Fog Computing Architecture and Their Analytical Models in Solving Problem Flows taking into account Failures and Features of Functioning

Aleksandr Matov

Institute for Information Recording of National Academy of Sciences of Ukraine, 2, Mykoly Shpaka Street, Kyiv, 03113, Ukraine

Abstract

Foggy computing complements cloud computing, bringing data processing and storage nodes closer to users. By reducing traffic, this avoids many of the problems in traditional cloud infrastructures that can arise when zettabyte data needs to be moved. At the same time, the delivery time of user task solutions is reduced. Analytical models of nebulous computing are proposed to calculate characteristics using many flows and many priorities of applications for tasks, different service disciplines and their combinations, taking into account failures and different disciplines of additional service and accumulation in queues for recovery.

Keywords

Reference architecture, nebulous computing infrastructure, node models, service characteristics, failures, priority service disciplines

1. Introduction

The developer of this relatively new nebulous computing technology is the OpenFog Consortium International Consortium. To the clouds - in order to bring these resources closer to end users and speed up decision-making processes. The OpenFog Reference Architecture (OpenFog RA) is designed for a specific class of business tasks for which cloud structures, or intelligent boundary devices, are in themselves inefficient. It complements the traditional model of cloud computing, ensuring the performance of their inherent functions at different levels of the network topology while maintaining such technological advantages as virtualization, containerization, orchestration, controllability, efficiency. Eight basic technological ones are briefly considered. principles (Pillars) of the OpenFog RA architecture, which characterize the belonging of systems to the OpenFog class: security, scalability, openness, autonomy, programmability, operability and maintainability, adaptability, hierarchy.

The development of mathematical models of nebulous calculations is an important area for identifying and improving their characteristics. The stochastic nature of the main factors and the need to quantify mass processes based on probability theory determines the use of queuing theory. Models of operation of nodes (*Fog Nodes*) with adaptation to failures are characterized by the following parameters: input flow of applications and flow of failures of nodes; discipline of application service; discipline of renewal of service of applications after restoration of the refused site; discipline of accepting applications to the queue during this recovery. The combination of one of the disciplines of service with one of the disciplines of renewal of service after the restoration of the failed node, and the behavior of the application, the service of which was interrupted by denial, sets the conditions for independent models. Six such models are considered. The models use an arbitrary law of distribution of random values of maintenance and node recovery, which provides additional opportunities in the

XXI International Scientific and Practical Conference "Information Technologies and Security" (ITS-2021), December 9, 2021, Kyiv, Ukraine
EMAIL: matov@ipri.kiev.ua (Aleksandr Matov)
ORCID: 0000-0001-7677-086X (Aleksandr Matov)

study of specific nebulous calculations. The considered models are suitable for the analysis of not only nebulous but also boundary (peripheral, extreme) nodes, as well as *logically isolated executive environments of different users of nebulous and boundary nodes*. Models are based on works [3–5].

2. The concept of nebulous computing

The concept of fog computing was formulated about ten years ago by Cisco Systems and provided for the expansion of cloud computing to the boundaries of the network. Foggy calculations complemented the centralized cloud model. The main meaning of nebulous calculations is to bring functional nodes closer to users. Bringing data processing and storage closer to end users allows you to solve many problems that arise with the exponential growth of the number of devices connected to the network.

In 2015, the international consortium OpenFog Consortium was established. Among the founders of the OpenFog Consortium - ARM Holdings, Cisco, Dell, Intel, Microsoft and Princeton University. Now it consists of about 60 organizations.

In February 2017, the consortium proposed a reference architecture for fog computing: OpenFog Reference Architecture [1]. The new IEEE 1934 standard regulates the use of reference architecture as a universal technology platform to support applications that require processing huge amounts of data, including the Internet of Things (IoT), the Industrial Internet of Things (IIoT), and artificial intelligence, 5G networks and a number of other modern technologies [1].

According to the consortium, digital innovations of the modern world – IoT, artificial intelligence, virtual reality, tactile Internet, 5G networks - can radically change production processes, business and people's lives. However, moving zetatabyte volumes of data generated by connected businesses, buildings, hospitals, and cars can lead to many problems in traditional cloud infrastructures.

The transfer of huge amounts of data to the clouds, their processing, the formation of control effects and their delivery in a reasonable time require very high productivity from cloud resources and the widest bandwidth from the network infrastructure. However, the construction, maintenance and development of such systems is associated with huge costs.

3. Foggy computing architecture

The reference architecture of nebulous computing complements the traditional model of cloud computing, ensuring the performance of their inherent functions at different levels of the network topology while maintaining such *technological advantages as virtualization, containerization, orchestration, controllability, efficiency*. This architecture eliminates the limitations of centralized cloud solutions, providing the necessary resources and communication channels for specific tasks [1].

The OpenFog Reference Architecture (OpenFog RA) is focused on a class of business tasks for which cloud structures or intelligent boundary devices alone are not efficient enough. It complements the traditional model of cloud computing, ensuring the performance of their inherent functions at different levels of the network topology while maintaining such technological advantages as virtualization, containerization, orchestration, controllability, efficiency.

The OpenFog RA architecture is based on eight basic technological principles (Pillars), which characterize the belonging of systems to the OpenFog class: security, scalability, openness, autonomy, programmability, operability and maintainability, adaptability, hierarchy [1, 2].

The OpenFog RA architecture allows you to create computing environments equipped with a wide range of *security features* that are used in all their components - from "things" and IoT devices to nebulous structures and cloud environments. Infrastructure protection has a set of requirements that must be met when developing solutions based on a reference architecture. Above all, it is *confidentiality, anonymity, integrity, trust, certification, verification and measurement*. Compliance with these requirements ensures that OpenFog solutions will be deployed in a secure computing environment that protects nodes, network components, management processes and orchestras.

Fog Nodes are physical and logical components that perform computational functions in nebulous networks. To some extent, nodes are analogs of cloud servers. Nodes located at the network boundary perform access control and data encryption. They must ensure their contextual integrity and isolation, as well as aggregate confidential data before they are sent to the next levels.

Although nodes are unified components of nebulous networks, their architectural elements (including CPUs and GPUs, as well as programmable logic arrays and network modules) vary depending on the location of nodes in the nebulous hierarchy and the functions they perform.

Nodes are able to form porous structures that balance the load, increase fault tolerance and reduce cloud traffic.

