Надежность № 2 2016
Dependability no.2 2016

Original article
УДК: 681.3
DOI: 10.21683/1729-2640-2016-16-31-38

Geometrical method for the operational control of the distributed solution of information-computing tasks in computer networks

Sergey A. Zhurbin, the 4th CSRI, Ministry of Defense of the Russian Federation, Korolev, Moscow Region, Russia, e-mail: serdgo4@yandex.ru
Gennady V. Kazakov, the 4th CSRI, Ministry of Defence of the Russian Federation, Korolev, Moscow Region, Russia, e-mail: kgv.64@mail.ru

Sergey A. Zhurbin

Gennady V. Kazakov

Abstract. Aim. *Some of the main performance indicators of ACS application are the operational efficiency and stability of the control of the above mentioned systems. The wide application of computing techniques in ACS as well as the organization of computer networks on this basis stipulate the necessity of effective control of distributed computation processes to ensure the required level of operational efficiency and stability while solving the specified tasks. The existing methods used to organize the computation process (method of dynamic programming, branch and bound method, sequential synthesis, etc.) may turn out to be bulky or less accurate in certain situations. These methods help to find a solution in the mode of interactive choice of an optimal variant to organize a computation process, i.e. consecutive approach to the required result and do not allow getting an a priori estimation of the time of computation process in a network. Application of the specified methods when solving research tasks in the course of design of computer networks presents itself as quite difficult. This article offers the application of a geometrical method that allows estimating the minimum time necessary to solve the set of information-computing tasks as well as ensuring their optimal assignment in a computing system. Besides, the method allows finding a full set of possible variants for the organization of a computation process in a network with an a priori estimation of time of the decision for each variant. The principle of the method is to represent the sets of all possible distributions of tasks by workstations in form of a broken hypersurface. To solve the indicated task the criterion and conditions of the optimality of the time spent to solve information-computing tasks have been introduced.* ***Results and conclusions.*** *This article describes many variants of realization of a computation process for homogeneous and non-homogeneous computing environments. Solution algorithm for a homogeneous computing environment is quite simple and makes it possible to define a minimum time necessary for a computation operations. It is based on a geometrical representation of the distribution of tasks by workstations in form of the hyperplane constructed in orthonormal space whose basis vectors are computation capacities of workstations. Besides, the algorithm for homogeneous computing environment can be successfully used for an approximate estimation of the minimum time necessary to solve a set of tasks in a network, for non-homogeneous computing environment as well. Minimum time necessary to solve functionally different tasks in a non-homogeneous computing environment is defined using a piecewise linear hypersurface that slightly complicates the algorithm, though in general, with consideration of computation capabilities of moderns computers, it is still simply realized. The estimations carried out in the course of preliminary researches, allowed concluding about the application of a geometrical method in a computer network for a large amount of workstations and information-computing tasks. The possibility of an a-priori estimation of the minimum time necessary to solve a set of tasks in the computer network allows using the offered method to solve research tasks at the stage of design of a computer network to estimate such indicators as operational efficiency, reliability, stability and etc. The possibility of an aprioristic assessment of the minimum time of the solution of a complex of tasks in the computer network allows to use, offered in work, a method in the solution of research tasks at a design stage of the computer network for an assessment of her such indicators as efficiency, reliability, stability, etc.*

Keywords: *automated workstation, computer network, quality indicators of computation process organization, control of computation process.*

For citation: *Zhurbin S.A., Kazakov G.V. Geometrical method for the operational control of the distributed solution of information-computing tasks in computer networks // Dependability, 2016, no.2, pp. 31-38. (in Russian) DOI: 10.21683/1729-2640-2016-16-31-38*

Currently the notion “computing systems” is no longer the innovation. Trends of the technology development are largely determined by the concerns to improve the performance of the current computing systems and the systems under development [1, 3, 4] by deployment of technical and technological novelties. Capacity of the first computers, as well as their functional reliability were not high enough, that is why they did not make it possible to solve many application tasks, or it took much time to solve them in view of all recovery processes required after failures. Just after the first computers appeared, different methods to combine several computers into one system were developed to improve the performance, functional reliability, and to reduce the time necessary to solve the tasks. The idea was simple: if the capacity of one computer was not enough to solve the tasks, then it was necessary to parallelize the whole set of tasks Ω among computers, and then each computer would solve its sub-set of the tasks Ω_i ($i = \overline{1, L}$), where L is the amount of computers in a computing system (CS). The same aim was pursued at the development of multiprocessor systems or, as they say, the systems with multi-core processors. With the advent of technical and technological possibility of information exchange between computers, a strong impetus was given to the development of the concept of CS construction which is a logical result of the evolution of computer technologies. So the computers were able to exchange information, they were provided with computation capacity and user-friendly interface, there was a little left to do – to teach them to control the computation process. The methods of distributed computation process, such as, for instance, the dynamic programming, the branch and bound method, sequential synthesis, etc. [1, 2, 3, 4] started their active development. These flexible generic approaches and methods became widely applicable in the efficient use of computing resources while solving a wide range of tasks: information, computing, technological and many other tasks. However, if we consider some one type of tasks, for example, computing tasks when it is necessary to distribute a certain set of one-type tasks in a multiprocessor or in a multi-core processor space, the generic methods may turn out to be bulkier or less accurate.

This article offers a geometrical method that allows estimating the time necessary to solve the tasks in homogeneous and non-homogeneous environments as well as ensuring their optimal assignment in CS. For a homogeneous environment, all computation processes in CS are assumed to be linear for all types of tasks, for a non-homogeneous environment – the linearity is observed only for the tasks of the same type, and there is no linearity for the set of different-type tasks. Linear and nonlinear relations mentioned above shall be described in more detail below.

