UDC 004. 05 **doi: 10.26906/SUNZ.2025.3.167-172**

Serhii Pyrozhenko, Viacheslav Radchenko

Kharkiv National University of Radio Electronics, Kharkiv, Ukraine

METHODS AND ALGORITHMS FOR VIRTUAL MACHINE MIGRATION

Abstract. The article discusses various aspects of virtual machine migration. The process of moving them between different hosts, data storages, or even between cloud environments. It can be performed in two modes: cold migration, when the virtual machine is previously turned off, and live (dynamic) migration, which occurs without stopping its operation. Thanks to live migration, it is possible to transfer active virtual machines, for example, between servers within a cluster, without interrupting the provision of services. To organize such a process, special administration tools are used, in particular Hyper-V Manager or System Center Virtual Machine Manager from Microsoft Learn.

Keywords: cold migration, virtual machine, host, data storage, cloud environment, dynamic migration.

Introduction

In the context of the rapid development of information technologies and the growth of computing workloads, virtualization technologies are gaining special importance, which ensure the rational use of hardware resources and increase the flexibility of infrastructure management [1, 2]. One of the key mechanisms of virtualization environments is the migration of virtual machines (VM), which is the process of moving VM between different physical hosts, data storage systems or cloud platforms in order to ensure the continuity of computing processes, load balancing and increase the fault tolerance of systems [3]. In the scientific and technical literature, two main approaches to migration are distinguished: cold, which involves stopping the operation of the virtual machine before transferring, and live (dynamic), which allows moving without interrupting service [4, 5]. The latter approach is of particular relevance in the context of critical services, where even minor delays can lead to significant losses.

The choice of optimal migration methods and algorithms is determined by the need to minimize transfer time, reduce latency, preserve data integrity, and ensure high performance of the computing environment [6, 7]. That is why conducting a systematic analysis of existing solutions in this area is an urgent scientific and practical task aimed at improving approaches to building and operating modern virtualized and cloud infrastructures.

Main part

The concept of cloud computing is a type of distributed data processing and storage aimed at promptly providing consumers with secure, high-quality, scalable and automatically configured IT services on demand based on the pay-as-you-go model [8]. The National Institute of Standards and Technology (NIST) defines cloud computing as a combination of three service models: SaaS (Software as a Service) - software as a service; PaaS (Platform as a Service) - platform as a service; IaaS (Infrastructure as a Service) - infrastructure as a service [9, 10]. The implementation of these types of models is carried out within the framework of the data center, which is owned or leased by the service provider. Currently, as part of improving the processes of organizing and functioning of the data center, the technology of their virtualization is used as the main technology for managing its computing and communication resources [11]. This type of data center is called virtualized or software-defined data center (SDDC) [12]. SDDC is a software add-on within a physical data center, which is a set of virtualized VRCs, such as servers, data storage systems (DSS), network components (telecommunication infrastructure), as well as a management system for them, which are deployed and managed as a single unit. The VCDC has the same functionality as a physical data center, while having additional benefits provided by virtualization technology. In fact, the VCDC, using virtualization technology, forms an abstract representation of the physical data center. In this case, the entire infrastructure of the physical data center, namely: processors (processor cores), RAM, data storage systems, network equipment, automation and information security tools are represented in the VCDC as a set (pool) of VRCs. In general, the concept of a virtual data center is formed on the basis of a generalized approach to virtualization of computing resources and already well-formed concepts of software-defined networks (SDN) and software-defined data storage systems (SDS). As in the case of SDN and SDS, such an approach allows us to present a virtualized data center in the form of IT as a Service (ITaaS) platforms.

Fig. 1 discusses the detection of underloaded or overloaded physical machines (PM) (steps 5 and 6) where one or a subset of VM must migrate between PM.

One of the goals of this migration is the need to eliminate SLA violations. The types of migration are presented in a generalized form in Fig. 2.

