UDC 004.7 **doi:** https://doi.org/10.20998/2522-9052.2026.2.04

Nina Kuchuk[1], Andrii Shyshatskyi[2], Viacheslav Radchenko[3], Yuliia Andrusenko[3], Sergii Klivets[4]

[1] National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine
[2] Kharkiv National Automobile and Highway University, Kharkiv, Ukraine
[3] Kharkiv National University of Radio Electronics, Kharkiv, Ukraine
[4] Ivan Kozhedub Kharkiv National Air Force University, Kharkiv, Ukraine

DESIGN OF A MULTILEVEL ARCHITECTURE FOR OPTIMIZING VIRTUAL MACHINE MIGRATION

Abstract. The paper considers the problem of designing a multi-tiered control structure for optimizing virtual machine migration processes in virtualized data centers. **The relevance of the study** is due to the growth of computing workloads, resource heterogeneity, and the need to ensure high performance and energy efficiency of the infrastructure while maintaining QoS (quality of service). Inefficient VM migration can lead to node overload, increased delays, and additional resource costs. **The purpose of the paper** is to develop a multi-tiered virtual machine migration management model that provides adaptive resource allocation, reduced downtime, and minimized costs for moving virtual machines. **The object of the study** is the processes of functioning of a virtualized data center, and the subject is methods and models for optimizing VM migration in a multi-tiered control architecture. The proposed structure provides for strategic, tactical and operational levels of management, which allows combining long-term resource planning with operational response to load changes. The work takes into account the criteria of load balancing, energy efficiency, network traffic minimization and SLA (Service Level Agreement) provision. **The results of the study** can be used in the design of cloud and grid infrastructures to increase the efficiency of computing resources and ensure stable operation of services under dynamic load conditions. **The areas of further research** are the implementation of intelligent decision-making algorithms and the use of simulation modeling to assess the effectiveness of the proposed structure.

Keywords: virtual machine; multi-tier architecture; resource optimization; virtualized data center; load balancing; cloud computing; heterogeneous environment; energy efficiency; quality of service; adaptive management**.**

Introduction

Problem relevance. The digital transformation of modern business and the rapid growth of data volumes have led to a massive transition to cloud computing and virtualized environments [1, 2]. Virtualization has become the foundation of IT infrastructure, allowing companies to flexibly manage resources, reduce equipment costs and ensure high availability of services [3, 4]. However, with the evolution of systems, there is a critical need for dynamic load redistribution, which makes virtual machine (VM) migration one of the most complex and important processes in data center (DCC) management [5].

The migration process, especially on the scale of large corporate networks, faces a number of technical challenges: limited bandwidth of communication channels, the risk of loss of data integrity and, most importantly, the need to minimize downtime [6, 7]. Traditional migration methods often do not take into account the specifics of loads or network topology, which leads to degradation of the performance of critical business applications. The relevance of this topic is due to the rapid transition of businesses to hybrid and multi-cloud infrastructures. In 2026, simple "migration" of data will no longer meet the needs of the market - it must be intelligent, fast and invisible to the user [8, 9].

Designing a multi-tier structure allows you to separate the control logic from the direct data transfer, implement intelligent compression and queuing algorithms, which ultimately optimizes the use of network and computing resources [10, 11]. Such an architectural approach is the key to creating fault-tolerant and adaptive IT infrastructures of the new generation.

Literature review. Paper [11] considers energy-efficient consolidation of virtual machines in cloud data centers. Adaptive heuristics for dynamic resource allocation are proposed, taking into account performance and energy consumption. However, it does not investigate multilevel control structures for VM migration processes. Study [13] proposes methods for managing overloaded hosts under QoS constraints. Mechanisms for initiating virtual machine migration to ensure system stability are analyzed. However, it does not address hierarchical decision-making levels in migration processes. In [14], an approach to virtual machine resource management using workload prediction is presented. The authors analyze the trade-off between performance and energy efficiency. However, it does not investigate multilevel models for VM migration control. Paper [15] presents the Sandpiper system for virtual machine resource management. An automated VM migration approach based on system state monitoring is proposed. However, it does not consider strategic levels of migration management. In [16], the impact of network traffic on virtual machine placement and migration is studied. Methods for optimizing VM distribution to reduce network congestion are proposed. However, it does not investigate comprehensive multilevel approaches to VM migration optimization. Study [17] addresses the problem of virtual machine placement considering multiple optimization criteria. Algorithms ensuring a balance between performance and resource costs are proposed. However, it does not consider adaptive multilevel control structures. Paper [18] presents an approach to resource pool management using reactive and proactive strategies. The effectiveness of different resource management policies is analyzed. However, it does not investigate VM migration processes within a multilevel architecture. In [19], general challenges of cloud computing and resource management

are discussed. Key directions for optimization, including virtual machine migration, are identified. However, it does not consider specialized multilevel models for VM migration control.

Research object: processes of functioning and resource management in virtualized computing environments during dynamic load redistribution.

Research subject: methods, architectural models and algorithms of multi-tier migration layer management to optimize the transfer of virtual machine states.

Purpose of the work: increasing the efficiency of cloud infrastructures by developing and implementing a multi-tier migration layer architecture that minimizes service downtime and rationally uses network resources.

To achieve the goal, it is necessary to solve the **following tasks**:

1. Analyze existing migration technologies (Live Migration) and identify their bottlenecks in conditions of highly loaded systems.
2. Develop the concept of a multi-level structure that separates the levels of management, processing and direct data transportation.
3. Propose an algorithm for dynamic selection of migration parameters depending on the priority of virtual machines and the state of communication channels.
4. Conduct modeling (or experimental verification) to confirm the effectiveness of the proposed architecture compared to standard solutions.

1. The role of virtualization in modern cloud infrastructures and data centers

At the current stage of information technology development, virtualization is a key tool for building flexible, scalable, and cost-effective IT infrastructures [20, 21]. It allows you to abstract hardware resources (processor, memory, disk space, network) from software, which ensures the launch of multiple independent operating systems on a single physical server [22, 23].

Let's consider the key aspects of the role of virtualization. Resource consolidation assumes that instead of using dozens of servers with a low level of load (usually 10-15%), virtualization allows you to combine them into powerful resource pools. This radically reduces the cost of equipment, power supply and cooling in data centers (DPCs) [24, 25].

