В. П. Мочалов [V. P. Mochalov],
Н. Ю. Братченко [N.Yu. Bratchenko],
И. С. Палканов [I. S. Palkanov],
Э. В. Алиев [E. V. Aliev]

УДК 004.031.43
DOI: 10.37493/2307-910X.2022.4.4

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ СИСТЕМЫ БАЛАНСИРОВКИ НАГРУЗКИ СЕРВЕРОВ КЛАСТЕРОВ ЦОД В УСЛОВИЯХ ФРАКТАЛЬНОЙ НАГРУЗКИ

MATHEMATICAL MODEL OF THE LOAD BALANCING SYSTEM OF DPC SERVER CLUSTERS UNDER FRACTAL LOAD CONDITIONS

ФГАОУ ВО «Северо-Кавказский федеральный университет», институт цифрового развития, кафедра инфокоммуникаций, n.b.20062@yandex.ru/ North Caucasus Federal University, Institute of Digital Development, Department of Infocommunications, n.b.20062@yandex.ru

Аннотация

Математическая модель используемая для центров обработки данных (ЦОД), обеспечивающая решение задачи оценки ее производительности с учетом степени загруженности. Работоспособность предложенной модели и проверка полученных результатов осуществлена путем имитационного моделирования. В качестве основных показателей качества использованы характеристики средней длины очереди, средней задержки и вероятности потери пакетов. Математическим аппаратом для оценки данных показателей качества является теория массового обслуживания. Система распределения и балансировки нагрузки представлена как многоканальная система с ограничением по длине очереди, включающая в свой состав неограниченный буфер (дисковую память) для всех серверов кластера, а также входные буферы ограниченной емкости для каждого сервера. Модель построена с учетом особенностей сетевого трафик современных инфокоммуникационных сетей, характеризуемого свойствами самоподобия, причем каждый вид трафика (HTTP/TCP, HTTPS. SMTP/TCP, VoIP, FTP/TCP, IP, Ethernet, ATM) описывается только характерным для него законом распределения как интервалов поступления пакетов, так и длинами протокольных блоков. Для учета особенностей поступающего в систему самоподобного сетевого трафика производится его описание фрактальным броуновским движением fBM/M/1/N и специальной функцией, зависящей от коэффициента самоподобия H (коэффициент Херста). Представленная модель может быть использована также для проведения исследований характеристик сетевого трафика с целью предотвращения сетевых перегрузок и минимизации потерь.

Ключевые слова: сетевой трафик, самоподобие, долговременная зависимость, кластеры серверов, балансировка нагрузки.

Abstract

A mathematical model of the system for distributing and balancing the load of servers of clusters of data processing centers (DPC) is proposed, which provides a solution to the problem of assessing its performance, taking into account the degree of workload. The performance of the proposed model and the verification of the results obtained were carried out by simulation. The characteristics of the average queue length, average delay, and packet loss probability were used as the main quality indicators. The mathematical apparatus for evaluating these quality indicators is the queuing theory. The load distribution and balancing system is presented as a multi-channel system with a limit on the length of the queue, which includes an unlimited buffer (disk memory) for *all servers in the cluster, as well as input buffers of limited capacity for each server. The model is built taking into account the features of the network traffic of modern infocommunication networks, characterized by self-similarity properties, and each type of traffic (HTTP/TCP, HTTPS. SMTP/TCP, VoIP, FTP/TCP, IP, Ethernet, ATM) is described only by its characteristic distribution law as packet arrival intervals and protocol block lengths. To take into account the features of the self-similar network traffic entering the system, it is described by the fractal Brownian motion fBM/M/1/N and a special function that depends on the self-similarity coefficient H (Hurst coefficient). The presented model can also be used to study the characteristics of network traffic in order to prevent network congestion and minimize losses.*

Key words: network traffic, self-similarity, long-term dependence, server clusters and load balancing.

