DOI: https://doi.org/10.26565/2304-6201-2022-55-06

УДК (UDC) 004.94

Telezhenko Denys *PhD student; V.N. Karazin Kharkiv National University, Svobody Square, 4, Kharkiv-22, Ukraine, 61022.*
e-mail: denisque75@gmail.com
https://orcid.org/0000000283778517

Tolstoluzka Olena *doctor of Engineering Sciences; Professor of Theoretical and Applied Systems Engineering Department; V.N. Karazin Kharkiv National University, Svobody Square, 4, Kharkiv-22, Ukraine, 61022.;*
e-mail: elena.tolstoluzka@karazin.ua
https://orcid.org/0000000312417906

A conceptual model for synthesizing the architecture of virtual distributed systems

Abstract. The article focuses on the development of a conceptual model for synthesizing the architecture of virtual distributed systems (VDS). It examines key aspects of VDS, including hardware, hypervisors, virtual machines, and management modules. The study highlights the methodological principles of architecture synthesis, starting from requirements analysis, architectural design, implementation and testing, and concluding with evaluation and optimization of VDS performance. Special attention is given to the role of hypervisors and virtual machines, their interaction with hardware, and resource management capabilities. This article provides valuable insights for researchers and practitioners involved in virtualization and computing systems development or optimization. **Purpose.** The purpose of this scientific article is to develop a conceptual model for the synthesis of architecture of virtual distributed systems. The focus is on key system components such as hardware, hypervisors, virtual machines, and management modules. The research aims to define methodological principles of architecture synthesis, covering requirements analysis, design, implementation, testing, evaluation and optimization of virtual distributed systems. **Research methods.** To achieve the goal, methods of analysis, synthesis, modeling and experimentation have been used. The system requirements and the standard practices in the field of virtualization and distributed systems have been analyzed. The methods of architecture design, technology selection and determination of interaction between system components have been applied. The models have been developed and the experiments have been conducted to evaluate and optimize the proposed conceptual model. **The results.** As a result of the research, a conceptual model of the synthesis of the architecture of virtual distributed systems has been developed. The key components and methodological principles have been taken into account. The study confirms the importance of a deep understanding of system requirements and the selection of appropriate technologies for successful architecture synthesis. Special attention is paid to the role of hypervisors and virtual machines in the system, as well as their interaction with hardware and resource management capabilities. **Conclusions.** This paper has significant practical and scientific value and can be useful for researchers and practitioners in the field of virtualization and development of virtual distributed systems.

Keywords: *virtual distributed systems (VDS), architecture synthesis, conceptual model, hardware, hypervisor, virtual machines, management module, requirements analysis, architecture design, implementation and testing, evaluation and optimization, virtualization, scalability, reliability, security, computing systems.*

Як цитувати: Тележенко Д.О., Толстолузька О.Г. Концептуальна модель синтезу архітектури віртуальних розподілених систем. *Вісник Харківського національного університету імені В.Н. Каразіна, сер. «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління»*. 2022. вип. 55. С.58-64. https://doi.org/10.26565/2304-6201-2022-55-06

How to quote: D. Telezhenko, and O. Tolstoluzka “A conceptual model for synthesizing the architecture of virtual distributed systems” *Bulletin of V.N. Karazin Kharkiv National University, series “Mathematical modelling. Information technology. Automated control systems*, vol. 55, pp 58-65, 2022. https://doi.org/10.26565/2304-6201-2022-55-06

1. Introduction

Virtual distributed systems are a key component of modern computing technologies. They make it possible to isolate various computational processes, thereby increasing resource utilization efficiency and ensuring security. This article proposes a conceptual model for the synthesis of VDS architecture based on a novel approach to virtualization. It is crucial to develop a conceptual model for the synthesis of architecture in virtual distributed systems that encompasses key components, their interactions, management strategies, technologies, optimization methods, and testing. Additionally, the model should be scalable and flexible, adaptable to changing conditions and new technological trends. It is expected that this work will help gain a better understanding of how to effectively design and develop virtual distributed systems, taking into account various factors that impact their performance, reliability, and efficiency.

