PRE-COPY LIVE VM MIGRATION TECHNIQUES IN CLOUD COMPUTING USING HDWHM ALGORITHM

N. Nirmala Devi (1), S. Vengatesh Kumar (2)

(1) Research Scholar (PT), Department of Computer Science; (2) Department of Computer Applications (PG),
Dr. SNS Rajalakshmi College of Arts and Science, Coimbatore
India

* Corresponding Author, e-mail: nirmalamca83@gmail.com

Abstract: Live Virtual Machine (VM) migration is a legacy of server virtualization in cloud computing. One of the difficulties in a Virtual Machine (VM) Environment is load balancing, and the precopy migration strategy is both time-consuming and expensive. In this paper, we propose an efficient VM migration strategy to compute the parameters for Linear Adam Algorithm overload detection and Interquartile Range (IQR) underloaded identification, the Central Processing Unit (CPU) utilization is sufficient. VMs require constant network traffic, storage, CPU, and memory demands. This study recommends using the novel Hamming Distance-Based Weighted Harmonic Mean (HDWHM) strategy for pre-copy live VM migration. We tested the suggested strategy using datasets from CloudSim-3.0.3 on a real PlanetLab. Simulations show that the suggested approach decreases host downtimes, energy consumption, VM migrations, and SLA degradation.

Key words: Cloud Computing, LB, VM migration, Pre-copy, SLA.

1. INTRODUCTION

Computer resources are treated as activities rather than products in cloud computing. Customers may use computer resources through a link while shifting capital investment, maintenance, and operating expenses to a third-party organization, the cloud platform. Users may escape restricted or overcrowded computer resources by requesting assets. Cloud computing removes server virtualization from physical infrastructure. Virtualization enables several servers to run on one machine. Shorter deployment timeframes, dynamic LB, hardware optimization, dynamic resource management, and diverse control systems are advantages of server virtualization. Live migration optimizes cloud infrastructure. Even yet, it's a hard operation made more so by a cloud management system that runs on the hypervisor and interfaces with migration. Each step of the migration process affects how long it takes to complete. The simulation accounts for the entire migration length without separating phases or assuming they are stable or predictable. Identifying and analyzing migratory mechanisms may enhance live migration comprehension, adjustments, and modelling. Large-scale Cloud Data Centres (CDCs) use virtualization to meet growing processing, memory, and connectivity needs.

VM migration enables LB, online system management, preventive fault tolerance, power control, and resource sharing. Live migration aims to reduce VM application disruption and migration time. VM data includes memory, CPU, and storage disc. LB guarantees that all networks or terminals function equally. LB becomes increasingly critical as cloud consumers and demand grow. LB maps every internet-based job to enhance response times and resource consumption. It reduces response times, improves flexibility, and eliminates bottlenecks. Resource allocation challenges include reducing costs and energy utilization, and It achieves by LB. Mapping with the host computer involves resource allocation.

This research aims to provide a cloud computing virtual machine migration approach with reduced downtime, VM migrations, and SLA violations. This research introduces a novel HDWHM pre-copy migration approach to migrate VMs to minimise energy usage and improve performance efficiently. The following demonstrates how this research paper [1] is structured: Section 2 discussed associated works, including LB, VM selection, VM allocation, and Live VM migration, Section 3 described the implementation of the proposed live VM migration approach using the HDWHM algorithm, and Section 4 illustrated the proposed technique's simulation findings. Finally, concluded in Section 5.

2. RELATED WORK

In this phase, we concentrated mainly on the most pertinent work regarding pre-copy Live VM migration ideas, which include:

Cloud infrastructure pools resources and makes them accessible to clients on demand. LB distributes work among computer nodes. This paper [2] proposed better Round Robin and changed Honey Bee Algorithms for efficient LB to control load via VMs. This article [3] proposes a strategy to enhance fault tolerance. The Ant Colony Optimization (ACO) method selects the optimum VM to transfer the cloudlet to improve execution time and energy usage. As cloud computing platforms have become more popular, accessibility and reliability have become more important as real-time computing systems. This study [4] proposes a proactive fault detection approach using LB for cloud computing. To reduce the turnaround time, this research proposed a method to achieve evenly distributed loads across VMs. The proposed task scheduling system's results are compared to currently used heuristic-based scheduling techniques [5].

