Providing the Resilience and Survivability of Specialized Information Technology Across Corporate Computer Networks

Mykola Stetsyuk[0000-0003-3875-0416], Leonid Bedratyuk[0000-0002-6076-5772], Bohdan Savenko [0000-0001-5647-9979], Vasyl Stetsyuk[0000-0001-9880-2666] and Oleg Savenko[0000-0002-4104-745X],

Khmelnytsky National University, Khmelnytsky, Ukraine
mikstt777@gmail.com, leonid.uk@gmail.com, savenko_bohdan@ukr.net, swmuau@gmail.com, savenko_oleg_st@ukr.net

Abstract. The approach to determining the effectiveness of IT based on quantitative values that characterize resilience and survivability is developed and can be expanded to include other characteristic values. To ensure the resilience and survivability of IT, a system of measures has been developed which results in the acquisition of highly specialized IT for various applications, where the accompanying processes are irrational or unrealistic times with sufficiently high parameters of resilience, survivability and overall resilience. at the same time, acceptable to the financial cost of its operation. The fault tolerance of the client part is ensured by performing a set of measures, including hardware and functional redundancy: power of client PCs from a separate line with security devices; use of lightning protection devices on computer network lines organization of automatic updating of client PC system software; development of algorithms of procedures that implement the critical functions of the client part of the IT, with the inclusion of non-trivial (intellectual) block of error processing in them, which is performed in parallel with the procedure itself; the use of non-trivial data editors that include an interactive pro-procedure in their algorithm that eliminates uncontrolled manipulation of the database data by the operator; implementation of critical resources, calculation procedures with the ability to promptly select the location of their execution, which does not allow overloading of hardware. The survivability of IT is ensured by: redundancy of the server part of the IT with the territorial separation of the main and backup server; redundancy of the client part software, the feature of the backup is the fact that the reserve is not specially dedicated computers, but the performance reserve of individual client computers, on which, according to the backup plan, the client part software is installed, which redundant, which at a critical moment will be used as a regular, preventing the loss of IT functionality.

Keywords: **Keywords:** Software, Fault tolerance, Resilience, Survivability, Computer.

1 Introduction

For information technologies (IT) that provide the livelihoods of an institution or enterprise, that is, specialized in, for example, areas such as financial and economic activities, the issues of survivability and resilience are more important, especially as their quantitative parameters increase in their function. zoning (increase of users, servers, volumes of information in databases of them) and level of complexity.

Under fault tolerance, we will consider the property of the system to maintain full or partial performance in cases of failure of individual elements that are not related to external unregulated activities. The survivability of an information system means its ability to remain operable with a permissible decrease in productivity in the face of negative external influences (unregulated actions). These concepts correspond to the State Standard of Ukraine [1]. They set one goal - to ensure the availability of IT, which is achieved in different ways. The effectiveness of all IT is directly dependent on providing these parameters. One of its parameters is the time of unavailability, that is, the time when the system is unable to perform its functions within certain requirements. For different systems this time is different and may range from zero to a certain, still acceptable value. So for automated control systems of complex technological processes, the time of inaccessibility is zero. For such critical systems, the probability of non-accessibility should be zero. For specialized IT operating in corporate computer networks and serving as information support in such highly specialized subject area as financial and economic activity in various fields of application, this parameter is well above zero, but the requirements for such IT are also high enough that it is impossible perform at low parameters of fault tolerance and survivability, especially with a steady increase in the number of users, increasing the complexity of information flows and volumes of data processed.

Ensuring high efficiency of specialized IT is carried out on the basis of the implementation of the principles of resilience and survivability in them, which is an urgent scientific problem, which begins to be solved in the process of developing specialized IT.

2 Related works

Known methods and techniques for providing resilience and survivability of specialized IT are focused on their different types, applications, application features, computer deployment and implementation features in various IT components. Also, an urgent area that needs research is the impact of external factors (distributed attacks, malware) on the functioning of IT and the resilience and survivability.

In [2], information technology was proposed to evaluate the structural reliability of technical objects, the structure of which corresponds to one of the known types of neural networks. The structure of information technology contains a morphological model, which allows to shape and change the structure of the object model studied by the rules, based on the determination of the probability of failure-free operation in the

theory of reality. The model developed allows you to modify the object model, which takes into account the resilience of IT.

In [3, 4] the reliability of information systems is considered. An approach based on risk assessment and risk mitigation is used to enhance the reliability of information systems. This approach allows for an early assessment of the risk of the software development process and identifies the most effective mitigation strategies.

In [5] it is shown how customizable software systems consist of many different, critical, non-critical and interdependent configurations. The reliability and efficiency of a configured system depends on the successful termination of communication or interaction between its configurations. Most of the time, users of configured systems are more likely to use critical configurations than non-critical configurations. The paper shows how failure of critical configurations will affect the reliability and performance of the system. To solve the problem, critical configurations that play a vital role are investigated, and a suitable candidate for failures is provided for each critical configuration. The article proposes an algorithm that identifies the optimal candidate failure for each critical configuration of the software system. Two schemes for classification of configurations are offered - critical and non-critical configurations based on: 1) frequency of configuration interactions, 2) characteristics and frequency of interactions. These schemes have played a very important role in achieving the reliability and resiliency of the software system in a cost-effective manner. Schema performance was tested using a file structure system.

In [6], clouds are considered to be an important platform for scientific work programs. However, as many nodes deploy in the clouds, managing resource reliability becomes a critical issue, especially for performing real-time, real-time workflows where deadlines are met. Therefore, cloud resilience is extremely important. PB (Primary Backup) scheduling is a popular technique for fault tolerance and is effectively used in cluster and network computing. However, applying this technique to real-time workflows in a virtualized cloud is much more complex and rarely explored. This work develops a real-time workflow failure model that extends the traditional PB model to include cloud performance. This model builds on task allocation and messaging approaches to make sure mistakes can be made while running a workflow. Also, a dynamic fault-tolerant scheduling algorithm is proposed, rather, for real-time workflows in a virtualized cloud. It has three key features: 1) it uses a backward displacement method to take full advantage of idle resources and uses overlapping tasks and VM migration for high resource utilization; 2) applies vertical / horizontal scaling technology to quickly secure resources 3) uses a vertical reduction scheme to avoid unnecessary and inefficient changes to resources due to fluctuations in workflow requests. The evaluation of the algorithm is based on synthetic workflows and workflows compiled from real scientific and business applications, and compared with six basic algorithms.

Articles [7, 8] investigate cloud applications that are considered to be components of several cloud services components that communicate with each other through web services interfaces, where each component performs certain functional capabilities. The lack of an effective failure resiliency scheme is one of the main obstacles to increasing the availability and efficiency of complex cloud systems for deployment. The paper proposes a comprehensive recovery scheme based on software rejuvenation

for cloud applications, which has three essential parts: adaptive fault detection, aging assessment, and component-based rejuvenation checkpoint. Preliminary and qualitative evaluations show that the new resiliency scheme brings improvements in the availability of cloud programs.

