

MODEL OF THE RELIABILITY ANALYSIS OF THE DISTRIBUTED COMPUTER SYSTEMS WITH ARCHITECTURE “CLIENT-SERVER”

I.V. Kovalev¹, P.V. Zelenkov¹, M.V. Karaseva¹, M.Yu. Tsarev¹,
R.Yu. Tsarev²

1 Siberian State Aerospace University, Krasnoyarsk, Russia

2 Siberian Federal University, Krasnoyarsk, Russia

E-mail: zelenkov@sibsau.ru

Abstract. The paper considers the problem of the analysis of distributed computer systems reliability with client-server architecture. A distributed computer system is a set of hardware and software for implementing the following main functions: processing, storage, transmission and data protection. This paper discusses the distributed computer systems architecture «client-server». The paper presents the scheme of the distributed computer system functioning represented as a graph where vertices are the functional state of the system and arcs are transitions from one state to another depending on the prevailing conditions. In reliability analysis we consider such reliability indicators as the probability of the system transition in the stopping state and accidents, as well as the intensity of these transitions. The proposed model allows us to obtain correlations for the reliability parameters of the distributed computer system without any assumptions about the distribution laws of random variables and the elements number in the system.

1. Introduction

The peculiarity of many applications “client-server” (CS-application) of modern distributed computer systems is their diversity and dispersal [1], [2]. There are some specific features in designing distributed computer systems. First of all it is the dependence of the architectural model from a some non-functional system requirements such as performance, security, safety, reliability [3]. In the architectural solutions analysis within the modern systems it is essential for the developer to have the opportunity to the evaluate architectural reliability of CS-applications, as an important part of the system [4-8].

In addition, in the distributed computer system different system components can be implemented in different programming languages and can be run in different processor types. The data models, information presentation and communication protocols are not uniform in the distributed systems; therefore architectural reliability is an important factor for design [9], [10]. As a rule, the intermediate components are formed from ready-made components and do not require developers to do special debugging.

In architecture “client-server” a server part is usually installed in a separate PC “server-client” part of the software in working places with the different functional structure of software. A server- client part of the software can operate in different operating environments. In this regard, problem of the parameters reliability analysis of distributed computer systems with client-server architecture is important and actual.



2. Scheme of a distributed computer system functioning with architecture “client-server”

Let's assume that the system is operating in normal (non-boundary) conditions, so you can avoid the independence of individual failures. In the proposed model the subsystems are not regenerated.

A distributed computer system (DCS) is understood to be a unity of hardware and software tools implementing the following main functions [11]: processing, storage, transmission and data protection. The structure of a computer system is presented in Fig. 1.

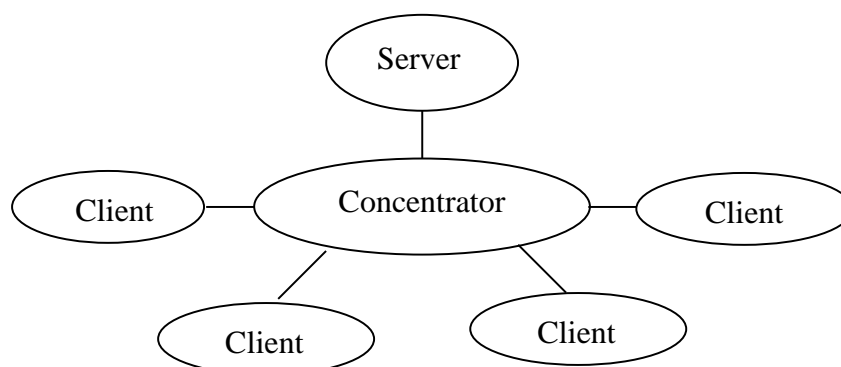


Fig.1. Structure of the distributed computer system with radial structure.

Here a server is an element included a system of data storage (SDS) with their system of data transition (SDT) and security system (SS), a client-element system containing data processing system (DPS) with their SDT and SS; and the concentrator serves to link a client and server and consists of SDT and SS. We consider the system with N clients and one server. We can distinguish the following types of the system failures security: hidden and false [12]. At the hidden failure SS does not parries the failures of other subsystems, with false failure SS spontaneously produces protective functions while normal operation of SDS, DPS and SDT and leads to the system stopping.