In addition to generating and improving VM allocation policies, the proposed system modifies the cloud workload [6]. There are only so many resources and VMs available in cloud computing that efficient job management is critical [7]. The approach proposed in this research supports the migration of intermodal virtual elements [8]. LB is a challenging issue in cloud computing that evenly distributes work to numerous internet VM nodes [9]. The VM becomes overloaded due to heavy resource usage, which prolongs the execution time of cloudlets [10]. The interquartile (IQR) span is a statistical dispersion measure that falls among the upper and lower QR and is represented as IQR=QU3-QU1 [11].

In a symmetric distribution, its Median Absolute Deviation (MAD) is a measure of half the IQR range [12]. This study describes a heterogeneous cloud computing framework for offering real-time (dynamic workload) in the cloud. Furthermore, this study proposes the SLA Aware Energy Optimized (SAEO) Algorithm called for executing dynamic workload applications such as big data and biomedical applications [13].

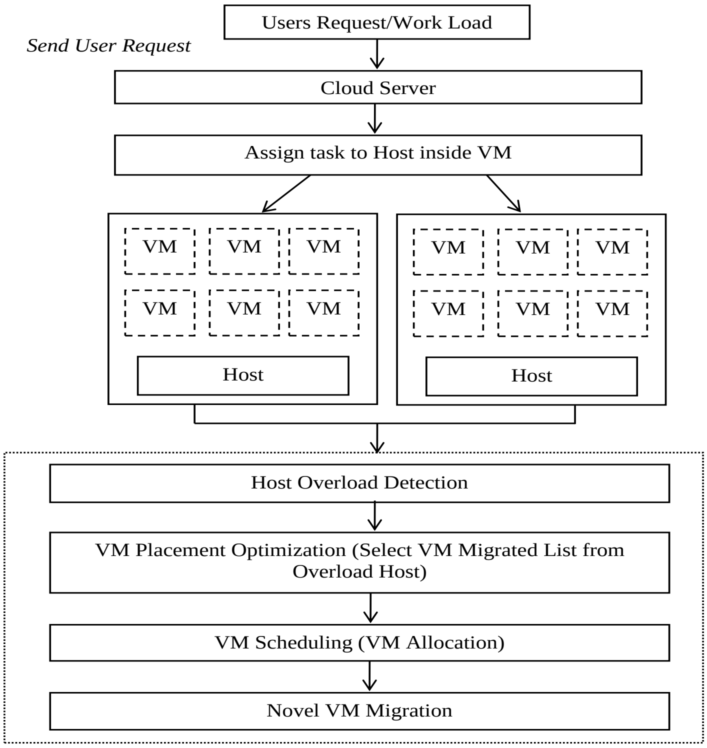

Fig. 1. Overall Flow Diagram for Proposed System

The purpose of this study [14] is to investigate a project consisting of a series of tasks that rely on one another and have varying execution times. The fact that various workers have to put in different amounts of time for their duties is a distinctive aspect of the activity. The Multi-Agent Deep Reinforcement Learning Model [15] VM Migration Algorithm is used in a system for managing energy usage by migrating VMs to other hosts. The VM scheduling problem [16] was solved by generating a hybrid Multi-Objective Whale Optimization Algorithm-Based Differential Evolution (M-WODE) technique.

3. PROPOSED METHODOLOGY

The proposed algorithm of Hamming Distance-based Weighted Harmonic Mean (HDWHM) strategy for live migration will next be explained. Figure 1 depicts a block diagram of the proposed method. It requires transferring the VM page from the source host to the destination host's VM. They propose a unique approach for VM migration in our research based on the criteria for lowering page faults, downtime, and total migration time. Researchers use HDWHM to calculate the Threshold value based on the projected value and adjustment speed history. If the current rate of page modification exceeds the threshold, the website will not be transmitted on the current iteration.