In more complex structures that support FaaS (Fog as a Service) service models, trust chains are formed from node to node and extend to cloud components. Since nodes can be dynamically created and dismantled, the software and hardware resources used for this must be trusted and certified.

Scalability is ensured by the dynamic compliance of the technological capabilities of foggy environments with business requirements, taking into account such factors as workload, productivity, cost indicators. Scalability involves making changes to individual network nodes by adding hardware and software, increasing the number of nodes at or near particularly busy levels, or reducing their number as needed, and adding storage systems and analytics tools. This allows you to increase performance in foggy structures, change the size of networks while increasing the number of applications, "things" or end users, expand the functionality of reliability and security. In addition, hardware configurations of node components and software are modified to adequately support applications, management infrastructure, and the orchestration of a large number of network-connected things and objects.

Openness is a fundamental principle that contributes to the formation of a large-scale nebulous computing ecosystem that deploys and supports IoT platforms and applications. It prevents the limitation of the number of suppliers and the predominance of proprietary "branded" solutions, which can lead to higher prices for products, reduce their quality and slow down the innovative development of systems. Thanks to openness, nodes can be placed in any segment of foggy networks, expand networks by adding nodes, use software-configured nodes that are dynamically formed.

Openness provides interoperability, supports the construction of a built-in infrastructure, giving downloaded applications the opportunity to use free resources, allows you to implement the principle of location transparency (Location Transparency). Transparency guarantees nodes their location at any hierarchical level of the nebulous network, and "things" IoT - optimization of network connections and the choice of the most convenient routes for access to computing and other resources.

Autonomy is determined by the ability of nodes to perform nebulous networks of the functions offered to them in the event of failure or lack of external services supporting them. In the OpenFog RA architecture, this principle applies to all levels of the network hierarchy. For example, in the case of operational autonomy, centralized cloud decision-making is not the only possible option. Due to the autonomous implementation of nodes, such functions are performed by local nodes at the network boundary on the basis of data processed by them.

Other areas where autonomy is needed include identifying and registering network connections, orchestration, management, and security.

Autonomy is one of the most important factors in improving the economic efficiency of nebulous structures, as it reduces the cost of transmitting large amounts of raw data to the cloud.

Programmability, which must be supported by hardware components, provides a high level of adaptability of programs deployed in foggy environments. This allows you to fully automate the re-setting of tasks to be performed by fog nodes or clusters consisting of multiple nodes. This result is achieved through the use of programming programming interfaces, which are described by universal interfaces of computing or accelerating components.

Programmability provides the formation of an adaptive infrastructure that meets the requirements of different scenarios for the deployment of IoT applications, as well as allows you to optimize the use of available resources, which, among other things, uses containerization. In the case of implementation in nebulous structures of service models, programmability allows you to logically isolate the execution environments of different users. It automates security updates and allows you to respond more quickly to emerging threats.

The *performance and maintainability* of foggy infrastructures is an integral attribute of modern system architecture, helping to reduce downtime, reduce the risk of business process disruption and simplify technical support.

The technologies required for their implementation, which are the most important components of the OpenFog RA architecture, include hardware systems, software and operations. They are especially needed when creating foggy infrastructures that operate in an industrial environment, an unfavorable natural environment, or remote solutions deployed in remote areas that are difficult to access.

With the OpenFog RA architecture, it is possible to bring data processing closer to the sources of data generation so that operational decisions can be made as soon as the data is translated into meaningful context. This property of architecture, its developers called *adaptability* (Agility). Adaptability allows you to make strategic decisions at different levels of the hierarchy of nebulous structures, quickly implement innovations and scale within the overall infrastructure.

Adaptability allows IoT decision makers to optimize the placement of applications that are components of the decision-making system. Data from sensors and other devices are generated in the form of arrays of different volumes, which arrive at different (not predetermined) times and thus create a significant amount of network traffic. Extract useful information from them and create a useful context, including recommendations for business, in most cases it becomes possible only after aggregation, comparison and analysis of data.

Such actions can be performed at the cloud level, but this significantly increases both the delays associated with the transmission of large amounts of data over long distances, and the cost of paying for communication channels.

OpenFog RA complements traditional cloud architectures, including *hierarchical support*. Each level of the hierarchy provides support for a set of functions required for the operation of IoT systems. The computing resources of the reference architecture can be represented as components of a logical hierarchical infrastructure that meets the requirements for the deployment of complex "end-to-end" IoT systems. Depending on their scale and tasks, the hierarchy may include a network of interconnected intelligent systems located at different physical or logical levels of architecture. At the same time, one physical system can handle IoT tasks, as provided by the principle of autonomy.

Monitoring and control are performed by microcontrollers: they are responsible for checking the status of ongoing processes, generating three-volume signals, launching applications to attract staff attention or automatically correcting the situation when there are significant deviations from the set values.

At the level of operational support, the analysis of telemetry data flows is performed with the presentation of the obtained results. Relevant analytical capabilities are small and focus mainly on the operational aspects of the physical environment for which the IoT system is responsible. This level compares data accumulated over short periods of time with the results of online analysis of new traffic.

At the business support level, the aggregate amount of IoT platform data from all of its many systems is analyzed, and information about events is stored (according to the rules established by the organization). Processing at this level of five-byte data sets helps to extract useful information for the organization, optimize business planning, compare and improve the operational efficiency of processes, using, among other things, machine learning methods.

Foggy calculations are often called limit calculations [2]. The developers of the OpenFog Reference Architecture consider this statement to be erroneous, because nebulous structures work in conjunction with cloud structures, and boundary structures (peripheral, extreme - Edge Computing) do not interact with them. Unlike hierarchical nebulous infrastructures, which extend to multilevel distributed integrated solutions that include many connected devices, "things" and clouds, boundary systems cover a small number of levels. They involve the deployment of data-intensive applications (eg, gateways, programmable automated controllers, etc.) at the network boundary. However, the latter can be significantly inferior in functionality to the nodes of nebulous structures. Some experts consider the limit calculations as a special case of nebulous.

An example of one of the market segments is unmanned autonomous vehicles, which are controlled by many interconnected components. These nodes must interact with nodes of other machines, nodes of road infrastructure, traffic control systems and cloud applications of the highest level. Thus, the resulting distributed fog systems will cover a fairly large area [2].