In [9] presented an approach for avoiding functional failures during execution in component application systems. The approach uses the internal redundancy of components to find workarounds as alternate sequences of operations to avoid crashes. The first Java prototype is presented and an evaluation plan developed as a preliminary result.

In [10] a proactive recovery scheme based on migration of services is proposed, tolerant systems are described. Active recovery is an important method of ensuring this. The main advantage of the developed proactive recovery scheme is the reduction of vulnerability in normal operation. This is achieved in two ways.

In [11], refusal tolerance is a major issue in guaranteeing the availability and reliability of critical services, as well as the implementation of applications. In order to minimize the impact of failures on the system and the execution of applications, deviations should be anticipated and acted upon. Failure tolerance methods are used to predict these failures and take appropriate action before the failures actually occur. This document discusses existing cloud computing resilience methods based on their policies, tools used, and research issues. A cloud-based virtualization-based system architecture is proposed. The proposed system implemented an autonomous rejection. The experimental results show that the proposed system can solve various software malfunctions for server-side applications in a cloud-based virtualized environment.

In [12], cloud computing offers new power and flexibility for high-performance computing applications with the provision of a large number of virtual machines for computing intensive applications. Fault Tolerance allows systems in a cloud with multiple nodes to complete computationally intensive applications at the moment of failure. The most common fault tolerance methods for such systems are checkpoint / restart. However, a checkpoint / restart increases program execution time, which increases the cost of running it. This paper introduces the framework of resiliency for high performance cloud computing. This framework proposes to use process level redundancy methods to reduce the runtime of computationally intensive applications.

In [13-16], cyber-resilience and cyber-viability are presented as closely related concepts that share similar technologies and practices. For historical reasons, these concepts have been built into different frameworks that define different constructs to describe problems and areas of solution. Cyber-resilience constructs allow you to define system requirements, identify metrics and security, and identify and analyze solutions. Identifying the relationships between cyber-resilience constructs and cyber-survivorship attributes is shown to use cyber-resilience to enhance cyber-resilience and vice versa.

In [17-22] show the impact on the resilience and survivability of IT of various types of malware and computer attacks.

Known methods and methods for providing resiliency and survivability of specialized IT are not sufficiently systematized and may not always be implemented due to the specific use and structure of specialized IT. Therefore, it is necessary to further research and develop new methods and techniques that can improve the resiliency and survivability of specialized IT, including cyber-attacks and malware.

3 Criteria for the effectiveness and sustainability of specialized information technologies in corporate computer networks

Let's introduce specialized IT in corporate computer networks with many components:

$$S_{IT} = \{S_1, S_2, \dots, S_n\}, \quad (1)$$

where S_i - i component of specialized IT in corporate computer networks, $i = 1, 2, \dots, n$, n – number of components.

For each component S_i we will apply a feature that will include all performance criteria for enterprise computer networks that need to be used for IT development in the future. In particular, such criteria will include the criteria for fault tolerance and survivability. Specify the performance criteria for a specialized IT vector whose components will be performance features that will meet specific criteria:

$$K_e = (f_1, f_2, \dots, f_m), \quad (2)$$

where f_j - j a function that sets one of the performance criteria, $j = 1, 2, \dots, m$, m - number of functions.

Given that, overall, the task of maximizing performance depends on specific criteria that may be related to each other and affect each other accordingly, while improving one's performance may impair the other. In addition, because specialized IT is made up of components that are subject to the same criteria from a given vector, the task is complicated by the fact that some IT components are different and, accordingly, the achievement of efficiency by the same set of criteria will be different. Therefore, choosing the best solution is a difficult multicriteria task. The general statement of the task of finding the best performance for specialized IT in corporate computer networks is formulated as follows:

$$\begin{cases} K_e(S_{IT}) \to max; \\ f_j(S_i) \to max, i = 1, 2, \dots, n, j = 1, 2, \dots, m \end{cases}, \quad (3)$$

In addition, some of the IT components may be functionally repetitive, depending on the tasks and deployment on corporate computer networks. This will affect the overall effectiveness of specialized IT. However, achieving performance by certain criteria in the same components of specialized IT does not have to be the same, because these components will solve different tasks or the same tasks, but at different times they will go through different stages. Keeping these features in mind is important, so we detail the task of finding the best performance for specialized IT on enterprise computer networks as follows:

$$\begin{cases} K_e(S_{IT}) \to max; \\ f_{j,q}(S_{i,p}) \to max, i = 1, 2, \dots, n, j = 1, 2, \dots, m, \\ q = 0, 1, \dots, n_q, j = 0, 1, \dots, n_p \end{cases} \quad (4)$$

where q - the number of specialized IT components in a particular corporate computer network node; j - an index for the criterion of performance of a component of specialized IT in a specific node of a corporate computer network; $q = 0, 1, \dots, n_q, j = 0, 1, \dots, n_p$; n_q - the number of identical components of specialized IT in a corporate computer network; n_p - the criterion number for the same components of specialized IT in the corporate computer network.

Let us introduce a function that will determine the maximum value of the criterion of effect:

$$F: K_e(S_{IT}) \to max; \quad (5)$$

The value of the efficiency criterion is given by the expression taking into account the weighting factors:

$$K_e(S_{IT}) = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{p=0}^{n_p} \sum_{q=0}^{n_q} \left(\alpha_{i,j,p,q} \cdot f_{j,q}(S_{i,p})\right), \quad (4)$$

where $\alpha_{i,j,p,q}$ - weighting factors.

Consider maximizing the criteria for failover and survivability in information technology configurations that are built on a client-server architecture with their provision across all systems from users (client side) to mission-critical server time. The choice to consider the client-server architecture depends on its features, which are manifested in the following: basic client-to-client functions are shared between the client and the server; the client computer's automated workplace software handles data through requests to the server software; full support for multi-user work; data integrity is guaranteed. This distinguishes it from other architectures and allows for the provision of fault tolerance and survivability to each of the units of the system separately.

The main directions for increasing the survivability and resiliency of IT are to make redundancy in the configuration of hardware and software, supporting infrastructure, redundancy of information resources (programs and da-them). In doing so, IT must meet the following basic requirements: the system must be built so that it does not have a component (resource), the failure of which will lead to a complete failure of the entire system. For real-time systems, in addition, time constraints are imposed on the result.

Consider specialized IT related to the systems of unrealistic cha-su. Therefore, time constraints are much less rigid for her than real-time systems. In view of the server-side and client-side IT implementation architecture chosen for consideration, we will accordingly consider the issue of resiliency and survivability in relation to the functions assigned to them. Despite the fact that these two parts, being components of a single, logically indissoluble IT, perform within their specific functions, ensuring resiliency for each component of IP is achieved in different ways.