Dynamic load management takes into account the fact that virtualization turns the physical infrastructure into a "software-defined" one. This allows you to instantly allocate resources for new tasks and automatically redistribute them in the event of peak loads [26, 27].

Thanks to virtualization, the failure of one physical node does not lead to a stop of business processes. Virtual machines (VMs) can automatically restart on other healthy nodes of the cluster.

Each VM runs in its own isolated environment. This is critical for multi-tenant cloud platforms, where the resources of one physical server can be used by different clients without posing a threat to each other's data [28, 29].

Virtualization has become a technological "engine" for IaaS (Infrastructure as a Service), PaaS, and SaaS models. It allows you to implement the concept of cloud elasticity - the ability of the system to automatically expand or contract depending on the real needs of the user [30, 31].

It is thanks to virtualization that the possibility of migrating computing processes in real time arose. Without this technology, it would be impossible to imagine modern load balancing strategies and server maintenance without stopping services, which leads us to the need to study VM migration mechanisms in detail [32, 33].

2. Analysis of existing migration technologies

Virtual machine migration is the process of transferring an active or stopped VM from one physical node (host) to another. Depending on the state of the VM and the data transfer method, migration technologies are divided into three main categories.

Offline Migration [34]. This is the simplest method, in which the VM is completely stopped or put into a deep sleep state (suspend). After that, all data (RAM contents and disk images) are copied to the target host, where the machine is started again. Its advantages include guaranteed data integrity. And the disadvantages are that there is significant downtime, which is unacceptable for critical services.

Live Migration [35]. Allows you to migrate VMs without noticeable interruption to user work. The main difficulty here is the synchronization of RAM, which is constantly changing. There are two approaches here: Pre-copy and Post-copy.

Pre-copy. The hypervisor copies memory pages to the new host while the VM continues to run on the old one. Pages that changed during the copy (dirty pages) are re-sent in several iterations. When the remaining memory becomes minimal, the VM is stopped for a moment for final synchronization. This is standard for VMware vMotion and KVM.

Post-copy. The VM is stopped on the original host, its minimal state (CPU, registers) is transferred to the new node, and it is immediately started there. The remaining memory pages are pulled from the original host on request ("page faults").

Storage Migration [36]. Transferring only the VM disks between different storage systems without changing the compute node. Often combined with live migration to completely transfer the VM between different data centers.

A comparative analysis of the considered methods is presented in Table 1.

Table 1 – **Comparative characteristics of methods**

Characteristic	Offline	Live (Pre-copy)	Live (Post-copy)
Downtime	High	Minimal	Very low
Failure Risk	Almost zero	Medium (due to iterations)	High (if the network crashes)
Network Load	Stable	Stable High (reusable)	Optimal

Although existing methods provide basic VM mobility, they have significant limitations: Pre-copy can become infinite at high memory write intensity, and Post-copy introduces large delays when accessing memory over the network. This confirms the need to develop an

optimized migration layer that can flexibly combine these methods and use additional optimization mechanisms.

3. Research into critical factors of migration efficiency

To objectively assess any virtual machine migration process, a set of quantitative and qualitative indicators are used. These factors determine how “seamless” the migration is for the end user and how it affects the overall network performance.

Let's consider the key performance metrics [37].

Downtime. The time interval during which the VM completely stops executing instructions and providing services. In modern systems, this indicator is tried to be reduced to milliseconds (usually < 100 ms to preserve TCP connections).

Total Migration Time. The time from the moment of initialization of the migration command to the complete release of resources on the source host. This indicator is critical for mass migrations (for example, during a planned shutdown of a server rack).

Migration Traffic. The total number of bytes transferred over the network. Due to the iterative copying of “dirty” pages in the Pre-copy method, this volume can exceed the actual amount of RAM of the VM several times.

Performance Degradation. Slowing down the speed of applications inside the VM during migration due to the use of processor and memory resources for the needs of the hypervisor.

Factors that negatively affect efficiency also play an important role.

Dirty page rate. If an application in a VM changes data in memory too quickly, the copying process can become endless, since new changes appear faster than the network can transfer them.

Network bandwidth. Limiting the communication channel between hosts is the main "bottleneck" that increases the overall migration time.

Latency. High ping between nodes critically affects the Post-copy method, causing the application interface to freeze with each access to remote memory.

The analysis shows that there is no ideal technology: minimizing one indicator often leads to a deterioration of the other. For example, reducing Downtime can lead to an increase in Migration Traffic. This creates a need for intelligent optimization, which, based on a multi-tiered structure, could adapt to the current network state and the type of load on the VM.

4. Justification of the need to transition to multi-tier architectures

Based on the metrics analysis performed in the previous subsection, we can derive a basic mathematical model of downtime for the classic Pre-copy algorithm. The downtime Tdown is defined as the time required to transfer the last portion of "dirty" memory pages and processor state:

$$T_{down} = \frac{V_{mem}^{(n)} + V_{state}}{B}, \quad (1)$$

where $V_{mem}^{(n)}$ is amount of memory remaining to transfer on the final iteration; V_{state} is amount of processor and device status data; B is available network bandwidth.

The problem is that with a high application memory update rate (D – dirty rate), the amount of data at each iteration may not decrease, but increase if $D > B$.

Standard solutions have exhausted their resource due to monolithicity, lack of intermediate processing and ignoring the network topology.

Monolithicity. Traditional hypervisors use hard-wired migration algorithms that cannot be quickly adapted to a specific type of traffic (for example, databases vs. web servers).

Lack of intermediate processing. In standard schemes, data is transferred "as is", without taking into account the possibility of intelligent on-the-fly compression or deduplication between similar VMs.

Ignoring network topology. Most systems consider the network as a "black box", not taking into account the multi-tiered structure of modern data centers (Leaf-Spine architecture).

The transition to a multi-tiered migration layer architecture allows dividing the holistic problem into sub-tasks, each of which is optimized separately. Analytics layer: predicts VM behavior and chooses the best moment to start. Data optimization layer: reduces the physical volume of traffic through adaptive algorithms. Transmission layer: dynamically manages network priorities (QoS). The review confirms that existing migration methods have fundamental limitations when working in highly loaded dynamic environments. This creates a scientific and practical need to develop a new architectural model – a multi-tiered migration layer, which, due to the specialization of layers, will provide a balance between migration speed and minimal impact on the operation of services.