The OpenFog RA architecture offers the ability to build an infrastructure that supports FaaS services that help simplify and accelerate the deployment of foggy solutions. According to the developers, FaaS will include not only such well-known services as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), but also many other services designed with specific requirements of cloud structures [1].

4. Mathematical formulation of models

The development of mathematical models of fog calculations or information systems created using fog is an important area for identifying and improving their characteristics [3-6]. Foggy calculations (FC) as well as cloud computing are objects with a high level of uncertainty in the functioning process, the main factors of which are [3, 5]:

- the probability of the flow of requests for computing resources (CR);
- the presence of the necessary CR and the randomness of the time of their use by customers;
- randomness of failures of maintenance infrastructure and time of their elimination.

The stochastic nature of the main factors and the need to quantify mass processes based on probability theory determines the use of queuing theory.

Analytical models are proposed to calculate the characteristics using different disciplines of service and their combinations, taking into account failures and different disciplines of after-sales service and accumulation in the queues at the time of recovery of the service device. The models use an arbitrary law of distribution of random values FC of maintenance and restoration of the maintenance node, which provides additional opportunities in the study of specific nebulous calculations. Models are based on works [3–5].

As previously noted, all computing, regulating, control functions in nebulous networks are performed in nodes that are analogous to servers in cloud structures. Moreover, thanks to the FC Pillar principle of autonomy, centralized cloud decision-making can be performed by nodes based on the data processed in them. Therefore, the FC (Fog Node) in the simulation system of queuing (SQ) is considered as a service device.

At the input of the node, as a single-channel SQ with anticipation, there are *N* Poisson streams of different types of applications to solve problems with intensity $\lambda_i i = \overline{1, N}$. The threads are numbered in descending order of importance of the requests, the requests of the i-th stream have the i-th priority in the service. Application service time is a random variable $B_i(t)$ with a distribution function and two endpoints b_i and $b_i^{(2)}, t = 1, N$.

The natural process of servicing applications in the SQ is disrupted by the failure of the service device. The service device is unreliable and may fail according to Poisson's law with the parameter λ_0. The recovery time of the device is a random variable with a distribution function $B_0(t)$ and two endpoints b_i and $b_i^{(2)}, t = 1, N$. The device can fail both when servicing applications (there are two possible cases: applications are returned to the queue; applications are lost), and in the free state. Failures of the service device lead to an increase in the queue of applications and additional delays in their maintenance. One of the possible ways for SQ to adapt to unproductive failures of computing resources is to prioritize the receipt of applications from various sources in the queue for these resources at the time of their diversion. Such control of the flow of flows can be achieved by feedback from the nodes to the sources of applications.

During the recovery period of the service device, the applications of some streams are accepted in the queue, and others are not accepted. This condition is set by a matrix-row of coefficients $n_i, i = \overline{1, N}$, and in the case if $n_i = 1$ the applications of the i-th stream are accepted in the queue, and if $n_i = 0$ the applications are rejected.

After the restoration of the service device, two disciplines of resumption of service are possible: from applications of higher priority and from applications whose service was interrupted by the failure of the device (provided that they are not lost during the failure).

The service of applications in the system can be organized according to the rules of relative, absolute, mixed and combined priorities.

In the case of relative priorities, the possibility of applications is taken into account only at the time of their assignment for service. The application with the highest priority is assigned to the released device and its further maintenance is not interrupted by other applications.

Absolute priorities involve interrupting the service of low-priority applications with applications received at the input of the system, higher priority. Interrupted applications in this case are returned to the queue and await additional service.

Intermediate to the disciplines discussed above is the discipline of service with combined priorities. In the case of combined priorities, the service time of all applications, except for higher priority applications, is divided into two time segments: the first segment has an absolute priority, the second - a relative one.

Mixed priorities are a combination of absolute and relative priorities, and non-priority service may be used for individual applications.

It is necessary to determine the following characteristics of application service in maintenance nodes FC: w_i - the average waiting time for the start of service of applications of the i-th stream in i-th turn; v_i - average time of stay of applications of the i-th stream in the system (response time of the system); q_i - the average number of applications of the i-th stream in the i-th queue; l_i - the average number of applications of the i-th stream in the system.

The combination of one of the disciplines of service with one of the disciplines of resumption of service after the restoration of the failed device, and the behavior of the application, the service of which was interrupted by failure, sets the conditions for independent tasks for priority SQ of the following types:

- systems with relative priorities and renewal of service on applications, the service of which was interrupted by failures (systems with relative priorities of the first type);
- systems with absolute priorities;
- systems with relative priorities and renewal of service from applications of higher priority systems (systems with relative priorities of the second type);
- systems with mixed priorities;
- systems with combined priorities;
- priority systems with losses.

Thus, the queuing system that simulates the computational process of operation of the maintenance unit FC with adaptation to failures, is characterized by the following parameters: input flow of requests and the flow of failures of the service device; discipline of application service; the discipline of resumption of service of applications after the restoration of the failed device, and the discipline of acceptance of applications in the queue during this recovery.

5. System model with relative priorities of the first type

The characteristics of the service of applications in the steady-state mode are interconnected by Little's formulas, which for systems with priority acceptance of applications in the queue during the recovery of a failed device are as follows:

$$l_i = \lambda_i^* v_i \ ,$$
$$q_i = \lambda_i^* w_i \ ,$$

where $\lambda_i^* = K_r \lambda_i (1 + n_i \rho_0)$ the intensity of the applications of the i-th stream in the system, taking into account the discipline of admission to the queue during the recovery of the failed device; $K_r = \frac{1}{1+\rho_0}$ is the probability that the service device is in good condition; $\rho_0 = \lambda_0 b_0$ is "loading" the system with failures.

The condition of steady state in systems of this class without losses is $\sum_{i=1}^{N} \rho_i^* < K_r$,where $\rho_i^* = \lambda_i^* b_i$ - the probability of employment of the device by servicing the application of the i-th stream.

In addition, the average residence time of the application of the i-th stream in the considered system $v_i = w_i + b_i(1 + \rho_0)$. Therefore, to find the necessary characteristics to calculate enough $w_i, i = \overline{1, N}$.