There are two approaches to building fault-tolerant IT. The first approach is based on the use of fault tolerant components. Such IT provides its functions in the event of failure of subcomponents of some components. This is the simplest method, but also the most expensive, because of the use of the most expensive components - the fault

tolerant components of IT. The second way is to build fault-tolerant IT using non-fault-tolerant components. Failure in such systems is achieved through the introduction of redundancy in them through redundancy of critical links of hardware, software, intercomponent communications and special algorithms for the functioning of IT, which provide for its reconfiguration when some components fail.

The main feature of fault tolerance is the transparency of failures of its individual components for the end user. This means that the fault-tolerant system automatically changes its configuration in the event of failure. Its runtime software is looking for workarounds, trying in failure conditions to bring the executable function to a successful completion. We define the function $f_1(S_i), i = 1,2,...,n$ quantification of fault tolerance in computer systems as follows:

$$f_1(S_i) = \frac{T_{f_1(S_i),1}}{T_{f_1(S_i),1} - (T_{f_1(S_i),2} + T_{f_1(S_i),3})}, \quad (5)$$

where i - number of components of specialized IT, $i = 1, 2, \dots, n$, $T_{f_1(S_i),1}$ – time between adjacent failures; $T_{f_1(S_i),2}$ – the time it takes to detect a failure and find a way around it; $T_{f_1(S_i),3}$ – the time required to recover IT after a failure.

As can be seen from formula (7), for IT with an automatic fault tolerance system, it will approach the maximum, due to the response speed. There are no theoretical obstacles to the construction of such systems, but in practice, when implementing them, a number of important factors must be taken into account: financial costs of implementing an automatic system to ensure survivability and fault tolerance; system complexity. For IT designed for information support in a narrow specialized subject area, such as financial and economic activities of a higher education institution, it will be appropriate to abandon the automatic fault management system in favor of the automated one. With this approach, some of the costly functions of managing redundancies present in IT will be entrusted to the individual, unless it threatens possible significant losses. Then, according to formula (7), the fault tolerance will be lower than in the first case. But the solution to the problem of building IT (similarly to other design problems) is not to ensure the maximum possible fault tolerance of the system, but to find an acceptable balance of system parameters, within a certain technological basis. And also, including taking into account the requirements of the criterion "fault tolerance / cost". Let us explore the solutions to ensure IT resilience when using such a strategy. Let us analyze the factors that negatively affect the client's IT resilience. This is necessary in order to assess and develop adequate countermeasures. The scheme of influence of negative factors on the failure of the client side of specialized IT is shown in Fig. 1.

As can be seen from the proposed model, the negative factors that affect the resiliency of the client side of the IT are divided into external and internal. Among the external factors, the greatest threats are power outages and natural phenomena that can lead to failures of computer components and computer networks. In order to avoid such cases, the power of the client computers of the IT must be performed from a separate line

equipped with security devices, such as arresters. You also need to use security devices to protect against lightning storms over long lines on computer networks.

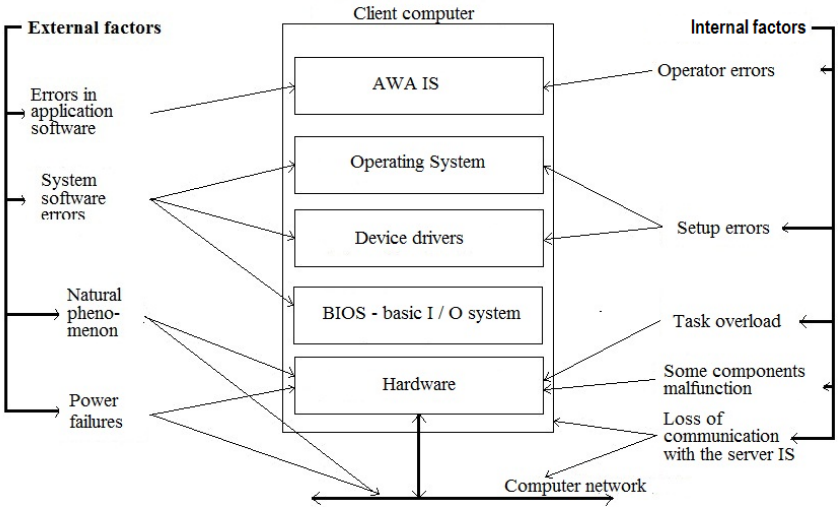

Fig. 1. The scheme of action of negative influencing factors on the fault tolerance of the client side of specialized IT

Another important factor is errors in the system software code. Previously, this factor was force majeure. Today the situation has changed and licensed operating systems can be configured with the automatic software update function enabled. This eliminates the human factor and reduces the burden on IT staff, although this does not fully solve the problem, but only reduces the likelihood of destructive behavior. The reason is that the system software is a complex system and each fix, a previously found error, does not guarantee the absence of a new one. This aspect of system software is another factor that can reduce fault tolerance. Due to its complexity, configuration errors may occur during software configuration. You can reduce its manifestations by using software with automatic adjustment, which is not always acceptable, and the involvement of more qualified personnel.

For application software, which includes client parts of specialized IT, the critical errors that occurred during the operation of jobs, are recorded together with their parameters in the system registry automatically and then used for analysis with me- the elimination of the causes that caused them. This is achieved through an approach that is based on introducing some redundancy into the software of the client part of IT. To this end, all the calculation procedures that may be critical to the functioning of the error, designed to comply with a certain type of template construction algorithm for its implementation. The essence of the algorithm is shown in Fig. 2.

In this structure, the algorithm for performing any non-trivial procedure is divided into two interacting blocks. The first block implements the function of the IT procedure, and in the second, the error handler. In the process of performing some procedure that implements one of the functions of the IT component, the two units interact with each other, transferring control of the computational process to each other until

the performed function is completed. Its essence is that the algorithm that implements the function of IT is divided by markers (label 1, ..., label n in Fig. 2) into fragments according to the principle of functional completeness.

