18. Lutsenko, I. Identification of target system operations. The practice of determining the optimal control [Text] / I. Lutsenko, E. Fomovskaya // Eastern-European Journal of Enterprise Technologies. – 2015. – Vol. 6, Issue 2 (78). – P. 30–36. doi: 10.15587/1729-4061.2015.54432
19. Lutsenko, I. Development of the method for testing of efficiency criterion of models of simple target operations [Text] / I. Lutsenko, E. Vihrova, E. Fomovskaya, O. Serduik // Eastern-European Journal of Enterprise Technologies. – 2016. – Vol. 2, Issue 4 (80). – P. 42–50. doi: 10.15587/1729-4061.2016.66307
20. Cirlin, A. M. Optimal'noe upravlenie tehnologicheskimi processami [Text] / A. M. Cirlin. – Jenergoatomizdat, 1986. – 400 p.
21. Lutsenko, I. Identification of target system operations. 1. Determination of the time of the actual completion of the target operation [Text] / I. Lutsenko // Eastern-European Journal of Enterprise Technologies. – 2014. – Vol. 6, Issue 2 (72). – P. 42–47. doi: 10.15587/1729-4061.2014.28040
22. Lutsenko, I. Identification of target system operations. 2. Determination of the value of the complex costs of the target operation [Text] / I. Lutsenko // Eastern-European Journal of Enterprise Technologies. – 2015. – Vol. 1, Issue 2 (73). – P. 31–36. doi: 10.15587/1729-4061.2015.35950
23. Lutsenko, I. Identification of target system operations. Development of global efficiency criterion of target operations [Text] / I. Lutsenko // Eastern-European Journal of Enterprise Technologies. – 2015. – Vol. 2, Issue 2 (74). – P. 35–40. doi: 10.15587/1729-4061.2015.38963
24. Bartuševičienė, I. Organizational Assessment: Effectiveness VS. Efficiency [Electronic resource] / I. Bartuševičienė, E. Šakalytė. – 2013. – Available at: http://stics.mruni.eu/wp-content/uploads/2013/06/45-53.pdf
25. Lutsenko, I. Principles of cybernetic systems interaction, their definition and classification [Text] / I. Lutsenko // Eastern-European Journal of Enterprise Technologies. – 2016. – Vol. 5, Issue 2 (83). – P. 37–44. doi: 10.15587/1729-4061.2016.79356

Досліджено концепції побудови існуючих систем моніторингу кластерних суперкомп'ютерів. Встановлено недоліки в системах моніторингу, що призводять не тільки до зниження ефективності обчислювальних кластерів, а й до порушення їх безпеки. Запропоновано підхід до створення системи моніторингу аномальних подій в суперкомп'ютерах з використанням нейронної мережі. Розроблено і описана формальна модель виявлення аномалій

Ключові слова: суперкомп'ютер, система моніторингу, виявлення аномалій, обчислювальні системи, багатоагентний підхід

Исследованы концепции построения существующих систем мониторинга кластерных суперкомпьютеров. Установлены недостатки в системах мониторинга, приводящие не только к понижению эффективности вычислительных кластеров, но и к нарушению их безопасности. Предложен подход к созданию системы мониторинга аномальных событий в суперкомпьютерах с использованием нейронной сети. Разработана и описана формальная модель обнаружения аномалий

Ключевые слова: суперкомпьютер, система мониторинга, обнаружение аномалий, вычислительные системы, многоагентный подход

UDC 004.056.2
DOI: 10.15587/1729-4061.2016.85433

DESIGNING A MONITORING MODEL FOR CLUSTER SUPER-COMPUTERS

I. Ruban
Doctor of Technical Sciences,
Professor, Head of Department
Department of Electronic Computers*
E-mail: ruban_i@ukr.net
V. Martovitsky
Postgraduate student
Department of Information Technology Security*
E-mail: martovytskyi@gmail.com
N. Lukova-Chuiko
PhD, Associate Professor
Department of Cybersecurity and
Information Protection
Taras Shevchenko National University of Kyiv
Volodymyrska str., 64/13, Kyiv, Ukraine, 01601
E-mail: lukova@ukr.net
*Kharkiv National University of Radioelectronics
Nauka ave., 14, Kharkiv, Ukraine, 61000

1. Introduction

At present, super-computer technologies solve the problems not only of scientific and technical activity but are also used in all fields of human activity. These technologies develop rapidly and have a large potential.