2. Basic concepts and definitions

Virtual distributed systems are a type of computer systems that enable the allocation of resources among multiple computational processes within a virtualized environment. VDS utilize virtualization technology, which allows for the partitioning of physical resources into virtual ones, and each virtual resource can be isolated from others. [1]

3. Conceptual model

The conceptual model for synthesizing VDS architecture comprises four main components: hardware, hypervisor, virtual machines, and management module (Fig. 1). These components interact with each other to ensure the proper functioning of the entire system.

1. *The hardware components:* these components form the foundation of virtual distributed systems. They include central processing units (CPUs), random-access memory (RAM), data storage devices, network interfaces, and other resources. For virtual distributed systems, it is important to have flexible and efficient hardware that can be dynamically allocated among virtual machines according to their needs. Without reliable, scalable, and high-performance hardware, the effective operation of VDS is not possible.

2. *Hypervisor:* It is software that provides hardware virtualization [2]. The hypervisor is responsible for resource allocation between virtual machines, ensuring their isolation and independence, and controlling their execution. The hypervisor is critically important for VDS as it allows multiple independent virtual machines to run on a single physical machine. It enables the efficient utilization of hardware resources by virtualizing them and providing a layer of abstraction for the virtual machines to operate on.

3. *Virtual Machines:* These are "virtual computers" that run on a single physical server and are managed by the hypervisor. Each virtual machine has its own operating system and set of programs. In the context of this topic, virtual machines are important as they enable the execution of multiple independent tasks on a single server, optimizing resource utilization and enhancing system flexibility. Additionally, the application of parallel processing methods in executing each individual task on a virtual machine can significantly reduce the overall task execution time. Currently, in the literature, five methods of parallel information processing are described [3]:

- Combining of Independent Operators (CIO) method;
- Pipeline Processing (PP) method;
- Decompositional Processing (DP) method;
- Code-Matrix Method (CMM);
- Method of Algorithm Mixture (MAM).

The essence of the Independent Operators Fusion method lies in simultaneously starting the execution of a certain number of operators of an algorithm at discrete time moments, under the following conditions:

a) the operators in question are not connected by any informational and/or control dependencies.

- for each of these operators, there are values for the respective operands up to the considered time moment.
- In the future, we will use the following definition of the method of combining operations - it is a method of parallel processing for which:
- objects of the operational level are bit codes of numbers or parts of codes (up to a single bit), considered as indivisible wholes;
- objects of the algorithmic level are fragments of algorithms or algorithms that are considered as indivisible goals;

- operational level operators are generally accepted data processing operations/functions (when working with number codes) and Boolean algebra operations (when working with one-bit data);
- algorithmic level operators are operations on graphs or time-parallel graph-schemes;
- time relations between (at least some) operators satisfy the requirement of having at least one pair of operators $P_i \in P$ and $P_j \in P$ of algorithm P with a non-empty intersection of their activity intervals IA_i and IA_j

$$IA_i \cap IA_j \neq O \text{ for } P_i, P_j \in P, \quad (1)$$

- the execution of which begins simultaneously, that is, for which tiH tjH;
  the nature of the time relations between the activity intervals I(SDk) of the implementation of the algorithm in different sets of input data SDk∈SD is determined by the following relation:
  I(SDk) ∈I(SDl) ≠O (2)

at least for some pair of numbers k, lj of input data sets of the algorithm where ki, l ∈ 1,2,...,sd; sd = |SD|, that is, there is an intersection of the time intervals of algorithm implementations for different data sets.

The method of Pipeline Processing (PP) is a method of parallel data processing, in which the following objects are defined:
- the algorithms for performing operators – generally accepted operations/functions of known programming languages;
- the objects of the algorithmic level are fragments of algorithms or algorithms considered as an indivisible whole.

The essence of the method is to perform the following transformations for the algorithm:

a) dividing the time algorithm for performing operations/functions (at the operational level) or the time algorithm for solving a problem (at the algorithmic level) into “pipeline” fragments (F) of equal time depth TD (pipeline clock), such that each previous fragment forms input data for the adjacent next fragment, and the parameter tн(Fj) of the beginning of each subsequent fragment Fj is determined by the parameter tк(Fi) of the end of the previous fragment

b) Fi(i,j ∈ NF;NF= 0.1,...,nf 1; where nf=| NF| is the number of pipeline fragments or the depth of the pipeline);

c) sequential implementation of fragments F0, F1, ..., Fnf-1 - in the case of a unit power (sd = 1) of the SD set of different input data sets of the algorithm;

d) combining (with sd>1) execution intervals of "different types" of algorithm pipeline fragments belonging to different sets of input data;

e) numbers ρ of combined fragments Fρ corresponding to input data sets with numbers δ (δ=1.2,....,nf-1, nf, nf+1,...) satisfy (in pipeline mode) the following relation ρ + δmod (nf +1) = nf.

