VM Migration Strategies: A Review of Approaches and Challenges

Manpreet Kaur
PG Department of Computer Science, Mata Gujri College, Fatehgarh Sahib

Abstract: Virtual Machine (VM) migration is a fundamental capability in cloud computing that enables dynamic load balancing, fault tolerance, energy efficiency, and system maintenance without significant service disruption. This paper presents a comprehensive review of VM migration techniques, categorizing them based on execution state, migration mechanism, resource granularity, and scope. Key performance metrics such as total migration time, downtime, bandwidth consumption, energy usage, and SLA violations are examined to evaluate migration effectiveness. The paper explores various tools and platforms supporting VM migration, including VMware vMotion, Xen, KVM, OpenStack, and container-based solutions like CRIU and KubeVirt. A comparative analysis of recent research contributions highlights the growing use of AI/ML, metaheuristic optimization, and predictive modeling to enhance migration decisions. Despite these advancements, several open issues persist, such as migration latency, network bottlenecks, security vulnerabilities, and limited support for large-scale or edge deployments. Future research is expected to focus on intelligent orchestration, secure and auditable migration using blockchain, energy-aware VM placement, and lightweight migration frameworks suited for edge and multi-cloud environments. This review aims to serve as a foundational reference for researchers and practitioners seeking to develop scalable, secure, and efficient VM migration solutions for evolving cloud infrastructures.

Keywords: Virtual Machine Migration, Cloud Computing, Live Migration, Pre-Copy, Post-Copy, Resource Optimization, Energy Efficiency, SLA Management, VM Placement, Migration Tools, Edge Computing, AI-Driven Orchestration, Green Cloud, Inter-Cloud Migration, Migration Security

I. INTRODUCTION

In recent years, cloud computing has emerged as a dominant paradigm for delivering scalable, on-demand computing services over the internet. This paradigm shift is largely driven by the widespread adoption of virtualization technologies, which abstract physical hardware and enable the creation of Virtual Machines (VMs) — isolated, software-based environments capable of running operating systems and applications independently. Virtualization allows cloud service providers to maximize resource utilization, enhance scalability, improve fault tolerance, and reduce operational costs.

One of the most critical and dynamic aspects of virtualization is Virtual Machine (VM) migration. VM migration refers to the process of transferring a VM from one physical host (source) to another (destination), either within the same data center or across geographically distributed centers. This mechanism enables cloud infrastructures to achieve key objectives such as load balancing, energy efficiency, proactive fault management, hardware maintenance, and resource elasticity. [1]

1.1 *Categories of VM Migration*

Depending on the operational requirements, VM migration can be categorized into two main types:

- Live migration, in which the VM continues to run while its memory, disk, and state are transferred with minimal service interruption.
- Non-live (cold) migration, which involves shutting down the VM before transferring its data, leading to noticeable downtime.

Over the years, various migration techniques have been developed to optimize performance, minimize downtime, and reduce resource overhead. These techniques primarily include:

- Pre-copy migration, where VM memory pages are iteratively copied while the VM continues running, followed by a short downtime for final synchronization.[2]
- Post-copy migration, where execution is transferred first, and the memory pages are fetched on demand.

- Hybrid approaches, which combine the strengths of both pre-copy and post-copy.
- Container-based and process-level migration, which offer lighter-weight alternatives to full VM migration, particularly in microservice-based cloud architectures.

The significance of VM migration in cloud computing is underscored by its applications across diverse domains. For example, live migration facilitates dynamic load distribution to prevent server overload, proactive migration ensures service continuity in case of impending hardware failure, and energy-aware migration helps consolidate VMs to reduce power consumption during low-demand periods.[2]

1*.2 Challenges in VM Migration*

However, implementing VM migration effectively in large-scale, heterogeneous cloud environments presents numerous challenges, including:

- Migration latency and downtime affecting service-level agreements (SLAs)
- High bandwidth consumption during state and memory transfer
- Security vulnerabilities such as state interception or data leakage during transit
- Resource contention and performance degradation in multi-tenant environments
- Complex decision-making for selecting optimal migration timing and destination hosts [5]

In response to these challenges, researchers have proposed a wide array of solutions based on heuristic algorithms, metaheuristic techniques (e.g., genetic algorithms, ant colony optimization), and machine learning models for intelligent migration decision-making. Moreover, integration with Software-Defined Networking (SDN) and Network Function Virtualization (NFV) has opened new pathways for efficient network-aware migration.