Let there be a CS whose nodes are computing machines generally of different technical characteristics (the speed of a processor and a front side bus, random access memory capacity, etc.). The computer network nodes are automated workstations (WS), solving a certain set of the entered information-computing tasks (ICT). ICTs solved in CS are independent of each other from the point of view of the pooled input and output data. The task of distribution of the whole variety of ICTs to be solved in CS is laid on a certain control center CS (CSCS), which can perform the computation process in CS in automatic or automated mode. In case of the high level of technical facilities, and if CS is provided with data communication channels, a high end server performing automatic control of CS may serve as a CSCS. If the level of technical facilities is insufficient or low, and in case there are no required communication channels, total control is taken by a human, and information exchange is carried out by means of courier service. This paper describes the CSs of the high technical level with data communication channels of high performance. The sample of an unspecified CS is shown in Figure 1.

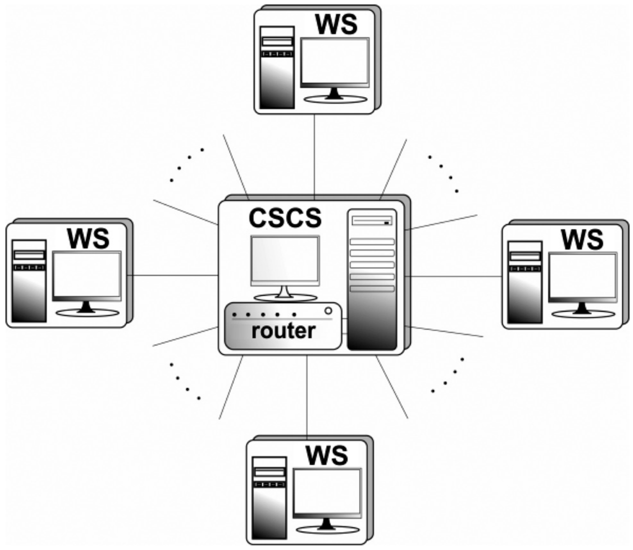

Fig. 1. Computer network. General view

The CS architecture may be rather diverse, but the CS itself should have several main properties:

WS of the network have the same rights and priorities in relation to the ICT solution;

The whole set of tasks solved in the CS can be solved at any WS;

WS, at which ICTs are solved, operate in parallel;

all WSs of the CS are thoroughly reliable.

Let the CS with m of WSs take M of ICTs including n types of tasks, i.e.

$$M = \sum_{j=1}^{n} \xi_j,$$

where ξ_j is the amount of tasks of the j-th type. The types of tasks differ, for instance, in scope and content of input and output data. Let for any pair of WSs the following equation hold true

$$\frac{t_{i,1}}{t_{j,1}} = \frac{t_{i,2}}{t_{j,2}} = \ldots = \frac{t_{i,n}}{t_{j,n}}, \quad (1)$$

where $t_{i,k}$ is the time to solve one task of the k-th type at the i-th WS; $t_{j,k}$ is the time to solve one task of the k-th type at the j-th WS ($i = \overline{1,m}$; $j = \overline{1,m}$; $i \neq j$; $k = \overline{1,n}$.).

Condition (1) specifies the fact that the time to solve an ICT at WS linearly depends only on the WS computation capacity, the time of bringing the initial information is not considered, as its scope is quite little and the times for information exchange between WSs are also negligible. For example, if the WS1 has a higher speed response than the WS2 by φ times, then for all types of tasks we can write

$$\frac{t_{1,1}}{t_{2,1}} = \frac{t_{1,2}}{t_{2,2}} = ... = \frac{t_{1,n}}{t_{2,n}} = \phi$$

Let us consider such CS as a homogeneous computing environment. Then the computation process should be organized in such a way, i.e. to distribute M of tasks between m WS of CS, so that the time to solve the whole set of tasks shall be optimal to some degree (optimality criteria will be described below).

Let us assume the whole set of ICTs being distributed somehow between WSs. The result of such distribution of the whole set of ICTs among WSs can be represented as a system of non-homogeneous linear equations

$$\begin{cases} v_{1,1}t_{1,1} + v_{1,2}t_{1,2} + ... + v_{1,n}t_{1,n} = T_1; \\ v_{2,1}t_{2,1} + v_{2,2}t_{2,2} + ... + v_{2,n}t_{2,n} = T_2; \\ \dots\dots\dots\dots\dots\dots \\ v_{m,1}t_{m,1} + v_{m,2}t_{m,2} + ... + v_{m,n}t_{m,n} = T_m, \end{cases} \quad (2)$$

where $v_{i,j}$ is the amount of the ICTs of the j-th type, distributed to the i-th WS; T_i is the total time of solution of the ICTs, distributed to the i-th WS.