It is obvious that, regardless of the type of migration, this process boils down to the transfer of the state of virtualized computing and communication resources of the VM between the original and target PM. The state of virtualized computing and communication resources includes:

- state of the processor or processor cores (VCPU State);
- content of the operational memory (Memory content);
- the contents of storage of virtual disks (Storage content).

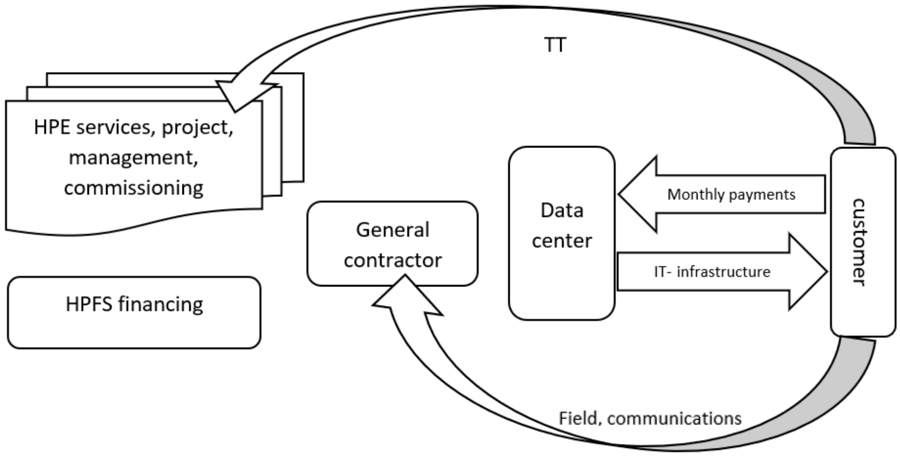

Fig. 1. Generalized representation of a virtualized data center as a FaaS service Data center as a FaaS service

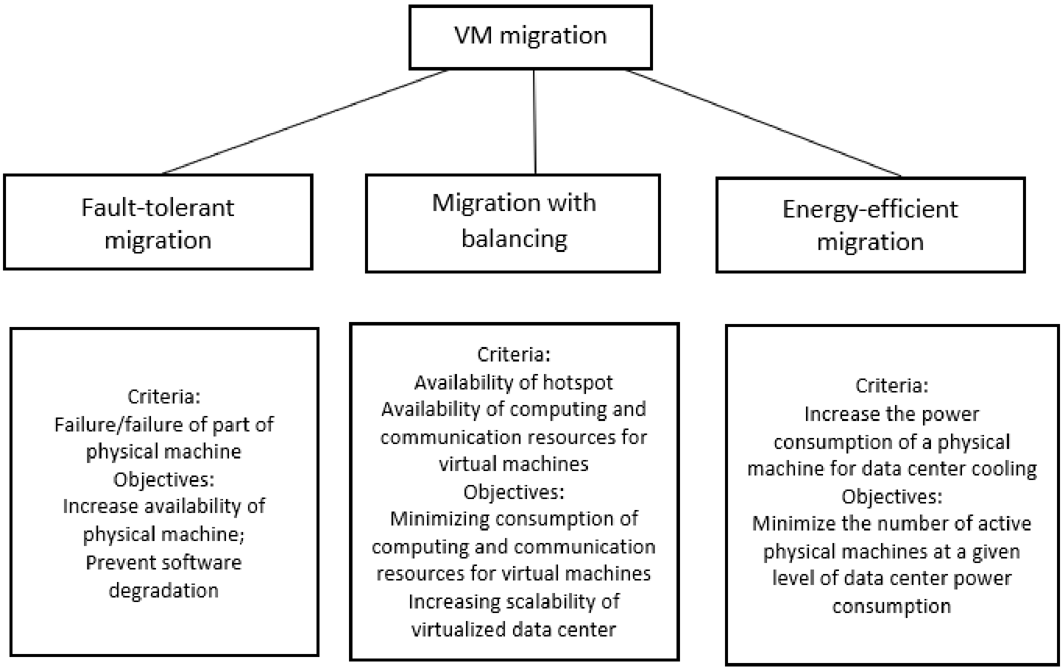

Fig. 2. Types of VM migration, their criteria and purpose

It is obvious that, regardless of the type of migration, this process boils down to the transfer of the state of virtualized computing and communication resources of the VM between the original and target PM. The state of virtualized computing and communication resources includes: state of the processor or processor cores (VCPU State); content of the operational memory (Memory content); the contents of storage of virtual disks (Storage content).

