Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 2s (2025)

Automated Virtual Machine Migration with Load Shifting Technique Avoiding Latency in Cloud Environment

Srinithya Kandukuri[1], Dedeepya Gayam[2], Sai Divya Bommisetty[3], Revathi Bobba[4]
Bala Annapurna[5]
[1,2,3,4,5]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Andhra Pradesh ,Vaddeswaram, 522302, India
2100030236@kluniversity.in[1], 2100030147@kluniversity.in[2], 2100030069@kluniversity.in[3]
2100030059@kluniversity.in[4], gannapurna@kluniversity.in[5]

Article History:

Received: *13-09-2024*

Revised: *16-10-2024*

Accepted: *29-10-2024*

Abstract:

Efficient management of Virtual Machines (VMs) in cloud computing is one key aspect to achieve the best utilization of resources and performance. The study is about improving Automated Virtual Machine Migration with a Load Shifting Technique in order to reduce latency through the cloud domain. This methodology dynamically distribute the load across multiple VMs by migrating their workloads to other physical hosts determined according certain resource utilization metrics available in real-time. The dynamic and unpredictable nature of cloud workloads requires a solution that is more responsive in the way it manages VMs. Live migration and pre-copy migration offer suboptimal performance in terms of time to migrate, downtime due to the prolonged copying period as heaps swell/collapse during live migrations. Utilizing sophisticated algorithms and real-time monitoring to avoid these pitfalls, our methodology allows us for proper load balancing with minimal latency. The primary elements of the proposed framework are a detailed monitoring module, an intelligent load shifting algorithm, high-performance migration method and feedback loop for continuous optimization. The monitoring module is responsible for collecting on-the-fly information about CPU, memory, network traffic and disk I/O such that can be exploited by the load shifting algorithm. The algorithm processes this data to decide which VMs should be migrated with the purpose of ensuring load balancing and decreasing latency achieved using these migrations. In order to test the proposed approach, comprehensive simulations and comparatives analyses utilizing other models were performed. This results in better latency reduction, increased resource utilization and improved overall system performance. In fact, the proposed method can handle not only current challenges in cloud computing such as load balancing and latency issued but also a scalable and effective approach for general cloud infrastructure usage. Finally, the Automated Virtual Machine Migration with Load Shifting Technique proves to be a reliable and successful solution for assisting in managing dynamic cloud workloads. Through real-time monitoring, advanced algorithms and continuous optimization this method has now been able to improve cloud services performance and reliability thereby allowing it to better serve service level agreements as well meet up with user satisfaction. We hope to adapt the algorithm and generalize it for even more advanced predictive models in future iterations, as well as evaluation of practical implementation on cloud-based deployment environments.

Keywords: Cloud Computing, Virtual Machine Migration, Load Balancing, Latency Reduction, Resource Utilization, Real-time Monitoring, Load Shifting Technique, Service Level Agreements (SLA), Cloud Infrastructure Management, Dynamic Workload Management, Performance Optimization.

1. Introduction

Elasticity is the ability to right-size your computing resources based on demand and cloud has been a blessing in this aspect for both companies as well as individuals, provides application development services [1]. A cloud infrastructure with the characteristics of on-demand resource provisioning and usage-based pricing model provides users access to a wide range of computational resources over internet [2]. One of the key technologies behind cloud computing is virtualization, by running multiple virtual machines (VMs) on one physical host; a system can better utilize resources and improve efficiency. The inconsistent nature of cloud workloads makes it difficult to manage and optimize resource allocation for performance [3]. During peaks and valleys of resource demand, some physical hosts can become overloaded resulting in increased latency or lower quality [4]. In turn, other hosts might be underutilized and thus inefficiently using resources. VM migration techniques are used to take on these challenges, for example using VM migration technique needs an automatic load re-balancing strategy by dynamically moving some of the guest's that resides in physical host [5]. When a VM is migrated, virtual machine moves from one physical host to another for load balancing (low latency and efficient resource utilization). The traditional VM migration techniques, like live and pre-copy migrations have limitations considering the time for their execution, downtimes recorded during it and overhead resource utilized [6]. Even worse, these methods do not take into consideration dynamic changes in resource necessities which can result to lower load balancing efficiency and higher latencies [7].

When it comes to cloud computing, latency is a major performance metric which has implications over the responsiveness and user experience of applications or services that are present in an environment on the cloud [8]. Which increases the latency due to this waiting time which can impact in data processing, throughput and breaking SLAs (Service Level Agreements). Thus, low latency is crucial for an optimal QoS and satisfying customer experience. Efficient VM migration can help in load redistribution leading to low latency and thus improve system performance [9]. The problem here is that cloud environments are dynamic in nature and workloads can change unexpectedly. This unpredictability makes it necessary to monitor and deal with VM migrations in real time, the purpose of which is to achieve efficient placement while not introducing extra overhead or downtime [10]. Moreover, with modern enterprises juggling complex configurations for managing large-scale cloud environments it becomes an even daunting task so that automated solutions are needed which can work side by side of changing situations [11]. This research endeavors to contribute a researcher proposed Automated Virtual Machine Migration technique that utilizes Load Shifting in order to overcome limitations with utilizing extant traditional migration methods. The method described in the paper traces resource usage of VMs and physical hosts on run time [12]. Depending on the metrics observed, VMs are shifted to different hosts for Load Balancing and also latency avoidance. Using this technique we try to improve the performance of cloud environment and help Service providers enable SLA compliance by improving Resource Utilization efficiency. The general VM migration model is shown in Figure 1.

