ADVANCES IN CYBER-PHYSICAL SYSTEMS

Vol. 4, No. 1, 2019

RESEARCH AND DESIGN OF THE MULTIFUNCTIONAL CYBER-PHYSICAL SYSTEM OF TESTING COMPUTER PERFORMANCE IN WAN

Iryna Pasternak

Lviv Polytechnic National University, 12, Bandera Str, Lviv, 79013, Ukraine.
Author's e-mail*: pas_irusj@ukr.net*

Submitted in 2019

Abstract: A multifunctional cyber-physical system for monitoring and testing remote computers in the WAN has been developed. This cyber-physical system has been built using microservice architecture. The system has used a website as a graphical interface, which in turn communicates with the main query separator, namely a web server. In addition, the database and the AES-256 encryption algorithm has been used to simplify data work, and to increase system security from external interventions. The algorithms of processing requests, algorithms of checking the efficiency of different modules of the computing node have been developed. A system of internal communication has been designed. The links between database tables have been designed. The utilities for various custom actions have been developed. The user has also provided the ability to group users and computing nodes to simplify the process of analyzing data on the performance check.

Index Terms: **module, computer network, global network, web-service, cyber-physical system.**

I.INTRODUCTION

Today, modern world cannot imagine itself without computer systems. Most companies work with the use of computers and, accordingly, they are accompanied by the emergence of health check systems. Most of these systems are set up to check the overall performance and need to either directly connect a specialist to this or install special software that will allow to remotely test the performance of computer systems. Nowadays, there is a lack of such software. Some software systems only allow to monitor certain parts of the computer. It may be monitored by the network status, and the status of disks or the current processor load. There are also programs that contain a set of tools to test performance. Such systems are called multifunctional, since the range of their capabilities does not end only with testing the performance of the specific computer part. Often, you can find programs that allow the system not only to check the performance, but also to carry out remote settings.

A modern computer needs to check the performance of several parts at once. For example, testing the performance of the system consists in analyzing processor load and disk space, being able to perform certain manipulations with the drives, analyzing the network traffic passing through this computer. These three parts actually build the foundation of computer system stability. Often, precisely due to the proper and in-depth health checks, it is possible to prevent problems such as abrupt system shutdown, due to lack of available resources, or disk overflow, system crash through the flow of broken and incorrect network packets. In addition to simple health check through use of software, it is also necessary to carry out a POST rewrite of working capacity using the system tools of the computer system. Testing the system's performance is quite a challenge, because of the amount of resources, tasks, whose periods of time increase and, accordingly, require improvement of the previous algorithms, that need to be monitored.

This article discusses the multifunctional cyber-physical system that will allow testing and tuning performance of computer systems using Windows or Linux systems. The multifunctionality of a cyber-physical system will consist in the fact that the system will be able both to collect information of the system as a whole, and its settings. It is also possible to call various procedures. The system will be able to work with the file system, namely, the ability to create, modify, view, and delete any files.

Multifunctional cyber-physical system is built using micro-service architecture. Using this architecture allows to have an independent system. That is, if the module that is responsible for analyzing the processor load of the system, stopped, or was on restart, then it will not affect the overall system. Only the functionality for module, which is responsible, will not be available. It allows to distribute the functionality of several processes, since the developed cyber-physical system also provides for a service monitor, that is, a separate service that will check one of the system services for a certain period of time.

The developed cyber-physical system also keeps a log of actions, that is, it will always be possible to see the actions that were performed on the computer using the system. The main idea of a cyber-physical system is that it is possible to group computers, from the point of view of enterprises. It allows the system administrator team to have a clear distribution of computers on the network. In addition, since the cyber-physical system

provides the use of the web mode, it will allow to check performance even outside the office. The system will include the principle of registration. That is, there is the ability to configure this system, so that each user can individually register themselves and the computers that he uses [1].

The developed cyber-physical system has a variety of settings that allow to configure them on each node as it was required by the user. The system can be configured in such a way that, for example, only a health check of the disk space is performed. Using optimal software solutions, the system has a set of utilities that are tied to each of the services. So, if we consider the service for analyzing the processor load of the system, it has a utility for starting and stopping processes, as well as the ability to start services.

Using one of the popular AES-256 encryption algorithms, all executed requests are encrypted in several levels to ensure high security of the system. Indeed, due to the availability of such a system outside the commercial network, there is a chance when an attacker tries to gain access to it and send malicious files using the file upload utility. Therefore, in addition to encryption, the system has several steps to verify the user to ensure that the user exists and is correct.

Summarizing the general view of the designed cyber-physical system, it can be singled out that the system from the environment of other systems is distinguished by a set of utilities, not only for testing the system, but also for general settings, performing archiving, defragmentation, starting and stopping processes, loading and unloading files. The developed system is distinguished by strong protection against external attacks. Simple configuration capability, ease of use and ease of expanding functionality, and adding new types of health checks as a separate module.

The architecture of the designed multi-functional cyber physical system. For the development of this type of cyber-physical system, the best option in terms of improvement, operability and performance is using micro-service architecture. This architecture will allow to distribute the health check into separate parts – services. In general, a software system is a combination of 4 components, namely:

1) user web pages (data input / output);

2) web server for processing incoming and outgoing requests;

3) microservice scan on each computational node;

4) database to store the necessary information.