And for system (2) the following condition should be satisfied

$$\begin{cases} \sum_{i=1}^{m} v_{i,j} = \xi_j,\ j = \overline{1,n}; \\ \sum_{j=1}^{n} \xi_j = M. \end{cases} \quad (3)$$

Let assuming (1)

$$\frac{t_{1,j}}{t_{i,j}} = \alpha_i,\ i = \overline{1,m},\ i \neq 1. \quad (4)$$

Then let us rewrite system (2) as follows

$$\begin{cases} v_{1,1}t_{1,1} + v_{1,2}t_{1,2} + ... + v_{1,n}t_{1,n} = T_1; \\ v_{2,1}t_{1,1} + v_{2,2}t_{1,2} + ... + v_{2,n}t_{1,n} = \alpha_2 T_2; \\ \dots\dots\dots\dots\dots\dots \\ v_{m,1}t_{1,1} + v_{m,2}t_{1,2} + ... + v_{m,n}t_{1,n} = \alpha_m T_m; \end{cases} \quad (5)$$

After the addition of all equations of system (5) as well as the reduction taking (3) into account we shall get

$$\xi_1 t_{1,1} + \xi_2 t_{1,2} + ... + \xi_n t_{1,n} = T_1 + \alpha_2 T_2 + ... + \alpha_m T_m. \quad (6)$$

Let us divide the right and the left parts of equation (6) by the left part of the same equation

$$1 = \frac{T_1 + \alpha_2 T_2 + ... + \alpha_m T_m}{\xi_1 t_{1,1} + \xi_2 t_{1,2} + ... + \xi_n t_{1,n}}. \quad (7)$$

Based on (4) and (5) the following equation holds true

$$\frac{1}{\alpha_i}\left(\xi_1 t_{1,1} + \xi_2 t_{1,2} + ... + \xi_n t_{1,n}\right) = \xi_1 t_{i,1} + \xi_2 t_{i,2} + ... + \xi_n t_{i,n}.$$

But

$$\xi_1 t_{i,1} + \xi_2 t_{i,2} + ... + \xi_n t_{i,n} = T^i_{\text{tot}}, \quad (8)$$

where T^i_{gen} is the time время to solve total amount of ICT s of all types at the i-th WS.

Then after simple transformation of (7) with consideration of (8) we shall get

$$1 = \frac{T_1}{T^1_{\text{tot}}} + \frac{T_2}{T^2_{\text{tot}}} + ... + \frac{T_m}{T^m_{\text{tot}}}. \quad (9)$$

As the distribution of all M ICTs by CS WSs is unspecified, then the variables $T_i \left(i = \overline{1,m}\right)$ are in general the variables taking the values in accordance with different variants of control of the computation process (loading of WS computation capacities with the tasks to be solved). Therefore, equation (9) is the expression of a hyperplane in segments [5, 6]:

$$1 = \frac{x_1}{T^1_{\text{tot}}} + \frac{x_2}{T^2_{\text{tot}}} + ... + \frac{x_m}{T^m_{\text{tot}}}. \quad (10)$$

Then hyperplane (10) determines the set of points corresponding to the variety of all possible distributions M of ICTs by m of CS WSs, and for any $x_i \left(i = \overline{1,m}\right)$, satisfying equation (10), the following inequation should be followed

$$0 \leq x_i \leq T^i_{\text{tot}},\ i = \overline{1,m}.$$

For descriptive reasons such plane is shown in Figure 2 for CS with 3 WSs.

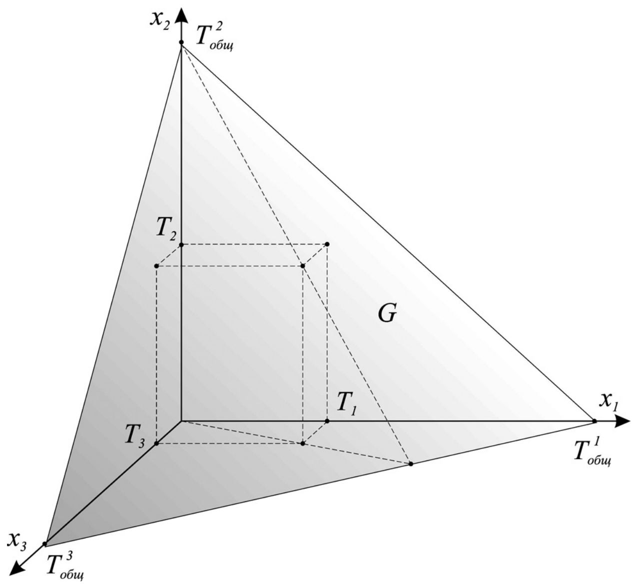

Fig. 2. Plane describing all possible distributions of ICTs by 3 WSs

Evidently, an arbitrary point of the hyperplane G can be assigned with a certain set R of possible variants of distri-

Table 1. Results of calculation of optimal parameters of reliability of the elements ensuring reliability of the system and initial data for the determination of allowable errors in calculation of derivatives.

No. of element	*T* = 2000			*T* = 3000			*T* = 4000			*T* = 5000		
	P = 0,9	*P* = 0,95	*P* = 0,99	*P* = 0,9	*P* = 0,95	*P* = 0,99	*P* = 0,9	*P* = 0,95	*P* = 0,99	*P* = 0,9	*P* = 0,95	*P* = 0,99
1	1,29	1,7	3,18	1,86	2,51	4,82	2,44	3,32	6,33	3,03	4,14	7,97
2	1,29	1,7	3,18	1,86	2,51	4,82	2,44	3,32	6,33	3,03	4,14	7,97
3	1,42	1,98	3,84	2,2	3,03	5,88	3	4,09	7,74	3,81	5,15	9,77
4	1,39	1,89	3,59	2,08	2,84	5,46	2,77	3,79	7,18	3,47	4,73	9,05
5	1,25	1,62	3	1,74	2,33	4,52	2,24	3,06	5,92	2,75	3,79	7,45
6	1,76	2,57	5,47	2,75	3,93	8,35	3,73	5,27	10,95	4,72	6,63	13,82
7	1,61	2,32	4,95	2,4	3,43	7,51	3,16	4,53	9,81	3,9	5,63	12,35
T_{av}	1,43	1,97	3,89	2,13	2,94	5,91	2,83	3,91	7,75	3,53	4,89	9,77
Maximum deviation of parameter from an optimal value, %												
Σ_n= 30 %	4,6	3,3	4,3	4,1	4,9	1,1	4,6	4,9	4,3	5,0	4,2	3,4
Σ_n= 50 %	10,1	10,4	9,3	12,5	13,9	6,6	3,9	8,2	7,7	12	10,5	5,4
Minimum values of components of the vector of derivatives μ_k (*10^{-3})												
μ_k min	3,8	1,86	0,31	3,35	1,52	0,2	3,84	1,27	0,15	2,63	0,97	0,12

bution of ICTs by WSs. In Figure 2 T_1, T_2, T_3 is the time necessary to solve several ICTs distributed to the 1-st, 2-nd and 3-d WS respectively taking (3) into account.

