DOI: 10.15439/2019F356
ISSN 2300-5963 ACSIS, Vol. 19

A Heterogeneous Parallel Processing System Based on Virtual Multi-Bus Connection Network

Piotr Hajder, Łukasz Rauch
AGH University of Science and Technology, Krakow, Poland
Email: {phajder, lrauch}@agh.edu.pl

Mariusz Nycz
Rzeszow University of Technology, Rzeszow, Poland
Email: mnycz@prz.edu.pl

Mirosław Hajder
University of Information Technology and Management
Rzeszow, Poland
Email: miroslaw.hajder@gmail.com

Abstract—**This work presents organization, architecture and synthesis as well as analysis of the reconfigurable heterogeneous parallel processing system. Reconfiguration takes place on two levels of the connection network: physical and logical. For its implementation, passive multi-channel optical networks were used. Due to its dynamic nature, the system is designed to handle computational and communication load of an explosive nature and is addressed in the first place to the production sphere of economy. The dynamically combined connection network not only prevents traffic bursts, but also based on the physical and logical circuit commutation gives the possibility of adapting to the existing traffic pattern. Although the described solution is addressed to the optical transmission environment, its effective functioning in the Ethernet networks with circuit switching and partly in wireless networks has been confirmed empirically. The theoretical foundations were verified in the design and construction of a reconfigurable super-microcomputer and the intelligent system detection of attacks addressed to industrial Internet.**

I. Introduction

THE progressive globalization of information systems and dynamic development of data transmission techniques have resulted in significant changes in the use of information systems and parallel computing. While earlier systems and parallel processing were the domain of scientific centers, they are now also widely used in industry and management. It is difficult to talk about any comparison of computer used in science and industry; however, they share a common feature: the parallelization of the computational process, which is implemented on many devices.

The performance of a parallel computer system is primarily determined by the performance of its components: computational elements and connections between them. Theoretically, the communication subsystem has less impact on the performance, but in practice it determines the size of entire system and thus its final efficiency. Computational efficiency, also referred as its performance, is a quantitative feature of the speed at which specific operations are performed. In turn, the most important parameter of the communication subsystem is its bandwidth defining the number of information units sent over time unit.

In the book [1] considered to be classical, the mathematical and methodological basis for estimating the above parameters has been comprehensively presented. Similar considerations for computer systems, networks and cloud computing can be found in [2]-[5]. This subject can be found in the latest publications in conference materials and magazines. They focus on determining power in heterogeneous systems, including IoT [6]-[8]. Solutions of this type are used in computations in the area of basic research, nuclear physics, medicine, and so on.

Assigning to all computer systems a statement known from the literature, that the only criterion for their evaluation is computational efficiency they offer, is inappropriate [9]. In supercomputers and clouds, the maximum efficiency is indeed an important factor of the entire solution [10], [11]. In other classes of systems, other properties may be desirable. For example, in management systems, information gathering, industrial IT, control systems, power and bandwidth should correspond to the actual demand and other characteristics or parameters are particularly desirable. The following statements can be distinguished:

a. In industrial IT crucial system properties change over time, including efficiency, bandwidth, availability, latency, etc.;
b. The effectiveness of inter-communication depends on the relationship between topology and supported traffic pattern, which is not constant and can change dynamically over time;
c. The size of computations made, and the volume of data transferred per time unit vary and may temporarily have a bursty character that can cause system's malfunctions;
d. In the system exist temporarily underloaded units that can participate in computations within any part of the system and tasks being solved.

From an economic point of view, the solution to the problem of insufficient resources cannot be handled by their redundancy as it generates significant costs both at the construction and operation stage. Most likely, these resources would be never used in satisfactory manner [9], [12].

In this work, the problems of insufficiency and redundancy of resources are solved by self-adjusting communication network based on multi-channeling, grouping and hierarchization. It is assumed that parameters and characteristics of the communication network are most important properties of entire computer system.

In the classification of the topology optimization methods presented in [13] among reconfigurable networks, there are demand-aware networks in which the adaptation to the load can be static or dynamic. In the first case, the topology is immutable while in the second one is reconfigurable. Reconfigurable networks are also called self-adjusting. Additionally, it can be noted that although reconfigurable architectures represent an interesting development paradigm, there are currently no analytical tools to study their potential [13].

The concept of distributed passive commutation developed in this work was proposed in [14] and the organization of experiments conducted is described and justified in [15]. The main goal is to present research on creating a budget reconfigurable heterogeneous system intended for the industry.

The paper starts with section 2, where main idea of the virtual multi-bus connection system and fundamental assumptions are presented. This section ends with classification of the communication efficiency improvement methods and the design task definition.

In section 3, a description of the design tasks is described and the methodology for designing a parallel processing system based on the virtual multi-bus connection system is provided. Three independent optimization criteria were considered: reliability, efficiency (computational and communication) and latency. For each, an appropriate algorithm were developed.

The paper ends with description of simulation tests performed, conclusions and further research goals.

II. Preliminary arrangements

A. Bus Systems and Their Representation

Let's assume that a parallel computer system with bus communication is a combination of two types of equal objects: computational nodes N and buses B. A node can be incidental with any number of buses. A network composed of n nodes and m buses is usually denoted as $[n,m]$ and describes an incident matrix $I = \{b_{ij}\}$ of size $n \times m$. The element $b_{ij} \in I$ is equal to 1 if the node N_i, $i = 1,\dots,n$ is incidental with bus $B_j, j = 1,\dots,m$, otherwise $b_{ij} = 0$. By the definition of incident matrix, bus networks do not allow loops for both buses and nodes. Therefore, if there are multiple connections between any two elements in the system (i.e. the selected node will be integrated with the selected bus by means of several connections), the traditional method of bus denotation cannot be used.