Follow the movement of some application coming into the system $j = \overline{1, N}$ the j-th stream, Before it gets to the device, the following must be done:

- completed renewal of the device (provided that the applications of the j-th stream are accepted in the queue during the restoration of the failed device);
- completed service on the device (if it is working) or in the queue (if the application service was interrupted by a refusal and the device is currently restored);
- all available applications from queues with numbers from 1 to j have been serviced;
- applications received again from streams with numbers from 1 to j - 1 are serviced.

For the average durations of these events, the equilibrium equation is:

$$w_j = n_j\sigma_0 + \sum_{i=1}^{N}\sigma_i + \sum_{i=1}^{j}\lambda_i^* w_i b_i(1+\rho_0) + w_j\sum_{i=1}^{j-1}\lambda_i^* b_i(1+\rho_0), \tag{1}$$

where $\delta_0 = K_r\rho_0\Delta_0$, $\delta_i = \rho_i(1+n_i\rho_0)\Delta_i(1+\rho_0)$; $n_jK_r\rho_0$ is the probability that at the time of receipt of the application of the j-th stream, the service device is restored; $\Delta_0 = \frac{b_0^{(2)}}{2b_0}$ is the average recovery time of the failed device; $\rho_i(1+n_i\rho_0)$ is the probability that the application coming into the system, the j-th stream will find in it the required application of the i-th stream; $\Delta_i = \frac{b_i^{(2)}}{2b_i}$ is the average time to service the application of the i-th flow without taking into account the failures of the service device.

After simple transformations of equation (1) we obtain the following recurrent relation:

$$w_j = \frac{1}{K_r - R_i}[n_jK_r\sigma_0 + \sum_{i=1}^{N}\rho_i\,(1+n_i\rho_0)\Delta_i + \sum_{i=1}^{j-1}w_i\rho_i^*], \tag{2}$$

where $R_j = \sum_{i=1}^{j}\rho_i^*$.

For recurrent view ratio

$$w_j = \frac{1}{K_r - R_j}(A_j + \sum_{i=1}^{j-1}w_i\rho_i^*)$$

it is easy to show that

$$w_j = \frac{K_rA_1 + \sum_{i=2}^{j}(A_i - A_{i-1})(K_r - R_{i-1})}{(K_r - R_j)(K_r - R_{j-1})}, \tag{3}$$

Using expression (3) from relation (2), we define explicitly:

$$w_j = \frac{1}{2(K_r - R_j)(K_r - R_{j-1})}[n_1K_r^2\lambda_0 b_0^{(2)} + \sum_{i=1}^{N}\lambda_i b_i^{(2)}\,(1+n_i\rho_0) + \\ +K_r\lambda_0 b_0^{(2)}\sum_{i=2}^{j}(n_i - n_{i-1})(K_r - R_{i-1})] \tag{4}$$

6. System model with absolute priorities

Absolute priorities are allowed to change the service of applications of lower priority by applications of higher priority. That is the middle hour of the j-th thread being interrupted in the system $v_j = w_j + u_j + b_j(1+\rho_0)$, where u_j — the middle hour of the j-th thread interrupting the requests in the service, which is subject to the service discipline.

Obsessively expressing for the purpose u_j. For which we know in advance the average valority of the period of employment of a non-necessary appli- ance for one i-th flow of applications.

In the old mode, the attachment can be more flexible, if there are no applications in the system, service jobs and renewals to service.

It seems that the middle hour at the fixed moment of the end of the period of work-to-stay will come before the first arrival at the warehouse system of the application of the i-th flow of the warehouse $\frac{1}{\lambda_i^*}$. For the great hour T, the service is scheduled for $T\rho_i^*$one hour, the service is renewed for $T\rho_i^*\rho_0$one hour and the last $T - T\rho_i^*(1+\rho_0)$ one for an hour. For the whole $\lambda_i^*T[1-\rho_i^*(1+\rho_0)]$ hour there is not enough time in the middle service cycle. Therefore, the middle period of employment of a nenadium will be:

$$\pi_i = \frac{T\rho_i^*(1+\rho_0)}{T[1-\rho_i^*(1+\rho_0)]\lambda_i^*} = \frac{b_i}{K_r - \rho_i^*}, \quad (5)$$

For an hour of servicing the request of the j-th thread, there will be $\Lambda_{j-1}b_j$ a retry, $\Lambda_{j-1} = \sum_{i=1}^{j-1}\lambda_i(1+n_i\rho_0)$. Skin such a rewriting - the whole period of employment will be applied with applications from streams with numbers from 1 to $j-1$. Average hour for servicing an application from the total flow

$$\sum_{i=1}^{j-1}\frac{\lambda_i^*}{\Lambda_{j-1}^*}b_i = \frac{R_{j-1}}{\Lambda_{j-1}^*},$$

where $\Lambda_{j-1}^* = \sum_{i=1}^{j-1}\lambda_i^*$.

The same applies to the formula (5) the period of employment will be applied with applications from flow-kiv with numbers from 1 to $j-1$

$$\pi_{j-1} = \frac{R_{j-1}}{\Lambda_{j-1}^*(K_r - R_{j-1})}, \quad (6)$$

and the middle hour is interrupted in the service of applications of the j-th flow, educated by the discipline of service,

$$u_j = \Lambda_{j-1}b_j\pi_{j-1} = b_j\frac{R_{j-1}}{K_r(K_r - R_{j-1})}. \quad (7)$$

Considering (7)

$$v_j = w_j + \frac{b_j}{K_r - R_{j-1}}.$$

Determine w_j from the equilibrium equation, for a system with absolute priorities may look like

$$w_j = n_j\sigma_0 + \sum_{i=1}^{j}\sigma_i + \eta_i + \sum_{i=1}^{j}\lambda_i^* w_i b_i(1+\rho_0) + w_j\sum_{i=1}^{j-1}\lambda_i^* b_i(1+\rho_0), \quad (8)$$

where, unlike formula (1), other additional insurance hour for additional service of requests from flows with numbers from 1 to j; $\eta_i = \sum_{k=2}^{j}P_k\Delta_k(1+\rho_0)$ is the middle hour of additional servicing of applications from numbers from 2 to j, the service of which was interrupted by the overdue applications of higher priorities; $P_k$is the mobility of that which is in the order of the k -th flow is interrupted by applications from the total flow, which will go to the $(k-1)$ first flows. Obviously, what

$$P_k = \lambda_k^* u_k. \quad (9)$$

After performing the incoherent transformations in echuals (8), wit improvisations in expressions (7) and (9) obsessive recursive singing for w_j:

$$w_j = \frac{1}{K_r - R_j}\left[n_j K_r\sigma_0 + \sum_{i=1}^{j}\rho_i(1+n_i\rho_0)\Delta_i + \sum_{k=2}^{j}\rho_k^*\frac{R_{k-1}}{K_r(K_r - R_{k-1})}\Delta_k + \sum_{i=1}^{j-1}\rho_i^* w_i\right]. \quad (10)$$