In Fig. 2 a graphical scheme of the distributed computer system functioning that reflects the failures of its subsystems and further developing situations is shown. Solid lines indicate the transferring of elements, dashed line indicate the situations development of elements failures.

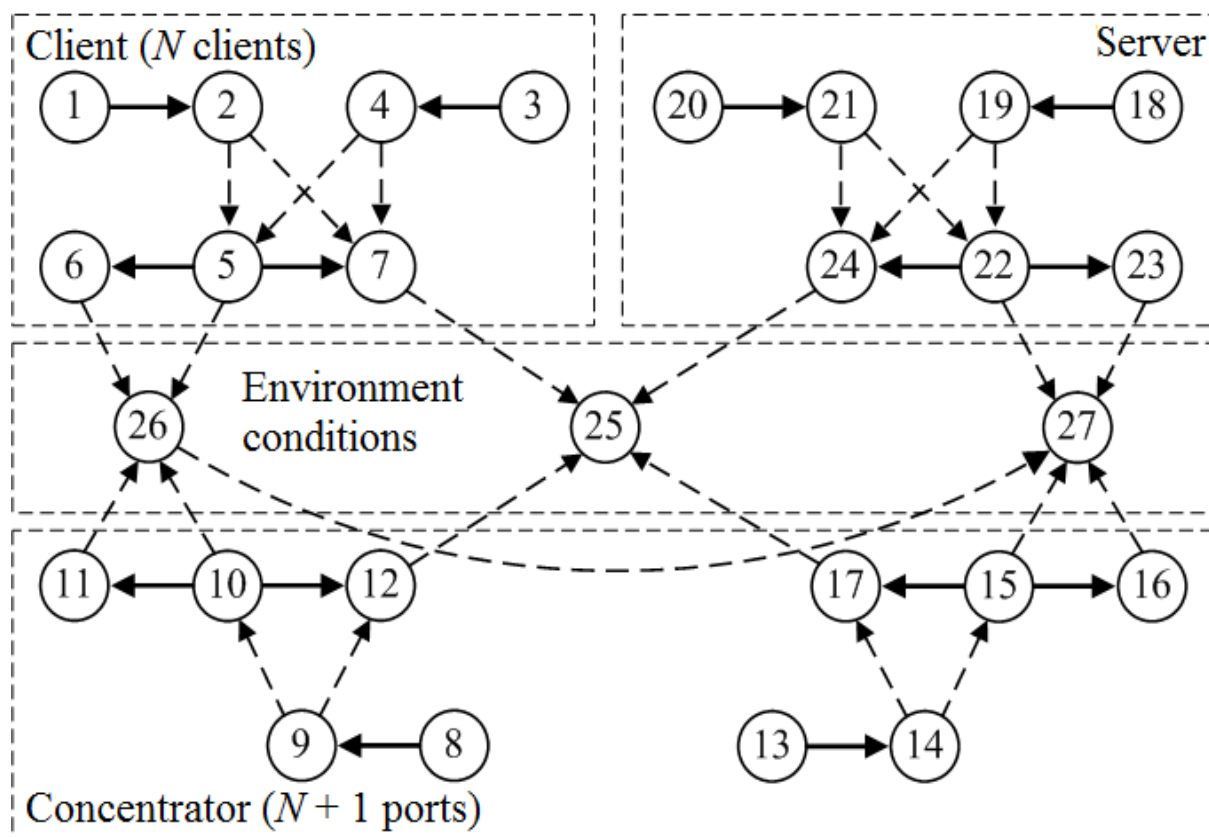


Fig.2. Scheme of the distributed computer system operating

State in the graph means the following:

- 1 – normal operation of DPS;
- 2 – failure of DPS;
- 3 – normal operation of SDT client;
- 4 – failure of SDT client;
- 5 – normal operation of SS client;
- 6 – false failure of SS client;
- 7 – hidden failure of SS client;
- 8 – normal operation of SDT port concentrator a client is attached to;
- 9 – failure of SDT port concentrator a client is attached to;
- 10 – normal operation of SS port concentrator a client is attached to;
- 11 – false failure of SS port concentrator a client is attached to;
- 12 – hidden failure of SS port concentrator a client is attached to;
- 13 – normal operation of SDT port concentrator a server is attached to;
- 14 – failure of SDT port concentrator a server is attached to;
- 15 – normal operation of SS port concentrator a server is attached to;
- 16 – false failure of SS port concentrator a server is attached to;
- 17 – hidden failure of SS port concentrator a server is attached to;
- 18 – normal operation of SDS;
- 19 – failure of SDS;
- 20 – normal operation of SDT server;
- 21 – failure of SDT server;
- 22 – normal operation of SS server;

- 23 – false failure of SS server;
- 24 – hidden l failure of SS server;
- 25 – emergency condition of DCS;
- 26 – state of the reduced efficiency of the DCS;
- 27 – stopping state of the DCS.

At failure of DPS or SDT on client or failure of SDT port concentrator a client is attached to at functioning of the relevant SS client or port concentrator is transferring into the stopping state; the system itself is transferring into the state of reduced efficiency (as with false failure of SS client or port concentrator a client is attached to). At failure SDS or SDT server or SDT port concentrator attached to server you are normally working corresponding SS a distributed computer system switches to transferring to the stopping state as and when the false failure of SS server or hub port concentrator, attached to and at stopping of all clients or port concentrators attached to clients. At any subsystems failure in a hidden failure responsible for its control SS transfers into the emergency state.

3. Mathematical analysis model of the distributed computer systems of the architecture “client-server”

Let's consider the system behavior on the interval $[0, t]$. Introduce the following notations. introduce the necessary. Let ψ is the time to failure of the DPS with the distribution

$$f_{\psi}(t) = P(\psi \leq t);$$

δ – time to failure of SDS with the distribution

$$F_{\delta}(t) = P(\delta \leq t);$$

$\gamma_1, \gamma_2, \gamma_3$ – time to failure of port concentrator client and server correspondingly with the distribution

$$F_{\gamma_1}(t) = P(\gamma_1 \leq t),$$

$$F_{\gamma_2}(t) = P(\gamma_2 \leq t),$$

$$F_{\gamma_3}(t) = P(\gamma_3 \leq t).$$

Introduce as ρ_1, ρ_2 , and ρ_3 the time to the hidden failures and as η_1, η_2 and η_3 the time to the false failures of SS of port concentrator client and server correspondingly with the distribution:

$$F_{\rho_1}(t) = P(\rho_1 \leq t), F_{\rho_2}(t) = P(\rho_2 \leq t),$$

$$F_{\rho_3}(t) = P(\rho_3 \leq t), F_{\eta_1}(t) = P(\eta_1 \leq t),$$

$$F_{\eta_2}(t) = P(\eta_2 \leq t), F_{\eta_3}(t) = P(\eta_3 \leq t).$$

The probability of the system transition in the stopping state and accidents is understood as indicators of reliability as well as the intensity of these transitions [13], [14]. First let's consider the reliability indicators for transitions in the stopping state.

Stopping state of the distributed computer system will come when in the stopping state, the clients or all ports concentrator clients, server, or hub port concentrator attached to will transfer.

The probability that on the interval $[0, t]$ the stopping state will occur could be written as follows:

$$P_o(t) = 1 - M(\bar{P}_{o\kappa 1}(t)\bar{P}_{o\kappa 2}(t)\bar{P}_{o\kappa 3}(t)\bar{P}_{oc}(t)),$$

where $P_{ok1}(t)$ is probability of the stopping state of all clients by the moment t ; $P_{ok2}(t)$ is probability of the stopping state of all port concentrator clients are attached to by the moment t ; $P_{ok3}(t)$ is probability of the stopping state of all port concentrator a server is attached to by the moment t ; $P_{oc}(t)$ – probability of the stopping state of the server by the t .