These components form a stable architecture will effectively test the performance of several compute nodes [3]. The web page is the graphical interface of this cyber-physical system. The choice of implementing the interface of this cyber-physical system using a web page was aimed at easing the usage and completing independence on the platform. Thus, the work with the software system is equally performed both on smartphones and computer systems. Also, the great advantage of choosing such an interface is the absence of the need to install additional software to work with the system. The web page allows to completely hide the entire software implementation from the user, thereby simplifying its implementation. The requests that the web page performs are in HTTP format in accordance with the standard in Fig. 1. These requests go to the web server, which in turn checks these requests for correctness and belonging to the session with which it works.

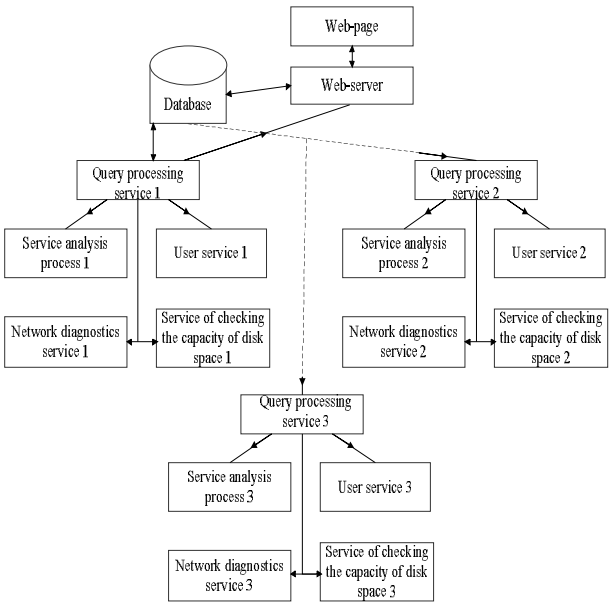

Fig. 1. General structure scheme diagram of the design decision

The web is a distribution point for this cyber-physical system. At the stage when the user logs in, the architecture has only three components. This is a web page, web server and database. Since the web page performs the actual role for data input / output, all of the actions for checking the user's login, getting access lists for the user of nodes and the work on creating requests for micro-service sweeps is done by the web service. Requests are turned into internal structures that microservice understands. These structures are in JSON format. This format is convenient both from the point of view of the user, and from the point of view of turning it into the structure of a programming language. This request passes the decryption stages of encryption before sending. This appointment is for all data to be concealed and in the case of interception it was not possible to get access to the work of the microservice scan.

Encryption is performed using the AES-256 encryption algorithm. This encryption algorithm provides encryption using the key, which, in turn, is stored in the database in accordance with the session. It is allowed for verification at the stage when the request came for microservice. Each part of the request is

encrypted with various keys that are stored in the database until the request is processed. That is, in general, encryption keys are stored for a few milliseconds and disappear immediately. Each request generates new keys, it allows to ensure high reliability of data transfer. On the web server, only websites are generated when data is encrypted / decrypted. In general, the query structure is simple. Session and password are needed in order to know whether this user is still active and whether he has access to this node, as well as whether the user data is correct. The request number is needed so that the request handler can select the component that must execute it and, accordingly, pass data to it. The database in this system has the main role, since it is the system that allows to store user's data, encryption keys, and micro-service placement on the network. Also, the database is used to save the user configuration of the microservice scan [2].

Microservice scan contains a set of services that are virtually independent of each other. This scan has a service that is responsible for processing the requests. It is the service that is registered in the database. It allowed to minimize the direct connection of the web with the health check services. That is, in general, the system has a linear form. The request processing service has internal components that allowed to work with the database, encrypting and decrypting requests, checking the user, checking the correctness of the requests and transferring them to the corresponding services. Since this service is the basis of the sweep, it must always be active. If the service is unavailable, the web server will not be able to send requests and, accordingly, the user will see that this node is unavailable due to the lack of connection to the data processing service. The request processing service reports a configuration sweep file. These files have service settings. However, the most important file is the connection file. This file has the communication settings of the services, that is, the ports used by the services, to listen for incoming requests. This cyber-physical health check system allows to have several identical services, it will allow the system to be loaded if several users want to check the system at the same time. The user was given the opportunity to change the configuration settings of services. That is, it was possible to set the settings in such a way that a health check was performed only for disk space.

The multifunctional of the cyber -physical system is that the system will include not only the monitoring of the computer system, but also the settings of individual nodes. For example, the available set of functions that is developed by the cyber-physical system is available:

- health check and network configuration;
- functional check and system disk drive setup;
- analysis of processor loading;
- archiving;
- grouping computers for one user;
- monitoring and running processes and tasks [5].

All of these functions are put in wrappers called services. These services are dependent only on the query processing service. In general, in addition to the request processing service, 4 additional were provided, namely:

1. Service test of the global network.
2. Service health check processes.
3. Service work with the user.
4. Service check of disk space.