Introduction

In connection with the widespread use of data processing centers (DPCs), the task of optimizing their parameters becomes relevant, the solution of which depends on the characteristics of the system for distributing and balancing the load of data center cluster servers, performance indicators and throughput of its communication system. It is known that the traffic of modern infocommunication networks with packet switching has statistical characteristics associated with fractal (self-similar) processes, characterized by a measure of stability of long-term dependence, fractal dimension, correlation parameters, spectral and fractal indicators [1-5]. The flow of applications entering the service in the data center can be described by fractal Brownian motion, since network traffic such as HTTP/TCP, FTP/TCP, SMTP/TCP, VoIP, IP, Ethernet, ATM has self-similar or fractal characteristics [6]. Classical queuing models, characterized by the exponential law of distribution of both the incoming load and its processing, are not applicable here. Therefore, one of the possible solutions to this problem is the development and application of algorithms and models for the efficient distribution of tasks within data center clusters in order to optimize the use of resources and reduce computation time under fractal load conditions. Taking into account the fractality (self-similarity) of the load entering the data center provides high-quality service for modern high-speed digital data streams, a rational choice of data center equipment, and the elimination of congestion and queues.

Despite the large number of publications that offer effective approaches to solving this problem [7,9,13,17], the problem of rational resource allocation under the conditions of a fractal (self-similar) traffic structure of various applications and services remains relevant. The article proposes analytical and simulation models of a load distribution and balancing system that allow determining the main probabilistic and temporal characteristics of the processes of its interaction with a cluster of data center servers under different server loads in a fractal structure of network traffic.

The purpose of this article is to develop analytical and simulation models of a load balancing system for servers in data center clusters under conditions of a self-similar structure of network traffic.

1. Research methodology

One of the most important elements of a data center server cluster is the load distribution and balancing system, shown in Figure 1, which provides solutions to the problems of managing cluster resources, distributing requests and their corresponding applications.

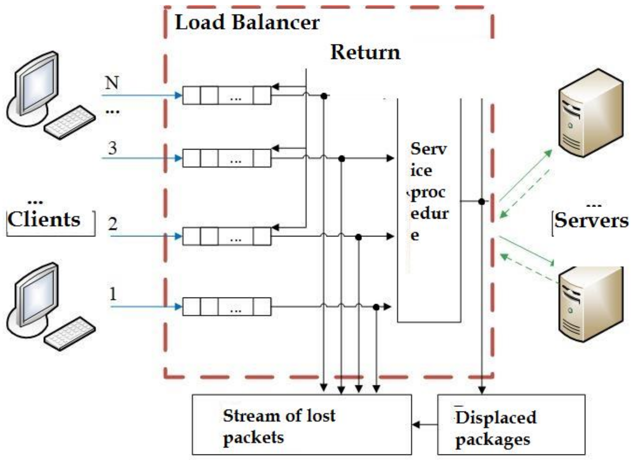

Figure 1 - Load distribution and balancing system

At the same time, the duration of servicing a data center request depends on the current state of the load balancer, the workload of the servers and the switching system. Overloading this system leads to uneven server load, a decrease in the efficiency of distribution and balancing algorithms, and a decrease in the quality of information services. The load balancing system model is shown in Figure 2.

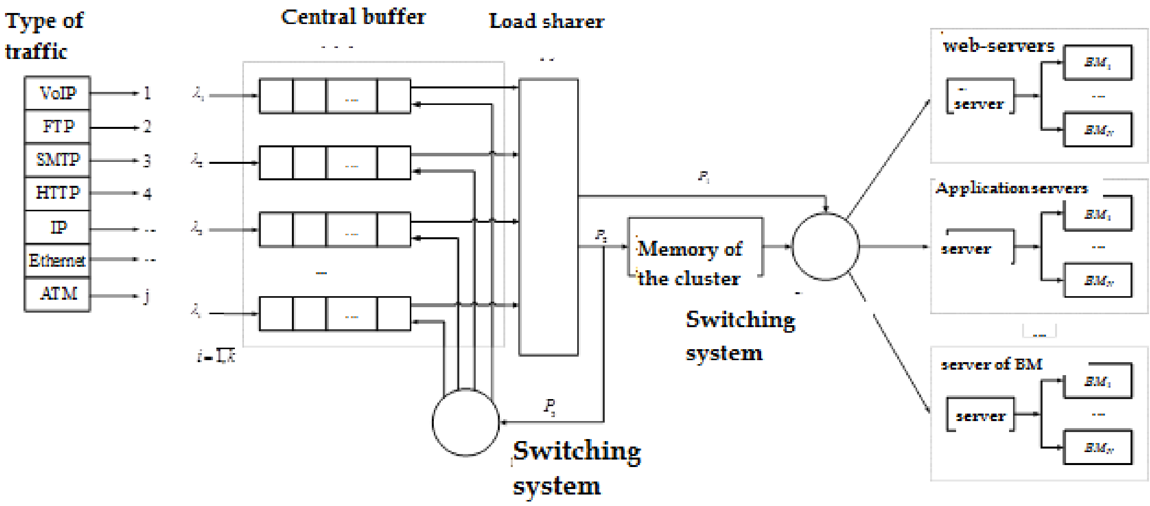

Figure 2 - Model of the load balancing system

The input of the load balancer receives independent streams of self-similar traffic of various types λ_i, $i=\overline{1,k}$, characterized by scale invariance, distributions with "heavy tails", parameters of fractal Brownian motion.

