UDC: 004.77, 004.728 https://doi.org/10.20535/2786-8729.4.2024.311595

METHOD OF LOAD BALANCING IN DISTRIBUTED THREE-LAYER IOT ARCHITECTURE

Anatolii Haidai
National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine
ORCID: http://orcid.org/0000-0001-9330-414X

Iryna Klymenko
National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine
ORCID: http://orcid.org/0000-0001-5345-8806

In connection with the constant growth of the number of Internet of Things devices, there is a need for more and more computing power. The traditional cloud computing model may not meet the computing speed needs of real-time systems. This problem is solved by Edge Computing, which involves moving computing resources closer to data sources, reducing data transfer and processing time. Adding new data processing devices at the system edge makes managing such networks much more difficult. This paper proposes a new method of load balancing on Edge nodes using the Zabbix monitoring system.

Key words: *load balancing, Internet of Things, Edge computing, Zabbix.*

1. Introduction

The growing number of IoT devices that have completely different functions and generate a large amount of data that needs to be transported, processed and stored poses new challenges for IoT specialists.

Although cloud solutions are attractive for most small systems, they are not the best for wider use, as they require rather voluminous communication channels and expensive equipment to process the incredible amount of information received every second of the system's operating time. Accordingly, in order to reduce the load on the network, reduce the time of processing and data transmission, calculations should be carried out as close as possible to the data source. Here, we are introduced to the concept of edge computing, which describes this distributed processing of received information near its source.

The transition to Edge and Fog technologies in IoT is currently quite attractive and widely used from the point of view of increasing system performance and reliability [1]. At the same time, an intermediate layer is added between the layer that contains many IoT devices and the data center, which is built on a certain number of new network devices. This, in turn, creates a new problem caused by the complexity of the IoT network infrastructure. Modern multi-level and heterogeneous IoT architectures are based on complex network infrastructure solutions, which are often dynamic structures. All this increases the complexity of network management and maintenance and makes it relevant and expedient to solve problems, to develop effective means of functioning of the network infrastructure of the IoT system, to develop means of load balancing, effective use of resources, and to increase reliability by reducing the number of bottlenecks and points of system failure.

Based on this, we get a new task, the main goal of which is to make a fairly effective method of managing available system resources and the load on computing nodes.

2. Literature review and problem statement

The authors of the paper [2] consider the methods of load distribution and energy saving when using fuzzy computing. Using the DRAM method for load balancing in fog computing has its

advantages, but it does not take into account energy consumption, which the authors say is important for IoT devices without a constant power source. Next, the authors demonstrate their method that will allow taking into account the above-mentioned factors and compare it with existing methods using simulations. As a conclusion, they have significant improvements in reducing the packet loss ratio and data transmission delay. In the following work [3], the authors draw attention to the fact that IoT devices can be quite mobile, and not static, since they can be placed in different vehicles, which in turn have both static routes and can have dynamic routes. This should be taken into account in system load balancing to reduce unnecessary overriding of processing devices for static routing of mobile IoT devices. Further, in [4], the authors propose their method of routing and balancing for IoT devices. Routing in the proposed method is faster, according to the authors' calculations, because not one route is selected when sending data further, but several, which in turn reduces the time when data is sent unsuccessfully and when a new route is searched. To build the network, beacon messages are used, which also contain information about the load of the node. Then the node with the lowest load according to the latest beacon messages is selected for balancing.

The following work [5], devoted to the review of balancing methods both in the cloud and in separated IoT networks. Thus, the authors highlight the main areas of further research that need to be conducted to improve the operation of IoT systems, namely:

1. Management of data priorities, according to their processing time.
2. Load balancing based on traffic.
3. It is necessary to take into account more parameters, determine their impact on the system as a whole, and determine parameters that can replace each other depending on the direction of system operation.
4. Optimization of already existing solutions that have high productivity.
5. The use of fog computing in the analysis of large volumes of data, with the distribution of results and data sets on fog network nodes.
6. Increasing interoperability between different solutions, such as cloud and fog computing, depending on the purpose of the system.
7. Improving the efficiency of load management.