If the number of buses incident with i-th node is designated as s_i^w, $i = 1,\dots,n$ (node degree) and the number of nodes incident with j-th bus as s_j^m (bus degree), then for any bus network $[n,m]$ there is relationship between the total degree of nodes and buses:

$$\sum_{i=1}^{n} s_i^w = \sum_{j=0}^{m} s_j^m = s\,. \qquad (1)$$

Expression (1) is the basis of network synthesis method developed by authors with single connections presented among others in [16], [17]. In contrast to the network with direct connections, for the bus network $[n,m]$ with incidence matrix M, there is always a network with a transposed incidence matrix M^T.

Bus networks are structurally equivalent to hypergraphs. They are used to analyze bipartite graphs [18]. The number of nodes in X_0 and X_1 parts of bipartite graph is equal to n and m, respectively. Its edges are local connections $l_{p,q}$ where p, q – the number of computational node and the bus respectively, whose purpose is to connect the computing node with the bus. This is, in fact, a description of the PBL neighborhood graph (**P**rocessor-**B**us-**L**ink).

The PBL graph $G = (V, B, L)$ containing $|V_G| = n$ nodes, $|M_G| = m$ buses and L_G link set is the bipartite graph G_{PBL} which can be described by following pair: $\left(V_{G_{PBL}}, B_{G_{PBL}}\right)$ and $V_{G_{PBL}} = VV_{G_{PBL}} \cup VB_{G_{PBL}}$, where $VV_{G_{PBL}} = V_G$ and $VB_{G_{PBL}} = B_G$. Connections in graph G between buses and nodes are represented by $B_{G_{PBL}}$. The nodes $v_{G_{PBL},i}$ and $b_{G_{PBL},j}$ are connected by the edge $l_{G_{PBL},k}$ if and only if in the source bus system, the bus b_i is connected to the node v_j with link l_k. Such a representation of the bus system is shown in Fig. 1.

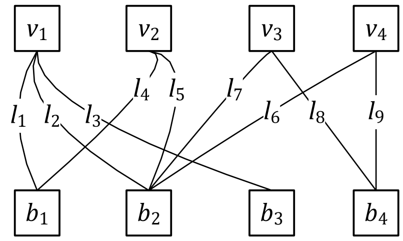

Fig. 1. The general form of multi-bus system

To describe graph in simulation programs a neighborhood matrix with a block structure was used:

$$A = \begin{bmatrix} 0_{n,n} & W \\ W^T & 0_{m,m} \end{bmatrix}, \qquad (2)$$

where A – matrix with size $n \times m$, $0_{n,n}$, $0_{m,m}$ – zero matrices with size $n \times n$ and $m \times m$ respectively. When storing in computer's memory, these elements are omitted and all information about the graph is contained in the sub-matrix W, sometimes called bi-neighborhood matrix. For a bipartite graph G with parts $V = \{v_1,\dots,v_n\}$ and $B = \{b_1,\dots,b_m\}$ the bi-neighborhood matrix W is a binary matrix of size $n \times m$ where $w_{i,j} = 1$ if and only if $\left(v_i, b_j\right) \in L$ [19].

The designed parallel processing system is heterogeneous, consists of server units with relatively large computing power, and dynamically connected low-efficient Raspberry, Arduino or similar units. Therefore, for the mathematical description of the system tripartite graph was used, whose components are servers S, support servers K and buses B. It was presented on Fig. 2.

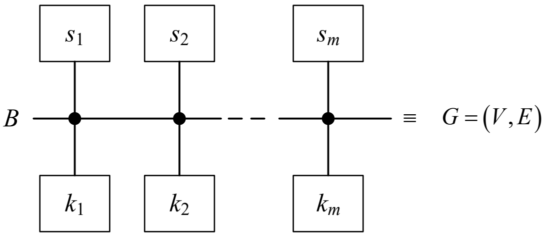

Fig. 2. Multi-bus system presented as tripartite graph

The denotation of tripartite graph in the matrix form was proposed in [18], [20]. Their representation is reduced to a diagonal graph, then presented as a subgraph of an upper graph which is, in fact, an algebraic graph.

As an alternative method for describing the bus network topology, graph algebra was developed for this purpose [20]. Let sets $S=\{s_1,s_2,\dots,s_r\}$, $B=\{B_1,B_2,\dots,B_m\}$, $K=\{k_1,k_2,\dots,k_n\}$ describe sets of primary servers, referred to as servers, buses and support servers, respectively. Informally, connections between elements can be described using following rules:

1. The recipient prefers the selected service provider. Preferences are not permanent and can be changed without restrictions during the work of Multi-bus Reconfigurable Computer System;
2. Connection between service recipients and service providers is performed by means of logical bus channel.

The formalization of this process takes place in the following way. Let's define a threefold relation $P \subseteq K \times S \times B$ on sets $K,S,B\colon (k,s,b) \in P \Leftrightarrow k$ prefers the service provider and these preferences are carried out using bus b $(k \in K, s \in S, b \in B)$. Relation P induces two relations of equivalence R and R^- on sets K and B respectively:

$$k_i R k_j \Leftrightarrow (k_i,s',B'),(k_j,s'',B'') \in P \text{ i } s'=s'', \tag{3}$$

$$\begin{gathered} b_i \overline{R} b_j \Leftrightarrow (k_i,s',b_i),(k_i,s',b_i),(k_s,s'',b_j) \in P \\ s'=s'' \end{gathered} \tag{4}$$