The pipeline processing method is characterized by the input of data sets with a TD discreteness interval and the output of algorithm execution results (for a set of SD input data sets) with a TD interval.

The essence of the method of algorithm mixture (MAM) is to create a mixture of tasks at the operational level and to use such a mixture to achieve 100% loading of the equipment and ensure potentially possible increase in the efficiency of the implementation of the analyzed set of algorithms. A mixture of algorithms is formally considered and performed as a single task. A mixture of tasks is used in cases where individually executed tasks, using all or part of the parallel processing methods discussed above, cannot ensure 100% hardware loading and achieving the potential value of performance.

Currently, only the method of combining independent operators is used in all modern processors. The method of pipeline processing at the hardware level is used in scalar and superscalar processors. The decomposition method, also at the hardware level, is used in Very Long Instruction Word (VLIW) processors with parallelism at the instruction level. The research conducted at the Faculty of Computer Sciences of the Kharkiv National University has showed that the application of a rational combination of methods of parallel information processing (multiparallel information processing) allows increasing the efficiency of computing systems of various classes. Table 1 shows a reasonable composition of different parallel processing methods for specific combinations of constraints and user requirements.

4. The management module is responsible for monitoring, managing and optimizing the operation of the VRS. This includes monitoring resource usage, managing resource allocation, ensuring system security, and responding to changes in system load or health. The control module is critically important for synthesizing the architecture of virtual distributed systems, as it ensures their stable and efficient operation. Taking into account the peculiarities of the organization of the computing process in individual nodes of the distributed system contributes to increasing the productivity of the entire virtual computing system. For example, parallel execution of successive threads of operations in scalar and superscalar processors is carried out using a hardware-implemented pipeline (or several pipelines working simultaneously for superscalar processors). In VLIW processors, the commands that are part of a long command word are executed in parallel, that is, the commands for which all input data are ready at the moment and there is a free resource for implementation.

Table 1. Requirements and limitation

Requirements/ limitation		Methods of parallel processing			
		Combining of Independent Operators method (c=1)	Pipeline Processing method (k=1)	Decompositional Processing method (d=1)	Code-Matrix Method (q=1)
Algorithm execution time t_0 (of algorithms t_ν)	Z=1	+	-	-	+
	Z>1	+	+	+	+
Input data refresh period T_{BX}	Z=1	-	+	+	+
	Z>1	-	+	+	+
Common requirements $(t_0 / t_\nu) \& T_{BX}$	Z=1	+	+	+	+
	Z>1	+	+	+	+
Requirements and restrictions $(t_0 / t_\nu) \& Q$	Z=1	+	-	-	+
	Z>1	+	+	+	+
Requirements and restrictions $T_{BX} \& Q$	Z=1	-	+	+	+
	Z>1	-	+	+	+

Requirements and restrictions $(t_0/t_{\wr})\&T_{BX}\&Q$	Z= 1	+	+	+	+
	Z> 1	+	+	+	+