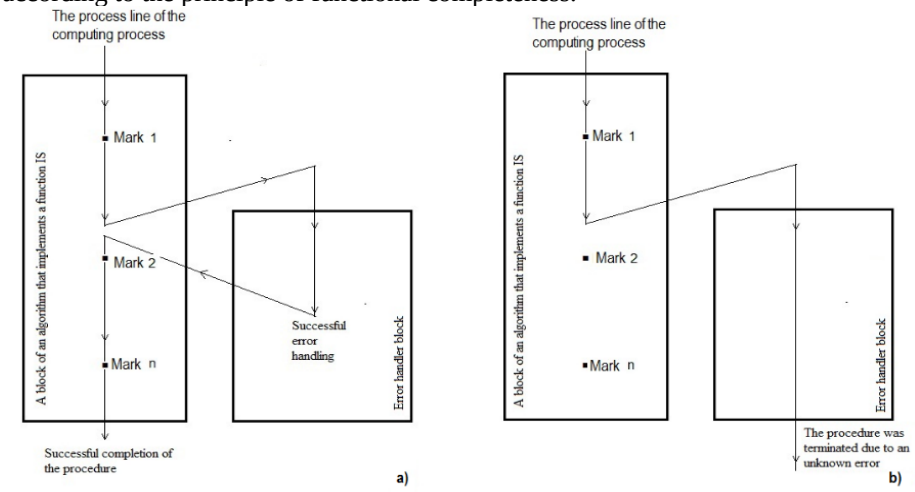

Fig. 2. Model of the algorithm procedure: a) in case of successful completion of the procedure; b) for the case where the error is unknown to the error handler

Before executing the current fragment of the algorithm, information about a hypothetically possible error (the client's workplace instance code, function code, tag number, time, etc.) is entered in the fatal error register. The following are possible developments in the future:

1. The fragment of the function algorithm was successfully executed. In this case, the information in the registry about the failed error is deleted and the computing process proceeds to the next fragment.

2. An error occurred while executing the fragment, but it was successfully localized by the error handler (Fig. 2a). In this case, the error information can also be deleted from the registry.

3. An error occurred during the execution of the fragment, which was not localized by the error handler (Fig. 2b). In this case, information about a possible error will remain in the registry.

Collected in this way, information about fatal errors that occurred in the process of the functioning of IT, allows them to classify and, in the process of further analysis, identify weaknesses in IT to address them by improving the software of the client side of IT.

The software of client IT parts during the IT life cycle, for various reasons, including due to the detection of errors in it, can change, going through their own update cycles (Fig. 3).

Consider the internal factors that affect the resilience of the client part of IT (Fig. 1) and the methods that were used to reduce it. The first of these, by frequency of occurrence, occurs because of operator errors. This problem is solved by using a standard data editor, in which all the procedures for making changes to the database are implemented using a template, the structure of which includes redundancies in the

form of blocks of algorithm, which allow to check the actions of the operator. Hereinafter, we will consider a basic element as a basic element, which is taken as a basis for the development of the entire set of editors used in IT.

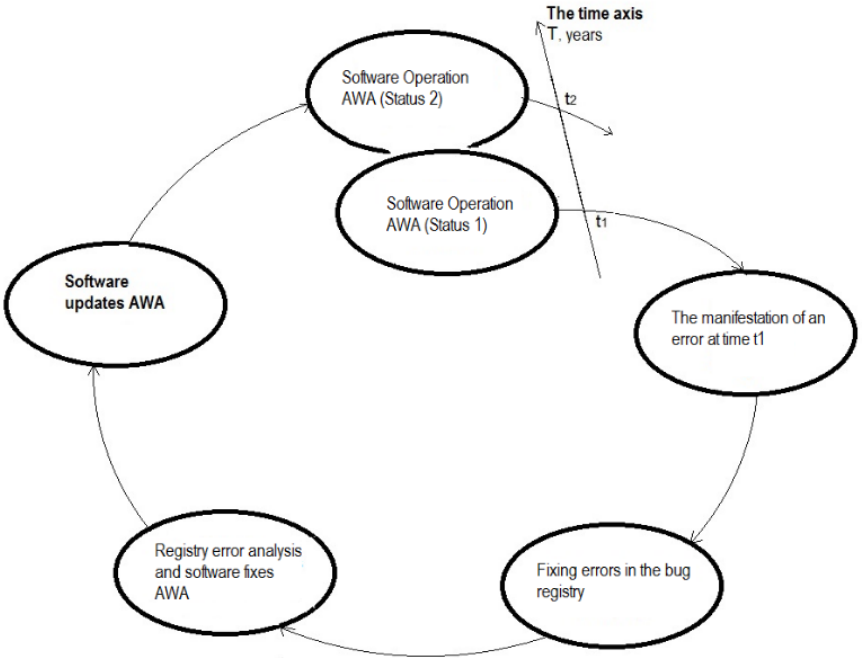

Fig. 3. Client software update cycles in the IT lifecycle

Another important internal factor that negatively affects the security is the overload of the client computer hardware platform with tasks, which can dramatically worsen the time parameters of the tasks performed by the client part of IT, or even make it impossible to work, due to depletion of technical resources . To neutralize the effect of this factor in IT, functional redundancy (Fig. 4) was applied in the development of the software, namely the part that is responsible for the implementation of "business logic".

The presence of a functional reserve of "heavy" calculation functions allows to maneuver the computing power of the IT hardware platform, in case of overloading of some of its links, thus increasing the failure-stability. Because the procedure that is functionally redundant (for example, Funk1 in Fig. 6) is developed in two variants by one algorithm, but in different software environments, to be performed in different technical means. This fact can be used to neutralize such a negative factor as the presence of an error in the application software of the client part, in the event that an error occurs in one of the variants of the procedure. This shows a positive multiplicity of the effect of functional redundancy, which increases the overall fault tolerance of information technology.

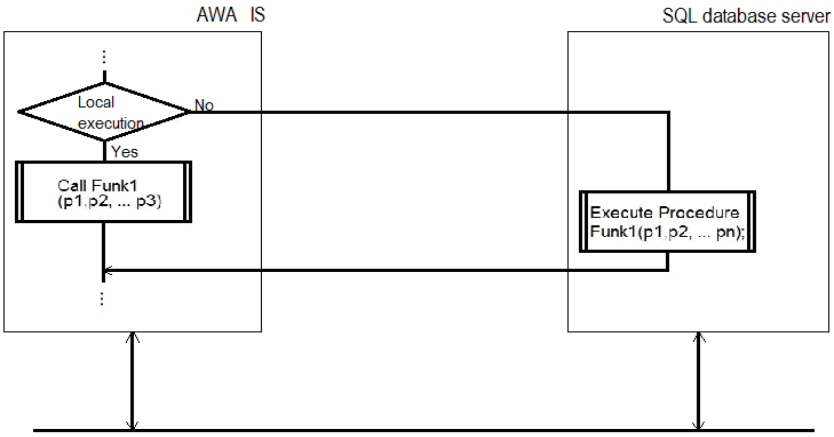

Fig. 4. Model of application of functional reservation of calculated functions of IS in environment "client - server" architecture

Thus, the fault tolerance of the client part is ensured by performing a set of measures, including hardware and functional redundancies: powering client PCs from a separate line with protection devices; use of lightning protection devices in computer network lines; organization of automatic updating of client PC system software; development of algorithms of procedures that implement critical functions of the client part of IT, with the inclusion of a non-trivial (intelligent) error handling unit, which is performed in parallel with the procedure itself; use of non-trivial data editors, which include in their algorithm an interactive procedure that eliminates uncontrolled manipulation of database data by the operator; implementation of critical resources, calculation procedures with the ability to promptly choose the location of their implementation, which does not allow overloading of hardware.

