UDC 681.3.06

DESIGNING LOCAL NETWORK STRUCTURE FOR DISTRIBUTED COMPUTER SYSTEM OF REAL TIME

A.V. Pogrebnoy, D.V. Pogrebnoy

TPU Institute «Cybernetic centre»
E-mail: Sasha@ad.cctpu.edu.ru

The task of designing local network structure in distributed computer system of real time has been formulated. The method of problem solving is suggested. It includes construction operations of data transmission graph between network stations, conflict presence matrices at the access to the network backbones, diagrams of combining parallel data transmission. Method description is accompanied by explanations with the examples.

Introduction

Multiprocessor computer system for controlling objects with geographically distributed equipment is examined. The system refers to a class of real-time systems with rigid constraints on computing time of control actions and constructed on the basis of typical stations (controllers) combined into a local network. Tendency to reduce system reaction time on input actions is a dominant criterion for all stages of designing such systems. The main time delays in the system fall on operation of station processors at performance of technological algorithm program modules implementing system applied functions and on data transfer between stations in local network.

Requirement in data transfer in the network occurs whenever resources of more than one station are used at of program module performance. For example, module is performed by one station processor and module program and required data are saved in other station memories. Time expenditures for data transfer between the network stations may turn out to be comparable with processor operation time at module performance. Therefore, designing computer system along with determining a number of stations capable of fulfilling applied functions at set time it is important to construct system local network and plan of using its resources so that time delays for data transfer are reduced.

Statement of problem of network structure construction

The problem of network construction is solved in conditions when minimally required number of stations for the network is determined [1] and algorithms of fulfilling applied functions being the main part of program load on computer system are presented in the form of information graph constructed for the model of program load [2]. Information graph $G=(D,F,R)$ is a bipartite weighted graph where D is the set of vertices of data $D=\{d_q\}$ with indication for each $d_q \in D$ the dimension of required memory P_q, $q=1,2,...,Q$; F is the set of vertices of modules $F=\{f_m\}$ with indication for each $f_m \in F$ the value of consumed processor time T_m, $m=1,2,...,M$; $R=\|r_{qm}\|_{Q\times M}$ is the matrix of volume of the data transferred between vertices of graph G.

On graph G cutting [3] into a set of subgraphs $\{G_i\}$, $i=1,2,...,n$ is specified. A number of subgraphs n corresponds to a number of computer system stations. Vertices of subgraph G_i by required memory P_q and processor time T_m do not totally exceed resources of station s_i by memory $P(s_i)$ and processor time $T(s_i)$. Value $T(s_i)$ equals to a number of slots which processor of station s_i may assign for fulfilling modules of subgraph G_i per one simulation cycle.

A set C of arcs of graph G, $C=\cup C_{ij}$ corresponds to cutting $\{G_i\}$ where c_{ij} is the set of arcs relating subgraphs G_i and G_j. Element r_{qm} of matrix R corresponds to each arc c_{qm}, connecting vertex d_q and f_m in graph G. Therefore data volume transferred in the network between stations s_i and s_j may be determined by value r_{ij},

$$r_{ij} = \sum_{c_{qm} \in C_{ij}} r_{qm}. \qquad (1)$$

Thus, total volume of data transferred in the network per one simulation cycle for cutting $\{G_i\}$ amounts to value r,

$$r = \sum_{i,j} r_{ij}. \qquad (2)$$

If network backbone capacity is denoted by value φ defining data volume transferred per one simulation slot then for successful operation of local network constructed on the basis of one backbone the condition should be fulfilled:

$$(r/\varphi)k_\varphi \le \mu. \qquad (3)$$

Here μ is the number of slots in one simulation cycle; k_φ is the coefficient accounting factor of decreasing value φ in real network.

