Adaptive Decision Support System for Scaling University Cloud Applications

Bakhytzhan Akhmetov¹, Valerii Lakhno², Boris Gusev², Miroslav Lakhno², Ivan Porokhnia², Gulnaz Zhiikishbayeva³, and Madina Akhanova¹

¹ Abai Kazakh National Pedagogical University, Almaty, Kazakhstan
² National University of Life and Environmental Sciences of Ukraine, Kiev, Ukraine
³ Yessenov University, Aktau, Kazakhstan

Abstract. The article proposes algorithms for an adaptive decision support system (DSS) related to the scaling of cloud applications (ClAPP) for the digital educational environment (DEE) of the university, as well as for determining the time for executing a network request. Based on information about the state of the DEE and the algorithms proposed for the DSS in the article, there are formed sets of rules for reactive scaling and assessment of the effectiveness of the DEE of the university. The proposed algorithms for adaptive DSS allow making flexible decisions about scaling the cloud application of the DEE of the university. There was also developed a model of the criterion function for the analysis of scaling variants of the DEE of the university. This allows getting an economic assessment of the cloud application effectiveness. This assessment is based on the cost of maintaining the infrastructure of the cloud application, as well as on the assessment of the number of users who are forced to use ClAPP due to its overload. The developed algorithm can be used as a constituent of the information technology for ClAPP scaling not only for universities, but also for other objects informatization.

Keywords: Digital educational environment of the university · Cloud technologies · Algorithms · Assessment of the effectiveness · Decision making

1 Introduction

In the context of the global pandemic of the coronavirus Covid-19 and due to the growing importance of social isolation, the use of cloud services has increased many times, including in educational institutions all over the world.

The use of cloud technologies and computing has entered the everyday practice of leading universities [1, 2]. This made it possible to significantly facilitate the support of the server infrastructure, to increase the speed of application deployment, and also to adapt cloud applications (ClAPP) to their load conditions with interchangeable peaks. The uneven intensity of use of the cloud application(s) determines the relevance of tasks to optimize the computing resources allocated to support the operation of the cloud...
infrastructure of a university or other educational institution. This task is being achieved by scaling cloud applications. This procedure is implemented on the basis of assessing the state of the cloud of the University’s digital educational environment (DEE), which allows to flexibly adapt the number of allocated computing resources to the workload of CIAPP and services, for example, of a large university.

CIAPP scaling systems mainly use reactive scaling (Reac) [3]. That is, the increase or decrease in computing power at reaching a certain threshold of the use of available computing resources is not always planned. This approach is effective in general, but in case of short and frequent load peaks, such as at testing a large number of students, or during the peak load on distance education systems that are in the cloud, there are observed problems in the operation of CIAPP (for example, Moodle, Classroom etc.) in the interval between the beginning of the scaling process and the allocation of computing resources. To avoid this disadvantage, there is used predictive scaling, based on the intensity prediction of the CIAPP load. Such a prediction can be created using the methods of time series prediction with their application to the time series of network requests received in CIAPP. At the same time, information about the possible load peaks of CIAPP can be used to increase the accuracy of the prediction by appropriate methods, depending on the state in which CIAPP is located. Most modern systems perform only horizontal scaling based on reactive rules. The combination of horizontal and vertical scaling using both reactive rules and CIAPP state prediction will increase the number of possible server infrastructure states and the scaling efficiency of the university’s cloud infrastructure or its distance learning system.

Therefore, the task of developing new models and information technology for CIAPP scaling using models for predicting the state of CIAPP, in order to reduce the reaction time to changes in workload and to increase the efficiency of CIAPP operation of the DEE of the university is relevant.

2 Literature Review and Analysis

Nowadays, many issues on improvement of the efficiency of virtual cloud resources (VCR) have not been fully investigated [1, 2]. In existing developments [3, 4], solutions to this type of problem have significant disadvantages. In particular, a number of researchers [4–6] note the lack of a universal method for ensuring the functional stability of cloud resources CS due to: 1) constant changes in the capabilities of existing virtual machines (VMs) in cloud information systems, including university ones; 2) the use of insecure application interfaces; 3) high resource intensity of existing systems.