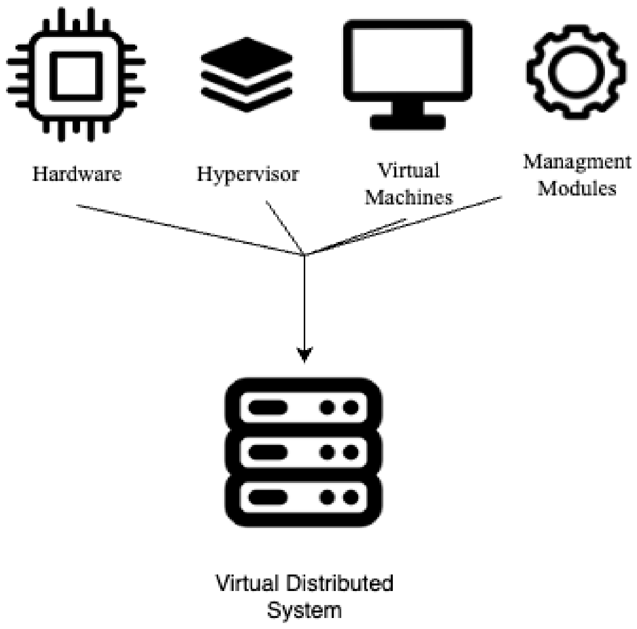

Figure 1. The scheme of the conceptual model of synthesizing the architecture of virtual distributed systems.

The conceptual model is an important part of the design process of virtual distributed systems. It defines the general structure of the system, reflecting the interactions between its various components, as well as the principles of their work. [4]

The conceptual model serves as the basis for detailed system design. It helps developers better understand the tasks the system should solve and the context in which it will be used. [5] This, in turn, allows you to determine the requirements for the system and choose the most suitable technologies for its implementation. The following aspects can be taken into account in the conceptual model of virtual distributed systems:

- Functionality: What tasks should the system solve? What services should it provide to its users?
- Interaction between components: How do the components of the system interact with each other? How do they work together to provide the functionality you need?
- Usage scenarios: In what conditions and scenarios will the system be used? What reliability, performance, scalability and security requirements should it meet?
- Data exchange and network interactions: How will data be transferred between different parts of the system? What protocols and technologies will be used for this?

A conceptual model is created at the initial stages of system development and can be updated throughout the project's life cycle to take into account new requirements or changes in the technological environment. [6]

4. Synthesis of architecture

Architecture synthesis is the process by which a conceptual model is transformed into a detailed plan or "architecture" of a virtual distributed system. This includes choosing specific technologies, designing the system structure, defining interfaces and protocols for interactions between components, and developing strategies to ensure system reliability, performance, scalability, and security. [7,8]

Architecture synthesis can be divided into several stages, including:

- *Technology selection:* Based on the system requirements and conceptual model, specific technologies are selected to implement virtual machines, hypervisors, management systems, and other system components.
- *Structure design:* at this stage, the detailed structure of the system is determined, including the network structure, placement of virtual machines, protocols for interaction between components, etc. [9]
- *Development of QoS strategies:* based on requirements for reliability, performance, scalability and security, strategies are developed to ensure these characteristics, such as backup strategies, load balancing, data encryption, etc. [8, 9]
- *Testing and Validation:* After the architecture is designed, it is tested and validated to ensure that it meets the system requirements.
- *Optimization:* Based on the results of testing and validation, necessary adjustments are made and various aspects of the architecture are optimized.

5. Conclusion

The conceptual model of synthesizing the virtual distributed system architecture plays a key role in creating efficient and reliable virtual distributed systems. The components specified in this model - a virtual machine, a hypervisor, a management system and network interactions - summarize important aspects that must be taken into account when designing such systems [10]. In the process of architecture synthesis, it is important to take into account the choice of technologies, structure design, development of service quality assurance strategies, testing and validation, as well as optimization to create an effective system [11].

This article provides important information for understanding and designing the architecture of virtual distributed systems, taking into account various aspects related to technology, design, quality of service strategies, and testing procedures.

Last but not least, the article emphasizes that the development of virtual distributed systems requires constant revision and updating of the architecture to adapt to new requirements and technological trends. Thus, the conceptual model and architecture synthesis must be flexible and scalable to ensure long-term success.