Let's introduce the following variables. Let ζ_i is the time to the stopping state of the i -th client, where $i = \overline{1, N}$. Then the probability that on the interval $[0, t]$ the stopping state of all clients will occur is determined by the following expression

$$P_{ok1}(t) = P\left(\bigvee_{i=1}^N \zeta_i \leq t\right),$$

$$\bigvee_{i=1}^N \zeta_i = \max(\zeta_1, \zeta_2, \dots, \zeta_N)$$

Using the features of indicators and mathematical expectation we receive:

$$P_{ok1}(t) = MI_{\bigvee_{i=1}^N \zeta_i \leq t} = M \prod_{i=1}^N I_{\zeta_i \leq t} = \prod_{i=1}^N MI_{\zeta_i \leq t} = \prod_{i=1}^N P(\zeta_i \leq t) = \prod_{i=1}^N (1 - \bar{P}_{ioo}(t)),$$

Where $P_{ioo}(t)$ is the probability that the stopping state of the i -th client will not occur by the moment t ; $I_{\zeta_i \leq t}$ is function-indicator ($I_{\zeta_i \leq t} = 1$ at $\zeta_i \leq t$ and $I_{\zeta_i \leq t} = 0$ at $\zeta_i > t$).

The probability that the i -th client will not be transferred into the stopping state:

$$\bar{P}_{ioo}(t) = P(I_{\rho_{1i} \geq \psi_i \wedge \gamma_{1i}} \psi_i \wedge \gamma_{1i} + I_{\rho_{1i} < \psi_i \wedge \gamma_{1i}} \eta_{1i} > t),$$

$$\text{where } \psi_i \wedge \gamma_{1i} = \min(\psi_i, \gamma_{1i}).$$

Having done some analytical transformations we receive:

$$\bar{P}_{ioo}(t) = \bar{F}_{\eta_{1i}}(t) (1 - F_{\rho_{1i}}(\psi_i) F_{\rho_{1i}}(\gamma_{1i})) \times \bar{F}_{\psi_i}(t) \bar{F}_{\gamma_{1i}}(t) + F_{\rho_{1i}}(\psi_i) F_{\rho_{1i}}(\gamma_{1i}). \quad (1)$$

Let τ_j is time to failure of j -th port concentrator a client is attached to, where $j = \overline{1, N}$. The probability that on the interval $[0, t]$ the stopping state of all ports concentrator will occur clients are attached to is determined by the following expression

$$P_{ok2}(t) = P\left(\bigvee_{j=1}^N \tau_j \leq t\right) = MI_{\bigvee_{j=1}^N \tau_j \leq t} = M \prod_{j=1}^N I_{\tau_j \leq t} = \prod_{j=1}^N MI_{\tau_j \leq t} = \prod_{j=1}^N P(\tau_j \leq t) = \prod_{j=1}^N (1 - \bar{P}_{j\tau}(t)),$$

where $\bar{P}_{j\tau}(t)$ is a server is attached to of i -th port concentrator will not occur by the moment t . $\bar{P}_{j\tau}(t)$ could be written as follows:

$$\bar{P}_{j\tau}(t) = P(I_{\rho_{2j} \geq \eta_{2j} \wedge \gamma_{2j}} \eta_{2j} \wedge \gamma_{2j} + I_{\rho_{2j} < \eta_{2j} \wedge \gamma_{2j}} \eta_{2j} > t) = \bar{F}_{\eta_{2j}}(t) (F_{\rho_{2j}}(\eta_{2j}) \bar{F}_{\gamma_{2j}}(t) + F_{\rho_{2j}}(\gamma_{2j})) \quad (2)$$

Similarly we can write down that the probability the stopping state of the port concentrate or a server is attached to on the interval $[0, t]$ will not occur:

$$\bar{P}_{ok3}(t) = \bar{F}_{\eta_{2(N+1)}}(t) (F_{\gamma_{2(N+1)}}(\rho_{2(N+1)}) \bar{F}_{\gamma_{2(N+1)}}(t) + F_{\rho_{2(N+1)}}(\gamma_{2(N+1)})). \quad (3)$$

