The graph-based analysis of structural delays in distributed multiprogram systems of information processing

E R Muntyan¹, E V Melnik²

¹ Institute of Computer Technologies and Information Security, Southern Federal University, 44, Nekrasovskiy Av., 347928, Taganrog, Russian Federation
² Federal Research Centre, Southern Scientific Centre of the Russian Academy of Sciences, 41, Chekhov Street, 344006, Rostov-on-Don, Russian Federation

E-mail: ermuntyan@sfedu.ru, evml7@mail.ru

Abstract. The paper discusses the issues related to the fault-tolerant computing systems design in terms of the structure and redundancy. It is shown that usage of prospective reservation methods significantly increases a complexity of the system design procedure. This is caused by the growth of the system configuration cases number because of the system redundancy and the need to test as many configurations as possible under the conditions of failures. To reduce the single configuration analysis complexity, an approach based on graphs with multiple edges such as vector allowing combining several edges of different types is proposed. Simultaneously, models based on such graphs allow representing the computer system structure, taking into account multiprogram data processing mode, and significantly reducing the calculation time of basic characteristics.

1. Introduction
Dependability is an important characteristic for a wide range of computing systems (CS), which are used in some failure-critical areas for solving complex tasks. The improvement of CS dependability can be achieved using highly reliable components, facilitating their operation mode and fault tolerance implementation.

The fault-tolerant CSs are in the focus of this paper. Let us consider CS as follows: it consists of n equal processor nodes (PN), which solve the computational task. The computational task consists of t programs, which are interconnected by the data transfer channels. Each i-th processor node PNᵢ (i = (1..n)) can operate in multiprogram mode, in which several programs of the task are being executed in it. There is at least one data transfer link between each pair of PN, including links with transit transfer through other nodes.

The following is quite important at the system design stage:
- determination of the structure, including the number of nodes and the number of direct data exchange links per node;
- provision of the appropriate level of characteristics that affect the CS fault tolerance and survivability, including:
  a) the structural survivability,
  b) the structural commutation,
  c) the structural delays in the data transfer between nodes.
To provide the fault tolerance and survivability of the system, it is necessary to parry failures by means of system redundancy. The structure of the CS that meets the above mentioned requirements will be considered optimal in the case of minimal redundancy in terms of additional PN and communication channels. The problem of synthesizing the optimal structure implementing a single-program nodes operation mode was successfully solved by calculating the characteristics of the graph [12]. However, in the case of a multi-program mode, the complexity of this task increases dramatically, and it determines the relevance of the study. To solve the problem, it is advisable to use modified graph approaches.

Various methods of structural redundancy are widely used [1, 2, 4]; however, in [3] a performance redundancy method that allows one to improve dependability characteristics (in particular, gamma-percentile operating time to failure) was proposed. The essence of this method lies in the fact that the system doesn’t have specially spare PNs, all nodes of CS take part in the task solving. In the case of PN failure, the programs from the failed PNs are deployed on the redundant resources of the operating ones. The absence of dedicated redundant nodes (such as with rolling redundancy [4]) allows one to distribute the load among the PNs evenly, which, in turn, due to the nonlinear dependence of the control components failure rate on the temperature (temperature depends on the load), allows one to improve the gamma percentile operating time to failure for CS. Thus, the performance redundancy along with the implementation of fault tolerance allows one to establish a rational load of CS components in order to increase the dependability. However, it requires solving the load allocation problem for nodes operating in multi-program mode, which complicates the forming of workable configurations in the case of failures, for example, in comparison with rolling redundancy. It requires the development of appropriate models, methods and algorithms, including those dedicated to study of the CS structure.

To study the structure properties that affect the CS dependability, it is expedient to use graph models. In particular, to solve such problem, graph models with the same type of non-oriented edges have been proposed [5, 6], where the vertices of the model act as CS nodes, and the data transfer links between the nodes correspond to the graph edges. Hereinafter a graph model with the non-oriented edges of the same type is considered as model 1.

However, such model allows one to study the interaction of nodes operating in a single-program mode. For nodes operating in a multiprogram mode and performance redundancy there is a need to use graphs with edges of different types, for example, described in [7, 8]. Such graphs allow one to specify heterogeneous vertices, heterogeneous, including multiple, edges between vertices (hereinafter we will call model 2). At the same time, the edges number increase leads to the computations planning time growth for the mentioned model.