The distribution density of the duration of traffic processing when it enters the load balancer, when all cluster servers are free, is equal to $f_{i1}(t)$. Let us denote the probability of such an event P_1.. The duration of traffic processing has a distribution density $f_{i2}(t)$ if it arrives at the moment when the cluster servers are busy and the memory is free. We denote the probability of such an event as P_2. The duration of traffic processing has a distribution density $f_{i3}(t)$ if it arrived at the time when both the servers and the cluster memory are busy. The probability of such an event P_3.

We consider that $n_{i1}(t)$ is the number of requests of the i -th type that arrived at the load balancer during the interval $(0,t)$ in an unloaded system, when the central buffer, internal memory, and cluster servers are free.

The number of requests of the i -th type, received in the time interval $(0,t)$, when the servers are busy, but there is no queue for processing, we denote $n_{i2}(t)$.

The number of requests of the i -th type, received in the time interval $(0,t)$, when the internal memory and servers are busy, will be denoted by $n_{i3}(t)$. The average service time for requests of the i -th type will be determined by the expression

$$T_{cp.} = n_{ij}(t) \cdot \overline{S}_{ij},$$

where $j = 1,2,3$, $\overline{S}_{ij}$ is the average processing time of an application.

balancer idle time in stationary mode

$$T_{cв.} = t - \sum_{i=1}^{k} \sum_{j=1}^{3} n_{ij}(t) \cdot \overline{S}_{ij} .$$

The total number of requests in the system at $t \to \infty$ will be equal to

$$n_{ij}(t) = \lambda_i \cdot (t - \sum_{i=1}^{k} \sum_{j=1}^{3} n_{ij}(t) \cdot \overline{S}_{ij}) .$$

The number of requests received during the server busy time interval will be equal to

$$n_{i2}(t) + n_{i3}(t) = \lambda_i \cdot \sum_{i=1}^{k} \sum_{j=1}^{3} n_{ij}(t) \cdot \overline{S}_{ij} .$$

The probability that the Δt system will not receive i -th requests for service in time will be equal to

$$P_{\Delta t} = \prod_{i=1}^{\Delta t} e^{-\lambda_i \Delta t} = e^{-\Lambda \Delta t},$$

where $\Lambda = \sum_{i=1}^{\Delta t} \lambda_i$.

The probability of receiving a service request if the system is busy during the time S is equal to

$$P_S = \sum_{h=0}^{S} \lambda_i e^{-\lambda h} = \frac{\lambda_i (1 - e^{-\lambda S})}{\Lambda} .$$

It follows from this that at $t \to \infty$, $n_{i2}(t)$ it will be equal to

$$n_{i2}(t) = \frac{\lambda_i}{\Lambda} \sum_{i=1}^{k} \sum_{j=1}^{3} n_{ij}(t) \left[1 - \sum_{S} f_{ij}(S) e^{-\lambda S} \right].$$

As follows from Figure 2, the probability of delay in processing an application P_d, is determined by the formulas [15]:

$$P_d = P_0 \frac{a_k^N}{N!(1-\rho_k)}; \quad P_0 = \left[\sum_{k=0}^{N-1} \frac{a_k^N}{k!} + \frac{a_k^N}{N!(1-\rho_k)} \right]^{-1},$$

where $a_k = \frac{\Lambda}{N\mu_k}, \quad \rho_k = \frac{\Lambda}{N\mu_k}, \quad \Lambda = \sum_{i \in I} \lambda_i, \quad \mu_k = \frac{\sum_{i \in I} \lambda_i}{\sum_{i \in I} \lambda_i \tau_i},$

Λ, μ_k, τ_i - the intensity of receipt, processing of applications and the time of their processing.