It is obvious that it is impossible to solve the problem of timely data transfer in the network decreasing number μ as the value r also depends on μ. Therefore, if the condition (3) is not fulfilled then it means that the network on one backbone with parameter φ is not operable and decision should be made on decreasing value r or increasing the magnitude of parameter φ. Among such decisions may be the following ones:

- to find another cutting with lower value r;
- to construct network on the basis of backbone with higher capacity;
- to select another structure of network constructed on the basis of several backbones.

Let us consider that two first types of decisions are exhausted as well as many others, connected, for example, with a change of information graph, conditions of arrival input and update of output data with other changes in program load. Let us examine decisions on selection of network structure constructed on the basis of several backbones with constant value φ, capable of transferring data volume r per a simulation cycle. Before describing the method of solving this problem let us give a number of examples of combining several backbones into network and conditions of their load at data transfer.

Analysis of basic variants of networks

A number of backbones in the network may be roughly determined on the basis of ratio $(r/\varphi)k_{\varphi}\leq\mu$. Let us round up the obtained result and take it as minimally possible number of backbones in the network. Let us construct the variants of network structures with different amount of backbones. Let us consider the variants of network structures to be basic ones and develop library for their saving. Basic networks reflect different configurations of several backbone bonds. The examples of four basic networks are given in Fig. 1.

Each example of basic network is attended by diagram which illustrates possible load of network backbones at data transfer between stations. In this case it is supposed that stations are connected to each network backbone and data are transferred both between the stations of one backbone and between the stations of different backbones. Length of sections reflecting the volumes of data transferred are taken arbitrarily. A list of backbones participating in data transfer is given over each section. For example, note 1-4-2 over diagram sections (Fig. 1, *г*) indicates load of backbones *M*1, *M*4, *M*2 at data transfer between stations connected to backbones *M*1 and *M*2. Let us notice as well that parallel with this transfer the data may be transferred between the stations connected to backbone *M*3 that is section 3 on a diagram Fig. 1, *г*, may be arranged parallel to sections 1-4-2.

The analysis of basic networks given in Fig.1 allows making a number of conclusions by appropriate diagrams. In network (*a*) stations should be connected to backbones *M*1 and *M*2 so that sections 1 and 2 are roughly equal and section 1-2 is of minimal length. For the network (*б*) backbone *M*2 is critical by load. In network (*в*) *M*2 and *M*3 are such backbones in network (*г*) backbone *M*4. The best variant of station connection to backbones for these networks is the variant which supports equal and minimal backbone load. For example, for network (*б*) the condition of equality of backbone load may be written down in the form: [1]+[1-2]=[2]+[1-2]+[2-3]=[3]+[2-3]. Here square brackets mean the length of appropriate sections. Similar conditions of backbone load equality may be written down for networks (*в*) and (*г*). Minimum backbone load is achieved in the case if station connection to backbones is managed to be performed so that data are not transferred between the stations connected to different backbones.

Method of solving the problem of network construction

Describing the method of solving the problem of selection of basic network and the variant of station connection to backbones let us follow the example of infor-

