Increasing the Speed and Performance of the Drupal CMS Server for Industrial IoT Technologies

Viktor Satsyk[1[0000-0002-7132-3363]], Dagmar Cagáňová[2[0000-0002-6834-6126]], Oleksandr Reshetylo[1[0000-0003-2555-4205]], Oleg Zabolotnyi[1[0000-0002-9169-9173]], Anatolii Tkachuk[1[0000-0001-9085-7777]]

[1] Lutsk National Technical University, 75, Lvivska street, 43018 Lutsk, Ukraine
[2] Comenius University Bratislava, Faculty of Management, Slovakia
volynasi@gmail.com

Abstract. In this paper, the methods of increasing the speed and performance of the Drupal CMS-based server in IIoT technologies were analyzed. Three main load parameters were taken into account: processor time (CPU), level of server RAM usage, and disk memory usage level; on three pages of the WEB site (main, list of blogs, and a page with one blog) and two research methods were applied: generalized and detailed. Based on the results of the analysis of four options for optimizing Drupal CMS for IIoT technologies: basic, using additional installed modules, servers, and configurations, and Drupal Hooks, effective, simple, cost-effective methods were synthesized that guarantee an increase in server speed and productivity. It was found that the optimization of configurations and Drupal Hooks is the most effective since after its implementation, the response of a page with one blog increased by 9.32 requests per second; built-in optimization is extremely simple to implement and inefficient, while server-side optimization is inefficient, most complex and expensive to implement.

Keywords: CMS Drupal, Productivity, Optimization, Modules, Server Load, Drupal Hooks, IIoT Technologies.

1 Introduction

The international company IDC conducted a global study of the data market for 2020 and made a forecast for the next five years. According to these forecasts, global Internet traffic will increase annually by 23%. The fastest-growing data segment is the Internet of Things. IDC specialists also note the growing popularity of clouds for storing and using data. The average annual growth rate of storage volume worldwide is forecast at 19.2% [1].

A cloud server becomes an effective tool for collecting and storing a large amount of information. Cloud servers allow you to store and process large amounts of data without significant investments in infrastructure, to have access to them from anywhere in the world, and not to worry about their storage [2]. This is especially relevant during the implementationof the Industrial IoT (IIoT) paradigm of Industrial 4.0.

WEB sites create a high load on the server. By reducing the load on the server created by the site - costs for hosting, VDS, and renting a dedicated server - are minimized accordingly [3]. If the site's goal is a fast user experience, like Google, then you should focus on simplicity and provide the resources to deliver pages in 1.5 seconds or less. If the goal is to have a visually appealing site, then a rich theme with lots of interactive JavaScript and longer loading times may be acceptable [4].

For non-standard, time-consuming, highly loaded services, you should use Drupal. According to the resource [5], 54% of the sites that were created on CMS Drupal got into the "Top Best Sites" rating. CMS Drupal site development is the optimal solution as it is a free open-source system, with high power, security, and reliability. The main advantages of Drupal include ease of installation, simple engine management through administration, multi-functionality, and expansion of capabilities due to the installation of additional modules from the official website [6]. In addition, CMS Drupal, widely used in the world, has full Ukrainian localization of the system core of all current versions [7].

2 Literature review

The practical and effective methods of reducing the load on the server of functioning and newly created sites based on CMS Drupal discloses paper [8]. For already functioning sites, the following ways of reducing the load on the server are recommended: application of simple methods (neutralization), use of built-in and additionally installed modules, configuration and maintenance of Drupal, and use of various server configuration functions.

In order to increase the speed and performance of the server of newly created sites based on the CMS Drupal, four mandatory steps must be taken:

1) apply the built-in optimization of Drupal based on caching;

2) use the possibility of additionally installed modules (installation of modules: Authcach; Cache Router, CSS Gzip and JavaScript Aggregator);

3) configure Drupal configurations and maintenance (reducing user session storage time, reducing the number of site logging messages;

4) use server configuration functions (in each case, server optimization settings will be different, as eAccelerator installation in Windows and Linux are significantly different).