5. The concept of hierarchical construction of the migration layer

Traditionally, migration is considered a function of the hypervisor. We propose to move it to a separate migration Layer, which works between the physical resource level and the virtual machine level.

The main idea is to decompose a complex process into three independent but interconnected levels. This allows you to optimize each stage of migration separately, without creating excessive load on the entire system at the same time.

Looking in detail at the logical structure of the layer, you can distinguish three components (Fig. 1).

The upper layer (Decision & Control Layer) is the “Brain” of the system. It collects metrics from the entire infrastructure and decides which VM to migrate, where exactly and according to what algorithm.

The middle layer (Processing & Optimization Layer) is the “Filter”. Here, data is prepared for transmission: compression, deduplication and caching of frequently used memory blocks.

The lower layer (Transport & Security Layer) is the “Channel”. Responsible for fast and secure transmission of bytes over the network, controlling the channel width and traffic priorities.

Let's take a closer look at the functional components of each layer to understand how optimization is achieved.

The management and orchestration layer uses machine learning methods or threshold algorithms to analyze the state of hosts. Its functions are to predict load "peaks", select the target node with the lowest latency, monitor the dirty page rate (memory change intensity) to select the migration method (Pre-copy or Post-copy).

At the data preparation and processing layer, the main savings in network traffic occur. Adaptive compression occurs if the host processor is free, complex compression algorithms are used (for example, LZ4 or Zstd). If the CPU load is high, compression is turned off to maintain performance. If we migrate 10 identical VMs (for example, from Windows 11), this layer notices identical memory blocks and transfers them only once, deduplication occurs on the fly.

At the transport layer, the so-called Traffic Shaping and Multi-path TCP interact directly with the network equipment.

Traffic Shaping allocates a guaranteed bandwidth for migration so that it does not "choke" client traffic.

Multi-path TCP uses multiple network interfaces simultaneously to accelerate the transfer of large amounts of data.

Implementing such a structure allows the system to be adaptive. For example, if the network is overloaded, the middle layer increases compression. If the network is free, but the processor is loaded, the system switches to direct transfer mode.

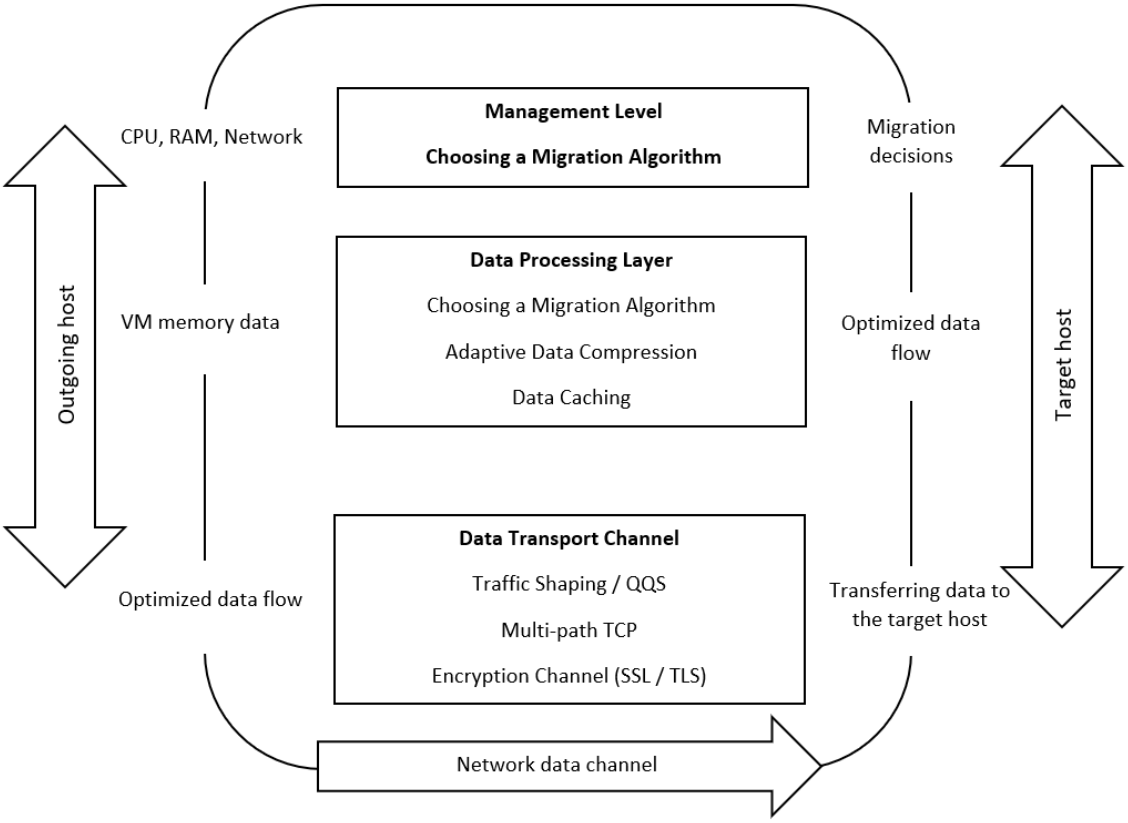

Fig. 1. Structural diagram of a multi-tiered migration layer optimization architecture

6. Mathematical model for optimizing the VM state transfer process

Even if the data update rate in a virtual machine (D) exceeds the physical bandwidth of the channel ($D > B$), migration is still possible and efficient if the product of the compression and deduplication coefficients ensures that the inequality is fulfilled. This mathematically proves that the multi-tier structure makes the system more resistant to high loads inside the VM.

Consider the data reduction model. Let: V_{mem} is the initial amount of RAM for the virtual machine; σ is data compression ratio ($0 < \sigma \leq 1$); δ is deduplication ratio, which characterizes the proportion of unique blocks ($0 < \delta \leq 1$). Then the actual amount of data that needs to be transmitted by the network after optimization is defined as

$$V_{opt} = V_{mem} \cdot \sigma \cdot \delta, \tag{2}$$

where V_{opt} is optimized data transfer volume.