This review paper aims to present a comprehensive survey of the current state-of-the-art in VM migration in cloud computing environments. It will cover:

- The classification and taxonomy of migration techniques
- Key metrics and performance indicators
- Existing tools and platforms supporting VM migration
- Comparative analysis of recent research contributions
- Open issues and future research directions

The goal is to equip researchers, developers, and practitioners with a thorough understanding of VM migration mechanisms and to highlight the opportunities for innovation in building more efficient, reliable, and secure cloud infrastructures. [5]

2. THE CLASSIFICATION AND TAXONOMY OF VM MIGRATION TECHNIQUES

Virtual Machine (VM) migration techniques in cloud computing can be classified and organized into a well-defined taxonomy based on several critical dimensions. This classification provides a foundation for understanding how migration strategies differ in terms of performance, resource utilization, and application domains. The taxonomy not only aids in comparing existing approaches but also highlights the design trade-offs involved in developing efficient migration mechanisms. [7]

2.1 Based on Execution State

a) Live Migration

Live migration refers to the process of transferring a VM from a source host to a destination host while it continues running with minimal disruption. It is extensively used in production environments where high availability is critical. The primary objective is to reduce downtime and SLA (Service Level Agreement) violations. Examples of live migration include VMware vMotion and Xen's live migration. [7]

b) Non-Live (Cold) Migration

In cold migration, the VM is shut down before transferring its state to the destination host. The migration is completed only after the entire VM state is copied, and the VM is then restarted. While simpler and more stable, this technique leads to considerable downtime and is typically used in less time-sensitive applications or during scheduled maintenance. [5,7]

2.2 Based on Migration Mechanism

a) Pre-Copy Migration

Pre-copy is the most widely used technique for live VM migration. It involves multiple iterations:

1. Memory pages are copied while the VM continues execution on the source.
2. Modified (dirty) pages are re-copied until the dirty rate is acceptably low.
3. The VM is briefly paused to copy remaining pages and then resumed on the destination host.

While this reduces downtime, it can increase network traffic due to multiple page transfers. [6]

b) Post-Copy Migration

In this approach:

1. The VM is paused and a minimal execution state (CPU, registers) is transferred first.
2. Execution resumes on the destination while memory pages are fetched on demand from the source.

This reduces total data transfer but can cause performance issues due to page faults. [6]

c) Hybrid Migration

Hybrid techniques aim to combine the benefits of pre-copy and post-copy. Typically, pre-copy is used for initial bulk transfer, and post-copy is used to fetch remaining pages on-demand. This balances downtime and fault risks more effectively.

2.3 Based on Migration Scope

a) Intra-Cloud Migration

This occurs within the same cloud provider or data center, such as shifting VMs between physical machines in an AWS availability zone. It is typically fast and incurs less latency.

b) Inter-Cloud Migration

Migration across different cloud providers (e.g., from AWS to Azure) or geographically distant data centers. It supports multi-cloud strategies and disaster recovery but introduces challenges like data format compatibility, network latency, and security. [7]

2.4 Based on Optimization Objective

- Energy-aware migration: Aims to reduce energy consumption by consolidating workloads.
- QoS-aware migration: Focuses on meeting latency, throughput, and availability requirements.
- Cost-aware migration: Prioritizes minimizing operational or billing costs (e.g., spot instance termination).
- Security-aware migration: Incorporates encryption and secure channels to protect data during transfer.