The step-by-step theoretical development and practical implementation of the algorithm of the data transformation system from local devices to the WEB interface based on CMS Drupal are disclosed [9], and the authors [10] describe the operation of the system in more detail. The technical capabilities offered by CMS Drupal and its support are described in detail in [11]. Special Drupal modules for artificial intelligence and their capabilities are presented on the site [12]. The analysis of the most common methods of load balancing for Web servers, which increase their speed and performance and ensure the stability of their work in IIoT, is carried out [13]. The existing methods of balancing the load of cloud systems have limited use and at the moment there is no universal system for balancing the load of networks in IIoT tech-

nologies [14]. Methods for increasing the performance, and speed of servers and ensuring data security are considered in [15]. It is recommended to apply quite practical steps to increase the productivity of high-load projects using CMS Drupal [16].

3 Researches Methodology

A web server is a server that accepts HTTP requests from clients, usually web browsers, and issues HTTP responses to them, usually along with an HTML page, image, file, media stream, or other data. The software that performs the functions of the Web server and the hardware on which this software product works are called Web servers [17]. In our research, we will use the most common web server, Apache.

The HTTP protocol in the IIoT concept is implemented using client-server technology and works on the request-response principle without saving the state [18]. The purpose of the request is one or another resource, which is determined by a single resource identifier - URI (Uniform Resource Identifier), HTTP uses one of the varieties of URI - URL (Uniform Resource Locator) - a universal pointer to the resource, which, in addition to information about the resource, determines its physical location, Fig. 1.

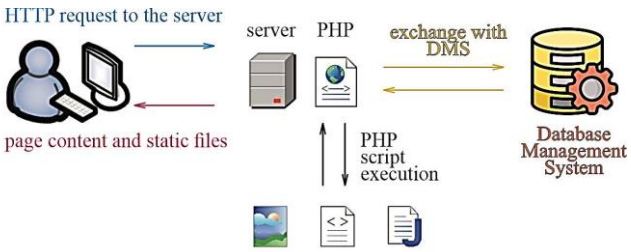


Fig. 1. Data exchange between the client and the server using the HTTP protocol in the IIoT concept.

Web server load parameters: Processor – processor time (CPU), or the time of information processing by the processor (in our case: 6; 3.0GHz); RAM (memory) – the level of RAM usage by the server (in our case: 8GB); Hard disk (disk) – the level of disk memory usage (in our case: 500Gb; 7200rpm; 64MB); Network interface (in our case: 10/100/1000).

The disk subsystem can be loaded when the application is working with files. In addition, the disks can be loaded by database work. Disk load is measured in Mb/s. However, for this work, we will translate these parameters into percentages. There are two types of load on the disk system: Actual DISK READ (the load that occurs when reading) and Actual DISK WRITE (the bag that occurs when data is written). Despite the fact that Actual DISK READ and Actual DISK WRITE create different functional loads in different time periods, and as practice shows, their value is small, in the study we will use the Actual DISK READ parameter. Each time a user accesses the site, the server receives information that it needs to process - perform queries to the database

and provide the user with one of the site's pages as a response. The amount of such information that the server is able to process depends on the RAM on the hosting server. This parameter is measured in Mb. For convenience in our work, we will translate the RAM load into %. Knowing that the server has 8GB of RAM, let's take this parameter as 100%.

The network interface is the amount of information that the server can process in one second and this value is measured in Mb/s. High network traffic is not a problem in itself. But close-to-peak indicators indicate the need to scale the server in the future. For example, an average traffic of 95 Mb/s on a 100 Mb/s interface will mean that the current server will soon be insufficient. However, in this work, we will not take this parameter into account, because this site does not access third-party resources on the back-end side, and therefore the load on the network interface does not exceed 3%.

Methods of server load analysis:

- generalized – using the online tool PageSpeed Insights,
- detailed – using access to the server via SSH (which provides the prediction of the occurrence of possible problems in certain functions). SSH is a protocol that uses a client-server model to authenticate remote systems and provide encryption for data exchanged over remote access.

Three pages are selected for server load analysis:

- main page,
- blog list page,
- one blog page.

