UDC 621.04 doi: 10.20535/2411-2976.12024.39-49

METHOD OF THE SERVER HORIZONTAL LOAD BALANCING FOR REDUCING ENERGY CONSUMPTION

[1]Mariia A. Skulysh, [2]Umakoglu Inci

[1]Educational and Research Institute of Telecommunication Systems
Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, Ukraine
[2]Electrical and Electronics Engineering, Kutahya Dumlupinar University, Turkey

Background. Server horizontal load balancing is a crucial aspect of modern computing systems, particularly in cloud computing environments. The efficient management of incoming flows of applications is essential to ensure optimal resource utilization and minimize energy consumption. This study focuses on developing a method for managing the incoming flow of applications to reduce energy consumption in server horizontal load balancing.

Objective. The primary objective is to develop a method for managing the incoming flow of applications to reduce energy consumption in server horizontal load balancing. This involves identifying the maximum permissible number of applications that can simultaneously enter the system for service, ensuring that the volume of resources used is close to the total maximum possible amount of resources. The method aims to minimize the variance of the elements of the sequence of maximum allowable numbers of applications and the dispersion of the elements of the sequences of volumes of resources used.

Methods. The method involves several key steps:

Input Load Smoothing Scheme: A static control method is proposed to smooth the incoming load. This involves developing a scheme for smoothing the incoming load, which is a set of values of the maximum allowable number of requests (sequence {ki}) arriving at the system input for a small time interval Δti. The sequence is selected to ensure that the volume of resources used is close to the total maximum possible amount of resources.

Genetic Algorithm: The selection of the sequence {ki} is carried out using a genetic algorithm. The algorithm involves crossover, mutation, and selection operations to minimize the variance of the elements of the sequence and the dispersion of the elements of the sequences of volumes of resources used.

Resource Allocation: The method involves allocating resources for the maintenance of a given type of service. The parameters of the server, which are characterized as the resources of the system serving the applications, are usually calculated for the average values of the parameters of the input stream.

Delay Introduction: To manage the application processing process and prevent resource shortages, a delay is introduced for a part of the applications that coincide with a surge in load. The delay time is determined so that delayed applications do not enter the system until the previous burst of load is successfully serviced in the resource-consuming functional block.

Results. The results of the study include the development of a method for managing the incoming flow of applications to reduce energy consumption in server horizontal load balancing. The method involves the use of a genetic algorithm to select the sequence {ki} that minimizes the variance of the elements of the sequence and the dispersion of the elements of the sequences of volumes of resources used.

Conclusions. The study concludes that the proposed method for managing the incoming flow of applications can effectively reduce energy consumption in server horizontal load balancing. The method involves the use of a genetic algorithm to select the sequence {ki} that ensures efficient use of system resources and minimizes the variance of the elements of the sequence and the dispersion of the elements of the sequences of volumes of resources used. The method can be applied in various scenarios where efficient use of system resources is crucial, such as in cloud computing environments.

Keywords: *Telecommunication services; cloud environment; resource management; load balancing; dynamic method; GPSS; application processing; energy efficiency.*

I. INTRODUCTION

In the context of hybrid telecommunication flows within cloud environments, efficient allocation of technical resources has become paramount due to the escalating demand for diverse telecommunication services.[1] This study presents a novel method for horizontal server load balancing aimed at reducing energy consumption while maintaining optimal service quality. The primary objective is to address the scientific problem of distributing limited computing resources across various virtual spaces, each serving different types of telecommunication services.

Through the analysis of service maintenance processes, the study highlights the technical feasibility of sustaining services in virtual environments with predetermined computing resources. The research employs a simulation approach using the General

Purpose Simulation System (GPSS) package to model the telecommunication service maintenance process. The model considers an online tariff system of a telecommunications operator, where cloud computing space is rented to support the online charging process. A key focus of the study is the management of heterogeneous and uneven application flows to ensure efficient resource utilization.[2] The simulation framework involves the division of service operations into logically distinct functional blocks, each requiring specific server resources such as RAM, processor time, and permanent memory. The research proposes a dynamic load balancing method that incorporates a delay mechanism to manage peak loads and prevent resource overconsumption.

The findings from the simulation indicate a significant reduction in application loss and an enhancement in economic efficiency of service maintenance. The proposed method ensures the optimal distribution of server resources, thus minimizing delays and improving overall service quality. The study also underscores the necessity of a two-level control system for managing the incoming application flow and the corresponding server resources.

II. Methods of allocation of technical resources for hybrid telecommunication flow

The analysis of service maintenance processes in the cloud environment showed that it was technically possible to maintain services in the virtual space with a given amount of computing resources. At the same time, the total amount of resources that is leased by the provider of telecommunication services is determined and limited by the lease agreement. The variety of telecommunication services and different requirements for the process of service maintenance determines the need to solve the scientific problem of distributing a limited number of computing resources between virtual spaces serving different types of telecommunication services.

To simulate the process of telecommunication service maintenance, the online tariff system of the telecommunications operator was considered. To meet the needs of the charging system, the telecommunications operator rents cloud computing space to support the online charging process. At the same time, software is deployed in the cloud environment, which serves applications for pricing of various services.[3] Each service provided by a mobile operator is characterized by a certain type of application received by the server. The paper will consider the request of the subscriber to perform a certain service as an application. At the same time, several important problems arise in the application service process. First, a large number of heterogeneous requests that require immediate processing were received at the server at the same time. Secondly, the incoming flow of applications is uneven. The third problem is related to the heterogeneity of the use of server resources when providing service for each application. Among the main resources of the server, you can highlight such as RAM, processor time, volume of permanent memory on disks.