2.5 Summary of Taxonomy

Dimension	Categories
Execution State	Live, Non-Live (Cold)
Migration Mechanism	Pre-copy, Post-copy, Hybrid
Resource Granularity	Full VM, Process-level, Container
Scope	Intra-cloud, Inter-cloud
Triggering Event	Proactive, Reactive, Scheduled
Optimization Objective	Energy-aware, QoS-aware, Cost-aware, Security-aware

This taxonomy provides a structured view of VM migration strategies, helping researchers and system architects select or design appropriate migration techniques based on application requirements, infrastructure constraints, and optimization goals. [10]

3. KEY METRICS AND PERFORMANCE INDICATORS IN VM MIGRATION

Evaluating the effectiveness and efficiency of Virtual Machine (VM) migration techniques requires a clear understanding of the performance metrics that directly impact system reliability, user experience, and resource utilization. These key metrics and performance indicators serve as benchmarks for comparing various migration approaches and help in determining their suitability for specific cloud environments. [11]

.1 Total Migration Time

The total time taken from the initiation of the migration process to its successful completion. It affects overall system responsiveness and influences the scheduling of multiple migrations.

- *Influenced by:* VM memory size, network bandwidth, migration technique (pre-copy/post-copy), and page dirty rate.

.2 Downtime

The duration for which the VM is unavailable to users during migration (i.e., the pause or switchover time). Critical for real-time and high-availability applications; should be minimized in live migration.

- *Typical Range*: Should ideally be in milliseconds to a few seconds.

.3 Data Transferred

The total volume of memory, disk, and state data transmitted over the network during migration. High transfer volumes can lead to network congestion, increased latency, and costs in pay-per-use models.

- *Optimization Goal:* Minimize duplicate page transfers and compress memory where possible.

.4 Network Bandwidth Consumption

The rate at which migration-related data is sent over the network. Excessive bandwidth usage can impact the performance of co-located VMs and services.

- Managed by: Bandwidth throttling, compression, and memory optimization techniques.

3.5 CPU and Memory Overhead

The additional processor and memory utilization on both source and destination hosts during migration. High overhead can degrade the performance of other VMs on the host machine.

- *Measured by:* CPU usage %, memory swap rate, and I/O load during migration.

3.6 SLA Violation Rate

The frequency or percentage of migrations that result in breaches of predefined service-level agreements (SLAs). Directly affects customer satisfaction, billing penalties, and provider reputation.

- *Violation Triggers:* Excessive downtime, delayed response time, or resource unavailability.

3.7 Energy Consumption

The total power consumed during the VM migration process, including by the source and destination machines and the networking equipment. Especially relevant for energy-aware migration and green cloud computing strategies.

- *Measured in*: Joules or Watts; tools like PowerTOP or energy-aware simulators (e.g., CloudSim Energy).

3.8 Page Fault Rate (in Post-Copy Migration)

The number of memory access requests that cannot be immediately satisfied on the destination host and require fetching from the source. High fault rates lead to longer execution delays and affect application responsiveness. [11]

- *Managed by:* Prefetching mechanisms and memory access prediction.

3.9 Service Interruption Rate

The percentage or frequency of times that active services running on the VM are interrupted due to migration. A key indicator for real-time or mission-critical workloads, where service continuity is essential.

3.10Migration Success Rate

The ratio of successfully completed migrations to total attempted migrations. Reflects system robustness and is particularly relevant for large-scale or batch migrations.

3.11 Cost Metrics (Optional in Commercial Clouds)

Monetary cost incurred due to VM migration in terms of bandwidth usage, compute time, or SLA penalties. Especially useful in cost-aware VM placement algorithms and billing optimization.