Informally, the relations R and R^- mean that the supporting servers k_i and k_j prefer the same server and joining it is carried out using the b_i and b_j buses. Connecting the selected server to the supporting server is performed by means of one and the same bus, i.e. $k_i\, Rk_j \Rightarrow b_t=b_l$, $i,j=1,\dots n$, $t,l=1,\dots,m$ and $k_i \neq k_l \Rightarrow s_i \neq s_j$. The model in the form of finite algebra of undirected connected graph can be described as:

$$\begin{aligned}
&\left(T^0_{s_1} * T^0_{B_1} \cup T^0_{B_1} * T^0_{B_1}\right) *_{f_1} T^0_{B_2} \cup \dots \\
&\cup \left(T^0_{s_{m-1}} * T^0_{M_{m-1}} \cup T^0_{M_{m-1}} * T^0_{k_{n-1}}\right) * \\
&*_{f_{n-1}} T^0_{M_m} \cup \left(T^0_{s_m} * T^0_{B_m} \cup T^0_{B_m} * T^0_{k_m}\right) = \\
&= \left(T^0_{s_1} * T^0_{B_1} \cup T^0_{B_1} * T^0_{k_1}\right) \cup \dots \cup \\
&\cup \left(T^0_{s_m} * T^0_{B_m} \cup T^0_{B_m} * T^0_{k_m}\right) *_{f_1} T^0_{B_1} *_{f_2} \dots *_{f_{m-1}} T^0_{B_m} = \\
&\left(T^0_{s_1 B_1} \cup T_{B_1 k_1} \cup T_{s_2 B_2} \cup T_{B_2 k_2} \cup \dots \cup T_{s_m B_m} \cup T_{B_m k_m}\right) * \\
&* f_1 T^0_{B_1} * f_2 \dots * f_{m-1} T^0_{B_m} .
\end{aligned} \tag{5}$$

Due to the alternation of connection operations and mutually unambiguous connection, it can be noticed that service providers and buses are equal and can be interchanged. The graphical representation of the organization described by expression (5) is shown in the form of tripartite graph in Fig. 2. The element ensuring its consistency is bus set B.

B. Organization of the Multi-bus System

The organization of a parallel reconfigurable computing system with multi-bus connections is shown in Fig. 3.

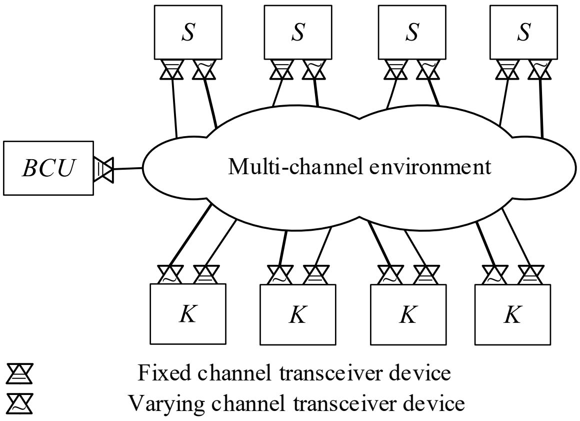
S
S
S
S
BCU
Multi-channel environment
K
K
K
K
Fixed channel transceiver device
Varying channel transceiver device

Fig. 3. The basic organization of parallel reconfigurable computing system: BCU – bus control unit.

Each of the computing nodes has been equipped with at least one fixed and one adjustable single-channel communication interface. If it is possible, the parameters of each fixed channels are unique in the whole organization, not limiting to setting up networks of any connection architecture. Deviation from the above rule is the management channel which all computational nodes are connected to. This channel always uses broadcast transmission and is intended only for sending information on the configuration of adjustable transceiver devices.

In the design of the connection system following assumptions were made:

1. The purpose of using multi-channel connections is to minimize the complexity of the design, construction and operation of computer systems using a set of processing elements. Instead of direct connections (which ale complex in design), broadcasting logical channels are used defining the architecture which is

relatively simple. In addition, multi-channeling should ensure maximum efficiency of using the bandwidth available in the physical transmission channel. Communication channels can work not only in the broadcast mode, but also two-point connections are possible. The only channel connected to all nodes and working only in broadcast mode is management channel. Depending on the acceptable construction and operating costs, it can be made in physical or logical form;

2. The bus channel can be built in the form of an active electronic or passive optical system. The use of all possibilities offered by the proposed architecture is possible in the second case. The proposed architecture, to a limited extent, can be made based on classic network switches equipped with a circuit switching. If it is acceptable to build a network with group broadcasting, the functional scope of the prototype increases. In the electrical implementation of the architecture, the management channel is redundant, because its role is fulfilled by the control systems of a switch. Part of the system's functions with proposed architecture can also be reconstructed based on wireless networks;
3. Each server is equipped with one fixed channel transceiver and several adjustable devices working with any channel available in the system. The greater the number of interfaces, the system capabilities are wider. The disadvantage of the tunable devices is the relatively long retuning time, as well as the higher purchase cost (theoretically). In turn, the use of fixed interfaces limits the possibilities of reconfiguration;
4. Logical channels are build based on the resources of physical communication channels. Physical channels use bus or star topology. The transmission in physical channel has a passive directed character. A broadcast transmission mode is used in the logical channel;
5. Reconfiguration consists in tuning the transceiver devices to work with another logical channel. This process is initiated by BCU and implemented by means of broadcast control bus;
6. BCU managing node, based on information flow and node load, designs in real time a new architecture of links and sends information about it to every computational node. Since each of the elements must be permanently connected to the management node, the use of fixed transceiver device is preferable;
7. Each node performs tuning operation on its transceiver interfaces, thus creating a new connection network. The condition for connecting the processing node to any other node or group of them is to work with the same logical channel available within the same physical channel;
8. In the presented architecture it is assumed that channel splitters (couplers) are passive devices. Further functionalities of the system expanding the scope of its reconfiguration can be obtained by using active devices. Their functionality has been proposed by the authors, among others in [14], [15]. In particular, it is interesting to divide logical buses into fragments, as has been used for physical ones.