REFERENCES

1. Smith, J., and Nair, R. "Virtual Machines: Versatile Platforms for Systems and Processes." *The Morgan Kaufmann Series in Computer Architecture and Design). In Morgan Kaufmann Publishers Inc.* 2005. https://www.sciencedirect.com/book/9781558609105/virtual-machines
2. Kivity, A., Kamay, Y., Laor, D., Lublin, U., and Liguori, A. "Kvm: the Linux virtual machine monitor." *In Proceedings of the Linux Symposium.* Vol. 1, pp. 225-230. 2007 https://www.researchgate.net/publication/228661937_KVM_The_Linux_virtual_machine_monitor
3. Tolstoluzka O. Study of the dependence of the time of the decision of the assignment problem on the width of the parallel process. Weapon systems and military equipment. Scientific journal. – Kh.: HUPS named after I. Kozheduba. – 2007. – Issue 3(11) - pp. 133-135. https://journal-hnups.com.ua/index.php/soivt
4. Barrett, R., Berry, M., Chan, T. F., Demmel, J., Donato, J., Dongarra, J., and Van der Vorst, H. "Templates for the solution of linear systems: building blocks for iterative methods", vol. 43. Siam. 1994 https://www.netlib.org/templates/templates.pdf
5. R. P. Goldberg. "Survey of virtual machine research." *IEEE computer magazine*, 7(6), 34-45. 1974 https://ieeexplore.ieee.org/document/6323581
6. Mell, P., and Grance, T. "The NIST definition of cloud computing." 2011 https://csrc.nist.gov/pubs/sp/800/145/final
7. Wood, T., Shenoy, P., Venkataramani, A., and Yousif, M. "Black-box and gray-box strategies for virtual machine migration." *In Proceedings of the 4th USENIX conference on Networked systems design & implementation*. p. 17. 2007 https://www.usenix.org/conference/nsdi-07/black-box-and-gray-box-strategies-virtual-machine-migration
8. Clark, C., Fraser, K., Hand, S., Hansen, J. G., Jul, E., Limpach, C., and Warfield, A. "Live migration of virtual machines." *In Proceedings of the 2nd conference on Symposium on Networked Systems Design*

& Implementation, vol. 2. pp. 273-286. 2005 https://www.semanticscholar.org/paper/Live-migration-of-virtual-machines-Clark-Fraser/2f2cdd7b0c98b5e43b61272d2ac3ebb5cd29041d
9. Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., and Brandic, I. “Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility.” *Future Generation Computer Systems*, 25(6), 599-616. 2009 https://www.sciencedirect.com/science/article/abs/pii/S0167739X08001957
10. Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., and Zaharia, M. A view of cloud computing. *Communications of the ACM*, 53(4), 50-58. 2010 https://www.researchgate.net/publication/220422375_A_View_of_Cloud_Computing
11. Sotomayor, B., Montero, R. S., Llorente, I. M., and Foster, I. “Virtual infrastructure management in private and hybrid clouds.” *IEEE Internet computing*, 13(5). 2009 https://www.researchgate.net/publication/224587421_Virtual_Infrastructure_Management_in_Private_and_Hybrid_Clouds

ЛІТЕРАТУРА

1. Smith, J., and Nair, R. “Virtual Machines: Versatile Platforms for Systems and Processes.” *The Morgan Kaufmann Series in Computer Architecture and Design). In Morgan Kaufmann Publishers Inc.* 2005. https://www.sciencedirect.com/book/9781558609105/virtual-machines
2. Kivity, A., Kamay, Y., Laor, D., Lublin, U., and Liguori, A. “Kvm: the Linux virtual machine monitor.” *In Proceedings of the Linux Symposium.* Vol. 1, pp. 225-230. 2007 https://www.researchgate.net/publication/228661937_KVM_The_Linux_virtual_machine_monitor
3. Tolstoluzka O. Study of the dependence of the time of the decision of the assignment problem on the width of the parallel process. Weapon systems and military equipment. Scientific journal. – Kh.: HUPS named after I. Kozheduba. – 2007. – Issue 3(11) - pp. 133-135. https://journal-hnups.com.ua/index.php/soivt
4. Barrett, R., Berry, M., Chan, T. F., Demmel, J., Donato, J., Dongarra, J., and Van der Vorst, H. “Templates for the solution of linear systems: building blocks for iterative methods”, vol. 43. Siam. 1994 https://www.netlib.org/templates/templates.pdf
5. R. P. Goldberg. “Survey of virtual machine research.” *IEEE computer magazine*, 7(6), 34-45. 1974 https://ieeexplore.ieee.org/document/6323581
6. Mell, P., and Grance, T. “The NIST definition of cloud computing.” 2011 https://csrc.nist.gov/pubs/sp/800/145/final
7. Wood, T., Shenoy, P., Venkataramani, A., and Yousif, M. “Black-box and gray-box strategies for virtual machine migration.” *In Proceedings of the 4th USENIX conference on Networked systems design & implementation*. p. 17. 2007 https://www.usenix.org/conference/nsdi-07/black-box-and-gray-box-strategies-virtual-machine-migration
8. Clark, C., Fraser, K., Hand, S., Hansen, J. G., Jul, E., Limpach, C., and Warfield, A. “Live migration of virtual machines.” *In Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation*, vol. 2. pp. 273-286. 2005 https://www.semanticscholar.org/paper/Live-migration-of-virtual-machines-Clark-Fraser/2f2cdd7b0c98b5e43b61272d2ac3ebb5cd29041d
9. Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., and Brandic, I. “Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility.” *Future Generation Computer Systems*, 25(6), 599-616. 2009 https://www.sciencedirect.com/science/article/abs/pii/S0167739X08001957
10. Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., and Zaharia, M. A view of cloud computing. *Communications of the ACM*, 53(4), 50-58. 2010 https://www.researchgate.net/publication/220422375_A_View_of_Cloud_Computing
11. Sotomayor, B., Montero, R. S., Llorente, I. M., and Foster, I. “Virtual infrastructure management in private and hybrid clouds.” *IEEE Internet computing*, 13(5). 2009 https://www.researchgate.net/publication/224587421_Virtual_Infrastructure_Management_in_Private_and_Hybrid_Clouds

