UDC 004.41:519.7
DOI: 10.15587/1729-4061.2023.274177

# DEVISING A METHOD FOR BALANCING THE LOAD ON A TERRITORIALLY DISTRIBUTED FOGGY ENVIRONMENT

**Nina Kuchuk**
Doctor of Technical Sciences, Professor*
**Oleksandr Mozhaiev**
*Corresponding author*
Doctor of Technical Sciences, Professor**
E-mail: mozhaev1957@gmail.com
**Serhii Semenov**
Doctor of Technical Sciences, Professor
Department of Cyber Security and Information Technologies
Simon Kuznets Kharkiv National University of Economics
Nauky ave., 9-A, Kharkiv, Ukraine, 61166
**Andrii Haichenko**
PhD, Deputy Minister of Justice of Ukraine for Executive Service
Ministry of Justice of Ukraine
Architect Horodetskyi str., 13, Kyiv, Ukraine, 01001
**Heorhii Kuchuk**
Doctor of Technical Sciences, Professor*
**Serhii Tiulieniev**
PhD, Director****
**Mykhailo Mozhaiev**
Doctor of Technical Sciences, Head of Laboratory
Laboratory of Copyright and Information Technologies****
**Viacheslav Davydov**
Doctor of Technical Sciences, Associate Professor*
**Oksana Brusakova**
Doctor of Law, Associate Professor
Department of Law Enforcement and Policing***
**Yurii Gnusov**
PhD, Associate Professor**
*Department of Computer Engineering and Programming
National Technical Uniersity «Kharkiv Polytechnic Institute»
Kyrpychova str., 2, Kharkiv, Ukraine, 61002
**Department of Cyber Security and DATA Technologies***
***Kharkiv National University of Internal Affairs
L. Landau ave., 27, Kharkiv, Ukraine, 61080
****Scientific Research Centre for Forensic
on Intellectual Property of the Ministry of Justice of Ukraine
L. Ukrainky blvd., 26, Kyiv, Ukraine, 01133

***This study solves the task to redistribute the load on a geographically distributed foggy environment in order to achieve a load balance of virtual clusters. The necessity and possibility of developing a universal and at the same time scientifically based approach to load balancing has been determined. Object of study: the process of redistribution of load in a foggy environment between virtual, geographically distributed clusters. A load balancing method makes it possible to reduce delays and decrease the time for completing tasks on foggy nodes, which brings task processing closer to real time. To solve the task, a mathematical model of the functioning of a separate cluster in a foggy environment has been built. As a result of modeling, the problem of finding the optimal distribution of tasks across the nodes of the virtual cluster was obtained. The limitations of the problem take into account the characteristics of the physical nodes of support for the virtual cluster. The process of distributing the additional load was also simulated through the graph representation of tasks entering virtual clusters. The task to devise a method for load transfer between virtual clusters within a foggy environment is solved using the proposed iterative algorithm for finding a suitable cluster and placing the load. The simulation results showed that the balance of the foggy environment when using the proposed method increases significantly provided the network load is small. The scope of application of the results includes geographically distributed foggy systems, in particular the foggy layer of the industrial Internet of Things. A necessary practical condition for using the proposed results is the non-exceeding the specified limit of the total load on the foggy medium, usually 70 %***

***Keywords: decentralized platform, cloud environment, Internet of Things, virtual cluster, iterative algorithm***

*Received date 07.12.2022*
*Accepted date 15.02.2023*
*Published date 28.02.2023*

***How to Cite:*** *Kuchuk, N., Mozhaiev, O., Semenov, S., Haichenko, A., Kuchuk, H., Tiulieniev, S., Mozhaiev, M., Davydov, V., Brusakova, O., Gnusov, Y. (2023). Devising a method for balancing the load on a territorially distributed foggy environment. Eastern-European Journal of Enterprise Technologies, 1 (4 (121)), 48–55. doi: https://doi.org/10.15587/1729-4061.2023.274177*

## 1. Introduction

The Industrial Internet of Things (IIoT), based on cloud computing technology [1], is being actively introduced in many production sectors. However, this causes a number of difficulties that are associated with the following factors [2]:

– large geographical distribution of systems and customers;
– unforeseen network delays;

– high cost of bandwidth;

– mobility of end devices.

To solve these problems, it is proposed to use the technology of foggy computing [3]. This technology makes it possible [4]:

– to reduce the communicative load on the network;

– to unload cloud data centers;

– to reduce the latency of objects related to the Industrial Internet of Things.

Foggy computing is a promising concept for organizing distributed computations. Foggy computing implies bringing data processing to the end devices of networks; it is the development of the cloud concept [5].

Typically, foggy infrastructure is implemented using multipurpose decentralized platforms. A key feature of foggy computing technologies is to perform most of the data processing at the edge of the network. They are usually used in distributed systems in which the response time of the system is one of the main characteristics. This reduces the load on the communication network and reduces the response time of the system.

In the process of functioning of such platforms, it is often necessary to solve the problem of system reconfiguration to balance its load [6, 7]. This procedure can be repeated many times due to the dynamism of the foggy environment. Reconfiguration tasks are associated with the reassignment of the subtasks to be solved on efficient computational nodes [8, 9]. The problem of transferring the computational load belongs to the class of NP-complete ones. Given the very large number of nodes of the fog layer, it leads to the need to solve the optimization problem of large dimensionality. This requires significant time costs, which can significantly reduce the effect of rearranging the computational load.