Modern data centers support a single storage of virtual disks, accessible through the telecommunications system of the data center in read-write mode for the whole set of VMs located on different PMs. The state VCPU State actually determines the context of the registers of the processor (processor core), which is not critical from the point of view of transfer between the source and target CPU. The most critical from the point of view of the operational efficiency of the VM relocation process is the content of the pages (pages) of its operational memory (Memory content). In general, the VM migration process is presented in Fig. 3.

As part of the process of VM migration presented in Fig. 3, it is classified into:

- non-live migration (VM Non-Live Migration);
- live (or dynamic) migration (VM Live Migration).

Non-live migration is performed offline. This means that VMs must be suspended on the source VM and the state of their virtualized computing and communication resources must be copied to the target VM, after which the VMs can be resumed. To transfer the state of Memory Content, non-live migration uses the stop-and-copy method, in which the execution of the VM must be suspended in order to copy its pages of the virtualized RAM to the target PM (stage 1 in Figure 3), and the VM can be resumed only after performing this copy operation. VM Live Migration refers to the transfer of the state of its virtualized computing and communication resources between the hardware computing and communication resources of a set of VMs located either within one data center node or on territorially distributed data center nodes.

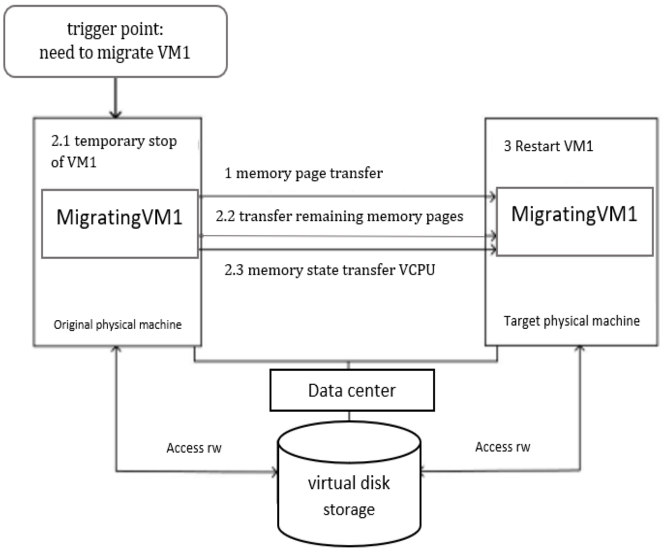

Fig. 3. General representation of the VM migration process

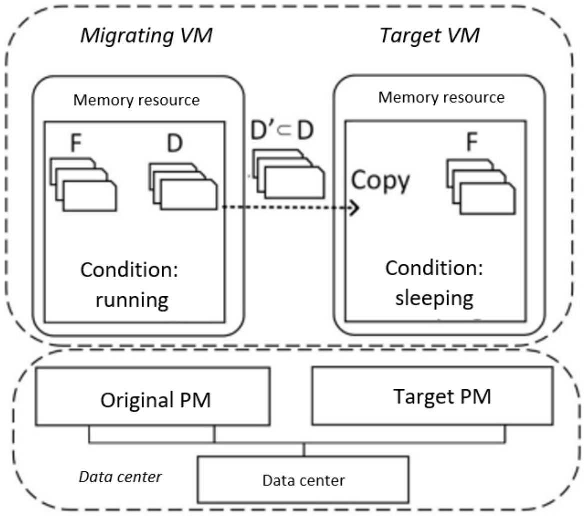

Fig. 4. General presentation of the technology of VM live migration

Existing VM Live Migration technologies mainly differ in the features of the transfer of the Memory Content state, which are based on the concept of memory organization of modern operating systems (OS) of server platforms that support the software and hardware technology of page memory allocation (Page Memory). According to this concept, the Memory content resource is represented by two sets of memory pages: Many free F pages (from Free); many busy ("dirty") pages D (from Dirty).