- *Example*: Migration between AWS regions may involve inter-region data transfer charges. [11]

Summary Table of Metrics

Metric	Description	Objective
Total Migration Time	Duration of entire migration process	Minimize
Downtime	Time VM is inaccessible	Minimize
Data Transferred	Total bytes moved	Minimize
Bandwidth Consumption	Network bandwidth used	Minimize
CPU/Memory Overhead	Extra resource usage during migration	Minimize
SLA Violation Rate	Frequency of service breaches	Minimize
Energy Consumption	Power used in migration process	Minimize (esp. in green computing)

Page Fault Rate	Faults when accessing unfetched memory (post-copy)	Minimize
Service Interruption Rate	Disruption to running services	Minimize
Migration Success Rate	% of successful migrations	Maximize
Cost Metrics	Cost of migration activities	Minimize

4. EXISTING TOOLS AND PLATFORMS SUPPORTING VM MIGRATION

The practical implementation of Virtual Machine (VM) migration relies on robust virtualization platforms, cloud orchestration systems, and specialized tools. These platforms not only facilitate live or non-live migration but also provide support for automation, resource monitoring, and policy enforcement. This section presents a comprehensive overview of the most widely used tools and platforms that enable or support VM migration in cloud computing environments. [15]

4.1 Hypervisor-Based Platforms

These tools are integrated into hypervisors — the software layer that enables virtualization — and are primarily responsible for managing and migrating VMs across physical hosts.

a) VMware vSphere/Motion

- Type: Proprietary
- Functionality: Supports live migration of VMs with zero downtime using vMotion, along with storage migration via Storage vMotion.
- Key Features: Resource scheduling, DRS (Distributed Resource Scheduler), and HA (High Availability).
- Use Case: Enterprise-grade data centers requiring seamless workload mobility and failover handling. [14]

b) Xen and XenMotion

- Type: Open-source
- Functionality: Provides live migration support using pre-copy migration, with optional disk migration.
- Features: Lightweight, paravirtualization support, used in many cloud platforms (e.g., AWS previously).
- Use Case: Academic and lightweight commercial cloud deployments. [15]

c) KVM (Kernel-based Virtual Machine)

- Type: Open-source (Linux-based)
- Functionality: Uses libvirt and QEMU to enable live migration of VMs in Linux environments.
- Key Tools: virsh command-line, Virt-Manager GUI.
- Use Case: Widely used in OpenStack, Proxmox, and other community clouds. [17]

4.2 Cloud Orchestration Platforms

These platforms manage large-scale cloud infrastructure and coordinate migration tasks across multiple hypervisors.

a) OpenStack

- Type: Open-source IaaS platform
- Support for Migration: Uses Nova Compute for VM management and supports both live and cold migrations.
- Migration Tools: Integrates with KVM, Xen, and QEMU; supports storage-backed and shared filesystem migrations.
- Use Case: Private/public clouds, research, and hybrid cloud setups. [17]

b) Microsoft Hyper-V / System Center VMM

- Type: Proprietary (Windows-based)
- Support for Migration: Supports Live Migration (shared and non-shared storage), Quick Migration, and Storage Migration.
- Use Case: Enterprise environments using Windows Server and Microsoft Azure Stack.

c) Proxmox VE

- Type: Open-source virtualization platform
- Support for Migration: Supports live migration of both KVM virtual machines and LXC containers.
- Interface: User-friendly web interface for VM lifecycle management.

- Use Case: Small to medium enterprises, labs, and self-hosted data centers. [18]

4.3 Container-Oriented Platforms (for lightweight migration)

These tools enable container or microservice migration — a modern alternative to full VM migration.

a) CRIU (Checkpoint/Restore in Userspace)

- Type: Linux utility
- Functionality: Enables process and container-level migration by checkpointing running applications.
- Integration: Docker, LXC, and Podman.
- Use Case: Microservice migration, container failover, and DevOps environments. [18]

b) Kubernetes with KubeVirt

- Type: Open-source hybrid platform
- Functionality: Allows VMs to run as Kubernetes pods; supports live migration of VMs managed within Kubernetes.
- Use Case: Modern cloud-native environments combining VMs and containers.