The probability that an application will be processed without a queue is

$$P_1 = 1 - P_d.$$

The probability of a request arriving in one of the segments of the central buffer of the system is equal to P_3 and is determined by the expression

$$P_3 = \frac{(1-\rho_k)\rho_k^S}{1-\rho_k^{S+1}}.$$

The probability that applications are waiting for service in the internal memory of the cluster, when the servers are busy, can be obtained from the condition

$$P_1 + P_2 + P_3 = 1.$$

Then, taking into account the degree of loading of the cluster servers, the average time for processing an application is equal to

$$t_k = \left[\tau_S P_1 + (\tau_w + \tau_S)P_2 + \tau_r P_3\right]\frac{1}{1-P_3}.,$$

where τ_S is the statistical service time of the request, τ_w is the statistical waiting time for service, τ_r is the fixed time of the request being in the central buffer.

The total input traffic of the data center, obtained by combining a large number of flows from various sources, each of which is characterized by its own distribution law, is characterized by fractal (self-similar) properties, ignoring which leads to a decrease in quality of service indicators (an increase in queue lengths on interfaces of buffer memory blocks, average delays packet flows, loss probabilities), causes an overload of the elements of the center. For example, approaching the coefficient of self-similarity of the input traffic H to 1, for systems oriented to the processing of Erlang flows, leads to significant traffic losses and requires a significant increase in the amount of buffer memory to maintain a given quality of service [8,11,14]. Therefore, to study this system, a queuing system was used, the load of which is described by the characteristics of the fractal Brownian motion. Fractal Brownian process with Hurst exponent $0,5 < H < 1$refers to a random process in which for a normally distributed random variable $A^H(t)$, the mean value $A^H(t)$is zero for any t , and for the covariance $A^H(t)$and $A^H(s)$, the expression [16,18] is true $A^H(0) = 0$

$$E\left[A^H(t)\cdot A^H(s)\right] = \frac{1}{2}\left(t^{2H} + s^{2H} - |t-s|^{2H}\right).$$

The dispersion $A^H(t)$is proportional t^{2H},and the trajectory has a fractal dimension $D = 2 - H$.

Thus, a fractal Brownian process is a Gaussian process of the form

$$A(s,t) = N\left(m(t-s), v(t-s)\right),$$

where $A(s,t)$is the amount of data; $m = 1,...,M$; $s < t$; $A(s,t) = A(t) - A(s)$; $Var\left(A(s,t)\right) = V(t-s)$.

For this process $\left\{A(t), t \geq 0\right\}$ $E\,A(t) = mt$, dispersion $V(t) = t^{2H}$, $H \in (0,1)$ is the Hurst exponent .

It is known that such a process is characterized by a long-term dependence at $H > 0,5$, and its parameters are close to those of a Gaussian process with mathematical expectation *nmt* and dispersion $V(t)$[10, 12]. This makes it possible to use Gaussian models to describe self-similar traffic.

The formulas for calculating the queue in the central buffer memory block, packet delay and the probability of its loss due to buffer overflow are presented below [19]:

$$n_{i,j}=\frac{\frac{\rho_{i,j}}{\pi}f(H)}{1-\left[\frac{\rho_{i,j}}{\pi}f(H)\right]^{N+2}}\cdot\frac{\left\{1-(N+1)\left[\frac{\rho_{i,j}}{\pi}f(H)\right]^{N}\right\}+N\left[\frac{\rho_{i,j}}{\pi}f(H)\right]^{N+1}}{1-\frac{\rho_{i,j}}{\pi}f(H)}-\frac{\rho_{i,j}}{\pi}f(H),$$

$$\tau_{i,j}=\frac{n_{i,j}+\frac{\rho_{i,j}}{\pi}f(H)}{f(H)\cdot\sum_{k\in K}x_{i,j}^{k}\cdot d_k}+\frac{1}{c_{i,j}}=\frac{\frac{1}{\pi c_{i,j}}}{1-\left[\frac{\rho_{i,j}}{\pi}f(H)\right]^{N+2}}\cdot\frac{\left\{1-(N+1)\left[\frac{\rho_{i,j}}{\pi}f(H)\right]^{N}\right\}+N\left[\frac{\rho_{i,j}}{\pi}f(H)\right]^{N+1}}{1-\frac{\rho_{i,j}}{\pi}f(H)}+\frac{1}{c_{i,j}},$$

$$p_{i,j}=\frac{1-\frac{p_{i,j}}{\pi}f(H)}{1-\left[\frac{p_{i,j}}{\pi}f(H)\right]^{N+2}}\left[\frac{p_{i,j}}{\pi}f(H)\right]^{N+1},$$

where $\rho_{i,j}$ is the channel load $(i,j)\in E$, namely: $\rho_{i,j}=\frac{\sum_{k=1}^{K}x_{i,j}^{k}\cdot d_k}{c_{i,j}}$; π– probability of no packet loss at the buffer memory input; $c_{i,j}$ – channel capacity $(i,j)\in E$; H is the Hurst exponent .