From relation (10), using equation (5), we determine explicitly w_j:

$$w_j = \frac{K_r}{2(K_r - R_j)(K_r - R_{j-1})}[n_1K_r^2\lambda_0 b_0^{(2)} +$$

$$+\sum_{i=1}^{j}\lambda_i b_i^{(2)}(1+n_i\rho_0) + K_r\lambda_0 b_0^{(2)}\sum_{i=2}^{j}(n_i - n_{i-1})(K_r - R_{i-1})].$$

7. System model with relative priorities of the second type

In the considered system of interruption of service of applications of the lower priority by applications of the higher priority are caused by discipline of renewal of service after restoration of the failed device. Therefore, the average residence time of the application of the j-th stream in the system

$$v_j = w_j + u_j^* + b_j(1+\rho_0),$$

Where u_j^* is the average time of all breaks in the service of the application of the j-th stream, due to the discipline of resumption of service after the restoration of the failed device.

To determine u_j^*, note that, in contrast to the system with absolute priorities, the applications of higher priority, received in the system during the service of the application of lower priority, will replace it on the device only if the service is interrupted by failure. So,

$$u_j^* = P_{OTK}(B_j)u_j \ ,$$

where $P_{OTK}(b_j)$ is the probability of interruption by the failure of the service process of the application of the j-th stream.

For calculation $P_{\mathrm{OTK}}(b_j)$, determine the average residence time of the application of the j-th stream on the device before the first interruption by denial of maintenance (b_{cpj}), because these values are related by the following relationship:

$$b_{cpj} = b_j[1 - P_{OTK}(b_j)] \ .$$

The function of distribution of time of stay of the application of the j-th stream on the device has a look

$$P_j(t) = [1 - B_j(t)]e^{-\lambda_0 t} \ ,$$

where $[1 - B_j(t)]$ - the probability that at time *t* the application of the j-th stream will not be serviced on the device; $e^{-\lambda_0 t}$ - the probability that no time will occur during time *t*.

Then

$$b_{cpj} = \int_0^{\infty} [1 - B_j(t)] \, e^{-\lambda_0 t} dt, \tag{11}$$

$$P_{OTK}\big(b_j\big) = 1 - \frac{\int_0^{\infty} \big[1 - B_j(t)\big] \, e^{-\lambda_0 t} dt}{b_j}. \tag{12}$$

The equilibrium equation w_j with respect to the considered system has the following form:

$$w_j = n_j\sigma_0 + \sum_{i=1}^{j} \sigma_i + \sum_{i=j+1}^{N} \rho_i^* \varDelta_i^* + \eta_j^0 + \sum_{i=1}^{j} \lambda_i^* w_i b_i (1 + \rho_0) + w_j \sum_{i=1}^{j-1} \lambda_i^* b_i (1 + \rho_0), \tag{13}$$

where, in contrast to expression (8), the third term takes into account the time of additional service applications from streams with numbers from j + 1 to *N;* $\varDelta_i^*$ is the average time of employment of the device for maintenance of the application of the *i*-th stream and recovery after failure that occurred during the maintenance of this application; $\eta_j^* = \sum_{k-2}^{j} P_k^* \varDelta_k^* \, (1 + \rho_0)$ is the average time of additional service of applications from the queues with numbers from 2 to j, service interruptions which are due to the discipline of renewal of service after the restoration of the failed device; $P_k^* = \lambda_k^* u_k^*$ is the probability that the queue of the *k*-th thread is in the queue, which was replaced by the request from the total thread connecting the $(k - 1)$ first streams on the device restored after failure.

From the definition $\varDelta_i^*$ it is obvious that

$$\varDelta_i^* = [1 - P_{OTK}(\varDelta_i)]\varDelta_i + P_{OTK}(\varDelta_i) b_0,$$

where $P_{OTK}(\varDelta_i)$ is the probability of interruption by failure of the process of servicing the application of the *i*-th stream.

To determine $P_{\mathrm{OTK}}(\varDelta_i)$ it is necessary to have the probability distribution function of the service time of the *i*-th stream on the device. It is known that the probability density distribution of this time

$$f_i(t) = \frac{1 - B_i(t)}{b_i}.$$

Then, taking into account this formula and expression (12):

$$P_{OTK}(\varDelta_i) = 1 - \frac{\int_0^{\infty} \left[1 - \int_0^{t} \frac{1 - B_i(x)}{b_i} dx\right] e^{-\lambda_0 t} dt}{\varDelta_i}.$$

After simple transformations in equation (13), using formula (3), we find explicitly:

$$w_j = \frac{K_r}{(K_r - R_j)(K_r - R_{j-1})} \left\{ n_1 K_r \sigma_0 + K_r \sum_{i=1}^{j} \sigma_i + K_r \sum_{i=j+1}^{N} \rho_i^* \varDelta_i^* + \right.$$

$$\left. + \sigma_0 \sum_{i=2}^{j} (n_i - n_{i-1})(K_r - R_{i-1}) + \frac{1}{K_r} \sum_{i=2}^{j} [\rho_i^* \varDelta_i^* - \sigma_i \overline{P_{OTK}} \, (b_i)] R_{i-1} \right\},$$

where $\overline{P_{OTK}}(b_i) = 1 - \frac{P_{OTK}(b_i)}{K_r}$.

8. Models of systems with mixed priorities

In systems with mixed priorities, adjacent application flows in *M* groups, between which there is an absolute, and within each - a relative priority in service. In this case, each *m*-th group of application streams contains streams with numbers from $(s_{m-1}+1)$ to $s_m, m = \overline{1, M}$. There are two possible disciplines for resuming applications from flows grouped into one group: applications that have been interrupted by device failure (type I system) and applications from higher priority applications (type II system).

