

Reconfiguration of the multi-channel communication system with hierarchical structure and distributed passive switching

Piotr Hajder¹[0000-0002-0437-3328] and Łukasz Rauch¹[0000-0001-5366-743X]

¹ AGH University of Science and Technology, Krakow, Poland
{phajder, lrauch}@agh.edu.pl

Abstract. One of the key problems in parallel processing systems is the architecture of internodal connections, thus affecting the computational efficiency of the whole. In this work authors describe proposition of a new multi-channel hierarchical computational environment with distributed passive switching. According to authors, improvement of communication efficiency should be based on grouping of system components. In the first type of clustering processing nodes are combined into independent groups that communicate using a dedicated channel group. The second type of clustering splits channels available in the system. In particular, they are divided into smaller independent fragments that can be combined into clusters that support selected users. In this work, a model of computational environment and basic reconfiguration protocol were described. The necessary components and management of reconfiguration, passive switching and hierarchization were discussed, highlighting related problems to be solved. Main reconfiguration restrictions were specified, using and combining together two structural complexity measures: Complexity B Index and Efficiency Index.

Keywords: Reconfiguration, Distributed systems, Multi-channel architecture.

1 Introduction

In modeling modern scientific and technical problems there are often tasks with high computational complexity, such as simulations of quantum mechanics, multiscale modelling, weather forecasting, dynamics of chemical processes, 3D plasma modeling (e.g. in a nuclear fusion reactor) and so others. Their high time complexity makes it impossible to use sequential processing to solve them [1, 2]. In this case, the solution of such tasks must be based on the parallel (centralized or distributed) processing. In both cases, an important part of task solving cost is internodal communication, especially in the case of distributed computing. This means that methods of improving the efficiency of simulation task solving should be primarily seen in its parallelization, while improving communication architecture.

One of the basic methods of improving communication efficiency is to adapt the topology of the connection network to the traffic pattern, characteristic for task being solved. However, it should be remembered that these patterns will be radically different

for various tasks. Taking into account the advances in information technology, it can be noticed that implementation of computational procedures should take place on the basis of MPP (*Massively Parallel Processing*) or CoW (*Cluster of Workstations*) processing models. In both cases, the efficiency of computational process requires a modern subsystem of internodal communication, which due to the variety of tasks to be solved should be tunable, i.e. it should be possible to reconfigure connections in preferably real time.

The increase in the number of nodes in large-scale computing systems reduces the quality of their use expressed by:

- Unacceptable increase in communication costs necessary to solve a computational task;
- Lack of uniformity of load on processing nodes and communication channels;
- Inefficiency of system resources reconfiguration in production environment.

The solution to above problem required solving following tasks:

1. Unification of terminology used in the area of multi-channel connection system with multi-channeling at the logical (virtual) level;
2. Designing of components for a parallel computational system, ensuring flexible re-configuration of internodal connections;
3. Preparation and verification of analysis methods and synthesis of dedicated reconfigured computational architectures and their examination.

From the available literature [3-6] many interesting architectures of parallel computing systems are known. In most cases, they use factory multiprocessors computing modules, combined into larger, more efficient structures [7]. Scaling such a system is based on various virtualization and switching techniques. Similar and in many cases better results can be obtained by replacing centralized active switching with distributed passive switching. The solutions proposed in the work are based on the following basic assumptions:

The basis of communication will be a multi-channel passive optical environment, similar in its idea (but not identical) to GPON (*Gigabyte Passive Optical Network*) technology [8]. The implementation of proposed solutions may also be based on multi-channel communication network using FDM (*Frequency Division Multiplexing*) or similar method, however, the obtained characteristics will be worse.

2 The role of multi-channeling and hierarchization in communication efficiency improvements

In the 1980s, K. Hwang formulated the hypothesis that the most influencing limitations of parallel and distributed system functioning result from internodal communication [9]. A similar thesis appears in other publications [10-12]. The solution of some of defined problems is the construction of computational architecture based on dynamically reconfigured connections, wherein each node should be supported by a set of

logically independent channels. The basic ways to improve the efficiency of modern distributed are defined (fig. 1). The bold font indicates the ones that can effectively take advantage of the new communication environment.

