Timeliness of the Reserved Maintenance by Duplicated Computers of Heterogeneous Delay-Critical Stream*

Vladimir A. Bogatyrev[1[0000-0003-0213-0223]], Stanislav V. Bogatyrev[2[0000-0003-0836-8515]] and Aleksey N. Derkach[1[0000-0002-0108-319X]]

[1] Saint-Petersburg National Research University of Information Technologies, Mechanics and Optics, Kronverksky prospect, 49, Saint Petersburg, 197101, Russian Federation
[2] NEO St. Petersburg Competency Center, Saint-Petersburg 194214, Russian Federation
vladimir.bogatyrev@gmail.com

Abstract. The research of ways to improve the functional reliability and timeliness of service requests in duplicate computer systems was conducted. Timeliness increases as a result of the reserved maintenance of delay-critical requests in queues. The timeliness of the calculations is determined by the probability that the delay in waiting for service requests do not exceed the established maximum permissible time. The effectiveness of the reserved execution of delay-critical requests during load balancing as a result of the distribution of non-reserved requests among workable computers is shown.

Keywords: timeliness, cluster, reserved service, heterogeneous stream.

1 Introduction

High reliability and fault tolerance [1-3] are supported during clustering and virtualization in real-time computer systems that do not allow breaks in the computational process. In such systems, the migration of virtual machines (VM) between physical nodes of the cluster is required in order to reconfigure to adapt the system to the accumulation of failures [4-6]. The migration of virtual resources and computational processes in real-time systems can be used while maintaining the continuity of the computational process after failures of physical nodes [7-8]. For fault tolerance, the system supports at least two copies of the VM in the memory of different physical computers. VM virtual disk images are stored on dedicated or distributed data storage with synchronous data replication. The backup computer must support an up-to-date copy of the RAM [7-9] of the active VM. Recovery time after failures depends on the structure and amount of data storage, which can be shared or local to each computer. When organizing a VM migration through a multi-tier network [10- 12], it is necessary to take into account network delays [13], including multi-path data transmission and reserved transfers through aggregated channels [14-15].

Models and organization of embedded redundant computer systems are considered in [16]. Markov models of reliability of cluster systems with VM migration that was proposed in [17] make it possible to determine stationary and non-stationary availability factors of the system characterizing cluster structural reliability. Markov duplicate computer systems, taking into account the influence of the control system on the readiness and safety of systems, are proposed in [18].

For real-time systems, the functional reliability of a cluster is determined not only by the probability of its readiness to perform the required functions but also by the probability of timely execution of delay-critical queries [18].

In [18], it was shown that redundant servicing in cluster systems potentially allows increasing the probability of timely servicing of delay-critical requests while reducing the average waiting time. The backup service of delay-critical requests is considered successful if at least one of the created copies of the request is executed in a timely manner. The redundant service of requests leads to a technical contradiction. On the one hand, it leads to an increase in load, which means additional expectations of requests\` copies in queues. On the other hand, to the potential possibility that a certain copy can be served much faster than the others.

The purpose of the research is to study the possibilities of increasing the functional reliability and timeliness of servicing duplicated computer systems as a result of redundant maintenance of delay-critical requests in queues.

The object of the study - Fault tolerance duplicate complex with a cluster architecture. The cluster contains two computers with local storage devices. Servers are connected through a switch. The data storage system is represented as local storage for each physical node as a hard disk. Synchronous data replication is maintained between local repositories to ensure fault tolerance [16].

2 Evaluation of the Requests\` Timeliness.

The timeliness of the calculations is determined by the probability that the delay in waiting for service requests do not exceed the established maximum permissible time t. Timeliness characterizes the functional reliability of the system.

We assume that the request stream is heterogeneous and contains requests of different criticality to the execution time.

For the two-machine cluster in question, inoperable and workable states are possible, in which the input request stream can be executed on two or one of the computers. Each computer node is represented as a single-channel queueing system with an infinite queue [19, 20].

Let us single out the requests of two grades of criticality to the waiting time t_1 and t_2 ($t_1 < t_2$), arriving with intensities Λ, $\Lambda\beta$ and executing for the same average time v.