Further, methods for load balancing in the cloud when using it as a data processing center for IoT devices are considered in [6]. Thus, the authors propose their method of load balancing, which, compared to existing balancing algorithms, distributes tasks and information more quickly between virtual machines in the cloud, according to the parameters defined by the authors.

The following works [7], [8] are devoted to data management in IoT, using SDN, which also allows balancing the load between different computing nodes of the IoT network. The division of the network into clusters, the routing and load of which is managed by the controller, can significantly improve performance and extend the time of uninterrupted operation of the system [8], which was verified in simulations based on existing algorithms and the algorithm proposed by the authors of the work. In [9], a comprehensive review of load balancing methods using SDN was carried out, and the authors conclude that in the future it is necessary to increase the efficiency of SDN for managing balancing in data types of systems and optimizing the data center.

Currently, there are several main problems that need to be solved in IoT systems. The first is the distribution of the load on the computing nodes for devices that have static and dynamic placement. This requires an analysis of data on the load of the computing node to which the device is connected and neighboring nodes to which data can be redirected. The second problem is the management of such a distributed system, it is necessary that the computing nodes can distribute the load from IoT devices themselves in order to reduce the dependence on the cloud server and increase the reliability of the system when there is no connection with the central server.

3. The aim and objectives of the study

The purpose of the study is to justify the three-layer architecture of the IoT system, which will balance the load and control the load using elements of the monitoring system. The implementation of the research goal is aimed at increasing the efficiency of load balancing and improving management

in the IoT system based on the proposed architectural solution, which is implemented using a mathematical model. Implementation of the research goal will make it possible to increase the efficiency of such systems and simplify the management of the IoT system.

To achieve the goal, the following tasks were set:

- To substantiate the concept of a three-layer IoT architecture to increase the effectiveness of implementing distributed management of network infrastructure and making management decisions. Develop a research prototype of a three-layer IoT architecture.
- To develop a method of load balancing in a distributed three-layer IoT architecture in a multi-level heterogeneous network infrastructure, which provides load balancing between Edge nodes.

4. Materials and methods for developing a load balancing system in a distributed three-layer IoT architecture

4.1. Justification of the architectural concept and parameters of influence on computing performance at the edge of IoT

This work examines the concept of Edge computing [10], in which the authors consider the architecture of the IoT system, which has three layers. The first layer is the devices that generate data, the second layer is the Edge, where the data is processed, and these Edges should be located closest to the data source, thus reducing the delay, the third layer is the cloud server that stores the data.