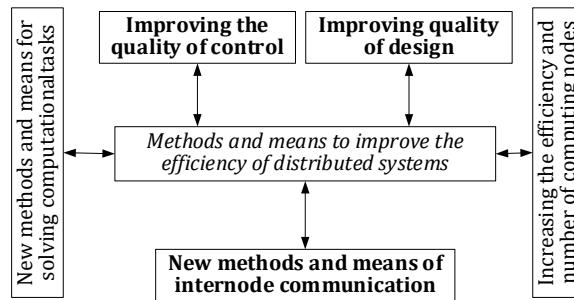


Fig. 1. Basic methods and means improving efficiency of distributed system

The concept of a reconfigurable computing environment is shown in fig. 2. Separated computing environments called clusters can be autonomous or connected to each other, as in the case of clusters 1 and 3. Connections between clusters are carried out by using the same means as internal connections and can be made as two-point, group or broadcast connections. Individual nodes (PN – processing, SN – storage) can be utilized in one cluster or shared between them. This enables information exchange between clusters through inter-process communication or shared memory areas (operational or mass). Besides the main functionality, storage nodes can provide an execution environment for I/O operations, not directly related to computations.

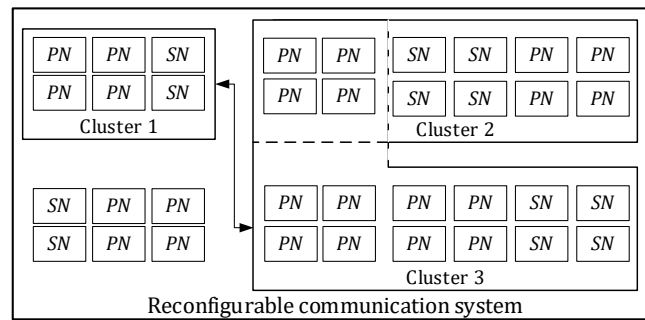


Fig. 2. Reconfigurable computing environment

According to fig. 1 and fig. 2, the improvement of communication efficiency should be based on grouping of system components (computing nodes or channels). In the first group, computing nodes, on the basis of the communication criterion, are combined into independent groups that communicate using a dedicated channel group. In addition to the traffic pattern generated by nodes forming a cluster, the division criteria may also include: classes of tasks solved within a given group, requirements regarding communication delays, error rates and others. The second type of clustering is based on

grouping the channels available in the system. In particular, they are divided into smaller independent fragments that can be combined into clusters that support selected users. For example, a separate group may be created by users using services insensitive to communication delays, another generating low traffic, yet another characterized by high explosiveness of generated traffic. The third method is a modification of previous ones and relies on the hierarchization of logical connection channels, where channels are combined into groups that support different sets of processing nodes, which can also be grouped. Due to the diversity of proposed methods, the architecture of connections can be adapted to the traffic patterns and proper communication requirements. Because in optical systems the change in the wavelength used by transceiver element takes milliseconds, adjusting the connection architecture to the current requirement of users can be dynamic and be implemented in real time. The classification of efficiency improvement methods in multi-channel architecture is shown in fig. 3. Each of the clustering methods is good and its selection depends on optimization criterion.

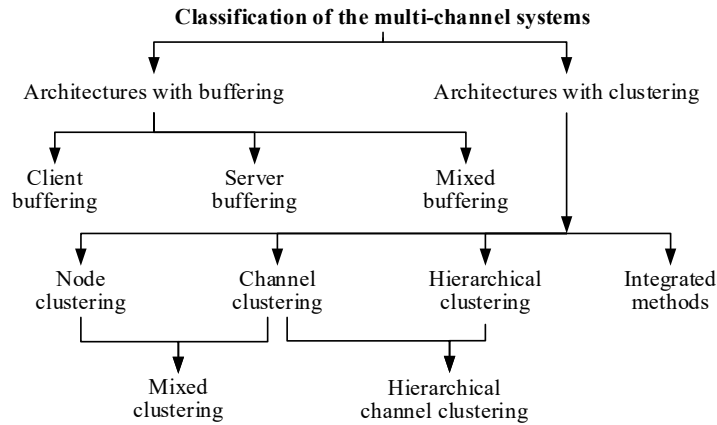


Fig. 3. Classification of efficiency improvement of multi-channel systems with clustering

In research on distributed computing systems, the concept of single- and multi-channeling are the most frequently occurring terms. Unfortunately, these terms are in many cases vague. Due to the fact that multi-channeling is the most important basic assumption, the own definition for both types of connections presented in this work were proposed.

Definition 1

Let m and n be the physical nodes of the communication system, and p and q its logical nodes built on the basis of physical nodes. If in any physical channel $[[m, n]]$ of the communication network only one logical channel $[[p, q]]$ is used, then the connections of the communication system are called **single-channel**.