A busy ("dirty") memory page is a page in which a part of the C software context, the software being executed, or its individual modules is written: software code (modules); the code of the libraries necessary for the functioning of the software code; software configuration data (modules, libraries); data processed by software (modules, libraries).

It is obvious that the power of the set D is a variable value, the value of which depends on: the current value of the power of the set F at the moment of loading the software (modules, libraries) into the PM memory; values of the C software context at the time of loading the software (modules, libraries) into memory; values of the dynamically loaded context tn of software C at moments of time tn. In a generalized form, the principle of VM live migration is presented in Fig. 4.

It can be seen from Fig. 4 that two types of VMs are used in the process of implementing VM Live Migration:

1. Migratable (initial) VM, the state of which requires transfer (migration).

2. The target VM is located on the target VM, which, when started, will implement the functions of the migrated VM.

It is obvious that the state of the target VM at the time of startup should not be active (sleeping). Otherwise, the VM hypervisor will not get access to the set F ee memory pages. In turn, the state of the migrated VM must be active (running), since the current context of the C software running on it must be transferred during the Live Migration process, both at the start of the VM Live Migration process and during its development.

The basic operation of VM Live Migration is the copying (Copy) of a set of pages D. At the same time, the VM hypervisor, having determined the target VM, iteratively selects a subset of pages $D' \subset D$ of the memory resource of the migrated VM, available for performing the Copy operation, at each moment of the VM Live Migration process. The elements of the subset D' include pages that, at the time of their selection, are not accessed for the execution of the write operation (Write) by the software (modules, libraries) operating on the migrated VM. It is obvious that the duration of the VM Live Migration process depends significantly on the activity of the software functioning (modules, libraries) specified above and can take a significant amount of time and can lead to an increase in the target quality indicator of the service provided to the user (software response time).

For example, studies of VM Live Migration of specific application services [6] showed that VM downtime varies depending on the implemented software workload template (software workload) from 60 milliseconds (gaming software) to up to 3 seconds in the case of high-performance computing software. Numerous studies have shown that the response time of software running on migrated VMs increases significantly during the entire Live Migration period. As part of the optimization of the VM Live Migration process in terms of the response time of the software, algorithms for its implementation were developed, using various approaches to the implementation of the Copy operation. Currently, there are implementations of VM Live Migration based on two approaches:

1. Pre-copy – based on a two-stage scheme that implements a preliminary iterative sending of the contents of the VM RAM pages to the source VM (stage 1), and its stopping, and copying the state of its other resources (processor and disk storage) to the target VM (stage 2);

2. Post-copy – based on the preliminary transfer of the minimum VM state (CPU state, state of registers and input/output subsystem) and subsequent iterative transfer of memory pages on demand (paging on

demand). Depending on the implementation, modern hypervisors of the first type support various of the VM Live Migration technologies considered above or their hybrid solutions.

A summary of the features of the functioning of various implementations of the pre-copy and post-copy Live Migration algorithms is presented in Fig. 5 and 6.