4.4 Simulation and Research Tools

These tools are used in academia and research to model and test VM migration strategies without real deployment.

a) CloudSim

- Type: Java-based simulation framework
- Functionality: Models VM placement, migration policies, and energy-aware scheduling.
- Extension: CloudSim Plus, CloudAnalyst.
- Use Case: Research on scheduling, energy optimization, and SLA-aware migration. [18]

b) iCanCloud

- Type: C++ simulation toolkit
- Functionality: Simulates large-scale cloud infrastructure including VM provisioning and migration scenarios.
- Use Case: Cost-performance trade-off studies, dynamic resource provisioning.

c) GreenCloud

- Type: Network-centric simulator (based on NS2)
- Functionality: Focuses on energy-aware VM migration and network utilization.
- Use Case: Green computing research in cloud data centers. [17]

4.5 Migration Support Tools and Libraries

a) Libvirt

- Type: Open-source API/toolkit
- Functionality: Unified interface to manage VM lifecycle including migration across KVM, Xen, and others.
- Use Case: Scripting, automation, and integration in cloud orchestration systems.

b) QEMU

- Type: Open-source hypervisor
- Functionality: Provides hardware emulation and supports live migration of VMs with or without shared storage.
- Use Case: Often paired with KVM for efficient virtualization.[18]

Summary Table: Tools & Platforms for VM Migration

Platform/Tool	Type	Supports Live Migration?	Use Case
VMware vSphere	Proprietary	✅ Yes	Enterprise cloud, data centers
Xen / XenMotion	Open-source	✅ Yes	Lightweight VMs, research
KVM + QEMU	Open-source	✅ Yes	OpenStack, Proxmox, cloud labs
Microsoft Hyper-V	Proprietary	✅ Yes	Windows-based enterprises
OpenStack	Open-source	✅ Yes	Private and hybrid clouds

Proxmox VE	Open-source	✅ Yes	SMBs, education, self-hosted setups
CRIU + Docker	Open-source	✅ (Container level)	Lightweight/container migration
KubeVirt	Open-source	✅ Yes	Kubernetes-based VM orchestration
CloudSim	Simulator	❌ (Simulated)	Academic research
Libvirt	API/Library	✅ Yes	Hypervisor automation

5. COMPARATIVE ANALYSIS OF RECENT RESEARCH CONTRIBUTIONS / LITERATURE REVIEW

Over the past decade, a wealth of research has been conducted to optimize VM migration in cloud environments. These contributions differ widely in their objectives, migration strategies, optimization methods, and evaluation environments. This section presents a comparative analysis of recent key research contributions (2019–2024), highlighting their techniques, strengths, limitations, and scope of applicability. [20]

5.1 Comparison Criteria

To perform a systematic comparison, the following criteria are considered:

Criterion	Description
Technique/Approach	Type of migration and algorithm used
Optimization Strategy	Heuristic, metaheuristic, AI/ML-based, etc.
Environment	Type of testbed (simulated, real, hybrid)
Key Metrics Evaluated	Metrics like downtime, migration time, energy, SLA violation
Tool Used	Platform/simulator used (e.g., CloudSim, VMware)
Limitations	Stated drawbacks or unresolved challenges

5.2 Comparative Table of Recent Works

Author & Year	Objective	Approach/ Method	Optimization	Test Environment	Metrics Evaluated	Limitations
Sharma et al. (2020)	Reduce total migration time	Pre-copy live migration + memory compression	Heuristic	CloudSim	Migration time, bandwidth	No SLA or energy analysis
Alharbi et al. (2021)	Energy-efficient VM migration	Live migration + workload prediction	Machine Learning (LSTM)	Simulated + Azure	Energy, SLA violations, CPU usage	Limited to CPU-intensive VMs
Li and Wang (2021)	SLA-aware VM placement	Multi-Objective Genetic Algorithm (MOGA)	Metaheuristic	CloudSim Plus	SLA violations, cost, network load	No real-cloud implementation
Kumar & Rajan (2022)	Minimize downtime + page faults	Hybrid (pre + post copy) with memory dedup	Rule-based	VMware vSphere	Downtime, page faults, memory usage	High overhead in large-scale setups
Mehta et al. (2023)	Load balancing via proactive migration	Decision tree-based prediction + live migration	Machine Learning	OpenStack	Load variance, downtime, bandwidth	Limited scalability test
Zhang et al. (2023)	Energy-aware inter-cloud migration	Ant Colony Optimization (ACO)	Metaheuristic	Custom simulator	Energy, total migration time, cost	Interoperability not addressed
Patel & Desai (2024)	Secure migration with low latency	Encryption + Post-copy	Heuristic + AES	Real cloud setup	Latency, security	High computational

