UDC 004.27+004.056
doi:10.31799/1684-8853-2022-1-19-29

Model of a distributed information system solving tasks with the required probability

V. V. Gryzunov[a]*, PhD, Tech., Assistant Professor, orcid.org/0000-0003-4866-217X, viv1313r@mail.ru*
[a]*Russian State Hydrometeorological University, 79, Voronejskaya St., 192007, Saint-Petersburg, Russian Federation*

Introduction: *Distributed in space-time Networks: IIoT and IoT, fog- and edge-computing tend to penetrate into all spheres of human activity. Enterprises, government, law enforcement agencies, etc. depend on the quality of those technologies.* ***Purpose:*** *To determine the composition of the Network that provides the required uptime probability.* ***Methods:*** *According to the concept of structural and functional synthesis, a distributed Network is presented as an unstable queuing system in which servicing devices are connected and disconnected at an arbitrary point in time. A simulation model of the Network has been built.* ***Results:*** *The state of the Network depends on the number of devices and tasks, their performance and lifetimes. The model does not use these quantities themselves, but their ratios. The values of the uptime probability of the Network are calculated for all possible combinations of ratios. The confidence interval has been calculated with a confidence level of 0.95. From the data obtained, it is clear: 1) what should be the minimum composition of the Network in order to provide the required probability; 2) what probability the current composition of the Network can provide; 3) what flow of tasks is admissible in order to solve tasks with the required probability. It is shown that the dependence of the mean tasks residence time on the Network on the composition of the Network has two inflection points. Using information about these points, the Network Management System forms pools of devices or increases the number of devices.* ***Discussion:*** *It is assumed that the Net has a fully connected structure. Consequently, for practical application, it is necessary: to expand the model with an adjacency matrix describing the connections between nodes, and hence the paths of propagation of tasks over the Network or consider that each node is a relay and is capable of transmitting the task to any other node on the Network. Overhead costs arising from this are taken into account by adjusting the original data.* ***Practical relevance:*** *The results allow minimizing costs in the design and operation of distributed systems, maximizing the likelihood of system uptime under given constraints for resource.*

Keywords *— fog computing, edge computing, distributed computing, failure-tolerant computing, QoS, availability.*

For citation: Gryzunov V. V. Model of a distributed information system solving tasks with the required probability. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2022, no. 1, pp. 19–29. doi:10.31799/1684-8853-2022-1-19-29

Introduction

Distributed computing IoT, IIoT (Industrial Internet of Things), edge, fog and others (hereinafter referred to as Networks) are integrated into the information systems of large enterprises: international meteorological data processing centers, control systems for operational-search activities [1], companies implementing Industry 4.0 standards, etc. On the one hand, the distribution of computations is an inevitable process [2], on the other hand, the welfare of humanity depends on whether the distributed Network can guarantee the solution of the tasks, that is, whether the Network ensures the required quality.

“Quality is a property or a set of properties of an object that determine its compliance with the purpose”.

Existing research model the Network differently and evaluate the quality of the Network differently.

At the same time, the Network is characterized by uncertainty associated: 1) with the tasks being solved, 2) with the structure of the Network. This uncertainty has a different nature: 1) stochastic, 2) non-stochastic, i. e. one that cannot be described using models and methods of probability theory, for which all alternatives or a probabilistic description are not known.

Existing studies model the Network differently, and measure the quality of the Network differently and struggle against uncertainties.

Research overview

A number of authors approach this problem by minimizing the time it takes to solve tasks by the Network. Only the uncertainty of the tasks to be solved is touched upon.

For example, researchers [3] reduce the time by using a special algorithm for randomly sending tasks between nodes. The decision on whether to take a task to work or send it to another node is taken independently by the Network node. It is assumed that if a node has accepted a task for execution, then it will necessarily execute it or send it to the one who will execute it. Cases with sudden failures or disconnection of nodes from the Network are ignored.

The authors [4] use the delay time of the Network’s response to a service request by the user. It is proposed to reduce latency by replicating data in a distributed geographic information system. If the Network is large, then the iFogStorP method must be used, otherwise the iFogStor. It is believed that the probability of solving tasks by the Network is 100%.

Researchers in [5] went further. They have developed mathematical models of the Network as a queuing system (QS) (Multiple Input Multiple Output) and propose to reduce the time to receive a service by changing the discipline of service. According to the results of the study, it can be seen that the minimum residence time on the Network is achieved when using a strategy that chooses to serve a subset of users with the maximum bandwidth. And in some cases, a strategy that organizes services in the order in which tasks are received. Denial of service is provided only if the queue does not reach the task.

The next group of works is focused on eliminating the structure uncertainty of a stochastic nature. The quality of service of the Network estimate through the fault tolerance of the Network. Fault tolerance is increased by introducing redundancy in the form of backups or clustering.

Research [6] concerns fault-tolerant energy-efficient routing in a distributed system based on IoT sensors. Clusters are created on the basis of two groups of nodes: normal and powerful. The powerful nodes control the load and activity of the normal nodes. The number of Network nodes is determined at the initial moment of time and does not increase during operation. The duration of solving tasks is ignored. The study concludes: the higher the network fault tolerance is required, the more nodes are needed; the higher the probability of node failure, the more nodes are needed to provide the required fault tolerance. However, the study has no data on how many nodes are needed.

On the other hand, the more nodes, the more overhead, the more network management overhead, the more difficult it is to balance the load. A similar problem was discovered by the authors of [7].

Overhead costs and load balancing to improve fault tolerance by optimizing the probability of failure-free operation are considered by the authors in [8]. The authors conclude that for an information and control system processing distributed data, a distributed structure is better suited (in terms of the uptime probability). At the same time, the authors' model considers the "fog layer" as a single device and takes into account the possibility of its failure only due to overheating caused by a large computational load. The possibility of restoring or changing the probabilistic characteristics of the "fog layer" during the operation period is not taken into account by the model.