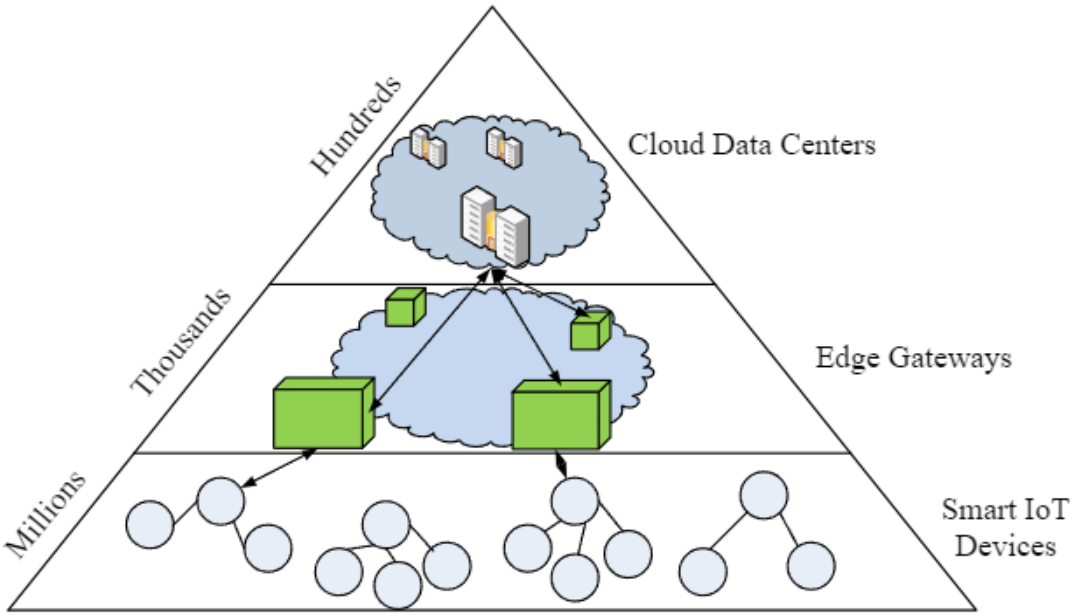


Fig. 1. Layered model of cloud edge-based IoT service delivery [10]

A description of the architecture and its advantages is provided in the following excerpt:"Given that data are rapidly produced at the edge of networks, dealing with these data at the edge of the network would be effective. Several approaches, such as cloudlet, fog computing, and mobile edge computing (MEC), provide complementary solutions to cloud computing to reduce data processing on the network edge. In short, edge computing is a general term that represents fog computing, MEC, cloudlets, and micro clouds. Storage, computing, and power are regarded as being on the edge of networks to increase availability, reduce latency, and eventually overcome cloud computing issues. Edge computing facilitates the processing of delay- sensitive and bandwidth-hungry applications near the data source. Figure 1 illustrates a layered model for cloud edge-based IoT service delivery [10].

The main parameters affecting computing performance include processor performance, amount of RAM, available storage space for data and computation results, and the quality of the network connection for receiving data and transmitting results.

If these four parameters are not provided at a sufficient level, the device will not be able to perform calculations with minimal delays and efficiently process all received data. In addition, there are other important parameters that affect the overall performance of a computing device, such as the reliability of its components and the temperature regime. These parameters should also be taken into account, as they can signal possible malfunctions of the device. This is especially important for timely transfer of calculations to another node during a failure and saving data on a cloud server.

For the work of the developed method, 3 parameters were chosen, namely:

– CPU load;

– Memory usage;

– Free disk space.

4.2. Description of the prototype of the three-layer IoT system

According to the architectural concept considered in the study (Fig. 1), a prototype consisting of several virtual servers was developed.

The prototype consists of four servers, two servers based on the Ubuntu Server 24.04 operating system, the other two using the Raspberry Pi Desktop operating system. These virtual servers are deployed on the ESXi virtualization platform, which, in turn, is placed on HP DL360 G7 physical servers. These servers have four gigabit interfaces that are connected to HP J9050A ProCurve 2900, and at the ESXi level to one virtual switch that provides connection of virtual machines to the network.

The structural diagram of the developed prototype is shown in fig. 2. Each Ubuntu Server 24.04 virtual machine has six cores and twelve gigabytes of RAM. The Raspberry Pi Desktop data server has 4 cores and 4 gigabytes of RAM.