Formula (2) reflects the reduction in data volume due to the use of compression and deduplication mechanisms.

Modified downtime model. Let η – transport layer efficiency coefficient ($\eta \geq 1$). Then, the optimized downtime of the virtual machine during migration, taking into account formula (1), is determined as follows:

$$T_{down}^{(opt)} = \frac{V_{mem}^{(n)} \cdot \sigma \cdot \delta + V_{state}}{B \cdot \eta}. \tag{3}$$

Formula (3) takes into account the reduction in the amount of transmitted data due to the optimization mechanisms of the second level of the architecture.

Pre-copy iteration convergence condition. In the classical Pre-copy migration model, the condition for convergence of iterations has the form: $D < B$. In the proposed multi-level architecture, taking into account data optimization, the convergence condition is modified:

$$D \cdot \sigma \cdot \delta < B \cdot \eta. \quad (4)$$

This means that applying compression and deduplication expands the allowable load range on the system.

Overall transmission optimization factor. To assess the efficiency of the architecture, we introduce an integral transmission optimization coefficient:

$$K_{opt} = \frac{V_{mem}}{V_{opt}}. \quad (5)$$

Substituting (2), we get

$$K_{opt} = \frac{1}{\sigma \cdot \delta}. \quad (6)$$

The greater the value K_{opt}, the higher the efficiency of optimizing VM state transfer.

7. VM Migration Management Algorithm

Now, having a mathematical basis, we can describe the algorithm by which the management layer works (Fig. 2).

First, monitoring is performed, i.e., collecting the values of *D*, *B* and the load on the host CPU. Then, a strategy is selected. If B (network) is a bottleneck: activate maximum σ (compression) and δ (deduplication). If the host CPU is overloaded: reduce the compression intensity (σ) so as not to slow down the operation of other VMs. If D (dirty rate) is critically high: switch from Pre-copy to Post-copy using the deduplicated cache.

Next, we describe the main processes that occur during the dynamic management of the migration layer.

Start and initialization. The system receives a VM migration command and starts the metrics monitoring process:

– *metrics collection block*, polling hypervisor sensors for values: Dirty Page Rate (*D*, memory refresh rate), Bandwidth (*B*), CPU Load (host processor load);

– *checking the convergence condition: Main logical branch:* if the amount of data generated is greater than the network capacity ($D > B$), the system switches to aggressive optimization;

– *parameter correction (Level 2):* deduplication is enabled to eliminate duplicate blocks; a high compression ratio is set (e.g. Zstd);

– *checking CPU usage:* if compression is too CPU intensive, the algorithm automatically reduces the complexity of the compression algorithm so as not to "slow down" the VM itself;

– *transport Management (Layer 3):* setting QoS priorities and opening additional TCP streams to maximize channel utilization;

– *loop to completion:* the algorithm repeats the check at each memory copy iteration until the remaining data is small enough to finally stop (Stop-and-Copy) and complete the migration.

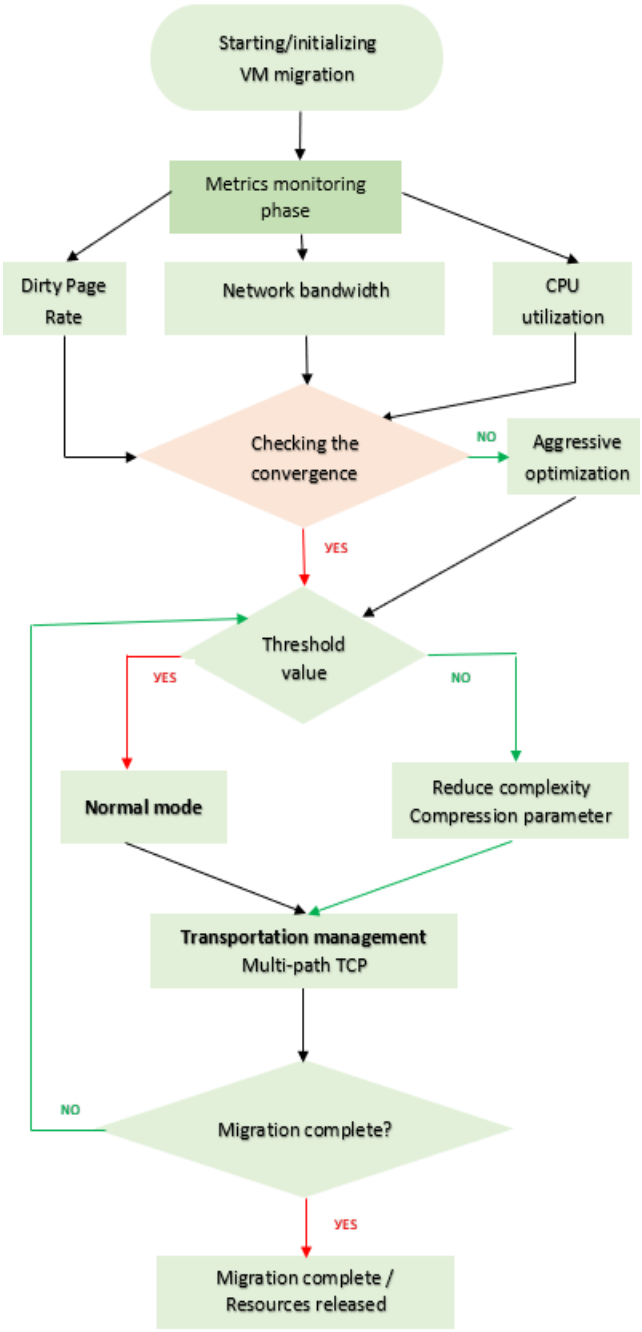

Fig. 2. Flowchart of the dynamic migration layer control algorithm

8. Discussion of the research results

The advantages of the developed algorithm include stability, autonomy and efficiency. Migration will not be interrupted due to the error "memory synchronization is impossible". The system itself balances the load on the processor and the network speed. Minimal downtime due to the accurate calculation of the final switching moment.

To implement and test a multi-tier structure, it is most appropriate to use one of two approaches: real environment and simulation modeling. Using the KVM/QEMU hypervisor with the Proxmox add-on. This allows you to programmatically manipulate migration

parameters and use real compression algorithms. Simulation modeling Using the specialized CloudSim framework. This is ideal for academic work, as it allows you to simulate hundreds of migrations in a large data center without a real server park. To obtain reliable results, it is proposed to simulate three scenarios.