The probability that a server was not transferred into the stopping state on the interval $[0, t]$:

$$\bar{P}_{oc}(t) = P(I_{\rho_2 \geq \delta \wedge \gamma_3} \delta \wedge \gamma_3 + I_{\rho_2 < \delta \wedge \gamma_3} \eta_3 > t).$$

Having done some transformations we receive:

$$\bar{P}_{oc}(t) = \bar{F}_{\eta_3}(t)(1 - F_{\rho_3}(\delta)F_{\rho_3}(\gamma_3))\bar{F}_{\delta}(t)\bar{F}_{\gamma_3}(t) + F_{\rho_3}(\delta)F_{\rho_3}(\gamma_3). \quad (4)$$

Then the probability that on the interval $[0, t]$ the stopping state will occur could be written as follows:

$$P_o(t) = 1 - M \left(\left(1 - \prod_{i=1}^N (1 - \bar{P}_{ioo}(t)) \right) \times \left(1 - \prod_{j=1}^N (1 - \bar{P}_{jt}(t)) \right) \bar{P}_{o\kappa 3}(t) \bar{P}_{oc}(t) \right).$$

As all the failures are independent this expression could be written as follows:

$$P_o(t) = 1 - \left(\left(1 - \prod_{i=1}^N (1 - M\bar{P}_{ioo}(t)) \right) \times \left(1 - \prod_{j=1}^N (1 - M\bar{P}_{jt}(t)) \right) M\bar{P}_{o\kappa 3}(t) M\bar{P}_{oc}(t) \right). \quad (5)$$

Having substituted (1), (2), (3) and (4) into (5) we receive:

$$P_o(t) = 1 - \left(1 - \prod_{j=1}^N (1 - \bar{F}_{\eta_{2j}}(t)(MF_{\rho_{2j}}(\gamma_{2j}) \times \bar{F}_{\gamma_{2j}}(t) + MF_{\rho_{2j}}(\gamma_{2j}))) \right) \\ \left(1 - \prod_{i=1}^N (1 - \bar{F}_{\eta_{1i}}(t) \times ((1 - MF_{\rho_{1i}}(\psi_i)MF_{\rho_{1i}}(\gamma_{1i}))\bar{F}_{\psi_i}(t) + MF_{\rho_{1i}}(\psi_i)MF_{\rho_{1i}}(\gamma_{1i}))) \right) \bar{F}_{\eta_{2(N+1)}} \\ (MF_{\rho_{2(N+1)}}(\gamma_{2(N+1)})\bar{F}_{\gamma_{2(N+1)}}(t) + MF_{\rho_{2(N+1)}}(\gamma_{2(N+1)})\bar{F}_{\eta_3}(t)(1 - MF_{\rho_3}(\delta\delta \times \\ MF_{\rho_3}(\gamma_3))\bar{F}_{\delta}(t)\bar{F}_{\gamma_3}(t) + MF_{\rho_3}(\delta\delta)M_{\rho_3}(\gamma_3)))$$

Now we can find the intensity dependence of the stopping failures from time that we designate as $\sigma_o(t)$:

$$\sigma_o(t) = \frac{dP_o(t) / dt}{1 - P_o(t)}.$$

So we received the expressions for calculation the probability of stopping state and its intensity of the distributed computer system on the interval $[0, t]$.

Now let's consider indicators of reliability referring the system transferring in accident conditions. The accident will occur if at least one client, port concentrate or a server will transfer into the accident conditions. A subsystem will transfer into the accident condition if after the hidden failure of SS the failure of the controllable subsystems will occur.