In the case of readiness (operability) of one computer out of two and the lack of priorities in service, the probability of fulfilling requests with waiting for criticality (maximum allowable waiting time) t1 and t2 (t1 < t2) is calculated respectively as:

$$p_1(t_1) = 1 - \Lambda(1+\beta)\mathrm{v}e^{\left(\Lambda(1+\beta)-\frac{1}{v}\right)t_1},$$

$$p_2(t_2) = 1 - \Lambda(1+\beta)\mathrm{v}e^{\left(\Lambda(1+\beta)-\frac{1}{v}\right)t_2}.$$

With the operability (readiness) of two computers, there are possible options for organizing the maintenance of a heterogeneous stream of requests, among which we consider options without and with redundant maintenance of critical requests.

For the case of non-redundant service, consider the options:

A_1: Stream maintenance is divided by computers — the first computer performs requests for t1 criticality, and the second computer performs t_2 criticality.

A_2: Stream maintenance is not divided among computers - any request can with a certain probability be sent both to the queue of the first and second computers (the probabilities of sending requests to different computers for different threads may differ).

A_3: Stream maintenance of criticality to t_2 delays is performed exclusively by the second computer, and criticality to delays t1 is divided between two computers with probability g.

For the maintenance organization option B_1, the probability of fulfilling requests with a waiting criticality (maximum allowable waiting time) t_1 and t_2 ($t_1 < t_2$) is calculated respectively as:

$$p_1(t_1) = 1 - \Lambda\mathrm{v}e^{\left(\Lambda-\frac{1}{v}\right)t_1},$$

$$p_2(t_2) = 1 - \Lambda\beta\mathrm{v}e^{\left(\Lambda\beta-\frac{1}{v}\right)t_2}.$$

The probability that requests for criticality to the delays of both t_1 and t_2 will be fulfilled in a timely manner will determine how:

$$p_{12}(t_1,t_2) = p_1(t_1)p_2(t_2). \tag{1}$$

For the organization of the computational process with duplication of requests of the first stream and with the execution of copies on different computers, select the following options:

B_1: the second stream is completely directed to the second computer for maintenance.

B_2: the second stream with probabilities g and (1-g) is sent to the second or first computers, respectively.

For option B_1, the probability of timely execution of a copy of requests of the first most critical waiting stream during time t_1 in the first and second computers is defined respectively as:

$$p_{11}(t_1) = 1 - \Lambda \mathrm{v} e^{\left(\Lambda - \frac{1}{v}\right)t_1},$$

$$p_{12}(t_1) = 1 - \Lambda\left(1+\beta\right)\mathrm{v} e^{\left(\Lambda(1+\beta) - \frac{1}{v}\right)t_1},$$

and the probability of timely execution of at least one of the two copies of the request during t_1 is as:

$$p_1(t_1) = 1 - \left(\Lambda \mathrm{v}\right)^2 e^{\left(\Lambda - \frac{1}{v}\right)t_1} \left(1+\beta\right) e^{\left(\Lambda(1+\beta) - \frac{1}{v}\right)t_1}.$$

For the second request stream, the probability of timely execution of requests in a time not exceeding t_2 is calculated as:

$$p_{12}(t_1) = 1 - \Lambda\left(1+\beta\right)\mathrm{v} e^{\left(\Lambda(1+\beta) - \frac{1}{v}\right)t_2}.$$

The probability of timely execution of requests for both streams is determined by the formula (1).