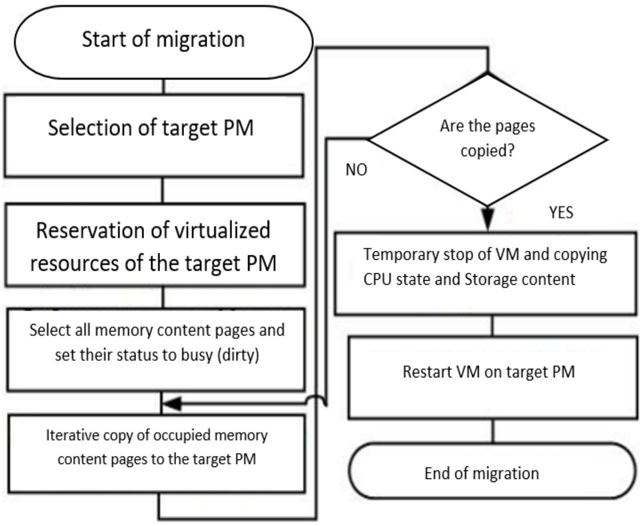

Fig. 5. Scheme of the VM Live Migration pre-copy algorithm

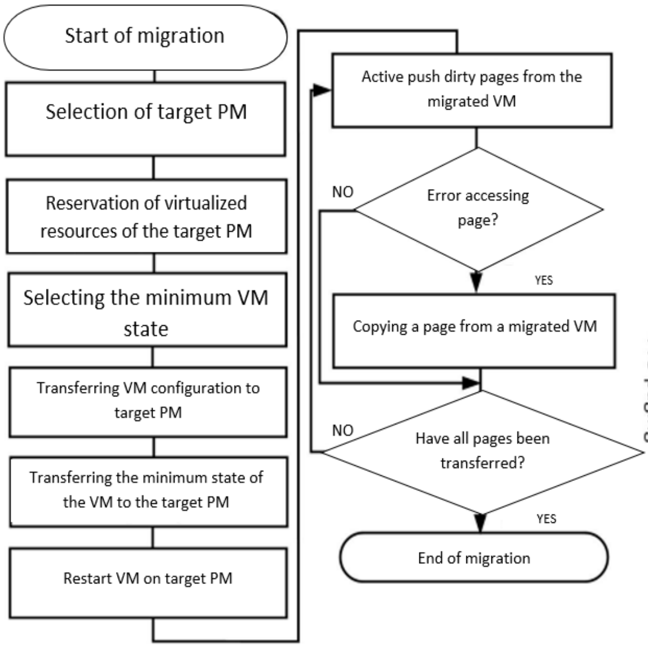

Fig. 6. Scheme of the post-copy VM Live Migration algorithm

Fig. 7 summarizes the time diagrams of the specified migration algorithms, reflecting the ratio of idle time to the total migration time.