The service for testing the global network includes the ability to analyze incoming outgoing packets, view the network interface settings, obtain a table of IP addresses, and test the global network. The analysis service process includes the ability to obtain a process tree, the amount of resources used by each process, to start processes and stop them. Since, for example, in the Windows operating system it is possible to start services, especially for this, the module has the ability to do this as well. This is a necessary order to fully utilize the functionality provided by the operating system.

The disk health check service involves working with the file system, which is retrieving a folder / file tree, creating, editing and deleting files, copying or moving them, defragmenting a disk or file to analyze the so-called "broken" disk clusters, archiving and unzipping files. The service for working with the user is configured to work with requests for changing the configuration of the service, loading / unloading files, monitoring micro-service sweep. That is, this service allows to get the status of services. The request processing service works like a micro-service sweep firewall. It ensures the protection of the system against incorrect requests, user verification; conducts communication between other services. The main feature of this service is the ability to run health check services. That is, if the service is not started and it is present in the configuration file, the query processing service will start the service and pass it upon the user's request. It can significantly increase the stability and validation of the operation of the health check system.

II. DESCRIPTION OF MULTI-FUNCTIONAL CYBER PHYSICAL INSPECTION SYSTEM OF MODULES PERFORMANCE.

Multi-functional development of cyber-physical health check system is divided into 3 parts: website and web server development, extending services for testing, development of SQL procedures to facilitate the work with the database. To build this software system, the following technologies were chosen:

1) for website development it is used: JavaScript, HTML \ CSS, ReactJS;

2) web server development uses: Java, Spring Boot;

3) for development of micro-service it is used: C \ C ++.

The software system is designed to work with Oracle database.Multifunctional cyber-physical diagram in Fig. 2. The system for monitoring and verifying the performance of remote compute nodes is as follows:

Multifunctional cyber-physical system uses services on Windows OS and so-called “daemon” on Linux OS. All the work of these services is performed in the background. It allows to freely use the compute node and not to interfere with the process of testing performance. Each service only performs that what it has, that is, services do not have overhead implementations that are present in the source code, but never used.

Request processing module. The core of the processing module is the verification of requests, and then the distribution of the request for services. Checking requests is that the service conducts a phased decryption of the request. That is, if the previous part of the request was decrypted, but the validity check found a data error, the request at this stage does not continue to be decrypted and an error is sent. Formation depends on the stage at which the work was suspended. Each decryption stage occurs by searching the key in the database corresponding to the session and the field to be decrypted. Session has a shared key, updated that at 3:00 [4].

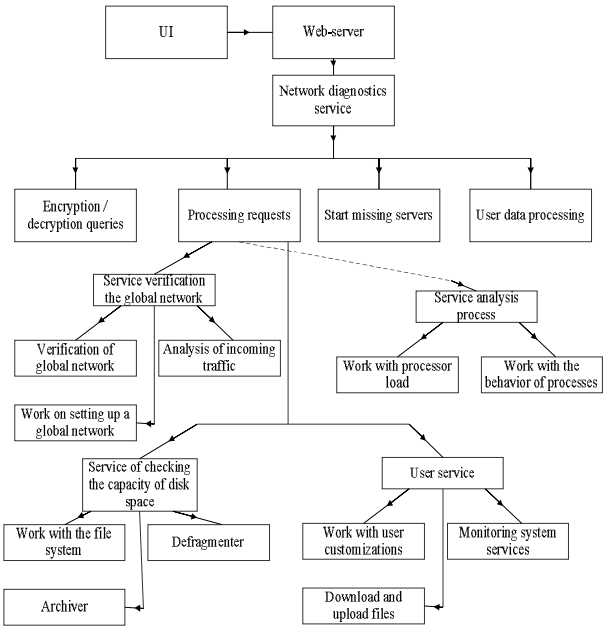

Fig. 2. General functional diagram of multi-functional cyber-physical systems

Encryption keys for session fields are different for each request. All keys that belong to the session, after the completion of the processing of the request, are deleted from the database. At the expense of the session, the user data is received, namely the real password. This password is compared to the one that was decrypted after the session. In case the passwords do not match, this request is not authorized and the service returns an authorization error.

After completing the verification of the user's authorization data, the type of the request is also decrypted, which, in turn, identifies the direction of the request. Each service has its own unique set of requests, which is checked using a mask. Each service has its own unique mask that is superimposed on this request. Because of this, you can determine where the request goes, and also this request by sending only one field. It reduces the traffic load, is very critical in this case. The reason for this is the time and volume of data that comes.

After all, during encryption, the amount of data is block-aligned, which leads to the fact that even a small amount of data is turned into a large encrypted block and, at constant needless transfer of information, this will slow down the system, so the main task is to reduce the amount of service information to one field in Fig. 3.