If not, it looks for changes to past behaviours on each page shown in Figure 2.

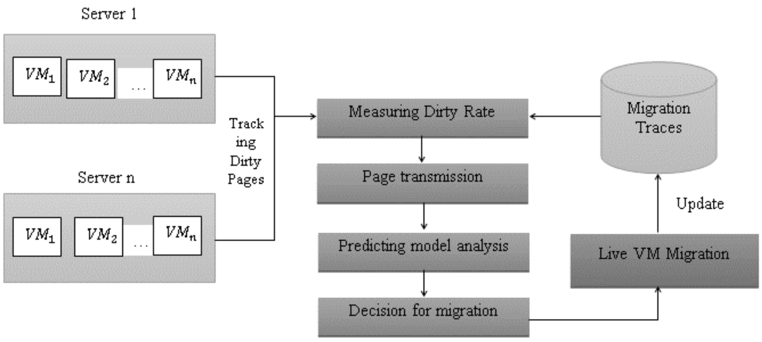

Fig. 2. Steps for Pre-copy Live VM Migration

3.1. Pre-copy techniques

It needs migrating the source host's VM webpage to the destination host's. Our research proposes a novel pre-copy VM migration method to reduce page faults, downtime, and migration time. The probability prediction-based Auto regression model predicts page dirtiness in the next iteration. Based on expected value and adjustment speed history, researchers use HDWHM to calculate threshold value. It requires numerous steps before deciding to migrate. LB and server consolidation needed VM migration from servers. Select the best VM or collection of VMs that fulfil the migration goal or selection criteria. We initially calculate each VM's memory consumption rate by analysing past and current page access patterns. The computer adjusts storage page transmission rate based on filthy rate. Performance metrics including migration time, migration cost, downtime, data transfer volume, and so on are used to predict VM efficacy. Performance indicators also determine VM migration.

3.2. Hamming Distance-Based Weighted Harmonic Mean (HDWHM)

Hamming length is essential in fault detection and overhaul. A message error may be measured by comparing the damaged and original bits. To compute a harmonic mean, the number of data items was divided by the sum of their inverses (1/xi). One of the Pythagorean means is harmonious. The harmonic average shows the Pythagorean minimum.

Harmonic mean is used to calculate the mean of ratios or rates. It's the best measurement for ratios and rates since it equalizes all data points. The geometric mean gives smaller data points less weight than the arithmetic mean, favouring big data. Eq. 1 calculates the total harmonic average.

$$\text{HM} = \text{n} / (\textstyle\sum_{i=1}^{k} 1/\text{yi}) \qquad (1)$$

Where HM- Harmonic Mean; n – a database's number of values; y_i –dataset point. The following equations could be used to determine the weighted harmonic mean given in eq 2:

$$\text{Weighted Harmonic Mean} = (\textstyle\sum x_i) / (\textstyle\sum x_i/y_i) \qquad (2)$$

Where x_i– is the weight of the data point; y_i – is the point in a dataset. A Hamming distance has been described as follows eq 3:

$$d(x,y) = \textstyle\sum_{i=1}^{n} \left|\mu_A(x_i) - \mu_B\left(x_{ji}\right)\right| \qquad (3)$$

The standardized Hamming distance between two interval-valued integers A and B using the following transfer function given in eq 4:

$$\mu_A(x_j) = \left[a_{x_j}^L, a_{x_j}^U\right], \mu_B(x_j) = \left[b_{x_j}^L, b_{x_j}^U\right] \; j = 1, 2, \ldots, n. \qquad (4)$$

The Hamming distance [17] is calculated using a weighted harmonic mean derived using an eq 5.

$$\text{Hamming distance} - \text{based weighted harmonic mean} = \frac{\sum_{i=1}^{k} |x_i - y_i|}{\sum\left(\frac{x_i - y_i}{X}\right)} \qquad (5)$$

Here, we proposed an algorithm for threshold forecasting method using HDWHM-based proposed Pre copy migration strategy to calculate mode of memory page rate. If the page is not modified, memory pages are directly sent to the virtual machine's destination.