Definition 2

Let m and n be the physical nodes of a computer system, $p_i, (i = 1, \dots, s_m^l), q_j (j = 1, \dots, s_n^l)$ – sets of logical nodes built on the basis of physical nodes m and

n respectively, where s_m^l, s_n^l are logical degree of m -th and n -th physical vertex. If in any physical channel $[[m, n]]$ more than one logical channel $[[p_i, q_j]]$ is used, then the connections in the system are called **multi-channel**.

Definition 1 shows that if in a system based on a set of physical channels, logical channels dependent of the physical environment are built, then such a system remains terminologically single-channel. According to second definition, in multi-channel system the physical communication environment is divided into an ordered set of independent logical channels. Logical degree s_n^l indicates the number of logical channels connected to n -th node.

The second basic solution used for the proposed communication environment is hierarchism, in particular hierarchical clustering. On the one hand, methods of the theory of hierarchical systems can be used to solve tasks of component selection and grouping [13], on the other hand it reflects the real conditions of functioning the backbone network. However, the effectiveness of existing methods may be irrelevant in a situation where processing operations are transferred to the edge of network.

In the available literature many successful applications of hierarchy can be found as means of improving communication efficiency in multiprocessor systems [14, 15]. In one of the papers, the author's proposal to introduce multi-level hierarchy of parallel system connections caused reduction in the number of point-to-point communication errors [16]. In turn, the hierarchization of computational clusters, including the manipulation of their size depending on the hierarchy level, and the separation (hierarchization) of communication channel groups favorably affects the level of latency, bandwidth and processing time of messages for each processor [17]. In the further work, a continuation of hierarchy approach can be noticed, with regard to reliability from the point of view of reconstructing interrupted computations in HPC architectures [18]. An additional advantage of this solution was the reduction of the overhead associated with message passing.

On the other hand, the hierarchy is also used in distributed systems, in particular the combination of Cloud, Fog and Edge Computing. However, entering in details, fog is a hierarchical structure [19], while the Edge is just heading in the direction where edge clusters are grouped against various criteria, including resource computational capabilities [20, 21]. It can therefore be expected that the hierarchy of system architecture components, i.e. nodes and communication channels, may bring beneficial effects due to operating parameters and structural measures of proposed architecture.

3 Reconfigurable communication environment

The model architecture presented in this work consists of two level hierarchy. The first level of computational system is a communication core with ring topology and consists of core nodes. They perform only communication operations and do not process information related to computational process. Following elements in each core node can be distinguished: communication processor, network interface (NI) and transceiver device. Each core node can be equipped with several transceiver devices supporting independent physical channels. In turn for the second level, computational system hierarchy

consists of processing nodes (PN), equipped with network interface and transceiver device. It is assumed that one physical channel will reach each of processing nodes.

The basic requirement for proposed computing environment is reconfiguration, guaranteeing the flexibility of connection network. Therefore, it is required to propose devices that enable reconfiguration to the full extent, both at the core level and edge devices, starting from direct connections, ending with broadcast communication. The set of all devices can be divided into two groups: passive switching devices with the possibility of being reconfigured (reprogrammed) and transmitting devices.

The first group of elements consists of passive switching devices. Each component has to establish static connections between logical channels during the reconfiguration process. Through this operation, switching during exploitation is carried out passively [22]. To guarantee all three modes of communication (uni-, multi- and broadcast), four classes of switching devices are singled out: coupling (CD), splitting (SD) and interconnected (ID). The device performing filtration can be integrated with other components, unless it can disturb or limit reconfiguration process. These devices are shown in fig. 4.

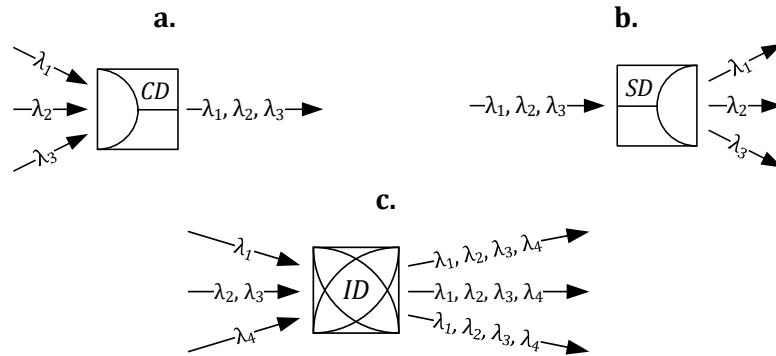


Fig. 4. Types of communication core devices. a. Coupling Device (CD); b. Splitting Device (SD); c. Interconnected Device (ID)

Presented elements belong to the network core, which mediates communication between processing devices. Regardless of their role in the system, each of them must be equipped with one or more communication processing units, a set of transceiver devices belonging to the second class of components in proposed architecture, and a network interface.