1. "Standard" scenario - basic migration without using additional optimization levels.

2. "Data-Optimized" scenario – migration with data processing layer enabled (compression + deduplication).

3. "Full Architecture" scenario – operation of all three layers, including dynamic traffic management.

The main output of the section should be graphs demonstrating the superiority of your architecture (Table 1, Fig. 3).

Table 1 – **Scenario comparison**

Parameter	Standard	Optimized	Effect
Downtime (ms)	150-300	40-60	~4 times reduction
Traffic (GB)	8.5	3.2	60% channel savings
Total Time (s)	120	45	2.5 times process acceleration

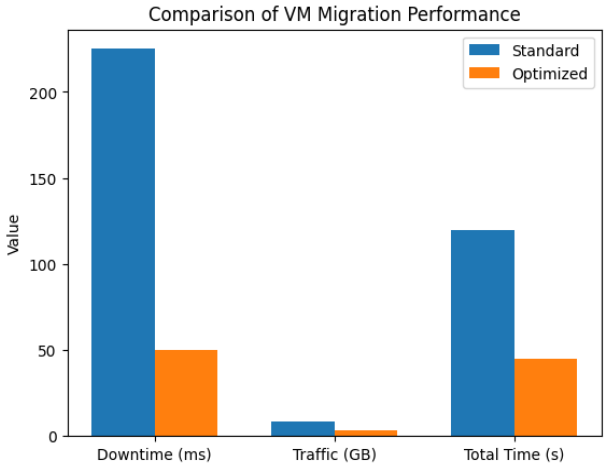

Fig. 3. Combined graph of comparison of architectures

Fig. 3 shows a comparison of the main parameters of the virtual machine migration process for two scenarios Standard and Optimized:

– standard – classic implementation of live migration without additional optimization mechanisms;

– optimized – proposed multi-tier architecture, which includes compression, deduplication and network transport optimization mechanisms.

The graph presents three main indicators.

VM downtime. This parameter characterizes the period during which the virtual machine is unavailable to users during the final migration phase (Stop-and-Copy). According to the results obtained, in the standard approach, downtime is approximately 225 ms, and in the proposed architecture it is reduced to 50 ms. Thus, the use of multi-tier optimization allows you to reduce downtime by approximately 4 times, which is critically important for highly available services.

Network Traffic. This indicator characterizes the total amount of data transmitted over the network during migration. Experimental results show that in a standard scenario, approximately 8.5 GB of data is transmitted, and when using an optimized architecture, the amount of transmitted data is reduced to 3.2 GB. Therefore, the use of compression and deduplication mechanisms allows you to reduce network traffic by approximately 60%, which significantly reduces the load on the data center network infrastructure.

Total Migration Time. This indicator reflects the total time required to complete the live migration process of a virtual machine. The results show that for the standard implementation it is approximately 120 seconds, and for the proposed architecture it is 45 seconds. Thus, the use of multi-level optimization allows you to speed up the migration process by approximately 2.5 times, which has a positive effect on the efficiency of load balancing in the data center.

Fig. 4 shows the dependence of the virtual machine downtime (Downtime) on the rate of modification of memory pages (Dirty Page Rate, *D*). This parameter characterizes the intensity of changes in the virtual machine memory during the execution of application programs. The graph presents two curves: Standard Migration (classical migration algorithm) and Optimized Architecture (proposed multi-level optimization system).

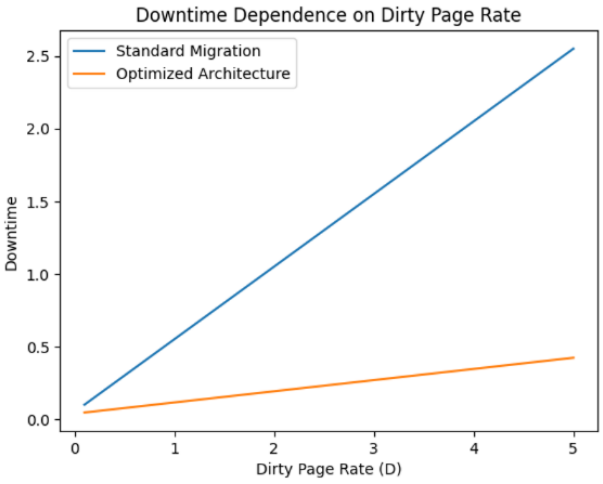

Fig. 4. Graph of Downtime vs. Dirty Page Rate

Analysis of the obtained results.

1, *For the standard architecture*. When the rate of memory modification increases, an almost linear increase in downtime is observed. This is explained by the fact that with a high Dirty Page Rate value, the system is forced to re-transmit a significant number of changed memory pages, which increases the volume of the final transfer. As a result, this can lead to a significant increase in downtime; the risk of incomplete migration.

2. *For the proposed architecture*.

The curve of the optimized algorithm has a much smaller slope, which means a slower increase in downtime with increasing Dirty Page Rate. This is achieved due to deduplication, which eliminates duplicate data blocks, memory compression, which reduces the amount of transmitted information, more efficient use of the network

channel. Thus, even at high Dirty Page Rate values, the system maintains a stable and predictable downtime.

The results obtained confirm that the proposed multi-level architecture for optimizing virtual machine migration provides a significant increase in the efficiency of the live migration process. In particular, its use allows:

– to reduce the downtime of a virtual machine by approximately 4 times;

– to reduce the amount of network traffic by approximately 60%;

– to speed up the overall migration process by approximately 2.5 times.

Furthermore, simulation results demonstrate that the proposed approach provides better system resilience to high memory modification rates, which is an important characteristic for cloud infrastructures and data centers.

Conclusions

In this study, an algorithm for dynamic management of the migration layer of virtual machines was developed and analyzed, which implements a multi-level approach to optimizing the process of transferring VM states. The proposed algorithm flowchart describes the sequence of system actions during live migration and takes into account key parameters of the functioning of the virtualized environment, in particular, the speed of changing memory pages, network bandwidth and processor load. The developed algorithm is based on constant monitoring of system metrics and provides for adaptive adjustment of migration parameters. In case the rate of generation of changed data exceeds the bandwidth of the network channel, optimization mechanisms are activated at the data processing level, in particular, deduplication and compression. In addition, the algorithm takes into account the current processor load and, if necessary, changes the compression intensity, which allows avoiding excessive load on the host's computing resources.