From the point of view of computer systems theory, reconfigurable bus systems should be classified into a group of reconfigurable multi-machine systems with tight or loose (more often) connections. Systems with tight connections will be called those in which transceiver elements will use the same type of logical channel. On the other hand, in the systems with loose connections transceiver components of the computational node are completely independent of each other. Theoretically, when designing a system both tight and loose connections can be used. In the analyzed solutions, in order to minimize costs, mainly organizations with a single connection of a computing element with a logical bus are used. In addition, transceiver devices should work with one type of logical channel. Otherwise, the bus input and output channels would be separated, and computational elements would be used for communication between nodes. Therefore, in the analyzed system it is recommended to use tight connections.

If the computational node is connected to the logical bus via several transceiver elements, there is a theoretical possibility of splitting the transmitter and signal receiver, and thus creating systems with loose connections. The above hypothesis, however, rises some doubts. First, the logical bus must use the same channel along its entire length. Secondly, connecting the processing node to the logical bus via several transceiver devices is only appropriate if the system is more resistant to failures. Thus, from the point of view of system definition with loose and tight connections, the analyzed systems are characterized by tight connections. Single, multiple and partial connections of computational element (service provider or recipient) with a logical bus is schematically shown in Fig. 4.

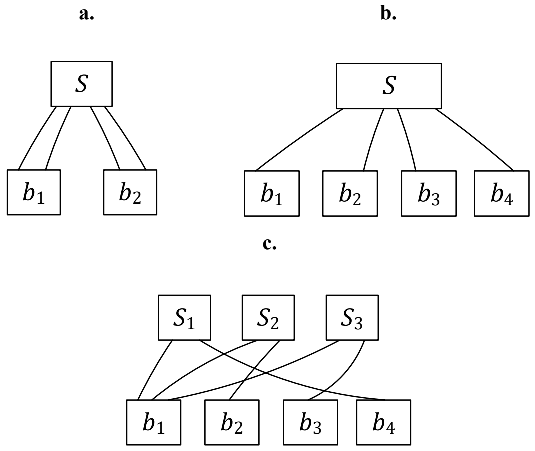

Fig. 4. Node connections to the virtual bus: a. Multiple (double) complete; b. Single complete; c. Partial

Let's consider the condition of the computer system's readiness with equivalent nodes. Let K_{in} specify the total number of processing nodes used in the system and K_{in}^{min}

– the minimum number of nodes necessary to implement all its functionalities and k_{in} – number of correctly working nodes in the system. Then, the condition of readiness has the form: $K_{in} \geq k_{in} \geq k_{in}^{min}$. In addition, it is necessary to operate a communication subsystem that ensures interoperability of at least k_{in}^{min} processing nodes. For systems with a server and supporting server, the following designations are used: K_k – the total number of supporting server; K_s – total number of servers, k_s^{min} – the minimum number of servers necessary to implement the systems functionality; k_k^{min} – the minimum number of supporting servers necessary to implement the systems functionality; k_s – the number of correctly working supporting servers; k_k – the number of correctly working supporting servers. The readiness condition can be written as: $K_s \geq k_s \geq k_s^{min}$ and $K_k \geq k_k > k_k^{min}$. In addition, the correctness of the communication subsystem ensuring interoperability of no less than k_s^{min} service provides and k_k^{min} service recipients should be guaranteed.

In this research, servers were usually desktop units with extensive configuration. As the supporting server, the most commonly used are self-made network traffic analyzers.

C. Classification of the multi-bus systems

In traditional bus networks, the communication network is single-channel and each of the system's element is connected to it only once. In the case of multi-channel buses, user is usually connected only to a specific channel. An exception to this rule is switchable optical networks in WDM technology, whose implementation costs exclude the possibility of their use in non-telecommunication applications.

Reconfigurable multi-bus parallel systems have several elements to effectively adapt to the demand for computational efficiency and bandwidth. As the first one should be mentioned the possibility to equip both types of computing nodes with a variable number of transceiver elements. This allows one or more times to connect any bus or a set of buses. Secondly, the use of transceiver devices tuning to a specific channel provides dynamic changes in connections and grouping of supporting servers around primary servers. The groups being build (node clusters) have their own communication environment and can be isolated from external interference. For example, a separate group can be created by users using services insensitive to latency, another generating low traffic, yet another characterized by bursty traffic. Thirdly, the buses are also grouped (clustered). In this way, not only the computational efficiency, but also bandwidth of communication channels can scale. The most useful is mixed grouping, where computational nodes and communication channels are simultaneously grouped. Fourthly, the multiplicity of node interfaces allows the use of folded buses (single, double or triple) ensuring better usage of the bus. Another method consists in the hierarchization of logical channels where channels are combined into groups that support different sets of servers. It is also possible to divide overloaded buses into smaller parts.

Thanks to the diversity of the proposed methods, the architecture of connections can be adapted to the traffic patterns and communication requirements. Because in optical systems the change in the wavelength used by transceiver element takes milliseconds, the adaptation of the connection architecture to current requirements can be dynamic and carried out in real time.

The classification of bus interconnection architecture is shown in Fig. 5.