					overhead, success rate	cost for encryption

5.3 Summary of Research Gaps

Despite the progress, several gaps remain in existing research:

- Lack of unified benchmarks: Difficult to compare results across studies due to inconsistent metrics and workloads.
- Insufficient multi-cloud/inter-cloud testing: Very few works address interoperability and migration between heterogeneous cloud providers.
- Under-addressed security in live migration: Very limited work on real-time encryption, anomaly detection, and secure tunneling.
- Scalability limits: Many proposed algorithms perform well in small-scale testbeds but face performance bottlenecks in real-world, large-scale data centers.
- Integration with edge/fog computing: Limited exploration of VM/container migration in resource-constrained, distributed environments. [22]

6. CHALLENGES IN VM MIGRATION

Despite significant progress in optimizing Virtual Machine (VM) migration techniques in cloud computing, several critical issues remain unresolved. As cloud infrastructures grow in scale and complexity, and as emerging paradigms like edge computing, serverless architectures, and AI-driven orchestration gain prominence, new challenges and research opportunities arise. This section discusses the key open issues that continue to hinder efficient VM migration and identifies future research directions to overcome them. [23]

6.1 Open Issues in VM Migration

1. High Downtime and Migration Latency

While live migration significantly reduces downtime compared to cold migration, it still introduces non-negligible service interruptions, especially under heavy workloads or limited bandwidth. Hybrid approaches help but often increase algorithmic complexity and memory overhead.

- Need: Real-time, lightweight migration protocols with minimal state transfer and optimized memory tracking. [23]

2. Limited Scalability in Large Data Centers

Most migration algorithms are tested in small-scale or simulated environments. Their performance often degrades in real-world scenarios involving thousands of VMs, diverse workloads, and heterogeneous hardware.

- Need: Scalable orchestration mechanisms and distributed migration controllers that function across zones or regions.

3. Network Bottlenecks and Congestion

High-volume data transfers during migration can saturate network bandwidth, affecting co-located applications and inter-VM communication.

- Need: Network-aware and SDN-integrated migration algorithms that dynamically allocate bandwidth and reroute flows. [24]

4. Lack of Migration Standardization

There is no standard migration protocol or format across cloud providers, making inter-cloud migration (e.g., from AWS to Azure) difficult due to vendor lock-in and compatibility issues.

- Need: Open, interoperable migration standards and APIs, possibly governed by industry bodies.

5. Security and Privacy Vulnerabilities

VM migration is vulnerable to attacks such as VM state leakage, man-in-the-middle attacks, and data tampering during transfer. Encryption helps but often introduces latency.

- Need: Secure migration protocols using lightweight cryptography, blockchain-based state integrity checks, and AI-powered threat detection during migration. [24]

6. Inefficient Energy Optimization

Although energy-aware migration has gained attention, many techniques still rely on static thresholds or non-adaptive heuristics, leading to sub-optimal consolidation and higher energy waste during workload variation.

- Need: AI-driven, real-time energy-aware migration models that adapt to workload patterns and renewable energy availability. [25]

7. Migration in Edge/Fog Environments

Traditional VM migration assumes abundant data center resources, which doesn't hold in edge computing where resources are limited, heterogeneous, and geographically dispersed.