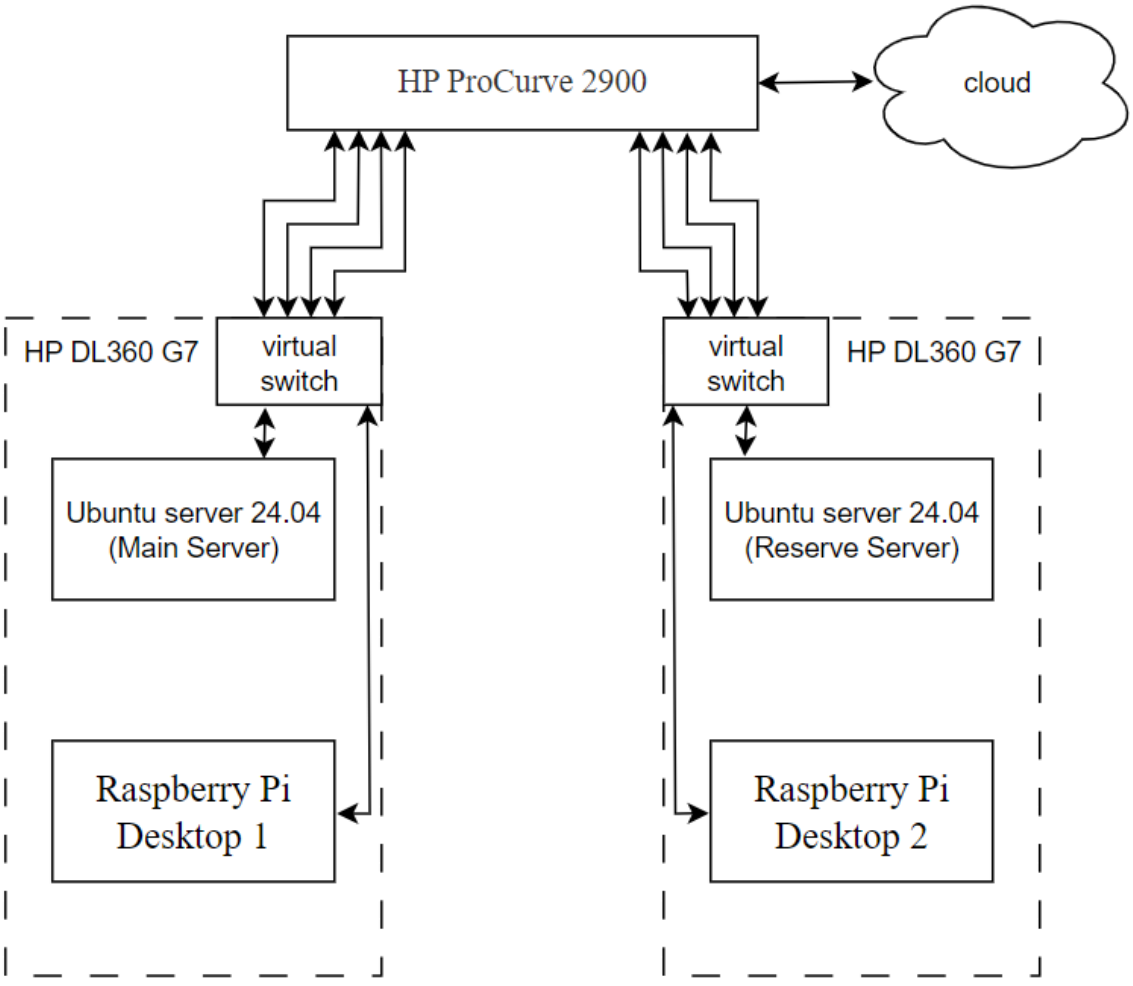


Fig. 2. Structural diagram of the prototype

Raspberry Pi Desktop 1 and Raspberry Pi Desktop 2 act as data servers that generate certain information for computing nodes. Ubuntu Server 24.04, Main Server and Reserve Server are edge devices that process data and act as monitoring devices for both data servers and the nearest processing devices, including themselves. This is required to distribute the load between computing devices, by obtaining the values of the workload parameters from them and running the corresponding scripts.

The cloud server in this model is used only to store the final results and to monitor the parameters of the calculation nodes, so that in the event of their failure, the loss of data coming from the sources is minimal.

The prototype consists of two virtual data processing servers, each of which has an installed instance of the Zabbix [11] monitoring system, which is additionally configured to execute scripts in accordance with the activation of the monitoring system to detect the specified server behavior patterns.

This prototype is designed for cases when the edge device has a significant load, which can negatively affect the speed of receiving and processing data.

4.3. Development of a mathematical model for server load monitoring

To monitor the server load, a mathematical model of the server load determination process is proposed, where n is the number of recent measurements of server parameters.