Current increase in computational equipment and methods of mathematical modeling provides for the possibility for the industrial and scientific research activity to reach higher level of development. Simulation of sophisticated structures, mathematical description and reproduction of natural processes, multiparametric optimization – all this is real today.

In industry, this is an increase in the competitiveness of enterprises in the world market; in science, this is an improvement in accuracy and speed when solving those problems, which used to be solved approximately. An application of super-computer technologies makes it possible to achieve better results in tackling fundamental problems of chemistry, physics, genetics, weather forecast and global change [1].

In the oil and gas industry this will make it possible to improve the accuracy of physical-mathematical calculations when simulating geophysical processes to solve the problems of exploration and extraction of minerals, as well as to solve those problems of this industry that were solved approximately or were not solved at all.

Contemporary super-computers consist of the sets of elements independent of each other, which are combined into a uniform system and can possess fairly complicated architecture, and each problem, processed in the cluster, has its internal mechanisms of parallelism.

An important role in a computational cluster is played by its system software, correct installation and configuration of which affect not only its technical characteristics (productivity, efficiency, scalability), but also such qualities as reliability and safety. In order to work with a cluster as the united system of collective use, specialized systems for clusters control are applied. Such systems work together with the basic operating system, installed in the nodes of the cluster, and offer the means of centralized control and monitoring, for the provision of trouble-free operation of the flows of tasks and allocation of resources of the cluster for solving the processed problems.

An increase in the volumes of information that passes through computational clusters and savings on staff requires the application of effective means of monitoring of computational resources, the result of which is an increase in the number of parameters, which are controlled by the system. Due to massive flows of data on measurements of different indicators, the probability of administrator failing to notice negative changes in the computational process grows. All this leads to the violations of safety in super-computer technologies. Fig. 1 demonstrates a number of cyber attacks by countries over the fourth calendar quarter of 2015 and for the first calendar quarter of 2016 [2].

Taking into account large flows of data of the measurements of different indices of computational clusters, and also quantity of cybernetic attacks in the world, by urgent task are creation of monitoring system, which on the basis of the data about the work of computational cluster could estimate the integrity of the functioning of computational process on the cluster.

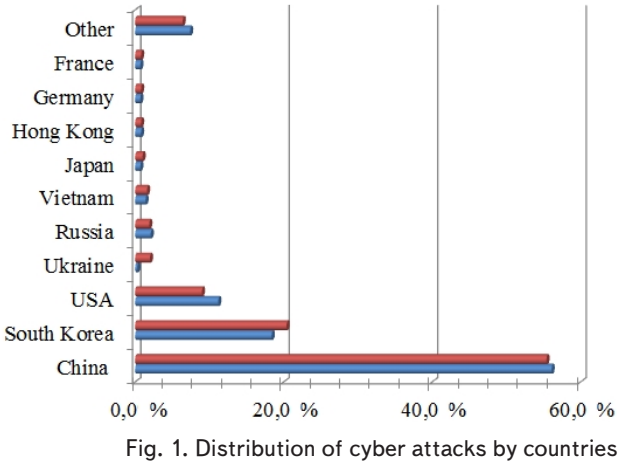


Fig. 1. Distribution of cyber attacks by countries

2. Literature review and problem statement

Situations of cyber attacks on the super-computer systems have recently appeared more frequently. Thus, in 2012, the natural oil and gas company Saudi Aramco, owned by Saudi Arabia, acknowledged that its computer systems were subjected to cyber attack. In this case, the infection spread to actually all workstations of its personnel, including their super-computer for the simulation of processes of distribution of oil in the extraction [3].

