Adv Syst Sci Appl 2020; **03**:105–112
Published online at https://ijassa.ipu.ru.

Estimating the Response Time of a Cloud Computing System with the Help of Neural Networks

Anastasia V. Gorbunova[1]*, Vladimir M. Vishnevsky[1]

[1] *V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Moscow, Russia*

Abstract: The article presents a new approach to assessing the average response time of a cloud computing system and its dispersion. A fork-join system or a system with request splitting was chosen as a queuing model, and artificial neural networks were used as a method for estimating a variable of interest. The analysis showed that the estimates obtained were more accurate than those previously known. Besides, the proposed approach allows expanding the analysis of the cloud system to the case of a model with a non-Poisson input stream and non-exponential service time, as well as obtaining estimates for a larger number of performance indicators of the cloud system, which was not previously possible.

Keywords: cloud computing, parallel computing, queuing system, parallel service of requests, average response time, multilayer perceptron, artificial neural networks, machine learning methods

1. INTRODUCTION

Cloud computing is a technology that allows a remote user to access various services that, as a rule, have significant computing resources upon request via the Internet [2]. At the same time, even despite the presence of powerful computing infrastructure, the task of ensuring the required level of quality of the services provided and, at the same time, a logical interest in reducing the overall load on the equipment continue to be relevant for cloud providers [1, 9, 10, 23]. This is due to the growing popularity of the services they provide for many objective reasons, as well as the understandable need to reduce costs due to, for example, increased energy efficiency.

One of the possible solutions to this problem is parallel data processing, which could be accomplished due to the use of virtualization technology, which results in a decrease in response time, and, accordingly, in maintaining or even improving the quality of the services provided in the cloud.

Thus, a cloud center consisting of several physical machines that can be used together to process complex user requests served in the order they arrive in the system is considered [5]. One of the most important indicators of the functioning of the cloud system is the response time, the evaluation of which has a large number of publications, for example, works [1,5,6,9,10,23] devoted to. Nevertheless, the study of this parameter is still in demand due to the relevance of the practical task, and as a consequence of the need to improve the quality of existing estimates of this characteristic.

The article is organized as follows: Section 2 describes the mathematical model of the cloud center, Section 3 discusses the possibility of applying a new approach to the analysis

[*] Corresponding author: avgorbunova@list.ru

of average response time, namely, artificial neural networks (ANNs), provides a numerical example, and finally, Section 4 highlights the prospects for further research.

2. THE MATHEMATICAL MODEL OF A CLOUD COMPUTING SYSTEM

A queuing system (QS) of the fork-join type is considered as a model of a cloud system. Systems of this type are also called “QS with parallel service of requests” [7, 8].

Fork-join systems are models for many real physical systems in which the parallelization of the tasks is performed. For example, using the method of datagrams in computer networks, when a message arrives at the source node, it is divided into packets that are transmitted by different routes to the address node, in the buffer of which the message is assembled. Other examples are the processes of assembling orders at the warehouse or the processes of functioning of high-performance applications in the production, medical, financial, scientific or any other areas, based on the concept of distributed or parallel computing [11, 15].

Features of the functioning of the QS with the splitting of requests are as follows (Fig. 2.1):

1) at the time the request enters the system, it is instantly split into K ($K \geqslant 2$) smaller components, i.e., sub-requests, each of which, in turn, becomes serviced by the device (if it is busy) or immediately begins to be serviced if the corresponding device is free. It is considered that each sub-request has its type, which must correspond to the number of the device on which it will be serviced;
2) after the service is ended the sub-request falls into the so-called synchronization buffer and remains there until all related sub-requests, that is, the sub-requests originally belonging to one request, finish their service. Then the whole request is instantly assembled and only after this the request is considered served and may leave the system.

Since modeling a cloud center in the context of solving complex computational problems is discussed in this case, virtual machines (VMs) will act as devices in the fork-join models. At the time of receipt, a complex user request is divided into K smaller components, each of which is processed on the corresponding VM. Then it enters the synchronization buffer, where it remains until the last of the subtasks is processed, after which the user request is considered served.