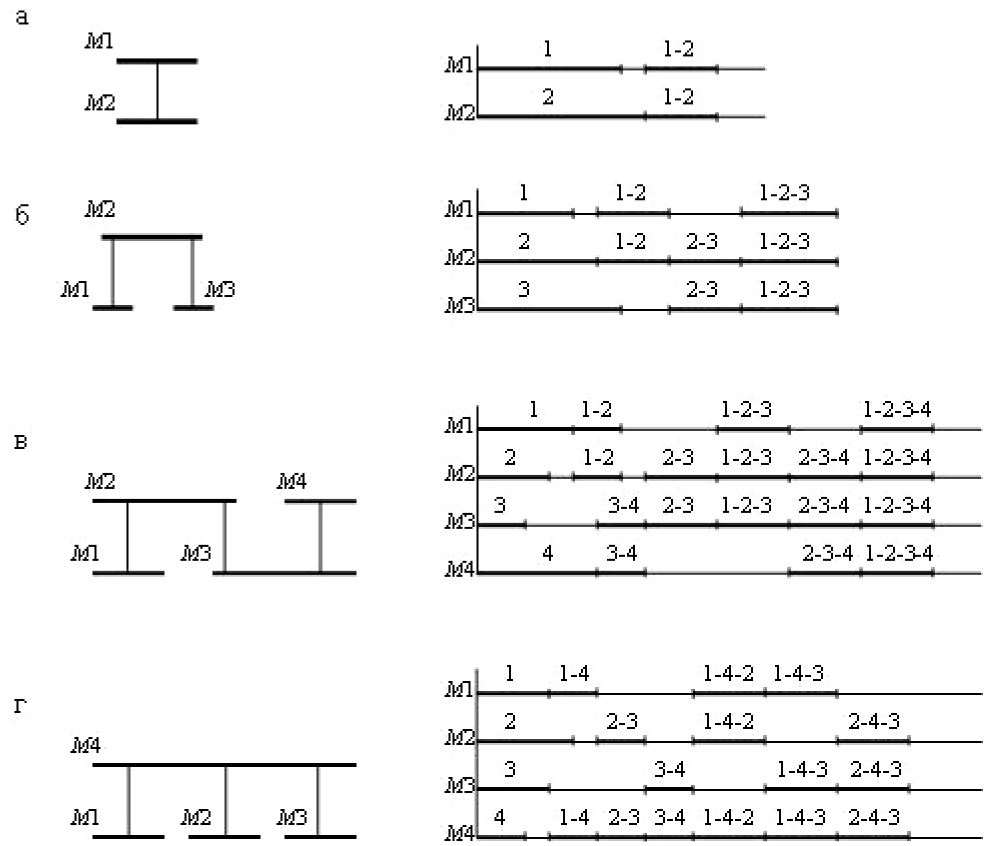


Fig. 1. *Examples of basic networks and diagrams of backbone load*

mation graph given in Fig. 2. Here data d_q, are shown by circles and modules f_m are shown by bars. At each arc connecting vertices d_q and f_m weights r_{qm} equal to data volume transferred between vertices d_q and f_m per one simulation cycle are given [2]. Subgraphs of cutting are marked out by dashed lines and numbers of stations resources of which occupy these subgraphs are given.

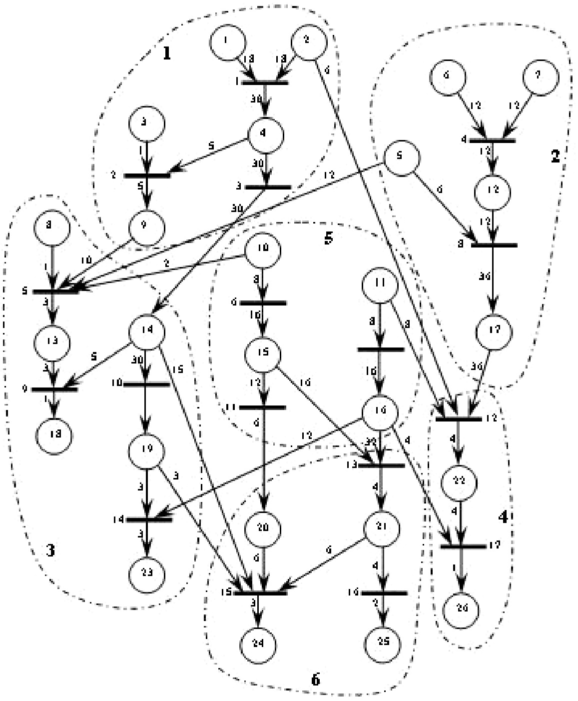


Fig. 2. *Example of information graph*

Let backbone capacity amount to 10 units of volumes of data transferred per 1 slot that is value φ=10, coefficient k_φ=1,3 and simulation cycle μ equals 12 slots. Then it is possible to determine approximate number of network backbones for the accepted variant of module and data distribution over stations. For this purpose on the basis of matrix R according to (2) total volume of data r transferred between stations is calculated. For this example r=198 units. Time for data transfer amounts to $(r/\varphi)k_\varphi$=(198/10)1,3=25,74 slots that exceeds twice the simulation cycle. Thus, not less than three backbones should be in the network.