For option B_2, the probability of timely execution of a request copies for the first stream during time t_1 in the first and second computers is defined respectively as:

$$p_{11}(t_1) = 1 - \Lambda\left(1+\beta g\right)\mathrm{v} e^{\left(\Lambda(1+\beta g) - \frac{1}{v}\right)t_1},$$

$$p_{12}(t_1) = 1 - \Lambda\left(1+\beta - g\beta\right)\mathrm{v} e^{\left(\Lambda(1+\beta - g\beta) - \frac{1}{v}\right)t_1},$$

the probability of the timely execution of at least one copy of the request for the first stream during time t_1:

$$p_1(t_1) = 1 - \left(\Lambda v\right)^2 \left(1+\beta g\right) e^{\left(\Lambda(1+\beta g) - \frac{1}{v}\right)t_1} \left(1+\beta - g\beta\right) e^{\left(\Lambda(1+\beta - g\beta) - \frac{1}{v}\right)t_1},$$

The probability of timely execution of requests of the second stream in a timeless than t_2 is calculated as:

$$p_{12}(t_1) = 1 - \Lambda\left(1+\beta - g\beta\right)\mathrm{v} e^{\left(\Lambda(1+\beta - g\beta) - \frac{1}{v}\right)t_2}.$$

The probability of timely execution of requests for the first and second threads is determined by the formula (1).

3 Evaluation of Structural Reliability

A Markov model of a duplicated cluster is constructed in accordance with [16] to evaluate the structural reliability determined by the operational readiness ratio.

The disciplines of maintenance for a duplicated computer system were investigated in [16]:

- D1 – operational recovery, which begins immediately after the failure;
- D2 – a recovery that begins after the system has passed into an inoperative state or a condition of a certain level of degradation.

The diagram of states and transitions for recovery discipline D_2 is shown in fig. 1. In the diagram, the failure intensity is denoted by λ_0 and restoration by μ_0 of the server; of the disk by λ_1, μ_1; of the commutator by λ_2, μ_2.

In the diagram, two lines cross out inefficient nodes, and workable ones that are not involved in the computation process by one line, the inoperative states of the system are darkened.

The system of differential equations in accordance with the diagram of states and transitions in Fig. 1 has the form:

$$\begin{cases} P_0^`(t) = -(2\lambda_0 + \lambda_2 + 2\lambda_1)P_0(t) + \mu_{40}P_4(t) + \mu_{50}P_5(t) + \mu_{60}P_6(t) + \mu_{70}P_7(t) + \mu_{80}P_8(t), \\ P_1^`(t) = -(\lambda_1 + \lambda_0)P_1(t) + 2\lambda_0 P_0(t), \\ P_2^`(t) = -(\lambda_1 + \lambda_0)P_2(t) + \lambda_2 P_0(t), \\ P_3^`(t) = -(\lambda_1 + \lambda_0)P_3(t) + 2\lambda_1 P_0(t), \\ P_4^`(t) = -\mu_{40}P_4(t) + \lambda_1 P_1(t) + \lambda_0 P_3(t), \\ P_5^`(t) = -\mu_{50}P_5(t) + \lambda_0 P_1(t), \\ P_6^`(t) = -\mu_{60}P_6(t) + \lambda_1 P_2(t), \\ P_7^`(t) = -\mu_{70}P_7(t) + \lambda_0 P_2(t), \\ P_8^`(t) = -\mu_{80}P_8(t) + \lambda_1 P_3(t), \end{cases}$$

Wherein

$$\mu_{40} = \left(1/\mu_5 + 1/\mu_0 + 1/\mu_1 \right)^{-1}$$

$$\mu_{50} = \left(1/\mu_5 + 2/\mu_0 \right)^{-1}$$

$$\mu_{60} = \left(\frac{1}{\mu_5} + \frac{1}{\mu_1} + \frac{1}{\mu_2} \right)^{-1}$$

$$\mu_{70} = \left(\frac{1}{\mu_5} + \frac{1}{\mu_0} + \frac{1}{\mu_2} \right)^{-1}$$

$$\mu_{80} = \left(\frac{1}{\mu_5} + \frac{2}{\mu_1} \right)$$

$$\mu_5 = \left(\frac{1}{\mu_3} + \frac{1}{\mu_4} \right)^{-1}$$

Where μ_5 data recovery intensity, while μ_3 - load intensity of the actual data replica, μ_4 - the intensity of work associated with the launch of the VM and user applications on the backup server.