The work [9] is devoted to the study of the dependence of the uptime probability on the relative speed of node recovery. The network is represented as a QS described by the Kolmogorov — Chapman system of differential equations. The dependences of the uptime probability of the system on the relative speed of recovery are obtained for various distributions of the repair time of nodes (Weibull — Gnedenko, Pareto and Lognormalnoe). However, the authors impose a rather strong restriction — they require that the system have a stochastic nature and, moreover, have a stationary probability distribution. In [10], the indicated restrictions are toughened by the requirement of the ordinariness of the flow of requests and the flow of refusals of nodes. Fault tolerance is a function of the network's mean response time to user requests.

The next line of works is exploring the possibility of improving the quality of service through compute scheduling, load balancing and routing.

The authors of [11] propose decentralized load balancing of heterogeneous nodes of the Network to increase the probability of solving tasks. Limitations adopted in the work: the structure of the Network is constant, the nodes solve the accepted tasks with a probability of 100%. Only the uncertainty of the tasks being solved is taken into account.

In [12], the load is controlled centrally based on templates. Templates are created in advance. Small variability of the structure of the Network is possible within the given templates. Tasks and nodes are classified according to their computational nature. Nodes accept tasks of their class.

In the articles studying the behavior of Mobile-edge Computing networks, some restrictions on the fixed structure of the Network are removed, the combinatorial complexity of load planning and the heterogeneity of nodes are taken into account [13]. In [14], researchers propose intelligent planning.

At the same time, the authors imply that the performance of devices and their number are predetermined, therefore, their research may be applicable only to some fairly stationary segments of the Network, such as clouds, but not fog, and even more so edge computing. The structure and functions of the Network, using edge computing, IoT, IIoT, are changing so much that the authors of the study [15] are forced to impose a ban on leaving node the Network until the node solves the task.

As can be seen from the analysis, the existing works operate with models that are not entirely adequate for the Network, do not fully take into account the uncertainty that is characteristic of the Network, namely: resource intensity of tasks; the ability of the nodes of the Network to arbitrarily connect and disconnect from the Network; options when a node takes a task, does not perform it, because it leaves the Network; situations with denial of service, when there are free nodes in the Network, but they refuse tasks, because their performance is not enough to solve the task.

A more adequate model of the Network will make it possible to more accurately predict the behavior of the Network in conditions of uncertainty and reasonably put forward requirements for its composition. To solve the indicated problem of modeling a distributed Network, solving the tasks with the required probability under conditions of strong uncertainty, it is necessary to formulate it, choose the appropriate indicator of the efficiency of the Network's functioning and methods of its calculation.

Strong uncertainty is understood as uncertainty associated with the tasks being solved and the structure of the Network and having a stochastic or non-stochastic nature.

"Efficiency is a property (quality) of the system functioning process, defined as its adaptability to solving the tasks set for the system" [16, p. 31].

Statement of the research problem

The formulation of the problem is based on the concept of structural and functional synthesis of systems, presented in [17] and the keep integrity law of an object, which states that there is a stable repeating connection between the properties of an object and its actions with a fixed purpose of the object:

$$I(Q) = F(Q, \Phi(R, U), t), \quad (1)$$

where:

I is an indicator of the effectiveness of the Network, the presentation of the required number of required characters at the required time (the number of simultaneously tracked targets, the number of web-portal users, the probability of image recognition, etc.);

Q — the set of required space-time states of the Network (Network model) is set by the metasystem / Network creator. Shows how exactly the elements of the Network should interact with each other and with the Network users. In the general case, it is a function of time and is set in various ways: by enumeration, analytically, by specifying characteristic properties, etc.;

Φ — the set of current space-time states of the Network, is a model of the current situation, shows how the elements of the Network interact with each other and the user of the Network in reality;

R is the set of capacities of the elements of the Network (model of actions of the elements of the system in space-time);

U is a set of control actions on the elements of the Network;

t is the Network operation time;

F is an operator expressing the basic laws of the existence of the Network.

All types of uncertainties of any nature inherent in an object are manifested through Q, R and Φ and are taken into account in expression (1).

It is advisable to begin the specification of expression (1) describing the Network with the formalization of the performance indicator (I).

Indicator of the effectiveness of the Network as a system operating in conditions of strong uncertainty

Systems with high uncertainty have their own performance indicators and methods of calculating them. The advantages and limitations of some indicators are analyzed in [18]. This study uses an indicator borrowed from [18] — the uptime probability of Network (UPN):

$$P = K^{*}/K, \quad (2)$$

where P — UPN; the indicator can be called a probability because it satisfies the corresponding axiomatics; K, K^{*} — the number of tasks assigned to the Network and performed by the Network respectively for the entire period of operation.

The indicator (P) is integral and uses only the number of tasks set and solved. The number of tasks set and their characteristics depend on the user's goals, the number of tasks solved depends on the characteristics of the tasks themselves, the available resources of the Network, their properties and configuration. Consequently, the indicator takes into account the uncertainty of any nature associated with both the tasks and the structure of the Network. And it can be called a probability, because it satisfies the corresponding axiomatics.

Axiom 1. The event consists in the solution of the i-th task by the Network. The solution by the Network of all assigned (K) tasks forms a complete group of events $\mathcal{F}$. An arbitrary system of subsets of the set $\mathcal{F}$ is closed under the operations of complement and union, and, therefore, is an algebra of events.

Axiom 2. Each i-th event, consisting in solving the i-th task, is associated with a non-negative real number K_i/K, where K_i is the same for all tasks and is equal to 1, therefore, each individual event is associated with the number $1/K$.

Axiom 3. $P(0) = 0/K = 0$, $P(K) = K/K = 1$, therefore $0 \le P \le 1$.

Axiom 4. For disjoint events $i \ne j$, $P(K_i) + P(K_j) = P(K_i + K_j)$.

It is not very convenient to calculate UPN in this way for a real Network, because it can be calculated only after the Network stops functioning. However, it is well applicable for studying the properties of the Network on the model.

In this case, the research problem is set as follows.

Verbal problem setting

Let's represent the Network as a set of QS $G/G/1$, that is, the laws of claims arrival and their servicing are arbitrary. The queue is endless. It is necessary to find the UPN of Network.