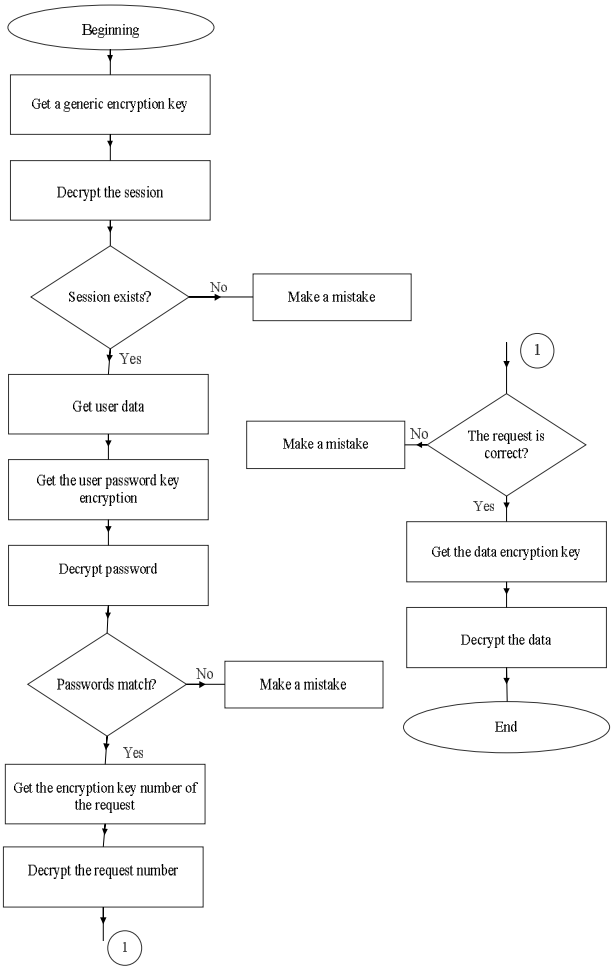

Fig. 3. Algorithm for checking the incoming request.

Upon completion of the preliminary processing of the request, the next step in the operation of the service is the establishment of the service to which the request is going. This is done using the configuration file containing the ports that use the services. If the service is disabled, a message will be generated stating that this

health check is not provided on this node. If the service that the request goes to is stopped, then the request processing service will try to start the health check service and send the request to it. Overall, the general algorithm of the service is quite simple. Its main task is to decrypt the request, check it and send it to the appropriate service [7].

The disk health check service consists of several components. In general, requests that perform this service are divided into requests for working with the file system, requests for working with archiving, requests for working with defragmentation. All requests go through a small request handler built into this service. Its job is solely to provide the appropriate module for processing the request.

Working with the file system consists in executing requests for revising the file tree, editing the file, deleting, moving, copying, and creating. To perform these requests, the operating system tools are used. The archiving work is performed using the ZLib library, which uses the Deflate algorithm for archiving. This is a compression algorithm without data loss. This algorithm is widely used in leading archivers. This algorithm includes a combination of two Huffman and LZ77 algorithms. In general, the archiving module, actually a wrapper for this library, conducts a preliminary check for the correctness of files, their existence, and accessibility.

The job of defragmenting a disk or a drive is performed by recursively defragmenting files. Accordingly, in order to get disk defragmentation, firstly, the service tree files of the disk, or drive, after recursively passing through the files, are defragmented. Since the files are recursively defragmented, the algorithm is at the heart of the cycle through the drive tree and there is no dependence on the file type. This algorithm is generalized and its internal structure changes depending on the type of file system that needs to be defragmented.

1) Analysis Module. This module load is the analysis of running processes. Each process has the resources it uses. The processor load takes account using all aspects, not just the amount of processor time that the process uses. The processes are running for a certain period of time and, accordingly, during the time of use of resources' changes, therefore, to obtain correct data, a certain period of cost analysis is selected. The health check service for this period of time has 1.5 seconds [6]. It allows to reduce the use of processor time by this service. Since the task of verifying the functionality is not only to conduct a complex and redefine the working capacity, but also the absence of a reason for downloading this or that type of working through the health check system.

The Analysis of CPU load always happens. Data on the workload of the system generated in a readable form is converted into kilobytes or bytes, as well as percentages, if we are talking about the percentage of the processor or memory. Upon receipt of the state of the system, the process identifiers are obtained and eaten. That is, the user will be able to change the priority of the process, stop it, look at the location of the file. Also, the user is given the opportunity to start the processes of this system. That is, there is the ability to remotely execute executable files.

2) WAN Health Check Module. For the global network, a health check is a process of viewing the traffic that passes through the system, also a rapid test of the system load, by sending out a large number of packets. The traffic analyzer is installed on the so-called "raw socket", which is inserted into the listening mode. Packages that pass through this socket fall into the so-called array of packets that is present in the service. The size of this array is configured in the service configuration file. Packages are saved in an array only if the background mode is enabled. Listening to packets in the background mode takes place simultaneously on all network interfaces. The user has the ability to start listening using only one network interface.

The user is also available to receive network interface settings, namely:

- the IP address (IPv4 and IPv6 if supported) of the network interface;
- physical address;
- subnet mask;
- time assignment of addresses;
- addresses of DNS and DHCP servers.

In addition to simply view the settings, the user has the ability to change them. After the changes, the network adapter reboots, so that all changes are made. The whole algorithm of work consists of the corresponding functions provided by settings. Each change has consequences and, accordingly, they can be critical. The user must understand these consequences.

3) User module. The service for working with the user has a functionality that is directed more at the management of this cyber-physical system. Management refers to the configuration of system service configurations on a compute node. Thus, with the user of the functional for processing such a file, the process is greatly facilitated, because it guarantees that the configuration of a specific module is divided. The developed cyber-physical system uses two types of configurations:

- General configuration.
- Service Location Configuration.