Service of the application on the server consists of the execution of some sequence of operations that can be divided into logically completed stages. In the following, these stages will be called functional blocks. The above-mentioned resources were used to service subscription applications in functional blocks. Successful completion of all functional blocks in the specified sequence ensured successful service of the application on the server. The service time on the server was limited, so if the request was in the system longer than the specified time, it would be removed from the service. In the same time for controlling energy efficiency according to the previous researches [4] it was necessary to check amount of resources used. Thus, in order to reduce the loss of applications to a minimum and maximize the economic efficiency of service, it is necessary to ensure the optimal distribution of server resources between different types of applications received by the operator.

Resources such as RAM, processor time, network resource (channel occupancy by signal traffic), volume of permanent memory on disks are used to service subscriber applications in functional blocks. Servicing each application in a given functional block requires a given number of resources. It is known how long each of the resources was used during maintenance in a given functional block. If the resource is busy, then the application is waiting for the resource to be released. Thus, there are delays in service, which lead to the loss of successfully served applications. That is why the system for monitoring the use of resources by different types of services was built taking into account the features of the application service stages on the server of the mobile application operator. It will be useful in the organization of the traffic management system.

III. The task of defining distribution of applications' number that are currently being serviced in the system

Formulation of the problem.

Applications for service are sent to the system *according to the law*. The process of servicing one request includes staying (serving) the request in one of *n* functional blocks, and G types of resources are used for service. Let there be service requests from *m* types of services. The statistics of the time of the application of the *i*-th type of service ($i = \overline{1,m}$) in the *j*-th functional block ($j = \overline{1,n}$) are known. **It is necessary to find** the distribution of applications between functional blocks and by types of services that are served in the system at the current moment in time.

It is necessary to define the matrix T={ t_{ij}}, each element of which corresponds to the mathematical expectation of the time of the application of the *i*-th type ($i = \overline{1,m}$) in the *j*-th functional block ($j = \overline{1,n}$). It is calculated by using the method of moments.

It is also needed to define a matrix $V^{Sg} = \{v_{ij}^{Sg}\}$, each element v_{ij}^{Sg} corresponding to the volume of the *g*-th resource type, which is used when serving the application of the *i*-th type ($i = \overline{1,m}$) in the jth functional block ($j = \overline{1,n}$). Then the total amount of the resource of the *g*-th type ($g = \overline{1,G}$) engaged in the *i*-th service at the current time is determined by the formula

$$v_i^{Sg} = \sum_{j=1}^{n} k_{ij} v_{ij}^{Sg}$$

where k_{ij}- the number of applications of the *i*-th type served at the current time in the *j*-th functional block;

v_i^{Sg}- the volume of the *g*-th type resource, which deals with requests of the *i*-th type of service at the current time.

Example 1.

Let m=1, n=4. That is, the system serves one type of applications in four functional blocks, only RAM is required for service (G=1, denoted by s1).

Let the mathematical expectation of the application's stay time in each functional block have already been calculated using the Method of Moments: t_{11}=1, t_{12} =3, $t_{13} = 4$, $t_{14} = 1$.

$$T = \begin{pmatrix} 1 \\ 3 \\ 4 \\ 1 \end{pmatrix}$$

That is, on average, the application spends nine units of time in the system, denoting it by T_Σ(T_Σ=9). Let K=100 requests arrive at the system within nine units of time and let the incoming flow be described by a uniformed distribution. Then the number of requests that got into the corresponding functional block and is served at the current moment of time will be as follows:

$$k_{1j} = \frac{t_{1j}}{T_\Sigma} K,$$

where $k_{1j}$is the number of applications of the first type in the *j*-th functional block.

It is known that to perform operations with the 1st type in the 1st functional unit, a volume of RAM is required one unit ($v_{11}^{s1} = 1$), to perform operations with the 1st information flow in the functional unit 2, a volume of RAM requires memory zero units ($v_{12}^{s1} = 0$), respectively: $v_{13}^{s1} = 4$, $v_{14}^{s1} = 2$. That is, the matrix V^{S1} with the form:

$$V^{S1} = \begin{pmatrix} 1 \\ 0 \\ 4 \\ 2 \end{pmatrix}$$

Fig. shows the use of resource *s1* during the service time of one application of the 1st type in all four functional blocks.