In addition, the foggy environment has features that impose their limitations on the models, methods, and algorithms used to solve the problem of transferring the computational load. The problems are associated with the need to repeatedly solve the task of transferring the computational load during the functioning of the system, which is a consequence of the dynamism and instability of the foggy environment. The solution to the reconfiguration problem is directly related to time costs, which negatively affect the efficiency of the system [10].

So, to improve the efficiency of the foggy environment, it is necessary to reduce the delay data, that is, to solve the task of transferring the computational load in the minimum time. Therefore, the issue of devising a method for balancing the load on a distributed foggy medium is a relevant one.

## 2. Literature review and problem statement

The task of balancing the load of any computer network is directly related to the task of transferring the load. Such a task in the absence of pre-prepared descriptions of configurations includes two stages:

– selection of computational nodes for load arrangement;

– resolving the issue of load arrangement.

In distributed decentralized systems, the task of load transfer is significantly complicated. As a result, there are many scientific studies related to this area.

Article [11] proposes a two-level method of resource planning in distributed systems, in particular, it can be used for foggy calculations. But this method is focused only on simple topologies. Therefore, its use for geographically distributed systems is problematic.

The mechanism of transfer of computational load to reduce delays during the execution of tasks in accordance with the characteristics of containers is discussed in [12]. But this algorithm does not take into account the characteristics of the computational problem, in particular the time of calculations in the fog.

The algorithm for forming a schedule based on heuristics is proposed in [13]. But this algorithm, when used in a foggy environment, has a very high computational complexity, which increases exponentially with an increase in the number of nodes.

Multi-level planning architecture, taking into account delays, is used when considering decentralized systems [14, 15]. But this approach does not take into account the costs associated with data transmission.

The unloading algorithm for performing tasks on free nodes, discussed in articles [16, 17], is used only in a homogeneous medium. But any foggy environment is almost always heterogeneous.

An approach to resource management based on dynamic planning using methods for classifying heterogeneous devices is given in [18]. But this approach is not intended for use in large-scale applications, and therefore cannot be used in a geographically distributed foggy environment. In addition, with this approach it is impossible to take into account the monetary costs associated with the use of foggy resources.

Methods of balancing computing resources in decentralized systems using neural networks are used in [19, 20]. But these methods are unsuitable for systems operating under a mode close to real time.

A genetic algorithm to achieve minimization of the total delay, personalized to the planning of Internet requests, is implemented in [21]. But its objective function does not include criteria for maximizing resource utilization and minimizing latency.

The optimized task scheduling algorithm discussed in article [22] to minimize delays in critical applications has significant limitations on the size of the transmitted data.

Consequently, the above scientific papers, in the redistribution of the load, do not fully enough take into account the characteristic features of the geographically distributed foggy environment. Therefore, it is advisable to devise a method for balancing the load on a geographically distributed foggy medium.

## 3. The aim and objectives of the study

The aim of our work is to devise a method for balancing the load on a distributed foggy environment based on load redistribution between virtual clusters. This will reduce delays and decrease task execution time on foggy nodes, which should bring task processing closer to real time.

To accomplish the aim, the following tasks have been set:

– to build a mathematical model of the functioning of a separate cluster within a foggy environment;

– to simulate the process of distribution of additional load;

– to devise a method of load transfer between virtual clusters within a foggy environment.

## 4. The study materials and methods

The object of our study is the process of load redistribution in a foggy environment between virtual, geographically distributed clusters.

The main hypothesis of the study assumes the possibility of bringing the process of task processing in all virtual clusters within a foggy environment to real time.

Proving the hypothesis put forward involves the use of a number of different tools. When formalizing the architecture of the foggy environment, methods to forming a cluster structure were applied. This choice is justified by the property of territorial distribution of the basic network. When formalizing network structures, the set theory methods were chosen. This choice is due to the heterogeneity of network structures. In the simulation, we employed methods focused on distributed decentralized systems.

Considering the discrete nature of the main variables of the obtained optimization problems, methods of discrete optimization and Boolean programming were applied to find solutions. For clarity of the statement of some problems and justification of the choice of mathematical apparatus, a graph representation of the relationships between the elements of problems was used. To analyze the load balance indicator of virtual clusters within a foggy environment, we employed statistical methods for analyzing random variables.

The application of this toolkit will make it possible to prove the specified research hypothesis.

The process of devising a method for balancing the load on a distributed foggy environment involved working with geographically distributed clusters. When devising the method, we were guided by the following conditions:

Condition 1. The foggy environment is territorially distributed.

Condition 2. There is no centralized monitoring of virtual cluster loading.

Condition 3. Each virtual cluster has a binding node that has statistics on the loading of basic physical nodes and has a constant connection with the cloud data center.

Condition 4. Some tasks involve performing subtasks in parallel.

For experimental evaluation of the results of our study, a model of the Industrial Internet of Things (IIoT) is proposed. It was planned to consider 15 non-intersecting virtual clusters of various configurations, which are based on geographically distributed groups of physical devices. This configuration was due to the real needs of IIoT customers.