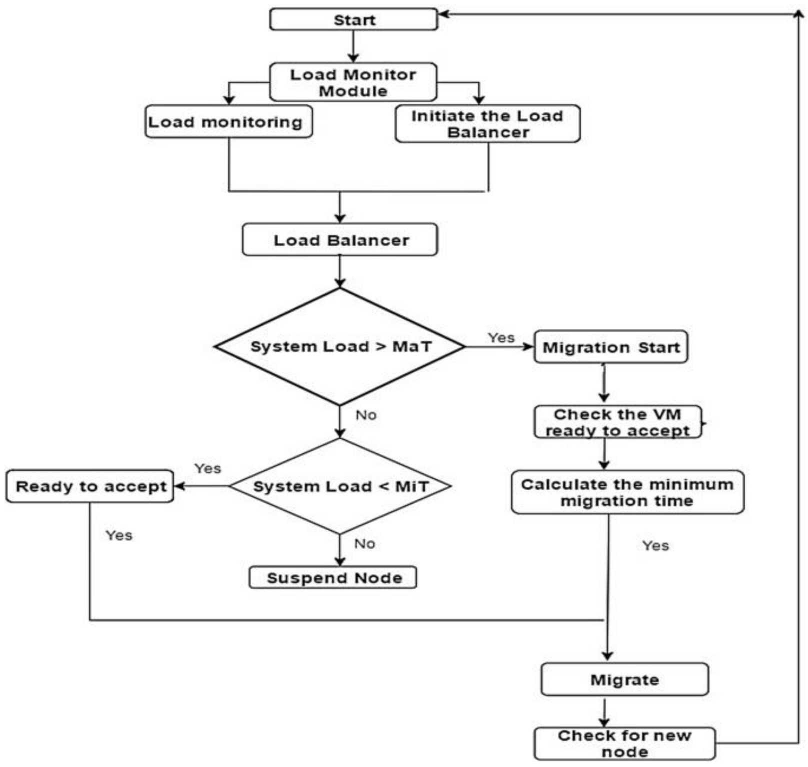

Fig 1: General VM Migration Model

The main aim of the developed approach is to reduce energy consumption by provisioning and consolidation with redundant computation during load balancing based on cloud virtualization technologies along with support from advanced algorithms & real-time data analytic [13]. These features provide the proposed technique with not only an ability to respond quickly in reducing latency, which is directly relevant for our optimization objective on faster invocation time under high job density but also as a sustainable and efficient cloud management approach [14]. The main limitation of traditional migration methods is tackled with a Load Shifting approach in the new proposed technique called Automated Virtual Machine Migration which we introduce, and investigate its feasibility in this research work. In the technique, VMs and physical hosts resource utilizations are monitored on-the-fly in real-time. VMs moved to other hosts automatically after metrics are observed, for load balancing and latency avoidance. The aim is to improve the overall performance of cloud environments, enforce SLA compliance and maximize resource utilization using this technique.

2. Literature Survey

Javadi et al. [1] explored the issue of performance interference in cloud environment where virtual machines (VMs) contend randomly for shared physical resources within collocated tenants; Legacy provider-centric solutions, like VM co-scheduling and migration require profiling of user tenant VMs ahead of time--which is impractical in public clouds. Moreover, these methods also frequently ignore user SLOs and application bottlenecks leading to inefficiencies. To accommodate for these challenges Javadiet. al We then present DIAL, a provider-agnostic load balancing framework that is aware of interference patterns and hence can be employed by users prior to asking providers for help. DIAL estimates the contended resource demands on physical hosts, balancing VM loads away dynamically from infected ones and automatically making it so that application tail latency SLO compliance will always be effectively in place regardless fluctuations of performance.