We will use a scale from 0 to 100 points to evaluate the speed of the site. Where zero means high loads, long page loading, and 100 means the opposite. This evaluation option uses PageSpeed Insights statistics. A single-threaded ApacheBench command-line computer program executed 1000HTTPGET requests with 100 concurrent requests completed in 204.6 seconds; the server responds to 4.89 requests in one second (Requests per second, which means that the server processed 0.489% of requests in one second Fig. 2

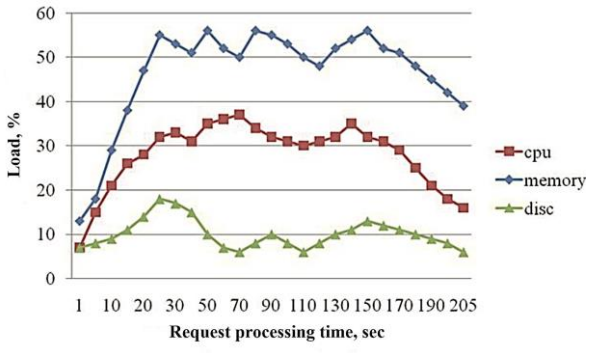


Fig. 2. Graphic representation of the percentage load of the main page of the site without optimization on the server depending on the processing time of 1000 HTTPGETs for 100 parallel requests.

Analyzing this graph, we can say that the distribution of the load on the server of the main page without optimization according to three parameters is not carried out evenly. The smallest (6…18%) load on the server by the main page is caused by the disk system of the server (disk). The maximum (13…56%) load on the server by the main page is observed in the used memory resource. The average (10…37%) load on the server by the main page is displayed during CPU usage. It should be noted that the speed of information processing by the server depends on its percentage load.

The blog list page was tested for the load. For this, the command was used: ab -n 1000-c 100http://

On this page, 1000 HTTPGET requests will be executed in 181.3 seconds. The server responds to 5.51 requests in one second. This means that the server processed 0.51% of requests in one second. The data collected after monitoring the results of the "htop" and "iotop" commands are shown in Fig. 3.

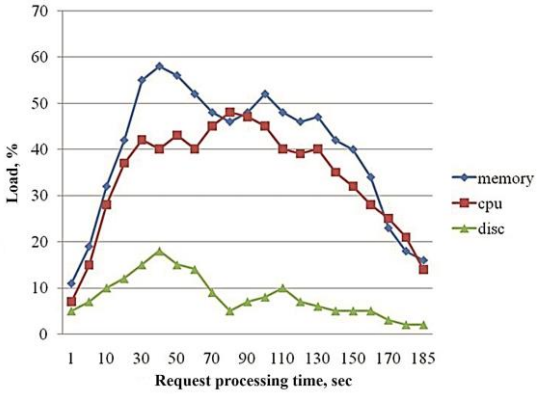


Fig. 3. Graphic representation of the percentage load of the blog list page of the site without optimization per server depending on the processing time of 1000 HTTPGET requests for 100 parallel requests.

In order to check the load of one blog page in the console, execute the following command: ab -n 1000-c 100http://. After running this command, the load on CPU will be from 88 to 92%, and on mysql from 32 to 78%. However, after some time, the error "Benchmarking (be patient) apr_poll: The timeout specified has expired (70007)" will be displayed in the console. The process ended without processing even 100 HTTPGET requests. This means that the page is too heavy and needs immediate optimization.

Therefore, at the moment, we will reduce the number of requests to 100 and parallel requests to 1. The command in the terminal will look like this: ab -n 100-c 1http://. The server processed these requests in 1662.7 seconds, the server answered 0.06 requests in one second. This means that the server processed 0.06% of requests in one second. The data collected after monitoring the results of the "htop" and "iotop" commands are shown in Fig. 4.

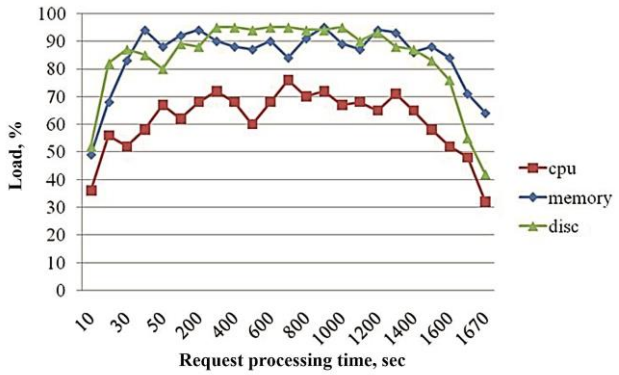


Fig. 4. Graphic representation of the percentage load of one blog site page without optimization per server depending on the processing time of 100 HTTPGET requests for 1 parallel request.