Where HDWHM- Hamming distance-based weighted harmonic mean; MPR – Modified Page Rate; MPDR - Modified Page Dirty Rate; CMPR – Current Modified Page Rate; TH – Threshold; PDR - Page Dirty rate. The modified page rate and HDWHM are calculated based on the threshold value.

Algorithm: HDWHM-based proposed Pre copy migration strategy
Input: No_memory_page, Max_iteration, Stop_Criteria, MemoryPages[]
Output: Page_Transfer_Count
Initialization

```
Page_Transfer_Count =0, To_Send_Last =[],
Skipped_iteration=0, MPR=[], MPDR=[][]
memorypagemode=Calc_MemoryPageMode(MemoryPages)
for(  iteration=0; iteration < Max_iteration; iteration ++)
        CMPR= CalModifiedPageRate(MemoryPages)
        MPR[iteration]= CMPR
        for(i=0;i< No_memory_page;i++)
               MPDR[i][iteration]=PageState
        end for
        if(iteration = 0)
            if(memorypagemode == fraction)
                for (i=0;i< No_memory_page;i++)
                    if(MemoryPages [i]==0)
                      MemoryPages[i] migrated to Destination Vm
                       Page_Transfer_Count =Page_Transfer_Count + 1
                     end if
                end for
            else if(memorypagemode == None_modification)
                for(i=0;i< No_memory_page;i++)
                    MemoryPages[i] migrated to Destination  Vm
                     Page_Transfer_Count =Page_Transfer_Count + 1
                 end for
             end if
        else
             if(Skipped_iteration <  stop_criteria )
                  Holt_Forcast=Holt_Winter(MPR)
                  TH=HDWHM(MPR, Holt_Forcast)
               if(CMPR < TH)
                      for(i=0;i< No_memory_page;i++)
                         PDR=CalcDirtyRate(MPDR[i])
                         PageRate=  PDR / iteration
                          if(PageRate <0.5)
                                 if(MemoryPages[i] in To_Send_Last )
                                 Remove[To_Send_Last [MemoryPages[i]]
                                 endif
                                  MemoryPages[i] migrated to Destination  Vm
                               Page_Transfer_Count =Page_Transfer_Count + 1
                           else
                               if(MemoryPages[i] not in To_Send_Last )
                                  To_Send_Last[]=MemoryPages[i]
                                end if
                            end if
                        end for
```

```
                else
                    skippediteration=skippediteration+1
                end if
    else
                Stop_and_copy (To_Send_Last)
                Page_Transfer_Count =Page_Transfer_Count +Len(To_Send_Last)
            end if
        end if
end for
```

Live Migration thresholds allow the servers when to begin and stop transferring data to the subsequent virtual disk in the storage structure. The threshold values are calculated by Hamming the distance between the first page and next page. This Hamming distance value is applied to weighed harmonic equation to calculate the page dirty rate.

It assumes page rate to calculate total page rate of modification. If the page rate is less than 0.5, the memory pages are directly migrated to the destination; otherwise, the modifications are done continuously. The complete pages are then stopped and sent to the destination. Finally, an output determines the page transfer count depending on the page rate of the migration from one VM to another. It is classed as 0 or 1: if the page is not modified, it is assumed to be 0; if the page is modified, it is considered to be 1.

4. RESULTS AND DISCUSSION

4.1. Experimental Setup

This section discusses the simulation setup. CloudSim simulates cloud data centre VM migration. Over 200 hosts. 1) Host category 1 uses HP ProLiant ML110 G4, Intel Xeon3040 CPU, 4GB RAM per PM, and 2CoresX1860MHZ per PM. 500 data centres may store VM files and user data using Network Attached Storage. Servers utilize LB. We use a random dataset to evaluate the suggested VM migration and predict the efficient output.