In works [7, 8], the authors analyzed the effectiveness of cloud technologies, on the basis of which it was concluded that their use provides a number of advantages over traditional IT technologies. In [8–10], there are shown the advantages of VCR in the tasks of efficient computing resources management of the organization. In works [9, 11], the advantages of VCR were substantiated in the tasks of reducing the cost on IT infrastructure and maintenance personnel, in the tasks of ensuring reliable data protection from losses and attacks. However, these works did not contain a description of specific techniques aimed at solving the problem of cloud applications scaling. In [10, 11], there were considered practical aspects of the use of VCS for the implementation of many
types of training activities, monitoring and assessment, and online testing. At the same time, the tasks related to the CS of VCS in educational institutions were not considered.

As an analysis of researches on this issue has shown, the most common approaches to automated CIAPP scaling are methods, models and technologies based on reactive scaling and scaling according to a schedule [1]. However, the disadvantage of these methods is the impossibility of scaling until the restart of the CIAPP of the university DEE and the possible inaccuracies in the schedule. Therefore, these approaches are ineffective under certain conditions of CIAPP operation of the university DEE.

3 The Purpose and Objectives of the Research

The purpose of the work is to develop models and algorithms for the computing core of the decision support system for cloud applications scaling of the university’s digital educational environment.

In order to achieve this goal, it is necessary to solve the following tasks:

1. To develop a model and appropriate algorithms for the computing core of the decision support system for cloud applications scaling of the university’s digital educational environment.
2. To develop a software application for assessing and predicting the need for proactive cloud applications scaling of the university’s digital educational environment.

4 Methods and Models

Problem statement. Let \( \langle S, N, P \rangle \) – the current state of the CIAPP infrastructure of the university DEE, where \( S \) – virtual machine (VM) size, \( N \) – number of allocated VMs; \( P \) – VM content for one minute. There are also set: \( X \) – prediction vector of the network requests number for the next \( h \) minutes; \( \tau_{sc} \) – time after the last CIAPP scaling; \( G \) – CIAPP infrastructure state graph of the university DEE; \( \tau_r \) – time period after CIAPP scaling during which new scaling is not carried out; \( C \) – information on the average time for executing a network request under various CIAPP operating conditions of the university DEE.

It is necessary to create an algorithm, which on the basis of the data on the CIAPP state of the university DEE returns a decision on CIAPP scaling \( e = \langle S', \Delta N \rangle \), where \( S' \) – the size of the VM after scaling; \( \Delta N \) – an increase in the number of VMs relative to the state of the CIAPP infrastructure of the university DEE at the beginning of scaling.

Let introduce the decision making algorithm for CIAPP scaling of the university DEE based on the use of reactive or proactive scaling methods (Proac). The reactive scaling method is a set of rules that are executed when there is an urgent need for scaling. Proactive scaling is carried out on the basis of the prediction of the CIAPP operation of the university DEE. However, the prediction may be inaccurate, then choosing the final solution, the priority is given to the results of the reactive scaling method.

The result of the algorithm is the decision on the need for scaling. In this case, the scaling can be vertical (the value \( N \) changes) or horizontal (the value \( S \) changes).
Let denote the results of CIAPP scaling of the university DEE by the pair \( e = (S', \Delta N) \). Positive values correspond to scaling up, negative values correspond to scaling down. The result of the algorithm may be a pair \( (S, 0) \) that corresponds to the absence of the need for scaling. We assume that such a result is equivalent to an empty scaling operation, i.e. \( (S, 0) = \text{null} \).

The work of the adaptive decision making algorithm for CIAPP scaling of the university DEE is based on the use of two algorithms representing the methods of reactive and proactive scaling. The input data of these algorithms are subsets of the input data of the main algorithm, and the result of their operation is the decision for scaling \( (S', \Delta N) \).