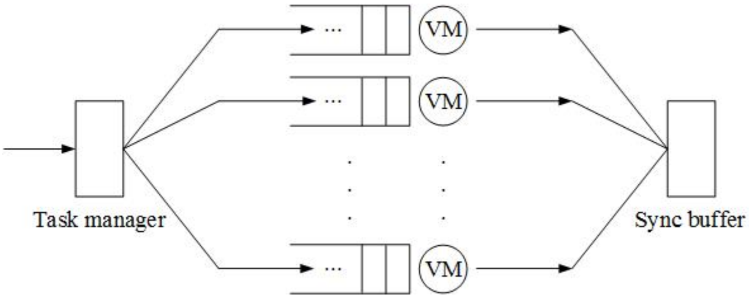

Fig. 2.1. The model of the cloud center system of the fork-join type

Note that other sub-request servicing schemes are known. For example, when devices are blocked until all parts of the request are serviced or the sub-requests are indistinguishable in the sense that any K sub-requests are required to assemble the request, etc. A more detailed overview can be found, for example, in [7, 20].

One of the main performance indicators of any queuing system is its response time. The fork-join system in this sense is no exception. Therefore, this concept is specified in the context of splitting requests. Since the request is considered to be served only at the moment when the last sub-request is serviced, to calculate the residence time of the request in the system, taking into account the fact that the moments of the appearance of all its sub-requests in the system coincide, it is sufficient to determine the maximum of all the times of the residence of its sub-requests.

However, the task of calculating the exact value of the average response time, even despite the Poisson nature of the incoming stream and the exponential service time on all devices in the case of splitting into more than 2 sub-requests, remains unresolved. The expression in closed form exists only for $K = 2$ [14], and for $K > 2$ only approximations of various degrees of accuracy were obtained [14, 21, 22]. This is explained by the complexity of the analysis of the residence times of the parts of the request in the system because of the existing relationship between them due to the general moments of receipt. The residence times of sub-requests in the system are positively associated random variables. Therefore, their maximum is stochastically no greater than the maximum of independent random variables with the same distribution.

Below are formulas that estimate the average response time in the system with splitting requests in the case of the same service intensities on all devices $\mu_k = \mu$, $k = \overline{1, K}$, and the incoming stream intensity equal to λ [14, 21, 22].

$$E[W_K] \approx \left[\frac{H_K}{H_2} + \frac{4}{11}\left(1 - \frac{H_K}{H_2}\right)\rho\right]\frac{12-\rho}{8}\frac{1}{\mu-\lambda}, \tag{2.1}$$

$$E[W_K] \approx \left[H_K + \left(\sum_{i=1}^{K}\binom{K}{i}(-1)^{i-1}\sum_{m=1}^{i}\binom{i}{m}\frac{(m-1)!}{i^{m+1}} - H_K\right)\frac{\lambda}{\mu}\right]\frac{1}{\mu-\lambda}, \tag{2.2}$$

$$E[W_K] \approx \frac{1}{\mu}\left[H_K + \frac{\rho}{2(1-\rho)}\left(\sum_{k=1}^{K}\frac{1}{k-\rho} + (1-2\rho)\sum_{k=1}^{K}\frac{1}{k(k-\rho)}\right)\right], \tag{2.3}$$

$$E[W_K] \approx \frac{1}{\mu-\lambda}H_K, \tag{2.4}$$

$$E[W_K] \approx \frac{1}{\mu-\lambda}\left(1 + \frac{K-1}{\sqrt{(2K-1)}}\right), \tag{2.5}$$

where $H_K = \sum_{i=1}^{K} 1/i$ is the partial sum of the harmonic series, and $\rho = \lambda/\mu < 1$ is the load factor. The first two of the above formulas are obtained due to the combination of the analytical approach with the empirical one, which consists in observing the behavior of the random variable under investigation through simulation and the subsequent use of the results of this observation in the preparation of the corresponding estimates. The third formula is derived by interpolation of high and weak input loads. Approximation (2.4) is the result of analyzing the average response time for K independent $M|M|1$ QSs operating in parallel, which is not only natural but also objectively justified assumption, since the expressions for the marginal probabilities of the number of k-type sub-requests ($k = \overline{1, K}$) in the original fork-join system coincide with the expressions for stationary probabilities in the $M|M|1$ system [5]. Finally, the last expression (2.5) from the above is the upper limit of the average value of the maximum of order statistics for independent exponentially distributed parameterized random variables with $(\mu - \lambda)$.