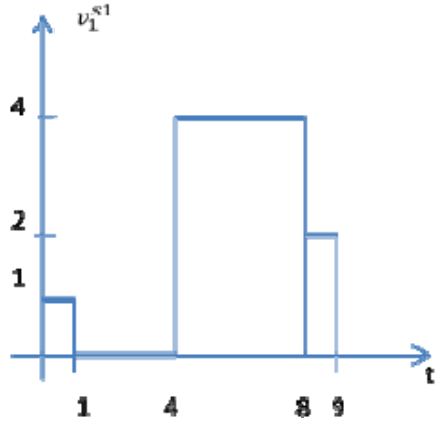

Fig. 1 Example of resource using over time

When a hundred requests were uniformly received to the system within nine times units, then it can be talked about the following distribution of the resource *s1* between the requests that are in service. The volume of

the resource *s1*, which is occupied by a hundred applications, is determined by the formula: $w_1^{s1} = \sum_{j=1}^{4} k_{1j} w_{1j}^{s1} = \sum_{j=1}^{4} \frac{r_{1j}}{r_\Sigma} K w_{1j}^{s1}$

$$w_1^{s1} = \frac{1}{9} * 100 * 1 + \frac{3}{9} * 100 * 0 + \frac{4}{9} * 100 * 4 + \frac{1}{9} * 100 * 2 = 211{,}11$$

. □

The assumption of uniformed input flow is essential to the model. In real systems, the incoming flow of requests from services can be described using the Poisson distribution of random variables. Thus, the task of analysing server resource load should be solved in two approaches:

a. To conduct an analysis of the use of system resources, provided that it is known how many requests and what type of service were received by the system in each time period, which allows one to build a monitoring system that will identify current problems by using the system resources.
b. Analyse the use of system resources over time, which requires analysing large-volume samples to identify "bottlenecks" in the system. This task should be considered in further works.

IV. PRE-REQUISITES FOR CREATION SYSTEMS MANAGEMENT INPUT FLOW TO THE NODE CLOUDY ENVIRONMENT

The charging process is multi-stage, while the operations that are sequentially performed by processing business logic with the involvement of various sub-systems are diverse and, accordingly, require different amounts of RAM, processor time, and disk space. When solving the task of managing the incoming flow, it is necessary to pay attention to the time of resource usage. It is also necessary to take into account both the total number of resources serving the server as a whole, and the separation of resources. In one hand it ensures efficient service of each stage of processing, and on the other one there is a limitation, since the sub-system uses only the resources allocated to it and does not have access to other ones.[5] It is also necessary to take into account the distribution of the average time of operations.

The second feature is that each type of service, despite the standardisation of operations performed in the pricing process, requires a different amount of resources to perform calculations.

As far as the required service procedure is in concern, all services can be divided into three groups:

- Tariff session with reservation (SCUR - Session Charging with Unit Reservation) the RAM is occupied for the entire duration of the session (can last up to a day - for example, GPRS)
- Instant pricing of the event (IEC Immediate Event Charging) does not save the status of its execution in memory - the evaluation and debiting of funds is performed at the same time (SMS).
- Event pricing with reservation (ECUR - Event Charging with Unit Reservation) – the RAM is occupied for the reservation period (for example, the time of content delivery to the subscriber: video, music).

Thus, SCUR and ECUR services are performed in several stages. The state of the application or the state of the call is stored in the MDP sub-system (Memory Database Provider). MDP is a module for saving the current state, which is a software-hardware complex that provides fast access to RAM (writing, reading, searching).

When servicing SCUR and ECUR services, the first and second stages, including extracting information about the subscriber and his location, are performed once, after which all information about the subscriber and the status of the call request are stored in the MDP sub-system.

The resources required by the system for servicing seven stages depend not only on the group of services, but also on its type. The difference in the speed and resource consumption of operations occurs when calculating the cost of the service and when extracting information about the status of the request-call from the MDP system. These conditions must be taken into account when calculating the plan for managing the incoming flow to the tariff server.

The third feature is that a large number of applications for tariffication of various types of resources are received at the same time.

Nowadays, mobile operators provide services to billions of subscribers. Under the condition of centralised service, the tariff system simultaneously serves up to one million subscribers who order or continue to use the services. A chain of operations described above is initiated for each subscriber request. During rush hours, the number of subscription requests increases several times.

The forth feature is the heterogeneity of the incoming flow of applications. It is customary to consider subscriber service systems as systems with a Poisson input flow of applications. The main feature is

the significant dispersion of the number of applications received. For the Poisson distribution, the variance is equal to the mathematical expectation. That is, bursts of load are possible for a short period of time, which is shorter than the service time of the application on the tariff server. Such surges lead to temporary server overload even in no rush hours conditions.

From the point of view of implementing the logic of the service process, the operation of the server sub-systems of the mobile operator can be represented as a multi-level system of mass service, where the flow of applications is managed on two levels.[6]

The first logical level of application software components.

Here, the applications received in the system are divided according to the type of service they represent. Queue maintenance is carried out according to the service scheme developed for the corresponding type of service. The management process includes the formation of queues by service type, the application of WRAD group methods, as well as other management schemes that take into account the specifics of service maintenance.[7] Thus, the mass service system of the first level represents applications of various types of services that come to the system for service, and can be called applications of the first level. Service devices in such a system are chains of functional blocks, where applications are serviced sequentially, each type of service is maintained in a separate chain.

The second level is the level of technical processing. The application service scheme by type of service involves the sequential execution of operations that require a specified amount of hardware resources. Each operation can be presented as a service application. One can talk about second-level applications, where the service devices are hardware resources. Here, second-level requests are organised into queues to the corresponding resources. Resource usage policies are determined by the resource management methods of the computing system.[8] Resource allocation architectures, second-level request service organisation, significantly affect service speed. However, this architecture of the second-level application processing system is permanent. Its operation can be judged by the statistical data of delays in the processing of first-level applications.