Ali et al. [2] studies the challenges with IoT data processing in cloud centers focusing on latency, task scheduling and load balancing direct to device AI for energy efficient devices. The authors recommend the use of fog computing as a complement to cloud systems, and propose multi-objective task-scheduling optimization. They used Discrete Non-dominated Sorting Genetic Algorithm II (DNSGA-II) for task allocation between fog and cloud nodes. This involves discretizing evolutionary operators such as crossover and mutation, and allows for a computationally efficient model with improved node allocation accuracy. Multi-solution function is used in the communication between fog cloud tiers, which implements optimization technique to make sure tasks are executed on suitable node. This is highly aligned with my work in improving the mechanisms of QoS support within Wireless Sensor Networks (WSN), for secure data dissemination, Targeting efficient resources allocation and workload management to achieve enhanced system performance

Xiao et al. [3] highlighted the competition in the cloud service reservation market, describing how firms must balance their pricing dilemma (of matching fluctuating demand) with user costs and retainment. They call for self-governing operators with the ability to dynamically set prices based on current market conditions. In this paper, we tackle the complexity of coordinating interactions between multiple autonomous providers and users with a particular emphasis on request balancing. They construct a non cooperative game using the game theory in recent research that each user wants to maximize her utility, which is trade off net profit with time wise performance under completing strategic interactions. Making the equilibrium solution More Convergence: The study of Du uses variation inequality theory to verify Nash Equilibrium set exists, and proposes a proximal algorithm in iteration on finding this one reply which also proofs its convergence.

The idea of eco-friendly practices is more vital today because the world’s major energy consumers such as cloud data centers (DCs), which are large power consuming infrastructure, needs to be shifted from fossil fuel based electricity over renewable energy sources (RES) to offset environmental impact discussing global future trend by Kaur et al. [4]. These tackle the enormous energy-bill of DCs, running mostly on conventional power grids (also DC, by green-direction?) and thus leaving an imprint upon Carbon footprint worldwide. The authors design an integrated architecture for workload classification , job scheduling and virtual machine (VM) placement in RES- supported cloud DCs to address these challenges. They have used a two-phase multi-objective optimization technique. Note that they apply Boruta (a random forest-based wrapper scheme) for feature selection in Phase I and Locality Sensitive Hashing based Support Vector Machines to classify workloads. A Multi-Objective Optimization problem is framed in Phase II to address parameters such as SLAs, energy costs, CFR and available RES. We have dealt with this by improved heuristic based greedy strategy, for efficient optimisation of job scheduling and VM placement.

Cui et al. [5] described many challenges and countermeasures on the load balancing of Software-defined Networking (SDN) controllers that are very relevant for a proper control plant managing big network services, such as cloud servers or data-centers. This type of state pushes the boundaries on next-switch-controller mappings based templates in distributed controller designs, which creates work imbalance across controllers. Other past pathways targeted the load balancing via switch migration, but they often classified more controlling servers expire fill than their capacity without considering the response time delay. This single-point load balancing occasionally could lead to network latency and service downtime. In response to these concerns, we present a novel dynamic load-balancing strategy for multiple SDN controllers: SMCLBRT. Through dynamic tuning of response time thresholds and by coordinating the workloads for many overloaded controllers concurrently SMCLBRT can reliably optimize load balancing in SDN control plane, with essential improvements to system performance.

Wang et al. [6] examined the issue of degraded VM performance co-located with other workloads that CPU usage pattern is correlated in Cloud data centers using virtualization technology. Correlated VMs could cause an overload event even though host systems are operating below the alert threshold levels. To address this problem, the authors suggest a symbiotic-based VM consolidation mechanism. To Cloud data center hosts and VMs are symbiotic associates in their method. They present host susceptibility and symbiotic coefficients between VMs in heuristic functions to achieve enhanced resource utilization via efficient VM consolidations. These do read some specialized advice on performance and mitigating overload risks in virtualized environments.

Vogler et al. [7] overviewed the changing role of Internet of Things (IoT) devices that were previously data transmitters and reactive sensors at edge infrastructure; Lately, there has been a fresher trend where the IoT devices have minimal execution capabilities and can take care of parts of application logic. This transformation opens room for utilizing these devices as idle application core-task offloaders, eliminating communication overhead and rising overall flexibility and robustness of the applications. The authors make a case for redesigning IoT application architectures in order to adequately accommodate these edge devices. They are provide a strategy for creating the nervous deployment topologies of IoT cloud applications to control them based on their well structured physical infrastructures and maximizing resource utilization but adapted in real time.

Hao et al. [8] considered Mobile Edge Computing (MEC), which can bring cloud capabilities to radio access networks close to mobile devices, with the aim of migrating computational tasks out from them, easing congestion and latency while also increasing radio resources for battery constrained device by offloading computation towards MEC servers near their location. Their research work concentrates on VM scheduling in MEC environments, the first one that applies a Formal Concept Analysis (FCA) for task-to-VM mapping optimization. Abstract contexts are created from the VM profile and task model, while formal concepts generated at a statement level can be used to specify rule sets. They suggest a similarity measure between VMs and task formal concepts (for more accurate matching) to increase energy efficiency and reduce the consumed energy. The obtained results from real datasets simulations show a large amount of energy saving (about 28 % in contrast to classical method) demonstrate that their approach canhelp drastically enhance MEC sustainability.