2. Model experiment

The simulation model of the load balancing system was developed and implemented in the AnyLogic simulation environment using the Java programming language. A simplified logical structure of the model is shown in Figure 3.

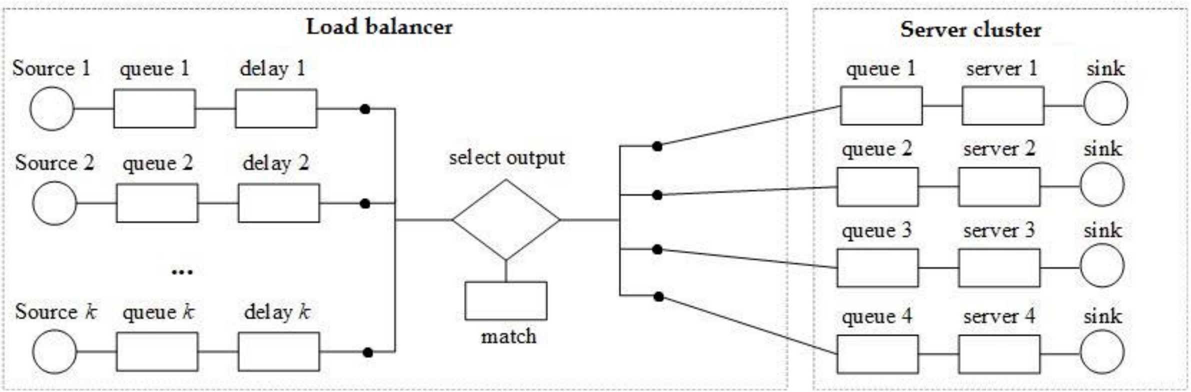

Figure 3 - Logical structure of a server cluster in the AnyLogic environment

Source - request generator;
queue – queue formation;
delay – delay generation;
match – transmission matching block;
select output – load balancing block;
sink - termination block.

To build the logic of the service request distribution system model, the Enterprise library was used. library. When implementing the load balancing module, the Least algorithm was used Connection, which takes into account the current load of the server cluster. The level of requested server resources is evenly distributed. The switching system and communication channels are limited to a bandwidth of 1 Gb/s. The cluster under study includes 10 servers with performance distributed in the range (0-1). An unlimited amount of segmented central buffer memory is assumed. The time spent by the request in the central buffer, when the cluster memory is occupied, is taken equal to 0.01 sec. The law of distribution of intervals for the arrival of packet streams is described by the characteristics of the fractal Brownian motion fBM / M /1/ N. The protocol block durations are described by a deterministic distribution law with an average value of 1000 bytes. The system input receives 50 input streams, with an intensity of 60 messages / hour=0.017 messages /sec. with uniform distribution between the elements of the central buffer. Load balancer service direction - switching system - server cluster is modeled by the system $M|M|N$, load balancer service direction - cluster memory - switching system - server cluster is modeled by the system $M|M|N|m$, service direction load balancer - central buffer is modeled system $M|M|N|m(V)$.

When estimating the memory size of a server cluster, the probability of message loss is taken equal $P_n = 0{,}01$and is estimated by the expression [20]: $P_n = 1 - R(V),$ where $R(V) = \gamma(p, qx)/\Gamma(p)$, $p = r_1^2/(r_2 - r_1^2)$, $q = r_1/(r_2 - r_1^2)$, $r_1 = \delta_1 + \varphi_1$, $r_2 = \varphi_2 + 2\varphi_1\delta_1 + \delta_2$, $\delta_1 = (1 - P_0)\, p/q$, $\delta_2 = (1 - P_0)\, p(p+1)/q^2$, where $\Gamma(p)$ is the gamma function; $\gamma(p, qx)$ is the element of the gamma function; p is the distribution parameter; r_1, r_2 is the moment of the total volume of messages; φ_1, φ_2 – moment of average message duration; $\varphi_1 = 1/p$; $\varphi_2 = (2 - p)\,p^2$; P_0 – probability of missing messages.