In addition to estimating the average response time, the approximations for variance, and higher moments of the studied value are worth noticing. These numerical characteristics make it possible to formulate a complete picture of the response time behavior. However, there is no longer such a variety of analytical expressions, and it is desirable to improve the existing estimates. So, in [6], an expression is presented for the variance of the average response time for both identical μ and different μ_k parameters of exponential service times, $k = \overline{1, K}$

$$D[W_K] \approx \sum_{l=1}^{K} \frac{2}{(\mu_l - \lambda)^2} - \sum_{1 \leqslant l < m \leqslant K} \frac{2}{(\mu_l + \mu_m - 2\lambda)^2} +$$

$$+ \sum_{1 \leqslant l < m < k \leqslant K} \frac{2}{(\mu_l + \mu_m + \mu_k - 3\lambda)^2} + \ldots + (-1)^{K-1} \frac{2}{(\mu_1 + \mu_2 + \ldots + \mu_K - K\lambda)^2} -$$

$$-\Bigg(\sum_{l=1}^{K} \frac{1}{\mu - \lambda} - \sum_{1 \leqslant l < m \leqslant K} \frac{1}{\mu_l + \mu_m - 2\lambda} + \sum_{1 \leqslant l < m < k \leqslant K} \frac{1}{\mu_l + \mu_m + \mu_k - 3\lambda} +$$

$$+ \ldots + (-1)^{K-1} \frac{1}{\mu_1 + \mu_2 + \ldots + \mu_k - K\lambda} \Bigg)^2, \tag{2.6}$$

$$D[W_K] \approx \frac{2}{(\mu - \lambda)^2} \sum_{i=1}^{K} \binom{K}{i} (-1)^{i-1} \frac{1}{i^2} - \left(\frac{1}{\mu - \lambda} \sum_{i=1}^{K} \frac{1}{i} \right)^2. \tag{2.7}$$

3. ARTIFICIAL NEURAL NETWORKS AND QUEUING SYSTEMS WITH REQUEST SPLITTING

Issues of using ANNs to study QS are currently poorly reflected in the world literature, although ANN methods can be effectively used to study several queues that are not solved in theory. In particular, this refers to the problem of forecasting or regression, which comes down to the problem of approximating the function of one or more variables. According to the theorems given in [3, 19], as well as in the survey [16], any continuous function can be represented as a combination of linear operations and a single nonlinear element. Thus, these theorems turn out to apply to the ANN, and, accordingly, to the multilayer perceptron, in which the so-called activation function acts as a nonlinear element. In the theory of queuing, there are many models, for example, non-Markovian models, for which classical analytical methods of solution are not applicable, and the numerical approach is rather laborious. In such conditions, addressing neural networks seems to be the best solution. Nevertheless, until recently, there have been very few works devoted to the application of ANNs to solving problems of queuing theory, namely, the analysis of complex queuing models. The articles [17,18] were devoted to building a model of the classic $M|M|1$ queuing system using an ANN and analyzing the adequacy of this simulation. The experiments showed that the values modeled using the ANN coincide with calculated values obtained using the classic mathematical approach. One of the first works (if not the very first), which affected the application of neural network apparatus for the analysis of non-Markovian QS models is the article [12], in which non-Markovian QS with heating [13] is researched since even in the case of QS $H_2|M|M|3$ or $M|H_2|M|3$ numerical algorithms for calculating stationary probabilities of states turn out to be very laborious and resource-intensive. The use of neural networks, in this case, made it possible to significantly reduce the complexity without losing accuracy in the calculations.

3.1. ANN model for fork-join

The multilayer perceptron will be used to approximate the average value and variance of the response time. Moreover, to increase the accuracy of the approximation, it is necessary to train not one perceptron with two output neurons corresponding to each of the indicated characteristics, but two perceptrons with one output neuron in each to estimate the mathematical expectation and standard deviation of the response time.

Despite the sufficiency of one hidden layer according to the above theorems on the approximation of functions, let us stay with the ANN with two hidden layers with the logistic activation function on each neuron $\varphi(x) = 1/(1 + e^{-x})$ (Fig. 3.2) from the estimate accuracy increase point of view. The load ρ and the K ($K = \overline{1, 20}$) number of sub-requests into which the request entering the system is split with a fixed value of $\lambda = 2$ (Table 3.1) will be used as input data.

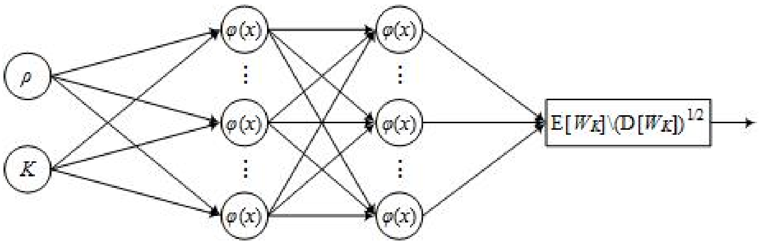

Fig. 3.2. Three-layer perceptron scheme

Table 3.1. Input data for calculating QS fork-join

ρ	0.1	0.2	...	0.9	0.1	0.2	...	0.9	0.1	...	0.9
K	3	3	...	3	4	4	...	4	5	...	20

The vectors of the values of the training and test samples were obtained using simulation in a specialized GPSS software environment (General Purpose Simulation System), and the construction of the neural network model itself and its training was carried out using the Python programming language. In the process of training ANNs with the backpropagation method, the best neural network configuration was selected based on the mean square error (MSE), mean absolute error (MAE), and average absolute relative error (MAPE):