As a result of the analysis of the factors that negatively affect the resilience of the server part of the IT, the model of the action of the negative factors, shown in Fig. 5.

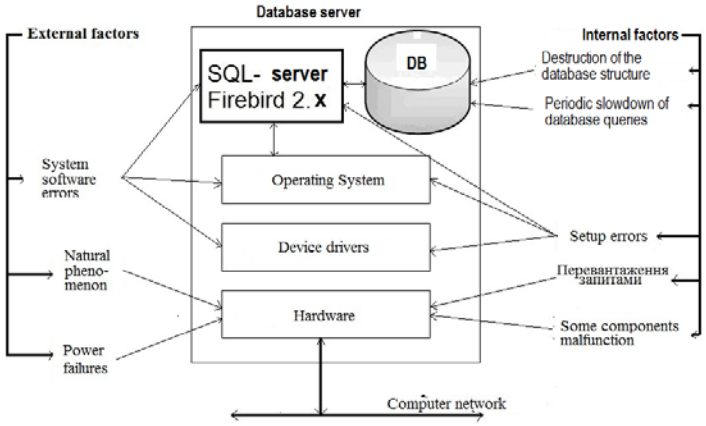

Fig. 5. Model of action of negative factors affecting the fault tolerance of the server of IS

In the presented model, all negative factors are divided into external, caused by causes that are outside the system, and internal. External factors that reduce server resiliency are neutralized in the same way as in the client side of IT. But due to its importance, this is not enough. Due to the fact that the IT server is the location of the database, where all the information processed in the system is concentrated, for him such a factor as failures in the power system is especially dangerous. This is due to the fact that the database, being the most complex system, is sensitive to violations of the technology of its handling. Sudden loss of power or failure of the server hardware due to voltage fluctuations can, with a high probability, damage the database with all subsequent negative consequences for the information. In order to prevent this from happening, a non-interruptible power supply unit with dual voltage conversion and sufficient time for server autonomous operation was introduced into the circuit of the power system. In addition, the uninterruptible power supply must have a status controller that includes an output with a serial RS-232 interface. It is necessary to send the server a signal of decay in the event that due to prolonged loss of external power, the internal power supply of the uninterruptible power supply device will be exhausted to an unacceptable limit. Upon this signal, the server will correctly terminate all applications that have been started without preventing the database from being destroyed.

No less threatening for the server part, which reduce its fault tolerance, are internal factors. Among them, the most severe in terms of consequences is the failure of hardware, namely drives. IT server drives are the most responsible part of it, the failure of which can lead not only to prolonged unavailability of information resources, but also irreversible data loss. This is due to the fact that the IT database is stored on storage devices, which in most cases are mechanical devices and, accordingly, have a smaller resource of operation than electronic circuits. Since the loss of the database is unacceptable, measures are needed to neutralize the threat of sudden loss of storage. This problem can be solved by backing up the drives. Instead of a separate drive for storing the database and other critical data, a RAID array of type 1 drives can be used. In addition, it is necessary to organize periodic diagnostics of drives, which will allow, in most cases, to identify drive problems in advance. To do this, you can use the smartmontools package, which is included in the official repositories of most distributions of the Linux operating system. It has convenient and quite flexible settings for the diagnosis schedule. This allows long-term procedures for diagnosing drives to take out of the working hours of the institution. Each diagnostic report sends to the specified email address or log file.

Another way to prevent database loss is to back up. In spite of the described measures of ensuring the resiliency of the server part of the IT, they still cannot claim to be absolute. Because the database and the entire sequence of software that ensure its operation is a complex system, the presence of errors in its operation remains quite high. To reduce the effects of destructions, such as unknown errors that can lead to database destruction, you can include a database backup subsystem in the server software loop. It operates automatically according to a schedule that records the backup frequency during office and non-working hours of the institution. The database copy repository is another computer located in a location remote from the main server.

Because the copy of the database is a fairly large array of information, so as not to depend on network traffic, the main server and the computer with the database copy repository have their own communication channel (Fig. 8). To prevent uncontrolled changes to the data in the database that could compromise the integrity of the IT data, all changes are performed under transaction management. This approach is guaranteed to ensure the transition of the database from one agreed state to another, when manipulating data.

Therefore, the fault tolerance process is continuous throughout the IT life cycle. It begins with the planning of measures to ensure the resilience of IT, which is designed and lasts until the end of its operation in general. In general, the task of ensuring the fault tolerance of the server part of IT is solved, as well as for the client part, as a set of measures to counteract the negative factors (Fig. 7). It includes: inclusion in the circuit of the subsystem of the power supply of an intelligent uninterruptible power supply unit, which interacts with the operating system of the server, providing automatic correct closing of all server applications, preventing a database crash, sudden power loss and elimination of power supply fluctuations; the use of a high-probability type 1 RAID storage array eliminates the loss of the IT database due to drive failure; automatic diagnostics of the condition of drives according to the established schedule, which allows to quickly identify the causes of future failures; organizing the work of the database backup subsystem in automatic mode, according to the schedule, with the territorial diversity of the main database and its copies, by its own network channel; using the client-side transaction subsystem software, which ensures that any manipulation of data in the database is performed with consistency of data at all times.

Vitality indicators in a complex system: multifunctionality of individual components; the existence of a single (main) purpose of the whole system; not only the possibility of information exchange between individual components, but also information interaction with users; availability of means of protection, control, diagnostics and self-organization. The task of structural survivability analysis requires the definition of: the system architecture required to fulfill the purpose of IT functioning at some point or time when undesirable effects on the system occur; requirements for particular types of system resources and their interconnection; requirements for functionality of system resources; the nature of the nature of the undesirable effects or their consequences. We define a function $f_2(S_i)$, in which $i = 1,2,...,n$ the definition of surviv bility in quantitative units in computer networks is expressed as follows:

$$f_2(S_i) = \frac{T_{f_2(S_i),1} + T_{f_2(S_i),2}}{T_{f_2(S_i),1}}, \quad (8)$$

where $T_{f_2(S_i),1}$ - time of operation of the IT process in standard mode, $T_{f_2(S_i),2}$ - time spent on survival processes, $i = 1, 2, \dots, n$.

This definition of the survivability function makes it possible to display the standard mode of operation with a unit value, and if there is a need to ensure survivability and in the case of a much longer time than the standard mode of operation, the function value will display a quantitative ordinal value.