In order to reduce the calculating time of characteristics of the graphs, which have edges of different types, it is proposed to use multiple edges such as vector, each of which allows one to combine several heterogeneous edges. In [9], a graph model is used to represent objects of complex technical systems, which takes into account homogeneous, heterogeneous, and multiple edges such as vector (hereinafter we will call model 3. Formally, such graph is given in accordance with (1):

\[ G' = (G'v, G'e), \]

where \( G'v = \{ gv_i | i = 1, 2, ..., n \} \), \( G'e = \{ ge_j | j = 1, 2, ..., m \} \) are, respectively, finite sets of vertices with \( gv_i \) (graph vertex) attributes and edges with \( ge_j \) (graph edge) attributes.

The vertices of the graph can be homogeneous and heterogeneous. Addressing to the graph vertices and edges is performed through their identifiers. The weighting coefficients can act as the vertices attributes, and the weights, orientation, and type as the edges attributes.

The edges can be oriented \((gv_i \rightarrow gv_j\) edges) or non-oriented \((gv_i \rightarrow gv_j\) edges), and also can be of the same type \((gv_i \rightarrow gv_j\) and \(gv_i \rightarrow gv_j\) edges), of different types \((gv_i \rightarrow tp \rightarrow gv_j\) and \(gv_i \rightarrow tp \rightarrow gv_j\) edges, where \( tp \rightarrow type \)) and multiple such as a vector \((gv_i \rightarrow \nu \rightarrow gv_j\) and \(gv_i \rightarrow \nu \rightarrow gv_j\) edges). An edge such as a vector allows one to combine \( t \) different types of edges and is represented in the form of \( \nu = \langle v_1, v_2, ..., v_t \rangle \), where \( t \) is the dimension of the vector or the multiplicity of the edge \( v \).

The weight of the vertex \((\nu \in [0, 1])\) and the edge \((\mu \in [0, 1])\) in the graph may give the meaning of
a certainty measure or the belonging degree to a fuzzy set. Degrees of affiliation are presented in the lists defining the graph. They can be determined by an expert or calculated through fuzzification [10].

Let us consider the possibilities and expedience of 1-3 models usage to represent the system structure. Firstly, the general requirements for graph models are formulated depending on the subject area tasks. Modern CS are characterized by a certain unification, both in terms of computing resources and in relation to data transfer links. To implement the calculations in the CS, usually the same processor nodes are used. Data transfer links between nodes are usually bi-directional. Consequently, the graphs representing CS must also have a certain unification degree. For example, in [3, 5, 12], the use of homogeneous (regular or circulant) graphs was proposed, i.e. graphs with vertices whose local degrees coincide: \( \forall i \neq j: s_i = s_j \). It is advisable to specify such graphs by a set of vertices of the same type and a set of unoriented edges.

Calculation of some graph characteristics allows one to determine a number of CS parameters. For example, structural delays in the data transmission between system nodes are estimated by the diameter and the average diameter of the graph. In [11, 12, 13], the improvement was noted in the structural survivability and structural commutability of the system, provided that the diameter and the average diameter of the graph were minimized. Even a slight average diameter decrease reduces the average data transfer time significantly. It is emphasized that structures with a minimum average diameter are optimal for implementing decentralized self-diagnosis algorithms in CS. In addition, at the system design stage, the need to minimize the graph diameter and / or an average diameter is noted, which is due to the need to reduce the number of inter-node data transfers at the task solving stage [12]. Therefore, to increase the dependability, it is important to find system structures with a minimal diameter and an average diameter of the graph. To solve this problem, some approaches were proposed in the works of a number of scientists [11, 12, 14]. Therefore, the questions of determining the system structure with minimal metric characteristics are beyond this article scope.

Thus, the task of multiple calculations of the diameter and the average diameter of the graph arises at the system design stage, as well as the same task takes place for CS fault-tolerance providing due to changes in the structure. This leads to the need to reduce time costs. In addition, non-standard graph models usage to provide a multiprogram mode of PN operation forces researchers to develop new or adapt existing methods for determining such characteristics.

Further, the authors propose approaches for determining the characteristics of a graph with multiple and heterogeneous relationships in order to study the structure properties that affect the CS dependability.