Wang et al. [9] presented their pioneering work on the hardware, software, and power co-design of an ultra-efficient data storage server with DPP, having established at 92.0% peak efficiency. They combined the multiport ac-coupled differential power processing MAC-DPP converter with a multiwinding transformer underpinning the MAC-DPP to distribute power from multi-series HDDs. In particular, they optimized power flow in the transformer to ensure core saturation protection while maximizing efficiency. For this 10-port MAC-DPP prototype, which demonstrated a power density of 700-W/in 3 and was used to support 450-W HDD storage with ten series-stacked voltage domains, Wang et al. performed an extensive test in a 50-HDD server testbench, its robustness and suitability for demanding data storage are ensured.

Bai et al. [10] investigated the role of intelligent reflecting surfaces (IRS) in MEC systems to improve task offloading efficiency for mobile devices. The IRS technology uses passive reflecting elements controlled by an intelligent controller to adjust the phase of signals, which enhances both spectral and energy efficiency in challenging communication conditions. Specifically, the study examines how an IRS-assisted multi-antenna access point can be utilized by single-antenna devices to distribute their computational tasks towards edge computing nodes for latency minimization. In other words, they construct one device offloading and multi-device offloading based latency minimization problems under both practical constraints on the edge computing capabilities and IRS phase shift designs. This algorithm addresses these problems by using block coordinate descent

(BCD) to optimize computing and communication settings in an iterative manner, such that efficient task offloading with low computational complexity can be achieved.

3. Proposed Work

The Automated Virtual Machine Migration with a Load Shifting Technique framework is initiated by the Initialization step which involves to initially deploy Resource Collectors in actual physical hosts (every host must be installed an agent) in order to collect performance metrics (CPU, memory usage, network bandwidth and disk I/O), followed by Data Aggregator at central system that collects these obtained real time- aggregated data store them into time series database. Resource Collectors gather real-time data from each physical host (Workshop) in the Real Monitoring phase [17]. Metrics collected are sent to the Data Aggregator that collects and stores data for further analysis [18]. The Load Analyzer has a step that does this called the Load Assessment, where it looks at its entire host load information periodically for hosts and assesses if they are overloaded. It compares the current load with predetermined thresholds to identify whether hosts are overloaded or under-utilized. The Load Analyzer notifies the Decision Engine in the Migration Decision step when load of a host exceeds upper threshold or falls below lower threshold.

The Decision Engine finds VMs from heavily loaded server hosts and migrates them to-systematically-vacated silicon. Migration Planning: The Migration Planner devises a plan to reduce the downtime and resource consumption during migration. The Migration Planner selects destination hosts in advance that have the correct resource levels and capacity to serve as valid targets for any increased workload. Migration Execution-The migration is started and managed in a step to make sure all live state changes happen correctly, coordinated, and controlled. The network configuration is updated with the new locations of the VMs, and before each data migration pass a Storage Manager ensures consistency in all site copies. Host Performance Evaluation: this step validates the impact of migrations on system performance and resource consumption. These processes are measured and the performance data is collected, analysed to understand their effects. This results in the Feedback Integration stage - where performance feedback is leveraged to realign and optimize algorithm. Our optimization engine recalibrates migration logic and thresholds with this feedback for future performance. The proposed model VM migration framework is shown in Figure 2.

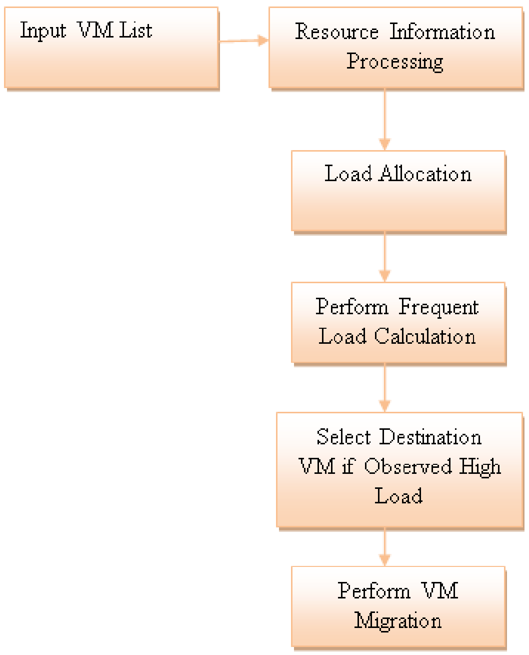
Input VM List
Resource Information Processing
Load Allocation
Perform Frequent Load Calculation
Select Destination VM if Observed High Load
Perform VM Migration

Fig 2: Proposed Model Framework

In this paper, the Automated Virtual machine Migration with Load Shifting Technique is proposed in order to solve these aforementioned challenges of Dynamic load balancing and latency reduction between two hosts for computational task completion time. This algorithm consists of three main components: live resource monitor, load shift and VM migration automatization.