If we determine the probabilities of states by solving the presented system of equations, then we can determine the availability factor as the sum of working states:

$$k = \sum_{i=0}^{3} P_i.$$

The probability of timely execution of requests R, taking into account the probability of requests when one or two computers are ready:

$$R = p_2 P_0 + p_1 \sum_{i=1}^{3} P_i,$$

where p_1 and p_2 are the probabilities of timely maintenance of requests when one and two computers are ready, taking into account the above organization of maintaining the request stream.

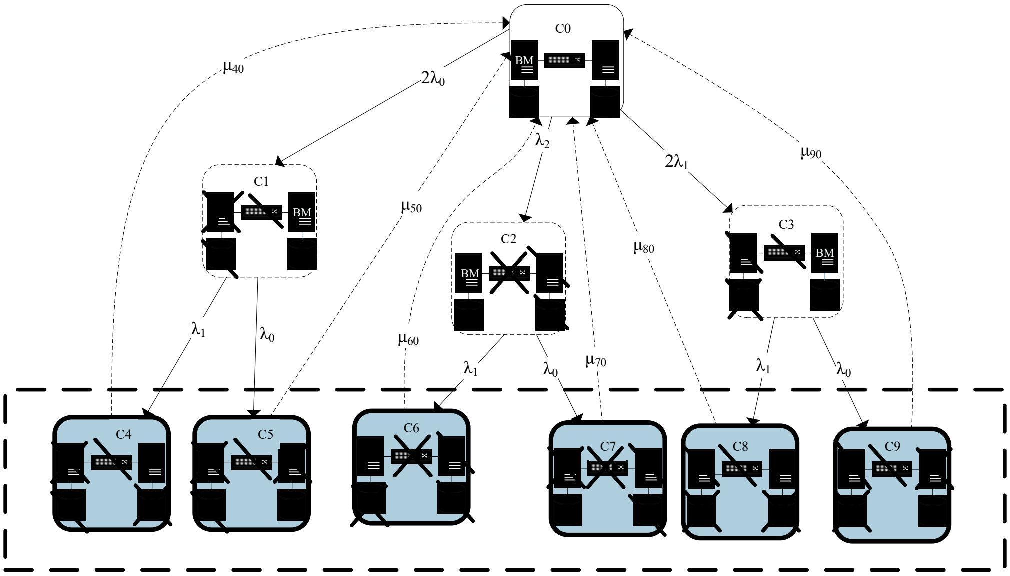

Fig. 1. Markov diagram of states and transitions of a cluster

4 The example of Calculating the Probability of Timely Execution of Requests.

Let us determine the probability of the timeliness of the redundant service with a heterogeneity of the input stream, which includes requests for two grades of criticality for the waiting time $t_1 = 0.2$ s, $t_1 = 0.4$ s for the same query execution time $v = 0.1$ s.

The probability`s dependence of requests timely execution of different criticality on the intensity of the requests stream Λ is shown in Fig. 3. In Fig. 3 at $\beta = 0.5, 0.8, 1.5$, the curves 1-3 correspond to the variant of maintenance organization streams without redundant maintenances of the first requests stream A_1, and the curves 4-6 with redundant maintenances according to option B_2.

Calculations show the expediency of requests' redundant maintenance that is most critical to the expectation, however, this effect is lost as the intensity of input streams Λ increases.