At first glance, the problem of specifying type of operator (F) in expression (1) looks like an ordinary problem of the theory of queuing, solved analytically. One of the typical works that solve such a problem [19] considers the Network as a set of QS $M/M/2$ with one reliable and one unreliable element, takes into account the variability of the Network structure and even delays in the dissemination of information on the Network about a failed device.

However, upon a more detailed study, it becomes clear that some details of real Networks cast doubt on the advisability of searching for expression (1) solutions in an analytical form. The most typical of these details:

— in a real Network, different tasks have different needs for the performance of service devices, including the minimum permissible performance of the processing device. In the QS, this is solved either by classifying claims, or by specifying a service flow that takes into account both the variability of the device in the form of available performance and various requirements of claims. Both are challenging in themselves;

— in the course of life on the Network, situations arise that are of a purposeful aggressive nature that cannot be described in terms of probability theory and mathematical statistics, for example, cyberattacks;

— nodes join the Network and leave it at an arbitrary moment in time;

— a node starts to solve a task and does not finish;

— a task arrives in the system through an arbitrary node of the Network.

There are ways to take into account the indicated features and obtain a solution in an analytical form: classification of claims and service devices, phase method, methods of working with devices with variable structure, etc. But the analytical solution is usually private with strong restrictions and assumptions, and the final system of equations is solved simply for 20–30 devices. And usually only numerically. The number of nodes in a real network can be hundreds of thousands of nodes.

Considering the above, it seems appropriate to solve the problem by drawing up a simulation model of the Network. For this, it is necessary to take into account the information and control connections of the tasks being solved.

Accounting for the information and control connections between tasks

As a rule, the tasks performed are linked by information and control links. And it can be assumed that the likelihood of completing one depends on the likelihood of completing the related task. Taking these dependencies into account greatly complicates the Network model. In the study [18], it was shown that for calculating the UPN, information and control dependencies can be neglected and the UPN can be calculated separately for each task:

"*Theorem*. Let a directed graph G be given, reflecting the dependencies between tasks. Then the events involved in solving tasks are independent for any directed graph.

Supposition. The considered technical systems are dynamic, therefore, the situation at the current time does not affect the situation at previous times (the situation at subsequent times does not affect the situation at the current times)".

Formal statement of the research problem

Performance is the number of tasks solved per unit of time.

Given:

1) T — Network lifetime (*MaxModelTime* in the model);

2) P^* — required UPN;

3) P — current UPN;

4) maximum available node performance (Ω_{node}) — describes the node's ability to provide its resources to the task;

5) the minimum performance required by the task (Ω_{task}) — the lower estimate of the node performance. If the performance of the node is less than the minimum required, then the node does not take the task, even if it is free: $\Omega_{node} \geq \Omega_{task}$;

6) maximum task execution time (T_{task}) — the time that cannot be exceeded by the node if the node has taken the task to work;

7) maximum lifetime of nodes (T_{node}) — describes the degradation of the Network, a node disappears from the Network if its residence time is greater than the maximum. If he solves the task, the task is considered lost;

8) maximum number of incoming tasks at each moment of time (A_{task}) — characterizes the load of the Network at each moment of time;

9) maximum number of nodes connecting to the Network at each moment of time (A_{node}) — describes

the increase in the resources of the Network at each moment of time.

Required:

get such a Network configuration for which the required UPN will be less than or equal to the current one:

$$P^* \leq P.$$

The simulation model was compiled and investigated in the MatLab language.

Network simulation model

Initial data

From the condition of the problem described above, it follows that the configuration of the Network is determined by nine values.

To simplify modeling, the model uses not the quantities themselves, but their ratios:

1) $CurrentTaskDuration = T_{task}/T_{node}$;

2) $CurrentTaskPerfomance = \Omega_{task}/\Omega_{node}$;

3) $CurrentRatio = A_{task}/A_{node}$.

Figure 1 has a simplified block diagram of the simulation model algorithm.

It can be seen from the given algorithm that the interaction of the same Network configuration with different task flows is simulated.