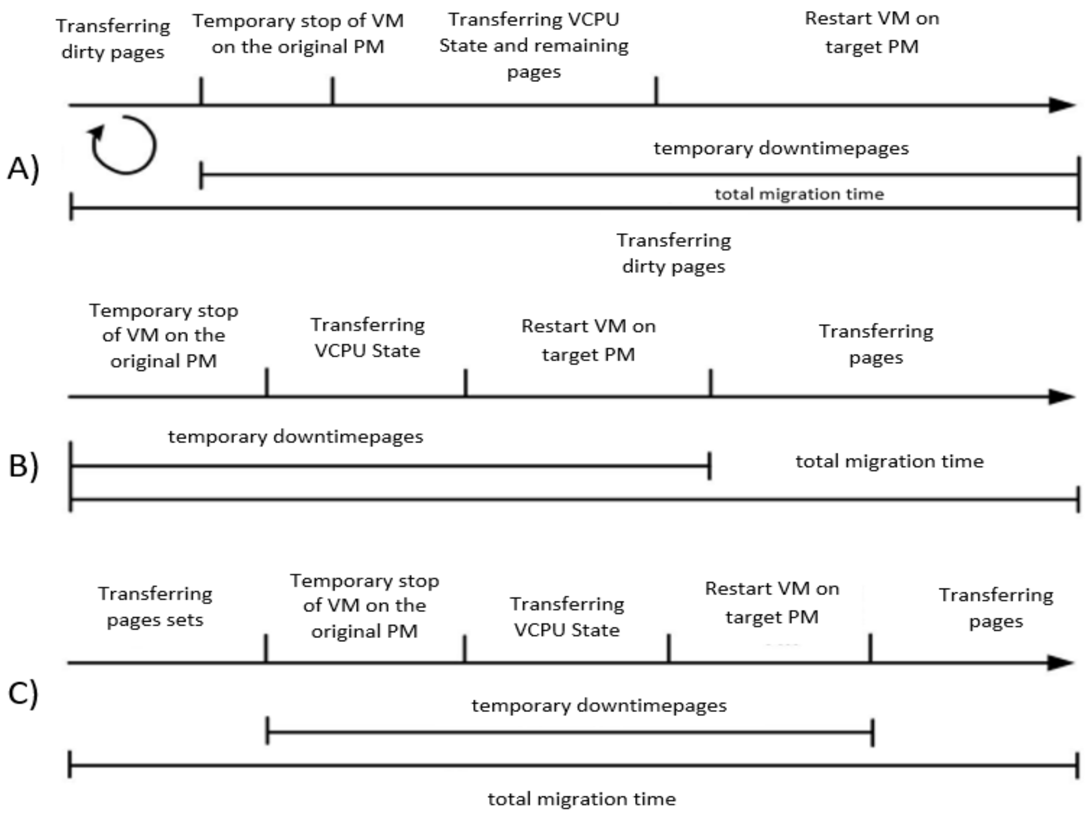

Fig. 7. Generalized timing diagrams of Live Migration algorithms: a) pre-copy, b) post-copy, c) hybrid Live Migration

In Table 1, a brief study of the features of the implementation of the specified algorithms is performed from the point of view of idle time, which leads to an increase in the response time performed on VM software. In particular, basic metrics reflecting the effectiveness of the Live Migration process implemented by pre- and post-copy algorithms are defined (Table 1).

Figure 8 shows the time diagrams of the implementation of the pre- (a) and post-copy (c) algorithms, showing the interrelationship of the metrics presented in Table 1.

Table 1 – **Metrics for evaluating the effectiveness of the live migration process of virtual machines**

The name of the metric	Description of the metric
t1 - preparation time	The time between the start of the migration and the transfer of the CPU (processor) context of the VM to the target VM, during which the migrated VM continues to execute and form new pages of set D. For the pre-copy algorithm, it includes an iterative procedure of copying the entire set of pages D. For the postcopy algorithm, this time is insignificant.
t2 - downtime	Time during which execution of the migrated VM is suspended. First of all, it includes the CPU context transfer time. For the pre-copy algorithm, this transfer time includes the transfer of the last subset of pages D ′⊂ D. For the post-copy algorithm, the time required by the VM to start on the target PM t3 - the time of resume
t3 - renewal time	The time between the resumption of VM execution on the target PM and the end of the migration as a whole. For the pre-copy algorithm, it consists of replanning the target VM and deleting the migrated VM. For the post-copy algorithm, it implements the main stages of migration (Figure 8)
Npage - the number of copied pages	The total number of copied pages of the set D, including their duplicates, according to the metrics tpod, simple t, t3. The pre-copy algorithm copies most of the pages for tpod. The postcopy algorithm copies most of the pages for t3.

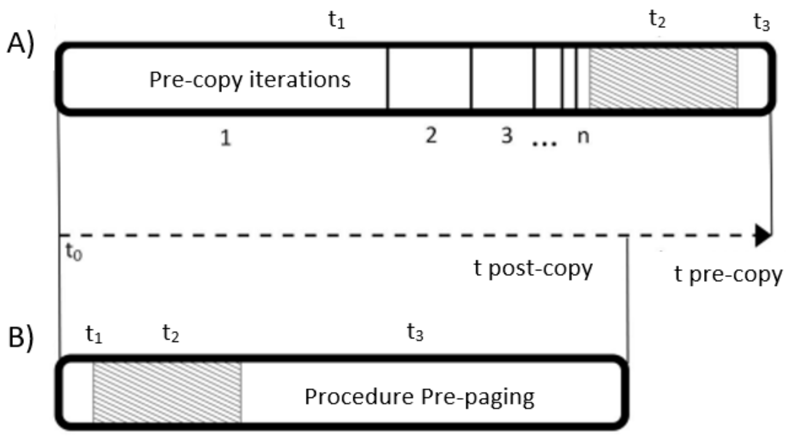

Fig. 8. Time diagrams of the implementation of live migration algorithms: a) pre-copy, b) post-copy