Additionally, the algorithm provides for transport layer management by configuring traffic priorities and using multiple TCP streams, which ensures more efficient use of the network channel.

Thanks to cyclic parameter control at each iteration of memory copying, the system determines the optimal moment for transitioning to the final Stop-and-Copy phase, which allows minimizing the downtime of the virtual machine. To verify the effectiveness of the proposed approach, two options for implementing the test environment were considered: using a real virtualization environment based on KVM/QEMU and Proxmox, as well as simulation modeling in the CloudSim environment. The latter approach is particularly appropriate for scientific research, as it allows modeling a large number of migration scenarios in a large-scale data center. As part of the experimental study, three system operation scenarios were simulated: basic migration without optimization, migration with data optimization, and full implementation of the proposed multi-tier architecture.

The results of the comparative analysis showed significant advantages of the proposed approach. In particular, the downtime of the virtual machine was reduced by about four times, the volume of transmitted network traffic was reduced by about 60%, and the total migration execution time was reduced by more than two times. The results obtained confirm the effectiveness of using a multi-tier architecture for managing virtual machine migration.

The proposed algorithm increases the system's resistance to high loads, ensures the rational use of computing and network resources, and allows significantly reducing the downtime of services during migration. This makes the proposed approach promising for use in modern cloud infrastructures and data centers.

Conflicts of interest

The authors declare that they have no conflicts of interest in relation to the current study, including financial, personal, authorship, or any other, that could affect the study, as well as the results reported in this paper.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

References

1. Mao, C. (2023), “Design of Computer Storage System Based on Cloud Computing”, *Lecture Notes in Electrical Engineering*, 1037 LNEE, pp. 651–659, doi: https://doi.org/10.1007/978-981-99-1983-3_59
2. Kuchuk, H., Mozhaiev, O., Kuchuk, N., Tiulieniev, S., Mozhaiev, M., Gnusov, Y., Tsuranov, M., Bykova, T., Klivets, S., and Kuleshov, A. (2024), “Devising a method for the virtual clustering of the Internet of Things edge environment”, *Eastern-European Journal of Enterprise Technologies*, vol. 1, no. 9 (127), pp. 60–71, doi: https://doi.org/10.15587/1729-4061.2024.298431
3. Li, Z. and Xiong, J. (2024), “Reactive Power Optimization in Distribution Networks of New Power Systems Based on Multi-Objective Particle Swarm Optimization”, *Energies*, vol. 17(10), 2316, doi: https://doi.org/10.3390/en17102316
4. Niu, Y.-F., Yan, Y.-F. and Xu, X.-Z. (2025), “A new MC-based method for the resource-constrained multi-distribution multi-state flow network reliability optimization problem”, RESS, 265, 111499, doi: https://doi.org/10.1016/j.ress.2025.111499
5. Kuchuk, N., Zakovorotnyi, O., Radchenko, V., Andrusenko, Y. and Lysytsia, D. (2025), “Load balancing of a multiprocessor computer system using the method particle swarm optimization”, *Advanced Information Systems*, vol. 9, no. 4, pp. 82–88, doi: https://doi.org/10.20998/2522-9052.2025.4.11
6. Sawalkar, V. and More, N. (20260, “Virtual Machine Migration Optimization in Cloud Data Centers: A Comprehensive Review”, L*ecture Notes in Networks and Systems*, 1647 LNNS, pp. 146–161, doi: https://doi.org/10.1007/978-3-032-06668-8_15
7. Shi, Q. and Zhao, F. (2024), “Research on Computer Cloud Intelligent System Based on Intelligent Virtualization Technology”, *2024 IEEE 3rd Int. Conf. on Eebda 2024*”, pp. 1245–1250, doi: https://doi.org/10.1109/EEBDA60612.2024.10485750
8. Kuchuk, G., Nechausov, S. and Kharchenko, V. (2015), “Two-stage optimization of resource allocation for hybrid cloud data store”, *International Conference on Information and Digital Technologies*, *Zilina*, pp. 266–271, DOI: http://dx.doi.org/10.1109/DT.2015.7222982