The probability that on the interval $[0, t]$ the stopping state will occur could be written as follows:

$$P_{ak}(t) = 1 - M(\bar{P}_{ak1}(t)\bar{P}_{ak2}(t)\bar{P}_{ac}(t)),$$

where $P_{ak1}(t)$ is the probability that the stopping state of one of clients will not occur by the moment t , $P_{ak2}(t)$ is the probability the accident of one of the port concentrates by the moment t ; $P_{ac}(t)$ is the probability the accident of a server by the moment t . Let's introduce the auxiliary variables. Let ω_i is time to the transferring of the i -th client into the accident conditions, where $i \in \overline{1, N}$. Then the probability that on the interval $[0, t]$ no one of clients will transfer into the accident conditions could be determined by the expression:

$$\overline{P}_{ak1}(t) = P\left(\bigwedge_{i=1}^N \omega_i > t\right).$$

Having done some analytical transformations we receive:

$$\overline{P}_{ak1}(t) = MI_{\bigwedge_{i=1}^N \omega_i > t} = M \prod_{i=1}^N I_{\omega_i > t} = \prod_{i=1}^N MI_{\omega_i > t} = \prod_{i=1}^N P(\omega_i > t) = \prod_{i=1}^N \overline{P}_{aki}(t),$$

$$\overline{P}_{ak1}(t) = P\left(\bigwedge_{i=1}^N \omega_i > t\right).$$

Having done some analytical transformations we receive:

$$\overline{P}_{ak1}(t) = MI_{\bigwedge_{i=1}^N \omega_i > t} = M \prod_{i=1}^N I_{\omega_i > t} = \prod_{i=1}^N MI_{\omega_i > t} = \prod_{i=1}^N P(\omega_i > t) = \prod_{i=1}^N \overline{P}_{aki}(t),$$

where $\overline{P}_{aki}(t)$ is the probability that the accident of the i -th client will not occur by the moment t .

The probability that on the interval $[0, t]$ the i -th client will not transfer into the accident conditions,

$$\overline{P}_{aki}(t) = 1 - P(\rho_{1i} \wedge t < \psi_i \wedge \gamma_{1i} \leq t).$$

Therefore

$$\overline{P}_{aki}(t) = 1 - I_{\rho_{1i} \leq t}(P(\psi_i \wedge \gamma_{1i} \leq t) - P(\omega_i \wedge \gamma_{1i} \leq \rho_{1i})).$$

Let ε_j is time to the failure of the of j -th port concentrator leading to the accident of DCS, where $j \in \overline{1, N+1}$. Then the probability that on the interval $[0, t]$ no one of port concentrators will transfer into the accident conditions could be determined by the expression:

$$\overline{P}_{\omega}(t) = P\left(\bigwedge_{j=1}^{N+1} \varepsilon_j > t\right) = MI_{\bigwedge_{j=1}^{N+1} \varepsilon_j > t} = M \prod_{j=1}^{N+1} I_{\varepsilon_j > t} = \prod_{j=1}^{N+1} MI_{\varepsilon_j > t} = \prod_{j=1}^{N+1} P(\varepsilon_j > t) = \prod_{j=1}^{N+1} \overline{P}_{je}(t),$$

where $\overline{P}_{je}(t)$ is the probability that of j -th port concentrator will not transfer into the accident conditions by the moment t .

$\overline{P}_{je}(t)$ could be written as follows:

$$\overline{P}_{je}(t) = 1 - P(\rho_{2j} \wedge t < \gamma_{2j} \leq t).$$

Having done some analytical transformations we receive:

$$\overline{P}_{je}(t) = 1 - (P(\gamma_{2j} \leq t) - P(\gamma_{2j} \leq \rho_{2j} \wedge t)) = 1 - I_{\rho_{2j} < t}(P(\gamma_{2j} \leq t) - P(\gamma_{2j} \leq \rho_{2j}))$$

The probability that on the interval $[0, t]$ a server will not transfer into the accident conditions,

$$\overline{P}_{ac}(t) = 1 - P(\rho_3 \wedge t < \delta \wedge \gamma_3 \leq t).$$

Let's do some transformations:

$$\overline{P}_{ac}(t) = 1 - (P(\delta \wedge \gamma_3 \leq t) - P(\delta \wedge \gamma_3 \leq \rho_3 \wedge t)) = 1 - I_{\rho_3 < t}(P(\delta \wedge \gamma_3 \leq t) - P(\delta \wedge \gamma_3 \leq \rho_3))$$