The algorithm consists of the following steps:

Step 1. Start of the work.
Step 2. Entering information about the CIAPP state of the university DEE.
Step 3. To assign the result of the reactive scaling algorithm to the current scaling decision \( e \).
Step 4. If \( e \) is empty, perform step 5, otherwise – step 6.
Step 5. To assign the result of the proactive scaling algorithm to the current scaling decision \( e \).
Step 6. To find the value of the current scaling decision \( e \).
Step 7. End.

The scheme of the algorithm is shown on Fig. 1.

A classic approach to decision making on CIAPP scaling of the university DEE is the use of the reactive scaling method. This method provides the determination of the limiting indicators of memory and processor utilization upon reaching of which the scaling occurs [3]. This scaling method is also called the static threshold method [4].

The core of the reactive scaling method is a set of rules \( r \in R \) of the form:

\[
\text{if } X_r|t > t_b, \quad \text{then}
\]

\[
\text{perform scaling } (f_r(S), \Delta N), t_{br} \leftarrow t + \Delta t,
\]

where \( t \) – current moment of time, \( t_{br} = \tau_{sc} + \Delta t \) – a point in time at which the stabilization period ends after the last scaling operation, \( \Delta t \) – a period of rest after scaling during which new CIAPP scaling operations of the university DEE are not performed. This period of time is necessary in order to begin to take indicators of the CIAPP load after changing the configuration of the university cloud infrastructure, \( f_r \) – a function determined on the basis of the current size of the VM (\( S \)). This function returns the necessary size of the VM (\( S', X_r \)) – for scaling, it can take a form:

\[
\text{processor load} > 80\% \quad \text{or a part of virtual memory} > 20\%.
\]

In most cases [3, 4], rules are created in pairs: for scaling up and down. At creating rules, it should be taken into account the cloud application hosting limitations: to make sure that the number and size of VMs are within the allowed range, do not use transitions that are not in the state graph of the CIAPP infrastructure of the university DEE, etc. All this leads to a complication of the conditions of the rules \( X_r \).
Carrying out reactive scaling, it is necessary to take into account the possibility of oscillations in the number of VMs. In order to avoid this undesirable effect, it should be carefully selected the limiting values of the scaling rules. In order the method not to issue scaling commands in response to unit load peaks, in the conditions of the rules $X_r$, there are used averaged indices for the last few time steps. For the same purpose, some varieties of the method issue a scaling command only if the condition $X_r$ is met sequentially in several steps.

In a set of rules $R$, we can add rules with conditions that are executed at a certain point in time, without reference to the level of system resources use. Such rules make it possible to scale on the eve of events that increase the CIAPP workload of the university DEE. These are, for example, certification periods and a session. To minimize costs, modifications to the rules can be made taking into account the hosting pricing policy. For example, in case of time-based billing, it makes no sense to scale CIAPP of the university DEE down within an hour after the up-scale operation.

The main advantages of using the reactive scaling method with respect to the CIAPP of the university DEE with periodic load peaks is the simplicity of the method, its high speed and the ability to start operation without collecting additional information about the infrastructure. The limitations of the method include the reactive nature, which is displayed in its name. The scaling command can only be given when a lack of resources is detected, and when new resources are allocated the operation of the cloud application

Fig. 1. Decision making algorithm for CIAPP scaling of the university DEE
can occur in an emergency mode. Also, a complex procedure is to set threshold values in the rules \((r)\), as well as the selection of a scaling action \(\langle S', \Delta N \rangle\). Let note that due to the large number of combinations the choice of optimal solution is a non-trivial task.

The ability to use the method without additional knowledge about the CIAPP operation of the university DEE in a specific infrastructure allows the use of the described method as a reserve for more complex scaling methods. If it is impossible to make a decision about scaling using your own resources, you can use the results of the reactive scaling method.

In case if CIAPP of the university DEE has a state \(\langle S_i, N_i, P_i \rangle\) from the state graph of the cloud application infrastructure, there are possible \(d(i)\) scaling variants, where \(d\) – a function that returns the degree of the graph vertex. Let denote the set of possible scaling variants as \(E_i\). In order to select the optimal scaling variant, we calculate the value of the criterion function \(G(e), e \in E_i\) as follows:

\[
G(e) = P_{inf} + k \cdot P_l,
\]

where \(P_{inf}\) – the cost of maintaining the CIAPP infrastructure of the university DEE within the next \(t_r\) minutes, \(P_l\) – the cost of losing one user, \(k\) – the number of users lost due to the excessive load of the cloud application during the following \(t_r\), \(e\) – the arc of the state graph of the CIAPP infrastructure of the university DEE - \(\langle S_i, N_i, P_i \rangle, \langle S_j, N_j, P_j \rangle, t_{ij}\)\.