The incoming flow of second-level applications is uniquely determined by the number of first-level applications served in the system. Therefore, the system of managing the incoming flow of applications of the first level, which is built taking into account the statistics of the resource load of the second level system, will allow reducing the loss of applications due to delays associated with a lack of resources.[9]

The question arises as to how to organise the system of managing the incoming flow of applications so that the flow of second-level applications is as uniformed as possible.

Service of requests of the first level in functional blocks generates a flow of requests of the second type, for the execution of which a given amount of server resources is used. Therefore, if a large number of requests of the first type are processed simultaneously in some functional block, while requests of the second type generated by the corresponding functional block require a significant amount of resources for their execution, then the problem of a lack of server resources may arise. It will lead to a delay in serving requests of the first type, and as a result of exceeding the allowable service time, loss of applications, decrease in the quality of service to subscribers.

The idea is not to allow two bursts of load during the time of serving requests in resource-consuming functional blocks. The suggested method involves tracking load peaks and introducing a delay for some parts of the requests of the second peak, which avoids overloading the server's resources.

In order to form a long-term load smoothing program, a static control method was proposed the incoming flow of applications for tariffication, involving the development of *a scheme for smoothing the incoming load.*

The input load smoothing scheme is a set of values of the maximum allowable number of requests (sequence $\{k_i\}$) arriving at the system input for a small time interval Δt_i in a given sequence. The number of elements of the sequence n is selected in such a way that the equation is fulfilled

$$t = \sum_{i=1}^{n} \Delta t_i,$$

where t is the average time of a first-level application staying in the system.

It is necessary to choose such a sequence $\{k_i\}$ where two conditions are fulfilled:

a. Applications that are simultaneously served in the system must use the volume of resources V close to the total maximum possible amount of resources V_{max}. Dispersion of sequences of such volumes should be minimal.

b. The variance of elements of the sequence $\{k_i\}$ should be minimal.

c. The length of stay of the first-level application in the functional blocks (FB) is a random value that depends on the processing speed of the second-level

applications generated in this FB. Based on the average statistical values obtained by the monitoring system one can say that the time the application is in the functional block is known (t_j , where j is the FB number). $t = \sum_{j=1}^{m} t_j$, where m is the number of functional blocks in the system.

The amount of resource (v_j, where j is the number of FB) is known, which is required to service the second-level requests generated by the given FB.

The way of calculating the volume of the resource V, which is used at the current moment of time, as the basic principle of reversing time reference system is used. If the time of the end of service of the application is zero, i.e. $t^0 = t$, then it can denote the time periods when applications pass between functional blocks: $t^1 = t^0 - t_1$, ..., $t^j = t^{j-1} - t_j$, ..., $t^m = t^{m-1} - t_m = 0$. All requests received during the $[t^1, t^0]$ time interval t^0 are served in the first functional block. Requests received by the system during the interval $[t^j, t^{j-1}]$ at time t^0, are served in the j-th functional block.

Thus, the volume of the resource V^0 used at the moment of time t^0 is the sum of the volumes of the resource v_j^0 occupied by applications that are in the j-th FB ($j = \overline{1,m}$) at the moment of time t^0.

$$V^0 = \sum_{j=1}^{m} v_j^0$$

The value v_j^0 depends on the number of applications received by the system during time $[t^j, t^{j-1}]$, and is defined as the product of the number of applications by the amount of resources required to service one application in the corresponding functional block. Since the input load smoothing scheme is used, the maximum allowable number of applications falling on this time interval is known. v_j^0 is the product of the number of applications that t^0 are in the j-th FB at the moment of time by the amount of resources required to service second-level applications generated by the j-th FB.

In order to ensure effective smoothing, it is necessary that condition 1 is fulfilled not only for the volume V^0, but also for all V^i, $(i = \overline{1,n})$.

The objective function includes three components:

- The variance of the elements of the sequence $\{k_{and}\}$ should be minimal.
- The dispersion of the elements of the sequences $\{V^{and}\}$ is minimal
- The average value of the elements of the sequence $\{V_j\}$ leads to the maximum possible amount of the resource V_{max} of a given type, which is allocated to service applications of the selected type of service

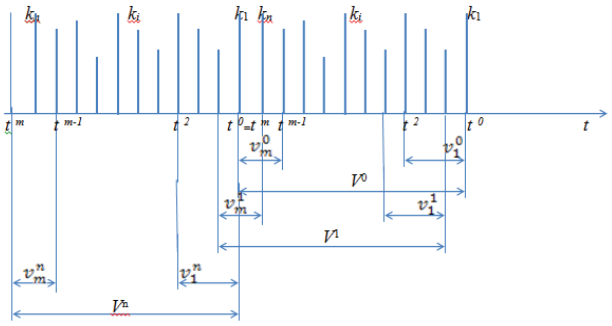

Fig. 2 Scheme of smoothing the input load, taking into account the volume of the resource used

According to the solution method, the selection of the sequence $\{k_i\}$ is carried out with the help of a genetic algorithm, in such a way that conditions 1 and 2 are met:

a. The sequence elements $\{k_{and}\}$ are the genome.

b. Crossover: changing the values of elements of the sequence $\{k_{and}\}$.

c. Algorithm completion conditions:

- by time
- by the number of considered generations
- population decline

As a result, the following sequence can be achieved.