4.2. Performance Metrics:

The following metrics can be employed to evaluate the effectiveness of pre-copy live VM migrations [18]:

Total Migration Time: Total memory transported from origin to destination and allowed bandwidth or network speed are deciding criteria (eq 6).

$$t_m = \frac{v_m}{b} \tag{6}$$

Where $t_m =$ time of total migration, $v_m =$ total memory as well as b = bandwidth

Downtime: The downtime depended on the amount of dirty pages, page size, time since previous pre-copy round.

$$t_d = \frac{d * l * t_n}{b} \tag{7}$$

Pages Transferred: Total pages communicated during VM migration and VM RAM both account for duplicate pages, which determine the page transmission rate by rounding the value from equation 8,

$$v_i = \begin{cases} v_{mem} & if\ i = 0 \\ d * t_{i-1} & otherwise \end{cases} \tag{8}$$

Where t_{i-1} is the time it takes to migrate dirty memory pages created only in previous rounds, and v_{mem} is the amount of VM storage.

The combination of Hamming distance and harmonic mean improves threshold forecasting ability. The existing VM migration strategy research has problem threshold detecting in precopy VM migration strategy, and the results justify the proposed HDWHM migration strategy has low energy consumption, down time, and SLA violation reduction machines gain knowledge the migration strategy by cooperating with the environment to minimize migration cost.

4.3. Experimental Evaluations

The proposed pre-copy live VM migration approach in terms of host termination, downtime, energy consumption, number of VM migrations, SLA violation, and degradation. We have looked at In the graphs below, our proposed method, Adam IGWO HDWHM-Precopy, is shown in red, while existing systems, Linear ACO Precopy method and Mad PABFD Precopy method, are shown in blue and green, respectively.

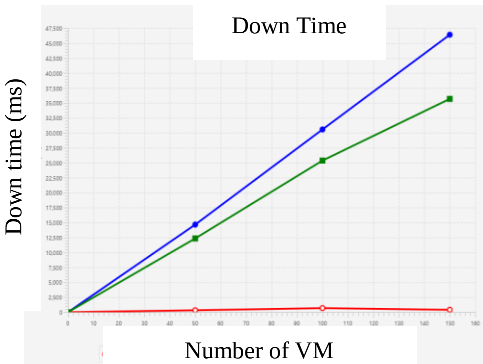

Fig. 3. Number of Host Shutdowns

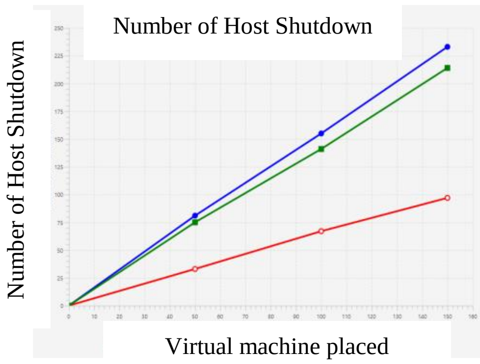

Fig. 4. DownTime

The rate of host's shutdown in proposed method was reduced considerably, as seen in Fig.3. The period of service disruption is defined as the VM downtime. Our proposed method is more effective than other existing methods as shown in Fig 4.