The following parameters are considered in the proposed model:

$$N_m = 100 - \frac{1}{n} * \sum_{i=1}^{n} x_i, \quad (1)$$

where x_i is the percentage value of the measurement of the amount of free memory on the device, N_m is the device's memory load.

$$N_p = 100 - \frac{1}{n} * \sum_{i=1}^{n} y_i, \quad (2)$$

where y_i is the percentage value of the processor idle time on the device, N_p is the load of the device on the processor.

$$N_d = 100 - \frac{1}{n} * \sum_{i=1}^{n} s_i, \quad (3)$$

where s_i is the percentage value of the amount of free space on the disk, N_d is the load of the device on the disk.

$$P = N_m * 0{,}15 + N_p * 0{,}8 + N_d * 0{,}05, \quad (4)$$

where P is the total server load.

According to the specified formula (4), the current server load is calculated, while the following coefficients are selected for each described parameter of the formula:

– the coefficient of 0.8 for the processor is chosen to run the scripts at 100% load, because at lower coefficients if the calculation task has used the entire CPU and other resources are free, data processing can take place with a significant delay;

– the coefficient of 0.15 for memory is chosen under the condition of performing calculations without using large buffers, to store data in memory during processing;

– the coefficient of 0.05 for disk space due to the necessary availability of space to save the results before transmission, but this parameter has the smallest impact on data processing.

Coefficients are selected depending on the level of importance of each parameter relative to the target task. These coefficients were chosen to implement a special script for emulating system operation when all processor resources, a small amount of memory resources and disk space are used.

Depending on the types of data that will be sent to the calculation nodes and the principles of their processing, the coefficients may change.

4.4. The method of load balancing in a distributed three-layer IoT architecture

To monitor the server load, a mathematical model of the server load determination process is proposed, where n is the number of recent measurements of server parameters.

A method of load distribution between edge nodes in the IoT system has been developed, which is based on the implementation of the proposed load distribution model (4). The reliability of the method was investigated based on the use of the developed prototype of the three-layer IoT system. According to the statement of the problem, the Edge Main processing server (Fig. 2) has a high level of workload.

Stages of the load distribution method:

Stage 1. According to expression (4), the load of the servers, Main Server and Reserve Server, to which the Raspberry Pi Desktop 1 and Raspberry Pi Desktop 2 IoT devices are connected, respectively, is calculated.

Stage 2. Depending on the calculation results, one of two scenarios is considered:

1. The server has a load factor of less than 80%, moving to stage 1.

2. The server has more than 80% load factor. Let's go to stage 3.

Stage 3. If the load condition of more than 80% is met, the data value evaluation trigger is triggered; where the trigger is a logical expression that evaluates the data values collected by the system and compares the obtained values according to the set coefficient (80%).

Stage 4. Checking the load of the Reserve server, if its load is less than 80%, it starts the load redistribution script when the data server connects to the Reserve server.

Stage 5. Creation of a new data source object on the Reserve server, while the new IP address of the processing server is transferred to the data server, Raspberry Pi Desktop 1.

Stage 6. Upon successful completion of stage 5, disconnect Raspberry Pi Desktop 1 from the Main server, followed by temporary disconnection of the data source object from the Main Server system.

Stage 7. Checking the server load according to expression (4), if the load is exceeded, return to stage 3.

5. Carrying out research on the method of load balancing in a distributed three-layer IoT architecture

The research of the method was carried out on the basis of the use of the prototype of the three-layer IoT system, which is described in point 4.2. To conduct experiments, scripts have been developed that are integrated into the Zabbix open source system software.

1. To conduct experiments, scripts were developed for system load emulation and load balancing, and virtual machines with the necessary software were prepared.