The best example of malware programs, used for the attacks on super-computer technologies rather than on individual users, is the worm Stuxnet. In reality, Stuxnet itself was a part of the set of several harmful programs, which complement each other from the point of view of their functional and the set targets [4].

An analysis revealed that hackers managed to obtain access to some of the most powerful super-computers in the world and their networks. Interesting fact is not that they succeeded in obtaining access to these super-computers but the fact that the break-in was not discovered by technical means. Therefore, monitoring system is one of the important components of the system software of a computational cluster. It makes it possible to control the level of the use of resources of the system, as well as detect the malfunctions, connected to the work of equipment, which is necessary for maintaining high degree of reliability of the clusters.

Hence an increase in the complexity of structure of a computational complex and at its scaling there appears the need in the system, capable of estimating independently condition of integrity of the process of the cluster performance.

Paper [5] proposed an approach to the analysis of complex systems, according to which the architectures of cluster computational systems were examined. This made it possible to identify vulnerable elements in their structure.

At present, there are many developed systems of monitoring. An example is the system Nagios – this is a system, designed for the monitoring of computer systems and networks. It scans specified nodes and notifies administrator when some of the services cease (or restart) the work [6]. A shortcoming in this system is:

– poor scalability;

– large interval between measurements of parameters;

– averaging of data (it is not possible to determine accurately the value of parameters, for example, a month ago);

– the means of automated expert data analysis.

Another example of similar systems is the product Zabbix, which is created for monitoring and tracking statuses of diverse services of computer network, servers and network equipment. Its drawback is poor scalability, low failure resistance and absence of the means of data analysis.

A monitoring system, proposed by author of article [7], with the aid of the multiagent approach solves the problems of scalability and failure resistance. The author proposes to create agents, which disperse on the computational nodes and collect data about productivity of the system. A shortcoming of this monitoring is the fact that only computational units are examined. Because of this, data analysis is conducted only for them and this provides for a possibility of criminals attacking not the nodes of the system, but its network.

In order to solve the problem of monitoring that is described in paper [7], it is proposed to use dynamically reconfigured distributed modular system of monitoring [8]. It is proposed here, together with an agent approach, to use statistical data collection from those systems of super-computer where for some reasons there is no possibility to create the agent. A drawback of this system is not standardized data, obtained from different sources, which leads to the complexity of their intellectual analysis.

Article [9] proposed an instrumental complex of meta-monitoring of the distributed computational environments. Its shortcoming is the fact that an expert system, developed with the use of the shell CLIPS, is used for intelligent data analysis. This expert system lacks a mechanism of self-learning. This leads to the fact that the system cannot identify new anomalies.

Author of paper [10], together with Nagios, uses neural networks for detecting anomalies, which solves the problem of learning, but it does not eliminate other drawbacks in this system.

An absence of the subsystem of monitoring, which, based on data about the work of computational cluster, could estimate the integrity of performance of computational process in the cluster, is the obvious drawback, revealed in the described papers.

3. The aim and tasks of the study

The aim of the work is the development of a model for monitoring cluster super-computers, which will allow us on the basis of parameters of performance of individual components of the system to estimate the condition of functioning of a computational cluster as a whole.

To achieve the set goal, the following tasks were formed:

– to examine peculiarities of architecture of super-computers and to identify basic components of systems;

– to determine the set of parameters for the evaluation of each element of the system;

– to design a model for the classification of condition of the system with the use of neural network technology for detecting anomalies of the computational cluster performance.

4. Principle of monitoring of cluster super-computers

A convenience in the construction of cluster computing systems consists in the fact that it is possible to flexibly regulate the required productivity of the system, adding to the cluster with the aid of special hardware and software interfaces standard serial servers until a super-computer with the necessary power is configured [11]. Clustering makes it possible to manipulate a group of servers as one system, simplifying control and increasing reliability of the system as a whole [12].

An important feature of clusters is the provision of access of each server to any block both of operative and disk memory. This task is successfully solved, for example, by combining systems of the SMP-architecture based on autonomous servers for the organization of common field of operative memory and by the use of the disk systems RAID for external memory [12].

