Larysa MURYTENKO[1], Valeriia SOLODOVNIK[2]

Scientific supervisor: Yanina SHESTAK[3]

OPRACOWANIE METODYKI ROZWOJU SPRZĘTOWYCH TECHNIK RÓWNOWAŻENIA OBCIĄŻENIA ROZPROSZONYCH SIECI INFORMACYJNYCH

Streszczenie: Uwzględniono specyfikę wysokowydajnego i energooszczędnego przetwarzania dużych rozproszonych sieci informacyjnych. Zaproponowano matematyczny model przetwarzania rozproszonych sieci informacyjnych w oparciu o graf zadań. W celu rozwiązania problemów związanych z harmonogramowaniem przeanalizowano podział systemu zadań i zasilanie. Porównano wydajność opracowanego modelu matematycznego przetwarzania rozproszonych sieci informacyjnych na dużą skalę z danymi statystycznymi.

Słowa kluczowe: sieci informacji rozproszonych, model matematyczny, sprzętowe równoważenie obciążenia, grupowanie, graf zadań, algorytmy partycjonowania, algorytmy partycjonowania

ALGORITHM OF LOAD BALANCE OPTIMIZATION ON HARDWARE RESOURCES OF INFORMATION SYSTEMS

Summary: The analysis of aspects of load balancing on hardware resources of distributed information systems is carried out. A mathematical apparatus for optimizing the load distribution of the network of a distributed information system based on the corresponding mathematical model is built. The method of constructing the graph of tasks of the information system based on the schedule of requests, and also parallelization of planning tasks and the multilevel scheme of distribution hardware and software resources is improved.

Keywords: distributed information networks, hardware load balancing, query generator, control center, relative rating, task graph.

[1] Taras Shevchenko National University of Kyiv, Faculty of Informaition Technology, Cyber Security and Information Protection: myrutenko.lara@gmail.com
[2] Taras Shevchenko National University of Kyiv, Faculty of Informaition Technology, Cyber Security and Information Protection: valeriamin6@gmail.com
[3] Taras Shevchenko National University of Kyiv, Faculty of Informaition Technology, Cyber Security and Information Protection: lucenko.y@ukr.net

1. Formulation of the problem

The effectiveness of the development and modification of distributed information systems (DIS) largely depends on solving systemic and architectural problems, such as the choice of algorithms for load balancing on hardware resources and the introduction of methods for parallel processing of incoming requests [1, 2].

The development of load balancing methods should be preceded by the development of an appropriate mathematical model, which at the basic level can be represented as a set of elements such as query generator (query generator, QG) and control center (CC). QG creates requests and sends them for processing through individual threads. CC processes requests received from QG. CC includes a computer, a request buffer, and dispatchers whose structure includes secondary control centers (SCCs). SCCs are started to process queues in the buffer. With the minimum traffic of requests, all calculations are performed at the CC level in accordance with the order of their submission to the QC. As the number of requests increases, the computing power CC may be insufficient for processing, and thus the requests accumulate in the buffer. When the number of requests in the buffer specified by the DIS administrator is reached, CC launches separate streams of dispatchers that balance the load on the system hardware resources. The manager calculates the optimal number of SCCs, each of which is then run in a separate thread.

When balancing the load on DIS hardware resources, the load distribution is determined both between the network clusters and within each cluster. The load level is determined by the load factor k_CP of the central processors (CP) of DIS servers, which in the construction of the mathematical apparatus can be represented as a function of the number of cycles per session and the total number of requests server n_Q, as well as the following server architecture:

- set of CP clock values as a one-dimensional matrix of values for individual servers $f_{CP}\colon\{f_{CP}^1, f_{CP}^2, \dots f_{CP}^i, \dots f_{CP}^I\}$, where I — is the total number of servers, or a one-dimensional matrix of averaged values for server clusters $\bar{f}_{CP}\colon\{\bar{f}_{CP}^1, \bar{f}_{CP}^2, \dots \bar{f}_{CP}^j, \dots \bar{f}_{CP}^J\}$, where J — is the total number of clusters;
- CP multicore coefficients as a one-dimensional matrix of values for individual servers $k_{MK}\colon\{k_{MK}^1, k_{MK}^2, \dots k_{MK}^i, \dots k_{MK}^I\}$ or a one-dimensional averaging matrix for server clusters $\bar{k}_{MK}\colon\{\bar{k}_{MK}^1, \bar{k}_{MK}^2, \dots \bar{k}_{MK}^j, \dots \bar{k}_{MK}^J\}$;
- operating system load factor by κ_Q requests;
- load factor associated with additional processes on the server κ_+.