The cost of maintaining the CIAPP infrastructure of the university DEE is determined as the sum of the cost of maintaining up to and after scaling. Each of the terms is calculated as the value of the tariff per minute \(P\) on the number of minutes \(t_r - t_{ij}\) and \(t_{ij}\):

\[
P_{inf} = P_j \cdot t_{ij} + P_j \cdot (t_r - t_{ij}).
\]

The number of users lost due to the excessive CIAPP workload of the university DEE is calculated as follows:

\[
k = \sum_{t=1}^{t_r} v_t \cdot q_l(t),
\]

where \(v_t\) – the number of CIAPP users of the university DEE over the period of time \((t; t + 1)\), \(q_l(t)\) – the probability that the user will stop using CIAPP of the university DEE. Substituting expressions (2) and (3) in (1) we obtain a more detailed expression for calculating the criterion function:

\[
G(e) = P_j \cdot t_{ij} + P_j \cdot (t_r - t_{ij}) + P_l \cdot \sum_{t=1}^{t_r} v \cdot q_l(t).
\]

To determine the probability of user loss \(q_l\), we use the probability distribution function, which represents the dependence of the probability of user loss on the time it takes to complete a network request \(F(\tau)\). The probability of user loss increases with the time it takes to wait for a response from the CIAPP of the university DEE. Studies [4–6] show that the number of users begins to decrease when \(\tau > 2\) s. In [5] there is shown the distribution function of the probability of user loss.
Let denote the random variable corresponding to the time of executing one network request of CIAPP of the university DEE as \( R \). Statistical characteristics of \( R \) depend on the CIAPP workload at a particular moment of time \( t \) and on the state of the CIAPP infrastructure. Let determine the probability of user loss through the mathematical expectation of the network query execution time \( M(R) \):

\[
q_l(t) = F(M(R_j, \tau)).
\]  

(5)

In case if \( t_{ij} \leq h \) and taking into account (5), expression (3) can be represented as follows:

\[
k = \sum_{t=1}^{t_{ij}} v_t \cdot F(M(R_{j, \tau})) + \sum_{t=t_{ij}+1}^{h} v_t \cdot F(M(R_{j, \tau})) + \sum_{t=h+1}^{t_r} v_t \cdot F(M(R_{j, \tau})).
\]  

(6)

Considering that the prediction of the number of network requests is made in \( h \) steps, it is necessary to determine the values \( v_t \) and \( R \) for the time interval \([h; t_r)\). We use the prediction at the moment of time \( h \) for this time period, then the last term in (6) can be represented by:

\[
k = \sum_{t=1}^{t_{ij}} v_t \cdot F(M(R_{j, \tau})) + \sum_{t=t_{ij}+1}^{h} v_t \cdot F(M(R_{j, \tau})) + (t_r - h) \cdot v_h \cdot F(M(R_{j, \tau})).
\]  

(7)

Therefore, expression (4) can be detailed using relation (7):

\[
G(e)\mid t_{ij} \leq h = P_j \cdot t_{ij} + P_j \cdot (t_r - t_{ij})
+ P_l \cdot \left( \sum_{t=1}^{t_{ij}} v_t \cdot F(M(R_{j, \tau}))
+ \sum_{t=t_{ij}+1}^{h} v_t \cdot F(M(R_{j, \tau})) + (t_r - h) \cdot v_h \cdot F(M(R_{j, \tau})) \right).
\]  

(8)

For the case when \( t_{ij} > h \) similar transformations lead to such results:

\[
k = \sum_{t=1}^{h} v_t \cdot F(M(R_{j, \tau})) + (t_{ij} - h) \cdot v_h \cdot F(M(R_{j, \tau})) + (t_r - t_{ij}) \cdot v_h \cdot F(M(R_{j, \tau})).
\]  