From Fig. 6 shows how the problem of increasing the survivability of IT was realized within the framework of the developed system by structural redundancy of its main components, namely its server part. In the event of a failure of the main IT server, its functions can assume a backup, which has exactly the same settings as the main one. The main and backup servers are geographically spaced and fed from different lines. Since the failure of two servers at once is an unlikely event, it ensures high survivability of the server part of IT. Reconfiguration of a real system takes no more than 10 minutes. The copy of the database is kept up to date by the replication service, so the replacement of the main database with the database - the copy is performed without loss of information. But a slight loss of information in such a scheme is still possible. This can happen if some sensitive components of the server hardware platform fail. Typically, these are recent transactions that will be terminated due to hardware failure. And if it is a transaction to change the information in the database, then in this case the information will be lost. But since such an event in the life cycle of the information system itself is rare, such a possible amount of information loss can be neglected. After the server part is restored, the operators whose transactions were lost need to perform the last operations again to recover the lost information. In the regular diagnosis of critical hardware of the server, in most cases it is possible to detect a ripening failure and to replace the corresponding component in a timely manner. Thus, the organization of work can reduce the likelihood of failure of the server-side part of IT and thereby negate the loss of information. As can be seen from Fig. 8, in addition to performing the backup function of the primary server, the backup server serves as a data source for the WEB server through which the IT publishes information to its remote users. In addition, another function is assigned to the backup server - it serves as a repository of database copies maintained by the backup service. Server backup guarantees sufficient IT survivability in general, but does not guarantee that it will lose some of its functions related to the failure of the client computer's hardware components, which critically affect the functioning of the client part as a whole. The solution to the problem, then, is to create some reserve. Analysis of the software of the client part of IT showed that some of them have a reserve of time. Therefore, it is natural to decide to use this reserve at critical moments in the work of the client part of IT. This approach does not allow you to keep a standalone computer as a standby, but also to have stockpiles of components that reduce operating costs without losing the overall viability of the system.

Typically, software modules, as configured, are stored in the IT software repository and on those client computers where they are scheduled to be used at critical times according to the backup plan. If critical computer hardware fails to make it impossible for the client software to perform its functions, it is transferred to a suitable other computer. The time spent on reconfiguring the client side is calculated in minutes, which is an acceptable value to ensure the viability of IT that provides information support in such a subject area, such as financial and economic activities of higher education institutions.

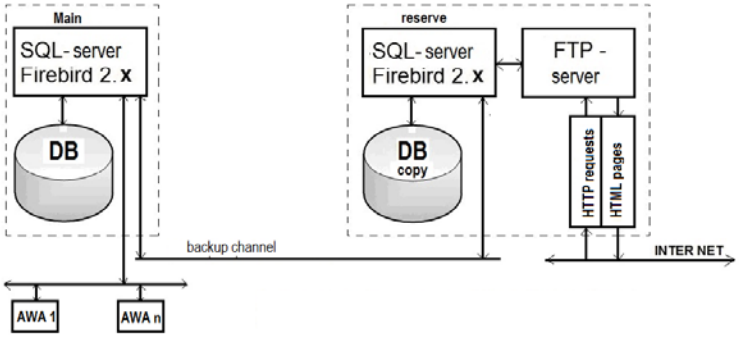

Fig. 6. Server backup scheme IS

This reconfiguration of the client part is made possible by the fact that absolutely no data is stored on the client computers running the software. And the software module itself, for convenience, is grouped into one file and does not require an installation procedure. All you have to do is copy it to another computer and it will be ready to go. This approach allows you to maintain the full functionality of IT even after the failure of several computers, which in itself has a low probability. There is only one limitation - each copy of the client-side software must be pre-registered with IT. Otherwise, an attempt to run such a program will be considered an attempt to gain unauthorized access to the system, even with the correct user credentials. IT's control over all instances of its client parts allows it to block attempts by malicious intruders who have managed to master user account data to gain access to the system. At the same time, the program mastered by the attacker does not gain access to the IT data, and the fact that such program attempts to connect to the systems is recorded in the registry of fatal errors with the corresponding data, which allows to use them to take organizational measures against the attacker.

Thus, IT survivability is ensured by: redundancy of the server-side IT with territorial separation of the main and backup server, the peculiarity of the redundancy is that the server function, at a critical moment, takes over the mirrored SQL server, which in regular mode provides the work FTP servers; redundancy of client part software, a feature of redundancy is that the reserve is not specially dedicated computers, but the performance reserve of individual client computers, which, according to the backup plan, installed the client client software, which is in the critical moment will be used as a regular, preventing loss of IT functionality.

Based on formulas (6) - (8) we obtain the value of efficiency for IT, taking into account the indicators of fault tolerance and survivability:

$$K_e(S_{IT}) = \sum_{j=1}^{m} \sum_{p=0}^{n_p} \sum_{q=0}^{n_q} \left(\alpha_{1,j,p,q} \cdot \frac{T_{f_1(s_i),1}}{T_{f_1(s_i),1} - (T_{f_1(s_i),2} + T_{f_1(s_i),3})} + \alpha_{2,j,p,q} \cdot \frac{T_{f_2(s_i),1} + T_{f_2(s_i),2}}{T_{f_2(s_i),1}} \right), \quad (9)$$

where $\alpha_{1,j,p,q}$ - coefficient for the value that determines the fault tolerance in quantitative units; $\alpha_{2,j,p,q}$ - coefficient for the value that determines the survivability in quantitative units; $\alpha_{1,j,p,q} + \alpha_{2,j,p,q} = 1$.

Similarly, the terms in formula (6) and its specification by formula (9) for the two quantities may be other indicators that characterize the effectiveness of IT.

As a result of the use of these measures was obtained IT narrowly specialized use for various applications, where the accompanying processes are unrealistic or unrealistic time with high parameters of fault tolerance, survivability and overall resilience and, at the same time, acceptable equal financial costs for its operation.

4 Experiments and evaluation

To determine how effective the proposed solutions to ensure fault tolerance and survivability, we will compare the criterion of efficiency for IT without ensuring fault tolerance and survivability and including these characteristics on the basis of formula (9).

The value of the value of the criterion of IT efficiency, which does not meet the requirements of fault tolerance and survivability, we obtain from formula (9) as follows: which constantly monitors the functioning of IT; problem situations are solved only when they are detected. In the first case, the calculation according to formula (9) can be similar and the value of the obtained value is orders of magnitude higher than the value of the criterion for IT, which provides fault tolerance and survivability. How to consider the second option, then $K_e(S_{IT}) = 1$. In this case, the relationship between the values is determined by formula (10) and allows to establish the effectiveness of the proposed solutions to ensure fault tolerance and survivability, as well as to improve efficiency by adjusting the coefficients:

$$\mu = \frac{1}{\sum_{j=1}^{m}\sum_{p=0}^{n_p}\sum_{q=0}^{n_q}\left(\alpha_{1,j,p,q}\cdot\frac{T_{f_1(s_i),1}}{T_{f_1(s_i),1}-(T_{f_1(s_i),2}+T_{f_1(s_i),3})}+\alpha_{2,j,p,q}\cdot\frac{T_{f_2(s_i),1}+T_{f_2(s_i),2}}{T_{f_2(s_i),1}}\right)}, \quad (10)$$

where the absence of specialized IT or external interruptions will mean that the time spent processing them will be zero and the ratio will be equal to one.