Analyzing the graph (Fig. 4), we can say that the distribution of the load on the server page of one blog without optimization according to three parameters is not carried out evenly. The least (10...75%) load on the server of one blog page is caused by the processor (CPU). The maximum (50...97%) load on the server of one blog page is observed when using the RAM resource (49...94%) (memory) and when using the disk system (52...95%) (disk). A generalized test of the load on the server by the statistical tool PageSpeed Insights (simplified) showed that the minimum load on the server without optimization is provided by the blog list page - 53 points. At that time, referring to the page of one blog carries the maximum load on the server - 28 points). The average between them is the main page - 47 points. This load on the server can be explained based on the structure of each element of the site.

Checking the load on the server using access to the server via the SSH protocol (detailed) showed the regularity of the load distribution. The minimum load on the server without optimization creates the blog list page (2...18%) and the main page (6...18%) using minimal server disk system resources. List of blogs: The maximum load on the server is created by using each of the pages of the site (one blog - 10...75%; list of blogs - 11...58%; the main one - 13...56%) using maximum memory resources; The page of one blog creates the maximum load both in the use of the RAM resource (memory) (49...94%) and when using the disk system (disk) (52...95%).

Research options:

Option 1. Built-in (basic) Drupal optimization

(Enabling caching, minifying and combining CSS and JS files, changing the quality of loaded images)

Option 2. Optimization using additionally installed Drupal modules

(Installing: Advanced CSS/JS Aggregation module to collect and optimize CSS and JS site files; page caching module – Boost; Clean block markup and Clean panels markup to disable some HTML5 elements; ImageMagick module to reduce image quality; execution of the “drush” command image-flush – all", which will delete old images and generate new ones).

Option 3. Server optimization.

(Installing PHP eAccelerator, enabling MySQL cache and size 64 megabytes)

Option 4. Optimization of configurations and Drupal Hooks

(Database normalization; optimization of Drupal configurations; optimization of Drupal hooks).

4 Results

4.1 The generalized method of analyzing the load on the server

Option 1. Built-in (basic) Drupal optimization

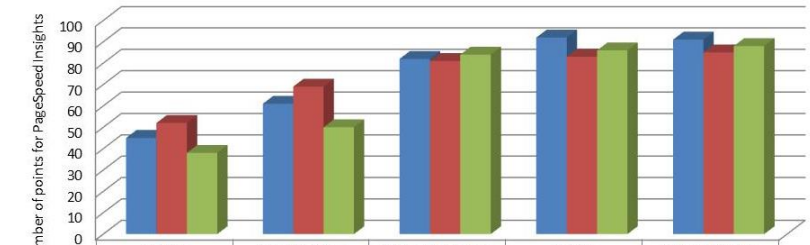
According to the PageSpeed Insights tool, after performing basic Drupal optimization, the main page, blog list page, and single blog scores increase by 16, 16, and 23 points, respectively. After this optimization, the site has not become more user-friendly. Pages still take a long time to load.

Option 2. Optimization with the help of additionally installed Drupal modules

However, after optimizing Drupal with the help of modules, these pages received an additional 20, 14, and 35 points, respectively. This optimization method can be classified as one of the most effective. After all, all pages received the "Good" category according to the scale of the PageSpeed Insights tool. All pages of the site load quickly except the pages with one blog. It should be noted that the results vary depending on the installed modules.

Option 3. Server optimization

The next step was to configure the server. As mentioned above, this method may not be available to most Drupal site owners. After all, hosting does not provide access to change server configurations. In this work, this service was provided by the owner of the server on which our site is located. After that, PageSpeed Insights showed the largest increase in scores for the main page at 10. However, the blog list page and the single blog page gained 2 and 5 points respectively. The data from these studies are shown in Fig. 5.