On the basis of available information on station amount, distribution of information graph modules over them, network total load and references on a number of backbones the task of selection of network structure may be stated in the following way. It is necessary to select the structure of basic network and variant of station connection to network backbones so that major part of data, if is possible, may be transferred parallel between network stations.

The method of solving the problem is based on detection of possibilities of parallel data transfer in the network structure for various variants of station connection to basic network backbones. For this purpose a set of operations on constructing the following objects is fulfilled:

- graph of data transfer between network stations;
- matrix of presence of clashes at access to network backbones;
- diagram of overlapping parallel data transfer.

Graph of data transfer P=(S,Z,R) is constructed on the basis of variant of distribution of modules and data of graph G over stations and matrices of weights $R=\|r_{qm}\|$. Vertex $s_i \in S$ of graph P corresponds to station s_i, to which modules and data of subgraph G_i of cutting $\{G_i\}$ are distributed. Presence of arc $z_{ij} \in Z$ corresponds to the fact that subgraphs G_i and G_j are connected to each other by arcs of information graph $c_{qm} \in C_{ij}$ that is $C_{ij} \neq \varnothing$. Value r_{ij} which is calculated by the expression (1) and determines the volume of data transferred between stations s_i and s_j per one simulation cycle corresponds to each arc z_{ij} of graph P.

The example of graph P constructed for cutting shown in Fig. 2 is given in Fig. 3. Graph P contains 6 vertices s_1–s_6 by a number of stations.

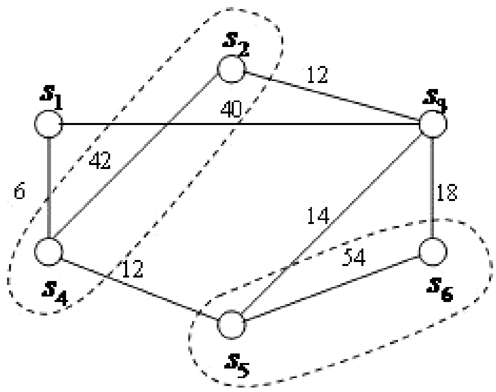


Fig. 3. *Graph of data transfer P*

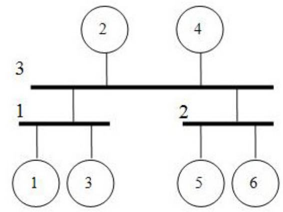
Weights r_{ij} of arcs z_{ij} given in Fig. 3 are obtained by the expression (4) and amounts to 198 units in sum.

Matrix of clash presence Q is constructed on the basis of graph P and variant of station connection to backbones of selected basic network. Let us show the technique constructing matrix Q on the examined example. Let us select a variant with three backbones given in Fig. 1, *б* as a basic network structure. One of the variants of stations connection to the backbones of this network is shown in Fig. 4, *a*.

Dimension of matrix Q is determined by a number of arcs of graph P. Let us denote a set of arcs ($z_{1,3}$, $z_{1,4}$, $z_{2,3}$, $z_{2,4}$, $z_{3,5}$, $z_{3,6}$, $z_{4,5}$, $z_{5,6}$) of graph P by a proper code numbers saving numbers of stations in them. We obtain a set of code numbers of arcs (13, 14, 23, 24, 35, 36, 45, 56) and, respectively, numbers of lines and columns of matrix Q. So, for example, the number of arc 24 indicates a presence of data transfers between the stations 2 and 4 with volume r_{24}=36 units. Element q_{vk} of matrix of clash presence $Q=\|q_{vk}\|$, $v,k \in$(13,14,23,24,35,36,45,56) is determined in the following way: q_{vk}=1, if pairs of stations of arcs v and k have clashes in access to backbone at data transfer in the network (Fig. 4, *a*); q_{vk}=0, if not.