The stationary distribution function of the memory size of the server cluster is [20]

$$D(x) = P_0 + (1 - P_0)\frac{\gamma(p, qx)}{\Gamma(p)}.$$

The load redistribution between the cluster servers is set by the normal law. The results of the experiment are the dependences of the average packet processing time by the load balancer and the probability of their losses from the load of the cluster servers, as well as the dependence of the message processing time by the load balancer on the amount of cluster memory

Many of these metrics provide a measure of how well a load balancer is performing. The research results are presented in tables 1, 2, 3.

Table 1 - Dependence of the average processing time of network packets on the load of the servers of the data center cluster

Server loading	0.51	0.55	0.63	0.69	0.75	0.82	0.85
T sec.	0.11	0.27	0.32	0.74	1.13	1.76	2.01

Table 2 - Dependence of the average processing time of network packets on the amount of memory of the data center cluster

Cluster memory size	5	10	15	20	30	40	50
T sec.	0.14	0.11	0.09	0.07	0.05	0.04	0.03

Table 3 - Dependence of the probability of network packet loss on the load of the servers of the data center cluster

Server loading	0.51	0.55	0.63	0.69	0.75	0.82	0.85
$P_{i,j}$	0.01	0.02	0.03	0.04	0.05	0.06	0.07

Conclusion

One of the most important components of a data center server cluster is a balancing system that implements efficient load distribution algorithms in order to reduce the time for executing user

requests. The performance of the balancing system has a significant impact on the quality indicators of the entire server cluster. The article proposes analytical and simulation models for solving the problem of studying the main indicators of the quality of a given system, depending on the degree of loading of the servers of the data center cluster. The solution of the problem is based on determining the parameters of queuing systems. The proposed model is investigated in terms of queue length in the central buffer memory block, packet delay, and packet loss probability. The performance of the proposed model was verified by simulation on the AnyLogic platform using the Java programming language and the Enterprise library. As a result of the model experiment, dependencies were identified that affect the quality of user service indicators. The presented model can be used to study the characteristics of network traffic in order to prevent network congestion and minimize losses in the load balancing system of data center cluster servers.

ЛИТЕРАТУРА /REFERENCES

1. Computing Center of the Institute of High Energy Physics (IHEP-CC) (2016). "VCondor – virtual computing resource pool manager based on HTCondor". Retrieved from https://github.com/hep-gnu/VCondor.
2. McNab A., Love P., and MacMahon E. (2015). Managing virtual machines with Vac and Vcycle, J. Phys.: Conf. Ser., Vol. 664.
3. Feller E., Rilling L., and Morin C. Snooze (2012). A scalable and autonomic virtual machine management framework for private Clouds, Proceedings of the 12th IEEE/ACMInternational Symposium on Cluster, Cloud and Grid Computing (CCGrid), pp. 482-489.
4. Beloglazov, R. Buyya (2015). OpenStack Neat: A Framework for Dynamic and Energy-Efficient Consolidation of Virtual Machines in OpenStack Clouds, Concurrency and Computation: Practice and Experience (CCPE), Vol. 27, No. 5, pp. 1310-1333.
5. Anne-C´ecile Orgerie, Laurent Lef`ever (2009). When Clouds become Green: The Green Open Cloud Architecture, International Conference on Parallel Computing (ParCo), pp. 228-237.
6. Ward J.S., Barker A. (2014). Observing the clouds: a survey and taxonomy of cloud monitoring, Jour-nal of Cloud Computing: Advances, Systems and Applications, Vol. 3.
7. Ward J.S., Barker A. (2015). Cloud cover: monitoring large-scale clouds with Varanus, Journal of Cloud Computing: Advances, Systems and Applications, Vol. 4.
8. Open Grid Forum (2016). "Open Cloud Computing Interface". Retrieved from http://occiwg.org/.
9. Baldoni M., Baroglio M., Martelli A., "Verifying the conformance of web servicesto global interaction protocols: A first step", International Workshop on WebServices and Formal Methods, 2005, P. 27.
10. Egawa T., "SDN standardization Landscape fromITU-T Study Group 13 / T. Egawa," ITU Workshop on SDNGeneva, Switzerland, 4 June 2013.
11. Mochalov V.P., Bratchenko N.Y., Yakovlev S.V., "Analytical model of object request broker based on Corba standard" (2018) Journal of Physics: Conference Series, 1015 (2). doi: 10.1088/1742-6596/1015/2/022012.
12. A. Vishnu Priya; N. Radhika Performance comparison of SDN OpenFlow controllers // International Journal of Computer Aided Engineering and Technology, 2019, Vol.11 No.4/5, pp.467 – 479.
13. Nageswara S. V. Rao Performance Comparison of SDN Solutions for Switching Dedicated Long-Haul Connections // ICN 2016: The Fifteenth International Conference on Networks. pp. 110-117.
14. Idris Z. Bholebawa, Upena D. Dalal Performance Analysis of SDN/OpenFlow Controllers: POX Versus Floodlight // Wireless Personal Communications. January 2018, Volume 98, Issue 2, pp 1679–1699. https://doi.org/10.1007/s11277-017-4939-z.