	Before optimization	Built-in Drupal optimization	Optimizing Drupal with modules	Server optimization	Optimization of configurations and Drupal Hooks
Main page	45	61	82	92	91
List of blogs	52	69	81	83	85
One blog	38	50	84	86	88

Fig. 5. Dynamics of changes in server load from step-by-step optimization (data from the PageSpeed Insights tool).

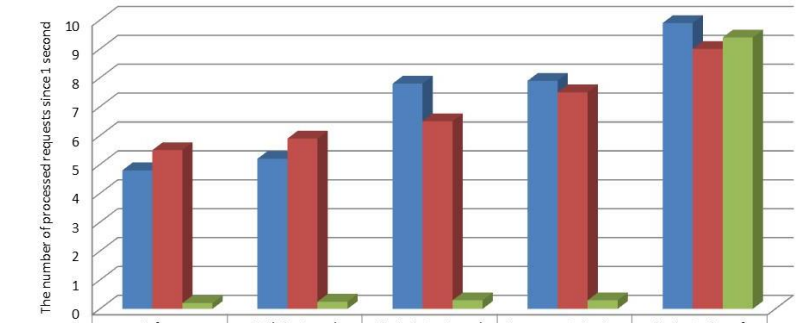
Option 4. Optimization of configurations and Drupal Hooks

After optimizing configurations and Drupal hooks, PageSpeed Insights showed no changes on any page. However, all pages have reduced load times in the browser.

4.2 A detailed method of analyzing the load on the server

Using access to the server via SSH, namely the load that the site creates on the processor (CPU), RAM (memory), and disk system (disk), using the "ab", "htop" and "iotop" utilities.

The results we got are much more accurate than the data from the PageSpeed Insights tool. After all, we can visually see all the problems that arise on the server. Let's depict the load on the server in the form of a correlation graph of the ratio of the number of requests that the server can process in one second to the optimization steps, Fig. 6.



	Before optimization	Built-in Drupal optimization	Optimizing Drupal with modules	Server optimization	Optimization of configurations and Drupal Hooks
Main page	4,8	5,2	7,8	7,9	9,9
List of blogs	5,5	5,9	6,5	7,5	9
One blog	0,2	0,25	0,3	0,3	9,4

Fig. 6. The ratio of the number of requests that the server can process in one second to 4 steps of optimization (1 - built-in Drupal optimization, 2 - optimization of Drupal using modules, 3 - optimization of the server, 4 - optimization of configurations and Drupal Hooks).

Option 1. Built-in (basic) Drupal optimization

Built-in optimization is extremely simple to implement and inefficient. After all, the maximum number of requests that the server can process on the main page and on the blog list page increased by 0.52. And a page with one blog added 0.019 requests in one second.

Option 2. Optimization using additionally installed Drupal modules

Optimizing Drupal using additional customizable modules is one of the most effective ways to reduce server load. After carrying out this optimization, the server increased the number of requests it can process in one second on the main page by 2.46 requests, on the blog list page by 0.77 requests, and on the page of one blog by 0.082. This method requires more time to implement, and it cannot be classified as a simple

way to reduce the load on the server. This method will be more effective for pages with many non-optimized images, CSS, and JS files. However, it is mandatory for all Internet sites.

Option 3. Server optimization

Optimizing the server is the most difficult way to reduce the load because most Drupal sites are hosted on hosts where this setting cannot be performed. You can solve this problem by renting a server. However, it is too expensive. Therefore, before launching the site, you should carefully choose the hosting and the space on the server that they provide. The server was optimized for us by the owner himself, after contacting him. This method made the server response faster for the main page by 0.16 requests per second, the blog list page by 0.91 requests per second, and the single blog page by 0.01 requests per second.

Option 4. Optimization of configurations and Drupal Hooks

Optimizing configurations and Drupal Hooks is also not an easy and slow way. After all, before that, you need to analyze the database architecture and site logic. After the actions are taken, the server increased the response to the main page of the site by 1.77 requests per second. The response to the blog list page increased by 1.4 queries per second. But the page with one blog received the biggest increase, with 9.32 requests per second. Table 1 presents numerical indicators of the effectiveness of the applied methods in increasing the speed and performance of the server.