## 5. Results of studying the process of load redistribution between virtual clusters within a foggy environment

### 5. 1. Construction of a mathematical model of the functioning of a separate cluster within a foggy environment

Foggy computing is a multi-level decentralized model that provides access to a common set of computing resources. Foggy computing minimizes the network response time of supported applications and provides end devices with local computing resources. If necessary, network connection to centralized services is provided. The architecture of foggy computing can be considered as a «layer» between the cloud and the end devices (Fig. 1).

Foggy computing has a limited number of temporary storages, making it possible to temporarily store the retrieved data for analysis and then send the necessary feedback to the output devices. Closely spaced devices are grouped together, forming virtual clusters. Based on their location, data sources can join and disconnect from any virtual cluster. In Fig. 1, the fog environment $M_{FOG}$ consists of $z$ of non-intersecting virtual clusters $K_i$, and the environment is fully connected with respect to them:

$$M_{FOG}=\bigcup_{i=1}^{z}K_i;\ \bigcap_{i=1}^{z}K_i=\varnothing. \tag{1}$$

Each virtual cluster can be represented as a set of elementary nodes, that is, such nodes that cannot simultaneously perform two or more operations.

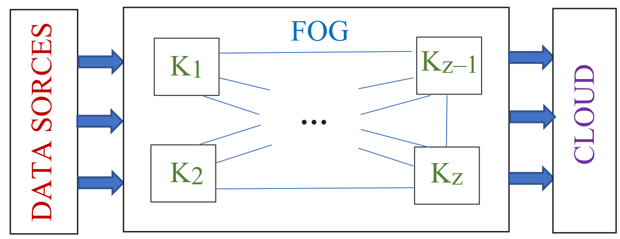


Fig. 1. An example of a fully connected foggy environment architecture

Consider a separate virtual cluster containing the set $M$ of foggy elementary nodes of power $m$. We shall build a mathematical model of its functioning, provided that it is necessary to pre-process before sending to the cloud a set of problems $N$ of power $n$.

Let $\xi_{ik}$ ($i=1... n$, $k=1... m$) is the duration of execution of the $i$-th ($i\in N$) task on the $k$-th ($k\in M$) node of the virtual cluster in question. The start time of this task on the $k$-th node is denoted as $T_{ik}$ ($i=1...n$, $k=1...m$).

Also note that each $i$-th task consists of $q_i$ components that can be performed independently.

We introduce the Boolean variable $\theta_{ilk}$, which will take a unit value if and only if the $l$-th component of the $i$-th task is performed on the $k$-th node. Otherwise, $_{ijk}\theta=0$.

Note that for any pair of tasks $i\in N$ and $j\in N$, only one of the following inequalities is valid:

$$T_{ik}-T_{jk}\geq\xi_{ik} \tag{2}$$

or

$$T_{jk}-Ti_{jk}<\xi_{ik}. \tag{3}$$

These inequalities are a consequence of the elementary property for the $k$-th node. That is, the execution of the $i$-th task on the $k$-th node precedes the execution of the $j$-th task or vice versa.

To determine the order of tasks on each node of the virtual cluster, we introduce boolean variables $y_{ijk}$. The variable $y_{ijk}$ is equal to one if and only if on the $k$-th node the $i\in N$ task precedes the $j\in N$ task, and not necessarily directly. Otherwise, $y_{ijk}=0$. Now constraints (2) and (3) can be written as two conditions, each of which must be met:

$$\left(\xi_{jk}+C\right)y_{ijk}+\left(T_{ik}-T_{jk}\right)\geq\xi_{jk}; \tag{4}$$

$$\left(\xi_{ik}+C\right)\left(1-y_{ijk}\right)+\left(T_{jk}-T_{ik}\right)\geq\xi_{ik}, \tag{5}$$

where $C$ is a sufficiently large constant, chosen so that only one of the equalities is satisfied: $y_{ijk}=0$ or $y_{ijk}=1$.

To comply with restrictions on the order of operations, we stipulate that:

$$\tau_{il} = \sum_{k=1}^{m} \theta_{ilk} T_{ik}, \tag{6}$$

where $i \in \overline{1,n}$, $l \in \overline{1,q_i}$, $\tau_{il}$ is the moment of commencement of the component $l$ of the task $i$.

For all components, except for the last component of each work, there must be an inequality:

$$\sum_{k=1}^{m} \theta_{ilk}\left(T_{ik} + \xi_{ik}\right) \le \sum_{k=1}^{m} \theta_{i,l+1,k} T_{ik}, \tag{7}$$

where $i \in \overline{1,n}$, $l \in \overline{1,q_i - 1}$.

The choice of indicator to optimize the planning process depends on the goals of a computer system that uses a foggy environment.

If you need to start completing tasks as early as possible, the functionality is used:

$$F_1(l) = \sum_{i=1}^{n} \sum_{k=1}^{m} \theta_{i,l,k} T_{ik}, \tag{8}$$

which needs to be minimized.

In most cases, there is a need to complete all the tasks as early as possible. Then the following functionality is minimized:

$$F_2(i) = \max_{l} \sum_{k=1}^{m} \theta_{i,l,k} T_{ik}. \tag{9}$$

So, we can formulate the problem of finding the optimal distribution of tasks across the nodes of the virtual cluster. The elements of the set $T=\{T_{ik}\}$ of the start time of performing tasks on cluster nodes are considered as variables. Also variables in this statement of the problem are elements of the set $Y=\{y_{ijk}\}$ – the set of preceding the time of execution of independent task components.