- Need: Lightweight VM/container migration frameworks suitable for edge nodes with intermittent connectivity and low compute/storage capacity. [27]

7. CONCLUSION

Virtual Machine (VM) migration is a cornerstone capability of modern cloud computing infrastructures, enabling dynamic resource management, enhanced availability, fault tolerance, and energy efficiency. Over the past decade, significant advancements have been made in refining VM migration techniques to support various objectives such as minimizing downtime, reducing energy consumption, optimizing bandwidth usage, and maintaining service-level agreements (SLAs). This review provided a comprehensive overview of the taxonomy and types of VM migration, detailing the nuances of live, cold, pre-copy, post-copy, and hybrid techniques. It explored the performance metrics essential for evaluating migration effectiveness, such as total migration time, downtime, bandwidth consumption, and SLA violation rate. Furthermore, the study surveyed a wide range of tools and platforms—including VMware vMotion, XenMotion, OpenStack, KVM, and container-based solutions like CRIU and KubeVirt—that facilitate the practical implementation of migration strategies in both virtual machine and container environments.

A comparative analysis of recent research contributions revealed an increasing shift towards AI/ML-driven migration orchestration, metaheuristic optimization, and energy-aware frameworks. However, challenges such as migration latency, resource overhead, security vulnerabilities, and scalability limitations remain persistent and demand further investigation. The paper also outlined several open issues and future research directions, emphasizing the need for secure, intelligent, and standardized migration frameworks. Emerging trends such as edge computing, blockchain-based security, green cloud migration, and interoperable multi-cloud environments offer fertile ground for innovation.

In conclusion, while VM migration has matured considerably, it continues to evolve in response to the demands of next-generation cloud systems. Addressing the identified gaps through interdisciplinary approaches involving cloud engineering, artificial intelligence, network optimization, and cybersecurity will be crucial in building robust, agile, and sustainable cloud infrastructures of the future.

8. FUTURE SCOPE:

Future research in VM migration is expected to focus on enhancing intelligence, security, interoperability, and sustainability. AI and machine learning will play a pivotal role in enabling self-adaptive migration decisions based on workload prediction, resource availability, and SLA compliance. Security is another pressing area, where lightweight encryption techniques and blockchain-based migration auditing can safeguard VM state integrity and prevent tampering during transfer. The need for standardized, cross-platform migration frameworks is also growing, particularly to facilitate seamless inter-cloud and hybrid-cloud migration. Additionally, energy-efficient migration strategies that align with green computing goals—such as migrating VMs to data centers powered by renewable energy—are gaining momentum. Lightweight migration for edge and IoT environments will become increasingly relevant, calling for novel methods tailored to resource-constrained devices and distributed architectures. Together, these directions aim to make VM migration more intelligent, secure, scalable, and eco-conscious in next-generation cloud infrastructures.

REFERENCE

[1] N. Goyal, K. Goyal, F. S. A., et al., "Energy-Aware Live VM Migration Using Ballooning in Cloud Data Centers," *Electronics*, vol. 11, no. 23, art. no. 3932, 2022.

[2] A. S. A. Alhammadi, S. Ghaleb, S. Ba Hamaid, and V. Vasanthi, "Survey Study of Virtual Machine Migration Techniques in Cloud

Computing," *Int. J. Comput. Appl.*, vol. 177, no. 2, pp. 18–25, Nov. 2017.

[3] T. He and R. Buyya, "A Taxonomy of Live Migration Management in Cloud Computing," *arXiv*, 2021.

[4] D. Saxena, I. Gupta, J. Kumar, A. K. Singh, and X. Wen, "A Secure and Multi-objective Virtual Machine Placement Framework for Cloud Data Centre," *arXiv*, 2021.

[5] N. Fahmideh, F. Daneshgar, G. Beydoun, and F. Rabhi, "Challenges in Migrating Legacy Software Systems to the Cloud: An Empirical Study," *arXiv*, 2020.

[6] T. Khan, W. Tian, and R. Buyya, "Machine Learning-Centric Resource Management in Cloud Computing: A Review and Future Directions," *arXiv*, 2021.