If a failure or external interference occurs, the value will be greater than one. The effective value is the value minimally deviated from one.

The results of ensuring the fault tolerance and survivability of specialized IT are shown in the implemented IT in Fig. 7.

For convenience, all lines of the log file fragment have been numbered and critical positions have been highlighted.

Position 19 revealed a fatal error in the operation of the network adapter "eth0" at the time of access of the user's computer with IP 192.168.168.2.

Position 35.36 closes the current session of the SWM user.

At position 37, the system notifies that reconfiguration of network devices is required.

At position 38 it is stated that the device "eth0" is switched off.

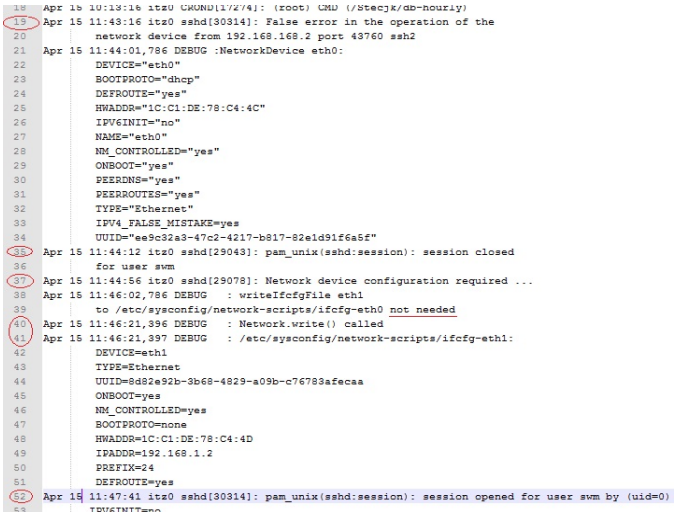

Fig. 7. Fragment of the log file of the subsystem of control of work of network devices

At positions 40,41 it is reported that the backup network adapter "eth1" is activated

At position 52 it is stated that a SWM user session has been opened which was terminated due to a failure of the eth0 network adapter.

This snippet (Fig. 8) reflects the operation of the database transaction subsystem during its backup. The backup procedure for the GBAK utility on February 4, 2019, performed a data error that led to a rollback of the transaction. Critical positions are highlighted.

Position 31. Creating a temporary database file.

Position 32-34. GBAK error message when trying to write to the [FK] field of the ORGILCSPISVSR table the default value of NULL.

Item 35. Rollback of the current transaction due to an error.

Item 40. Notification that the backup procedure was completed incorrectly.

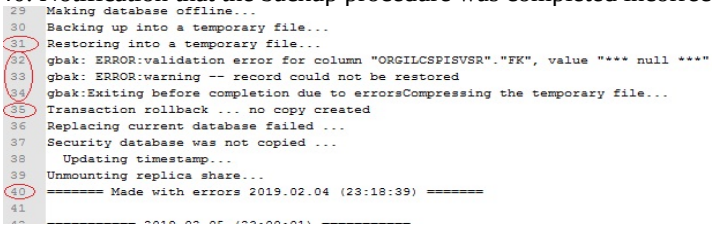

Fig. 8. A fragment of the log file of the database backup subsystem

The graphs (Fig. 9) obtained according to the calculations according to the formula (10) for the results of fault tolerance (a), survivability (b) and for the case of combined manifestations and resilience and survivability (c).

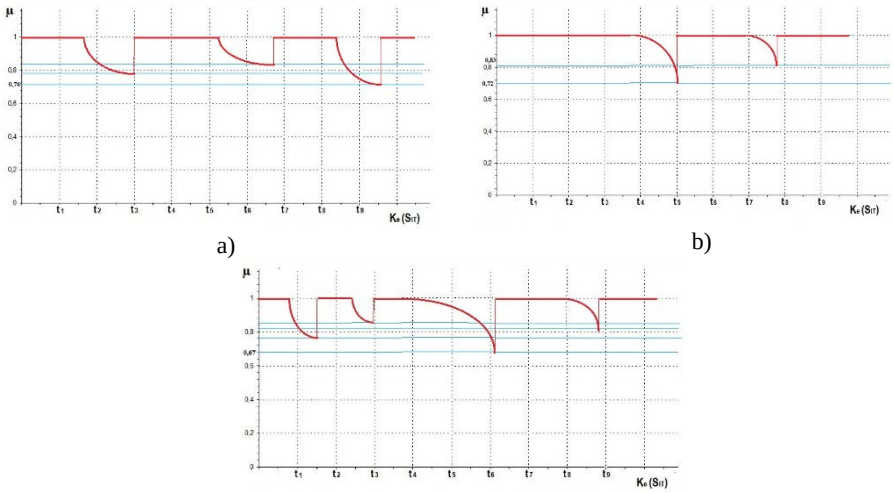

Fig. 9. A schedule for fault tolerance (a); schedule for manifestations of survivability (b); graph of reflection of simultaneous manifestations and fault tolerance and survivability (c)

The results of the study confirm the high level of resiliency and survivability in corporate computer networks, which is more than 75%.

5 Discussion and Future work

An important area of further research to improve the efficiency of IT is to develop a method to ensure effective protection of information directly in the structure of IT and computational processes that take place in computational processes. Their inclusion in the general criterion for determining the effectiveness of IT will balance such values as survivability, resilience and protection of information, expressed in quantitative form, and will become the basis for the development of specialized IT with improved performance.

6 Conclusion

Thus, an approach has been developed to determine the effectiveness of IT based on quantitative values that characterize resilience and survivability, and can be expanded to include other characteristic values. To ensure the resilience and survivability of IT, a system of measures has been developed which resulted in highly specialized IT applications for various applications, where the accompanying processes are unrealistic or unrealistic time with fairly high parameters of resilience, survivability and overall resilience and, at the same time time, acceptable equal to the financial cost of its operation.

Fault tolerance is continuous throughout the IT life cycle and includes: inclusion in the power subsystem of the intelligent uninterruptible power supply that interacts with the server operating system, ensuring automatic correct closure of all server relationships, preventing database crashes, sudden power failure and exclusion of fluctuations in supply voltage; the use of RAID array of type 1 drives, which with a high probability, eliminates the loss of the IT database due to the failure of the drive; automatic diagnostics of the state of drives according to the established schedule that allows to reveal the reasons of future failures quickly; organization of the database reservation subsystem in automatic mode, according to the schedule, with territorial distribution of the main database and its copies, on its own network channel; using the transaction subsystem of the client software, which ensures that any manipulation of the data in the database is performed with data consistency at all times.