Тележенко Денис Олександрович *аспірант; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: denisque75@gmail.com; ORCID: 0000-0002-8377-8517.*

Толстолузька Олена Геннадіївна *д. т. н., с. н. с.; професор кафедри теоретичної та прикладної системотехніки; Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022; e-mail: elena.tolstoluzka@karazin.ua; ORCID: 0000-0003-1241- 7906.*

Концептуальна модель синтезу архітектури віртуальних розподілених систем

Актуальність. У сучасному інформаційному середовищі віртуальні розподілені системи є ключовим компонентом для ефективного використання ресурсів, забезпечення гнучкості та забезпечення надійності. Швидкий розвиток технологій віртуалізації та зростання потреб у розподілених обчисленнях створює необхідність у розробці концептуальної моделі синтезу архітектури віртуальних розподілених систем, що враховує сучасні вимоги та технологічні тренди.
Мета. Метою даної наукової статті є розробка концептуальної моделі синтезу архітектури віртуальних розподілених систем. Основна увага зосереджується на ключових компонентах системи, таких як апаратне забезпечення, гіпервізори, віртуальні машини та модулі управління. Дослідження має на меті визначити методологічні принципи синтезу архітектури, що охоплюють аналіз вимог, проектування, реалізацію, тестування, оцінку та оптимізацію роботи віртуальних розподілених систем.
Методи дослідження. Для досягнення поставленої мети використовуються методи аналізу, синтезу, моделювання та експерименту. Аналізуються вимоги до системи та стандартні практики у галузі віртуалізації та розподілених систем. Застосовуються методи проектування архітектури, вибору технологій та визначення взаємодії між компонентами системи. Розробляються моделі та проводяться експерименти для оцінки та оптимізації пропонованої концептуальної моделі.
Результати. У результаті проведеного дослідження розроблена концептуальна модель синтезу архітектури віртуальних розподілених систем, що враховує ключові компоненти та методологічні принципи. Дослідження підтверджує важливість глибокого розуміння вимог до системи та вибору відповідних технологій для успішного синтезу архітектури. Особлива увага приділена ролі гіпервізорів та віртуальних машин в системі, а також їх взаємодії з апаратним забезпеченням та можливостями управління ресурсами.
Висновки. Дана стаття має значний практичний і науковий внесок і може бути корисною для дослідників та практиків у галузі віртуалізації та розробки віртуальних розподілених систем.

Ключові слова: *віртуальні розподілені системи (ВРС), синтез архітектури, концептуальна модель, апаратне забезпечення, гіпервізор, віртуальні машини, модуль управління, аналіз вимог, проектування архітектури, реалізація та тестування, оцінка та оптимізація, віртуалізація, масштабованість, надійність, безпека, обчислювальні системи.*

Надійшла у першій редакції 02.08.2022, в останній - 03.09.2022.