2. The description of specified graph characteristics

Some metric characteristics of an undirected graph with homogeneous connections \( G \) are defined below. Possible interpretation for representing a number of CS parameters is given too. The graph diameter \( d(G) \) is the maximum distance of all shortest paths between vertices pairs [15] and it can be determined using standard way:

\[
d(G) = \max_{i,j \in V} \{d(gv_i, gv_j)\}
\]  

(2)

where \( d(gv_i, gv_j) \) is the minimum distance between vertices \( gv_i \) and \( gv_j \).

The graph diameter is a quantitative characteristic for the maximum structural delays and corresponds to the maximum nodes number required for inter-node data transfer.

The average diameter of the graph \( d(G) \) in accordance with [12] can be calculated as:

\[
d(G) = \frac{\sum_{i \neq j \in V} d(gv_i, gv_j)}{n(n-1)}.
\]  

(3)

The average diameter allows one to estimate the average delay of inter-node interactions.

Obviously, both metrics depend on the distance between the graph vertices. The shortest path or distance between the vertices in an undirected graph with the same type of the edges \( d(gv_i, gv_j) \) is
considered as the smallest of all possible edges sequences $S_g = (\ldots, g_{e1}, g_{e2}, \ldots, g_{em}, \ldots)$ so as every two adjacent edges $g_{e1}$ and $g_{e2}$ have the same endpoint [15]. However, the question of determining the distance in a graph with multiple heterogeneous relationships arises.

The search for paths in such graph is possible for all edges types (1st case if the entire graph is examined) or for the selected type of edge (2nd case if a subgraph is examined), for example, for a given type of edge or for an edge in a vector with a certain index. The search for ways in the second case is expedient for the study of the selected edges that the researcher needs at the moment. The movement along the graph is carried out only on a given edges subset, while the remaining edges are temporarily ignored. Summarizing the above, we can formulate a criterion for calculating the shortest path for such graph: the shortest path in a graph with multiple heterogeneous edges is calculated taking into account the complete or truncated set of edges. Consequently, the graph metrics can be calculated for the entire graph or for each individual edges type depending on the domain problems requirements, it does not impose restrictions on (2) and (3) usage for the model specified in accordance with (1).

3. CS presentation possibilities on the graph base

Further, we will consider homogeneous graphs of degree $S(G)$ with vertices of the same type ($g_{v1}$) and undirected edges ($g_{e1}$) using models 1-3. For an example, let us consider a CS consisting of nine processor nodes $PN1 - PN9$, having communication links with four neighboring nodes. Modeling and calculation of the characteristics of the graphs discussed below were performed by means of the developed software [16].

Model 1. Figure 1 shows the graph $G1=(G1\text{'}v, G1\text{'}e)$, where $G1\text{'}v = \{g_{v1}, g_{v2}, \ldots, g_{v9}\}$ is the set of vertices, $G1\text{'}e = \{g_{e1}, g_{e2}, \ldots, g_{e18}\}$ is the set of the same type edges. The graph degree is $S(G1\text{'})=4$. In this model, each vertex $g_{v1}$ from the set $G1\text{'}v$ corresponds to $PNi$, and the weight of the vertex $\eta_i$ has the meaning of the node load degree. The edges between the vertices $g_{e1}$ correspond to the communication links between nodes, and the weight of the edges corresponds to the links load degree. This model reflects the least time-consuming case of calculating CS graph structural characteristics and is presented for comparison with the model proposed in the article (model 3). To evaluate the weights values of the vertices and edges of the graph $G1\text{'}$, a verbal description is introduced (table 1).

![Figure 1. Graph $G1\text{'}$ with homogenous edges.](image_url)
Graph $G'_1$ allows one to represent CS, consisting of $PN_1 - PN_9$, operating in a single-program mode. Let us consider two cases:

1) at the system design stage on which, in order to ensure survivability, performance redundancy is provided;

2) at the reconfiguration stage of CS operating at the limit of capabilities (when there is no reserve of PN performance).

| Table 1. Vertices and edges weights value evaluation. |
|-----------------------------------------------|
| Possible verbal degree description | Numerical weights values range |
|------------------|-----------------------------|
| Absent           | 0                           |
| Weak             | [0.1; 0.4]                  |
| Moderate         | [0.41; 0.7]                 |
| Strong           | [0.71; 1]                   |