Let us go back to the optimality criterion related to the time necessary to solve ICTs in the CS. Let us assume the time necessary to solve a set of ICTs in CS to be the time interval between the start of a computation process and its termination at all the WSs. Let us write this definition in form of the following mathematical equation

$$T_S = \max \{T_i\}_{i=1}^{m}, \tag{11}$$

where T_s is the total time necessary to solve ICTs in CS; T_i is the time necessary to solve the ICTs distributed to the i-th WS; m is the amount of CS WSs. Then taking (11) into account, the minimum time necessary to solve ICTs in CS shall be achieved when $\forall T_i \left(i = \overline{1,m}\right)$, the equation will be followed

$$T_1 = T_2 = ... = T_m. \tag{12}$$

Based on (10) and (12) the minimum time necessary to solve ICTs in CS will be

$$T_{\min} = \frac{1}{\sum_{i=1}^{m} \frac{1}{T_{\text{tot}}^{i}}}. \tag{13}$$

Equation (13) holds true when the linearity condition is satisfied (1). Let us consider the case then the linearity condition will be satisfied within one type of ICTs, and not satisfied for all types of ICT. It becomes possible when the time necessary to solve ICT at WS includes the time of bringing of the initial information, or when not only computing tasks are being solved, but also graphic and information tasks, i.e. the tasks which are essentially different.

In this case the following equation holds true

$$\frac{t_{i,1}}{t_{j,1}} \neq \frac{t_{i,2}}{t_{j,2}} \neq ... \neq \frac{t_{i,n}}{t_{j,n}}, \tag{14}$$

where $t_{i,k}$ is the time necessary to solve one task of the k-th type at the i-th WS; $t_{j,k}$ is the time necessary to solve one task of the k-th type at the i-th WS ($i = \overline{1,m};\ j = \overline{1,m};\ i \neq j;\ k = \overline{1,n}$). Thus we see that the linearity was observed in relation to all types of tasks in (1), but in (14) it found only in relation to one arbitrary type of tasks and not found in relation to all the tasks. Such CS shall be considered as a non-homogeneous computing environment. Let us show it on a graphical example for two WSs with the numbers i and j.

The relations expressed by (1) and (14) are the slope ratio of the straight lines [7], specifying the functional dependence of times T_i and T_j of the variants of distribution of tasks between the i-th and the j-th WSs, for instance, in Figure 2 it is a segment, connecting the points $T_{\text{tot}}^{i}, T_{\text{tot}}^{j}$ with $i = \overline{1,3};\ j = \overline{1,3};\ i \neq j$. Then let us write (14) as follows

$$\text{tg}\alpha_1 \neq \text{tg}\alpha_2 \neq ... \neq \text{tg}\alpha_n,$$

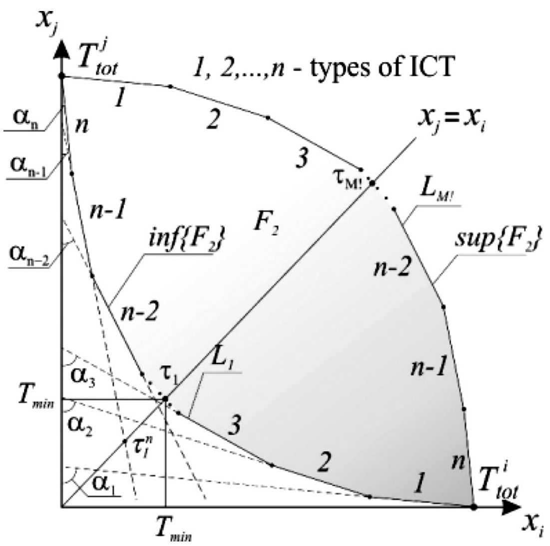

Fig. 3. Graph of change of the time necessary to solve ICTs distributed between the i-th and the j-th WSs depending on sequence of the task transfer

where α_k is a slope of the k-th straight line to the axis x_j. Let the control of the computation process in the network be such, so that all M of tasks of m types are assigned to the i-th WS, after which they are transferred in a certain sequence to the j-th WS. Then the dependence of time necessary to solve the whole set of tasks distributed between the i-th and the j-th WSs shall be represented by a broken line that connects the points T^i_{tot} and T^j_{tot}, and the form of this broken line will depend on the sequence in which the tasks will be transferred from the i-th WS to the j-th WS. Geometric interpretation of the latter thesis is shown in Figure 3.