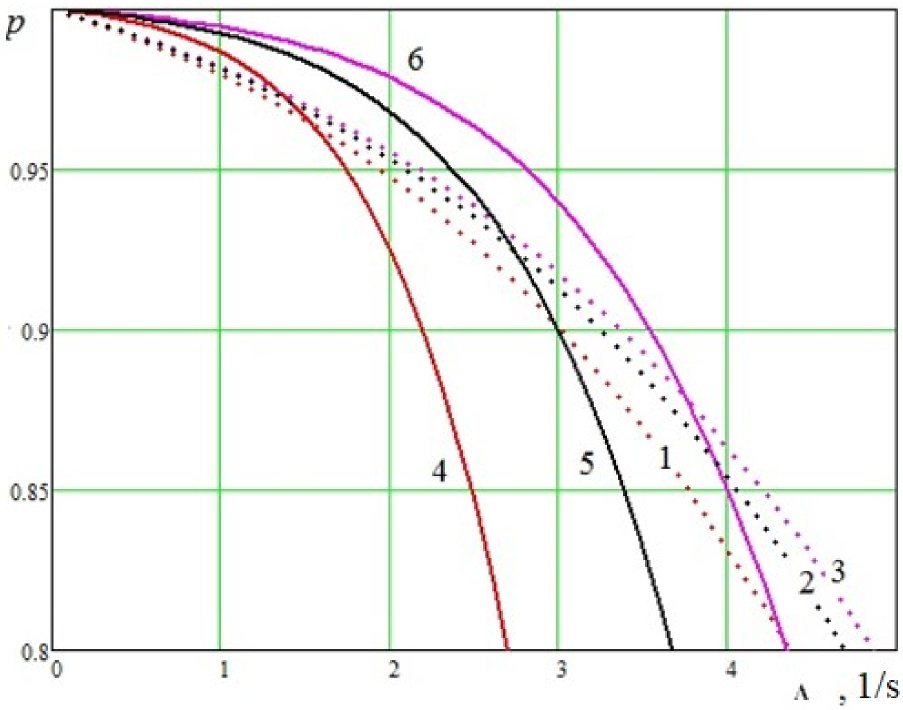

Fig. 2. Dependence of the probability of requests' timely execution on the intensity of the input stream with redundant and non-redundant maintenance requests

For the disciplines of redundant maintenance of the first stream B1 and B2 there are in Fig. 4 the dependence of the probability of timely servicing the requests of two streams on the intensity of requests Λ and the requests` shares g and (1-g) of the second stream which are sent in discipline B2 to the queue of the first or second computers. In fig. 2 the curves 1-3 for discipline B1 correspond to β = 0.5, 0.8, 1.5, and the curves 4-6 for discipline B2 with g = 0.8, correspond to values β = 0.5, 0.8, 1.5. The presented dependencies show the importance of the stream control`s effect (change g) on the timeliness of maintaining requests for the total requests` stream of different criticality to delays. The presented dependencies show the importance of the influence of the control of requests` streams, which have different criticality to delays (change g), on the timeliness of service of requests of the total stream.

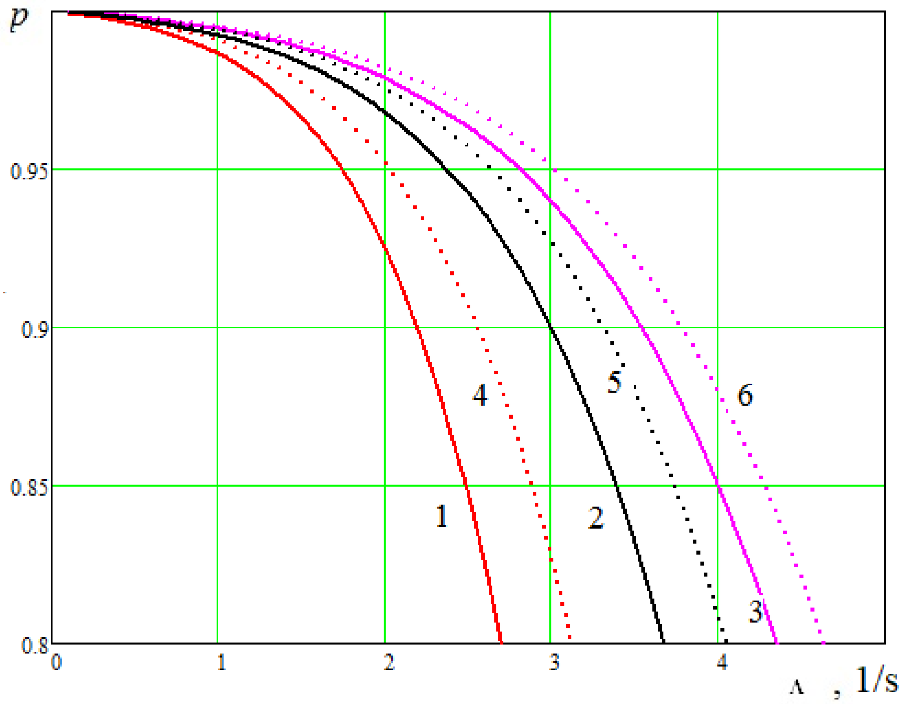

Fig. 3. Probability`s dependence of timely maintenance requests of two threads with redundant service for option B2