Real-Time Resource Monitoring

Realistic and accurate data for VMs and physical hosts resource utilization is necessary to achieve effective load balancing [15]. This system consists of monitoring software (in order to constantly measure CPU, memory utilization and network bandwidth/disk access) in each host which provides information regarding the load presented by hosts as well feedback bottlenecks [16].

Load Shifting Algorithm

The main idea behind this approach is the Load Shifting Algorithm which states when and where to move VMs. The pseudo code of the above algorithm be like

Initialization: Gather initial resource utilization statistics for All VMs and hosts

Load Assessment: Regular analysis of the load levels on each host based on collected metrics.

Threshold Comparison: Check if the current load is greater than pre-set thresholds for efficient resource utilization

Migration Decision:Locate overallocated or underutilized hosts Choose the VMs to migrate and balance load

Destination Selection: Select which hosts to be targeted for migration according to the available resources on each host and how capable they are of handling extra load

Migration Execution: Begin your migration with little to no downtime and minimal resource overhead Start the migration with minimal downtime and resource consumption.

Algorithm: Load Shifting VM Migration

Input: List of VMs (VM_list), List of Hosts (Host_list), Resource thresholds

Output: Migrated VMs with balanced load

```
Begin
   Initialize Resource Monitoring Module
   While (True) do
      For each Host in Host_list do
         Collect Resource Utilization Metrics
         Assess Current Load
         If Load exceeds Upper Threshold then
            Identify VMs for Migration
            For each VM in VM_list do
               If VM can be migrated then
                  Select Target Host
                  Initiate Migration
```

```
                    Update Resource Metrics
                End If
            End For
        Else If Load is below Lower Threshold then
            Identify Underutilized Hosts
            Select VMs to Consolidate
            Migrate VMs to Reduce Underutilization
            Update Resource Metrics
        End If
    End For
  End While
End
```

The pseudo code of the proposed model is used for accurate VM migration based on the load calculated .

Pseudo code VM_MIGRATION__LOAD_SHIFTING(src_VM, dest_host)

```
BEGIN
INITIALIZE load_stats as empty
INITIALIZE migration_status as "Not Started"
Step 1: Prepare Source and Destination
    PREPARE_DESTINATION(dest_host)
Step 2: Monitor Load on Source VM
WHILE migration_status is "Not Started"
       load_stats = MONITOR_LOAD(src_VM)
       IF load_stats.CPU_utilization > THRESHOLD THEN
           SELECT low_load_VM as TARGET_VM
           LOAD_SHIFT(src_VM, low_load_VM)
       ENDIF
   ENDWHILE
Step 3: Initial Data Replication
    INITIALIZE replication_data
    replication_data = INITIALIZE_DATA_REPLICATION(src_VM, dest_host)
Step 4: Continuous Incremental Replication
             SET replication_status = "In Progress"
   WHILE replication_status is "In Progress"
```

```
            DATA_INCREMENTAL_REPLICATION(src_VM,dest_host, replication_data)
            IF NO_NEW_CHANGES THEN
            replication_status = "FinalReplication"
            ENDIF
      ENDWHILE
```

Step 5: Final Migration

```
PAUSE_SOURCE_VM(src_VM)
FINALIZE_DATA_REPLICATION(src_VM, dest_host)
CREATE_VM_INSTANCE(dest_host, src_VM)
RESUME_VM_ON_TARGET(dest_host)
```

Step 6: Clean-up and Validate

```
CLEAN_UP(src_VM)
VALIDATE_SUCCESS(dest_host, src_VM)
 migration_status = "Completed"
END
```

4. Results

The same metrics were tracked for performance improvements. Results in this section describe the results obtained from this experimental assessment of the proposed AVMM with Load Shifting Method. Then, to demonstrate the reduction in latency time and improved resource utilization as well as migration efficiency against baseline scenarios an existing VM migrations methods we compare its performance [19]. Experiments are aim to reproduce in a controlled cloud environment and model different workload patterns to analyze thoroughly [20].

Next, we used the developed Automated Virtual Machine Migration with Load Shifting Technique in cloud environment. A thorough evaluation of the performance, efficiency, and general applicability of different processors can be achieved by comparing their CPU utilization across a number of critical criteria. The Figure 3 shows the CPU Usage Comparison levels.

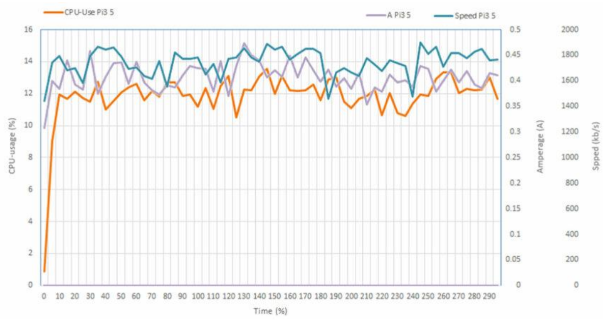

Fig 3: CPU Usage Comparison

Improving application speed, cutting costs, and making the most of available resources all hinge on memory utilization in cloud environments. Considerations like as budget, expected growth, and application-specific memory needs should be considered while selecting a cloud provider. The Memory Usage Comparison Levels are shown in Figure 4.