It is clear that the amount of variants for the distribution of ICTs between WSs (amount of broken lines) shall be equal to the number of shifts of the total amount of tasks, i.e. $M!$ that will be inside the area F_2 (Figure 3) ("2" in F_2 is the area dimensions). Broken lines L_1 and $L_{M!}$ (Fig.3) are the lower and the upper bounds of the area F_2, i.e.

$$\begin{aligned} &L_1\left\{|\text{tg}\,\alpha_1| > |\text{tg}\,\alpha_2| > \ldots > |\text{tg}\,\alpha_{n-1}| > |\text{tg}\,\alpha_n|\right\} = \inf\{F_2\} \\ &L_{M!}\left\{|\text{tg}\,\alpha_1| < |\text{tg}\,\alpha_2| < \ldots < |\text{tg}\,\alpha_{n-1}| < |\text{tg}\,\alpha_n|\right\} = \sup\{F_2\} \end{aligned}, \quad (15)$$

where $L_1\left\{|\text{tg}\,\alpha_1| > |\text{tg}\,\alpha_2| > \ldots > |\text{tg}\,\alpha_{n-1}| > |\text{tg}\,\alpha_n|\right\}$ and $L_{M!}\left\{|\text{tg}\,\alpha_1| < |\text{tg}\,\alpha_2| < \ldots < |\text{tg}\,\alpha_{n-1}| < |\text{tg}\,\alpha_n|\right\}$ are the broken lines consisting of the variety of segments $\{l_i\}_{i=1}^n$, for which the following inequations hold true

$$\begin{aligned} &|\text{tg}\,\alpha_1| > |\text{tg}\,\alpha_2| > \ldots > |\text{tg}\,\alpha_{n-1}| > |\text{tg}\,\alpha_n| \text{ и} \\ &\text{и } |\text{tg}\,\alpha_1| < |\text{tg}\,\alpha_2| < \ldots < |\text{tg}\,\alpha_{n-1}| < |\text{tg}\,\alpha_n|, \end{aligned}$$

where $\text{tg}\alpha_j$ is a slope ratio of the straight line, on which the segment l_j lies.

We can see from Figure 3 that for any $L_i \in F_2$, $i = \overline{1, M}!$ there is a point τ_i, which shall fulfill condition (12), but it is evident that

$$T_{\min} = \tau_1 \in \inf\{F_2\}. \quad (16)$$

Therefore $\inf\{F_2\}$ will determine the sequence of ICTs transfer between two WSs, under which the condition (16) will be valid. Let us consider the algorithm of evaluation of $T_{\min}$ in accordance with criterion (12) for $\inf\{F_2\}$, as it is this broken line that will set an optimal in sense of criterion (12) and condition (16) functional dependence of change of the time necessary to solve ICTs at their distribution between WSi and WSj. Without loss of generality let us consider the numbers of the tasks' types corresponding to the segments l_j, to correspond to the numbers α_j in definition (15).

Let all the ICTs entering the system be allocated at the WSi, then the time necessary to solve them shall be T^i_{tot}. Let us start to transfer the ICT of the 1st type from the WSi to the WSj. Then the dependence of change of the time necessary to solve ICTs at the redistribution between WSs will be expressed by the segments l_1, lying on the straight line represented by the equation

$$f(x_i) = x_j = a \cdot x_i + b,$$

where $a = \text{tg}\alpha$; α is a slope of the straight line in relation to the coordinate axis; b is an absolute term, whose value depends on the values of the coordinates of the points x_i and x_j on the plane crossed by a straight line.

For the segment l_1 $a=\text{tg}\alpha_1$ and $x_i = T^i_{\text{tot}} = \left(T^i_n + T^i_{n-1} + \ldots + T^i_2 + T^i_1\right)$, $x_j = 0$, then for l_1 the following equation holds true (see Figure 3)

$$\begin{aligned} f_1(x_i) &= x_j = \text{tg}\,\alpha_1 \cdot \left(x_i - T^i_{\text{tot}}\right) = \\ &= \text{tg}\,\alpha_1 \cdot \left(x_i - \left(T^i_n + T^i_{n-1} + \ldots + T^i_2 + T^i_1\right)\right), \end{aligned}$$

where T^i_j is the time necessary to solve all tasks of the j-th type at the WSi. Then for an arbitrary segment $l_k \in \inf\{F_2\}$ it is not difficult to form a straight line for which $a=\text{tg}\alpha_k$, $x_i = \left(T^i_n + T^i_{n-1} + \ldots + T^i_k\right)$ and $x_j = \left(T^j_1 + T^j_2 + \ldots + T^j_{k-1}\right)$, we shall have the following expression of the straight line

$$\begin{aligned} f_k(x_i) &= x_j = \text{tg}\,\alpha_k \cdot x_i + \\ &+ \left(\left(T^j_1 + T^j_2 + \ldots + T^j_{k-1}\right) - \text{tg}\,\alpha_k \cdot \left(T^i_n + T^i_{n-1} + \ldots + T^i_k\right)\right), \end{aligned}$$

where $(T^j_1 + T^j_2 + \ldots + T^j_{k-1})$ is a total time necessary to solve the tasks from the 1-st to the (k–1)-th type at the WSj (with k=1 $(T^j_1 + T^j_2 + \ldots + T^j_{k-1}) = 0$); $(T^i_n + T^i_{n-1} + \ldots + T^i_k)$ is a total time necessary to solve all the tasks from the k-th to the n-th type at the WSi.