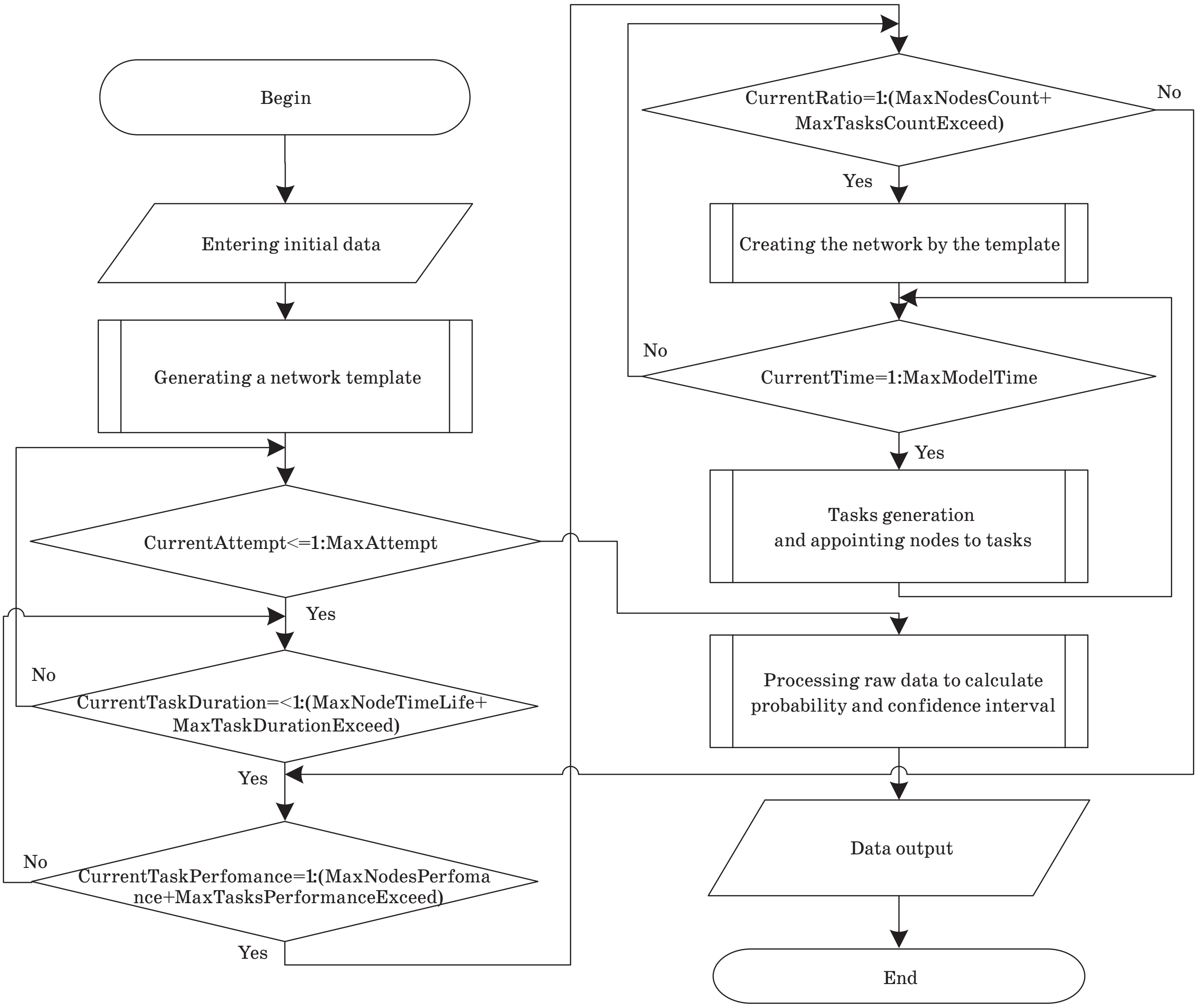
Network configurations are generated randomly according to a uniform distribution law with the following parameters:

$T_{node} \in [1; MaxNodeTimeLife]$;

$\Omega_{node} \in [1; MaxNodesPerfomance]$;

$A_{node} \in [1; MaxNodesCount]$.

The Network configuration is invariant. Relative to it, other values of the model, generated randomly according to the uniform distribution law, change:


Begin
Entering initial data
Generating a network template
CurrentAttempt<=1:MaxAttempt
Yes
No
CurrentTaskDuration=<1:(MaxNodeTimeLife+ MaxTaskDurationExceed)
Yes
No
CurrentTaskPerfomance=1:(MaxNodesPerfomance+MaxTasksPerformanceExceed)
Yes
CurrentRatio=1:(MaxNodesCount+ MaxTasksCountExceed)
No
Yes
Creating the network by the template
No
CurrentTime=1:MaxModelTime
Yes
Tasks generation and appointing nodes to tasks
Processing raw data to calculate probability and confidence interval
Data output
End

■ ***Fig. 1.*** Simplified block diagram of a simulation model

$T_{task} \in [1; (MaxNodeTimeLife+MaxTaskDurationExceed)]$;

$\Omega_{task} \in [1; (MaxNodesPerfomance+MaxTasksPerformanceExceed)]$;

$A_{task} \in [1; (MaxNodesCount+MaxTasksCountExceed)]$.

The required UPN (P^{*}) is also constant for all variations. The network lifetime (T) is set by the simulation time, is stored in the *MaxModelTime* variable and takes on a significantly larger value than other times in the model. This improves confidence in the simulation results.

Tasks appear and are executed at an arbitrary moment of the Network functioning, there are no deadlines, therefore, the resulting UPN is the upper estimate of the real UPN.

The task is lost if the node accepts the task and leaves the Network without completing the solution.

Of all the available nodes for solving the task, the node is selected whose maximum available performance is closest to the minimum performance required by the task. The model implements the FIFO (first input first output) algorithm, i. e. the task that entered the Network earlier is assigned first. Among the devices, the one that connected to the Network earlier is selected.

Real nodes of the Network have physical restrictions on the number of connections with other nodes, therefore, in the model, the value describing the number of nodes is random and bounded from above.

The minimum performance required by the i-th task ($\omega_{i,\ task} \in (0; \Omega_{task}]$), the time to solve the task when this productivity is obtained $t_{i,\ task} \in (0; T_{task}]$ and the maximum available performance of the j-th node ($\omega_{j,\ node} \in (0; \Omega_{node}]$) determine the time during which the node is considered busy and cannot accept other tasks$(t_{j,\ busy})$:

$$t_{j,\ busy} = t_{i,\ task}\omega_{i,\ task}/\omega_{j,\ node}.$$

Simulation results

Due to the large amount of data received, in Tables 1 and 2, only a part of them is shown for *CurrentTaskDuration* = 6/10. The rest of the data, if necessary, can be provided upon request. The mean UPN values for all possible Network configuration options are given in the Table 1. The confidence interval with a confidence level of 0.95 increases with the growth of the *CurrentTaskDuration*, *CurrentTaskPerfomance*, *CurrentRatio* ratios and lies in the interval [0.0013; 0.056]. An increase in the confidence interval is associated with an increase in the intervals in which the simulated values are located:

$T_{task} \in [1; (MaxNodeTimeLife+MaxTaskDurationExceed)]$;

$\Omega_{task} \in [1; (MaxNodesPerfomance+MaxTasksPerformanceExceed)]$;

$A_{task} \in [1; (MaxNodesCount+MaxTasksCountExceed)]$.

It can be seen from the figure that the smaller the task/node ratio, the higher the UPN. At the same time, the unit costs for one task increase [7]. However, the UPN is influenced not only by this ratio, but also by the nature of the tasks performed, therefore, the Network Management System can provide the required UPN, based on the duration of incoming tasks and their performance requirements.