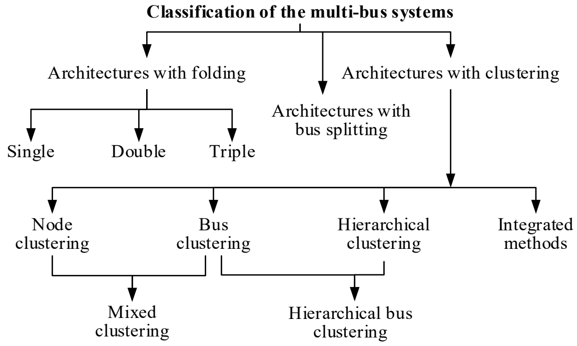
Classification of the multi-bus systems
Architectures with folding
Architectures with bus splitting
Architectures with clustering
Single
Double
Triple
Node clustering
Bus clustering
Hierarchical clustering
Integrated methods
Mixed clustering
Hierarchical bus clustering

Fig. 5.Classification of multi-bus architectures

D. Definition of the project task

For a multi-bus communication system, a connection architecture should be designed to ensure **a.** Maximization of reliability; **b.** maximization of computational efficiency; **c.** minimization of latency while ensuring the minimum (maximum) level of other system characteristics.

Many parameters affect the design process to a greater or lesser extent. The first of them is the generalized construction cost [18], [21]. It is assumed that the generalized costs of constructing the system will be proportional to the number of informational inputs. In a given case, the parameter is the summary logical degree S_Σ^{LF} of all physical (processing) nodes connected to the logical buses used. From a technical point of view S_Σ^{LF} is the total number of transceiver devices used in the nodes of service providers and system users. If the analyzed architecture is based on expensive tunable devices, the value S_Σ^{LF} determines costs of the system. If an optimization design procedure is performed, the generalized costs will be one of the optimization criteria.

The second important parameter is the communication channel's load. In a special case, the system may use two-point communication channels with one server and one supporting server. This case, however, should be regarded as unique. In fact, a set of units of both types will be added to the logical bus, whose efficiency will determine the bus load. The parameter of a system using logical buses is branching factor W_{Ri}. The branching factor of the i-th bus is defined as a sum of transceiver devices in which the processing nodes connected to the common logical bus are equipped. From the point of view of load minimization, it is advantageous to use only single connections (see Fig. 4). In this case, the branching factor W_{Ri} of the i-th logical bus depends only on the number of processing nodes connected to it.

If all the processing nodes load the transmission channel identically, the W_{Ri} factor determines the level of logical bus occupancy. The value W_{Ri} manifests a key influence on the bus channel efficiency, i.e. it determines the size of coupling-splitting physical communication channels. In the presented organization, the processing elements are connected with each other by means of a set of logical buses functioning in one or a set of physical buses.

Connection of the computational element to a specific logical bus is done by setting the parameters of transceiver device. Therefore, from the point of view of performance and bandwidth analysis, parameters related with their reconfiguration time are important. Further, the symbol t_n indicates the time of setting the transceiver element, i.e. the required time to adapt to the indicated logical channel after processing node is started. In turn, t_p will mean the transceiver's device tuning time, i.e. the time required to adapt to specific logical channel. It can be assumed that for the analyzed system with tight connections, the setting and tuning times are identical, i.e. $t_n = t_p$. In a system using tunable transceiver devices, a situation may arise when the currently tuned device should stop the logical channel change and adapt to another channel. As t_s the time necessary to stop the tuning process and start to work with another channel will be marked. It is assumed that $t_s \leq t_p$. If $t_s = 0$, the stopping of tuning process is immediate.

Another important parameter of the computational system, especially for industrial application, is its reliability. The research used the reliability model developed by the authors assuming the equivalence of processing nodes and the interrelationship between physical and logical components of the connection network [15], [18], [22].

III. Design process of multi-bus architecture

A. The Idea of Methodology

Numerous algorithms for designing computational system with multi-channel bus connections are known from the literature. Most of them focus on ensuring a certain level of performance. From the point of view of connection system, these algorithms are aimed at creating an architecture with a specific level of performance or latency occurring between computational elements considering the condition of non-transferability of given costs.

In contrast, the proposed methodology additionally takes into account the level of reliability of computational system. First, it allows to create computing systems with connections characterized by maximum reliability and minimum acceptable communication efficiency and maximum communication delay. Secondly, it ensures the load balancing in bus channels.

Considering the requirements for ensuring minimum acceptable value of bandwidth and reliability as well as the maximum acceptable construction costs and latency, three design tasks can be defined.

B. Task 1: Maximization of reliability

Determine the minimum number K_B of communication channels and the method of connecting computing nodes to them, which ensure the maximum level of reliability with a limited total cost of technical devices, minimum acceptable efficiency D_C and transmission time *t*.

Task 1 can be defined as construction of a maximum reliable computing system, which can be written as an optimization task:

$$R(K_B) \to \max \qquad (6)$$

for the following restrictions:

$$t(K_B) \leq t_{max}\,,\; D_C(K_B) \geq D_C^{min}\,,\; U_\Sigma(K_B) \leq U_{max}$$

where: t_{max} – the maximum acceptable value of latency between any pair of computational nodes; D_C^{min} – minimal acceptable computational efficiency of the system; U_{max} – maximum acceptable cost of technical devices.

The task described in (6) should be solved using procedure consisting of three basic steps. To achieve the goal, those steps should be described in the following form:

$$R(K_B^{min}, K_B) \to \max$$

for the following restrictions:

$$1 \leq K_B \leq K_B^{min}\,,\; t\left(K_B^{min}, K_B\right) \leq t_{max}\,,$$

$$D_C\left(K_B^{min}, K_B\right) \geq D_C^{min}\,,\; U_\Sigma\left(K_B^{min}, K_B\right) \leq U_{max}$$

where: K_B^{min} – minimum number of fully operational buses, which ensures the maximum acceptable latency t_{min} and efficiency D_C^{min}. Next, each step will be described.

Step 1. Create the basic system configuration composed of K_B^{min} buses. The sequence of actions performed within a given step is shown in Algorithm 1.