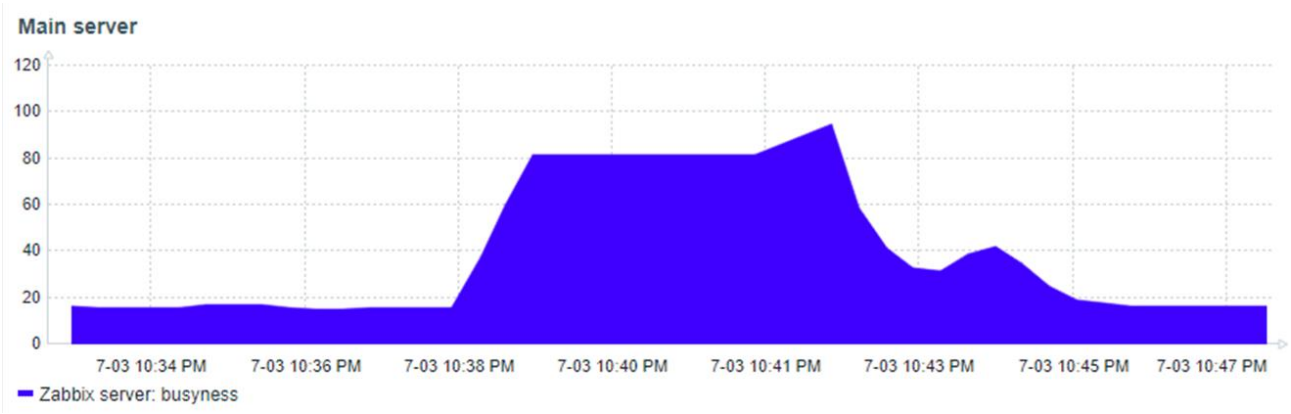
2. On the Edge servers, a load emulation script was configured to run when the Raspberry Pi Desktop 1 data server was connected.

3. Every 20 seconds, according to the mathematical model, load calculations are carried out on the Main Server and Reserve Server, n in this simulation is equal to 9.

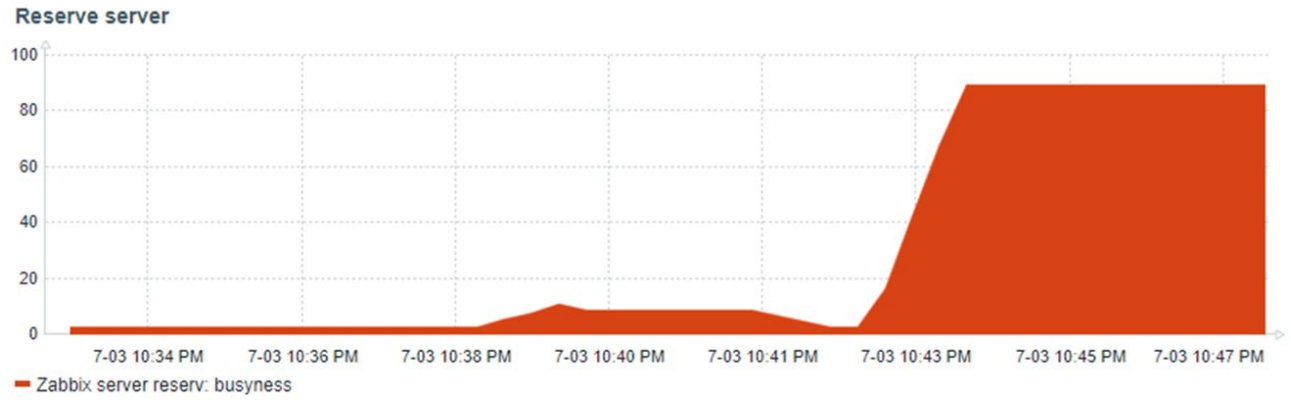
4. Fig. 3 shows the load graphs of the Main Server, it can be seen that at 10:38 PM a massive calculation task was launched, the result display time is slightly different from the current one due to load measurements every 20 seconds.

5. After 3 minutes of operation of the load script on the Main server, the trigger is triggered.

6. Next, a script is launched to transfer the data generation node to the Reserve server, this adds to the load on the Main server, as can be seen from the graph.



a



b

Fig. 3. Load graph of two servers: a – Main Server; b – Reserve Server

7. After transferring the data node, it can be seen that the Reserve Server starts calculations and increases its load by more than 80%, at the same time, the load of the Main Server can be seen to drop, because it has given the data node and is no longer engaged in processing its information.