When performing Live Migration algorithms, Xen VMM, KVM Hypervisor, Microsoft Hyper-V, ESXi hypervisors are connected. As part of the process of functioning of the data center, the implementation of the VM Live Migration process (regardless of the algorithm of its implementation) is performed for a certain subset of VMs on the basis of a previously formed reassignment plan (VM-PM Remapping). Usually, it is presented in the form of a migration matrix (MM) - a bit map that determines the mapping of the nth VM to the kth PM.

Figures 4–8 show that, depending on the Live Migration algorithm used, the time characteristics of this process, as well as the characteristics of computing and communication resources involved in its implementation, differ significantly. They determine the overheads that are part of the data center administration process and can influence the overall efficiency of the data center operation.

It is obvious that the dynamics of the functioning of VMs in large virtualized data centers, which are generally random in nature, in combination with the automatic solution of the VM hypervisor to the task of balancing the load on multiple VMs using the VM Live Migration algorithms discussed above, over time can lead to the fragmentation of a set of VMs by a set of distributed VCRs of VMs. Usually, such a fragmentation problem arises during the operation of the formed structure of virtualized computing and communication resources, depending on the intensity and specifics of the tasks solved by the users of these VMs.

Thus, it is important to consider methods that support the VM relocation process, taking into account the influence of Live Migration algorithms on this process.

Conclusion

The analysis shows that virtual machine migration is a key mechanism of modern virtualized and cloud environments, which ensures the continuity of service provision, optimal use of computing resources and increased fault tolerance of the infrastructure. The main approaches to its implementation are cold and live migration, with the latter gaining the most popularity due to its ability to provide minimal downtime and high availability of critical applications.

Research into modern methods and algorithms has shown that the efficiency of the migration process depends on optimizing transfer time, reducing network costs, preserving data integrity and maintaining quality of service (QoS).

The development of new approaches in this area is focused on the use of intelligent control algorithms, load forecasting technologies and integration with cloud and edge computing infrastructure.

Thus, further scientific research should be aimed at improving existing algorithms and developing new migration methods, which will increase the scalability and reliability of virtualized systems, as well as ensure their effective integration into the architectures of next-generation computing networks.