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y}_i)^2, \quad MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \widehat{y}_i|,$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{(y_i - \widehat{y}_i)}{y_i}\right| \cdot 100\%,$$

where y_i is the desired (actual) output value for the i-th input element from the training set, $\widehat{y}_i$ is the real value produced by the system, N is the number of elements in the training set, $i = \overline{1, N}$.

The choice of the latter of the measures presented here is since it was used by the authors to assess the quality of the approximate formulas obtained by them. Since the best approximation for the average response time is given by formula (2.1), in particular, the error of this formula for $K \leqslant 32$ does not exceed 5%, the results of the ANN are compared precisely with the results of applying this formula. So, in Table 3.2, the values of the above types of errors calculated already on the test sample in the case of using simulation, neural network, and formula (2.1) for calculations are presented. It is the trained perceptron that shows the best result even despite the relatively small number of vectors in the training sample, which is measured in hundreds and probably does not improve the quality of approximation in a general case. Nevertheless, the obtained result speaks for itself and only confirms the potential of the proposed approach.

In the case of estimating the standard deviation of the response time, the well-known formulas show a large error of approximation, in contrast to estimating the average value of this quantity. Therefore, ANNs predictably give the best result (Table 3.3).

Table 3.2. Comparison of error values in calculating the average response time using formula (2.1) and ANN.

Error type	MSE	MAE	MAPE, %
Formula (2.1)	0,022164	0,081634	1,592066
Neural network	0,000649	0,015875	0,739039

Table 3.3. Comparison of error values in calculating the standard deviation of the response time using formula (2.7) and ANN.

Error type	MSE	MAE	MAPE, %
Formula (2.7)	0,080494	1,696545	6,896426
Neural network	0,000642	0,010495	0,363507

4. CONCLUSION

Due to applying to a new approach to estimating the random value of the response time of a cloud computing system, a more accurate approximation is obtained for the mathematical expectation and variance of this characteristic.

Application of neural networks to the analysis of indicators of service quality in the fork-join type QSs allows expanding the class of models under consideration. The reason is the absence of restrictions on assumptions about the type of incoming flow or distribution of service times, which, as a rule, greatly complicates the analytical or numerical process of solving the problem, while there is no loss in estimation quality of required parameters.

Also, it becomes possible to increase the number of analyzed parameters. So, for example, in the future, it is possible to clarify the well-known estimate of the average time spent by the first sub-request in the synchronization buffer in anticipation of the last of its related components [6]. That is, in fact, the average residence time of related sub-requests in the buffer until exiting the system. Information about this value is important, for example, for designing a cloud system when planning the size of the waiting buffer, because the longer this time, the larger the buffer should be.

ACKNOWLEDGEMENTS

The authors are grateful to A.V. Lebedev (Moscow State University) for valuable recommendations and useful comments.

REFERENCES

1. Alhamad, M., Dillon, T., Wu, Ch. & Chang, E. (2010) Response time for cloud computing providers, *Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services*, 603–606.
2. Buyya, R., Broberg, J. & Goscinski A.M. (2011) *Cloud computing: Principles and paradigms*. New Jersey, USA: John Wiley & Sons.
3. Gorban' A.N. (1998) Obobshchennaya approksimacionnaya teorema i vychislitel'nye vozmozhnosti nejronnyh setej [Generalized approximation theorem and computational capabilities of neural networks], *Sibirskij zhurnal vychislitel'noj matematiki [Siberian J. of Numer. Mathematics]*, **1**(1), 12–24, [in Russian]
4. Gorbunova, A.V. & Lebedev, A.V. (2020) Bivariate distributions of maximum remaining service times in fork-join infinite-server queues, *Problems of Information Transmission*, **56**(1), 73–90.
5. Gorbunova, A.V., Zaryadov, I.S., Matushenko, S.I. & Samuylov, K.E. (2015) Approksimaciya vremeni otklika sistemy oblachnyh vychislenij [The approximation of

response time of a cloud computing system] *Informatika i ee primeneniya [Informatics and Applications]*, **9**(3), 32–38, [in Russian].