15. Tong Li, Jinqiang Chen, Hongyong Fu Application Scenarios based on SDN: An Overview // IOP Publishing. IOP Conf. Series: Journal of Physics: Conf. Series 1187 (2019) 052067 doi:10.1088/1742-6596/1187/5/052067.
16. Boev V. Kompjuternoe modelirovanie: Posobie dlja prakticheskih zanjatij, kursovogo i diplomnogo proektirovanija v AnyLogic7 [Computer modeling: A manual for practical classes, course and diploma projects in AnyLogic7] St. Petersburg, VAS Publ., 2014, 432p. (In Russian)
17. Taihoon K., Soksoo K. Analysis of Security Session Reusing in Distribution Server System. Computational Science and Its Applications - ICCSA 2006. Springer, 2006, 1045 p.
18. Khritankov A. Modeli i algoritmy raspredelenija nagruzki. Algoritmy na osnove setej SMO [Models and algorithms of load balancing. Algorithms on the basis of networks of queuing systems]. Informacionnye tehnologii i vychislitel'nye seti – Information technologies and computer networks. 2009, vol. 3. (In Russian)
19. Ivanisenko I., Kirichenko L., Radivilova T. Metody balansirovki s uchetom multifraktal'nyh svojstv nagruzki [Balancer multifractal methods considering load characteristics]. International Journal "Information Contentand Processing". 2015, vol. 2, no. 4, pp. 345–368. (In Russian)
20. Panchenko T.V. Genetic Algorithms [Text]: teaching aid /Yu.Yu. Tarasevich. Astrakhan: «Astrakhanskiy Universitet»,2007. – 87 p. (In Russian)

ОБ АВТОРАХ | ABOUT THE AUTHORS

Мочалов Валерий Петрович, профессор, доктор технических наук профессор кафедры инфокоммуникаций ФГАОУ ВО «Северо-Кавказский федеральный университет», институт цифрового развития, кафедра инфокоммуникаций, n.b.20062@yandex.ru, +7 9624004447, (8652) 95-69-97

Mochalov Valeriy Petrovich, Doctor of Technical Sciences, Professor, SI NCFU, Stavropol, Russia, n.b.20062@yandex.ru, +7 9624004447, (8652) 95-69-97

Братченко Наталья Юрьевна кандидат физико-математических наук, доцент, n.b.20062@yandex.ru, +7 9887405050, (8652) 95-69-97

Bratchenko Natalia Yurievna, Candidate of Physical and Mathematical Sciences, Associate Professor, SI NCFU, Stavropol, Russia, n.b.20062@yandex.ru, +7 9624004447, (8652) 95-69-97

Палканов Илья Сергеевич, программист, ФГАОУ ВО «Северо-Кавказский федеральный университет», институт цифрового развития, кафедра инфокоммуникаций, n.b.20062@yandex.ru+7 9289633235

Palkanov Ilya Sergeevich, programmer, SI NCFU, Stavropol, Russia n.b.20062@yandex.ru+7 9289633235

Алиев Эльдар Вадимович, аспирант, ФГАОУ ВО «Северо-Кавказский федеральный университет», институт цифрового развития, кафедра инфокоммуникаций, n.b.20062@yandex.ru, (8652) 95-69-97

Aliev Eldar Vadimovich, graduate student, SI NCFU, Stavropol, Russia n.b.20062@yandex.ru, (8652) 95-69-97

Дата поступления в редакцию: 19.10.2022
После рецензирования:13.11.2022
Дата принятия к публикации:07.12.2022