References

1. Silva, D., Rafael, J. & Fonte, A., 2023. Virtualization Maturity in Creating System VM: An Updated Performance Evaluation. *International Journal of Electrical and Computer Engineering Research*, 3(2), pp. 7–17. https://doi.org/10.53375/ijecer.2023.341
2. Zhang, S., Wang, Y., Wan, X., Li, Z. & Guo, Y., 2023. Virtualization Airborne Trusted General Computing Technology. *Applied Sciences*, 13(3), 1342. https://doi.org/10.3390/app13031342
3. Altahat, M.A., Daradkeh, T. & Agarwal, A., 2025. Virtual machine scheduling and migration management across multi-cloud data centers: blockchain-based versus centralized frameworks. *Journal of Cloud Computing*, 14, Article number: 1. https://doi.org/10.1186/s13677-024-00724-7
4. Kuchuk, N., Mozhaiev, O., Semenov, S., Haichenko, A., Kuchuk, H., Tiulieniev, S., Mozhaiev, M., Brusakova, O. and Gnusov, Y. (2023), "Devising a method for balancing the load on a territorially distributed foggy environment", *Eastern-European Journal of Enterprise Technologies*, 1(4 (121), pp. 48–55. doi: https://doi.org/10.15587/1729-4061.2023.274177.
5. Chen, J., Wang, Y. and Liu, T. (2021), "A proactive resource allocation method based on adaptive prediction of resource requests in cloud computing", *J. Wireless Com Network*, 24, doi: https://doi.org/10.1186/s13638-021-01912-8.
6. Petrovska, I. and Kuchuk, H. (2022), "Static allocation method in a cloud environment with a service model IAAS", *Advanced Information Systems*, vol. *6, is.* 3, pp. 99–106, doi: https://doi.org/10.20998/2522-9052.2022.3.13.
7. Kuchuk, N., Shefer, O., Cherneva, G. and Alnaeri, F.A. (2021), "Determining the capacity of the self-healing network segment", *Advanced Information Systems*, vol. 5, no. 2, pp. 114–119, Jun. 2021, doi: https://doi.org/10.20998/2522-9052.2021.2.16.
8. Petrovska, I. and Kuchuk H. (2022), "Features of the distribution of computing resources in cloud systems", *Control, Navigation and Communication Systems*, No. 2, pp. 75-78, doi: http://dx.doi.org/10.26906/SUNZ.2022.2.075.
9. Kuchuk, G., Kovalenko, A., Komari, I.E., Svyrydov, A. and Kharchenko, V. (2019), "Improving big data centers energy efficiency: Traffic based model and method", *Studies in Systems, Decision and Control*, vol. 171, Kharchenko, V., Kondratenko, Y., Kacprzyk, J. (Eds.), Springer Nature Switzerland AG, pp. 161-183, doi: https://doi.org/10.1007/978-3-030-00253-4_8.
10. Nechausov, A., Mamusuĉ, I. and Kuchuk, N. (2017), "Synthesis of the air pollution level control system on the basis of hyperconvergent infrastructures", *Advanced Information Systems*, vol. 1, no. 2, pp. 21–26, doi: https://doi.org/10.20998/2522-9052.2017.2.04.
11. Tan, B., Ma, H. and Mei, Y. (2017), "A NSGA-II-based approach for service resource allocation in cloud", *IEEE Congress on Evolutionary Computation (CEC)*, 17013723, pp. 2574–2581, doi: https://doi.org/10.1109/CEC.2017.7969618.
12. Wang, J., Bewong, M. & Zheng, L. (2024), "SD-WAN: Hybrid Edge Cloud Network between Multi-site SDDC", *Computer Networks*, 250, Article number 110509, doi: https://doi.org/10.1016/j.comnet.2024.110509

Received (Надійшла) 16.05.2025
Accepted for publication (Прийнята до друку) 30.07.2025

Відомості про авторів / About the Authors

Пироженко Сергій Станіславович – аспірант кафедри електронних обчислювальних машин, Харківський національний університет радіоелектроніки, Харків, Україна;
Serhii Pyrozhenko – PhD candidate of Department of Electronic Computers, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine;
e-mail: serhii.pyrozhenko@nure.ua; ORCID Author ID: http://orcid.org/0009-0006-4209-2144;
Scopus Author ID: https://www.scopus.com/authid/detail.uri?authorId=59675302400&origin=resultslist.

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

Методи та алгоритми міграції віртуальних машин

С. С. Пироженко, В. О. Радченко

Анотація. У статті розглянуто різні аспекти міграції віртуальних машин, процес їх переміщення між різними хостами, сховищами даних або навіть між хмарними середовищами. Вона може здійснюватися у двох режимах: холодна міграція, коли віртуальна машина попередньо вимикається, та жива (динамічна) міграція**,** яка відбувається без зупинки її роботи. Завдяки живій міграції можливо переносити активні віртуальні машини, наприклад, між серверами всередині кластера, не перериваючи надання послуг. Для організації такого процесу застосовують спеціальні інструменти адміністрування, зокрема Hyper-V Manager **або** System Center Virtual Machine Manager від Microsoft Learn.

Ключові слова: холодна міграція, віртуальна машина, хост, сховище даних, хмарне середовище, динамічна міграція.