Type I system model. It is easy to see that, unlike the system with absolute priorities, the application of the *j*-th stream of the *m*-th group, arriving at the input of the system, must wait in line for additional service requests from streams with numbers from 1 to s_m, and its service may be interrupted priorities in the event that $j > s_1$ (the total flow of applications that interrupts its service, includes flows with numbers from 1 to s_{m-1}). Therefore, the equilibrium equation of the relative average waiting time for the start of service of applications of the *j*-th stream of the *m*-th group will be as follows:

$$w_j^{(m)} = n_j\sigma_0 + \sum_{i=1}^{s_m}\sigma_i + \eta^{(m)} + \sum_{i=1}^{j}\lambda_i^* w_i b_i(1+\rho_0) + w_j\sum_{i=1}^{j-1}\lambda_i^* b_i(1+\rho_0), \tag{14}$$

where $\eta^{(m)} = \sum_{r=1}^{m-1}\sum_{k=s_{r+1}}^{s_{r+1}}\rho_k^*\frac{R_{s_r}}{K_r - R_{s_r}}\Delta_k(1+\rho_0)$, and the average residence time of the application of the *j*-th current of the *m*-th group in the system:

$$v_j^{(m)} = w_j^{(m)} + u_j^{(m)} + b_j(1+\rho_0), \tag{15}$$

where $u_j^{(m)} = b_j\frac{R_{s_{m-1}}}{K_r(K_r - R_{s_{m-1}})}$.

Performing simple transformations in expression (15), we obtain:

$$v_j^{(m)} = w_j^{(m)} + \frac{b_j}{K_r + R_{s_{m-1}}}.$$

From equation (14), taking into account equation (3), we find the expression for explicitly w_j:

$$w_j^{(m)} = \frac{K_r}{2(K_r - R_j)(K_r - R_{j-1})}[n_1 K_r^2\lambda_0 b_0^{(2)} +$$

$$+\sum_{i=1}^{s_m}\lambda_i b_i^{(2)}(1+n_i\rho_0) + K_r\lambda_0 b_0^{(2)}\sum_{i=2}^{j}(n_i - n_{i-1})(K_r - R_{i-1})].$$

Type II system model. In contrast to the type I system, interruptions in the service of applications in this system are due to both the discipline of service and the discipline of renewal of service after the restoration of the device that failed. So

$$v_j^{(m)} = w_j^{(m)} + u_j^{(m)} + u_j^{*(m)} + b_i(1+\rho_0),$$

where $u_j^{*(m)} = b_i P_{\mathrm{OTK}}(b_i)\frac{\Delta R_{j-1}^{(m)}}{K_r(K_r - \Delta R_{j-1}^{(m)})}$, $\Delta R_{j-1}^{(m)} = R_{j-1} - R_{s_{m-1}}$, and the equilibrium equation w_j for has the following form:

$$\omega_j^{(m)} = n_l\sigma_0 + \sum_{i=1}^{j}\sigma_i + \sum_{i=j+1}^{s_m}\rho_i^*\Delta_i^* + \eta^{(m)} + \eta^{*(m)} +$$
$$+\sum_{i=1}^{j}\lambda_i^* w_i b_i(1+\rho_0) + w_j\sum_{i=1}^{j-1}\lambda_i^* b_i(1+\rho_0) \tag{16}$$

where $\eta_j^{*(m)} = \sum_{r=1}^{m-1}\sum_{k=s_{r-1}+2}^{s_r}P_k^{*(r)}\Delta_k(1+\rho_0) + \sum_{k=s_{m-1}+2}^{j}P_k^{*(m)}\Delta_k(1+\rho_0)$, $P_k^{*(m)} = \lambda_k^* u_k^{*(r)}$.

Using equations (16) and (3), it is easy to obtain the expression $w_j^{(m)}$ for explicitly.

9. Models of systems with combined priorities

In the system with combined priorities, the service time of all applications, except for the applications of the first stream, is divided into two segments (stages): on the first segment there is an absolute priority, on the second - relative. The duration of the first stage of service of the application of the k-th stream - a constant value: $z_{ik}, i = 1, k-1, k = \overline{2, N}$. In the second stage of application service, two disciplines of renewal of service are possible: from applications whose service was interrupted by the failure of the device (type I system), and from applications of higher priority (type II system).

Type I system model. The average residence time of the j-th flow application in the system $v_j = w_j + u_j + b_j(1+\rho_0)$, where u_j is the average time of all interruptions in the service of the j-th flow application, due to the action of absolute priority in the first stage of its maintenance.

We obtain an expression for definition u_j. During the service of the application of the j-th current in the first stage there will be $\sum_{i=1}^{j-1}(z_{ij} - z_{i+1,j})\Lambda_i$ interrupts, and at each time period $(z_{ij} - z_{i+1,j})$ of service of the application, the interrupt flow combines i the first flows. Using expression (6), similarly to systems with absolute priorities, we find u_j:

$$u_j = \sum_{i=1}^{j-1}(z_{ij} - z_{i+1,j}), \tag{17}$$

The equilibrium equation for this system is as follows:

$$\begin{gathered} w_j = n_j\sigma_0 + \sum_{i=1}^{j}\sigma_i + \sum_{i=j+1}^{N} P_{\text{OTH}ji}\,\Delta_{\text{OTH}ji}(1+\rho_0) + \eta_j + \\ + \sum_{i=1}^{j}\lambda_i^* w_i b_i(1+\rho_0) + w_j\sum_{i=1}^{j-1}\lambda_i^* b_i(1+\rho_0), \end{gathered} \tag{18}$$

where; $\eta_j = \sum_{k=2}^{j}\sum_{i=1}^{k-1} P_{ik}\,\Delta_{ik}(1+\rho_0)$; $P_{\text{OTH}ji} = \lambda_i(1+n_i\rho_0)x_{ji}$ is the probability that the application received in the system of the j-th stream, will find in it underserved in the second stage of the application of the i-th stream; $\Delta_{\text{OTH}ji} = \frac{x_{ji}^{(2)}}{2x_{ji}}$ is average time of additional service at the second stage of applications of the i-th stream in relation to applications of the j-th stream; $x_{ji}, x_{ji}^{(2)}$ is respectively, the first and second moments of the duration of the second stage of service of the application of the i-th stream in relation to the applications of the j-th stream; P_{ik} is the probability that the queue is the application of the k-th stream, the service of which at the first stage was interrupted by the application of the i-th stream; $\Delta_{ik} = b_k - \frac{z_{ik}}{2}$ is the average time of service of the application of the k-th stream, the service of which was interrupted at the first stage by applications from the total flow, combining the i first flows.