So, in a common configuration, cyber-physical systems are common to all settings. In particular, level of logging (the information flow to be stored), the maximum size of the log file, the number of threads in the pool "pool", the background health check mode, the maximum number of simultaneous connections.

Service-dependent configuration, including overload of all settings are present in the general configuration, as well as service-specific settings. For example, a disk space health check service allows to limit the size of files to be archived, prohibit defragmentation of certain disk volumes, or prohibit read / write operations [8]. Since the

configuration settings can be duplicated, so that regardless of the values that override the service configuration files, it was possible to use the common settings by all services, the StrictSetupUsage parameter is supported, which tells the service to use the general settings, rather than being overloaded. In addition, the service works with the settings and with the download \ upload files. Files are transferred one by one, that is, several files cannot be downloaded at once, they need to be archived and then transferred over the network in Fig. 4.

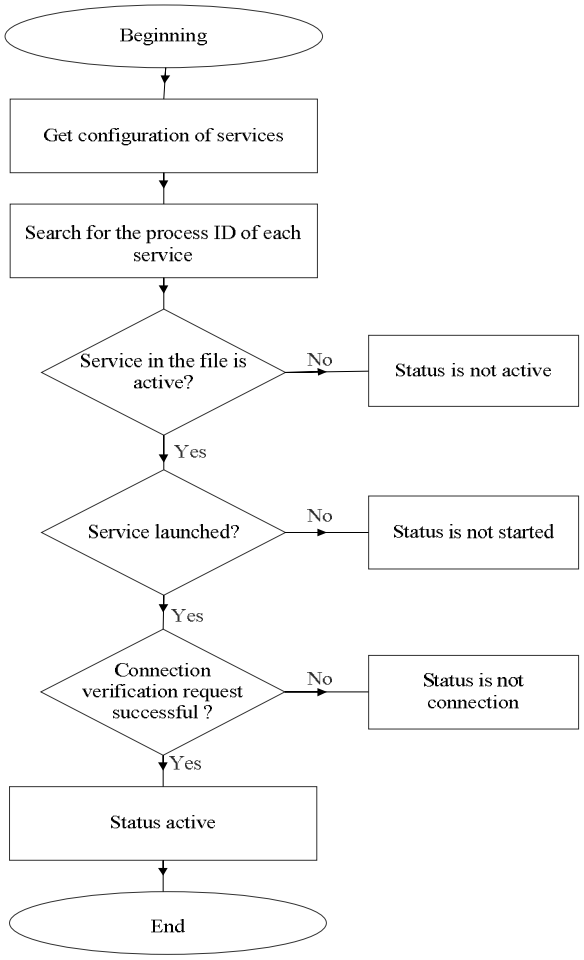

Fig. 4. Algorithm of monitoring of multifunctional cyber-physical systems

In fact, the main task of this service is to monitor a multi-functional system. Monitoring consists in checking the openness of requests, and running of services. Also, given the user's settings, the service is allocated or is in 3 states: active, connection is not accepted, stopped.

Service-dependent configuration, including overload of all settings are present in the general configuration, as well as service-specific settings. For example, a disk space health check service allows to limit the size of files to be archived, prohibit defragmentation of certain disk volumes, or prohibit read / write operations [8]. Since the configuration settings can be duplicated, so that regardless of the values that override the service configuration files, it was possible to use the common settings by all services, the StrictSetupUsage parameter is supported, which tells the service to use the general settings, rather than being overloaded. In addition, the service works with the settings and with the download \ upload files. Files are transferred one by one, that is, several files cannot be downloaded at once, they need to be archived and then transferred over the network in Fig. 4.

In fact, the main task of this service is to monitor a multi-functional system. Monitoring consists in checking the openness of requests, and running of services. Also, given the user's settings, the service is allocated or is in 3 states: active, connection is not accepted, stopped.

III. REALIZATION OF THE GRAPHIC INTERFACE MULTI-FUNCTIONAL CYBER-PHYSICAL TESTING PERFORMANCE OF WEB SERVER.

The graphical interface, as such an algorithm does not work, is used exclusively to send a request and display its result. The interface is implemented using ReactJS, this framework allows to simplify the creation of a website, and also allows to perform web requests to call the functionality of a web server. The framework also allows for a smooth interface to provide a more aesthetic appearance in Fig. 5.