Table 1. Performance indicators of the applied methods of increasing the speed and performance of the server

№	Site pages	Optimization methods (number of processed requests in 1 second)			
		Built-in (basic) Drupal optimization	Optimization using additional installed Drupal modules	Server optimization	Optimization of configurations and Drupal Hooks
1	Main page	0.52	2.46	0.16	1.77
2	Blog List Page	0.52	0.77	0.91	1.4
3	One blog page	0.19	0.082	0.01	9.32

This method has shown that it is one of the most effective and necessary. Also, this method highlights problems with certain modules, in our case, it is the Advanced CSS/JS Aggregation module, which reduces the number of requests to the server by combining, minifying, and caching CSS and JS files. This module stores the cache in the database. However, the tables of this module are not normalized. This increases the response of the server and increases the load on MySql.

Optimizing configurations and Drupal Hooks is a must for every Drupal site. In our case, the normalization of databases was carried out, which reduced the load on the database management system, the configuration settings were correctly performed,

which protects the site from unwanted bots and users, and also reduces the load on the database, reduces the storage of user sessions, and logging messages of site operation.

5 Conclusions

The analysis of publications on this issue shows that CMS Drupal is a sufficiently developed content management system. The effort of developers to provide their system with universality and increased functionality compared to competitors leads to an increase in the load on the server and a decrease in its speed. Thus, in this CMS there is a demand for server resources. In most cases, this problem is solved by choosing specialized expensive hosting services by allocating servers or several servers. Our work will be especially interesting and useful for developer practitioners who receive graphic and digital indicators of research on increasing the speed and performance of the server using effective, relatively simple, and free methods for existing sites and sites under development based on CMS Drupal.

Research results:

- built-in optimization is extremely simple to implement and inefficient.
- optimizing Drupal using add-on modules is one of the most effective ways to reduce server load. This method will be the most effective for the main page on which there are many non-optimized images, CSS, and JS files. However, it is mandatory for all Internet sites.
- server optimization is one of the most difficult, inefficient, and expensive methods.
- optimizing configurations and Drupal Hooks is complex and long-term, but at the same time the most effective. Also, this method highlights problems with certain modules, in our case, it is the Advanced CSS/JS Aggregation module, which reduces the number of requests to the server by combining, minifying, and caching CSS and JS files.
- optimizing configurations and Drupal hooks are mandatory for every Drupal site.

In the future, we can use methods to increase the speed and performance of the Drupal CMS-based server to balance the load on IIoT servers. Load balancing is required to evenly distribute the processing work of two or more devices, network links, storage devices, or other devices, ultimately providing faster service with greater efficiency. Analysis and distribution of load in cloud systems used in IIoT is an urgent task since most cloud systems with open access use simple load schedulers of their physical servers.

References

1. Metinvest Digital, https://metinvest.digital/ua/page/globalne-dosl-dzhennya-idc-shcho-cheka-rinok-danih-u-nastupn-5-rok-v, last accessed 2022/12/04.
2. Drivas, I., Kouis, D., Kyriaki-Manessi, D., Giannakopoulos, G.: Content Management Systems Performance and Compliance Assessment Based on a Data-Driven Search Engine