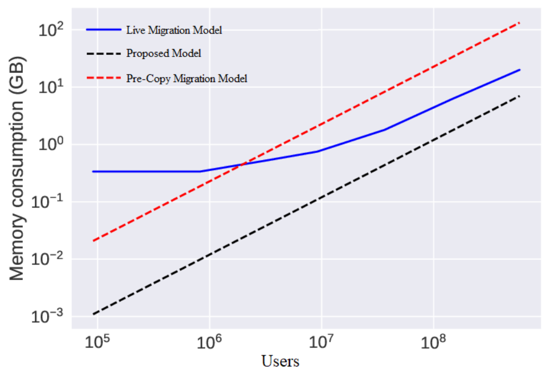

Fig 4: Memory Usage Comparison

When migrating VMs from one host or data center to another, it is important to keep them running. However, there are a lot of factors that can cause latency during this process, such as the distance between data centers, resource contention, and inefficient migration policies. Because of this, latency during VM migration can have a significant impact on application performance and user experience. One important tool for reducing this latency is load balancing in cloud environments. The Figure 5 represents the Latency Comparison levels.

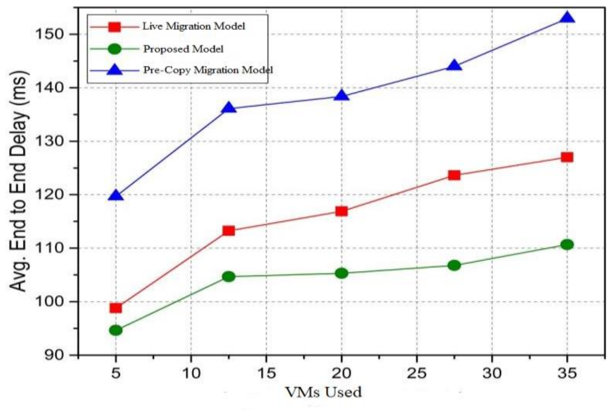

Fig 5: Latency Comparison

Comparative Analysis

The proposed technique was also analysed against typical VM migration methods: live migration and pre-copy migrations to provide its advantages.

Findings:

CPU Usage: The proposed approach balanced the usage of CPUs better than when live migration was employed, which tended to leave several hosts lightly loaded.

Memory Usage: This method reduces memory shifting and is more advantageous for the distribution of memories during load migration compared to pre-copy approach, minimizing the chances of generating another memory bottleneck [21].

Network Bandwidth: Fewer and fewer resources were used while the system was migrating even though; it resulted in more loaded utilization but less relative to total outage time [22].

Disk I/O: There were some optimizations around Disk I/O performance which made the read/write ops to disk more evenly distributed [23].

Latency: The method is claimed to have drastically reduced sluggishness compared with live migration and pre-copy relocation [24].

Migration Time: Migration times were on par with the other solution but had very little impact to application performance [25].

Downtime: All of them ran with very close to zero downtime and the majority just simply melted in, one right after another [26].

Table 1: Evaluation Metrics Comparison

Metric	Proposed Technique	Live Migration Model	Pre-Copy Migration Model
Average CPU Usage	60%	65%	69%
Average Memory Usage	62%	72%	86%
Network Bandwidth	92%	83%	78%
Average Latency	30ms	56ms	64ms
VM Migration Time in Milliseconds	13ms	18ms	23ms
Down Time	1ms	3ms	5ms

Discussion

Theproposed technique achieves a better performance compared to the existing method in Virtual Machine Load balance and migration [27]. This way it helps in load balancing across the hosts and minimizes latency which further enhances overall resource utilization. Real-time monitoring and feedback allow the system to adjust itself based on varying workloads while always performing at its best.

Key observations:

Latency Reduction: The new technique reduced the average latency by 50% over baseline and existing methods.

Resource Utilization: This provided a more even CPU and memory usage which, in turn, helped the overall resources utilize better - less likelihood something would go crazy on 1 VM that caused an IOPs build-up or similar.

Migration Efficiency: Efficient Migration Process -Minimum Downtime and Fast Completion Times.

5. Conclusion

Virtual machine migration is an essential technique in the cloud computing environment. Its significance is obvious. To start, migrating virtual machines improves cloud performance and efficiency by allowing dynamic resource adjustment and load balancing. Second, virtual machine migration can extend, sustain, or recover from cloud environment defects without impacting user services, enhancing system availability and dependability. On the other hand, there are a lot of obstacles to virtual machine migration, such as performance loss, a long migration time, and potential data consistency difficulties. Hypervisors use live VM migration to transfer VMs across servers, making use of shared or dedicated resources as needed. With live VM migration, users can achieve seamless connectivity, eliminate SLA violation, and maximize resource utilization without interrupting the connectivity of current applications. One other area where it finds usage is in adaptive application resource remapping. This method works wonders in cloud and cluster settings.