In Fig. 5 there is the probability`s dependence of timely servicing of requests for two streams with redundant services for option B2 on the fraction g, requests of the second stream which sent to the second computer. In the figure at $\beta = 1.5$, the curves 1–3, and at $\beta = 1$, the curves 4–6 correspond to the intensity of the input stream $\Lambda = 3, 3.5, 4$ 1/s.

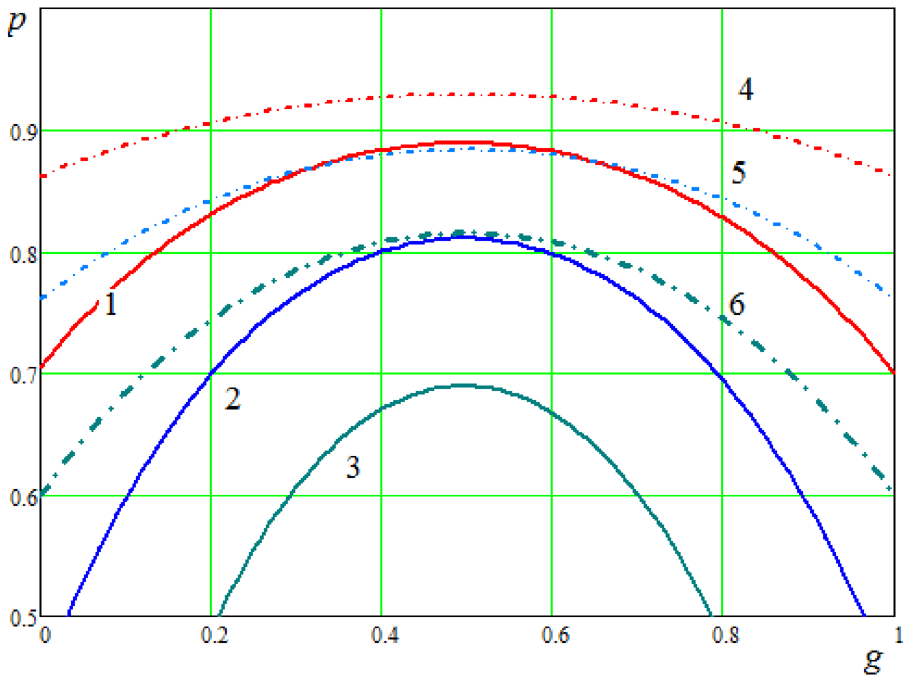

Fig. 4. The probability`s dependence of timely servicing of requests for two streams with redundant services for option B_2 on the fraction g, requests of the second stream which sent to the second computer

5 Conclusions

For duplicate cluster real-time systems, the options for organizing the maintenance of a heterogeneous requests` stream that has different criticalities to their delays in queues are proposed.

The influence`s materiality of the organization of maintaining requests with varying criticality to delays in the computer nodes` queue, that have remained operable, on functional reliability and the probability of timely execution of requests for a heterogeneous stream is shown.

The effectiveness of the redundant execution of delay-critical requests during load balancing as a result of the distribution of other requests among workable computers is shown.

References

1. Kopetz, H.: Real-Time Systems: Design Principles for Distributed Embedded Applications. 2nd edn. Springer, Germany (2011). DOI: 10.1007/978-1-4419-8237-7
2. Utkin, L.V., Coolen, F.P.A., Robust weighted SVR-based software reliability growth model.