9. Rezanov, B., and Kuchuk, H. (2023), "Model of elemental data flow distribution in the Internet of Things supporting Fog platform", *Innovative Technologies and Scientific Solutions for Industries*, no. 3(25), pp. 88–97, doi: https://doi.org/10.30837/ITSSI.2023.25.088
10. Kuchuk, H., Kovalenko, A., Ibrahim, B.F. and Ruban, I. (2019), "Adaptive compression method for video information", International Journal of Advanced Trends in Computer Science and Engineering, vol. 8(1), pp. 66-69, doi: http://dx.doi.org/10.30534/ijatcse/2019/1181.22019
11. Kuchuk, H., Kalinin, Y., Dotsenko, N., Chumachenko, I. and Pakhomov, Y. (2024), "Decomposition of integrated high-density IoT data flow", *Advanced Information Systems*, vol. 8, no. 3, pp. 77–84, doi: https://doi.org/10.20998/2522-9052.2024.3.09
12. Zhang, K., Lyu, Y., Zheng, D., Chen, Y. and Xu. J. (2025), "Adaptive Virtual Machine Consolidation Based on Autoformer and Enhanced Double Q-Network for Energy-Efficient Cloud Data Center", *International Journal of Advanced Computer Science and Applications* (IJACSA), vol. 16(10), pp. 104–119, doi: http://dx.doi.org/10.14569/IJACSA.2025.0161011
13. Archana and Kumar, N. (2025), "A Modified Bat Mechanism for Virtual Machine Migration in a Cloud Environment", *SN Computer Science*, vol. 6, article number 74, doi: https://doi.org/10.1007/s42979-024-03627-1
14. Ferreto, T. C., Netto, M. A. S., Calheiros, R. N., and De Rose, C. A. F. (2011), "Server consolidation with migration control for virtualized data centers", *Future Generation Computer Systems*, vol. 27, issue 8, pp. 1027–1034, doi: https://doi.org/10.1016/j.future.2011.04.016
15. Wood, T., Shenoy, P., Venkataramani, A. and Yousif, M. (2009), "Sandpiper: Black-box and gray-box resource management for virtual machines", Computer Networks, vol. 53(17), pp. 2923–2938, doi: https://doi.org/10.1016/j.comnet.2009.04.
16. Meng, X., Pappas, V. and Zhang, L. (2010), "Improving the scalability of data center networks with traffic-aware virtual machine placement", *2010 Proc. IEEE INFOCOM*, doi: https://doi.org/10.1109/INFCOM.2010.5461930
17. Ma, D., Cao, X., Hu, J., Xia, T., Zhou, Y., Liu, K., Zhu, L.,. Su, L. and Gao, F. (2026), "Topology-aware virtual machine placement for improving cloud servers resource utilization", *Future Generation Computer Systems*, vol. 179, article number 108361, doi: https://doi.org/10.1016/j.future.2025.108361
18. u, L., Zhu, X., Griffith, R., Shah, P., Smirni, E. (2014), "Application-driven dynamic vertical scaling of virtual machines in resource pools", *2014 IEEE Network Operations and Management Symp.*, doi: https://doi.org/10.1109/NOMS.2014.6838238
19. Shankar, S. and Anbarasan, M. (2025), "An Intelligent Approach for Cloud Infrastructure With Improved Multi-Objective Graywolf Optimization and Resource Allocation for Dynamic Virtual Machine Placement", *Transactions on Emerging Telecommunications Technologies*, vol. 36(6), e70172, doi: https://doi.org/10.1002/ett.70172
20. Rajammal, K. and Chinnadurai, M. (2025), "Dynamic load balancing in cloud computing using predictive graph networks and adaptive neural scheduling", *Scientific Report*s, vol. 15(1), 22181, doi: https://doi.org/10.1038/s41598-025-97494-2
21. Khyzhniak, A. V. and Kazymyr, V. V. (2025), "Analysis of Methods for Supporting Personalization in IT Education", *Herald of Advanced Information Technology*, vol. 8, no.3, pp. 366–381, doi: https://doi.org/10.15276/hait.08.2025.24
22. Kuchuk, G.A., Akimova, Yu.A. and Klimenko, L.A. (2000), "Method of optimal allocation of relational tables", *Engineering Simulation*, vol. 17(5), pp. 681–689, available at: https://www.scopus.com/record/display.uri?eid=2-s2.0-0034512103&origin=resultslist
23. Lee, B.M. (2025), "Efficient Resource Management for Massive MIMO in High-Density Massive IoT Networks", *IEEE Transactions on Mobile Computing*, vol. 24 (3), pp. 1963–1980, doi: https://doi.org/10.1109/TMC.2024.3486712
24. Grybniak, S. S., Leonchyk, Y. Y., Mazurok, I. Y., Nashyvan, O. S., Shanin, R. V. and Vorokhta A. Y. (2025), "Virtually Unlimited Sharding for Scalable Distributed Ledgers", *Herald of Advanced Information Technology*, vol. 8, no. 1, pp. 67–86, doi: https://doi.org/10.15276/hait.08.2025.5
25. Kuchuk, N., Mozhaiev, O., Mozhaiev, M. and Kuchuk, H. (2017), "Method for calculating of R-learning traffic peakedness", *2017 4th International Scientific-Practical Conference Problems of Infocommunications Science and Technology*, PIC S and T 2017 – Proceedings, pp. 359–362, doi: https://doi.org/10.1109/INFOCOMMST.2017.8246416
26. Humeniuk, A. O. (2025), "Development and Optimization of Distributed High-Performance Systems With Real-Time Data Consistency", *Herald of Advanced Information Technology*, vol. 8, no. 3, pp. 326–340, doi: https://doi.org/10.15276/hait.08.2025.21
27. Kuchuk, G., Kovalenko, A., Kharchenko, V. and Shamraev, A. (2017), "Resource-oriented approaches to implementation of traffic control technologies in safety-critical I&C systems", *Studies in Systems, Decision and Control*, vol. 105, pp. 313–337, doi: https://doi.org/10.1007/978-3-319-55595-9_15
28. Taneja, M. and Davy, A. (2017), "Resource aware placement of IoT application modules in fog-cloud computing paradigm", *Proc. 2017 IFIP/IEEE Symposium on Integrated Network and Service Management*, INSM, pp. 1222–1228, doi: https://doi.org/10.23919/INM.2017.7987464
29. Petrovska, I., Kuchuk, H. and Mozhaiev, M. (2022), "Features of the distribution of computing resources in cloud systems", *2022 IEEE 4th KhPI Week on Advanced Technology*, KhPI Week 2022 – Conference Proceedings, 03-07 October 2022, Code 183771, doi: https://doi.org/10.1109/KhPIWeek57572.2022.9916459
30. Kortas, N. and Youssef, H. (2025), "A Bayesian neural network study for virtual machine migration within cloud environment", *Journal of Supercomputing*, vol. 81(16), 1505, doi: https://doi.org/10.1007/s11227-025-07974-5
31. Semenov, S., Mozhaiev, O., Kuchuk, N., Mozhaiev, M., Tiulieniev, S., Gnusov, Yu., Yevstrat, D.,Chyrva, Y., Kuchuk, H. (2022), "Devising a procedure for defining the general criteria of abnormal behavior of a computer system based on the improved criterion of uniformity of input data samples", *Eastern-European Journal of Enterprise Technologies*, vol. 6(4-120), pp. 40–49, doi: https://doi.org/10.15587/1729-4061.2022.269128
32. Liu, C., Ma, L., Zhang, M. and Long, H. (2025), "Optimizing cloud resource management with an IoT-enabled optimized virtual machine migration scheme for improved efficiency", Journal of Network and Computer Applications, 237, 104137, doi: https://doi.org/10.1016/j.jnca.2025.104137
33. Kuchuk, G., Kharchenko, V., Kovalenko, A. and Ruchkov, E. (2016), "Approaches to selection of combinatorial algorithm for optimization in network traffic control of safety-critical systems", *Proceedings of 2016 IEEE East-West Design and Test Symposium*, EWDTS 2016, 7807655, doi: https://doi.org/10.1109/EWDTS.2016.7807655
34. Cocos, H.-N. (2024), "Offline-first strategies - a novel concept for the migration of workloads using virtual machines omiting limitations of traditional service deployment concepts", *Proc. of the Int. Conf. on Applied Computing and Www Internet*,