6. Gorbunova, A.V., Zaryadov, I.S., Matushenko, S.I. & Sopin E.S. (2016) The estimation of probability characteristics of cloud computing systems with splitting of requests, *Proceedings of the Nineteenth International Scientific Conference Russia: Distributed computer and communication networks: control, computation, communications (DCCN-2016)*, **3**: Youth School-Seminar, 467–472.
7. Gorbunova, A.V., Zaryadov, I.S., Samuylov, K.E. & Sopin E.S. (2017) Obzor sistem parallel'noj obrabotki zayavok [A survey on queuing systems with parallel serving of customers], *Vestnik Rossijskogo universiteta druzhby narodov. Seriya: Matematika, informatika, fizika [RUDN Journal of Mathematics, Information Sciences and Physics]*, **25**(4), 350–362, [in Russian].
8. Gorbunova, A.V., Zaryadov, I.S. & Samuylov, K.E. (2018) Obzor sistem parallel'noj obrabotki zayavok. CHast' II [A survey on queuing systems with parallel serving of customers. Part II], *Vestnik Rossijskogo universiteta druzhby narodov. Seriya: Matematika, informatika, fizika [RUDN Journal of Mathematics, Information Sciences and Physics]*, **26**(1), 13–27, [in Russian].
9. Jitendra, S. (2014) Study of Response Time in Cloud Computing, *International Journal of Information Engineering and Electronic Business*, **6**, 36–43.
10. Keller, M. & Karl, H. (2017) Response-time-optimized service deployment: MILP formulations of piece-wise linear functions approximating bivariate mixed-integer functions, *IEEE Transactions on Network and Service Management*, **14**(1), 279–294.
11. Kemper, B. & Mandjes, M. (2012) Mean sojourn time in two-queue fork-join systems: Bounds and approximations // *OR Spectrum*, **34**, 723–742.
12. Khomonenko, A.D. & Yakovlev, E.L. (2015) Nejrosetevaya approksimaciya harakteristik mnogokanal'nyh nemarkovskih sistem massovogo obsluzhivaniya [Neural network approximation of characteristics of multi-channel non-Markovian queuing systems], *Trudy SPIIRAN [SPIIRAS Proceedings]*, **41**(4), 81–93, [in Russian].
13. Khomonenko, A.D., Adadurov, S.E. & Gindin, S.I. (2013) CHislennyj raschet mnogokanal'noj sistemy massovogo obsluzhivaniya s rekurrentnym vhodyashchim potokom i "razogrevom" [Numerical calculations of multichannel queuing system with recurrent input and "warm up"], *Izvestiya Peterburgskogo universiteta putey soobscheniya [Proceedings of Petersburg Transport University]*, **37**(4), 92–101, [in Russian].
14. Nelson, R. & Tantawi, A.N. (1988) Approximate analysis of fork/join synchronization in parallel queues // *IEEE Transactions on Computers*, **37**, 739–743.
15. Rashid, Z.N., Zebari, S.R.M., Sharif, K.H. & Jacksi, K. (2018) Distributed cloud computing and distributed parallel computing: a review, *International Conference on Advanced Science and Engineering (ICOASE) Proceedings*, Duhok, 167–172.
16. Shvedov, A.S. (2018) Approksimaciya funkcij s pomoshch'yu nejronnyh setej i nechetkih sistem [Functions approximating by neural networks and fuzzy systems], *Problemy upravleniya [Control Sciences]*, 1, 21–29, [in Russian].
17. Sivakami Sundaria, M. & Palaniammalb, S. (2015) Simulation of $M|M|1$ queuing system using ANN // *Malaya Journal of Matematik: Special Issue*, 1, 279–294.
18. Sivakami Sundaria, M. & Palaniammalb, S. (2015) An ANN simulation of single server with infinite capacity queuing system // *International Journal of Innovative Technology and Exploring Engineering*, **8**(12), 4067–4071.
19. Stone, M.N. (1948) The generalized Weierstrass approximation theorem, *Mathematics Magazine*, **21**(4), 167–183.
20. Thomasian, A. (2014) Analysis of fork/join and related queueing systems, *ACM Computing Surveys (CSUR)*, **47**(2), 17:1–17:71.
21. Varki, E., Merchant, A. & Chen, H. The M/M/1 fork-join queue with variable subtasks (unpublished).

22. Varma, S. & Makowski, A.M. (1994) Interpolation approximations for symmetric fork-join queues, *Performance Evaluation*, **20**, 245–265.
23. Xiong, K. & Perros, H. (2009) Service performance and analysis in cloud computing, *5th IEEE World Congress on Services (Services-1'09) Proceedings*, Los Angeles, 693–700.