V. SETTING THE MANAGEMENT METHOD PROBLEM OF INCOMING STREAM FOR PROCESSING

The method of managing the incoming flow of applications for pricing, the feature of which is the control of the number of applications that are at the service stage, in case of exceeding their permissible number, applications are delayed at the entrance, which allows avoiding resource overload and to prevent inefficient loading of the resource for servicing applications that will be processed more than for the time allocated in the system.

The input data in the task of managing the flow of requests that arrive for service at the server of the mobile operator are:

- Information about the volume of resources that is necessary for the implementation of operations provided by the functional block for servicing the application of the given type of service.

- Information on the duration of resource use when serving a request of a given type of service in each functional block.
- Statistical information about the duration of service of the application of the given type of service in each functional block.
- The volume of resources is allocated for the maintenance of a given type of service.

The parameters of the server, which are characterised as the resources of the system serving the applications, are usually calculated for the average values of the parameters of the input stream. However, the system has peak values of the number of applications received at the same time.

By a surge in the load of the incoming stream, it means the simultaneous arrival of such a number of applications that is bigger than the above calculated permissible value.

To manage the application processing process in order to prevent resource shortages in the management system, the following strategy is suggested:

✓ two or more input stream load bursts were not served simultaneously in functional blocks, the processing of which requires a significant amount of resources:

✓ for this, a delay is introduced for a part of the applications, the receipt of which coincided with a surge in load. The delay time should be determined so that delayed applications do not enter the system until the previous burst of load is successfully serviced in the resource-consuming functional block.[10][11]

Applications for service are sent to the system according to the law. The process of servicing one request includes staying (serving) the request in one of *n* functional blocks, *G* types of resources are used for service. If there are requests for service from *m* types of services, the statistics of the time of the application of the *i*-th type of service ($i = \overline{1,m}$) in the *j*-th functional block ($j = \overline{1,n}$) are known.

The known mathematical expectation (t_{ij}) of the time of application of the *i*-th type of service ($i = \overline{1,m}$) in the *j*-th functional block ($j = \overline{1,n}$), these data are summarized in the matrix T=$\{t_{ij}\}$. It is known that during the service of the application of the *i*-th type of service ($i = \overline{1,m}$) in the *j*-th functional block ($j = \overline{1,n}$), the resource of the *g*-th type is engaged for a time τ_{ij}^{Sg}($\tau_{ij}^{Sg} \le t_{ij}$). Information on the duration of service is summarized in the matrix $T^{Sg} = \left\{\tau_{ij}^{Sg}\right\}_{i=1,m,j=1,n}$.

There are G pieces of such matrices, each matrix corresponds to one of the resources under consideration.

A matrix is known $V^{Sg} = \{v_{ij}^{Sg}\}$, each element v_{ij}^{Sg} of which corresponds to the volume of the resource of the *g*-th type, which is used in the service of the application of the *i*-th type ($i = \overline{1,m}$) in the *j*-th functional block ($j = \overline{1,n}$).

Within the framework of this study, the detailing of business processes that take place in the functional block is not considered. It is not specified at which stage of service of the application in the middle of the functional block, which resource is used. Therefore, an assumption is made: all applications that are currently served in the functional block use resources evenly, that is, the volume of the *g*-th resource used by the *i*-th type of service in the *j*-th FB decreases in proportion to the ratio of the time of use of the resource to the time of the application in functional block, then the following formula is valid: $v_{ij\ new}^{Sg} = v_{ij}^{Sg}\frac{\tau_{ij}^{Sg}}{t_{ij}}$, where $v_{ij\ new}^{Sg}$ is the indexed volume of the *g*-th type resource used during the service time of the *i*-th type of service request in the *j*-th FB. New matrices are formed $V_{new}^{Sg} = \{v_{ij\ new}^{Sg}\}$.

It is necessary to determine_a method of managing the incoming flow of applications, which allows avoiding a shortage of system resources.

VI. ALGORITHM OF THE METHOD

Deciphering the symbols used in the algorithm is given after it.

a. For each *i*-th type of service, set the permissible number of applications that can simultaneously enter the system for service ($k_{t\,доп}$). The number of admissible applications depends on the time discretisation interval, the discrete time reference system must be the same for the entire system. Remarks. The method of determining the permissible number of applications, which is solved as a dynamic programming problem (machine loading problem), was being considered.

b. The set $F = \{\emptyset\}$ is given. For each type of resource , $g = \overline{1,G}$ there is a maximum element $v^{sg}_{i_g j_g} = \max\{v^{sg}_{11\,new}, v^{sg}_{12\,new}, \ldots, v^{sg}_{mn\,new}\}$ in the matrix V^{sg}_{new}, pairs of indices $(i_{g1} j_{g1})$ of the corresponding elements are added to the set F. If the matrix contains two or more $(gmax \geq 1)$ maximum elements $v^{sg}_{i_{g\,1} j_{g\,1}} = \cdots = v^{sg}_{i_{g\,gmax} j_{g\,gmax}}$, then all pairs of indices are added to the set F and denoted by ($i_{g1} j_{g1}$, ..., $i_{g\,gmax} j_{g\,gmax}$). Indices of maximum volume values for different types of resources may coincide; repeated values are not added to the set F. For example, $i_{11} j_{11} = i_{21} j_{21} = 2\,3$, this means that for resource 1 and for resource 2, the first maximum element corresponds to the service process of the service request of the second type of service in the third functional block. That is, this service is the most expensive for the resources of the first and second types, in this case, the pair (2 ,3) will enter the set F once. Thus, the set F is filled with pairs, where the first position is the number of the service, the service of which is resource-consuming in the functional block, whose number is in the second position. *Remark.* Number pairs do not store the resource type, as this is irrelevant for this control method.