So, or example, element $q_{24,45}$=1 as at simultaneous data transfer between stations s_2 and s_4 and stations s_4

а



б

	13	14	23	24	35	36	45	56
13		1	1		1	1		
14	1		1	1	1	1	1	
23	1	1		1	1	1	1	
$Q=$ 24		1	1		1	1	1	
35	1	1	1	1		1	1	1
36	1	1	1	1	1		1	1
45		1	1	1	1	1		1
56					1	1	1	

Fig. 4. *Variant of network: a) variant of station connection to the basic network; б) matrix of clash presence Q*

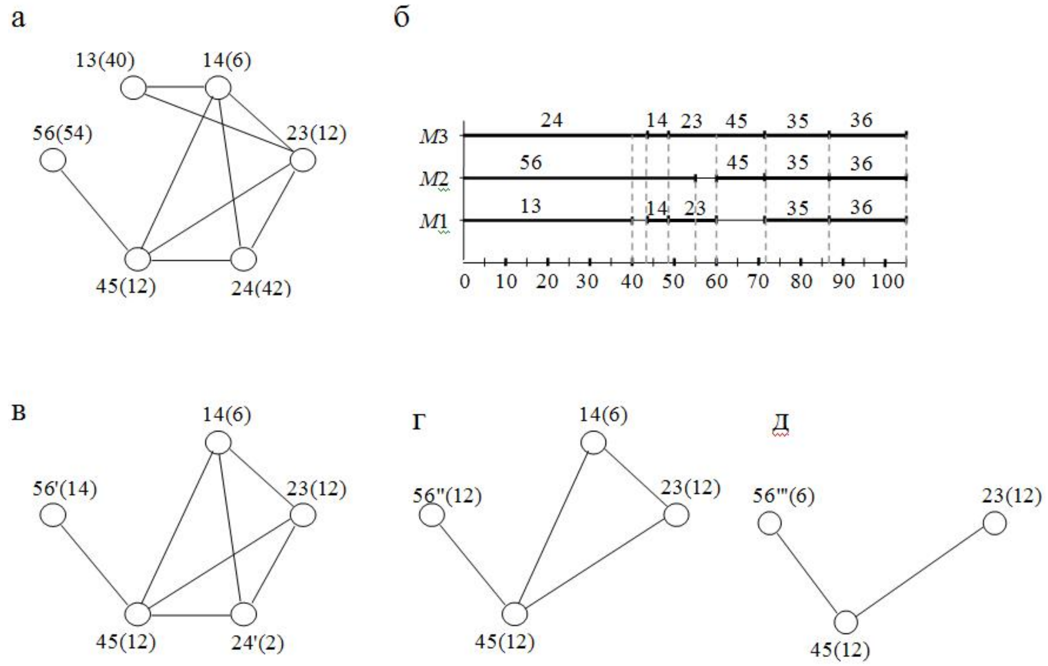


Fig. 5. *Construction of overlap diagram: a) graph Q; б) diagram of overlapping data transfer; в-д) conversion of graph Q at diagram construction*

and s_5 the clash for access to backbone occurs. On the contrary, element $q_{56,13}$=0 as there is no clash for access to the backbone at simultaneous data transfer in pairs of stations s_5,s_6 and s_1,s_3. It is explained by the fact that different backbones are used in this case. Matrix Q constructed in this manner is given in Fig. 4, *б*.

Lines and columns 35, 36 in matrix Q are marked out in Fig. 4, *б*. They differ in the fact that all their elements q_{vk}=1, $v\neq k$. It means that at data transfer between the appropriate stations, for example, for line 35 they are stations s_3 and s_5 all three backbones are occupied and data can not be transferred between stations in all other pairs parallel with pair s_3, s_5. Therefore, the marked out lines and columns may be excluded of matrix Q.