Then the probability that on the interval $[0, t]$ the accident will occur could be written as follows:

$$P_a(t) = 1 - M \left(\left(\prod_{j=1}^{N+1} \bar{P}_{j\varepsilon}(t) \right) \left(\prod_{i=1}^{N+1} \bar{P}_{aki}(t) \right) \bar{P}_{ac}(t) \right).$$

Due to statistical independence of failures this expression could be written as follows:

$$P_a(t) = 1 - \left(\prod_{j=1}^{N+1} M\bar{P}_{j\varepsilon}(t) \right) \times \left(\prod_{i=1}^N M\bar{P}_{aki}(t) \right) M\bar{P}_{ac}(t). \quad (6)$$

Having omitted intermediate transformations let's write down the final correlations:

$$M\bar{P}_{aki}(t) = 1 - \left(MF_{\psi_i}(\gamma_{1i}) \int_0^t F_{\rho_{1i}}(y) dF_{\psi_i}(y) + MF_{\gamma_{1i}}(\psi_i) \int_0^t F_{\rho_{1i}}(y) dF_{\gamma_{1i}}(y) \right); \quad (7)$$

$$M\bar{P}_{j\varepsilon}(t) = 1 - \int_0^t F_{\rho_{2j}}(y) dF_{\gamma_{2j}}(y); \quad (8)$$

$$M\bar{P}_{ac}(t) = 1 - \left(MF_{\delta}(\gamma_3) \int_0^t F_{\rho_2}(y) dF_{\delta}(y) + MF_{\gamma_3}(\delta) \int_0^t F_{\rho_2}(y) dF_{\gamma_3}(y) \right). \quad (9)$$

Having substituted (7)-(9) into (6), we receive:

$$P_a(t) = 1 - \left(\prod_{i=1}^N \left(1 - \left(MF_{\psi_i}(\gamma_{1i}) \int_0^t F_{\rho_{1i}}(y) dF_{\psi_i}(y) + MF_{\gamma_{1i}}(\psi_i) \int_0^t F_{\rho_{1i}}(y) dF_{\gamma_{1i}}(y) \right) \right) \right) \times \quad (12)$$

$$\times \left(\prod_{j=1}^{N+1} \left(1 - \int_0^t F_{\rho_{2j}}(y) dF_{\gamma_{2j}}(y) \right) \right) \times \left(1 - \left(MF_{\delta}(\gamma_3) \int_0^t F_{\rho_2}(y) dF_{\delta}(y) + MF_{\gamma_3}(\delta) \int_0^t F_{\rho_2}(y) dF_{\gamma_3}(y) \right) \right).$$

Let $\sigma_a(t)$ – intensity of accidents. Then

$$\sigma_a(t) = \frac{dP_a(t) / dt}{1 - P_a(t)}.$$

So, we have received the expressions for calculation the accident probability and intensity of the distributed computer system on the time interval $[0, t]$.

The fact that reliability indicators were received without any suppositions about a number of clients in the system and about laws about operation time for failure allows us to speak about accuracy of results.

4. Conclusion

The analysis of the distributed computer systems modeled as a set of services provided by a server to a client processes is an important problem. Collection and data storage are expensive procedures as data often cost less than a distributed computer system they are processed on. The proposed procedure of reliability analysis of distributed computer systems allows us even at the design stage to avoid unnecessary duplication of data (to prevent their loss due to the unreliability of the system) and to avoid additional efforts and financial expenses.

The proposed models can be used in modern technologies of the distributed computer systems development critical about security. The considered in the paper indicators applied in the architectural reliability of distributed computer systems analysis may be adjusted as the

most appropriate indicators for a particular system are determined depending on the system type and subject field of knowledge. Furthermore different indicators are applied for different systems.

In the result we can make the conclusion that the analysis model of the distributed computer systems of the architecture “client-server” allows us, unlike the existing models and methods of reliability estimation, without any assumptions about the laws of random variables distribution and a number of elements in the system to obtain expressions for the reliability parameters of the distributed computer system.