The next group are transceiver devices. All system processing components, including core elements, are equipped with a set of devices. Their main role is to send and receive packets. To enable communication in processing node, a dedicated network interface must be supplied, which has to appropriately convert the optical signal coming from many channels to an acceptable form. It is worth mentioning that in the general case the model is not limited only to the optical signal. It was only assumed that the transmission medium has to guarantee multi-channeling.

Among the transceiver element, there are two types of division:

- With respect to the transmission type: transmitting, receiving and transmitting-receiving (transceiver device);
- With respect to the nature of switching: fixed wavelength devices (=), adapted to a specific channel, and varying wavelength devices (~), capable of modulation.

Each of the transceiver devices currently supports exactly one logical channel, and therefore the number of logical channels of a physical device is equal to the number of transceiver devices available for a given node. It should be noted that from the point of view of connection flexibility, varying wavelength devices is more justified solution, but considering the construction costs and its reliability, fixed wave devices seems to be a more cost-effective solution.

With components defined, the relationships between them can be defined. The processing node having an appropriate communication module (transceiver devices and network interface NI) can be connected to the system. The general architecture of the communication system is presented in fig. 6. The network core has ring topology, but it is not imposed. It is used for illustrating the relationship between components of the system.

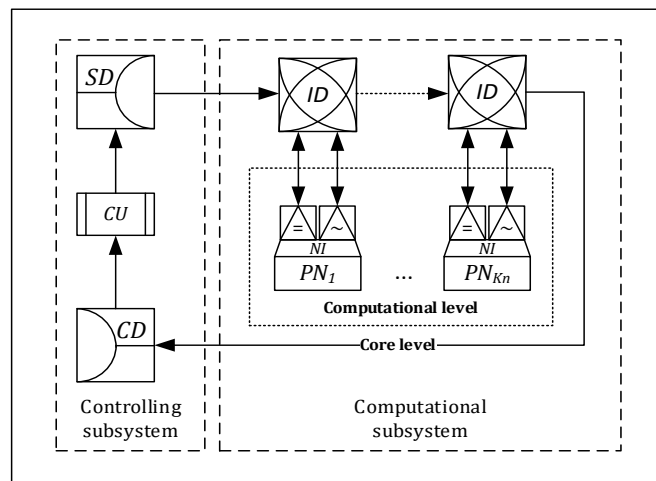


Fig. 5. Architecture of the computational system in a passive hierarchical communication environment

Two subsystems are singled out in the connection system: computational and control. In the computational subsystem K_n computational nodes are connected to passive switching core elements, the number of which is directly dependent on the number of logical channels required. Due to the flexibility of internodal connections, each device can be both core (ID) and computational node (PN). In addition, the properties of the proposed architecture make it ideal for edge computing systems, including sensor networks and IoT [23, 24].

The proposed internodal connection architecture is not critical from the point of view of medium access protocols. These can be both autonomous and centrally controlled,

deterministic or probabilistic. The protocols used can organize a channel with unicast, multicast or broadcast traffic. In previous work aimed at building a prototype of device, a central control unit (CU) connected by a common dedicated logical channel to each node of the system was used. From the CU level, decisions about the architecture of connections are made. Such a solution is particularly effective if the system dynamics is moderate, which is assured for tasks being solved now. In the future, it is planned to create a set of protocols ensuring autonomous implementation of the topology reconfiguration procedure. It is possible that processing nodes will be connected with a dedicated control channel, however the control unit will no longer be used in the system.

4 Basic reconfiguration algorithm

Reconfiguration in the architecture is distributed – both core and endpoint devices are the subject of the process. Core devices do not perform computational tasks, but only control and monitor the reconfiguration process and determine the optimal logical paths between devices.

For reconfiguration purposes, an additional channel may be allocated, in which information about changes that must be performed in each individual transceiver devices will be sent. Situations that may cause reconfiguration are, among other:

- Computational group requires an additional component (the component may be a single device, a group of devices with predefined topology or supercomputer);
- Detection of damage to the communication channel or device;
- Inefficiency of current connection architecture.

The occurrence of any of reasons for reconfiguration causes core devices to search for the optimal solution and as a result a set of logical paths is obtained, which must be added, changed or removed. Among the optimization criteria, one can distinguish in particular: communication delays, reliability of the communication between devices and minimal acceptable bandwidth.

Having a new set of logical paths prepared, a message is sent to each of devices, containing dedicated tuning rules for transceiver elements. The message reaches all devices so that they can assess whether the tuning process interferes with their communication paths.