The objective function of the problem takes the form:

$$F_1(l) \to \min_{l}, \text{ or } F_2(i) \to \min_{i}. \tag{10}$$

In this case, the constraints of the problem are given by expressions (2) to (7).

So, this subsection proposes a mathematical model of an autonomous virtual cluster of a foggy network. But in the case of overloading of the cluster under consideration, it is necessary to use transit sections of the network to unload it. Therefore, it is necessary to simulate the process of distribution of excess load.

**5. 2. Modeling additional load distribution**

The mathematical model discussed in the previous chapter is focused only on the autonomous virtual cluster of the foggy network. Therefore, the model does not take into account the fact of the presence of transit sections of the network, which significantly affect the speed of solving computational problems.

Consider the statement of the problem of transferring the computational load, taking into account the presence of transit sections of the network, which is characteristic of a foggy environment. Let us single out the computational problem $Z$, the connection between the components of which and the sequence of actions are described by graph $G$. Execution of the task is scheduled on the virtual cluster $K'$. But monitoring the loading of virtual fog clusters showed that the $K'$ cluster was currently overloaded. Among the available transit areas, the most favorable is the $K''$ virtual cluster.

The graph G is divided into two subgraphs: $G'$ and $G''$ (Fig. 2). It is necessary to place the subgraph $G''$ on the computing devices of the virtual cluster $K''$. The computational subtasks of the $G'$ subgraph components will continue to be performed on the computing devices of the basic virtual cluster $K'$.

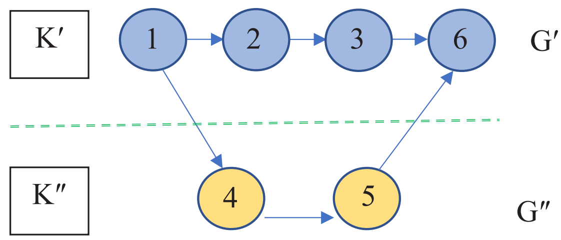


Fig. 2. Virtual cluster unloading diagram

Accordingly, such a transfer of computational subtasks involves the solution of an optimization problem with certain criteria and limitations, which are partially formed by the requirements of the computing system itself. The main criterion for solving the problem of transferring the computational load from the basic virtual cluster is the reliability of the system.

Consider the graph description of the set of subtasks of the fixed problem $Z$:

$$G=\{\langle i, x_i, w_i\rangle\}, \tag{11}$$

where $i$ is a unique identifier of the computational subtask, which has the computational complexity $x_i$ and transmits the amount of information $w_i$.

Computational subtasks of the graph $G$ are performed on the nodes of computing devices of the set $K$, which is described by the graph structure:

$$K=\{\langle j, c_j\rangle, BW\}, \tag{12}$$

where $j$ is the node identifier, $c_j$ is the node performance, $BW$ is the throughput matrix of communication channels between the incident nodes of the foggy network.

Now consider the computational subtasks of subgraph $G''$, which need to be transferred while the computational subtasks of subgraph $G'$ continue their execution. Between these subgraphs there are several input ($F\_in$) and outgoing ($F\_out$) information flows associated with the subgraph $G''$.

The flow of input information to the nodes of the subgraph $G''$ is described by the set of such tuples:

$$F\_in=\{F\_in\_k\}=\{\langle ik_{out}, ik_{in}, wk_{out\_in}\rangle\}, \tag{13}$$

where from the set of nodes $\{ik_{out}\}$ of the subgraph $G'$ to the set of nodes$\{ik_{in}\}$ of the subgraph $G''$ the volumes of information recorded in the set $\{wk_{out\_in}\}$ are transmitted.

Similarly, the flow of source information from the nodes of the subgraph $G''$ to the nodes of the base cluster of the foggy environment is described:

$$F\_out=\{F\_out\_k\}=\{\langle ik_{in}, ik_{out}, wk_{in\_out}\rangle\}. \tag{14}$$

Consider the problem of placing the load. Let there be a subgraph of computational problems $G'$, associated with cluster $K'$, and the sets are formed of flows $F\_in$ and $F\_out$.

It is necessary to place the computational subtasks of the subgraph $G''$ on the set of devices of the cluster $K''$ so that the total execution time of computational subtasks $G$ is less than the specified time $T$ taking into account the fulfillment of the criterion of reliability of the system.

The solution to this problem is to establish a connection between the computational subtasks of the subgraph $G''$ and the computational nodes of the $K''$ cluster, which can be described by the matrix of tuples for the allocation of resources of the cluster $K''$:

$$\gamma = \left\| \left\langle t_{ij}^{(0)}, u_{ij} \right\rangle \right\|, \tag{15}$$

where $i \in \overline{1,n''}$, $j \in \overline{1,\varphi''}$, $\varphi'' = \mathrm{card}\, G''$, $t_{ij}^{(0)}$ is the moment of relative time (from the beginning of the implementation of load transfer) when the calculation of the $i$-th subtask on the $j$-th node begins, $u_{ij}$ – the fraction of the total performance of the $j$-th node to perform the $i$-th subtask, $n''$ – the total number of computational nodes of the cluster $K''$, $\varphi''$ – the total number of subtasks execution of which is migrated from the base cluster.