According to criterion (12) for each straight line $f_k(x_i)$ let us find the point $\tau^k_1 \left(k = \overline{1, n}\right)$, which is a crossing of the straight line $f_k(x_i)$ and the straight line $x_j = x_i$

$$\tau^k_1 = \frac{\left(T^j_1 + T^j_2 + \ldots + T^j_{k-1}\right) - \text{tg}\,\alpha_k \cdot \left(T^i_n + T^i_{n-1} + \ldots + T^i_k\right)}{1 - \text{tg}\,\alpha_k}.$$

Further on for all values of $\tau^k_1 \left(k = \overline{1, n}\right)$ it is necessary to choose the maximum values corresponding to $T_{\min}$, i.e.

$$T_{\min} = \max\left\{\tau^k_1\right\}_{k=1}^n. \quad (17)$$

Figure 3 shows that for two WSs, the equation $T_{\min}$ divides the whole set of tasks into two subsets. The first subset are the tasks to be solved at the WSi, the second subset includes the tasks to be solved at the WSj. By analogy with the computation process at two WSs, the task of estimation of the value $T_{\min}$ for m WS is reduced to the construction of $\inf\{F_m\}$, which shall be a piecewise linear hypersufrace. Analytically, as in the case of (10), it is difficult to represent this hypersurface, that is why it is offered to used a geometrical method to construct it.

Let us consider the algorithm of $\inf\{F_m\}$ on the example for m=3 and n=4. Let us assume all ICTs to be distributed