In order to create clusters, usually either simple single-processor personal computers are used or the two-processor or four-processor SMP-servers. In this case, there are no limitations in terms of composition and architecture of the nodes. Each of the nodes can function under control of its own operating system. Standard OS are most frequently used: Linux, FreeBSD, Solaris, UNIX, and Windows. When the nodes of a cluster are not uniform, then we deal with heterogeneous clusters.

There may be two approaches when creating clusters:

– the first approach is applied when creating small cluster systems. A cluster combines full-featured computers, which continue to work as independent units. For example, computers of a training class or workstations of a laboratory;

– the second approach is used when a powerful computational resource is created on purpose. Then the system blocks of computers are placed compactly at special racks, and to control the system and for starting the tasks, one or several full-featured computers are assigned, which are called host-computers.

An analysis of cluster solutions for creating super-computers by the leading companies (IBM, Intel, HP) revealed the following elements:

– computational nodes;

– fabric of high-speed network;

– auxiliary networks (control/monitoring);

– system of data storage;

– auxiliary servers (nodes of access/compilation/monitoring, etc.);

– infrastructure (systems of uninterrupted electric power supply and cooling).

Fig. 2 demonstrates a standard structure of super-computer, built according to the principles of cluster architecture.

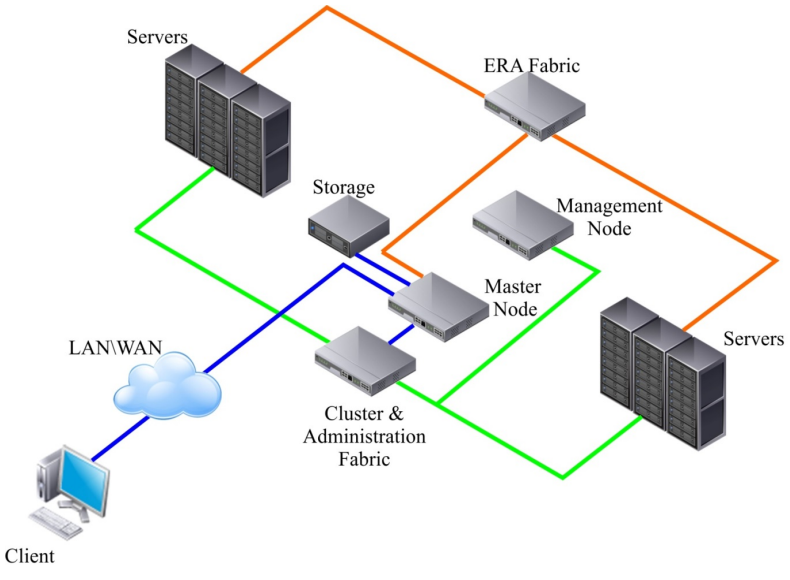


Fig. 2. Structure of a super-computer of cluster architecture

By analyzing the architecture of a cluster super-computer (Fig. 1), it is possible to draw a conclusion that the subsystem of monitoring must meet the following conditions:

– capacity of integration with the means of local monitoring of computational nodes;

– representation of applied software interfaces based on open standards for the incorporation into other software sets;

– inclusion of tools for the standardization of data, obtained from different sources;

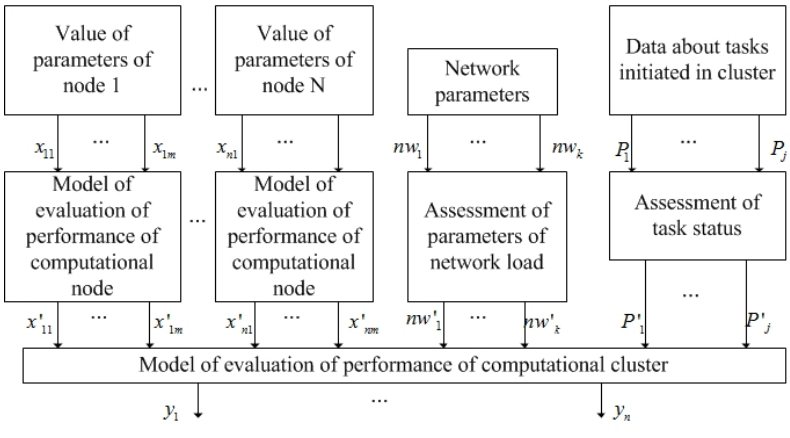
– presence of tools for the collection and data analysis about performance of equipment at each functional element of cluster;