When each of the physical nodes participating in reconfiguration reports its readiness, then in any logical path which includes any element from the set of reconfigured physical nodes, communication should be stopped. The standard solution used in classical network architectures in such case is buffering: during reconfiguration, each device can buffer traffic on the active parts of the path (fig. 3). With low saturation of the physical channel, creating new connections may not require stopping communication in any of available paths. It can therefore be concluded that an important factor influencing the optimization of connection paths is the degree of introduced changes in relation to the underlying topology, which affects the selection of optimization methods.

After reconfiguration, each device reports its readiness to work. After obtaining mutual consent, the core sends information about the transition to the work state, fully

restoring the communication to all nodes again. The core itself returns its original state of monitoring the connection architecture.

The basic reconfiguration algorithm in the proposed architecture has been presented in fig. 6 in the form of a flowchart.

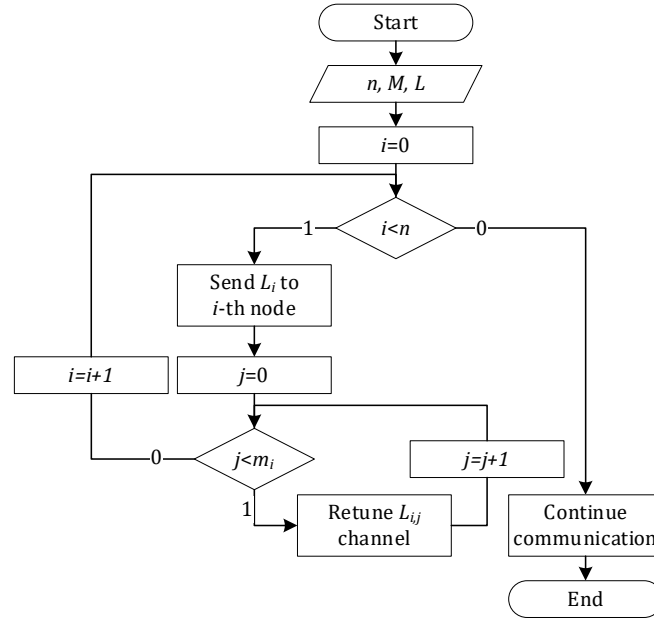


Fig. 6. Reconfiguration flowchart. Denotations: n – number of physical nodes being retuned; m_i – number of changes in i -th node, $m_i \in M$, where $M = (m_0, \dots, m_{n-1})$; L_i – list of changes in i -th node, $L_i \in L$, where $L = (L_0, \dots, L_{n-1})$

Optimization process generates following data: list of n physical nodes to be reconfigured and L_i lists of reconfiguration messages, describing changes in i -th node's communication channels. Each reconfiguration message $L_{i,j} = \{p, q, \lambda\}$ contains information about physical transmitting node p (entry point), physical receiving node q (exit point) and logical communication channel λ between p and q nodes (e.g. a single wavelength or part of the band in optical fiber). There are three possible cases, which can take place during reconfiguration of a single node (retunePath in fig. 6):

- Addition of new communication channel if $\lambda \neq \emptyset$ and path $\llbracket p, q \rrbracket$ between nodes does not exist;
- Modification of existing channel if $\lambda \neq \emptyset$ and path $\llbracket p, q \rrbracket$ between nodes exists;
- Deletion of existing connection between nodes if $\lambda = \emptyset$.

The message $L_{i,j}$ can be sent to any type of device. If any of the physical nodes (p, q) is identical to the node being reconfigured, it means that the node is a computational device (endpoint). Otherwise, it is a core device.

5 Restrictions on reconfiguration

The reconfiguration, however, has some technological and cost limitations. The maximum number of channels coexisting in the communication medium is technologically limited and at the moment it is about 2000 in the case of optical fiber [25]. The increasing number of channels also affects the complexity of architecture, which makes real-time reconfiguration difficult: more time is required to detect a triggering factor and to compute a new logical topology. In addition, the cost of constructing multi-channel architecture increases non-linearly with the increasing number of channels served.

The second important aspect is signal attenuation, occurring with an excessive number of devices connected to one channel or cluster. There may be a downtime in communication between two or more devices if a channel is overloaded, referring to low power signal source. The solution to this problem may require providing a signal regenerator, thus adversely affecting the costs of construction. The transceiver devices have a direct impact on the attenuation. For a low cost architecture, low energy transmitters, e.g. semiconductor lasers, should be used, which in turn limits the maximum range. The main conclusion from considerations is identification the number of channels as a bottleneck in the synthesis process of reconfigurable multi-channel architecture, which strongly influences complexity and construction costs of the system.