The mean tasks residence time in the Network for *CurrentTimeRatio* = 6/10 is presented in the

■ ***Table 1.*** Mean UPN for all experiments performed, *CurrentTimeRatio* = 6/10

CurrentRatio	CurrentTaskPerfomance										
	1	2	3	4	5	6	7	8	9	10	11
1	0.717	0.677	0.622	0.599	0.566	0.520	0.517	0.524	0.548	0.476	0.437
2	0.771	0.704	0.672	0.623	0.598	0.583	0.554	0.562	0.547	0.487	0.465
3	0.786	0.727	0.695	0.671	0.629	0.603	0.582	0.593	0.577	0.526	0.466
4	0.821	0.760	0.709	0.688	0.669	0.618	0.638	0.607	0.611	0.527	0.491
5	0.828	0.773	0.721	0.688	0.662	0.662	0.630	0.625	0.624	0.550	0.480
6	0.843	0.777	0.748	0.709	0.682	0.668	0.654	0.646	0.631	0.569	0.516
7	0.835	0.794	0.757	0.728	0.697	0.674	0.660	0.648	0.651	0.580	0.507
8	0.859	0.799	0.756	0.717	0.716	0.690	0.680	0.669	0.656	0.602	0.527
9	0.864	0.805	0.777	0.753	0.718	0.702	0.692	0.686	0.675	0.595	0.536
10	0.869	0.822	0.782	0.764	0.728	0.717	0.707	0.693	0.672	0.605	0.537
11	0.868	0.826	0.793	0.751	0.737	0.720	0.709	0.696	0.684	0.606	0.544

Table 2. Times are calculated only for those tasks that were performed by the Network. Lost tasks or tasks that were not recruited for any reason were not counted.

The simulated Network was tested 30 times. The duration of the Network functioning is 100 units of model time. Fig. 2 contains a surface, inside which all configurations of the Network are located, which solves the assigned tasks with an UPN of at least 0.8.

Consequently, this simulation result can be interpreted as follows: the tasks entering the Network are solved with probability (see the Table 1). For tasks that are solved by the Network, they are on the Network on mean (see the Table 2) units of model time. Tasks are denied service for the following reasons:

1) the minimum required performance for the task is greater than the current performance of any node on the Network;

2) the node leaves the Network, taking the task and not finishing its solution. The reason for leaving is not important for the simulation results — it can be a failure of the software or hardware of the node, disconnection of the node from the Network, disabling the node during a cyber attack, etc.

Figure 3 shows the dependence of the mean tasks time on the Network on various Network configurations. The graph was built as follows:

1) each *CurrentTaskDuration* ratio has its own legend (line of the same color);

2) found the mean value for each row from the Table 2 (*MeanTimeForCurrentRatio*). This value shows the mean tasks residence time on the Network with a fixed *CurrentTaskPerfomance*;

3) found the mean value for each column from the Table 2 (*MeanTimeForCurrentTaskPerfomance*). This value shows the tasks residence time on the Network with a fixed *CurrentRatio*;

4) the difference between the values obtained in steps 2 and 3 was calculated: *MeanTimeFor CurrentRatio–CurrentTaskPerfomance* for each row-column pair;

5) a graph of the difference was built for each number of the pair from step 4.

The graph in Fig. 3 shows a pattern: there are clearly pronounced inflection points numbered 6 and 9. These points correspond to the 6/10 and 9/10 ratios for the *CurrentRatio* and *CurrentTaskPerfomance* ratios. It is interesting that through the same points the surface corresponding to UPN 0.8 (see Fig. 2) cuts the plane formed by the lines *CurrentRatio* and *CurrentTaskPerfomance*.

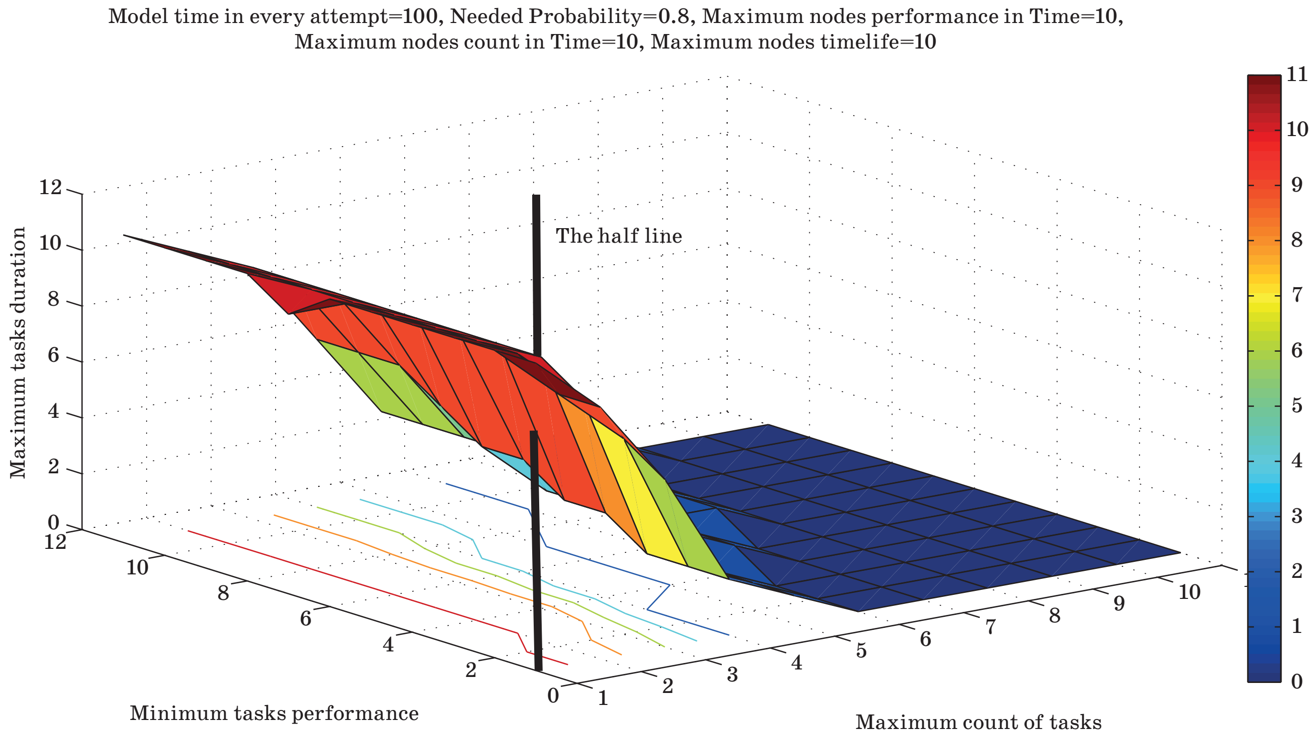
Figure 3 shows two distinct situations:

1. Graph in the positive half-plane. This means that with an increase in the number of tasks, the mean tasks residence time on the Network grows faster than with an increase in the minimum required performance. This happens as long as the *CurrentRatio* < 6/10.

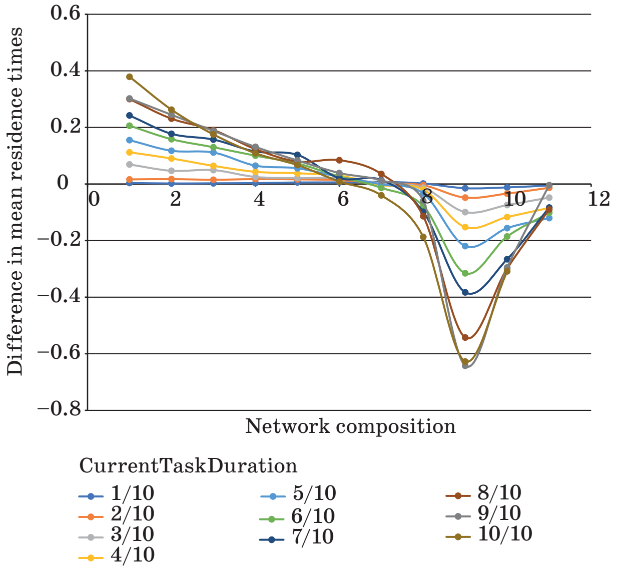
2. Graph in the negative half-plane. This means that with an increase in the number of tasks, the

■ ***Table 2.*** Mean residence time on the Network of tasks solved by the Network, *CurrentTimeRatio* = 6/10

The set number	The set number											*Mean Time For Current TaskPerfomance*
	1	2	3	4	5	6	7	8	9	10	11	
1	0.708	0.760	0.824	0.852	0.928	0.939	1.007	1.082	1.188	1.183	1.253	**0.97**
2	0.729	0.782	0.835	0.911	0.947	0.970	1.056	1.084	1.248	1.369	1.162	**1.01**
3	0.741	0.800	0.865	0.926	0.978	1.081	1.086	1.175	1.417	1.243	1.387	**1.06**
4	0.739	0.821	0.885	0.973	1.017	1.110	1.146	1.240	1.500	1.424	1.348	**1.11**
5	0.753	0.836	0.919	1.013	1.063	1.126	1.196	1.304	1.558	1.562	1.470	**1.16**
6	0.755	0.847	0.950	1.019	1.102	1.195	1.232	1.357	1.635	1.536	1.422	**1.19**
7	0.767	0.867	0.960	1.034	1.131	1.164	1.282	1.371	1.708	1.575	1.525	**1.22**
8	0.812	0.892	0.963	1.058	1.171	1.230	1.309	1.430	1.665	1.758	1.644	**1.27**
9	0.815	0.902	0.981	1.090	1.189	1.258	1.370	1.479	1.890	1.711	1.791	**1.32**
10	0.811	0.899	1.047	1.109	1.211	1.332	1.399	1.569	2.070	2.033	1.749	**1.38**
11	0.825	0.946	1.041	1.116	1.261	1.351	1.452	1.698	2.066	1.878	1.822	**1.41**
Mean Time For CurrentRatio	**0.77**	**0.85**	**0.93**	**1.01**	**1.09**	**1.16**	**1.23**	**1.34**	**1.63**	**1.57**	**1.51**	*Mean Time For CurrentRatio*
Difference in mean residence times	**0.21**	**0.16**	**0.13**	**0.10**	**0.07**	**0.03**	**–0.01**	**–0.08**	**–0.32**	**–0.19**	**–0.10**	*Difference in mean residence times*



■ ***Fig. 2.*** Network configuration solving the assigned tasks with a UPN of at least 0.8



■ ***Fig. 3.*** Dependence of the mean tasks residence time on the Network on various Network configurations

mean tasks residence time on the Network grows more slowly than with an increase in the minimum required performance. This happens when *CurrentRatio* > 7/10.

Hence it follows that with an increase in the load on the Network and *CurrentRatio* < 6/10, it is better to first of all create pools [20] from the existing nodes. Better in the sense of reducing the mean residence time on the Network while maintaining the required UPN. If the *CurrentRatio* > 7/10, it is better to focus on connecting new nodes.

In [21], a method is described, which can be used to use the time and accuracy errors admissible for tasks and to weaken the requirements for the Network, while maintaining the required UPN. In Fig. 2, this appears as a shift of the surface to the right along the ray outgoing from the origin. The beam tilt angles depend on the tolerances for the task.

The work [22] describes an example of patrolling of a naval base by unmanned aerial vehicles “Orlan”. As a result of information and technical impact, a situation arises when the available resource of the unmanned aerial vehicles is not enough to solve the assigned task of monitoring the perimeter, and the control system forms a pool with the required performance. A similar situation of resource scarcity and a decrease in UPN occurs with an increase in the number of tasks or their resource intensity. Using the data from the proposed study, we can say that in order to achieve the required UPN, it is necessary to keep the ratios *CurrentTaskDuration*, *CurrentTaskPerfomance*, *CurrentRatio* in the specified range. This is achieved either by forming resource pools or by using task reserves in terms of time and accuracy [21].