– provision of tools for the collection and data analysis about distribution of tasks for the applications of each user;

– control of the state of computer network, applied during calculations;

– provision of tools for the automated expert data analysis of monitoring and generation of controlling influences.

Taking into account the set conditions, we formed a general structure of the monitoring system (Fig. 3).



x – controlled parameter of computational node;
nw – controlled parameter of computational network;
P – parameter of task initiated in cluster;
x' – evaluation parameter of computational node;
nw' – evaluation parameter of computational network;
P' – parameter of evaluation of task initiated in cluster;
y – parameter of evaluation of status of computational cluster

Fig. 3. Structure of the model of monitoring of super-computers of cluster architecture for detecting anomalous events

To describe the subsystem of monitoring, it is proposed to use the following cortege:

$$SM = \{\{Ag\},\{S\},NET\}, \quad (1)$$

where {Ag} is the set of agents of monitoring, intended for the collection and primary data analysis about key parameters that describe the process of functioning of computational cluster: characteristics of the initiated tasks, information on the users, network connections and their characteristics, parameters of configuration of the modules of the OS core of each node; {S} is the set of statuses of the system, which is formed based on data collected by the agents of monitoring; NET is the artificial neural network.

When describing states of the cluster, it is possible to define: current state of system $S_t \in \{S\}$ that is formed based on data collected by the agents of monitoring $Ag_i \in \{Ag\}$ in the process of "real" time and the set of templates of normal states of the system $S_{norm} \subseteq \{S\}$ compiled based on statistical data, obtained in the process of training the system when all states of the system are considered to be conditionally "safe".

Each element of the state of the system is described by the following cortege:

$$S = \{\{P\},\{X\},\{NW\}\}, \quad (2)$$

where {P} is the set of tasks initiated in the cluster, {X} is the set of the states of computational nodes, {NW} is the set of parameters of network connections.

In turn, each task $P_i \in \{P\}$ is possible to represent in the form of the set of parameters:

$$P_i = \{UID, TID, \{NODE\}, C\}, \quad (3)$$

where UID is the identifier of the user who initiated tasks, TID is the identifier of task, {NODE} is the list of nodes, which participate in processing the task, C is the time of execution of task.

Each of the states of computational nodes of cluster $X_i \in \{X\}$ can be represented in the form of cortege:

$$X_i = \{\{UIDOS\},\{PID\},\{NOS\}\}, \quad (4)$$

where {UIDOS} is the set of users in the operating system (OS) of the node, {NOS} is the set of network connections of the node, {PID} is the set of processes of OS of the node.

$PID_i \in \{PID\}$ is possible to write down in the form of parameters [13]:

$$PID_i = \{UIDOS_i, ID, \{API\}, \{CNP\}\}, \quad (5)$$

where $UIDOS_i \in \{UIDOS\}$ is the identifier of the user who initiated the process (this identifier determines rights of the process when fulfilling the API functions of the operating system), ID is the identifier of process in the system, {API} is the set of API functions of the operating system, which were assigned by the application, including parameters of these assignments, {CNP} is the set of network connections, initiated by a particular application.

Next, obtained and grouped in the course of monitoring data enter artificial neural network NET, which performs classification of the states of the system.