(9)

\[
G(e)\mid t_{ij} > h = P_j \cdot t_{ij} + P_j \cdot (t_r - t_{ij})
+ P_l \cdot \left( \sum_{t=1}^{h} v_t \cdot F(M(R_{j, \tau}))
+ (t_{ij} - h) \cdot v_h \cdot F(M(R_{j,h})) + (t_r - t_{ij}) \cdot v_h \cdot F(M(R_{j,h})) \right).
\]  

(10)

Therefore, on the basis of the prediction of the number of CIAPP users of the university DEE and the average execution time of the network request, it is possible to calculate the value of the criterion function for all possible ways to scale \( e \in E_i \) of CIAPP at a
time \( t \). The scaling method with the lowest value of the criterion function is used as the selected scaling variant.

The formal record of the scaling rule in DSS has the following form:

\[
\text{if } t > t_b \text{ then }
\]

\[
\text{perform scaling } e_{\min} | G(e_{\min}) = \min G(e),
\]

\[
\text{if } e_{\min}, S' \neq S_i \text{ or } e_{\min}, \Delta N \neq 0 \text{ then } t_b \leftarrow t + \Delta t
\]

where \( t \) – current moment of time, \( t_b \) – point in time at which the stabilization period ends after the last scaling operation, \( \Delta t \) – the period of rest time after ClAPP scaling of the university DEE during which new scaling operations are not carried out.

In order to determine the value of the criterion function of the ClAPP scaling efficiency of the university DEE, it is necessary to have information about the number of ClAPP users per minute and about the average time of the network request. Using the DSS computational core, we will create a prediction of these values based on the prediction of the number of network requests and statistical information on the ClAPP operation of the university DEE [11, 12].

The ratio of the number of network requests to the number of unique users with a large number of users is a constant value for a particular ClAPP. Therefore, the prediction of the number of users \( v_t \) can be calculated as follows [13]:

\[
v_t = x_t \cdot \psi_u,
\]

where \( x_t \) – prediction of the number of network requests for a period of time \( t \), \( \psi_u \) – coefficient showing the frequency of requests from one user per unit of time for a specific ClAPP. The value of the coefficient \( \psi_u \) is determined by periodical collecting statistics on the number of unique users and the number of network requests during ClAPP operation of the university DEE.

In order to predict the execution time of a network request based on the number of network requests per unit of time, it is necessary to have information about the allocated computing power. Such information is contained in the description of ClAPP infrastructure of the university DEE \( \langle S, N, P \rangle \), namely, the size of the VM \( S \) and their number \( N \). Let present possible combinations of variable values \( x, S \) and \( N \) using a hypercube of cloud application states [9, 10, 14]. The number of measurements of such a hypercube is three; an array is stored in the cells, each element of which represents the execution time of a separate network request. The cell also stores the average value of the array elements. Such a representation will speed up the calculations. To reduce the number of hypercube cells and to facilitate prediction [9, 14, 15], it is necessary to discretize the variable \( x \), dividing the range of its values to 3–10 segments.

The advantage of using a hypercube for storing data on the average time of a network request is the possibility of organizing data in such a way when it becomes possible to use information from neighboring cells in case of the absence of information in the cell describing the ClAPP state.
In case if the increase in the execution time of the network request changes linearly at moving along one of the dimensions of the hypercube, it is possible to introduce into DSS an algorithm for determining the execution time of the network request. This algorithm allows to supplement the information about empty hypercube cells using linear interpolation.