pp. 158–166, available at: https://www.henrycocos.de/Veroeffentlichung/Praesentation_Paper_Applied_Computing_17_2024.pdf
35. Li, JL. and Li, SW. (2025), “Performance Implications of SEV Virtual Machine Live Migration”, *Lecture Notes in Computer Science*, vol 15385. Springer, Cham, doi: https://doi.org/10.1007/978-3-031-90200-0_11
36. Ogura, N., Duolikun, D., Enokido, T., Watanabe, R. and Takizawa, M. (2019), “A Virtual Machine Migration for Storage Processes”, Lecture Notes on Data Engineering and Communications Technologies, vol 25. Springer, Cham, doi: https://doi.org/10.1007/978-3-030-02613-4_67
37. Thorpe, J., Swiler, L.P., Hanson, S., Cruz, G., Tarman, T. Rollins, T. and Debusschere, B.J. (2022), “Verification of Cyber Emulation Experiments Through Virtual Machine and Host Metrics”, *ACM International Conference Proc. Series*, pp. 71–80, doi: https://doi.org/10.1145/3546096.3546115

Received (Надійшла) 12.12.2025
Accepted for publication (Прийнята до друку) 25.03.2026

ВІДОМОСТІ ПРО АВТОРІВ/ ABOUT THE AUTHORS

Кучук Ніна Георгіївна – доктор технічних наук, професор, професорка кафедри комп’ютерної інженерії та програмування, Національний технічний університет “Харківський політехнічний інститут”, Харків, Україна;
Nina Kuchuk – Doctor of Technical Sciences, Professor, Professor of Computer Engineering and Programming Department, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine;
e-mail: nina_kuchuk@ukr.net; ORCID Author ID: http://orcid.org/0000-0002-0784-1465;
Scopus Author ID: https://www.scopus.com/authid/detail.uri?authorId=57196006131.

Шишацький Андрій Володимирович – доктор технічних наук, старший дослідник, Харківський національний автомобільно-дорожній університет, Харків, Україна;
Andrii Shyshatskyi – Doctor of Technical Sciences, Senior Researcher, Kharkiv National Automobile and Highway University, Kharkiv, Ukraine;
e-mail: ierikon13@gmail.com; ORCID Author ID: https://orcid.org/0000-0001-6731-6390;
Scopus Author ID: https://www.scopus.com/authid/detail.uri?authorId=57201773300.

Радченко В'ячеслав Олексійович – старший викладач кафедри електронних обчислювальних машин, Харківський національний університет радіоелектроніки, Харків, Україна;
Viacheslav Radchenko – Senior Lecturer of the Department of Electronic Computers, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine;
e-mail: viacheslav.radchenko@nure.ua; ORCID Author ID: https://orcid.org/0000-0001-5782-1932;
Scopus Author ID: https://www.scopus.com/authid/detail.uri?authorId=57189376280.

Андрусенко Юлія Олександрівна – асистент кафедри електронних обчислювальних машин, Харківський національний університет радіоелектроніки, Харків, Україна;
Yuliia Andrusenko – Assistant Lecturer of the Department of Electronic Computers, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine;
e-mail: yuliia.andrusenko@nure.ua; ORCID Author ID: https://orcid.org/0000-0001-7844-2042.

Клівець Сергій Іванович – кандидат технічних наук, науковий співробітник Наукового центру, Харківський національний університет Повітряних Сил імені Івана Кожедуба, Харків, Україна;
Sergii Klivets – Candidate of Technical Sciences, Research Scientist at the Scientific Center, Ivan Kozhedub Kharkiv National Air Force University, Kharkiv, Ukraine;
e-mail: 1546.hnups@gmail.com; ORCID Author ID: https://orcid.org/0000-0002-8109-0639;
Scopus Author ID: https://www.scopus.com/authid/detail.uri?authorId=58983477300.

Проектування багаторівневої структури для оптимізації процесів міграції віртуальних машин

Н. Г. Кучук, А. В. Шишацький, В. О. Радченко, Ю. О. Андрусенко, С. І. Клівець

Анотація. У роботі розглядається задача проектування багаторівневої структури керування для оптимізації процесів міграції віртуальних машин (ВМ) у віртуалізованих центрах обробки даних. **Актуальність дослідження** зумовлена зростанням обсягів обчислювальних навантажень, гетерогенністю ресурсів та необхідністю забезпечення високої продуктивності й енергоефективності інфраструктури при збереженні якості обслуговування (QoS). Неефективна міграція ВМ може призводити до перевантаження вузлів, збільшення затримок та додаткових витрат ресурсів. **Метою роботи** є розроблення багаторівневої моделі керування міграцією ВМ, яка забезпечує адаптивний розподіл ресурсів, зменшення часу простою та мінімізацію витрат на переміщення віртуальних машин. **Об’єктом дослідження** є процеси функціонування віртуалізованого центру обробки даних, а предметом — методи та моделі оптимізації міграції ВМ у багаторівневій архітектурі керування. Запропонована структура передбачає стратегічний, тактичний та операційний рівні управління, що дозволяє поєднати довгострокове планування ресурсів із оперативним реагуванням на зміни навантаження. У межах роботи враховано критерії балансування навантаження, енергоефективності, мінімізації мережевого трафіку та забезпечення SLA. **Результати дослідження** можуть бути використані при проектуванні хмарних та грід-інфраструктур для підвищення ефективності використання обчислювальних ресурсів і забезпечення стабільної роботи сервісів в умовах динамічних навантажень. **Напрямами подальших досліджень** є впровадження інтелектуальних алгоритмів прийняття рішень та застосування імітаційного моделювання для оцінки ефективності запропонованої структури.

Ключові слова: віртуальна машина; багаторівнева архітектура; оптимізація ресурсів; віртуалізований центр обробки даних; балансування навантаження; хмарні обчислення; гетерогенне середовище; енергоефективність; якість обслуговування; адаптивне управління.