This model of the task makes it possible to connect more than one task that comes to one node at a time.

For further development of the model, it is necessary to consider the following parameters regarding the loading of the $j$-th node:

– $Lp_j(\gamma)$ – loading of the $j$-th node generated by transferring the computational subtask to this node;

– $Ldist_j(\gamma, F\_in, F\_out)$ – load of the $j$-th node generated by the exchange of information between subtasks included in the structures specified by subgraphs $G'$ and $G''$;

– $Ltr_j(\gamma, F\_in, F\_out)$ – load of the $j$-th node generated by the use of this node as a transit node when implementing load transfer between virtual clusters;

– $D_{lk}$ – a list of edges of the graph $K$, which determines the route between the nodes $l$ and $k$;

– $L(D_{lk})$ is a matrix that describes the bandwidth of communication channels between nodes $l$ and $k$.

The loading of nodes of the virtual cluster $K''$ depends on the distribution of computational tasks over the nodes of the network, which is described by the matrix $\gamma$. In this case, the load of the $j$-th node depends on such parameters as $Lp_j(\gamma)$, $Ldist_j(\gamma, F\_in, F\_out)$, $Ltr_j(\gamma, F\_in, F\_out)$.

Thus, the full load of the $j$-th node is described by the following formula:

$$L_j(\gamma) = Lp_j(\gamma) + Ldist_j(\gamma, F\_in, F\_out) + $$
$$+ Ltr_j(\gamma, F\_in, F\_out). \tag{16}$$

Consider the objective function as a set of values of the reliability functions of the involved nodes of graph $K$:

$$F_j(t) = \exp(-\lambda_j t), \tag{17}$$

where $\lambda_j$ is the failure intensity of the $j$-th node, and $t$ is the operating time of the service device.

Since the intensity of failures depends both on the load of the node at the current time and on its technical characteristics, then,

$$\lambda_j(\gamma) = \lambda_{0j}\psi(k_j, L_j(\gamma)), \tag{18}$$

where $\lambda_{0j}$ – the primary value of the failure intensity of the $j$-th node, $k_j$ – the coefficient, which depends on the type of service device; $L_j(\gamma)$ – the current load of the $j$-th node when using the distribution $\gamma$, calculated by expression (16).

In addition, for each distribution $\gamma$, we introduce a subset $K''(\gamma) \subset K''$, to which those and only those elements of the set $K''$ belong, on which the subtasks included in the subgraph $G''$ are planned to be performed in this distribution.

So, it is possible to form a generalized functionality of the objective function for the task of transferring the load to an additional virtual cluster in the form of a multiplicative indicator:

$$F_{mult}(\gamma, \Delta\tau) = \prod_{j \in K''(\gamma),\, t \in \Delta\tau} \eta_j \exp\left(-\left(\lambda_{0j}\psi\left(k_j, L_j(\gamma, \Delta\tau)\right)\right)t\right) =$$
$$= \eta \cdot \exp\left(-t \cdot \sum_{j \in K''(\gamma)} \lambda_{0j}\psi\left(k_j, L_j(\gamma, \Delta\tau)\right)\right), \tag{19}$$

where $\eta = \prod_{j \in K''(\gamma),\, t \in \Delta\tau} \eta_j$; $\eta_j$ is the weighting factor of the $j$-th node of the virtual cluster $K''$; $\Delta\tau$ – time period.

Then the objective function of the optimization problem under consideration will take the form:

$$\min_{t \in \Delta\tau} F_{mult}(\gamma, \Delta\tau) \to \max_{\gamma}. \tag{20}$$

The main limitation for the optimization problem is the maximum possible time $T$ planned to perform the graph of the computational task $G''$. Therefore, this restriction will take the form:

$$\forall\left(i \in \overline{1,n''}, j \in K''(\gamma)\right) \Rightarrow \left(t_{ij}^{(0)} + \frac{x_{ij}}{u_{ij} \cdot p_j} + t_i^{(\text{transfer})}\right) < T, \tag{21}$$

where $t_i^{(\text{transfer})}$ is the maximum time of delivery of information from the $i$-th subtask to all subtasks that receive its initial data.

The formulated optimization problem underlies the method of load transfer between virtual clusters of the foggy environment, which is discussed in the next subsection.

**5. 3. Development of a method for load transfer between virtual clusters within a foggy environment**

The specificity of the foggy environment is the absence of any centralization between its components. Usually, independent virtual clusters are considered as components. Clustering is carried out on a territorial basis, with each cluster having a binding node. The functions of such a node include constant operational communication with the cloud environment and monitoring the state of the physical components of this virtual cluster.

To solve the task of transferring part of the computational load, it is necessary to find an additional virtual cluster (or a group of virtual clusters). The corresponding method is implemented using the following algorithm. The algorithm considers the computational task $Z$, described by the graph $G$, the execution of which is planned on a partially overloaded virtual cluster $K'$. To unload $K'$, you need to find the best available transit section:

Step 1. The graph $G$ is divided into two subgraphs: $G'$ and $G''$, planning the execution of the tasks of the subgraph $G''$ outside the basic virtual cluster. The distribution of subtasks described by the subgraph $G'$ is modeled in the environment of the basic separate foggy environment.