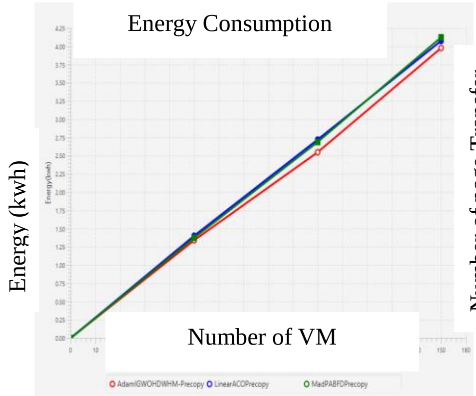

Fig. 5. Energy Consumption

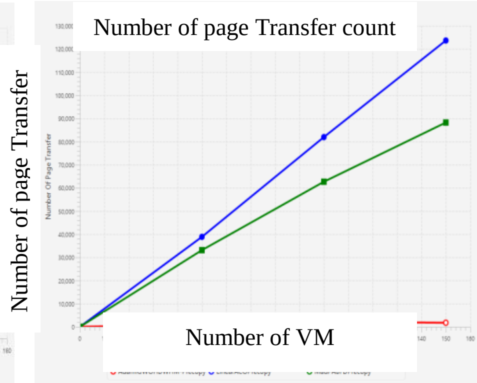

Fig. 6. Page Transfer Count

We evaluated the proposed pre-copy live VM migration technique regarding energy consumption. When the proposed and existing approaches are compared in Fig.5, it is determined that overall energy consumption is greatly decreased in our proposed system. The number of page transfer count is calculated based on number of virtual machines and page transfer count. Our proposed method using HDWHM algorithm is more efficient than existing algorithm, as shown in Fig 6.

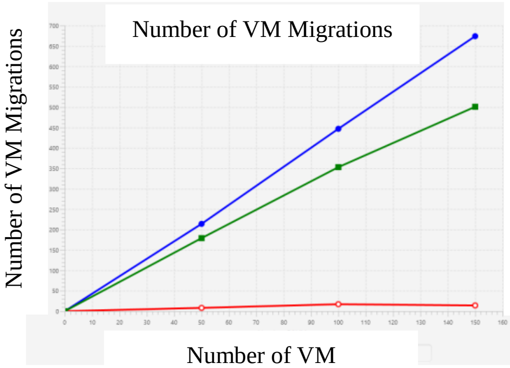

Fig. 7. Number of VM Migrations

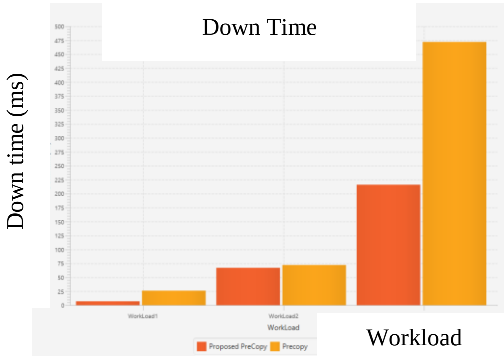

Fig. 8. Down Time comparison

Fig. 7 shows a reduction in the total number of Virtual machines for new methods. Our suggested approach, however, outperforms all others. This result shows that the proposed methodology outperforms better than existing.

The comparison of basic pre-copy and proposed pre-copy for downtime, migration time and page transfer rate is shown in Figures 8, 9, and 10, respectively. Effective VM migration is critical to reducing power usage in the data centre. SLA is an additional crucial parameter to consider since it can potentially decrease the effectiveness of the proposed approach, as shown in Figures 11 and 12. Under the performance metrics, a comparison is made with the present algorithm.