Functioning of multilayer artificial neural network is described by the following cortege:

$$(Net_{ij}, Out_{ij}, In_{ijk}, x_k), \quad (6)$$

where

$$Net_{ij} = \sum_k w_{ijk} In_{ijk}, \quad (7)$$

$$Out_{ij} = f(Net_{ij} - \theta_{ij}), \quad (8)$$

$$In_{ijk} = Out_{i-1k}, \quad (9)$$

$$In_{0jk} = x_k, \quad (10)$$

where x is the set of input values of neural network (a set of the states of the system); In is the set of input values of neu-

ron; Out is the set of output values of neuron; I is the number of layer of neural network; j is the number of neuron in the layer of neural network; k is the number of input of neuron; f is the function of activation of neuron; w is the weight of input of neuron; θ is the level of activation of neuron.

Effectiveness of solving a problem depends on the selection of architecture of neural network and its training. A selection of optimum architecture comes down to finding the network that solves the set problem with minimum target error:

$$E(w)=\frac{1}{2}\sum_{j=1}^{p}(y_j-d_j), \quad (11)$$

where y_i is the value of the j-th output of neural network; d_j is the target value of the j-th output of neural network; p is the number of neurons in the output layer.

The most suitable for this purpose are the following types of neural networks are: multilayer perceptron, network with radial basic elements, probabilistic, generalized-regressional or linear networks.

A neural network is trained by the method of reverse propagation. Training is conducted also with the minimization of target error (11).

An algorithm of training a neural network is shown in Fig. 4 and it includes the following steps.

1. Assign input vector (2) to the neural network input and determine values of the neural network outputs with the help of system (7).

2. Calculate auxiliary variable $\delta^{(N)}$ for the neural network output layer:

$$\delta_j^{(N)}=\left(y_i^{(N)}-d_i\right)\times\frac{dy_i}{dS_i}, \quad (12)$$

where $y_i^{(N)}$ is the value of the i-th output of neuron of the output layer; S_i is the weighted sum of output signals; N is the number of layers.

Change the weights $\Delta w_{ij}^{(n)}$ of the i-th layer of the j-th neuron:

$$\Delta w_{ij}^{(n)}=-\eta\times\delta_j^{(N)}\times x_i^n, \quad (13)$$

where η is the parameter that determines the speed of training; x_i^n is the value of the i-th input of neuron of the n-th layer.

3. Calculate expressions:

$$\delta_j^{(N)}=\left[\sum_k\delta_k^{(n+1)}\times w_{jk}^{(n+1)}\right]\times\frac{dy_i}{dS_i}, \quad (14)$$

and (12) in line with $\delta_j^{(N)}$ and $\Delta w_{ij}^{(n)}$ for other layers of neural network, n=N–1...1.

4. Correct all weights of synapses of neural network:

$$w_{ij}^{(n)}=w_{ij}^{(n)}+\Delta w_{ij}^{(n)}. \quad (15)$$

5. Determine value of the indicator of correspondence according to expression (11). If the value does not fit the assigned interval, then proceed to step 1. Determining the indices of significance criterion of the error of approximation depends on the particular parameter of the solved tasks and in future there will be an approach proposed for determining the indices of criteria.

5. Discussion of results of research into a model for the estimation of state of a computational cluster

The developed model makes it possible to monitor a cluster not only as a united computational system, but also all its components individually, which provides for the possibility to thoroughly estimate state of the system as a whole.

