scheduling Optimization for Load Balancing in Cloud Computing Environment Using Hybrid Artificial Bee Colony Algorithm With Reinforcement Learning," in *IEEE Access*, vol. 10, pp. 17803-17818, 2022, doi: 10.1109/ACCESS.2022.3149955.
- [7].S. A. Javadi and A. Gandhi, "User-Centric Interference-Aware Load Balancing for Cloud-Deployed Applications," in *IEEE Transactions on Cloud Computing*, vol. 10, no. 1, pp. 736-748, 1 Jan.-March 2022, doi: 10.1109/TCC.2019.2943560.
- [8].M. Shifrin, R. Mitrany, E. Biton and O. Gurewitz, "VM Scaling and Load Balancing via Cost Optimal MDP Solution," in *IEEE Transactions on Cloud Computing*, vol. 10, no. 3, pp. 2219-2237, 1 July-Sept. 2022, doi: 10.1109/TCC.2020.3000956.
- [9]. Siddiqua ,Jahangeer, Sundari, "A Study on Different Deep Learning Algorithms Used in Deep Neural Nets: MLP SOM and DBN". *Wireless Personal Communication* (2022). <https://doi.org/10.1007/s11277-022-10079-4>.
- [10].Kaviarasan, R., P. Harikrishna, and A. Arulmurugan. "Load balancing in a cloud environment using enhanced migration and adjustment operator based monarch butterfly optimization." *Advances in Engineering Software* 169 (2022): 103128.
- [11].Milan, Sara Tabaghchi, et al. "Nature inspired meta-heuristic algorithms for solving the load-balancing problem in cloud environments." *Computers & Operations Research* 110 (2019): 159-187.
- [12].Thakur, Avnish, and Major Singh Goraya. "RAFL: A hybrid metaheuristic based resource allocation framework for load balancing in a cloud computing environment." *Simulation Modelling Practice and Theory* 116 (2022): 102485.
- [13].Balaji, K., P. Sai Kiran, and M. Sunil Kumar. "An energy efficient load balancing on cloud computing using adaptive cat swarm optimization." *Materials Today: Proceedings* (2021).
- [14].Muteeh, Arfa, Muhammad Sardaraz, and Muhammad Tahir. "MrLBA: multi-resource load balancing algorithm for cloud computing using ant colony optimization." *Cluster Computing* 24.4 (2021): 3135-3145.
- [15].Mubeen, Aroosa, et al. "Alts: An Adaptive Load Balanced Task Scheduling Approach for Cloud Computing." *Processes* 9.9 (2021): 1514.
- [16].Abdulhammed, Omar Younis. "Load balancing of IoT tasks in the cloud computing by using a sparrow search algorithm." *The Journal of Supercomputing* 78.3 (2022): 3266-3287.
- [17].Velde, Venkateshwarlu, Kiran Enumala, and Krishna Bandi. "Optimized Adaptive load balancing algorithm in cloud computing." *Materials Today: Proceedings* (2021).
- [18].Alqahtani, Fayez, Mohammed Amoon, and Aida A. Nasr. "Reliable scheduling and load balancing for requests in cloud-fog computing." *Peer-to-Peer Networking and Applications* 14.4 (2021): 1905-1916.
- [19].Arulkumar, V., and N. Bhalaji. "Performance analysis of nature inspired load balancing algorithms in cloud environments." *Journal of Ambient Intelligence and Humanized Computing* 12.3 (2021): 3735-3742.
- [20].Miao, Zhang, et al. "A discrete PSO-based static load balancing algorithm for distributed simulations in a cloud environment." *Future Generation Computer Systems* 115 (2021): 497-516.
- [21]. Kanthimathi, M., and D. Vijayakumar. "An enhanced approach of genetic and ant colony based load balancing in cloud environment." *2018 International Conference on Soft-computing and Network Security (ICSNS)*. IEEE, 2018.
- [22]. Janakiraman, S., Priya, M.D. Improved Artificial Bee Colony Using Monarchy Butterfly Optimization Algorithm for Load Balancing (IABC-MBOA-LB) in Cloud Environments. *J NetwSyst Manage* 29, 39 (2021).
- [23]. Wang, Gai-Ge, Suash Deb, and Zhihua Cui. "Monarch butterfly optimization." *Neural Computing and applications* 31.7 (2019): 1995-2014.
- [24]. He, Jing. "Cloud Computing Load Balancing Mechanism Taking into Account Load Balancing Ant Colony Optimization Algorithm." *Computational Intelligence and Neuroscience* 2022 (2022).
- [27]. Rathidevi, R., N. Nachiyar, and Jahangeer, "Secure multi-owner data sharing for dynamic groups in the cloud", *International journal of innovations in engineering research and technology*, pp.2394-3696 vol.2,no.5, 2015.
- [25]. Feng, Yanhong, et al. "Monarch butterfly optimization: a comprehensive review." *Expert Systems with Applications* 168 (2021): 114418.
- [26]. Ghetas, Mohamed. "A multi-objective Monarch Butterfly Algorithm for virtual machine placement in cloud computing." *Neural Computing and Applications* 33.17 (2021): 11011-11025.