Step 2. Ranking is carried out on the basis of remoteness from the input data of all virtual clusters of the foggy environment, which are transit for the base cluster $K'$. As a result, we obtain the following sequence:

$$K_1'', K_2'', \ldots, K_i'', \ldots. \tag{22}$$

Step 3. Cluster $K_1''$ is selected as the current transit cluster.

Step 4. The base cluster establishes a connection with the binding node of the current transit cluster and receives information on the possibility of placing an additional load.

Step 5. If a negative response is received (if there are not enough resources in the current cluster), then the next element of sequence (22) is selected as the current cluster. If the previous current element was the last in the sequence, then proceed to step 7, otherwise – to step 4.

Step 6. If a positive response is received, then the current cluster is fixed. The optimal distribution of additional load on the current cluster is simulated. Go to step 8.

Step 7. Sequence (22) is converted as follows:

$$K_1'' \cup K_2'', K_2'' \cup K_3'', \ldots, K_{i-1}'' \cup K_i'', K_{i1}'' \cup K_{i+1}'', \ldots. \tag{23}$$

As the current transit cluster, the cluster $K_1'' \cup K_2''$ is selected and we proceed to step 4.

Step 8. The process of redistributing the load for task $Z$ is completed.

Note. If a pair of virtual clusters is not found in sequence (23), then the expediency of redistributing part of the load of the base cluster looks questionable. Therefore, the model of distribution of subtasks of a separate task is performed on a basic cluster for all subtasks of the computational task $Z$.

Evaluation of the quality of the proposed algorithm will be carried out using the indicator of the balance of loading of virtual clusters of the foggy environment. To do this, consider the discrete random variable $X$ – the fraction of the current workload of the virtual cluster. Considering the features of this environment as an indicator of balance, the variance of this random variable was chosen.

The value of the random variable $X$ can be calculated based on expressions (12) and (16):

$$X = \left\{ x_j, j = \overline{1,z} \right\}, \tag{24}$$

where $x_j = L_j(\gamma)/c_j$.

Then the mathematical expectation of a random variable $X$, that is, the average workload of the cloud environment, is calculated as:

$$M[X] = \frac{1}{z} \sum_{j=1}^{z} \frac{L_j(\gamma)}{c_j}, \tag{25}$$

and the variance, that is, the generalized equilibrium indicator, is equal to:

$$D[X] = \frac{1}{z} \sum_{j=1}^{z} \left( \frac{L_j(\gamma)}{c_j} - \frac{1}{z} \sum_{j=1}^{z} \frac{L_j(\gamma)}{c_j} \right)^2. \tag{26}$$

To compare the proposed and existing algorithms, the Industrial Internet of Things (IIoT) foggy environment was modeled. 15 non-intersecting virtual clusters of different configurations, which are based on geographically distributed groups of physical devices, were considered. Randomly, a load was applied to the input of all virtual clusters of the model. 500 different input load options were simulated. The average current workload value of one virtual cluster ranged from 15 to 85 %. The results of calculating the dependence of the variance of the fraction of the current workload of the virtual cluster on the congestion of the foggy network for the proposed and existing algorithms are shown in Fig. 3.

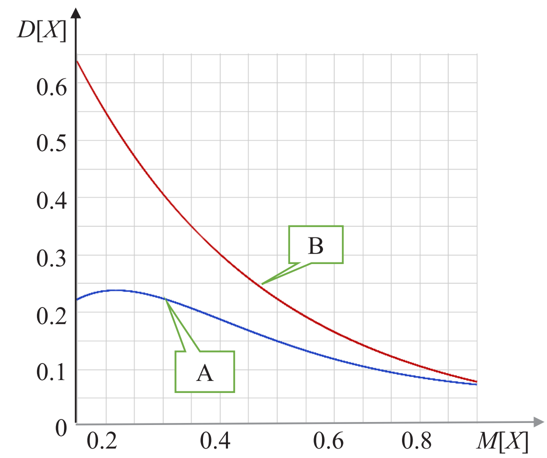


Fig. 3. Results of the analysis of the balance of the foggy environment: A – a proposed algorithm; B – an existing algorithm

Analysis of the simulation results showed that with an average load within a foggy environment of 15 to 30 %, it is possible to achieve a level of balance greater than twice as high as the existing one. This is due to the underloading of transit routes of the base network. Quite a tangible advantage (from 50 %) is achieved with an average load within a foggy environment from 30 to 60 %. But with an increase in the average workload of 60 %, there were no significant successes in balancing the network during the operational unloading of some clusters. It should be noted the inexpediency of the proposed approach with an average load within a foggy environment of more than 75 %. With such a load, the balance of the network practically does not change but unproductive losses on the parallelization of information flows increase. Consequently, the balance of the foggy environment when using the proposed algorithm increases significantly if the load of the base network does not exceed 60 %.

## 6. Discussion of the results of improving the balance of the foggy environment

The mathematical model of the functioning of a separate cluster within a foggy environment was built by representing a virtual cluster as a set of elementary nodes. As a result of the simulation, the problem of finding the optimal distribution of tasks across the nodes of the virtual cluster with the objective function (10) and the constraints specified by formulas (2) to (7) was obtained.