Directions for further research

The developed model assumes that the Network has a full-mesh topology. In the future, it is supposed to use the adjacency matrix and study the behavior of Networks with different topologies, compare the work of the method offered in [20] with others.

It is considered that the node that accepts the task solves it with 100% probability. The reliability of the nodes themselves can be taken into account either by creating a special module that thins the flow of nodes according to the required pattern, or by making changes to the initial data, on the basis of which the Network model is generated.

Events consisting in the appearance of tasks and the connection of nodes to the Network and disconnection of nodes from it are random in the presented model. This is well suited to describe normal operation: solving planned tasks, failures and failures caused by natural causes. But this does not fully reflect the targeted aggressive effects on the Network: cyberattacks, information technology intrusions, a surge in tasks in case of emergencies, etc.

Conclusion

With the help of the approach proposed in the article, a significant step has been taken in concretizing the expression for the structural and functional synthesis of the Network (1), namely: such characteristics of a distributed Network as the UPN and the average time spent on the Network were formulated and studied through the simulation model; the variables on which the state of the Network depends were indicated. The number of variables has been reduced almost twice by introducing the ratios *CurrentTaskDuration*, *CurrentTaskPerfomance*, *CurrentRatio*; shown, that with further concretization of expression (1) Information-control dependencies between tasks can be neglected.

Each required UPN of Network is matched its own surface. To ensure the solution of task with the required UPN, the Network control system needs to keep the values of the *CurrentTaskDuration*, *CurrentTaskPerfomance*, *CurrentRatio* ratios under this surface.

The graph describing the behavior of the mean residence time of tasks on the Network has two distinct inflection points. Knowledge of these points helps the control system to make a decision on the choice of an acceptable ratio of *CurrentTaskDuration*, *CurrentTaskPerfomance*, *CurrentRatio*.

By simple modifications of the model, it is possible to study the behavior of the Network under conditions of non-stochastic uncertainty, which cannot be described in terms of probability theory: cyberattacks, information technology intrusions, a surge in tasks in case of emergencies, etc.

Financial support

The reported study was funded by Russian Ministry of Science (information security), project No 08/2020.

References

1. Kudelkin V. A., Denisov V. F. Experience of integration of distributed information systems. *IT-standart*, 2017, no. 1. Available at: http://journal.tc22.ru/wp-content/uploads/2018/02/opit_integracii_raspredelennih_informacionnih_sistem.pdf (accessed 03 September 2021) (In Russian).
2. Abdulqadir H. R., Zeebaree S. R., Shukur H. M., Sadeeq M. M., Salim B. W., Salih A. A., Kak S. F. et al. A study of moving from cloud computing to fog computing. *Qubahan Academic Journal*, 2021, vol. 1, no. 2, pp. 60–70. doi:10.48161/qaj.v1n2a49
3. Yousefpour A., Ishigaki G., Jue J. P. Fog Computing: Towards minimizing delay in the Internet of Things. *Proceedings — 2017 IEEE 1st International Conference on Edge Computing*, Edge, 2017, pp. 17–24. doi:10.1109/IEEE.Edge.2017.12
4. Naas M. I., Lemarchand L., Raipin P., Boukhobza J. IoT data replication and consistency management in fog computing. *Journal of Grid Computing*, 2021, vol. 19, iss. 33. doi:10.1007/s10723-021-09571-1
5. Gorbunova A. V., Medvedeva E. G., Gaidamaka Yu. V., Shorgin V. S., Samouylov K. E. Effective user service strategies in a multi-user MIMO system. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2019, no. 4, pp. 69–81 (In Russian). doi:10.31799/1684-8853-2019-4-69-81
6. Lin J., Chelliah P. R., Hsu M., Hou J. Efficient fault-tolerant routing in IoT wireless sensor networks based on bipartite-flow graph modeling. *IEEE Access*, 2019, vol. 7, pp. 14022–14034. doi:10.1109/ACCESS.2019.2894002
7. Grover J., Garimella R. M. Reliable and fault-tolerant IoT-edge architecture. *IEEE Sensors,* IEEE, 2018, pp. 1–4. doi:10.1109/ICSENS.2018.8589624
8. Melnik E. V., Klimenko A. B., Ivanov D. Y. Distributed information and control system reliability enhancement by fog-computing concept application. *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, 2018, vol. 327, no. 2, pp. 022070. doi:10.1088/1757-899X/327/2/022070
9. Kinmanon U. G. Zh., Kozyrev D. V. Analytical and simulation modeling of the reliability of a closed ho-

mogeneous system with an arbitrary number of data sources and limited resources for their processing. *Modern Information Technology and IT-education*, 2018, vol. 14, no. 3, pp. 552–559 (In Russian). doi:10.25559/SITITO.14.201803.552-559