6. Discussion of the results of the proposed load balancing method in the three-layer IoT architecture.

In the previous sections, the three-layer architecture of the IoT system was considered, which consists of three levels: the first level is responsible for creating data, the second level deals with its processing, and the third, the cloud level, provides data processing and storage. To evaluate the effectiveness of the proposed load distribution method, a system prototype was created based on virtual servers using the latest version of the Ubuntu operating system. The load distribution method was configured based on data obtained from the Edge server's load monitoring system.

Analysis of the results, presented in the form of graphs, showed that the method worked according to the developed algorithm: in case of overloading of the Main Server, the data server is transferred, for further calculations and load reduction, to the Reserve Server, where data processing continued. This confirms the effectiveness of the proposed approach for improving the allocation of computing resources in IoT systems. The results also show the possibility of adapting this method for different use cases, for example, to include additional computing units, such as GPUs, by making appropriate changes to the load distribution formula, leaving other system components unchanged. This makes the proposed method potentially useful for integration into systems using artificial intelligence.

The Zabbix monitoring system plays a key role in the proposed method, as it allows the integration of different data-generating devices using a wide range of protocols. Zabbix can accumulate this data for further processing or transmission to a cloud server, and also provides the ability to run scripts based on predefined formulas and algorithms. This makes the system flexible and scalable, which ensures its adaptation to the specific needs of various IoT infrastructures.

7. Conclusion

The considered architecture of the distribution of the IoT system into layers, with the allocation of computing processes to Edge nodes, has significant advantages in the speed of data transmission for processing. However, this model complicates system management, as it increases the number of nodes that require monitoring and management.

To test the effectiveness of the proposed load balancing method, a prototype of a three-layer IoT system was created. On this prototype, the operation of the method, which is guided by the developed mathematical model for managing data servers and Edge nodes, was tested. The method allows you to dynamically distribute the load based on the current performance indicators of computing nodes, which increases the overall efficiency of the system.

Using the proposed mathematical model allows you to estimate the load on Edge servers, taking into account the type of data and the used system resources. Data processing at intermediate nodes with subsequent transfer to cloud servers ensures an even load distribution in the system, which allows the use of less powerful and cheaper communication channels. In addition, such a system is more resistant to failures and network problems, as it can temporarily store the results of data processing on Edge nodes, which reduces the risk of information loss and ensures the preservation of intermediate results before they are transferred to storage servers.

References

[1]I. Klymenko, A. Gaidai, S. Nikolskyi, V. Tkachenko, "The Architectural Concept Of The Monitoring System On The Basis On A Neuron Module IoT Data Analytics", Adaptive systems of automatic control, p. 111-123, 2022, doi:10.20535/1560-8956.41.2022.271355

[2] M. Shuaib, S. Bhatia, S. Alam, R. K. Masih, N. Alqahtani, S. Basheer, and M. S. Alam, "An Optimized, Dynamic, and Efficient Load-Balancing Framework for Resource Management in the Internet of Things (IoT) Environment," Electronics, vol. 12, no. 5, pp. 1104, 2023. doi: 10.3390/electronics12051104.

[3] N. Shivaraman, J. Fittler, S. Ramanathan, A. Easwaran and S. Steinhorst, "WiP Abstract: Mobility-based Load Balancing for IoT-enabled Devices in Smart Grids," 2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS), Sydney, NSW, Australia, 2020, pp. 192-193, doi: 10.1109/ICCPS48487.2020.00029.

[4] H. Tseng, "Multipath Load Balancing Routing for Internet of Things," Journal of Sensors, vol. 2016, pp. 1-8, 2016. doi: 10.1155/2016/4250746.

[5] D. Kanellopoulos and V. K. Sharma, "Dynamic Load Balancing Techniques in the IoT: A Review," Symmetry, vol. 14, no. 12, p. 2554, 2022. doi: 10.3390/sym14122554.