Algorithm 1. The first step of designing reliable connection system

```
input: $D_C^{min}$, $K_B^{max}$, $t_{max}$, $K_{in}$
for $l \leftarrow 1\ to\ K_B^{max}$
    if $\left(D_C(l) \geq D_C^{min} \wedge t(l) \leq t_{max}\right)$
        if $(l \leq K_{in})$
            return $partialConnections$
        else
            return $completeConnections$
        end if
end for
return $\emptyset$
```

Afterwards, configurations are created containing $l = 1,2,\dots,K_B^{max}$ channels, where K_B^{max} – maximum acceptable number of channels based on technical and economic criteria. Then, using the methods proposed in [13], [15], their performance and latency are evaluated. If $K_B^{min} > 1$. Then, it is assumed that the communication environment is functioning in the load sharing mode between all K_B^{min} channels.

First, complete connections are analyzed in which the number of channels $K_{in} > K_B^{min}$. If any configuration with complete connections that meet all the restrictions cannot be found, architectures with partial connections are created and examined $\left(K_{in} < K_B^{min}\right)$. The searching process ends, when the first configuration that meets the restrictions is found (i.e. with the minimal number of channels K_B^{min}).

Step 2. Improve the reliability of the connection system by introducing additional transceiver elements and verify restrictions on the total cost of the system. The sequence of actions performed at given step is shown in Algorithm 2.

Step 3. Evaluate computational efficiency and latency as well as the reliability of the synthesized system. If these requirements are not met, the design process is repeated using modified inter-node connections classified in Fig. 5.

Algorithm 2. The second step of designing reliable connection system

```
input: $K_{in}$, $U_{max}$
for $K_B \leftarrow K_{in}$ to $K_B^{min}$
    if $(U_\Sigma(K_B) \le U_{max})$
        return 1
    end if
end for
return 0
```

C. Task 2: Maximizing system efficiency

Determine the minimum number of K_B channels and the method of connecting computing nodes to them, ensuring maximum computational efficiency D_C.

Task 2 can be defined as the task of building computing system with maximum performance. The relevant optimization task has the following form:

$$D_C\left(K_B^{min},K_B\right)\to\max$$

for the following restrictions:

$$1\le K_B\le K_B^{min},\ t\left(K_B^{min},K_B\right)\le t_{max},$$

$$R\left(K_B^{min},K_B\right)\ge R_{min},\ U_\Sigma\left(K_B^{min},K_B\right)\le U_{max}$$

where R_{min} – minimal acceptable reliability of the computing system. The solution of this project task is carried out in three steps.

Step 1. Design an architecture with complete connections that meets the requirements regarding costs and reliability. The maximum computational efficiency is characterized by the architecture in which all the transceiver elements are used. In particular, the system should meet the following condition: $K_B^{min} = K_B = K_{in}$. However, because all buses are used, such architecture is also the most unreliable. Therefore, at the beginning the architecture with maximum number of channels from the range $1 \le K_B \le K_{in}$ is sought. The cost limit $U_\Sigma\left(K_B^{min}, K_B\right)$ should be met.

Further, the configurations with the minimal number of buses K_B^{min} from the range $K_B - 1, K_B - 2, \ldots, 1$ are gradually determined. Reliability is calculated for each of them. The searching procedure is continued until a configuration meeting the reliability condition is found. Among them, one is selected, and it should satisfy all other requirements. This step of design procedure is shown as Algorithm 3. If all the requirements are met, optimal number of buses K_B^{opt} and maximum value of efficiency D_C^{max} are returned. Otherwise (i.e. $K_B^{opt} = 0$), appropriate connection system does not exist.

Algorithm 3. The first step of designing computationally efficient system

```
input: $K_{in}$, $R_{min}$, $t_{max}$, $U_{max}$
for $l \leftarrow 0$ to $K_{in}$
    if $(U_\Sigma(l) \le U_{max})$
        exit for
    end if
end for
if $(l > K_{in})$
    return $\emptyset$
end if
$D_C^{max} \leftarrow 0$, $K_B^{opt} \leftarrow 0$
for $K_B \leftarrow l$ to 1
    if $(R(K_B) \ge R_{min} \wedge t(K_B) \le t_{max})$
        if $(D_C(K_B) > D_C^{max})$
            $D_C^{max} \leftarrow D_C(K_B)$
            $K_B^{opt} \leftarrow K_B$
        end if
    end if
end for
return $\left(K_B^{opt}, D_C^{max}\right)$
```

Step 2. A configuration with complete connections is required. It should meet the requirements for total costs U_Σ and reliability R. Note that the architecture with partial connections $(K_B > K_{in})$ is usually unacceptable due to the construction cost. Therefore, determining architecture with partial connections ensuring redundancy of transceiver devices is possible only if the architecture with complete connections $(K_B = K_{in})$ satisfies cost limit. The basis for further considerations is the fact that the architecture with partial connections with $K_B = \lceil 3K_{in}/2 \rceil$ is characterized by maximum performance and cost as well as minimal latency and reliability. Therefore, for the implementation of configurations that meet the cost and reliability constraints, structures are created sequentially for which number of channels K_B is respectively equal to: $\lceil 3K_{in}/2 \rceil, \lceil 3K_{in}/2 \rceil - 1, \ldots, K_{in} + 1$. For each variant, the cost and reliability are determined. The given process is continued until the configuration is defined meeting both criteria at the same time. This step is described as Algorithm 4. According to algorithm, system can consist of complete or partial connections. In the case, when requirements cannot be met, construction of the system is impossible.