10. Skoba A. N., Mikhaylov V. K., Panfilov A. N. The problem of determining the optimal fault tolerance of distributed information processing systems. *Engineering Journal of Don*, 2020, no. 4. Available at: http://www.ivdon.ru/uploads/article/pdf/IVD_28__3_scoba_mikhaylov_ayesh_loganchuk.pdf_fe4c698775.pdf (accessed 06 September 2021) (In Russian).
11. Beraldi R., Canali C., Lancellotti R., Mattia G. P. Distributed load balancing for heterogeneous fog computing infrastructures in smart cities. *Pervasive and Mobile Computing*, 2020, vol. 67, no. 6, pp. 101221. doi:10.1016/j.pmcj.2020.101221
12. Kapsalis A., Kasnesis P., Venieris I. S., Kaklamani D. I., Patrikakis C. Z. A cooperative fog approach for effective workload balancing. *IEEE Cloud Computing*, 2017, vol. 4, no. 2, pp. 36–45. doi:10.1109/MCC.2017.25
13. Pham Q.-V., LeAnh T., Tran N. H., Seon Hong C. Decentralized computation offloading and resource allocation in heterogeneous networks with mobile edge computing. *IEEE Transactions on Mobile Computing*, March 2018. doi:arXiv:1803.00683
14. Peng Q., Wu C., Xia Y., Ma Y., Wang X., Jiang N. DoSRA: A decentralized approach to online edge task scheduling and resource allocation. *IEEE Internet of Things Journal*. doi:10.1109/JIOT.2021.3107431
15. Jia B., Hu H., Zeng Y., Xu T. Double-matching resource allocation strategy in fog computing networks based on cost efficiency. *Journal of Communications and Networks*, 2018, vol. 20, no. 3, pp. 237–246. doi:10.1109/JCN.2018.000036
16. *Elementy teorii testirovaniya i upravleniya tekhnicheskimi sistemami* [Elements of the theory of testing and control of technical systems]. R. M. Yusupov Ed. Leningrad, Energetika Publ., 1978. 191 p. (In Russian).
17. Burlov V. G., Gryzunov V. V., Tatarnikova T. M. Threats of information security in the application of GIS in the interests of the digital economy. *Journal of Physics: Conference Series*, 2020, vol. 1703, pp. 012023 doi:10.1088/1742-6596/1703/1/012023
18. Burlov V. G., Gryzunov V. V. Evaluation of the effectiveness of geographic information systems adaptation to destabilizing factors. *Journal of Physics: Conference Series,* 2020, vol. 1703, pp. 012016. doi:10.1088/1742-6596/1703/1/012016
19. Tananko I. E., Fokina N. P. Method of analysis of queuing networks with unreliable devices and information delay. *Vestnik Tomskogo gosudarstvennogo universiteta. Menedzhment, komp'yuternyye tekhnologii i informatika*, 2020, no. 52, pp. 90–96 (In Russian). doi:10.17223/19988605/52/11
20. Gryzunov V. V. Dynamic aggregation of pools in military computing systems. *Informatsionno-upravlyayushchiye sistemy* [Information and Control Systems], 2015, no. 1, pp. 13–20 (In Russian). doi:10.15217/issn1684-8853.2015.1.13
21. Gryzunov V. V. Problem solving method of measuring and calculating tasks under conditions of data computing system degradation. *Vestnik SibGUTI*, 2015, no. 1 (29), pp. 35–46 (In Russian).
22. Gryzunov V. V. FIST geoinformation system model using fog computing in destabilization. *Herald of Dagestan State Technical University. Technical Sciences*, 2021, no. 48 (1), pp. 76–89 (In Russian). https://doi.org/10.21822/2073-6185-2021-48-1-76-89

УДК 004.27+004.056
doi:10.31799/1684-8853-2022-1-19-29

Модель распределенной информационной системы, решающей задачи с требуемой вероятностью

В. В. Грызунов[a], канд. техн. наук, доцент, orcid.org/0000-0003-4866-217X, viv1313r@mail.ru
[a]Российский государственный гидрометеорологический университет, Воронежская ул., 79, Санкт-Петербург, 192007, РФ

Введение: распределенные в пространстве-времени сети IIoT и IoT, fog- и edge-вычисления имеют тренд проникать во все сферы человеческой деятельности. От качества работы этих технологий зависят предприятия, органы власти, силовые структуры и т. д. **Цель:** определить состав сети, обеспечивающий требуемую вероятность безотказной работы. **Методы:** согласно концепции структурно-функционального синтеза распределенная сеть представлена как нестабильная система массового обслуживания, в которой обслуживающие устройства подключаются и отключаются в произвольный момент времени. Построена имитационная модель сети. **Результаты:** состояние сети зависит от количества устройств и задач, их производительностей и времен жизни. В модели рассматриваются не сами эти величины, а их соотношения. Рассчитаны значения вероятности безотказной работы сети для всех возможных сочетаний соотношений. Рассчитан доверительный интервал с уровнем доверия 0,95. Из полученных данных понятно: 1) каким должен быть минимальный состав сети, чтобы обеспечить требуемую вероятность; 2) какую вероятность может обеспечить текущий состав сети; 3) какой поток задач допустим, чтобы решать задачи с требуемой вероятностью. Показано, что зависимость среднего времени нахождения задачи в сети от состава сети имеет две точки перегиба. Используя сведения об этих точках, система управления сетью формирует пулы из устройств или увеличивает количество устройств. **Обсуждение:** подразумевается, что сеть имеет полносвязную структуру. Следовательно, для практического применения необходимо расширить модель матрицей смежности, описывающей связи между узлами, а значит, пути распространения задач по сети, или считать, что каждый узел является ретранслятором и способен передать задачу на любой другой узел сети. Накладные издержки, возникающие при этом, учитываются через корректировку исходных данных. **Практическая значимость:** результаты позволяют минимизировать издержки при проектировании и эксплуатации распределенных систем, максимизировать вероятность безотказной работы систем при заданных ограничениях на ресурсы.

Ключевые слова — туманные вычисления, edge-вычисления, распределенные вычисления, отказоустойчивые вычисления, QoS, доступность.

Для цитирования: Gryzunov V. V. Model of a distributed information system solving tasks with the required probability. *Информационно-управляющие системы*, 2022, № 1, с. 19–29. doi:10.31799/1684-8853-2022-1-19-29

For citation: Gryzunov V. V. Model of a distributed information system solving tasks with the required probability. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2022, no. 1, pp. 19–29. doi:10.31799/1684-8853-2022-1-19-29

Финансовая поддержка

Исследование выполнено при финансовой поддержке Министерства науки России (информационная безопасность), проект № 08/2020.

ПАМЯТКА ДЛЯ АВТОРОВ

Поступающие в редакцию статьи проходят обязательное рецензирование.

При наличии положительной рецензии статья рассматривается редакционной коллегией. Принятая в печать статья направляется автору для согласования редакторских правок. После согласования автор представляет в редакцию окончательный вариант текста статьи.

Процедуры согласования текста статьи могут осуществляться как непосредственно в редакции, так и по e-mail (ius.spb@gmail.com).

При отклонении статьи редакция представляет автору мотивированное заключение и рецензию, при необходимости доработать статью — рецензию.

Редакция журнала напоминает, что ответственность
за достоверность и точность рекламных материалов несут рекламодатели.