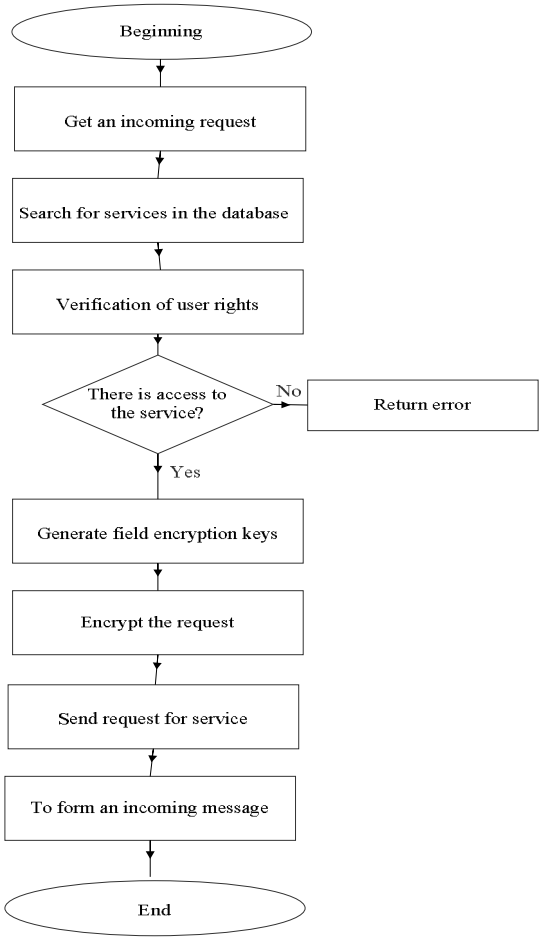

Fig. 5. Algorithm web server processing service requests

The web of this cyber-physical system is the basis and distributor. The web server has a direct connection with the database, from where it can get a list of services, a list of users, insert data about the session. Also, the web checks for user correctness [10]. All actions for grouping services, changing the password, user login, making requests are performed on the web server. On the web server, the main functions are related to links. That is, when a website requests for a specific link, an event handler is called that does not call a function. The data on request can be placed both in the link, in the form of parameters, and in the body of the request. As it was described earlier, the requests that the site sends are standard HTTP requests. Request processing is divided into two parts:

- request processing for web server;
- request processing for services.

Processing the request for the web server is quite simple, since the functions that immediately work out this request are called. As for the processing of service requests, it is much more complicated. Since the input is processed, the service is determined where the request should be sent and the data is encrypted accordingly.

The web server is operating in asynchronous mode, that is, it does not expect a response from the service. However, after processing, the service sends a response to the web server, which calls the so-called "callback", which processes the response and returns it to the site. Callback is a feedback system, from the point of view of a programming language, it is a function that is called when a response is returned from a service. Requests from the site are also asynchronous, they allow to avoid GUI hangs because a response from the web server is expected. Accordingly, for the duration of the request, the user can make several more requests for processing.

The web uses the additional set of session functionality built into the Spring framework. Sessions are created at the stage of user login to the system. All sessions are stored in the database and at the end of their lifetime, are removed from it. The session is necessary in order not to constantly send out in the Ask site, username and password of the user, as well as for greater security. Session lifetime and maximum idle time are configured using system settings. In addition, the lifetime of the common cryptographic key is configured. By default, the multifunctional system uses a value of 2:00 for the life of the session, 20 minutes of possible idle time, that is, a period when the user has not sent a single request, after this time the session is considered to be completed and the user needs to be logged in again. The cryptographic key lifetime is 3:00 [9]. After the time expires, the key is removed from the database and a new key is generated.

To monitor all temporary dependencies, the web server has additional streams using data time from the database. Also, in the case of a change in hourly values when the system is running, automatic reconfiguration of the stream data occurs, which in turn makes it possible not to restart cyber-physical system.

A multifunctional cyber-physical system uses a database to store basic data. In fact, the main part of the protection of this cyber-physical system is the database. It stores data about the encryption keys of the request fields. It stores data about the registered users, groups of users, addresses of request processing services of each computational node. Also, since the database has the ability to save functions and procedures, it allows to reduce almost all the work of obtaining the recorded data, updating services to a minimum, since all the functionality will be described by procedures. The database has four main tables and several secondary tables. The main tables are: user table; table of services; session table; table for storing cryptographic keys. The secondary tables include:

- table of system settings;
- query history table;
- table of current settings;
- table of service groups;
- table of user groups;
- table of service belonging to the user.

The interrelation of the four main tables builds the basis of the developed cyber-physical system. These tables are associated with a unique user identifier, provided to it at registration, and which will appear in most tables. Also, each service has its own unique identifier, to add its own settings to the system and the session is always unique.

Only the table in which cryptographic data is stored for each field does not have a key, that is, a unique identifier by which only one record can be obtained. Such a solution could be circumvented by expanding the number of fields, but this would increase the size of the table and slow down the search for queries. Therefore, several records are inserted into the label [11]. Each request has its own unique identifier, which is not repeated, thereby keeping 3 records for one request. In the table, a cryptographic key was used for: user password, request number, data.

The request number is not its unique identifier, it is the number that determines which service the request should be written to. A unique request identifier is assigned to it at the formation stage and arrives at the service just before the request. It is formed using the created procedure for incrementing this identifier in the database. Plates usually store less important information. All labels, depending on the destination, are associated with unique identifiers of services, user and sessions.

IV. TESTING THE SERVICE FOR VERIFYING THE PERFORMANCE OF THE GLOBAL NETWORK OF A CYBER-PHYSICAL AND MULTIFUNCTIONAL SYSTEM

Since the service writes a log file to the results, actual data on the execution of all scenarios will be used from it.