[6] K. Devi, D. Sumathi, V. Venkataraman, A. Chunduru, K. Kataraki, and S. Balakrishnan, "CLOUD load balancing for storing the internet of things using deep load balancer with enhanced security," Measurement: Sensors, vol. 28, p. 100818, 2023. doi: 10.1016/j.measen.2023.100818.

[7] Z. Eghbali and M. Zolfy Lighvan, "A hierarchical approach for accelerating IoT data management process based on SDN principles," Journal of Network and Computer Applications, vol. 181, p. 103027, 2021. doi: 10.1016/j.jnca.2021.103027.

[8] E. Hajian, M. R. Khayyambashi and N. Movahhedinia, "A Mechanism for Load Balancing Routing and Virtualization Based on SDWSN for IoT Applications," in IEEE Access, vol. 10, pp. 37457-37476, 2022, doi: 10.1109/ACCESS.2022.3164693.

[9] M. R. Belgaum, S. Musa, M. M. Alam and M. M. Su'ud, "A Systematic Review of Load Balancing Techniques in Software-Defined Networking," in IEEE Access, vol. 8, pp. 98612-98636, 2020, doi: 10.1109/ACCESS.2020.2995849.

[10] N. Hassan, S. Gilani, E. Ahmed, I. Yaqoob, and M. Imran, "The Role of Edge Computing in Internet of Things," IEEE Communications Magazine, vol. PP, 2018. doi: 10.1109/MCOM.2018.1700906.

[11] Zabbix Documentation, Zabbix, 2023. [Online]. Available: https://www.zabbix.com/documentation/current/en/manual. [Accessed: Sep. 4, 2024].

УДК 004.77, 004.728 https://doi.org/10.20535/2786-8729.4.2024.311595

МЕТОД БАЛАНСУВАННЯ НАВАНТАЖЕННЯМ В РОЗПОДІЛЕНІЙ ТРИШАРОВІЙ АРХІТЕКТУРІ IOT

Анатолій Гайдай
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0001-9330-414X

Ірина Клименко
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0001-5345-8806

У зв'язку зі зростанням кількості IoT пристроїв і необхідністю швидкої обробки даних з мінімальними затримками, традиційні хмарні обчислення стають менш ефективними. Для вирішення цієї проблеми застосовується концепція граничних обчислень, яка, хоча й підвищує продуктивність, ускладнює управління системою та вимагає ефективного розподілу навантаження для забезпечення балансу між використанням ресурсів Edge-вузлів та швидкістю обчислень. Метою даної роботи було розроблення методу балансування навантаження в тришаровій архітектурі IoT-системи, враховуючи фактичне завантаження вузлів.

Проведено огляд літератури та обрано архітектурну концепцію яка відповідає новим тенденціям та складається з трьох шарів: пристроїв, що генерують дані, граничних вузлів, що обробляють інформацію, та хмари, яка зберігає дані і надає їх користувачам. Було створено прототип системи, який включає кілька Edge-вузлів на базі операційної системи Ubuntu Server 24.04 та сервери даних на основі Raspberry Pi Desktop. Розроблено математичну модель, що дозволяє оцінити навантаження на вузли залежно від типу виконуваних завдань. На створеному прототипі проведено перевірку працездатності методу з використанням математичної моделі.

Результати дослідження показали, що розроблений метод успішно розподіляє навантаження між Edge-вузлами за допомогою спеціальних скриптів та елементів системи моніторингу, що відображено на графіку навантаження серверів. Запропонований метод може підвищити продуктивність системи завдяки автоматичному розподілу навантаження між вузлами. Цей підхід може стати частиною більш комплексної стратегії підвищення продуктивності та надійності IoT-систем із використанням граничних обчислень. Використання компонентів системи моніторингу для різних платформ з різною потужністю дозволяє знизити вартість системи, застосовуючи дешевші та менш потужні обчислювальні пристрої.

Ключові слова: *балансування навантаженням, інтернет речей, граничні обчислення, Zabbix.*