Case 1. In the graph $G'_1$, the vertices and edges weights take the following values: $\eta_i = 0.7$, $\mu_j = 0.7$. Such graph characteristics values correspond to a moderate degree of nodes and communication channels loading. Then the diameter of the graph is $d_1(G'_1) = 1.4$, the average diameter of the graph is $\bar{d}_1(G'_1) = 1.05$. Thus, at the system design stage, a node performance reserve of $\approx 30\%$ is provided, and communication links are generally underloaded by about $30\%$. At the same time, the maximum structural delay for the inter-node data exchange corresponds to the value “1.4”, and the average delay in inter-node interactions corresponds to the value “1.05”.

Case 2. If at a certain moment the system doesn’t have a node performance reserve and communication links are loaded as much as possible, then this can be reflected in the model using certain weighting coefficients values. Then, in the graph $G'_1$, the vertices and edges weights take the following values: $\eta_i = 1$, $\mu_j = 1$. As a result, we obtain the values of the graph diameter $d_2(G'_1) = 2.0$ and the graph average diameter $\bar{d}_2(G'_1) = 1.33$. This means the maximum and average structural delays of the inter-node interaction at the peak of performance are determined. The time taken to calculate the shortest paths table and determine the metric characteristics in the PN for the graph $G'_1$ is 6 ms.

Model 2. The graph $G'_2 = (G'_2, G'_e)$ is defined by the vertices set $G'_2 = \{g_{v1}, g_{v2}, \ldots, g_{v9}\}$ and the heterogeneous edges set $G'_e = \{g_{e1}, g_{e2}, \ldots, g_{e34}\}$. The degree of the graph is $S(G'_2) = 4$. In such model, each vertex $g_{vi}$ from the set $G'_2$ corresponds to $PN_i$, and the vertex weight $g_{vi}$ corresponds to the node load degree. In model 2, pairs of vertices are connected by $tp$-fold edges of different types, where $tp$ corresponds to the possible number of programs running in PN. Let’s explain the purpose of the heterogeneous edges using an example. If 3 programs can be executed on each node in parallel, then $tp = 3$ data transfer channels should provide the calculation results transfer. Then, taking into account the above, for this model each edge from the graph $G'_1$ is actually replaced by three edges of different type. Taking into account the multiprogram mode, the heterogeneous edges between the vertices $g_{ej}$ correspond to the programs data transfer channels, and the edge weight corresponds to each channel load degree. The verbal description presented in Table 1 also can be used to estimate the graph $G'_2$ vertices and edges weights values.

Further, we will assume that the information-related programs are located on the PNs having a direct communication links; therefore, there is no transit data transmission through the nodes. If necessary, such a transmission can be reflected taking into account the utilities for transmitting transit data in each of the nodes. In this case, there will be one more data transfer channel.

Graph $G'_2$, unlike $G'_1$, allows one to represent nodes operating in the multiprogram mode. However, the drawback of model 2 compared to model 1 is the need of multiple edges in the graph usage in accordance with the multi-program mode of PN operation and the CS fault tolerance requirements. So, the number of edges in graph $G'_2$ is three times larger compared to $G'_1$; therefore the
time spent on calculations increases. For example, the time taken to calculate the shortest paths table and metric characteristics in the PN for the graph $G_2$ is 36 ms.

To solve this problem, the usage of graphs with multiple edges such as vector is proposed [9]. Each multiple edge such as a vector is a union of several different type of edges. Model 3 provides such opportunities.

Model 3. If no more than 3, for example, programs can be launched in each node, then, taking into account multiple edges such as vectors, the given CS model can be represented as a graph $G_3 = (G_3v, G_3e)$, where $G_3v = \{gv_1, gv_2, \ldots, gv_9\}$ is the set of vertices, $G_3e = \{ge_1, ge_2, \ldots, ge_{18}\}$ is the set of edges defined by the vector $v = \langle v_1, v_2, v_3 \rangle$ (Figure 2). The degree of the graph is $S(G_3) = 4$.

![Figure 2. Graph $G_3$ with relationships defined by the vector.](image)

The edges inside the vector have their own identification and separate weight, which corresponds to the channel load when task is performed. The verbal description presented in Table 1 is used to evaluate the vertices and edges weights values for the graph $G_3$.