Optimization Methodology. Information. 12(259) (2021). https://doi.org/10.3390/info12070259.
3. Satsyk, V., Smolyankin, O., Grudetskym, R.: Decrease of load on server by means of CMS Drupal. Sciences of Europe 32, 74-81 (2018).
4. Jagamogan, R., Ismail, S., Hafizah, N., Abas, H.: A Review: Penetration Testing Approaches on Content Management System (CMS). 7th International Conference on Research and Innovation in Information Systems (ICRIIS), 1-6 (2021) doi: 10.1109/ICRIIS53035.2021.9617087.
5. McDonald, C., Burkhardt, H.: Library-Authored Web Content and the Need for Content Strategy. Information Technology and Libraries, 38(3), 8-21 (2019) https://doi.org/10.6017/ital.v38i3.11015.
6. Krouska A., Troussas C., Virvou M.: Comparing LMS and CMS platforms supporting social e-learning in higher education. 8th International Conference on Information, Intelligence, Systems & Applications (IISA), 1-6 (2017) doi: 10.1109/IISA.2017.8316408.
7. Lapteva, U., Kuzyakov, O.: Rationale for Principles of Developing Control and Protection of Web Content Using CMS Drupal. International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon-2018), 1-6 (2018) doi: 10.1109/FarEastCon.2018.8602487.
8. Satsyk, V., Grudetsky, R., Kuzmych, O., Bahniuk, N., Hlynchuk L., Melnychuk, Y.: Reduction of Server Load by Means of CMS Drupal. 2020 10th International Conference on Advanced Computer Information Technologies (ACIT), 523-528 (2020) doi: 10.1109/ACIT49673.2020.9208874.
9. Ferrnandes, J., Lourenço, A.: Bringing Named Entity Recognition on Drupal Content Management System. In: Saez-Rodriguez, J., Rocha, M., Fdez-Riverola, F., De Paz Santana, J. (eds) 8th International Conference on Practical Applications of Computational Biology & Bioinformatics (PACBB 2014). Advances in Intelligent Systems and Computing, 294. Springer, Cham (2014) https://doi.org/10.1007/978-3-319-07581-5_31.
10. Skulimowski, A., Badecka, I.: Software Innovation Dynamics in CMSs and Its Impact on Enterprise Information Systems Development. In: Tjoa, A., Xu, L., Raffai, M., Novak, N. (eds) Research and Practical Issues of Enterprise Information Systems. CONFENIS 2016. Lecture Notes in Business Information Processing, 268, Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49944-4_23.
11. Li, C., Zhang, Q., Huang, C. et al.: Optimal Service Selection and Placement Based on Popularity and Server Load in Multi-access Edge Computing. J Netw Syst Manage 31, 15 (2023). https://doi.org/10.1007/s10922-022-09703-2.
12. Bouflous, Z., Ouzzif, M., Bouragba, K.: Analysis of Load Balancing Algorithms Used in the Cloud Computing Environment: Advantages and Limitations. In: Arai, K. (eds) Proceedings of the Future Technologies Conference (FTC). Volume 3. FTC 2022 2022. Lecture Notes in Networks and Systems, 561. Springer, Cham. (2023) https://doi.org/10.1007/978-3-031-18344-7_13.
13. Hessen, S., Abdul-kader, H., Khedr, A., Salem, R.: Load balancing based on multi-agent framework to enhance cloud environment. Computers, Materials & Continua, 74 (2), 3015–3028 (2023).
14. Xiong, Z., Zhao, M., Yuan, Z., Jianlong, C.: Energy-saving optimization of application server clusters based on mixed integer linear programming, Journal of Parallel and Distributed Computing, 171, 111-129 (2023) https://doi.org/10.1016/j.jpdc.2022.09.009.
15. Moroz, S., Tkachuk, A., Khvyshchun, M., Prystupa, S., Yevsiuk, M.: Methods for Ensuring Data Security in Mobile Standards. Informatyka, Automatyka, Pomiary W Gospodarce I Ochronie Środowiska, 12(1), 4-9 (2022). https://doi.org/10.35784/iapgos.2877.

16. Kumar, N., Sharma, B., Narang, S.: Emerging Communication Technologies for Industrial Internet of Things: Industry 5.0 Perspective. In: Singh, P.K., Wierzchoń, S.T., Tanwar, S., Rodrigues, J.J.P.C., Ganzha, M. (eds) Proceedings of Third International Conference on Computing, Communications, and Cyber-Security. Lecture Notes in Networks and Systems, vol 421. Springer, Singapore (2023). https://doi.org/10.1007/978-981-19-1142-2_9.
17. Bakhovskyy, P., Yevsiuk, M., Zabolotnyi, O., Cagáňová, D., Tkachuk, A.: Stages of the Virtual Technical Functions Concept Networks Development. In: D. Cagáˇnová et al. (eds.), Advances in Industrial Internet of Things, Engineering and Management, EAI / Springer Innovations in Communication and Computing, 119-135 (2021) https://doi.org/10.1007/978-3-030-69705-1_7.
18. Rafique, F., Obaidat, M., Mahmood, K., Ayub, J., Ferzund, M., Chaudhry, S.: An Efficient and Provably Secure Certificateless Protocol for Industrial Internet of Things. In IEEE Transactions on Industrial Informatics, 18(11), 8039-8046 (2022) doi: 10.1109/TII.2022.3156629.