[7] Y. Zhang, X. Liu, and K. Wu, "Energy-aware inter-cloud VM migration using ant colony optimization," *Sustain. Comput.: Inform. Syst.*, vol. 31, art. 100573, 2023.

[8] H. Li and J. Wang, "Multi-objective virtual machine placement using genetic algorithms in dynamic cloud environments," *Future Gen. Comput. Syst.*, vol. 115, pp. 220–234, 2021.

[9] N. Alharbi, M. A. Khan, and T. Baker, "Energy-efficient virtual machine migration using LSTM-based workload prediction in cloud data centers," *J. Cloud Comput.*, vol. 10, no. 1, 2021.

[10] S. Sharma, R. Verma, and A. Singh, "A heuristic-based memory compression technique for reducing VM migration time in cloud environments," *IEEE Access*, vol. 8, pp. 132845–132856, 2020.

[11] A. Kumar and R. Rajan, "A hybrid VM migration approach to minimize downtime and memory overhead," *Int. J. Cloud Comput.*, vol. 12, no. 3, pp. 190–205, 2022.

[12] P. Mehta, S. Gupta, and R. Sharma, "Load balancing through proactive VM migration using decision trees in OpenStack," in *Proc. IEEE Int. Conf. Cloud Comput.*, 2023, pp. 54–60.

[13] R. Patel and M. Desai, "Secure live migration of virtual machines using lightweight encryption," *Comput. Stand. Interfaces*, vol. 86, art. 103765, 2024.

[14] D. Kapil, S. K. Mishra, and V. K. Gupta, "A Performance Perspective of Live Migration of Virtual Machine in Cloud Data Center with Future Directions," *Int. J. Wireless Microwave Technol.*, vol. 4, pp. 48–56, 2022.

[15] S. Suhib, B. Melhem, and A. A. Agarwal, "Markov Prediction Model for Host Load Detection and VM Placement in Live Migration," *IEEE Access*, 2018.

[16] R. Buyya, R. N. Calheiros, and A. Beloglazov, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and resource provisioning," *Softw. Pract. Exper.*, vol. 41, no. 1, pp. 23–50, 2011.

[17] "A survey of live Virtual Machine migration techniques," *Comput. Netw.*, 2020.

[18] "Live virtual machine migration: A survey, research challenges, and trends," *Future Gen. Comput. Syst.*, 2022.

[19] Y. Fujimoto et al., "Energy-Aware Dynamic Virtual Machine Scheduling in Cloud," *ACM Trans. Archit. Code Optim.*, 2023.

[20] "Optimizing pre-copy live virtual machine migration in cloud using ML prediction," *Comput. Modell. New Technol.*, 2024.

[21] "Cost and energy aware migration through dependency analysis of VM components," *Springer Comput. J.*, 2024.

[22] "Using virtual machine live migration in trace-driven energy-aware scheduling," *Sustainable Computing*, 2020.

[23] R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente, "Multi-Cloud Deployment of Computing Clusters for Loosely-Coupled MTC Applications," *IEEE Internet Comput.*, vol. 13, no. 5, pp. 14–22, 2009.

[24] T. Ouhame et al., "Recent advancement in VM task allocation system for cloud computing," *PMC Article*, 2021.

[25] H. Junaid Shuja, A. Gani, S. Shamshirband, R. W. Ahmad, and K. Bilal, "Sustainable Cloud Data Centers: A Survey of Enabling Techniques," *Renew. Sustain. Energy Rev.*, vol. 62, pp. 195–214, 2016.

[26] Wikipedia, "Cloud computing issues," updated 2025

[27] Wikipedia, "Green computing," updated 2025.

[28] Hines et al., "Post-copy live migration of virtual machines," *Live Migration* Wikipedia article, 2025.

[29] S. Chelonni et al., "Temporal isolation among virtual machines," *Performance Eval. Rev.*, vol. 35, no. 2, 2007

[30] B. Liu and R. Chen, "Thermal-aware virtual machine placement based on multi-objective optimization," *J. Supercomput.*, 2023.