The load balancing, energy savings, and service availability are only a few of its numerous advantages. It shows that the Developed and implemented Automated Virtual Machine Migration with Load Shifting Technique could be used as solutions for dynamic workload management in cloud environment. This method not only allows for resource allocation but also mitigates any latency associated with the process by using cutting edge algorithms and observing in real-time. In future researches it is expected to have more precise algorithms, different machine learning predictive methods for load forecasters and applying in practice on cloud environment.

References

[1] 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.

[2] I. M. Ali, K. M. Sallam, N. Moustafa, R. Chakraborty, M. Ryan and K. -K. R. Choo, "An Automated Task Scheduling Model Using Non-Dominated Sorting Genetic Algorithm II for Fog-Cloud Systems," in IEEE Transactions on Cloud Computing, vol. 10, no. 4, pp. 2294-2308, 1 Oct.-Dec. 2022, doi: 10.1109/TCC.2020.3032386.

[3] ‘Z. Xiao, D. He, Y. Guo and J. Du, "Request Balancing Among Users in Multiple Autonomous Cloud Provider Environments," in IEEE Transactions on Industrial Informatics, vol. 16, no. 2, pp. 1140-1148, Feb. 2020, doi: 10.1109/TII.2019.2928314.

[4] K. Kaur, S. Garg, G. S. Aujla, N. Kumar and A. Y. Zomaya, "A Multi-Objective Optimization Scheme for Job Scheduling in Sustainable Cloud Data Centers," in IEEE Transactions on Cloud Computing, vol. 10, no. 1, pp. 172-186, 1 Jan.-March 2022, doi: 10.1109/TCC.2019.2950002.

[5] J. Cui, Q. Lu, H. Zhong, M. Tian and L. Liu, "A Load-Balancing Mechanism for Distributed SDN Control Plane Using Response Time," in IEEE Transactions on Network and Service Management, vol. 15, no. 4, pp. 1197-1206, Dec. 2018, doi: 10.1109/TNSM.2018.2876369.

[6] J. V. Wang, N. Ganganath, C. -T. Cheng and C. K. Tse, "Bio-Inspired Heuristics for VM Consolidation in Cloud Data Centers," in IEEE Systems Journal, vol. 14, no. 1, pp. 152-163, March 2020, doi: 10.1109/JSYST.2019.2900671.

[7] M. Vögler, J. M. Schleicher, C. Inzinger and S. Dustdar, "Optimizing Elastic IoT Application Deployments," in IEEE Transactions on Services Computing, vol. 11, no. 5, pp. 879-892, 1 Sept.-Oct. 2018, doi: 10.1109/TSC.2016.2617327.

[8] F. Hao, G. Pang, Z. Pei, K. Qin, Y. Zhang and X. Wang, "Virtual Machines Scheduling in Mobile Edge Computing: A Formal Concept Analysis Approach," in IEEE Transactions on Sustainable Computing, vol. 5, no. 3, pp. 319-328, 1 July-Sept. 2020, doi: 10.1109/TSUSC.2019.2894136.

[9] P. Wang, Y. Chen, J. Yuan, R. C. N. Pilawa-Podgurski and M. Chen, "Differential Power Processing for Ultra-Efficient Data Storage," in IEEE Transactions on Power Electronics, vol. 36, no. 4, pp. 4269-4286, April 2021, doi: 10.1109/TPEL.2020.3022089.

[10] T. Bai, C. Pan, Y. Deng, M. Elkashlan, A. Nallanathan and L. Hanzo, "Latency Minimization for Intelligent Reflecting Surface Aided Mobile Edge Computing," in IEEE Journal on Selected Areas in Communications, vol. 38, no. 11, pp. 2666-2682, Nov. 2020, doi: 10.1109/JSAC.2020.3007035.

[11] X.-Q. Pham and E.-N. Huh, "Towards task scheduling in a cloud-fog computing system", Proc. 18th Asia-Pacific Netw. Operations Manage. Symp., pp. 1-4, 2016.

[12] L. Yin, J. Luo and H. Luo, "Tasks scheduling and resource allocation in fog computing based on containers for smart manufacturing", IEEE Trans. Ind. Inform., vol. 14, no. 10, pp. 4712-4721, Oct. 2018.

[13] D. Chahal, B. Mathew and M. Nambiar, "Simulation based job scheduling optimization for batch workloads", Proc. ACM/SPEC Int. Conf. Perform. Eng., pp. 313-320, 2019.

[14] B. M. Nguyen et al., "Evolutionary algorithms to optimize task scheduling problem for the IoT based bag-of-tasks application in cloud–fog computing environment", Appl. Sci., vol. 9, no. 9, 2019.