The process of modeling the distribution of additional load is based on the graph representation of tasks entering virtual clusters. As a result, a generalized functional (19) was formed for the objective function (20) of the load transfer problem to an additional virtual cluster. Optimization is carried out according to the criterion of reliability with strict time constraints specified by formula (21).

The proposed method of load transfer between virtual clusters within a foggy environment is based on an iterative algorithm for finding a suitable cluster and placing the load.

The iterative algorithm finds the best available transit section, selected by criterion (20).

In contrast to the two-level method of resource planning in distributed systems [11], the proposed method can be used for geographically distributed foggy systems. This becomes possible through the use of an iterative virtual cluster search algorithm. In contrast to [12] where the mechanism of computational load transfer is considered, the proposed method takes into account the characteristics of the computational problem, in particular the time of calculations in the fog. At the same time, the computational complexity of the method is significantly less than in the heuristic approach [13]. In addition, the costs associated with data transfer are taken into account, in contrast to the methods proposed in [14, 15]. This becomes possible through the use of specific features of the foggy environment. Also, in contrast to the approaches to unloading the load given in [16, 17], balancing can be used in a heterogeneous environment. Large-scale applications are also possible to implement this approach, in contrast to the method of resource management based on dynamic planning [18]. In addition, in contrast to intelligent balancing methods [19–21], the proposed method minimizes latency and helps bring the process closer to real time. This is made possible by using the partitioning of the foggy environment into disjointed virtual clusters according to expression (1).

Thus, our solutions close the problem part regarding the redistribution of the load, taking into account the characteristic features of the geographically distributed foggy environment. In particular, the preprocessing time of tasks in the fog before entering the cloud has been reduced due to load balancing between virtual clusters. This is due to the division of a geographically distributed foggy environment into disjointed virtual clusters and the use of an iterative algorithm for finding a transit route for unloading.

The quality assessment of the proposed algorithm was carried out using the proposed indicator of the load balance of virtual clusters of the foggy environment. The simulation results showed that the balance of the foggy environment when using the proposed method increases significantly if the network load is small.

The current study has limitations on the network congestion within a foggy environment. With an average network load of more than 60 %, the use of the proposed method is impractical since the balance practically does not change. But, at the same time, there are additional delays associated with both the search for an additional virtual cluster and the transfer of the load.

The disadvantage of the study is the need to conduct an iterative survey of several virtual clusters while increasing the imbalance of the foggy environment. This deficiency is planned to be eliminated through the use of a partial operational monitoring procedure.

The development of the study involves the development of a method for determining the optimal algorithm for eliminating the imbalance of the foggy environment.

## 7. Conclusions

1. The task of developing a mathematical model of the functioning of a separate cluster within a foggy environment is solved by representing the cluster as a set of elementary nodes. As a result of modeling, the problem of finding the optimal distribution of tasks across the nodes of the virtual cluster was obtained. The limitations of the problem take into account the order of tasks and the characteristics of the physical support nodes of the virtual cluster.

2. The task of modeling the process of additional load distribution is solved by graph representation of tasks entering virtual clusters. This made it possible to divide the load of the allocated task on an overloaded cluster and place part of it for parallel execution on another virtual cluster. As a result, a generalized functionality of the objective function was formed for the task of transferring the load to an additional virtual cluster. Optimization is carried out according to the criterion of reliability with strict time constraints.

3. The task of developing a method for load transfer between virtual clusters within a foggy environment is solved using the proposed iterative algorithm for finding a suitable cluster and placing the load. The quality assessment of the proposed algorithm was carried out using the proposed indicator of the load balance of virtual clusters of the foggy environment. With an average load within a foggy environment of 15 to 30 %, the achieved level of balance is more than twice as high as the existing one. An increase in the level of network balance from 50 % is observed with an average load within a foggy environment from 30 to 60 %. With an increase in average workload from 60 %, no significant success in balancing the network during the operational unloading of some virtual clusters is observed. With an average load within a foggy environment of more than 75 %, the balance of the network almost does not change.

## Conflicts of interest

The authors declare that they have no conflicts of interest in relation to the current study, including financial, personal, authorship, or any other, that could affect the study and the results reported in this paper.

## Funding

The study was conducted without financial support.

## Data availability

All data are available in the main text of the manuscript.

## References

1. Kumar, N., Sharma, B., Narang, S. (2022). Emerging Communication Technologies for Industrial Internet of Things: Industry 5.0 Perspective. Lecture Notes in Networks and Systems, 107–122. doi: https://doi.org/10.1007/978-981-19-1142-2_9
2. Qayyum, T., Trabelsi, Z., Waqar Malik, A., Hayawi, K. (2022). Mobility-aware hierarchical fog computing framework for Industrial Internet of Things (IIoT). Journal of Cloud Computing, 11 (1). doi: https://doi.org/10.1186/s13677-022-00345-y