One of the possible approaches for complexity estimation are structural complexity measures. They can be applied during synthesis process of a given architecture. In this work, two hierarchical multi-channel architectures were proposed and analyzed for structural complexity using two measures: Complexity B Index (CBI) and Efficiency Index (EI) [26]. For indices calculation, the QuACN package in R was used [27]. Connection architectures were represented as graphs and generated as adjacency lists. Exemplary multi-channel topologies were presented in fig.7 and fig.8. Symmetrical architectures were analyzed for simplicity. Following denotations were applied: B - number of channels, α - number of nodes per cluster, n - number of clusters.

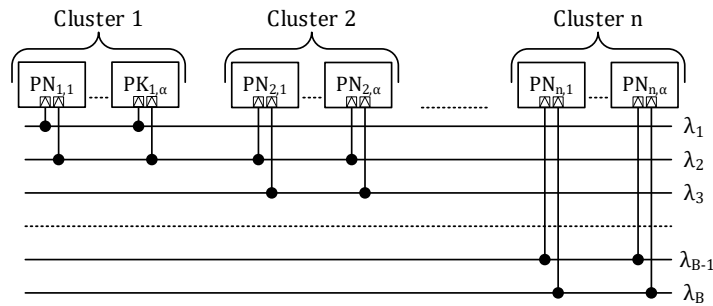


Fig. 7. Exemplary architecture topology with fully connected nodes between neighboring clusters, denoted as *Architecture A*

In the next step, structural complexity measures were calculated. Results are presented in fig. 9. Each architecture consists of 1024 nodes, with changing number of nodes per cluster (and hence the number of clusters).

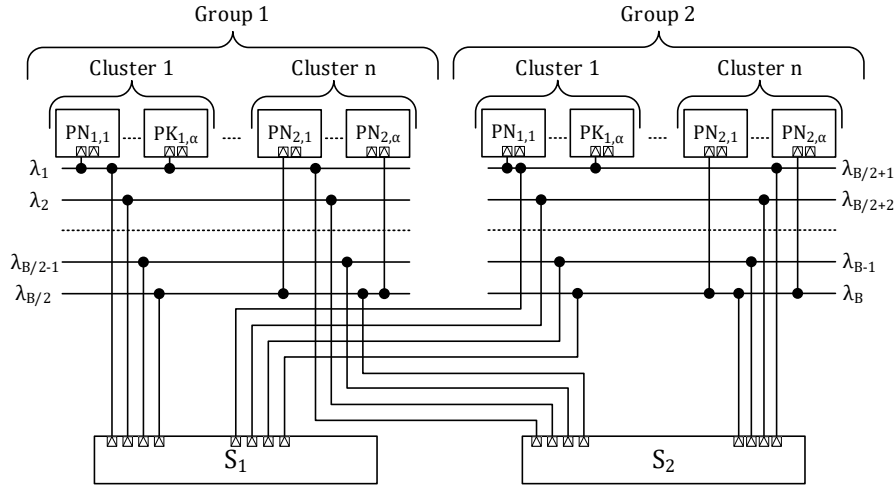


Fig. 8. Exemplary architecture topology with clustering of nodes and channels with two groups and two server devices, denoted as *Architecture B*

In the first topology complete connections within each clusters and between neighboring clusters were proposed. Appropriate neighborhood can have influence on load balance in the system. The second one is similar to the dragonfly topology, which is used in MPP systems [28]. The devices are divided into two disjoint groups, thus implementing channel hierarchy. Intergroup communication is possible through server devices. On the other hand, hierarchization of nodes was carried out in the same way as in first topology – by node clustering.

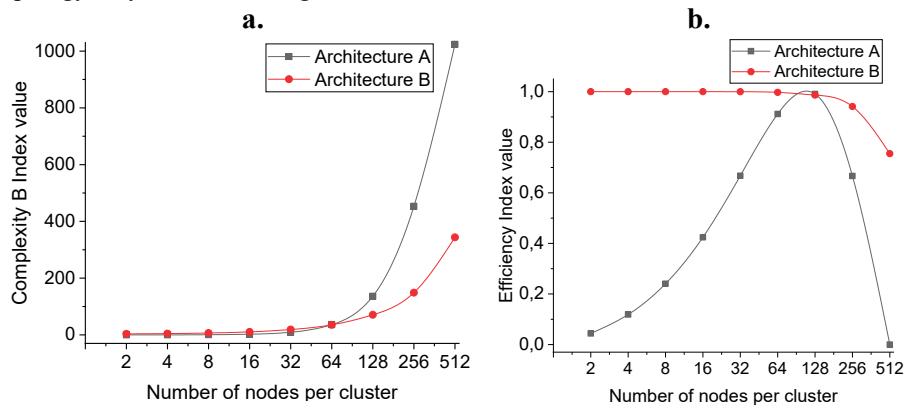


Fig. 9. Structural complexity measures calculated for proposed topologies: **a.** Complexity B Index B; **b.** Efficiency Index