c. The elements of the set *F* are ordered by the first element. The set F is divided into *m* subsets, so that F_1 includes pairs where the first element is equal to 1, F_2 includes pairs where the first element is equal to 2, etc. If some *r*-th subset $(r \epsilon \overline{1,m})$ is empty ($F_r = \{\emptyset\}$), then requests of the *r*-th service type will not be subject to delays of requests arriving in bursts of the input flow load. For all subsets of F_d ($d \epsilon \overline{1,m}$), which contains one element, the actions of item 4 are performed. For all subsets F_p $(p \epsilon \overline{1,m})$, which contains two or more elements, the actions from point 5 are performed.

d. The task of this point is to determine the maximum delay of the excess number of requests of the *d*-th type of service, which arrived at the moments of peak loads of the incoming flow. If the set F_d contains one element (d, f_d), this means that for the application of the *d*-th type of service, two load peaks cannot be allowed during the service time in the functional block f_d, the duration of which is determined from the matrix T and is equal to $t_{d\,f_d}$. Go to point *f*.

e. The task of this point is not only to prevent two load peaks from occurring on one critical (resource-consuming) functional block, but also to avoid superposition when two load peaks are served in two resource-consuming functional blocks. For this, the elements of the subset F_p are ordered by the second element. Supposed that the set F_p consists of two elements: $(p, f1_p)$ and $(p, f2_p)$, problems with a larger number of elements are unlikely and are solved in a similar way. This means that when servicing applications of the *p*-th type of service, functional blocks with the numbers $f1_p$ and are resource-consuming $f2_p$. The elements with the corresponding indices are selected from the matrix T: $t_{p\,f1_p}$, $t_{p\,f2_p}$. The conditions under which two load peaks will not fall on one functional unit are the following:

- the distance between load peaks cannot be less than the value of $t_{p\,f1_p}$.

- the distance between load peaks cannot be less than the value of $t_{p\,f2_p}$.

- if $f2_p - f1_p = x > 1$, then the distance between load peaks is not allowed to be greater than $\sum_{q=1}^{x-1} t_{p\,(f1_p+q)}$.

f. During the operation of the monitoring system, moments of peak loads are recorded, when the number of applications received in the system is greater than the permissible value (according to clause 1). Moments of time when a load spike is detected are added to the sets $T_{i\,max}$, where and is the type of service for which the load spike was recorded. For services of the *r*-th type (see clause 3), the set $T_{r\,max}$ is not created.

For services of type d, for elements of the set $T_{d\,max}$, the condition of item 4 is checked, that is, for each new element of the set, $t_{d\,maxw+1}$the value is checked $t_{d\,f_d} - (t_{d\,maxw+1} - t_{d\,maxw}) = y1$, if *y*1>0, then part of the applications $(k_d(t_{d\,maxw+1}) \quad k_{d\,gen})$is delayed for time y1. If at the moment $(t_{d\,maxw+1} + y1)$, the number of applications received $k(t_{d\,maxw+1} + y1)$ plus the balance $(k_d(t_{d\,maxw+1}) - k_{d\,gen})$ in the amount give a value greater than the permissible value $k_{d\,gen}$. Then the

excess is transferred to the next moment of time $(t_{d\,maxw+1} + y1 + 1)$, the load smoothing procedure.

For services of type p, for elements of the set $T_{r\,max}$, the conditions of Clause 5 are checked, that is, for each new element of the set, $t_{p\,maxw+1}$ the conditions are checked:

- value $t_{p\,f1_p} - (t_{p\,maxw+1} - t_{p\,maxw}) = y2$, if y2>0, then part of the requests $(k_p(t_{p\,maxw+1}) - k_{p\,доп})$ is delayed for time y2, if necessary, the load smoothing procedure is applied
- value $t_{p\,f2_p} - (t_{p\,maxw+1} - t_{p\,maxw}) = y3$, if y3>0, then part of the applications $(k_p(t_{p\,maxw+1}) - k_{p\,доп})$ is delayed for time y2, if necessary, the load smoothing procedure is applied.
- if $f2_p - f1_p = x > 1$, then the value of is examined $\sum_{q=1}^{w-1} t_{p\,(f1_p+q)} - (t_{p\,maxw+1} - t_{p\,maxw}) = y4$, if y4<0, then part of the requests $(k_p(t_{p\,maxw+1}) - k_{p\,доп})$ is delayed for time $(t_{p\,f2_p} + y4)$, if necessary, the load smoothing procedure is applied.

Thus, in the case of the second load peak arrival event registration during the time defined by the conditions, the excess number of applications is delayed for the time determined by the algorithm of the method, after which the delayed applications are sent to the system in such a way as to prevent the creation of a load peak.

The number of applications that is permissible for a given type of service is calculated by the method of redistribution of technical means between applications of different types of services described above, while taking into account the efficiency of servicing all types of services with the available amount of system resources.