Step 1. Start of the work.
Step 2. Entering information about the dimension of the hypercube of ClAPP states \((k_S, k_N, k_x)\), the set of values of its cells \((C)\) and the target state \((iS, iN, ix)\), as well as about the search depth \(z\).
Step 3. If \((C_{iS,iN,ix})\) has the value – go to the step 4, otherwise – step 5.
Step 4. Find \((C_{iS,iN,ix})\), go to step 13.
Step 5. For \(j\) from 1 to \(z\) perform the steps 6 – 11.
Step 6. If there is \((C_{iS-j,iN,ix})\) and has the value and if there \((C_{iS+j,iN,ix})\) and has the value – go to the step 7, otherwise – go to the step 8.
Step 7. Find \([((C_{iS-j,iN,ix} + C_{iS+j,iN,ix})/2]\), go to step 13.
Step 8. If there is \((C_{iS,iN-j,ix})\) and has the value and if there \((C_{iS,iN+j,ix})\) and has the value – go to the step 9, otherwise – go to step 10.
Step 9. Find \([((C_{iS,iN-j,ix} + C_{iS,iN+j,ix})/2]\), go to step 13.
Step 10. If there is \((C_{iS,iN,ix-j})\) and has the value and if there \(d\) has the value – go to the step 11, otherwise – go to step 5.
Step 11. Find \([((C_{iS,iN,ix-j} + C_{iS,iN,ix+j})/2]\), go to step 13.
Step 12. Find the undefined result.
Step 13. End.

A diagram of the algorithm for determining the execution time of a network request is shown on Fig. 2. By introducing a variable \(z\), the algorithm allows prediction based on hypercube cells, the values of which are undefined.

For clarity of the operation of the algorithm, the main variables are written in the corresponding blocks of the block diagram, respectively, on Figs. 1 and 2.

Algorithms on Fig. 1 and Fig. 2 allow forming a classification criterion for the current operating mode of ClAPP. The procedure for forming the criterion is based on the Paige-Hinckley method and the calendar of events, which are associated with the operation of ClAPP. The ClAPP performance criterion was determined on the basis of information about the execution time of a network request, the algorithm on Fig. 2, the number of users and the cost of maintaining the infrastructure. The proposed criteria made it possible to compare and combine dissimilar metrics of ClAPP operation of the universities. Using the obtained criteria, we can evaluate the efficiency of ClAPP functioning when using different scaling strategies, as well as compare the feasibility of using different scaling methods.
Fig. 2. Algorithm for determining the execution time of a network request in ClAPP of the university DEE

The model and decision making algorithms described in the article for CIAPP scaling of the university DEE, as well as determining the network request execution time were implemented in a software product – a decision support system on the need to scale CIAPP of the university DEE. The DSS interface is shown on Fig. 3.

In the computational core of adaptive DSS, a criterion function is used to analyze the variants for CIAPP scaling of the university DEE. This allows to get an economic assessment of the CIAPP effectiveness, which is based on the cost of maintaining the CIAPP infrastructure. Moodle systems, Blue Jeans Web conferences, the http://ideone.com portal and etc. were analyzed as cloud applications for learning programming languages. At the same time, the work was organized in heterogeneous collaboration groups. Students also had the opportunity to collaborate on Google Docs in order to create group presentations and reports on thematic projects. During the quarantine period, and the increase in the load on cloud services and applications of distance learning systems of the universities, there was made a comparison of the work offered by DSS for CIAPP scaling (based on the European University (Ukraine), Yessenov University and Abai University (Kazakhstan)) with a CloudMonix automation system.
Since the purpose of the work was to design the DSS computational core for scaling cloud applications of the university’s digital educational environment (DEE CA of the university), during the experiments, it was only required to qualitatively show the convergence and correctness of the algorithms shown on Fig. 1 and 2. In order to evaluate the algorithms in the DSS, it was proposed to implement the following approach: 1. For test CAs, the input data, the DEE CA configuration and the corresponding operating time were randomly selected. Test CAs were launched with a different number of nodes in the university’s DEE. Then, in the cycle, the input parameters of the problems, solved using the CAs, were randomly selected. These parameters, together with the history of previous launches, were fed to the input of the algorithms, Fig. 1 and 2. After each such request, the algorithms returned the DEE configuration and the time parameters for the execution of network requests to the DEE CA of the university. These values were saved in the history of DSS launches and were used for subsequent runs of the algorithm and decision making on scaling the university’s DEE cloud application.

It is shown that the developed DSS for CIAPP scaling can increase the efficiency of the cloud application by 12–15%.