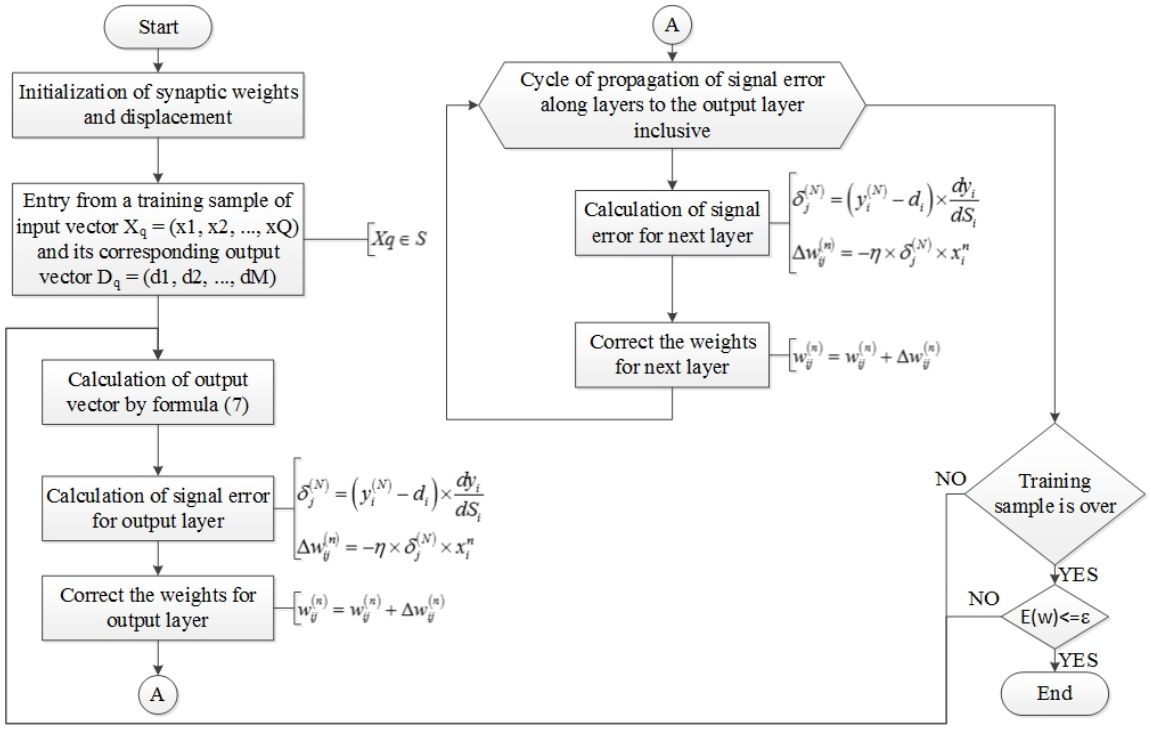


Fig. 4. Algorithm of training a neural network

We carried out an estimation of effectiveness of the proposed model only for detecting anomalous behavior of network traffic based on the set of parameters of network connections (formula (2)), which is realized by the analysis of incoming traffic with the help of neural network, represented in the form of cortege of formula (6). In order to evaluate parameters of load on a network, neural network was trained according to the algorithm, represented in Fig. 4.

Next, we conducted experimental research into performance effectiveness of the proposed model for the detection of anomalous behavior, the results of which are given in Table 1.

Table 1

Experimental research into effectiveness of the proposed model in the presence of anomalies in traffic

Number of tests	Number of valid solutions	Number of invalid solutions	
		Errors of the Ist kind	Errors of the IInd kind
10000	9696	203	101

Tests were conducted in the computational cluster, created on the workstations of a training laboratory under the action of slow DDoS-attack and attack over entire time interval. The number of tests was limited to 10000 because results of the research upon reaching this number of tests demonstrated a steady trend. Probability of making a valid solution about the existence of anomalous traffic was 97 %, errors of the first kind – 2 % and errors of the second kind – 1 %.

Thus, the proposed model is efficient for detecting anomalies in the performance of a computational cluster.

Data of research can be used for the improvement of already existing subsystems of monitoring of super-computer technologies, as well as form a foundation for creating fundamentally new neural network multi-agent system for monitoring the detection of anomalous events.

In future we plan to develop a monitoring system, which will be an adaptive subsystem of active monitoring, built on the base of artificial neural network using a multi-agent approach. This system can be used as a tool of control of the state of a cluster and actions of users in the system for the purpose of detecting potential anomalies, both internal and external.

5. Conclusions

We analyzed structure of the widely used architectures of computational clusters, which allowed us to propose two approaches for the creation of super-computers. The first one is applied for the organization of small cluster systems. The cluster combines full-featured computers that continue to work as independent units. The second approach is used when a powerful computational resource is configured on purpose. In this case, system blocks of computers are placed compactly on special racks while one or several full-featured computers are allocated to control the system and initiate the tasks. Based on the aforementioned, we established basic elements that need monitoring of anomalous events.