The proposed approach novelty is the ability to determine the metric characteristics for a subgraph taking into account a certain connection type, for example, an edge according to the vector $v_1$ or $v_2$ or $v_3$. This corresponds to the criterion for calculating the shortest path for graphs with multiple edges of different types, denoted in section 2.

Let us consider cases 1 and 2, similarly to the graph $G_1$.

Case 1. At the system design stage, in graph $G_3$, the vertex weights are $\eta_i = 0.7$ that corresponds to a moderate degree of the system nodes load (performance reserve is about 30%). The edges weights in the vector $v_1 \mu_{j1} = 0$, and in the vectors $v_2$ and $v_3$ take different values corresponding to a strong degree of the data transfer channel load, for example, $\mu_{j2} = 0.8, \mu_{j3} = 1.0$. It is obvious the data transfer channel corresponding to the vector $v_1$ is not loaded, i.e. in the full hardware configuration, one program is not running, respectively, one data transfer channel is also not involved. In this case, the node performance reserve and a free data transfer channel can be used to execute programs (subtasks) in case of a failure of no more than 30% of the CS computing and communication resource.

The graph diameter and the average diameter characteristics for each type of edge in the vector are obtained separately (see table 2). An analysis of the graph metrics calculation results allows one to conclude that at the design stage the system has a backup (unloaded) data transfer link that corresponds to edges from $v_1$ of the graph $G_3$. At the same time, the PN performance reserve is about 30%. 

![Figure 2. Graph $G_3$ with relationships defined by the vector.](image)
Case 2. As a result of two nodes failure, the available computing reserve was used. Now, the vertices and edges weights in the graph $G'$ take the following values: $\eta_i = 1$, $\mu_{j.1} = 0.8$, $\mu_{j.2} = 0.9$, $\mu_{j.3} = 1.0$. The metrics of such a graph are presented in Table 2. As a result of the current graph characteristics analysis, we can conclude the system doesn't have a node performance reserve, and communication links load corresponds to a strong degree. The advantage of the proposed model 3 lies in the possibility of CS representing in the multiprogram mode case, as in model 2, while maintaining the graph edges number, as in model 1. At the same time, the calculation of shortest paths and metric characteristics in the PM for graph $G'_{3}$ is 12ms.

**Table 2.** Graph $G'_{3}$ metric characteristics for individual subsets of edges.

| Accounting for the edge vector $v_1/v_2/v_3$ index | Vertices weight numerical values, $\eta_i$ | Edges weight numerical values, $\mu_j$ | Diameter numerical values, $d(G')$ | Average diameter numerical values, $d(G_i)$ |
|--------------------------------------------------|------------------------------------------|---------------------------------------|-----------------------------------|-------------------------------------|
| Case 1:                                          |                                          |                                       |                                   |                                     |
| $v_1$                                            | 0.7                                      | 0                                     | 0                                 | 0                                   |
| $v_2$                                            | 0.7                                      | 0.8                                   | 1.6                               | 1.06                                |
| $v_3$                                            | 0.7                                      | 1.0                                   | 2.0                               | 1.33                                |
| Case 2:                                          |                                          |                                       |                                   |                                     |
| $v_1$                                            | 1.0                                      | 0.9                                   | 1.8                               | 1.20                                |
| $v_2$                                            | 1.0                                      | 0.8                                   | 1.6                               | 1.06                                |
| $v_3$                                            | 1.0                                      | 1.0                                   | 2.0                               | 1.33                                |

4. The models based on results of graphs experimental studies

For experiments on graphs, models 2 and 3 were selected, since they allow setting the multiprogram mode of nodes operation in the system. For each model 10 graphs of various dimensions (10-100 vertices) were used. Studied graphs metrics calculation was performed by means of the developed software.

The graph of the metric characteristics calculation time dependence, including the formation of the shortest paths table, on the size of the graph in models 2 and 3 in Figure 3 is presented. Analysis of the experimental studies on graphs results obtained by means of developed software shows the advantages of the model 3 using multiple edges such as vectors. For example, for graphs 10-100 vertices in size, the time to find the shortest paths and calculate metrics for model 3 is 3.4-5.6 times less than for model 2.