Values x_{ji} and $x_{ji}^{(2)}$ can be found by the following formulas:

$$x_{ji} = b_i - z_{ji},$$
$$x_{ji}^{(2)} = b_i^{(2)} - 2z_{ji}b_i + z_{ji}^{(2)}.$$

Let's define P_{ik}. Taking into account expression (17), the average time of interruptions in the service of the application of the k-th flow, due to interruptions of its service by applications from the total flow, combining the first i flows:

$$u_{\Sigma i,k} = \sum_{n=1}^{i-1}(z_{nk} - z_{n+1,k})\frac{R_n}{K_r(K_r - R_n)} + z_{ik}\frac{R_i}{K_r(K_r - R_i)}.$$

Then:

$$P_{ik} = \lambda_k^*\left(u_{\Sigma i,k} - u_{\Sigma(i-1)'k}\right) = \lambda_k^* z_{ik}\frac{\rho_i(1+n_i\rho_0)}{(K_r - R_i)(K_r - R_{i-1})}.$$

Using equations (3) and (18), it is easy to obtain $w_j$explicitly:

$$w_j = \frac{1}{2(K_r - R_j)(K_r - R_{j-1})} \sum_{i=1}^{j} [K_r^2 \lambda_0 b_0 (n_i - n_{i-1}) + + \lambda_i (1 + n_i \rho_0) b_i^{(2)} +$$

$$+ \sum_{k=i+1}^{N} \lambda_k (1 + n_k \rho_0) x_{ik}^{(2)} - - \sum_{k=1}^{N} \lambda_k (1 + n_k \rho_0) x_{i-1,k}^{(2)} + \sum_{k=1}^{i-1} P_{ki} (2b_k - z_{ki})](K_r - R_{i-1}).$$

Type II system model. In contrast to the type I system, in the system under consideration,

$$v_j = w_j + u_j + u_j^* + b_i (1 + \rho_0) ,$$

where u_j^* is the average time of interruptions in the service of the application of the *j*-th stream, due to the discipline of renewal of service after the restoration of the failed device in the area of relative priority.

Obviously

$$u_j = \sum_{i=1}^{j-2} (z_{ij} - z_{i+1,j} \frac{\Delta R_{i+1}^{j-1}}{K_r (K_r - \Delta R_{i+1}^{j-1})} P_{\text{OTK}} (z_{ij}, z_{i+1,j}) + x_{1j} \frac{R_{j-1}}{K_r (K_r - R_{j-1})} P_{\text{OTK}} (x_{1j}) \ ,$$

where: $\Delta R_{i-1}^{j-1} = R_{j-1} - R_{i-1}$; $P_{\text{OTK}}(z_{ij}; z_{i+1}), P_{\text{OTK}}(x_{1j})$ - according to the probability that the service device will fail on time $(z_{ij} - z_{i+1,j})$ and x_{1j} intervals.

Failure probabilities $P_{\text{OTK}}(z_{ij}, z_{i+1,j}), P_{\text{OTK}}(x_{1j})$ can be calculated by the following formulas:

$$P_{\text{OTK}}(z_{ij}; z_{i+1,j}) = e^{-\lambda_0 z_{i+1,j}} - e^{-\lambda_0 z_{ij}} \ , P_{\text{OTK}}(x_{1j}) = 1 \frac{\int_0^\infty [1 - F_{1j}(t)] e^{-\lambda_0 t} dt}{x_{1j}},$$

where $F_{1j}(t)$ is the function of the distribution of the duration of the second stage of maintenance of the *j*-th stream in relation to the applications of the first stream.

The function $F_{1j}(t)$ can be determined using the direct Laplace transform of the density distribution of the duration of the second stage: $f_{1j}(p) = b_i(p) e^{z_{1j} p}$, where $b_i(p)$ is the direct Laplace transform of the function $\frac{d}{dt} B_j(t)$.

The equilibrium equation w_j for in this system is as follows:

$$w_j = n_j \sigma_0 + \sum_{i=1}^{j} \sigma_i + \sum_{i=j+1}^{N} P_{\text{OTH}ji} \Delta_{\text{OTH}ji}^* + \eta_j + \eta_j^* +$$

$$+ \sum_{i=1}^{j} \lambda_i^* w_i b_i (1 + \rho_0) + w_j \sum_{i=1}^{j-1} \lambda_i^* b_i (1 + \rho_0) \ ,$$

where $\eta_i^* = \sum_{k=2}^{j} \sum_{i=1}^{k-1} P_{ik}^* \Delta_{ik}^*$; $\Delta_{\text{OTH}ji}^* = [1 - P_{\text{OTK}}(\Delta_{\text{OTH}ji})] \Delta_{\text{OTH}ji} + P_{\text{OTK}}(\Delta_{\text{OTH}ji}) b_0$ is the average time of employment of the device of pre-service of the application i -th flow in the area of relative priority in relation to applications j -th flow and recovery after failure that occurred during this additional service; $P_{\text{OTK}}(\Delta_{\text{OTH}ji}) = 1 \frac{\int_0^\infty \left[1 - \int_0^t \frac{1 - F_{ij}(x)}{x_{ji}} dx\right] e^{-\lambda_0 t} dt}{\Delta_{\text{OTH}ji}}$ is the probability of failure of the device in the remaining time to service the application of the *i*-th stream in the area of relative priority in relation to the applications of the *j*-th stream; P_{ik}^* is the probability that the queue of the *k*-th stream is in the queue, which was replaced by the application of the *i*-th stream on the device restored after failure$(i < k)$; $\Delta_{ik}^* = \Delta_{\text{OTH}ik}^* = \frac{x_{ik}^{(2)}}{2x_{ik}}$ is the average service life of the application of the *k*-th stream, which was replaced on the restored after failure of the device by the application of the *i*-th stream in the area of relative priority.