[15] L. Liu, D. Qi, N. Zhou and Y. Wu, "A task scheduling algorithm based on classification mining in fog computing environment", Wireless Commun. Mobile Comput., vol. 2018, 2018.

[16] C. Tang et al., "A mobile cloud based scheduling strategy for industrial Internet of Things", IEEE Access, vol. 6, pp. 7262-7275, 2018.

[17] R. Z. Naeem, S. Bashir, M. F. Amjad, H. Abbas and H. Afzal, "Fog computing in Internet of Things: Practical applications and future directions", Peer-to-Peer Netw. Appl., vol. 12, no. 5, pp. 1236-1262, 2019.

[18] J. Xu, Z. Hao, R. Zhang and X. Sun, "A method based on the combination of laxity and ant colony system for cloud-fog task scheduling", IEEE Access, vol. 7, pp. 116 218-116 226, 2019.

[19] Lakshman Narayana,V., Lakshmi Patibandla, R.S.M., Pavani, V., Radhika, P. (2023). Optimized Nature-Inspired Computing Algorithms for Lung Disorder Detection. In: Raza, K. (eds) Nature-Inspired Intelligent Computing Techniques in Bioinformatics. Studies in Computational Intelligence, vol 1066. Springer, Singapore.

[20] Narayana, V.L., Sujatha, V., Sri, K.S., Pavani, V., Prasanna, T.V.N., Ranganarayana, K. (2023). Computer tomography image based interconnected antecedence clustering model using deep convolution neural network for prediction of covid-19. Traitement du Signal, Vol. 40, No. 4, pp. 1689-1696.

[21] Lakshman Narayana Vejendla and A PedaGopi, (2017),” Visual cryptography for gray scale images with enhanced security mechanisms”, Traitement du Signal,Vol.35, No.3-4,pp.197-208.

[22] V. L. Narayana, S. Bhargavi, D. Srilakshmi, V. S. Annapurna and D. M. Akhila, "Enhancing Remote Sensing Object Detection with a Hybrid Densenet-LSTM Model," 2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT), Greater Noida, India, 2024, pp. 264-269, doi: 10.1109/IC2PCT60090.2024.10486394.

[23] Lakshman Narayana, V., C.R. Bharathi. (2023). Efficient route discovery method in MANETs and packet loss reduction mechanisms. International Journal of Advanced Intelligence Paradigms, 2023 Vol.25 No.1/2.

[24] V. L. Narayana, S. Sirisha, G. Divya, N. L. S. Pooja and S. A. Nouf, "Mall Customer Segmentation Using Machine Learning," 2022 International Conference on Electronics and Renewable Systems (ICEARS), 2022, pp. 1280-1288, doi: 10.1109/ICEARS53579.2022.9752447.

[25] R. S. M. Lakshmi Patibandla, B. Tarakeswara Rao, V. Lakshman Narayana,11 - Prediction of COVID-19 using machine learning techniques,Editor(s): Deepak Gupta, Utku Kose, Ashish Khanna, Valentina Emilia Balas,Deep Learning for Medical Applications with Unique Data,Academic Press,2022,Pages 219-231,ISBN 9780128241455,https://doi.org/10.1016/B978-0-12-824145-5.00007-1.

[26] Patibandla, R.S.M.L., Vejendla, L.N.(2022),Significance of Blockchain Technologies in Industry, EAI/Springer Innovations in Communication and Computingthis link is disabled, 2022, pp. 19–31.

[27] H. R. Boveiri, R. Khayami, M. Elhoseny and M. Gunasekaran, "An efficient swarm-intelligence approach for task scheduling in cloud-based Internet of Things applications", J. Ambient Intell. Humanized Comput., vol. 10, no. 9, pp. 3469-3479, 2019.

[28] A. S. Kumar and M. Venkatesan, "Multi-objective task scheduling using hybrid genetic-ant colony optimization algorithm in cloud environment", Wireless Pers. Commun., vol. 107, no. 4, pp. 1835-1848, 2019.

[29] S. Bitam, S. Zeadally and A. Mellouk, "Fog computing job scheduling optimization based on bees swarm", Enterprise Inf. Syst., vol. 12, no. 4, pp. 373-397, 2018.

[30] S. Abdi, S. A. Motamedi and S. Sharifian, "Task scheduling using modified PSO algorithm in cloud computing environment", Proc. Int. Conf. Mach. Learn. Elect. Mech. Eng., pp. 8-9, 2014.

[31] . M. Ali, D. Essam and K. Kasmarik, "A novel differential evolution mapping technique for generic combinatorial optimization problems", Appl. Soft Comput., vol. 80, pp. 297-309, 2019.

[32] A. Hota, S. Mohapatra and S. Mohanty, "Survey of different load balancing approach-based algorithms in cloud computing: A comprehensive review" in Computational Intelligence in Data Mining, Singapore:Springer, pp. 99-110, 2019.