CBI takes into account vertex degree and fairness to calculate the measure. Complexity of the system is proportional to the CBI value. Branches and cycles increases CBI value, for clique is equal to number of nodes. Values close to zero are obtained for linear or

monocyclic topologies. In the case of EI, its increasing value raises general communication efficiency, i.e. reliability, resilience, bandwidth, resulting from the occurrence of multiple paths connecting two nodes, but the complexity of the system also increases. Values is equal to one for clique and zero for isolated nodes. Both measures depend directly on the number of edges in the model graph. Thus, the increase in complexity of the communication system increases the number of B logical channels, necessary for establishing the connections.

On the basis of the CBI value it can be concluded that for a small α architecture A is characterized by lower complexity. However, above $\alpha = 64$ architecture B has lower CBI value, reducing the number of channels B .

The main and noticeable difference in EI values is the complexity for small α . Architecture B from the very beginning is characterized by high efficiency, expressed in length (cost) of the path between two nodes. Bearing in mind the CBI results, it can be determined that the system is characterized by complexity with an average number of edges. In turn, the low EI value for architecture A with small α results from the path length between nodes of cluster 1 and n .

Analyzing the dependence of structural measures on the number of clusters it can be noticed that the hierarchy decreases complexity of the connection system. It can be concluded that during the reconfiguration, clusters with a reduced number of nodes should be considered. Very complex structures are more difficult to reconfigure and require more time for detection of the triggering factor.

6 Summary

On the basis of conducted research, it can be concluded that the use of multi-channel environments in the implementation of hierarchy in computing system components can bring measurable benefits in the form of delay reduction in selected network segments by adapting the architecture to the current operating conditions and traffic pattern. Introduction of hierarchy has so far been successfully used, both in the case of parallel and distributed systems. Therefore a beneficial effect on the system's reliability and overall performance can be expected. However, before the implementation of such environment occurs, a number of tasks should be solved, e.g. reconfiguration conditions or formalization of the path selection process.

During the generation of graphs, it was noticed that the classic representation of the connection network as a vertex graph, where vertices represent nodes and edges communication channels, is inefficient and does not fully reflect the actual architecture, because links to the communication channels are thus omitted. In further research, other graph representation methods will be tested, e.g. hypergraphs and total graphs, with fixed number of channels.

The idea incorporated in the authors' research assumes that the result of their work will be development of a low cost micro-supercomputer characterized by flexibility of the connection system. Therefore, most important works planned include:

1. Improving the scalability of solution, in particular increasing the number of supported logical channels. In general, improving the efficiency of the use of processing

- nodes and communication channels will be based on clustering (grouping) and hierarchization of system components: nodes and channels;
2. Due to NP-hard complexity of reconfiguration task, it is necessary to develop optimization algorithms, enabling core devices optimal communication paths finding (regarding the appropriate optimization criteria), in particular considering number of changes in logical topology;
 3. Computational nodes forming micro-supercomputer, based on Raspberry or Arduino technologies, will be equipped with tunable optical transceivers, settings of which will decide on the architecture of connections – its change will always involve the necessity of retuning the set of transceiver devices;
 4. Preparation and implementation of algorithms for load balancing of computational nodes and channels;
 5. Preparation of the basic scalable processing module used to build system of any size;
 6. Expanding the scope of applications for proposed architecture.

The proposed environment can be particularly interesting in optimization of communication environment in tools dedicated for containerization and cluster management, such as OpenShift, Mesos, Docker or Kubernetes.