P_{ik}^* is determined similarly P_{ik} for type I system: $P_{ik}^* = \lambda_k^* (u_{\Sigma i,k}^* - u_{\Sigma(i-1),k}^*) = \lambda_k^* (z_{i-1,k} - z_{ik}) \frac{\rho_i (1 + n_i \rho_0)}{K_r - \rho_i^*} P_{\text{OTK}}(z_{i-1,k}, z_{ik}) + P_{ik} \frac{x_{1k}}{x_{ik}} P_{\text{OTK}}(x_{1k})$

10. Models of priority systems with losses

A distinctive feature of this class of systems is the loss of applications, the service of which was interrupted by the failure of the device. Therefore, the probability of employment of the device by servicing the application of the *i*-th stream $P_{\text{cp}i}^* = \lambda_i^* b_{\text{cp}i}$, which $b_{\text{cp}i}$ is determined from expression (11).

The condition of a steady state in systems with losses is $\sum_{i=1}^{N} \rho^*_{\text{cp}_i} < K_r$.

Consider the following types of priority systems with losses:

- system with relative priorities;
- system with absolute priorities;
- a system with mixed priorities.

System model with relative priorities. The average residence time of applications of the *j*-th stream in this system $v_j = w_j + b_{\text{cp}j}$.

The expression for the definitionw_j is obtained from the equation

$$w_j = n_j\sigma_0 + \sum_{i=1}^{N} \rho^*_{\text{cp}i} + \sum_{i=1}^{j} \lambda^*_i\, w_i b^*_i + w_j \sum_{i=1}^{j-1} \lambda^*_i\, b^*_i,$$

where $b^*_i = b_{\text{cp}i} + b_0 P_{\text{OTK}}(b_i)$ is the average time of employment of the device by servicing the application of the *i*-th stream and recovery after failure that occurred during this service; $P_{\text{OTK}}(b_i) = 1 - \frac{b_{\text{cp}i}}{b_i}$ is the probability of interruption by refusing to service the application of the *i*-th stream.

Finally explicitly:

$$w_j = \frac{n_i\sigma_0 + \sum_{i=1}^{N} \rho^*_{\text{cp}i}\, \Delta^*_i + \sigma_0 \sum_{i=2}^{j} (n_i - n_{i-1})(1 - R^*_{i-1})}{(1 - R^*_i)(1 - R^*_{i-1})},$$

where$R^*_j = \sum_{t-1}^{j} \lambda^*_i b^*_i$.

System model with absolute priorities. Using the same method of determining *wj* and *vj* as for the system with absolute priorities without losses, the following results can be obtained. The average period of employment of the device by servicing the application of the *i*-th stream and recovery after failure that occurred during this service:

$$R^*_j = \sum_{t-1}^{j} \lambda^*_i b^*_i,$$

where $\rho^{**}_i = \lambda^*_i b^*_i$.

The average time of all interruptions in the service of the application of the *j*-th stream, due to the discipline of service:

$$u_j = b_{\text{cp}j} \frac{R^*_{j-1}}{1 - R^*_{j-1}}, v_j = w_j + \frac{b_{\text{cp}j}}{1 - R^*_{j-1}},$$

$$w_j = \frac{n_i\sigma_0 + \sum_{i=1}^{j-1} \rho^*_{\text{cp}i}\Delta^*_i + \sigma_0 \sum_{i=2}^{j} (n_i - n_{i-1})(1 - R^*_{i-1})}{(1 - R^*_j)(1 - R^*_{j-1})}.$$

Model of a system with mixed priorities. By analogy with the eponymous system without losses (system with mixed priorities of type I) the necessary characteristics of the service of applications of the *j*-th stream of the *m*-th group are found by the formulas:

$$w_j^{(m)} = \frac{n_1\sigma_0 + \sum_{i=1}^{S_m} \rho^*_{\text{cp}i} + \sigma_0 \sum_{i=1}^{j} (n_i - n_{i-1})(1 - R^*_{i-1})}{(1 - R^*_j)(1 - R^*_{j-1})},$$

$$v_j^{(m)} = w_j^{(m)} + \frac{b_{\text{cp}j}}{1 - R^*_{S_{m-1}}}.$$

11. Conclusions

1. Analytical models for calculating characteristics using many streams and many priorities of applications for solving problems, different service disciplines and their combinations, taking into account failures and different disciplines of after-sales service and accumulation in queues for the time of service recovery node FN - Fog Node. The models use an arbitrary law of distribution of random values of maintenance and restoration of the maintenance node FN, which provides additional opportunities in the study of specific nebulous calculations. Models are based on works [3–5].

2. Models suitable for the analysis of not only nebulous but also boundary (peripheral, extreme) nodes, as well as logically isolated executive environments of different users of nebulous and boundary

nodes are considered. The last possibility is provided by the programmability of implementation in nebulous and boundary structures of service models.

3. The efficiency of fog nodes can be determined by the quality of organization of computational processes, which are quantified by efficiency indicators [3, 4, 6], based on estimating the average or maximum time of tasks in the system, response time of the computer system, delays allowable deadlines, etc. The obtained formulaic expressions of characteristics of different models are suitable for evaluating the efficiency of nebulous nodes.

References

[1] OpenFog Reference Architecture for Fog Computing, 2017. URL: https://www.iiconsortium.org/pdf/OpenFog_Reference_Architecture_2_09_17.pdf

[2] Alexey Chernobrovtsev. Foggy infrastructure standardization. Journal of Network Solutions, 2018. URL: https://www.osp.ru/lan/2018/04/13054572

[3] A. Y. Matov, V. N. Shpilev, A. D. Komov, Organization of computational processes in ACS, Kyiv, 1989.

[4] A. Y. Matov, I. O. Khramova, "Problems of use and mathematical modeling of cloud computing for an integrated information-analytical system of public administration." Data Recording, Storage & Processing 12.2 (2010): 113-127.

[5] A. Matov, Mathematical models of cloud computing with absolute-relative priorities of providing of computer resources to users in conditions of functioning features and failures, in: CEUR Workshop Proceedings (ceur-ws.org). Vol-2318 urn:nbn:de:0074-2318-4. Selected Papers of the XVIII International Scientific and Practical Conference «Information Technologies and Security» (ITS 2018), 2318, 2018, pp. 150–159.

[6] A. Matov, Adaptation of cloud computing as optimization of the process of rendering services to users in the conditions of limited computing resources, in: CEUR Workshop Proceedings (ceur-ws.org). Vol-2318 urn:nbn:de:0074-2318-4. Selected Papers of the XIX International Scientific and Practical Conference «Information Technologies and Security» (ITS 2019), 2577, 2019, pp. 210–221.