5 Conclusions

Therefore, on the basis of information about the state of the cloud application of the university’s digital educational environment (DEE) there was formed an algorithm based on a set of reactive scaling rules and effectiveness assessment. The proposed algorithm allows making decisions on cloud application scaling of the university DEE. There was developed a model of the criterion function for the analysis of scaling variants, which allows to obtain an economic assessment of efficiency. This assessment is based on the cost of maintaining the cloud application infrastructure, as well as an assessment of the number of users who are forced to use the cloud application due to its overload. The
developed algorithm can be used as a composite information technology for an adaptive decision support system for cloud applications scaling not only for universities, but also for other informatization objects.

Therefore, the scientific novelty of the work lies in the improvement of the decision making method for CIAPP scaling. The improved method, in contrast to the existing ones, is based on a combination of reactive and advanced predictive approaches, taking into account data on possible peaks of CIAPP load. The proposed approach increases the capacity and efficiency of many CIAPP scaling solutions.

For the first time, there was developed an adaptive decision support system for CIAPP scaling. This system provides analysis of information about the CIAPP state and implements an improved method for decision making on its scaling, which can improve the efficiency of the CIAPP operation.

References

1. Yang, S., Huang, Y.: Teaching application of computer multimedia cloud sharing technology in hand-painted performance course in colleges and universities. In: International Conference on Education Innovation and Social Science, ICEISS. Atlantis Press (2017)
2. Liu, S., Zeng, W., Li, Y.: Study on the preliminary construction of the cloud of mental health education in Chinese colleges and universities. In: 2016 International Conference on Public Management. Atlantis Press (2016)
3. Lorido-Botran, T.: Auto-scaling techniques for elastic applications in cloud environments. In: Lorido-Botran, T., Miguel-Alonso, J., Lozano, J.A. (eds.) Department of Computer Architecture and Technology, University of Basque Country, Tech. Rep., EHU-KAT-IK-09, vol. 12 (2012)
4. RightScale Cloud Management. http://www.rightscale.com/
5. How a Slow Website Impacts Your Visitors and Sales. http://www.peer1.com/knowledgebase/how-slow-website-impacts-your-visitors-and-sales
6. Nah, F.F.H.: A study on tolerable waiting time: how long are web users willing to wait? Behav. Inf. Technol. 23(3), 153–163 (2004)
7. How Loading Time Affects Your Bottom Line. https://blog.kissmetrics.com/loading-time/
8. Menasce, D.: Load testing of web sites. Internet Comput. 6(4), 70–74 (2002)
9. Tamimi, A.A., Dawood, R., Sadaqa, L.: Disaster recovery techniques in cloud computing. In: 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology, JEEIT, pp. 845–850. IEEE (2019)
10. Duncan, B., Happe, A., Bratterud, A., Sen, J.: Cloud cyber security: finding an effective approach with unikernels. Security in Computing and Communications, vol. 31. IntechOpen, London (2017)
11. Akinsanya, O.O., Papadaki, M., Sun, L.: Current cybersecurity maturity models: how effective in healthcare cloud? In: CERC, pp. 211–222 (2019)
12. Srinivasan, S., Raja, K.: An advanced dynamic authentic security method for cloud computing. In: Cyber Security, pp. 143–152. Springer, Singapore (2018)
13. MacLennan, J., Tang, Z., Crivat, B.: Data mining with Microsoft SQL server 2008 (2008)
14. Savchuk, T.O., Kozachuk, A.V.: Vyznachennya dotsIlnostI vikoristannya bagatovimIrnogo pldhodu do prognozuvannya stanu tehnogenno. VimIryuvvalna ta obchislyuvalna tehnIka v tehnologIchnih protsesah. Hmelnitskiy 2(47), 179–182 (2014)
15. Lakhno, V.A., Malyukov, V.P., Satzhanov, B., Tabylov, A., Osypova, T.Yu., Matus, Yu.V.: The model to finance the cyber security of the port information system. Int. J. Adv. Trends Comput. Sci. Eng. 9(3), 2575–2581 (2020)