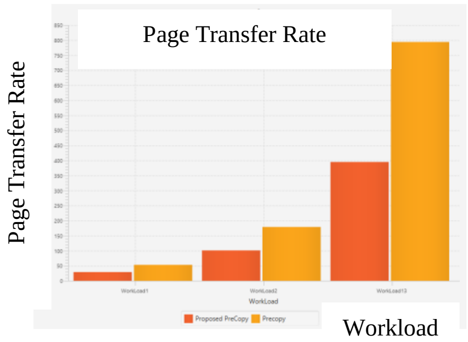

Fig. 9. Page Transfer Rate

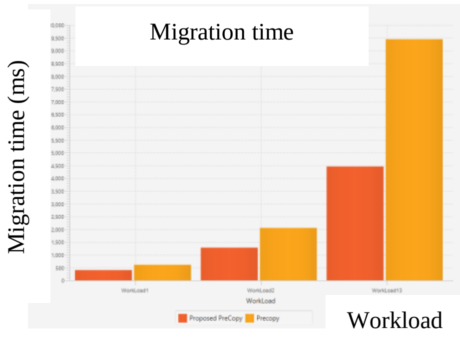

Fig. 10. Migration Time

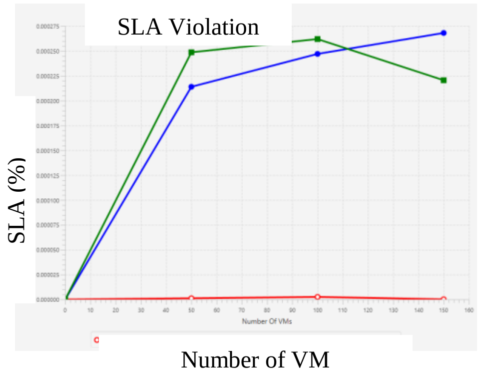

Fig. 11. SLA Violation

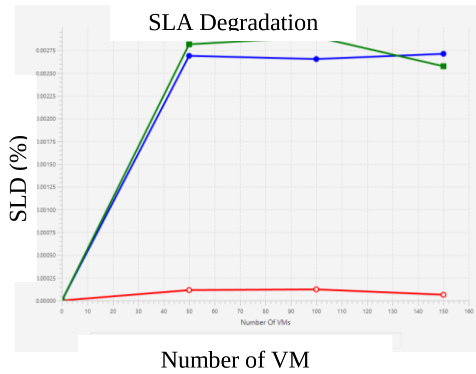

Fig. 12. SLA Degradation

5. CONCLUSION

The purpose of this research is to solve the complex process of live VM migration in the cloud using HDWHM-based Pre copy Live migration algorithm and analyze the performance with its metrics. Live migration seems to be the transfer of VM information from source to destination with as little disruption to the VM's functions as feasible. Both the post copy and the pre-copy are active migration techniques. Post-copy migration takes a lengthy time. Pre copy is preferable to post copy. This research examines the wide views of load balancing techniques and presents performance using Linear Adam and IQR methods. To evaluate the performance metrics, we ran a simulated experiment on the proposed LB methods utilizing Real PlanetLab as well as Random workloads. Our Proposed method also increases the speed of the pre-copy live VM migration by optimizing the migration mechanism by using HDWHM algorithm based on threshold values to calculate page transferring rate. In the future, we will present a revolutionary strategy that will decrease downtime as well as overall migration time.

REFERENCES

[1] Misra, S. A step by step guide for choosing project topics and writing research papers in ICT related disciplines. *Communications in Computer and Information Science*, 2021, pp.727–744. DOI:10.1007/978-3-030-69143-1_55

[2] Jeyalaksshmi, S., Smiles, J.A., Akila, D., Mukherjee, D., Obaid, A.J. Energy-efficient load balancing technique to optimize average response time and data center processing time in cloud computing environment. *Journal of Physics: Conference Series*, vol. 1963, no. 1, 2021, p. 012145. IOP Publishing.

[3] Alyouzbaki, Y.A.G., Al-Rawi, M.F. Novel load balancing approach based on ant colony optimization technique in cloud computing. *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 4, 2021, pp. 2320-2326.

[4] Attallah, S.M., Fayek, M.B., Nassar, S.M., Hemayed, E.E. Proactive load balancing fault tolerance algorithm in cloud computing. *Concurrency and Computation: Practice and Experience*, vol. 33, no. 10, 2021, e6172.

[5] Moghaddam, S.M., O'Sullivan, M., Unsworth, C.P., Piraghaj, S.F., Walker, C. Metrics for improving the management of coud environments - Load balancing using measures of quality of service, service level agreement violations and energy consumption. *Future Generation Computer Systems*, vol. 123, 2021, pp. 142-155.

[6] Ghafir, M.S., Alam, A., Siddiqui, F., Naaz, S., Virtual machine allocation policy for load balancing. *Journal of Physics: Conference Series*, 2021, vol. 2070, no. 1, 2021, p. 012129. IOP Publishing.