1) The result of testing the scenario of obtaining the current network settings:

- The obtained data is in Fig. 6.

```
----------------Starting to retrieve network parameters info----------------
      -> Host Name: Zero-PC
      -> Domain Name:
      -> DNS Servers:
            -> 192.168.0.1

      -> Node Type:
            -> Hybrid
      -> NetBIOS Scope ID:
      -> IP Routing Enabled: No
      -> WINS Proxy Enabled: No
      -> NetBIOS Resolution Uses DNS: No
----------------Finished retrieving network parameters info-----------------
```

Fig. 6. One expected network information

- Expected data is in Fig. 7.

```
DNS Servers . . . . . . . . . . . : 192.168.0.1
                                    0.0.0.0
NetBIOS over Tcpip. . . . . . . . : Enabled
```

Fig. 7. Expected network information

2) The result of testing the scenario of obtaining statistical data TCP and IP:

- The obtained data is in Fig. 8.

```
--------Starting to retrieve tcp statistic info---
-> Number of active opens:  10774
-> Number of passive opens: 557
-> Number of segments received: 22532798
-> Number of segments transmitted: 15104406
-> Number of total connections: 62
```

Fig. 8. Obtained statistics on the TCP protocol

- - Expected data is in Fig. 9.

```
TCP Statistics for IPv4

  Active Opens                        = 10999
  Passive Opens                       = 557
  Failed Connection Attempts          = 691
  Reset Connections                   = 1386
  Current Connections                 = 29
  Segments Received                   = 22557986
  Segments Sent                       = 15125888
  Segments Retransmitted              = 29065
```

Fig. 9. Expected statistics on the TCP protocol

The data is slightly different due to the fact that the expected results were read, with some delay after receiving data from the service.

3) Test result of adapter data script

- The obtained data is in Fig. 10.

```
--------------Starting to retrieve adapters info-----------
  -> Adapter Name: {A62B898C-0ED5-440F-9458-46FFBECC405E}
  -> Adapter Desc: Realtek PCIe GBE Family Controller
  -> Adapter Addr: 14-DA-E9-EC-AC-34
  -> IP Address: 192.168.0.101
  -> IP Mask: 255.255.255.0
  -> Gateway: 192.168.0.1
  -> Lease Obtained: Tue, 06.06.2017 06:42:36
  -> Lease Expires: Tue, 06.06.2017 08:42:36
  -> DHCP Enabled: Yes
          -> DHCP Server: 192.168.0.1
--------------Finished retrieving adapters info------------
```