Modelling of the dynamic method management incoming load

Simulation modelling of the method of managing the flow of tariff applications was carried out. The GPSS package was used for simulation.

In the process of simulation modelling, a model for two resources and a flow of services was studied. Resources taken into account during simulation - RAM and Permanent storage.

The application processing process includes four functional blocks. The work of the functional blocks simulated such operations as: extracting the subscriber's information from the database, calculating the cost of the service, generating a notification for the subscriber, final calculation and debiting of funds.

To provide service, a given amount of resources was allocated, calculated for the simultaneous service of 50,000 applications for tariffication, provided that the number of applications is evenly distributed between functional units. During the service of the application in the functional block, the corresponding amount of the resource was blocked, and was released when moving to the next functional block. If the request is received for service in the functional block, but the resource is not enough for the service, then the request is delayed until the necessary amount of resource is released. At each stage, the time the application remains in the system is checked and compared with the allowable service time. The values were chosen as close as possible to real systems.

The inlet flow was modelled according to Poisson's law. Based on the analysis of the operation of the real system, the most resources were spent during the formation of the message to the subscriber. Therefore, the model monitored the number of applications that were served at the current moment in time in the third functional block and delayed messages until the number of applications became less than the maximum allowable number.

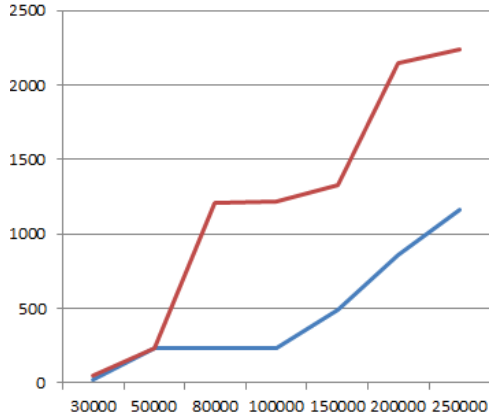

Fig. 3 Number of lost applications

The dynamics of the lost applications number depending on the intensity of the incoming flow was shown in Fig. It shows a decrease in the loss of applications due to exceeding the permissible service time. The line of packet loss without applying the proposed method of controlling the incoming flow was

marked in red, and the result of simulation using the proposed control method in blue.

VII. CONCLUSION

This study has successfully developed and validated a novel method for horizontal server load balancing aimed at reducing energy consumption while maintaining optimal service quality in hybrid telecommunication flows within cloud environments. The primary contribution of this research lies in addressing the scientific problem of distributing limited computing resources among virtual spaces that cater to various types of telecommunication services. Through comprehensive simulation using the General Purpose Simulation System (GPSS) package, the research has demonstrated the technical feasibility of maintaining telecommunication services in virtual environments with predetermined computing resources. The analysis of an online tariff system of a telecommunications operator provided a realistic context for the simulation, highlighting the critical role of efficient resource management in supporting the online charging process.

The study's key innovation is the dynamic load balancing method that incorporates a delay mechanism to manage peak loads and prevent resource overconsumption. By dividing service operations into distinct functional blocks, each requiring specific server resources such as RAM, processor time, and permanent memory, the method ensures a more balanced and efficient utilization of these resources. This approach significantly reduces application loss and enhances the economic efficiency of service maintenance. Moreover, the proposed two-level control system for managing incoming application flows and corresponding server resources has been shown to effectively smooth out load peaks, ensuring a uniform application flow. This system's architecture not only prevents server resource overloads but also improves the overall quality of service provided to subscribers.

REFERENCES

1. Smith, J., & Johnson, R. Dynamic Load Balancing in Cloud Computing Environments: A Review. International Journal of Cloud Computing, 2020, 12(3), pp. 245-261.
2. Wang, L., & Li, H. Energy-Aware Load Balancing Techniques for Cloud Computing: A Survey. Proceedings of the IEEE International Conference on Cloud Computing, 2019, pp. 158-165.
3. Islam, S., Keung, J., Lee, H., & Huh, E. N. "A Review on Distributed Application Processing Frameworks in Smart Mobile Devices for Mobile Cloud Computing." Wireless Personal Communications, vol. 82, no. 1, 2015, pp. 597-619.
4. Saez, S., & Garcia-Valls, M. "Mobile Edge Computing: A Survey." IEEE Internet of Things Journal, vol. 5, no. 1, 2018, pp. 450-465.
5. Li, X., Chen, M., & Li, M. "Resource Management for Mobile Edge Computing: A Survey." IEEE Access, vol. 7, 2019, pp. 66598-66610.
6. Dinh, H. T., Lee, C., Niyato, D., & Wang, P. "A Survey of Mobile Cloud Computing: Architecture, Applications, and Approaches." Wireless Communications and Mobile Computing, vol. 13, no. 18, 2013, pp. 1587-1611.
7. Meng, X., Isci, C., Kephart, J., Zhang, L., & Bouillet, E. "Efficient Resource Management in Computer Clouds." Proceedings of the 2010 IEEE International Conference on Data Engineering, 2010, pp. 828-831.
8. Garg, S. K., & Buyya, R. "NetworkCloudSim: Modelling Parallel Applications in Cloud Simulations." Proceedings of the 2011 Fourth IEEE International Conference on Utility and Cloud Computing, 2011, pp. 105-113.
9. Kaur, K., Garg, S., & Singh, H. "Energy Efficient Resource Allocation in Cloud Computing: A Survey of Various Techniques." Proceedings of the 2016 International Conference on Computing, Communication and Automation (ICCCA), 2016, pp. 48-53.
10. Wu, H., & Buyya, R. "Load Balancing in Cloud Computing: A Taxonomy, Survey, and Future Directions." IEEE Transactions on Cloud Computing, vol. 9, no. 1, 2021, pp. 238-261.
11. Wang, S., Zhou, A., & Yang, P. "Efficient Resource Allocation in Mobile Edge Computing: A Reinforcement Learning Approach." IEEE Wireless Communications, vol. 25, no. 3, 2018, pp. 115-121.
12. Merseedi, K. J., & Zeebaree, S. R. (2024). The Cloud Architectures for Distributed Multi-Cloud Computing: A Review of Hybrid and Federated Cloud Environment. Indonesian Journal of Computer Science, 13(2).
13. Globa, L., Skulysh, M., Romanov, O., & Nesterenko, M. (2018, November). Quality control for mobile communication management services in hybrid environment. In The International Conference on Information and Telecommunication Technologies and Radio Electronics (pp. 76-100). Cham: Springer International Publishing.
14. Skulysh, M. A., Globa, L. S., & Sulima, S. V. (2016). Model for Efficient Allocation of Network Functions in Hybrid Environment.
15. Skulysh, M. A., Romanov, O. I., Globa, L. S., & Husyeva, I. I. (2019). Managing the process of servicing hybrid telecommunications services. Quality control and interaction procedure of service subsystems. In Advances in Soft and Hard Computing (pp. 244-256). Springer International Publishing.