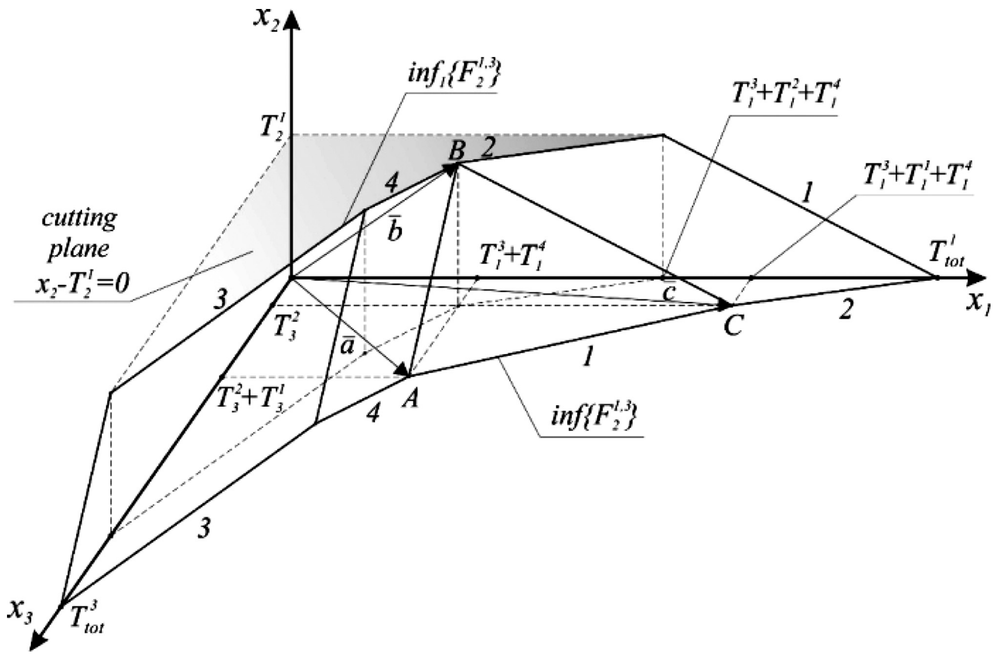

Fig. 5. Fragment of the piecewise linear surface *Q*

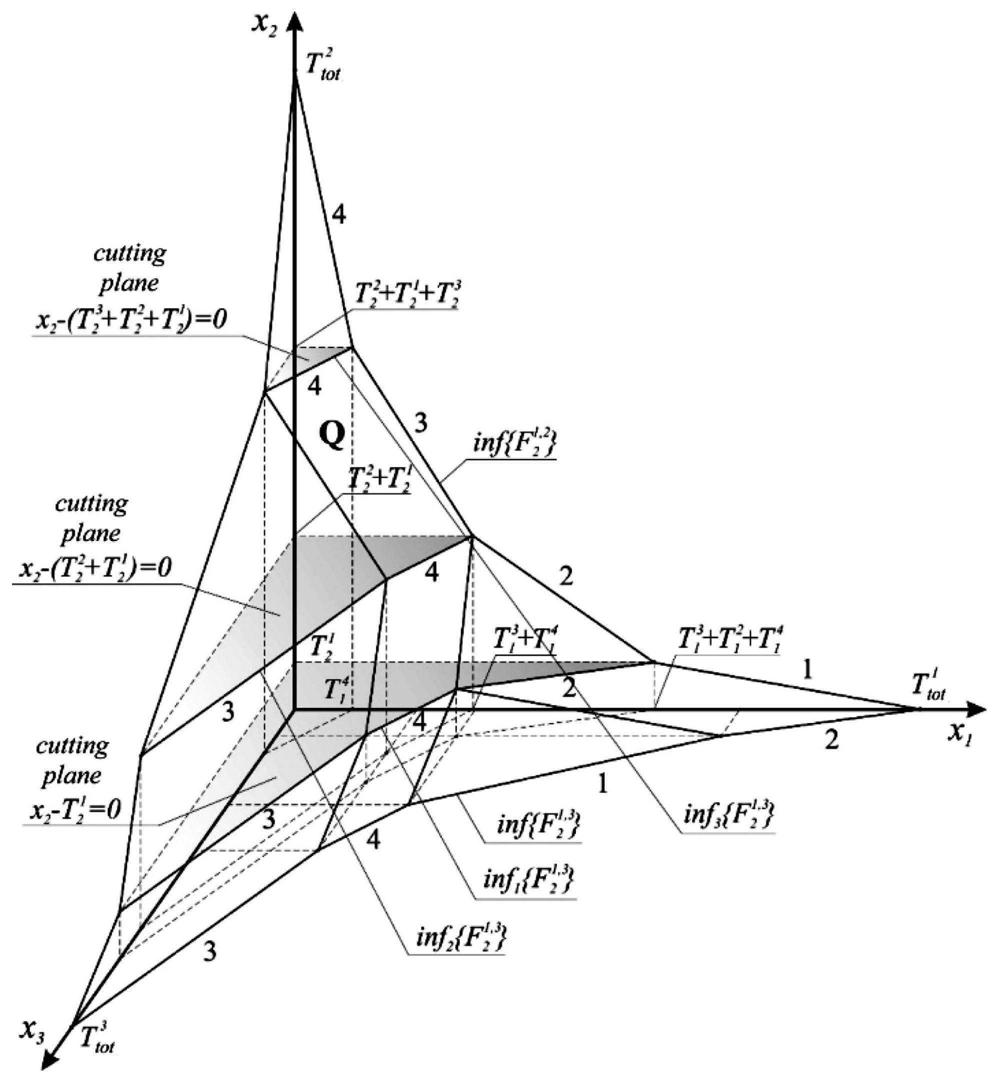

Fig. 4. Piecewise linear surface *Q* describing the variety of possible variants for the distribution of ICTs between three WSs

to the WS1. Then the time necessary to solve all the ICTs at the WS1 shall be equal to T^1_{tot}. There is a potential to transfer the ICTs from the WS1 to the WS2 and WS3, and the dependence of the change of the solution time whole transferring the ICTs from the WS1 to the WS2 will correspond to $\inf\{F_2^{1,2}\}$, and while transferring the tasks from the WS1 to the WS3 – $\inf\{F_2^{1,3}\}$. In general the sequences of ICT transfer that determine $\inf\{F_2^{1,2}\}$ and $\inf\{F_2^{1,3}\}$, may be different. Without loss of generality let us accept that the sequences of ICT transfer that determine $\inf\{F_2^{1,2}\}$ and $\inf\{F_2^{1,3}\}$, are such as it is shown in Figure 4.

Let us transfer the ICTs of the 1st type from the WS1 to the WS2 in accordance with the sequence defined by $\inf\{F_2^{1,2}\}$.

Then the dependence of the change of time necessary to solve the ICTs of the 1st type, distributed between WS1 and WS2, will be expressed by a linear function $x_2(x_1)=l_1$, where l_1 is the segment 1 in the plane x_1x_2, connecting the points $(T^1_{tot}, 0)$, $(T_1^3+T_1^2+T_1^3, T_2^1)$ (Figure 4), i.e. we can write that

$$l_1 \in \inf\{F_2^{1,2}\}.$$

As the result there will be the ICTs of the 2nd, 4th and 3rd types left, allocated in the sequence that fulfills condition (15) and specifies $\inf_1\{F_2^{1,3}\}$ (Figure 4).

It is evident that $\inf\{F_2^{1,3}\}$ is a trace of the piecewise linear surface Q by the plane x_2=0 and $\inf_1\{F_2^{1,3}\}$, $\inf_2\{F_2^{1,3}\}$, $\inf_3\{F_2^{1,3}\}$ are traces of the piecewise linear surface Q by the planes $x_2-T_2^1=0$, $x_2-\left(T_2^2+T_2^1\right)=0$ and $x_2-\left(T_2^3+T_2^2+T_2^1\right)=0$ respectively, parallel to the plane x_2=0 (Figure 4). Figure 4 shows that the surface Q between the traces, for instance between $\inf\{F_2^{1,3}\}$ and $\inf_1\{F_2^{1,3}\}$, are the intercrossing planes formed by parallel or by crossing segments (coplanar vectors). By the example of the plane ABC, lying between $\inf\{F_2^{1,3}\}$ and $\inf_1\{F_2^{1,3}\}$ (Figure 5), let us consider the construction of an arbitrary plane forming a piecewise linear surface Q.

We know [5, 6], that in orthonormal space the plane is definitely set by a normal vector $\overline{N}$ and by the point lying on the plane. In our case let us choose the point A, referring to the plane, with the coordinates $(T_1^3+T_1^4, 0, T_3^2+T_3^1)$ (Figure 5).

Let us consider three vectors $\overline{a}$, $\overline{b}$, $\overline{c}$, whose coordinates coincide with the points A,B,C respectively (Figure 5). As the indicated points are on the plane, any linear combination $\overline{a}$, $\overline{b}$, $\overline{c}$ will represent the set of coplanar vectors [5, 6], i.e.

$$\overline{AB}=\overline{b}-\overline{a},\quad \overline{AC}=\overline{c}-\overline{a}.$$

The normal to the plane is known to be the result of a vector product of a pair of coplanar vectors [5, 6], therefore, we can write

$$[\overline{AB}, \overline{AC}]=\overline{N}, \tag{18}$$

where $[\overline{AB}, \overline{AC}]$ is a vector product of the vectors $\overline{AB}$ and $\overline{AC}$.