Fig. 10. Five received adapter data

- - Expected data is in Fig.11.

```
Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Realtek PCIe GBE Family Controller
   Physical Address. . . . . . . . . : 14-DA-E9-EC-AC-34
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::34d6:eb73:e393:8856%3(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.0.101(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Sunday, June 4, 2017 2:41:52 PM
   Lease Expires . . . . . . . . . . : Tuesday, June 6, 2017 8:42:37 AM
   Default Gateway . . . . . . . . . : 192.168.0.1
   DHCP Server . . . . . . . . . . . : 192.168.0.1
   DHCPv6 IAID . . . . . . . . . . . : 51698409
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-1E-E4-FE-72-14-DA-E9-EC-AC-34

   DNS Servers . . . . . . . . . . . : 192.168.0.1
                                       0.0.0.0
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

Fig. 11. Expected data about adapters

The service working with the user is the actual warden for the system. After all, thanks to this service, you can get information about services, and carry out their settings. Before the service works with the user, there are following list of tasks:

- getting the status of services;
- getting service configuration;
- change service settings.

All actions are performed using a graphical interface. The user experiences service acts as a monitor for services and, accordingly, will not appear in the list of active services, because it is with its use that the current information is obtained. So, the service should return data about such services: service test of the global network; service analysis processor load; service check of disk drive. These services have MDS prefix in the name of the executable file and, accordingly, the name of the process. It allows to simplify testing and search data processes in the process manager. Configuration files are stored in a separate installation folder. That is, each service has a separate folder.

Consequently, when you click on the username in the upper right corner of the site, it automatically redirects to the user page. When the page is loaded, it immediately makes a request for the status of the services, so that the user does not spend time on pressing the buttons to get this data. In the current installation, these services have the following names:

1. Global Network Health Check Service – MDSNetworkDiagSrv.exe

2. Service analysis of the processor load of the computer "UTERA- MDSProcDiagSrv.exe.

3. Service check of disk drive – MDSDiskDiagSrv.exe.

These services have been recently launched, so they do not have a lot of resources to use in Fig. 12.

According to the data from the computer, the services are currently running and are available for communication and work. The next stage of testing is to view and modify service configurations. To do this, select a service whose settings will be changed.

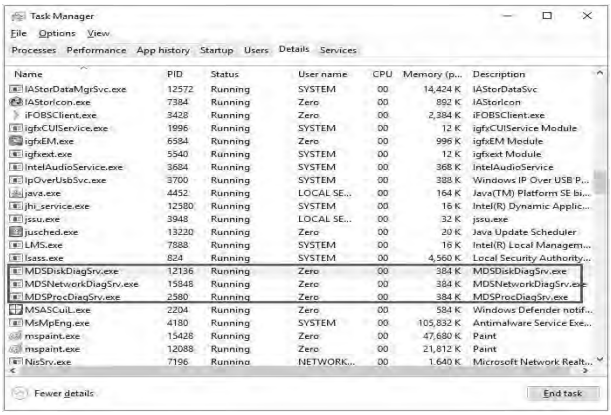

Fig. 12. Eight data task manager environment

Configuration files are fully analyzed and divided into three categories, namely the parameter, their value and a comment in them. To confirm the correctness of the data, you need to revise the configuration file, which is located in the folder with the service. Thus, the data that is in the configuration file in the environment coincides with the data that was returned to the site. The final task of this service is to change configurations. After making changes on the site and saving them, these data must be entered into the configuration file.

So, the service coped with all the tasks. All data was correct, and changes were acquired not only on the site, but also in the user interface and in the service configuration file.

V. CONCLUSIONS

In this article the proposed multifunctional cyber - physical system used efficient query processing algorithms. It had methods of protection, and through the use of the architecture, microservices had greater flexibility to change, and resistance to environmental problems. It was argued that this type of cyber physical systems had many configuration aspects that allowed to change settings in various ways. The article was studied and analyzed; the availability of the possibility of disabling certain services sometimes played an important role, since not all systems might require a complete functional check. And also, the existing nodes were inserted, which had a greater dependence on the workload of the processes and, accordingly, a decrease in the number of fund performance checked up to 1 would allow to carry out the necessary performance checks and not to violate the stability of the system by running additional services that were not used. So, the stable operation of a multi-functional cyber physical system depended on the database. Placing it in regions with a little delay in requests would increase the overall performance of the system.

Moreover, a multifunctional cyber-physical system that checked the performance of remote computers in the WAN, although easy to use, but required pre-setting of the environment with which it would work. Thus, in a multi-functional cyber-physical system, it depended on the libraries for working with the hardware of the computer system, and the library for working with threads. In this regard, before establishing the services of a multi-functional cyber-physical system, you must firstly install a distribution kit with libraries, if the environment was based on the use of the operating system Windows, and install a library for working with streams, for environments based on Linux OS. Also, a multifunctional cyber-physical system required these operating systems to be supported with Oracle database operations. To do this, there must be a client with drivers for working with databases.

REFERENCES

[1] Geyer J. O. Wireless Networks. First step: Trans. from English. – M.: Publishing House “Williams”, 2011, page 192 (In English).

[2] Esposito D.E. Development of modern web applications: analysis of subject areas and technologies. – Dialectics-Williams, 2017, page 464 (In English).

[3] Olifer V. G. , Olifer N. A. “Computer networks. Principles, technologies, protocols. Edition 4th.”, Uch. manual for universities. Petersburg, 2010, page 943 (In English).

[4] Ratcha E. J. IBM AT A Beginner's Guide. – Radio and communications, 2000, page 154 (In English).

[5] Banks A. O., Porcello E. L. Learning React: Functional Web Development with React and Redux 1st Edition. – O'Reilly Media 1 edition, 2017, page 350 (In English).

[6] Guild. H. H. Fully iterative fast array for binary multiplication and addition. Electronics Letters, Volume 5, Issue 12, 12 June 1969, page 263 (In English).

[7] Pasternak I. I. Modular client / server interaction interface [Structural interface in network]. *Visnyk Natsional'noho universytetu “L'vivs'ka politekhnika” “Komp"yuterni systemy ta merezhi”*. Lviv, Ukraine, 2012, vol. 745, pp. 160–163 (In Ukrainian).

[8] Pasternak I. I., Morozov Yu. V. Modular network interface for distributed information navigation systems [Evaluation of structural complexity of multisection multiplier for Galois field elements]. *Visnyk Natsional'noho universytetu “L'vivs'ka politekhnika” “Computational problems of electrical engineering”*. Lviv, Ukraine, 2014, vol. 3, pp. 47–56 (In English).

[9] Pasternak I. I. Principles of designing a social network with minimal load on servers [Definition of client / server interac]. *Visnyk Natsional'noho universytetu “L'vivs'ka politekhnika” “Komp"yuterni systemy ta merezhi”*. Lviv, Ukraine, 2016, vol. 857, pp. 74–82 (In Ukrainian).

[10] Pasternak I. I. Diagnostics and debugging of nodes of the corporate network of cyber-physics systems [Definition of the extended minimal load on servers].*Visnyk Natsionalnoho universytetu “Lvivska politekhnika” “Informatsiini systemy ta merezhi”*, vol. 872. Lviv, Ukraine, 2017. pp. 3–9 (In Ukrainian).

[11] Pasternak I. I. Means of checking nodes of the communication network of the cyber-physics system [Definition of the information navigation systems and cyber-physics system]. *Visnyk Natsionalnoho universytetu “Lvivska politekhnika” “Informatsiini systemy ta merezhi”*, vol. 881. Lviv, Ukraine, 2017. pp. 107–119 (In Ukrainian).

Iryna I. Pasternak is an assistant professor of the Department of Computer Engineering at Lviv Polytechnic National University, Ukraine. Iryna Pasternak was born in 1988 in Lviv, Ukraine. She received the B.S. degree in computer engineering at Lviv, Polytechnic National University in 2010 and M.S. degree in system programming at Lviv Polytechnic National University in 2011. In 2015 she obtained her Ph.D. at Lviv Polytechnic National University. Her research interests include network interface, diagnostics of computer networks, client-server interaction, modular programming navigation and cyber-physical systems.