Shortcomings of the existing systems of monitoring of cluster super-computers are examined, conditions are determined, described in chapter 4, which a subsystem of monitoring must meet. This enabled us to propose a set of parameters for the evaluation of each element of the system.

In the paper we proposed a model for the classification of the state of the system, which makes it possible to define the sets of states depending on functional tasks, to separate processes of targeted functioning of the system from the interface processes of interaction with the network infrastructure and to use them in the neural network technology for detecting anomalies in the performance of computational cluster. This model provides for the possibility to ensure local control of parameters for each process and, based on the formed vector, to determine anomalous influence on system as a whole.

References

1. Voevodyn, V. V. Superkompiuternie tekhnolohyy v nauke, obrazovanyy y promishlennosty [Text] / V. V. Voevodyn. – Moscow: Yzdatelsvo Moskovskogo universiteta, 2012. – 232 p.
2. DDOS-ataky v pervom kvartale 2016 hoda [Electronic resource]. – Available at: https://securelist.ru/analysis/malware-quarterly/28429/ddos-ataki-v-pervom-kvartale-2016-goda (Last accessed: 22.07.2016).
3. Bronk, C. The cyber attack on Saudi Aramco [Text] / C. Bronk, E. Tikk-Ringas // Survival. – 2013. – Vol. 55, Issue 2. – P. 81–96. doi: 10.1080/00396338.2013.784468
4. Knopová, M. The Third World War? In The Cyberspace. Cyber Warfare in the Middle East [Text] / M. Knopová, E. Knopová // Acta Informatica Pragensia. – 2014. – Vol. 3, Issue 1. – P. 23–32. doi: 10.18267/j.aip.33
5. Ruban, I. V. An approach to cyber security support [Text] / I. V. Ruban // Information processing systems. – 2015. – Vol. 11. – P. 6–8.
6. Kora, A. D. Nagios based enhanced IT management system [Text] / A. D. Kora, M. M. Soidridine // International Journal of Engineering Science and Technology (IJEST). – 2012. – Vol. 4, Issue 4. – P. 1199–1207.
7. Cigala, V. Job-Oriented Monitoring of Clusters [Text] / V. Cigala, D. Mahale, M. Shah, S. Bhingarkar // International Journal on Computer Science and Engineering. – 2011. – Vol. 3, Issue 3. – P. 1333–1337.
8. Stefanov, K. Dynamically Reconfigurable Distributed Modular Monitoring System for Supercomputers (DiMMon) [Text] / K. Stefanov, V. Voevodin, S. Zhumatiy, V. Voevodin //Procedia Computer Science. – 2015. – Vol. 66. – P. 625–634. doi: 10.1016/j.procs.2015.11.071
9. Sydorov, Y. A. Ynstrumentalnii kompleks metamonytorynha raspredelennikh vichyslytelnikh sred [Text] / Y. A. Sydorov, H. A. Oparyn, V. V. Skorov // Parallelnie vichyslytelnie tekhnolohyy. – 2014. – P. 159–167.
10. Tarasov, A. G. Integration of computing cluster monitoring system [Text] / A. G. Tarasov // Proc. of the First Russia and Pacific Conference on Computer Technology and Applications (RPC 2010), 2010. – P. 221–224.
11. Tirenko, A. IT na rubezhe epokh [Text] / A. Tirenko // Otkritie systemi. SUBD. – 2016. – Vol. 1. – P. 46–47.
12. Nemniuhyn, S. A. Parallel′noe prohrammyrovanye dlia mnohoprotsessornikh vichyslytel′nikh system [Text] / S. A. Nemniuhyn. – St. Petersburg, 2002. – 255 p.
13. Olad'ko, A. Iu. Podsystema monytorynha y audyta ynformatsyonnoĭ bezopasnosty v operatsyonnoĭ systeme Linux [Text] / A. Iu. Olad'ko // Yzvestyia Iuzhnoho federal noho unyversyteta. Tekhnycheskye nauky. – 2012. – Vol. 137, Issue 12. – P. 22–28.