Let x_1, x_2, x_3 be an orthonormal basis of the vector space, then let us rewrite the expression (18) in a coordinate form.

$$\overline{a}=(T_1^3+T_1^4,\ 0,\ T_3^2+T_3^1);\ \overline{b}=(T_1^3+T_1^4,\ T_2^1,\ T_3^2);$$
$$\overline{c}=(T_1^3+T_1^1+T_1^4,\ 0,\ T_3^2);\ \overline{AB}=\overline{b}-\overline{a}=(0,\ T_2^1,\ -T_3^1);$$
$$\overline{AC}=\overline{c}-\overline{a}=(T_1^1,\ 0,\ -T_3^1);$$
$$\overline{N}=[\overline{AB}, \overline{AC}]=\begin{vmatrix} x_1 & x_2 & x_3 \\ 0 & T_2^1 & -T_3^1 \\ T_1^1 & 0 & -T_3^1 \end{vmatrix}=x_1\cdot\begin{vmatrix} T_2^1 & -T_3^1 \\ 0 & -T_3^1 \end{vmatrix}-$$
$$-x_2\cdot\begin{vmatrix} 0 & -T_3^1 \\ T_1^1 & -T_3^1 \end{vmatrix}+x_3\cdot\begin{vmatrix} 0 & T_2^1 \\ T_1^1 & 0 \end{vmatrix}=$$
$$=-x_1\cdot T_2^1T_3^1+x_2\cdot T_3^1T_1^1-x_3\cdot T_2^1T_1^1.$$

Therefore, normal vector to the plane surface ABC shall be as follows

$$\overline{N}=(-T_2^1T_3^1,\ T_3^1T_1^1,\ -T_2^1T_1^1).$$

Canonical expression of the plane in the space with an orthonormal basis x_1, x_2, x_3, going through the point $M^0(x_1^0, x_2^0, x_3^0)$ and having the normal vector $\overline{N}=(A,B,C)$ is as follows

$$A(x_1-x_1^0)+B(x_2-x_2^0)+C(x_3-x_3^0)=0.$$

Then the expression of the plane ABC with the normal vector $\overline{N}=(-T_2^1T_3^1,\ T_3^1T_1^1,\ -T_2^1T_1^1)$ passing through the point $A\left(T_1^3+T_1^4,\ 0,\ T_3^2+T_3^1\right)$ will be as follows

$$-T_2^1T_3^1(x_1-(T_1^3+T_1^4))+T_3^1T_1^1x_2-$$
$$-T_2^1T(x_3-(T_3^2+T_3^1))=0.$$

Similar way is used to construct the rest planes forming a piecewise linear surface Q. Then for each plane of the surface Q in accordance with the criterion (12) we shall find the point of crossing of the plane with a straight line x_1=x_2=x_3. The amount of planes forming the surface Q can be found using equation [7]

$$L=\frac{2+(n-1)}{2}\cdot n,$$

where L is the amount of the planes forming the surface Q; n is the amount of types of ICTs distributed between WSs. Let us denote the point of crossing of the straight line x_1=x_2=x_3 with the k-th plane τ^k. Then by analogy with (17)

$$T_{\min}^{WS1}=\max\{\tau^k\}_{k=1}^{L},$$

where $T_{\min}^{WS1}$ is the value of minimum time necessary to solve ICTs provided all ICTs are potentially allocated at the WS1. The value $T_{\min}^{WS1}$ was obtained provided that the distribution of ICTs between WSs started with the WS1, i.e. all ICTs were virtually allocated at the WS1 and then distributed between the WS2 or WS3. If to form the surface Q provided that the distribution of ICTs starts with the WS2 of the WS3, we will have the values $T_{\min}^{WS2}$ and $T_{\min}^{WS3}$, for which the following inequation will generally be fulfilled

$$T_{\min}^{\text{APM1}} \neq T_{\min}^{\text{APM2}} \neq T_{\min}^{\text{APM3}}. \quad (19)$$

It is evident that for a complete solution of the task it is necessary to choose the minimum value from the obtained values (19). I.e., for m WS we will have a minimax task

$$T_{\min} = \min\left\{T_{\min}^{\text{WS}\,i}\right\}_{i=1}^{m} = \min\left\{\max\left\{\tau_{\text{WS}\,i}^{k}\right\}_{k=1}^{L}\right\}_{i=1}^{m}. \quad (20)$$

Therefore, having obtained (20) $T_{\min}$, which is the point of an absolute minimum of the task, through the coordinates of this point $T_1=T_2=\ldots=T_m$ corresponding to the axes $x_1, x_2, \ldots, x_m$, we will get the best (in terms of the criterion (12) variant of the organization of a computation process in CS.

References

1. Voevodin V.V., Voevodin Vl. V. Parallel computing. – SP.: BHV – Petersburg, 2002. – 608 p.

2. Voevodin V.V. mathematical models and methods in parallel processes. – M.: Science, 1986, 296 p.

3. Kozlov M.V., Malashenko Y.E., Nazarova I.A. Assured estimations of the distribution of computational resources under uncertainty. – M:, Computing Center of RAS, 2011, 46 p.

4. Ishakova E.N. Theory of computation processes: study guide. – Orenburg: OSU, 2007. -160 p.

Voevodin V.V. Linear algebra. – M.: Science, 1980, 400 p.

6. Kagan M.L., Samokhin M.V. Mathematics in an engineering higher school. Algebra and geometry. – M.:Stroyizdat, 2003, 207 p.

7. Nikolsky S.M. Course of mathematical analysis. T. 1. –M.: FISMATLIT, 1983, 448 p.

About the authors

Sergey A. Zhurbin – senior researcher in the 4th CSRI, Ministry of Defense of the Russian Federation. 13-224, Pushkinskaya street, Korolev, Moscow region, 141092, Russia. Phone +7 (910) 408-97-04;

e-mail: serdgo4@yandex.ru.

Gennady V. Kazakov – Ph.D. in engineering, associate professor, director of the 4th CSRI, Ministry of Defence of the Russian Federation. 17-103, Pushkinskaya street, Yubileinyi, Moscow region, 141092, Russia. Phone +7 (910) 467-71-66; e-mail: kgv.64@mail.ru.