Algorithm 4. The second step of designing computationally efficient system

```
**input**: $K_{in}$, $R_{min}$, $U_{max}$, $t_{max}$, $D_C^{min}$
**for** $l \leftarrow 1$ $to$ $K_B^{max}$
  **if** $(l \leq K_{in})$
    **if** $\left(D_C(l) \geq D_C^{min} \wedge t(l) \leq t_{max}\right)$
      **for** $k \leftarrow \lceil 3K_{in}/2 \rceil$ $to$ $K_{in}$
        **if** $(U_\Sigma(l) \leq U_{max} \wedge R(l) \geq R_{min})$
          return $partialConnections$
        **end if**
      **end for**
      exit for
    **end if**
  **else**
    return $completeConnections$
  **end if**
**end for**
return $\emptyset$
```

D. Task 3: Minimization of latency

Specify the minimum number of K_B^{min} buses and the method of connecting computational nodes to them ensuring minimal latency with additional requirements: total cost U_Σ of its technical devices, reliability R and minimal computational efficiency D_C.

Task 3 is essentially the task of building a computing system with a minimum communication delay. The optimization task can be written as:

$$t\left(K_B^{min}, K_B\right) \rightarrow \min$$

with following restrictions:

$$1 \leq K_B^{min} \leq K_B \,,\; D_C\left(K_B^{min}, K_B\right) \geq D_C^{min} \,,$$

$$R\left(k_B^{min}, K_B\right) \geq R_{min} \,,\; U_\Sigma\left(K_B^{min}, K_B\right) \leq U_{max} \,,$$

where: R_{min} – minimum acceptable reliability of the computational system.

Similar as in the previous one, task 3 is solved in three steps. The first one is based on searching for an architecture with complete connections, meeting cost and reliability constraints, characterized by minimal communication delay. The second is to search for an architecture with partial connections that meets above requirements while the third is to choose the architecture with highest efficiency.

IV. SIMULATION TESTS

The research was conducted in two independent areas. The first included theoretical studies based on models prepared and verified for this purpose. The super-microcomputer will be based on Raspberry devices connected via a multi-channel bus. For reconfiguration, transceiver devices with fixed channels were used and tuned in various configurations. The research has shown that it is possible to ensure a linear increase in the computing power of the system in a very wide range, depending only on the number of available logical buses and the possibility of their management by BCU. The studies simulated a parallel solution to classical tasks in the field of algebra, set theory and graph theory. For example, double reduction in solving time for a system of linear homogeneous Diophantine equations using the Contejean-Devie algorithm occurred after addition of 4 additional support servers. Similar results were obtained for the Pottier and Demenjoud methods. Empirical verification of modelling results was not possible now. The research confirmed the author's expectations regarding the flexibility of reconfiguration. For a dynamically changing assortment of short tasks, the computational efficiency obtained in parallel system (i.e. using support servers) was smaller than expected. For this reason, the simulations indicate a relatively long reconfiguration time using tunable transceiver devices.

The second analyzed architecture is a real commercial IT system designed to detect attacks on availability in a heterogeneous system containing both classic PC computers as well as tablets and smartphones. The research was conducted in *Dominet* corporation and has shown that in the case of an attack or the occurrence of bursty traffic, remote Raspberry modules, thanks to flexibility of connection system, can effectively perform both of their functions: a network traffic analyzer and an additional computational element. Threat detection was performed using among others AI and graph theory. Research has also shown that supporting servers are required only at the learning stage of neural network. Other system tasks are performed within an acceptable time. It was also found that adding less than 5-6 servers to the system is pointless. When solving learning tasks and searching for a perfect match in a n-partite graph, saturation of 1Gb communication channels followed the inclusion of a dozen or more support servers. The results of the experiments showed that the usefulness of the electrical system implementation to solve typical computational tasks requiring parallelism is satisfactory, but worse than the possibilities offered by optical solutions. Available reconfiguration options are small in relation to the optical environment.

V. CONCLUSIONS AND FURTHER WORK

The approach presented in this work was used to create a methodology for designing multi-channel bus connections addressed to the communication service of heterogeneous distributed system. In contrast to existing methodologies [23], which focus on ensuring a certain level of bandwidth in entire computational system omitting reliability parameters, the proposed methodology allows to determine the architecture of connections characterized by:

a. maximum reliability with minimum acceptable efficiency and maximum acceptable latency in the connection network;
b. minimum latency with a minimum acceptable reliability and efficiency;
c. maximum computational efficiency with a defined acceptable level of reliability and latency.

The maximum construction costs are limited for each of the solutions. The construction of a distributed system uses buses with complete and partial connections, both flat and hierarchical.

Further research will focus on the formalization of the specific architecture selection from a set of acceptable solutions, considering the incompleteness of information necessary in decision-making process. For this purpose, the connection network will be presented in a form of n-partite hypergraph. The solution assessment will be multi-criteria. The set of them will be flexible and its selection will depend on the designer's needs, in particular on the nature of future operation of the connection network.

Based on hypergraph model, a set of $A(a)$ acceptable solutions will be considered. For each of them, the following quality indicators will be defined:

1. The first optimization criterion applies to computational efficiency $\Phi_1(a) = \max_{a \in A} \min_{e \in E_a} \omega(e)$, where E_a – a set of hypergraph edges belonging to the solution a. Maximized of minimum level of performance (computational or communication) of the system will be considered;
2. The second optimization criterion applies to communication latency: $\Phi_2(a) = \min \sum_{e \in E_a} \xi(e)$ ensuring the search of the connection network with a minimum total latency. For systems with different importance, the value $\xi(e)$ of the expected latency change will be scaled to the node's priority;
3. The reliability criterion: $\Phi_3(a) = \max \sum_{e \in E_a} \psi(e)$. As in the case of latency criterion Φ_2, this criterion ensures the search for the network architecture with maximum total reliability.

The possibilities of this method are not limited to the summation criterion in min or max form. To evaluate the quality of the final solution, any methods of convolution can be used, including methods that consider weights of individual subparameters.