The given model instead distribution functions expressed in analytical form allows us to use their statistical equivalents found experimentally; that is especially helpful at the reliability calculation of the distributed computer systems.

The reliability analysis of the distributed computer systems of architecture “client-server” can be applied to assess the systems reliability for any possible architectural changes and reliable architecture election from some alternatives as depending on a number and size of the components the conditional and unconditional probability of failure, access, analysis and recovery time as well as the time of the various components use.

5. Acknowledgment

This work was supported by Russian Science Fund. The name of the project is “Distributed system development of storage, providing with and processing personalized multilingual information for the enterprises of the aerospace industry”.

References

- [1] Nikolai Matni, Yoke Peng Leong, Yuh Shyang Wang, Seungil You, Matanya B. Horowitz, John C. Doyle 2014 Resilience in Large Scale Distributed Systems *Procedia Computer Science* **28** pp. 285-293.
- [2] Kovalev I., Zelenkov P., Ognerubov S. 2013 The minimization of inter-module interface for the achievement of reliability of multi-version software *Proceedings of the 2013 International Conference on Systems, Control and Informatics* (Venice, Italy) pp. 186-189.
- [3] Anton V. Uzunov, Eduardo B. Fernandez 2014 An extensible pattern-based library and taxonomy of security threats for distributed systems *Computer Standards & Interfaces* Volume 36, Issue 4 pp. 734-747.
- [4] Tariq Omari, Greg Franks, Murray Woodside, Amy Pan 2007 Efficient performance models for layered server systems with replicated servers and parallel behavior *Journal of Systems and Software* Volume 80, Issue 4 pp. 510-527.
- [5] Berrade M.D., Scarf P.A., Cavalcante C.A.V., Dwight R.A. 2013 Imperfect inspection and replacement of a system with a defective state *A cost and reliability analysis, Reliability Engineering & System Safety* Volume 120 pp. 80-87.
- [6] Kovalev I.V., Polyanskiy K.V., Zelenkov P.V., Brezitskaya V.V., Sidorova G.A. 2013 System of search, analysis and processing of multilinguistic texts, integrated with data retrieval systems *Vestnik SibGAU* **1(47)** pp. 48-52.
- [7] Raspopin N.A., Karasyova M.V., Zelenkov P.V., Kayukov E.V., Kovalev I.V. 2012 Models and methods of data collecting and processing *Vestnik SibGAU* **2** pp. 69-72.
- [8] Zelenkov P.V., Kovalev I.V., Karaseva M.V., Rogov S.V. 2008 Multilingual model of the distributed system on the thesaurus basis *Vestnik SibGAU* **1** pp. 26-28.

- [9] Waseem Ahmed, Yong Wei Wu 2013 A survey on reliability in distributed systems *Journal of Computer and System Sciences* Volume 79, Issue 8 pp. 1243-1255.
- [10] Xiaoyong Tang, Kenli Li, Meikang Qiu, Edwin H.-M. Sha 2012 A hierarchical reliability-driven scheduling algorithm in grid systems *Journal of Parallel and Distributed Computer* Volume 72, Issue 4 pp. 525-535
- [11] Vařeková P., Vařeková I., Černá I. 2010 Automated Computer of the Maximal Number of Handled Clients for Client-Server Systems *Electronic Notes in Theoretical Computer Science* Volume 260 pp. 243-259.
- [12] Gregory Levitin, Liudong Xing, Suprasad V. Amari, Yuanshun Dai 2013 Reliability of non-repairable phased-mission systems with propagated failures *Reliability Engineering & System Safety* Volume 119 pp. 218-228.
- [13] Changyou Zheng, Xiaoming Liu, Song Huang, Yi Yao 2011 A parameter estimation method for software reliability models *Procedia engineering* Volume 15 pp.3477-3481.
- [14] Cong Jin, Shu-Wei Jin 2014 Software reliability prediction model based on support vector regression with improved estimation of distribution algorithms *Applied Soft Computer* Volume 15 pp. 113-120.