Dynamic load balancing in real time should determine the proportion of distribution of requests between servers, for which this study proposes to introduce the concept of relative server rating (relative rating, RR) as a function of k_CP and n_Q and the average RR value for the DIS cluster:

$$R_i = \frac{n_Q^i}{k_{CP}^i \cdot \sum_{i=1}^{I}\left(n_Q^i\right)}, R_j = \frac{n_Q^j}{\bar{k}_{CP}^j \cdot \sum_{j=1}^{J}\left(n_Q^j\right)}. \qquad (1)$$

In fig. 1 shows a model of a complex of three servers capable of processing 1 × Q, 2 × Q and 3 × Q requests, respectively. If the server receives 6 × Q requests, then under conditions of uniform load distribution, the first server is overloaded, the second - loaded optimally, and the third - insufficiently loaded (Fig. 1-a). Moreover, if one of the servers breaks down, only one of the three will work effectively (Fig. 1-b).

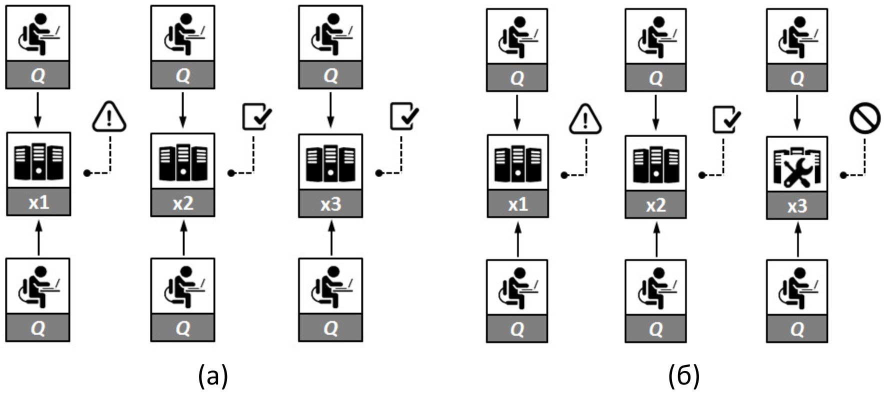

Figure 1. Diagram of organization of DIS operation without load balancing in normal (a) and emergency (b) operating mode.

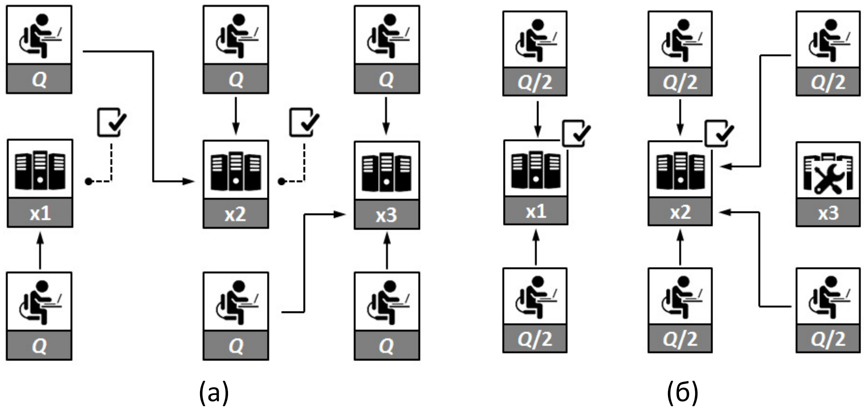

Figure 2. Scheme of organization of DIS work with load balancing in normal (a) and emergency (b) mode of operation.

When using the load balancing scheme, the first server receives 1 × Q requests, the second - 2 × Q requests, and the third - 3 × Q requests, in accordance with the relative rating of each of the three servers (Fig. 2-a). In the case of emergency network operation, in which the most powerful server fails, the computing power drops twice, but the resources are distributed evenly among the remaining servers (Fig. 2-b). Thus, the developed algorithm allows to dynamically determine the proportions of the distribution of requests to maintain a uniform load of available hardware resources.

At the level of solving system problems DIS should also be noted opportunities to optimize the operation of the computer system and load coordination provided by the multicore processors of the server complex. The computing power of a multi-core processor with the number of cores n_MK is close to the value of the power n_MK of single-core processors, or the power of one single-core processor with n_MK times higher frequency. However, the level of electricity required for its operation is less than the level of electricity used by a single-core processor with a higher clock speed or a multiprocessor server:

$$E_{MK} = \frac{E_0}{(n_{MK})^{\sigma}} \text{ , для } \begin{cases} \sigma \in \mathbb{N} \\ \sigma > 1 \end{cases}, \tag{2}$$

where E_0 — is the level of energy consumed by single-processor servers, σ is a parameter that depends on the construction of a multi-core processor. Thus, with proper organization of DIS based on multi-core servers, it is possible to significantly increase the performance of query processing by the system, by building a parallel architecture, at which the interaction between cores can be through data transfer and shared memory, and the degree of parallelism. Thus increasing with the number of nuclei. This opens up a wide range of possibilities for creating multitasking algorithms that are able to handle a larger volume of requests per unit time. In the case of DIS significant number of parallel queries, the main task is to allocate cores for each task by creating a task graph (TG), which provides balancing the load on hardware resources and the optimal level of power consumption [3-7].