Partial criteria will be combined using the following function: $\Phi(a) = \left(\Phi_1(a), \Phi_2(a), \Phi_3(a)\right)$. The multi-criteria objective function $\Phi(a)$ determines in the set of acceptable solutions A the Pareto set A^p containing Pareto solutions a^p. If two solutions $a_1, a_2 \in A$ of the vector objective function $\Phi(a)$ are equivalent, then from the set A^p a full set of alternatives will be extracted, which is, in fact, the maximum system of vector Pareto optimizations.

Acknowledgment

The work was realized as a part of fundamental research financed by the Ministry of Science and Higher Education, grant no. 16.16.110.663, task 8.

References

[1] R. Jain, The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling, Hoboken, New Jersey, USA: Wiley & Sons, 1991.

[2] N. J. Gunther, Analyzing Computer System Performance with Perl::PDQ, Heidelberg: Springer-Verlag, Germany, 2005.

[3] P. J. Fortier and H. E. Michel, Computer Systems Performance Evaluation and Prediction, Amsterdam, NL: Digital Press, 2003.

[4] B. Greg, Systems Performance. Enterprise and the Cloud, Upper Saddle River, NJ, USA: Prentice Hall, 2014.

[5] A. Oppermann, M. Esche, F. Thiel, J.-P. Seifert, "Secure Cloud Computing: Risk Analysis for Secure Cloud Reference Architecture in Legal Metrology," in Proceedings of the Federated Conference on Computer Science and Information Systems, pp. 593-602, 2018, DOI: 10.15439/2018F226

[6] A. D. Malony, S. Biersdorff, S. Shende, H. Jagode, S. Tomov, G. Juckeland, R. Dietrich, D. Poole and C. Lamb, "Parallel Performance Measurement of Heterogeneous Parallel Systems with GPUs," in International Conference on Parallel Processing, 2011.

[7] M. U. Asharif, F. A. Eassa, A. A. Albeshri and A. Algarni, "Performance and Power Efficient Massive Parallel Computational Model for HPC Heterogeneous Exascale," IEEE Access, vol. 6, pp. 23095-23107, 2018.

[8] K. Vanishree and M. Purnaprajna, "Work in Progress: Performance Modeling for Data Distribution in Heterogeneous Computing Systems," in International Conference on Compilers, Architectures and Synthesis for Embedded Systems (CASES), Torino, Italy, 2018.

[9] M. Hajder, H. Loutskii and W. Stręciwilk, Informatyka. Wirtualna podróż w świat systemów i sieci komputerowych, M. Hajder, Red., Rzeszów: Wydawnictwo Wyższej Szkoły Informatyki i Zarządzania, 2002.

[10] T. Sterling, M. Anderson and M. Brodowicz, High Performance Computing. Modern Systems and Practices, Cambridge, MA, USA: Morgan Kaufmann, 2018.

[11] S. Bhowmik, Cloud Computing, Cambridge: Cambridge University Press, 2017.

[12] M. Hajder and M. Kiełbus, „Matematyczny model opóźnień w sieci z komutacją pakietów," w XV Konferencja Sieci i Systemy Informatyczne, Łódź, 2007.

[13] C. Avin and S. Schmid, "Toward demand-aware networking: a theory for self-adjusting networks," SIGCOMM Comput. Commun. Rev., vol. 48, no. 5, pp. 31-40, 2018.

[14] P. Hajder and L. Rauch, "Reconfiguration of the Multi-channel Communication System with Hierarchical Structure and Distributed Passive Switching," in ICCS 2019, Part II, LNCS 11537 proceedings, 2019.

[15] M. Hajder, Reconfigurable multichannel network communication systems architecture, Rzeszow: Wydawnictwo Wyższej Szkoły Informatyki i Zarządzania w Rzeszowie, 2018.

[16] M. Hajder and M. Bolanowski, "Connectivity Analysis in the Computational Systems with Distributed Communications in the Multichanel Environment," POLISH JOURNAL OF ENVIRONMENTAL STUDIES, vol. 17, no. 2A, pp. 14-18, 2008.

[17] M. Hajder, M. Bolanowski and L. Byczkowska-Lipińska, "A set of connection network synthesis basis on the linear diophantine constraints solution in area {0,1}," in Ogólnopolskie Warsztaty Doktoranckie. Materiały konferencji, Nałęczów, 2008.

[18] M. Hajder, Methods and facilities increasing effectiveness of designing distributed systems based on multichannels. Dissertation for the degree Doctor of Technical Science, Kyiv: National Technical University of Ukraine "Kyiv Polytechnic Institute", 2006.

[19] J. Kunegis, "Exploiting the structure of bipartite graphs for algebraic and spectral graph theory," Internet Mathematics, vol. 11, no. 3, pp. 201-321, 2015.

[20] M. Hajder, J. Kolbusz and T. Bartczak, "Undirected graph algebra application for formalization description of information flows," in Human System Interaction (HSI), 2013 The 6th International Conference, Kraków, 2013.

[21] M. Hajder, H. Loutskii and W. Stręciwilk, Informatyka. Wirtualna podróż w świat systemów i sieci komputerowych, Rzeszów: Wydawnictwo Wyższej Szkoły Informatyki i Zarządzania w Rzeszowie, 2002.

[22] M. Hajder and P. Dymora, "A novel approach to fault tolerant multichannel networks designing problems," ANNALES UNIVERSITATIS MARIAE CURIE-SKŁODOWSKA, SECTIO AI: INFORMATICA, no. 1, pp. 66-77, 2011.

[23] М. Хайдер, "Множественный подход в системах каналов связи," Вісник Національного технічного університету України "Київський політехнічний інститут". Інформатика, управління та обчислювальна техніка, vol. 41, pp. 120-132, 2004.