Acknowledgement: 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

References

1. L. Rauch, D. Bachniak, Dynamic load balancing for CAFE multiscale modelling methods for heterogeneous hardware infrastructure, *Procedia Computer Science* 108C, pp. 1813-1822, 2017.
2. L. Rauch, K. Bzowski, M. Pietrzyk, D. Krol, R. Slota, J. Kitowski, Application of HPC based VirtRoll system to design laminar cooling of strips according to material properties in the coil, Tenth ACC Cyfronet AGH HPC users' conference, pp. 25-26, 2017.
3. A. Pajankar, *Raspberry Pi Supercomputing and Scientific Programming*, Apress, 2017.
4. T. Sterling, M. Anderson and M. Brodowicz, *High Performance Computing: Modern Systems and Practices*, Morgan Kaufmann, 2018.
5. InfiniBand Trade Association, [Online]. Available: <https://www.infinibandta.org/>. [Accessed 5 January 2019].
6. D. Gritzalis, M. Theocharidou and G. Stergiopoulos, *Critical Infrastructure Security and Resilience*, Springer, 2019.
7. R. Trobec, R. Vasiljević, M. Tomasević, R. Milutinović and M. Valero, Interconnection Networks in Petascale Computing Systems: A Survey, *ACM Computing Surveys*, vol. 49, no. 3, 2016.
8. M. Radivojević, P. Matavulji, *The Emerging WDM EPON*, Springer, 2017.
9. K. Hwang and J. Ghosh, Hypernet: A communication-efficient architecture for constructing massively parallel computers, *IEEE Transactions on Computers*, vol. C-36, no. 12, pp. 1450-1466, 1987.
10. B.-J. Chang and R.-H. Hwang, Performance analysis for hierarchical multirate loss networks, *IEEE Journals & Magazines*, pp. 187 - 199, 2004.

11. M. D. Mesarovic, Multilevel systems and concepts in process control, *IEEE Journals & Magazines*, vol. 58, no. 1, pp. 111-125, 1970.
12. M. V. Gubko, *Mathematical models of optimization of hierarchical structures*, LENAND, 2006.
13. B. Korte and J. Vygen, *Combinatorial Optimization. Theory and Algorithms*, III ed., Berlin: Springer-Verlag, 2006.
14. M. Hajder, M. Bolanowski, *Connectivity Analysis in the Computational Systems with Distributed Communications in the Multichannel Environment*, *Polish Journal of Environmental Studies*, vol. 17, no. 2A, pp. 14-18, 2008.
15. J. A. Rico-Gallego, J. C. Diaz-Martin, R. R. Manumachu and A. L. Lastovetsky, *A Survey of Communication Performance Models for High-Performance Computing*, *ACM Computing Surveys*, vol. 51, no. 6, 2019.
16. L. Yuan, Y. Zhang, Y. Tang, L. Rao and X. Sun, *LogGPH: A Parallel Computational Model with Hierarchical Communication Awareness*, 13th IEEE International Conference on Computational Science and Engineering, 2010.
17. F. Cappello, P. Fraigniaud, B. Mans and A. L. Rosenberg, *An Algorithmic Model for Heterogeneous Hyper-Clusters: Rationale and Experience*, *International Journal of Foundations of Computer Science*, vol. 16, no. 2, pp. 195-215, 2005.
18. L. Bautista-Gomez, T. Ropars, N. Maruyama, F. Cappello and S. Matsuoka, *Hierarchical Clustering Strategies for Fault Tolerance in Large Scale HPC Systems*, *Proceedings of the 2012 IEEE International Conference on Cluster Computing*, 2012.
19. F. Bonomi, R. Milito, J. Zhu and S. Addepali, *Fog Computing and Its Role in the Internet of Things*, *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, New York, 2012.
20. A. J. Ferrer, J. M. Marques and J. Jorba, *Towards the Decentralized Cloud: Survey on Approaches and Challenges for Mobile, Ad hoc, and Edge Computing*, *ACM Computing Surveys*, vol. 51, no. 6, 2019.
21. S. E. Kafhali and K. Salah, *Efficient and Dynamic Scaling of Fog Nodes for IoT Devices*, *Journal of Supercomputing*, vol. 73, pp. 5261-5284, 2017.
22. X. Li, Z. Shao, M. Zhu and J. Yang, *Fundamentals of Optical Computing Technology*, Springer, 2018.
23. W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu, *Edge Computing: Vision Challenges*, *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637-646, 2016.
24. O. Michail and P. G. Spirakis, *Elements of the Theory of Dynamic Networks*, *Communications of the ACM*, vol. 61, no. 2, pp. 72-81, 2018.
25. G. P. Agrawal, *Optical Communication: Its History and Recent Progress*, in *Optics in Our Time*, Springer, pp. 177-199, 2016.
26. A. Kamisiński, P. Chołda and A. Jajszczyk, *Assessing the Structural Complexity of Computer and Communication Networks*, *ACM Computing Surveys*, vol. 47, no. 4, 2015.
27. QuACN: Quantitative Analysis of Complex Networks [Online]. Available: <https://rdrr.io/cran/QuACN/>. [Accessed: 1 April 2019].
28. M. Mubarak, C. D. Carothers, R. B. Ross, P. Carns, *Enabling Parallel Simulation of Large-Scale HPC Network System*, *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 1, pp. 87-100, 2017.