On the basis of matrix Q the diagram of overlapping parallel data transfer is constructed. For convenience of diagram construction matrix Q is taken as a matrix of connection of graph vertices. An appropriate graph Q is given in Fig. 5, *a*. Vertices 35, 36 were not included into graph Q by the above mentioned reason. There is no sense in including these vertices into graph Q as all network backbones are captured at data transfer in proper pairs of stations and, therefore, data transfers are not possible in other pairs of stations parallel with the given ones. Data volumes transferred between the appropriate stations are given in brackets near vertices of graph Q.

To construct diagram in graph Q maximal empty subgraphs are singled out serially [3] and volumes of transfers of appropriate vertices are overlapped on diagram (Fig. 5, *б*). So, maximal empty subgraph is singled out at the first stage, for example, with vertices 13, 56, 24 and appropriate volumes of 40, 54 and 42 units may be parallel transferred in network and, therefore, they are overlapped on diagram. Vertices with minimal volume of data transfer are excluded of graph Q. In this case it is vertex 13 and vertices 56′ and 24′ are saved in graph with new volumes of 14 and 2 units (Fig. 5, *в*) and marked with bars. Forming the next maximal empty subgraph of a vertex those marked with bars are selected in the first place. Vertices 56′ and 24′ are selected for graph in Fig. 5, *в*. Vertex 24′ is excluded and the process continues for the graph in Fig. 5, *г*. Here vertices 56″ and 14 are selected. Vertex 14 is excluded and for graph in Fig. 5, *д*, maximal empty subgraph includes vertices 56″ and 23. Both of them are serially excluded of graph and the rest vertex 45 is reflected on a diagram.

Let us assume that constructing diagram (Fig. 5, *б*) volumes of transferred data for each vertex are reflected on all backbones participating in these data transfer. Diagram construction is completed by reflection of volumes of transfers for vertices 35 and 36 marked in matrix Q and excluded of graph Q.

It follows form the diagram that total volume of data transfer in the network taking into account their overlaps amounts to 104 units or $104/(\varphi=10)=10{,}4$ of slots that is included in simulation slot of 12 slots. However, in this case coefficient k_φ achieves only the value 1,15. Total volume of data transfers according to graph P amounts to 198 units. Thus, series circuit of data transfer for the selected local network and variant of station connection Fig. 4, *a* is reduced from 198 to 104 units due to the use of parallel transfers. Therefore, time of transfers is reduced from 19,8 to 10,4 slots. Estimating decrease of transfer time it should be thought that a gain in 9,4 slots is reduced by a value of delays in adapters connecting network backbones.

Conclusion

The transfer from a plan of using resources obtained at solving the problem of distribution of modules and information graph data over the stations of computer system to the task of selecting structure of its local network with minimal time expenditures for data transfer was managed to be formalized.

The result obtained at solving the problem of selecting network structure should be considered in general case as one of possible. Really, if station connection to backbones is changed in the obtained network then clash matrix and diagram, respectively will be changed. To find the best variant of network structure giving the greatest overlap of parallel data transfer it is necessary to search the majority of acceptable basic networks and form and estimate a set of variants of station connection for them. Forming variants of station connection the decisions are made on the basis of analysis of data transfer graph and basic network structure.

The experiments showed that for networks on the basis of 2–4 backbones at connection up to 10 stations the search of basic networks and the best variants of station connection using the stated rules does not result in great volumes of calculations. At further increase of network dimension it is necessary to develop, along with suggested rules, the additional more efficient rules of elimination of unpromising network variants comparing arc weights of data transfer graph and clash graph structure of basic network.

REFERENCES

1. Pogrebnoy A.V. Determining the number and topology of arrangement of stations of multiprocessor computer system // Bulletin of the Tomsk Polytechnic University. – 2006. – V. 309. – № 7. – P. 160–164.
2. Pogrebnoy A.V. Determining data transfer volumes in computer system network for specified model of program load // Bulletin of the Tomsk Polytechnic University. – 2007. – V. 310. – № 3. – P. 103–107.
3. Kristofides N. Theory of graphs. Algorithm approach. – Moscow: Mir, 1978. – 432 p.

Received on 31.10.2007