![Figure 3](image_url)

**Figure 3.** Graph of the dependence of metric characteristics calculation time on the size of the graph in models 2 and 3.
5. Conclusion
The graph model with multiple edges such as vectors for representing nodes and communication links in homogeneous computing systems supporting the multi-program mode is proposed. Such model allows one to reduce time costs at the design stage of the structure and configurations of fault-tolerant CS based on the PN performance and communication channels load reserve. In particular, it allows one to measure structural delays in the data transmission that is necessary to solve the problem of determining model metrics, including the graph maximum and average diameter. The article focuses on the features of the calculating metric characteristics process in the proposed graphs.

For graphs with multiple edges such as vectors a criterion for calculating the shortest path is formulated. The proposed approach novelty is the ability of calculating the graph metric characteristics taking into account the edges selected subset.

The authors conducted experimental studies on graph models that allow one to represent the multiprogram mode of CS nodes operation. The results of the experiments provided by means of the developed software showed the advantages of the proposed model through the multiple edges such as vectors usage. For graphs with the size of 10-100 vertices, the time of search for shortest paths and calculation of metrics in the proposed model is reduced by 3.4-5.6 compared to another model that supports the multi-program mode.

6. Acknowledgments
The reported study was funded by RFBR projects 18-05-80092, 19-08-00152.

References
[1] Avizienis A, Laprie J C, Randell B, Landwehr C 2004 Basic concepts and taxonomy of dependable and secure computing IEEE Trans. Dependable Secure Comput. 1 11–33
[2] Zhang Y, Jiang J 2008 Bibliographical review on reconfigurable fault-tolerant control systems Annual Reviews in Control 32(2) 229–252
[3] Kalyaev I, Melnik E, Korovin I, Klimenko A and Schaefer G 2018 7th Int. Conf. on Informatics, Electronics and Vision & 7th Int. Symp. in Computational Medical and Health Technology (Piscataway, New Jersey: IEEE) pp 582-587
[4] Akkaya M 2015 Fault tolerance Mechanisms in Distributed Systems Int. J. Communications, Network and System Sciences 8 478-482
[5] Kuznetsov A A, Kishkan V V 2018 The Cayley graphs of finite two-generator burnside groups of exponent 7 Siberian J of Science and Technology 19(2) 217–222
[6] Parkhomenko P P 2005 Shaped lattice graphs as models of the multiprocessor computer systems Automation and Remote Control 66 492–502
[7] Boccalettia S, Bianconic G, Criado R, Genio C I, Gómez-Gardeñes J, Romance M, Sendiña-Nadal I, Wang Z, Zanin M 2014 The structure and dynamics of multilayer networks Physics Reports 544(1) 1-122
[8] Contractor N S, Monge P R, Leonardi P M 2011 Multidimensional networks and the dynamics of sociomateriality: bringing technology inside the network Int. J. of Communication 5 682–720
[9] Muntyan E R 2019 Realization of fuzzy model of objects interaction inside a complex technical systems based on graphs Software & Systems 32(3) 411–418
[10] Rosenfeld A 1975 Fuzzy Sets and Their Applications ed Zadeh LA, Fu K S, Shimura M (New York: Academic Press) pp 77–95
[11] Sampes M 1996 Caylay graphs as interconnection networks: A case study Proc. of the Inter. Conf. Parsells 96 (Berlin:Akademie-Verlag) pp 97–106
[12] Monakhov O G, Monakhova E A 2000 Parametrically described regular semigroup interconnection networks for large-scale multicomputers Proc. Int. Symp. on Parallel Architectures, Algorithms and Networks (Piscataway, New Jersey: IEEE) pp 202–209
[13] Bermond J C, Comellas F, Hsu DF 1995 Distributed loop computer networks: a survey J. Parallel Distributed Comput. 24 2-10
[14] Bhavana M, Krishnan P 2019 On Index coding for Complementary Graphs with focus on Circular Perfect Graphs Proc. 2019 IEEE Int. Symp. on Information Theory (Piscataway, New Jersey: IEEE) pp 2319–2323

[15] Berge C 1964 The Theory of Graphs and its Applications (New York: Wiley)

[16] Muntyan E R 2018 Software module for representation of actors and relations between actors based on graphs Certificate of registration of the computer program of EC № 2018665499 (Moscow: Rospatent)