According to the TG model, queries located at different levels of the system are considered independent, and therefore typical algorithms can be used to solve them, which makes it possible to solve the problem of queuing when planning tasks and simplifies system analysis. The implementation of TG must be preceded by a procedure for clustering DIS hardware resources. Within this architecture, each task queries multiple cores, so it is necessary to divide the processor cores into clusters. In addition, the scheme takes into account the mechanism of organization of resource allocation, ie determining the optimal proportions when providing for a particular task of hardware resources in order to ensure reliable operation in processing requests.

In this approach, the design of DIS should be based on the distribution for CP power servers, the level of which is determined by the value of the utilization factor CP - η_CP, clock speed, load capacity C and supply voltage U, and these parameters are not independent (supply voltage is a function of frequency, etc). Next, the performance of the CP as a function of f_CP and the processor architecture is determined. Accordingly, the power consumption index of the CP server complex DIS is calculated as:

$$P = \eta_{CP} \cdot (f_{CP} \cdot CU^2) = \eta_{CP} \cdot b^2 \cdot C \cdot f^{2\sigma+3} . \tag{3}$$

When processing n parallel queries, the queued task is a subset of the set of task graphs. All tasks i are performed in parallel and are characterized by the following set of parameters: (i) the number of CP cores used in the task is μ_i; (ii) the maximum level of requirements for the computing power of the cores is δ_i and is determined due to the number of commands to be executed when processing the request; (iii) the task operation is calculated as $w_i = \mu_i \cdot \delta_i$; (iv) the processing speed of the request S_n and the power provided for its execution are related through the parameter α, where $P_n = S_n^{\alpha}$; the query processing time t_n is determined by the parameter $\delta_n = t_n \cdot S_n$. Thus, for DIS with average values of CP parameters, which when balancing the load operate with one speed s_n for one t_N, the electricity consumed to process the request n is defined as:

$$e_n = \mu_n P_n t_N = w_n S_n^{\alpha-1} \quad (4)$$

Therefore, the DIS optimization process consists of two basic stages: the stage of construction of the task schedule, which is characterized by the smallest size, taking into account the limit on the amount of total energy E_Σ and the stage of determining the minimum of the total energy function $E_\Sigma\Sigma$, taking into account the restrictions on the size of the task schedule.

2. Conclusion

The analysis of key aspects of load balancing on hardware resources of the infrastructure of distributed information systems is carried out. The peculiarities of the operation of these systems arising from the introduction of modern information technologies, such as the development of multicore processors, process parallelization and system clustering, are considered.

A mathematical apparatus for optimizing the load distribution of the network of a distributed information system based on the corresponding mathematical model is constructed. The presented model included a request generator and a control center, which in turn consists of a computer, a request buffer and dispatchers. The efficiency of functioning of this system at work of infrastructure of information system in a normal and emergency mode is shown.

The method of construction of the graph of tasks of the information system based on the schedule of inquiries, and also parallelization of planning of tasks and the multilevel scheme of distribution of hardware and software resources is considered and improved. It is shown that the procedure of optimization of the distributed information system consists of stages of construction of the schedule of tasks characterized by the smallest size, taking into account the limit on the amount of total energy and determining the minimum function of total energy, taking into account restrictions on the schedule of tasks.

REFFERENCES

1. GUPTA H.: Load Balancing In Cloud Computing. International Journal of Recent Trends in Engineering and Research, *3*(2017)3, 260-267. doi:10.23883/ijrter.2017.3076.uimcu.
2. FATEMI M., SALIMI S., SALAHI A.: Anonymous roaming in universal mobile telecommunication system mobile networks. IET Information Security, 4(2010)2, 93. doi:10.1049/iet-ifs.2009.0154.
3. YANG C., LIN C.: Time-Varying Network Measures in Resting and Task States Using Graph Theoretical Analysis. Brain Topography, 28(2015)4, 529-540. doi:10.1007/s10548-015-0432
4. LI K.: Energy efficient scheduling of parallel tasks on multiprocessor computers, Journal of Supercomputing, 60(2012)2, 223–247.
5. LI K.: Power allocation and task scheduling on multiprocessor computers with energy and time constraints, Energy-Efficient Distributed Computing Systems, A. Y. Zomaya and Y. C. Lee, eds., Chapter 1, pp. 1-37, John Wiley & Sons, 2012.
6. KHAN S. U.: Handbook on Data Centers. New York, NY: Springer. 2015.

7. LI K.: Algorithms and analysis of energy-efficient scheduling of parallel tasks, Handbook of Energy-Aware and Green Computing, Vol. 1 (Chapter 15), I. Ahmad and S. Ranka, eds., pp. 331-360, CRC Press/Taylor & Francis Group, 2012.