Скулиш М.А., Umakoglu Inci

Метод горизонтального балансування навантаження на сервер для зменшення енергоспоживання

Проблематика. Горизонтальне балансування навантаження на сервер є важливим аспектом сучасних обчислювальних систем, особливо в хмарних середовищах. Ефективне управління вхідними потоками додатків має важливе значення для забезпечення оптимального використання ресурсів і мінімізації енергоспоживання. Це

дослідження присвячено розробці методу управління вхідним потоком заявок для зменшення споживання енергії при горизонтальному балансуванні навантаження на сервер.

Мета досліджень. Основною метою є розробка методу управління вхідним потоком заявок для зниження енергоспоживання при горизонтальному балансуванні навантаження на сервер. Це передбачає визначення максимально допустимої кількості додатків, які можуть одночасно надходити в систему для обслуговування, забезпечуючи при цьому, щоб обсяг використовуваних ресурсів був близький до сумарного максимально можливого обсягу ресурсів. Метод спрямований на мінімізацію дисперсії елементів послідовності максимально допустимої кількості заявок та дисперсії елементів послідовностей обсягів використаних ресурсів.

Методика. Метод включає декілька ключових кроків:

Схема згладжування вхідного навантаження: Для згладжування вхідного навантаження пропонується статичний метод управління. Для цього розробляється схема згладжування вхідного навантаження, яка являє собою набір значень максимально допустимої кількості заявок (послідовність {ki}), що надходять на вхід системи за малий інтервал часу Δti. Послідовність вибирається таким чином, щоб обсяг використовуваних ресурсів був близьким до сумарного максимально можливого обсягу ресурсів.

Генетичний алгоритм: Вибір послідовності {ki} здійснюється за допомогою генетичного алгоритму. Алгоритм включає операції кросинговеру, мутації та відбору для мінімізації дисперсії елементів послідовності та дисперсії елементів послідовностей обсягів використаних ресурсів.

Розподіл ресурсів: Метод передбачає виділення ресурсів для обслуговування заданого типу сервісу. Параметри сервера, які характеризуються як ресурси системи, що обслуговує додатки, зазвичай розраховуються для середніх значень параметрів вхідного потоку.

Введення затримки: Для управління процесом обробки заявок і запобігання дефіциту ресурсів вводиться затримка для частини заявок, які збігаються зі сплеском навантаження. Час затримки визначається таким чином, щоб затримані заявки не потрапляли в систему до тих пір, поки попередній сплеск навантаження не буде успішно обслужений в ресурсоємному функціональному блоці.

Результати. До результатів дослідження можна віднести розробку методу управління вхідним потоком заявок для зменшення енергоспоживання при горизонтальному балансуванні навантаження сервера. Метод передбачає використання генетичного алгоритму для вибору послідовності {ki}, що мінімізує дисперсію елементів послідовності та дисперсію елементів послідовностей об'ємів використаних ресурсів.

Висновки. В результаті дослідження зроблено висновок, що запропонований метод управління вхідним потоком заявок дозволяє ефективно зменшити енергоспоживання при горизонтальному балансуванні навантаження на сервер. Метод полягає у використанні генетичного алгоритму для вибору послідовності {ki}, що забезпечує ефективне використання системних ресурсів та мінімізує дисперсію елементів послідовності та дисперсію елементів послідовностей об'ємів використаних ресурсів. Метод може бути застосований у різних сценаріях, де ефективне використання системних ресурсів є критично важливим, наприклад, у середовищах хмарних обчислень.

Ключові слова: *Телекомунікаційні послуги; хмарне середовище; управління ресурсами; балансування навантаження; динамічний метод; GPSS; обробка додатків; енергоефективність.*