3. Chalapathi, G. S. S., Chamola, V., Vaish, A., Buyya, R. (2021). Industrial Internet of Things (IIoT) Applications of Edge and Fog Computing: A Review and Future Directions. Advances in Information Security, 293–325. doi: https://doi.org/10.1007/978-3-030-57328-7_12
4. Lu, S., Wu, J., Wang, N., Duan, Y., Liu, H., Zhang, J., Fang, J. (2021). Resource provisioning in collaborative fog computing for multiple delay-sensitive users. Software: Practice and Experience, 53(2), 243–262. doi: https://doi.org/10.1002/spe.3000
5. Özdoğan, E. (2022). Cloud, Fog, and Edge Computing for IoT-Enabled Cognitive Buildings. IoT Edge Solutions for Cognitive Buildings, 23–52. doi: https://doi.org/10.1007/978-3-031-15160-6_2
6. Kuchuk, G. A., Akimova, Yu. A., Klimenko, L. A. (2000). Method of optimal allocation of relational tables. Engineering Simulation, 17 (5), 681–689.
7. Attar, H., Khosravi, M. R., Igorovich, S. S., Georgievan, K. N., Alhihi, M. (2021). E-Health Communication System with Multiservice Data Traffic Evaluation Based on a G/G/1 Analysis Method. Current Signal Transduction Therapy, 16 (2), 115–121. doi: https://doi.org/10.2174/1574362415666200224094706
8. Kovalenko, A., Kuchuk, H., Kuchuk, N., Kostolny, J. (2021). Horizontal scaling method for a hyperconverged network. 2021 International Conference on Information and Digital Technologies (IDT). doi: https://doi.org/10.1109/idt52577.2021.9497534
9. Raskin, L., Sukhomlyn, L., Sagaidachny, D., Korsun, R. (2021). Analysis of multi-threaded markov systems. Advanced Information Systems, 5 (4), 70–78. doi: https://doi.org/10.20998/2522-9052.2021.4.11
10. Svyrydov, A., Kuchuk, H., Tsiapa, O. (2018). Improving efficienty of image recognition process: Approach and case study. 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT). doi: https://doi.org/10.1109/dessert.2018.8409201
11. Li, G., Liu, Y., Wu, J., Lin, D., Zhao, S. (2019). Methods of Resource Scheduling Based on Optimized Fuzzy Clustering in Fog Computing. Sensors, 19 (9), 2122. doi: https://doi.org/10.3390/s19092122
12. Proietti Mattia, G., Beraldi, R. (2023). P2PFaaS: A framework for FaaS peer-to-peer scheduling and load balancing in Fog and Edge computing. SoftwareX, 21, 101290. doi: https://doi.org/10.1016/j.softx.2022.101290
13. Hoang, D., Dang, T. D. (2017). FBRC: Optimization of task Scheduling in Fog-Based Region and Cloud. 2017 IEEE Trustcom/BigDataSE/ICESS. doi: https://doi.org/10.1109/trustcom/bigdatase/icess.2017.360
14. Sharma, S., Saini, H. (2019). A novel four-tier architecture for delay aware scheduling and load balancing in fog environment. Sustainable Computing: Informatics and Systems, 24, 100355. doi: https://doi.org/10.1016/j.suscom.2019.100355
15. Khudov, H., Diakonov, O., Kuchuk, N., Maliuha, V., Furmanov, K., Mylashenko, I. et al. (2021). Method for determining coordinates of airborne objects by radars with additional use of ADS-B receivers. Eastern-European Journal of Enterprise Technologies, 4 (9 (112)), 54–64. doi: https://doi.org/10.15587/1729-4061.2021.238407
16. Malik, U. M., Javed, M. A., Frnda, J., Rozhon, J., Khan, W. U. (2022). Efficient Matching-Based Parallel Task Offloading in IoT Networks. Sensors, 22 (18), 6906. doi: https://doi.org/10.3390/s22186906
17. Liu, L., Chen, H., Xu, Z. (2022). SPMOO: A Multi-Objective Offloading Algorithm for Dependent Tasks in IoT Cloud-Edge-End Collaboration. Information, 13 (2), 75. doi: https://doi.org/10.3390/info13020075
18. Ghenai, A., Kabouche, Y., Dahmani, W. (2018). Multi-user dynamic scheduling-based resource management for Internet of Things applications. 2018 International Conference on Internet of Things, Embedded Systems and Communications (IINTEC). doi: https://doi.org/10.1109/iintec.2018.8695308
19. Yaloveha, V., Hlavcheva, D., Podorozhniak, A., Kuchuk, H. (2019). Fire Hazard Research of Forest Areas based on the use of Convolutional and Capsule Neural Networks. 2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON). doi: https://doi.org/10.1109/ukrcon.2019.8879867
20. Podorozhniak, A., Liubchenko, N., Kvochka, M., Suarez, I. (2021). Usage of intelligent methods for multispectral data processing in the field of environmental monitoring. Advanced Information Systems, 5 (3), 97–102. doi: https://doi.org/10.20998/2522-9052.2021.3.13
21. Aburukba, R. O., AliKarrar, M., Landolsi, T., El-Fakih, K. (2020). Scheduling Internet of Things requests to minimize latency in hybrid Fog-Cloud computing. Future Generation Computer Systems, 111, 539–551. doi: https://doi.org/10.1016/j.future.2019.09.039
22. Jamil, B., Shojafar, M., Ahmed, I., Ullah, A., Munir, K., Ijaz, H. (2019). A job scheduling algorithm for delay and performance optimization in fog computing. Concurrency and Computation: Practice and Experience, 32 (7). doi: https://doi.org/10.1002/cpe.5581