Reliability Engineering & System Safety 176, 93-101 (2018), DOI: 10.1016/j.ress.2018.04.007.
3. Sorin, D.: Fault Tolerant Computer Architecture. Morgan & Claypool, USA (2009), 104 p. DOI: 10.2200/S00192ED1V01Y200904CAC005.
4. Sahni, S., Varma, V.: A hybrid approach to live migration of virtual machines. Proc. IEEE Int. Conf. on Cloud Computing for Emerging Markets (CCEM 2012), 12–16, Bengalore, India (2012), DOI: 10.1109/CCEM.2012.6354587.
5. Knowledge sharing portal UNIX/Linux-systems, open-source systems, networks, and other related things. Available at: < http://xgu.ru/wiki/Kemari > Last accessed 2019/03/25.
6. Dittner, R., Rule, D.: The Best Damn Server Virtualization Book Period. 2nd edn. Syngress, USA (2011).
7. Chandak, A., Jaju, K., Kanfade, A.: Dynamic Load Balancing of Virtual Machines using QEMU-KVM. International Journal of Computer Applications (0975 – 8887) 6(46), 10-14 (2012).
8. Adamova, K.: Anomaly Detection with Virtual Service Migration in Cloud Infrastructures. Master Thesis 263-0800-00L (2012).
9. Hu, L., Zhao, J., Xu G., Ding Y. HMDC: Live Virtual Machine Migration Based on Hybrid Memory Copy and Delta Compression, Appl. Math. Inf. Sci. 7, No. 2L, 639-646, China. (2013), DOI: 10.12785/amis/072L38.
10. Poymanova, E.D., Tatarnikova, T.M. Models and Methods for Studying Network Traffic. 2018 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF) (26-30 Nov. 2018), DOI: 10.1109/WECONF.2018.8604470.
11. Kutuzov, O.I., Tatarnikova, T.M. On the Acceleration of Simulation Modeling. XXI International Conference on Soft Computing and Measurements (SCM'2018) (23-25 May 2018)
12. Korobeynikov, A.G., Fedosovsky, M.E., Zharinov, I.O., Shukalov, A.V., Gurjanov, A.V. Development of conceptual modeling method to solve the tasks of computer-aided design of difficult technical complexes on the basis of category theory. International Journal of Applied Engineering Research 6(12), 1114-1122 (2017).
13. Zhmylev, S., Martynchuk, I.G., Kireev, V.I., Aliev, T. Analytical methods of nonstationary processes modeling. CEUR Workshop Proceedings, IET - 2019, Vol. 2344 (2019).
14. Bogatyrev, A.V., Bogatyrev, S.V., Bogatyrev, V.A. Analysis of the Timeliness of Redundant Service in the System of the Parallel-Series Connection of Nodes with Unlimited Queues. 2018 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF), pp. 1-4 (2018), DOI: 10.1109/WECONF.2018.8604379.
15. Bogatyrev, V.A On interconnection control in redundancy of local network buses with limited availability. Engineering Simulation, 16 (4), pp. 463-469 (1999).
16. Bogatyrev, V.A., Aleksankov, S.M., Derkach, A.N. Model of Cluster Reliability with Migration of Virtual Machines and Restoration on Certain Level of System Degradation. 2018 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF), pp. 1-4 (2018), DOI: 10.1109/WECONF.2018.8604317.
17. Ageev, A.M.: Configuring of Excessive Onboard Equipment Sets. Journal of Computer and Systems Sciences International 4 (57), 640-654 (2018). DOI: 10.1134/S1064230718040020
18. Bogatyrev, V.A., Bogatyrev, A.V. Functional Reliability of a Real-Time Redundant Computational Process in Cluster Architecture Systems. Automatic Control and Computer Sciences, Vol. 49 No. 1, pp. 46-56 (2015), DOI: 10.3103/S0146411615010022.
19. Kleinrock, L. Queueing Systems, Volume I: Theory. Wiley Interscience, USA. p. 417 (1975), DOI: 10.1002/net.3230060210.
20. Kleinrock, L. Queueing Systems, Volume II: Computer Applications. Wiley Interscience, USA. p. 576 (1976), doi: 10.1002/net.3230070308.