The survivability of IT is ensured by the following methods: redundancy of the server part of IT with the territorial separation of the main and backup server, the peculiarity of redundancy is that the server function, at a critical moment, takes over the mirror SQL server, which provides FTP -servers; redundancy of client software, a feature of redundancy is that the reserve is not a dedicated computer, and the performance reserve of individual client computers, which, according to the redundancy plan, is installed software of the client client, which is the critical moment will be used as a regular, preventing the loss of IT functionality.

References

1. DSTU 3396.2-97 Protection of information. Technical protection of information. Terms and definitions. State Committee of Ukraine, Kyiv (1997) [in Ukrainian]
2. Savelyeva, O. S., Krasnozhon, O. M., Lebedeva, O. U. (2014). Using the structural fault-tolerance index in project designing. Odes'kyi Politechnichnyi Universytet. Pratsi, 2, 130–135. doi: 10.15276/opu.2.44.2014.24.
3. S. Boranbayev, S. Altayev, A. Boranbayev. Applying the method of diverse redundancy in cloud based systems for increasing reliability, in Proceedings of the 12th International Conference on Information Technology: New Generations (ITNG 2015) (Las Vegas, Nevada, 2015), pp. 796–799.
4. Boranbayev A., Boranbayev S., Yersakhanov K., Nurusheva A., Taberkhan R. (2018) Methods of Ensuring the Reliability and Fault Tolerance of Information Systems. In: Latifi S. (eds) Information Technology - New Generations. Advances in Intelligent Systems and Computing, vol 738. Springer, Cham.
5. Chinnaiah, M., Niranjan, N. Fault tolerant software systems using software configurations for cloud computing. *J Cloud Comp* **7,** 3 (2018). https://doi.org/10.1186/s13677-018-0104-9.
6. Zhu X, Wang J, Guo H, Zhu D, Yang LT, Liu L (2016) Fault-tolerant scheduling for real-time scientific workflows with elastic resource provisioning in virtualized clouds. IEEE Trans Parallel Distrib Syst 27(12):3501–3517. https://doi.org/10.1109/TPDS.2016.2543731.
7. Liu J, Zhou J, Buyya R (2015) Software rejuvenation based fault tolerance scheme for cloud applications In: 2015 IEEE 8th International Conference on Cloud Computing, 1115–1118, New York. https://doi.org/10.1109/CLOUD.2015.164.

8. Liu J, Wang S, Zhou A, Kumar SAP, Yang F, Buyya R (2016) Using Proactive Fault-Tolerance Approach to Enhance Cloud Service Reliability. IEEE Trans Cloud Comput PP(99):1–1. http://dx.doi.org/10.1109/TCC.2016.2567392.
9. Nicolo P (2013) A frame work for self-healing software systems In: IEEE 35th International Conference on Software Engineering (ICSE), 1397–1400. https://doi.org/10.1109/ICSE.2013.6606726.
10. Zhao W, Wenbing Z, Melliar-Smith PM, Moser LE (2010) Fault Tolerance Middleware for Cloud Computing In: 2010 IEEE 3rd International Conference on Cloud Computing, 67–74, Miami. https://doi.org/10.1109/CLOUD.2010.26.
11. Bala A, Chana I (2012) Fault tolerance- challenges, techniques and implementation in cloud computing, ISSN (Online): 16940814. IJCSI Int J Comput Sci 9(1). www.IJCSI.org.
12. Egwutuoha IP, Chen S, Levy D, Selic B (2012) A fault tolerance framework for high performance computing in cloud, Cluster, Cloud and Grid Computing (CCGrid) In: Proceedings of the 12th IEEE/ACM international symposium. 13-16 May, 709–710. https://doi.org/10.1109/CCGrid.2012.80.
13. S. Pitcher, "New DoD Approaches on the Cyber Survivability of Weapon Systems (25 March 2019)," 25 March 2019. [Online]. Available: https://www.itea.org/wp-content/uploads/2019/03/Pitcher-Steve.pdf.
14. D. J. Bodeau, R. D. Graubart, R. M. McQuaid and J. Woodill, "Cyber Resiliency Metrics and Scoring in Practice: Use Case Methodology and Examples (MTR 180449)," The MITRE Corporation, Bedford, MA, 2018.
15. D. Fitzpatrick, D. Bodeau, R. Graubart, R. McQuaid, C. Olin and J. Woodill, "(DRAFT) Cyber Resiliency Evaluation Framework for Weapon Systems: Foundational Principles and Their Potential Effects on Adversaries," The MITRE Corporation, Bedford, MA, 2019.
16. NIST, "Initial Public Draft of NIST SSP 800-160 Volume 2, Systems Security Engineering: Cyber Resiliency Considerations for the Engineering of Trustworthy Secure Systems," 21 March 2018. [Online]. Available: https://csrc.nist.gov/CSRC/media/Publications/sp/800-160/vol-2/draft/documents/sp800-160-vol2-draft.pdf.
17. Lysenko, S., Savenko, O., Kryshchuk, A., Kljots, Y. Botnet detection technique for corporate area network. In: Proc. of the 2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems, pp. 363-368 (2013).
18. Savenko, O., Lysenko, S., Nicheporuk, A., Savenko, B.: Metamorphic Viruses' Detection Technique Based on the Equivalent Functional Block Search. CEUR Workshop, Vol. 1844, pp. 555–569 (2017).
19. Savenko, O., Lysenko, S., Nicheporuk, A., Savenko, B.: Approach for the Unknown Metamorphic Virus Detection. In: 9-th IEEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems. Technology and Applications, pp. 453–458 (2017).
20. Pomorova, O., Savenko, O., Lysenko, S., Kryshchuk, A. and Nicheporuk, A.: A technique for detection of bots which are using polymorphic code. In: 21st International Conference, CN, Springer, Brunów, Poland, pp. 265-276 (2014)
21. Kondratenko, Y., Kondratenko, N.: Soft Computing Analytic Models for Increasing Efficiency of Fuzzy Information Processing in Decision Support Systems. Chapter in book: Decision Making: Processes, Behavioral Influences and Role in Business Management, R. Hudson (Ed.), Nova Science Publishers, New York, 41-78 (2015)
22. Balyk, A., Karpinski, M., Naglik, A., Shangytbayeva, G., Romanets, I.: Using graphic network simulator 3 for DDoS attacks simulation. International Journal of Computing. Vol. 16, Issue 4, pp. 219-225 (2017).