[7] Shafiq, D.A., Jhanjhi, N.Z., Abdullah, A., Alzain, M.A. A load balancing algorithm for the data centres to optimize cloud computing applications. *IEEE Access*, vol. 9, 2021, pp.41731-41744. DOI: 10.1109/ACCESS.2021.3065308

[8] Ramanathan, S., Kondepu, K., Razo, M., Tacca, M., Valcarenghi, L., Fumagalli, A. Live migration of virtual machine and container based mobile core network components: A comprehensive study. *IEEE Access*, vol. 9, 2021, pp.105082-105100. DOI: 10.1109/ACCESS.2021.3099370.

[9] Tuli, K., Kaur, A. Load balancing scheme for optimization of virtual machine migration using swarm in cloud environment. *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*, pp. 1-6,2021, IEEE. DOI: 10.1109/ICRITO51393.2021.9596463.

[10] Kaur, G., Sachdeva,R. Virtual machine migration approach in cloud computing using genetic algorithm. *Advances in Information Communication Technology and Computing*, pp. 195-204, 2021, Springer, Singapore.

[11] Devi N., Kumar S.V. Novel load balancing approach towards cloud computing by reducing SLA degradation and energy consumption, *Indian Journal of Computer Science and Engineering*, vol. 13, no. 3, 2022, pp.197–207.

[12] Devi N., Kumar S.V. SLAV mitigation and energy-efficient VM allocation technique using improvised grey wolf optimization algorithm for cloud computing. *2022 8th*

International Conference on Advanced Computing and Communication Systems (ICACCS), vol. 1, pp. 155-160, 2022. DOI: 10.1109/ICACCS54159.2022.9785078.

[13] Jambigi, M.V., Kumar, M.V., Ashoka, D. V. Service level agreement aware energy optimized scheduling algorithm for cloud computing environment. *International Journal on Information Technologies & Security*, vol. 14, no. 1, 2022, pp.17-28.

[14] Oleinikova, S. A., Kravets, O.Ja., Aksenov, I.A., Frantsisko, O.Yu., Rahman, P.A., Atlasov, I.V. The general scheme of the genetic algorithm for solving the task scheduling problem for a multistage system and assigning time for jobs. *International Journal on Information Technologies and Security*, vol.13, no. 4, 2021, pp. 47-58.

[15] Dai Y, Zhang Q, Yang L. Virtual machine migration strategy based on multi-agent deep reinforcement learning. *Applied Sciences*. Vol. 11, no. 17, 2021, p.7993. https://doi.org/10.3390/app11177993

[16] Rana, N., Abd Latiff, M. S., Abdulhamid, S. M., Misra, S. A hybrid whale optimization algorithm with differential evolution optimization for multi-objective virtual machine scheduling in cloud computing. *Engineering Optimization*, vol.54, no. 12, 2021, pp.1999–2016. DOI:10.1080/0305215x.2021.1969560

[17] Wu, L., Xia, H. Particle Swarm optimization algorithm for container deployment. *Journal of Physics: Conference Series*, vol. 1544, no. 1, 2020, IOP Publishing.

[18] Choudhary, A., Govil, M.C., Singh, G., et al. A critical survey of live virtual machine migration techniques. *Journal of Cloud Computing*, *vol.* 6, no 1, 2017, p/1-41. DOI: 10.1186/s13677-017-0092-1

Information about the authors:

Nirmala Devi N is presently working as Assistant Professor, Department of Computer Science in Auxilium College (Autonomous), Vellore, India. She is pursuing Ph.D. Computer Science in Dr.SNS Rajalakshmi College of Arts & Science (Autonomous), Bharathiar University, Coimbatore, India from July 2019 to till date. Her dissertation work is carried out in Virtual Machine Migration in the field of Cloud Computing.

Dr. Vengatesh Kumar S is presently working as Associate Professor, Department of Computer Applications in Dr. SNS Rajalakshmi College of Arts & Science (Autonomous), Coimbatore, India, and his research area is Cloud Computing, Image Processing, and Data Mining applications. He is reviewer for Springer journals and member in International Association of Engineers (IAENG) & ACM.

Manuscript (first version) received on 20 October 2022
Revised manuscript re-submitted on 03 November 2023