

# The Model of Device Community Forming Problem for the Geographically-Distributed Information and Control Systems Using Fog-computing Concept

Eduard V. Melnik

Information Technologies and  
Control Processes department  
Southern Scientific Center of the  
Russian Academy of Sciences (SSC  
RAS)  
Rostov-on-Don, Russia

Anna B. Klimenko

Scientific Research Institute of  
multiprocessor computing systems  
Southern Federal University  
Taganrog, Russia  
Anna\_klimenko@mail.ru

Donat Ya. Ivanov

Scientific Research Institute of  
multiprocessor computing systems  
Southern Federal University  
Taganrog, Russia

**Abstract**—This paper focuses on the fog-computing concept usage in the geographically-distributed information and control systems. Fog-computing concept improves the dependability of the information and control system, system latency and the communicational environment load. The key issue of fog-computing application to the information and control systems is the formation of device communities to solve management tasks. In the current paper the new device community forming problem model is presented and described as a multicriteria optimization problem. A brief model discussion is given, and some problem solving techniques are described.

**Keywords** —*fog-computing, information and control system, dependability, reliability, device community forming.*

## I. INTRODUCTION

Information and control system (ICS) is an integral part of a large amount of complex robotic and mechatronic objects, geographically distributed, in particular. Besides this, the trend of stand-alone mechatronic object integration under the ICS management is in usage nowadays. It allows to utilize computational resources more efficiently and to increase the general potential of such objects[1].

Complex mechatronic objects are used in nuclear energy production, oil and gas production, transportation, robotics, and production with aggressive environments, where the humans are not supposed to locate, but in case of failure the economic and environmental losses can be significant[2,3].

The good example of such systems is an oil-production. For the oil wells exploitation the complex mechatronic objects on the basis of electric centrifugal pumps (ECP) are used. The ECP downtime leads to the economic losses from 250000 to 1000000 rubles[4].

The progress in the microelectronic and sensor device area allows to provide such devices with a network interface. It leads to the possibility of integration of all types of sensors, computational units and actuators into the network. Such

approach to the ICS design is called the network-centric one [5].

The network-centric approach to the ICS design and implementation provides the possibility to re-distribute the functional tasks among the system elements, and so increase the system dependability level. The number of spare elements can be rather small, or the performance redundancy can be used [6,7].

Contemporary ICSs base on the distributed computations. The typical simple ICS structure is given in Fig.1.

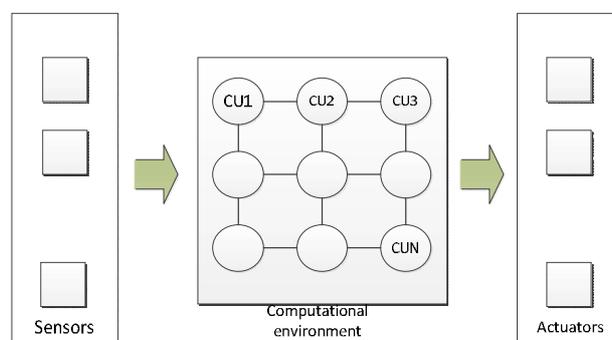


Fig. 1. The ICS general scheme

If sensors are located remotely from the computational environment, and amount of data to be processed is large, the following issues are of current interest:

- System latency. The main reasons are the network hops and the large amount of data to be transferred to the computational environment;
- High computational environment load, which leads to the reliability function decreasing;
- High communicational environment load, which leads to the reliability function degradation too;
- Low system scalability.

The fog-computing concept is quite new and supposed to solve the latency, scalability and other problems of Internet of Things [8-10]. Indeed, in the conditions of data exchange with cloud, it is expedient to delegate some computations to the nearest devices.

It is useful to mention that fog computing and edge computing are the different terms, nevertheless they are frequently confused [11].

Fog computing pushes intelligence down to the local area network level of network architecture, processing data in a fog node or IoT gateway.

Edge computing pushes the intelligence, processing power and communication capabilities of an edge gateway or appliance directly into devices like programmable automation controllers.

As it is impossible to fully delegate the process of decision making in ICSs to the edge of the network, the fog computing concept is rather useful to delegate some preliminary data processing from the CE to the sensors and other network devices nearby. Similar idea was proposed in [12].

The obvious advantages of the fog computing concept usage are:

- System latency decreasing;
- Computational environment load decreasing (and so the reliability function improves);
- Communication network load decreasing;
- System scalability improving.

With all these advantages the key issue of such computations is the **dynamic device community forming**. The requirement of dynamics is due to the some uncertainty of the computational resources presence. Devices of the edge of the network can be busy with another computational tasks, and, besides this, can be disfunctioning or simply turned off.

The possibility of the similar problem solving is presented in details in studies [13-15]. In [14] the efficiency of the task distribution among the set of computational units by the parallel simulated annealing is proved experimentally (10-25 iterations for the one-criterion problem to find a viable solution). Techniques of the solution quality improving from the load balancing point of view through the criteria number reducing is given in [16].

The problem of the device community forming (DCF) for the preliminary data processing in the geographically distributed ICSs is in the focus of this paper. The following sections of the paper contain:

- A new formal model of DCF description;
- A brief DCF problem model discussion;

A review of the solution search techniques for the similar problem solved in the previous work.

## II. A GENERALIZED DCF PROBLEM MODEL FOR THE DATA PRE-PROCESSING IN THE GEOGRAPHICALLY DISTRIBUTED ICSs WITH FOG-COMPUTING CONCEPT USAGE.

The to continue, some new terms must be defined.

**Preliminary data processing device (PDPD)** is a device, which can participate the preliminary data processing as a member of device community.

**Preliminary data processing task (PDPT)** is a set of operations, which must be performed before the pre-processed data is sent to the **Computational Environment (CE)**.

Also, the **computational environment** is defined as a set of computational nodes, where the final data processing is performed. It is supposed, that the distance between CE and PDPDs is more than one transitive network fragment.

The DCF problem model bases on the following assumptions:

1. We presuppose some degree of uncertainty of the PDPD number, but if PDPD solves a task, it must finish it.
2. All PDPDs can exchange the information about their states.

Let's take a look on the preferable criteria for the optimization problem under consideration.

1. It is expedient to minimize the community cost. This criterion relates to the possibility and necessity to solve a large amount of PDPTs in a distributed manner on the edge of network.

2. As PDPT can be performed in a distributed manner, its structure can be described as an information graph. In such graph vertexes are the sub-operations, and ribs are the information flows between sub-operations. So, the next criterion is the amount of data exchange during the PDPT performing with the condition of sub-operation distribution through the device community.

3. The criterion of device load is due to the dependence between the load and reliability function [7]. The dependency is exponential, so, the load minimization is rather desirable component of the general objective function.

The DCF problem can be preliminary formulated as follows: having PDPT parameters, planning PDPT completion time and PDPD identifier given, the device community and the PDPT subtasks performing schedule must be formed with completion time constraint and previously formulated criteria optimization.

The formal model of DCF problem is presented below.

A set of PDPDs is incorporated to the network of the arbitrary topology.

PDPD is characterized by the tuple given below:

$$\langle j, p_j, E_j, \{R_{jk}\}, \{F_{jk}\}, u_j, s_j \rangle,$$

Where

$j$  – PDPD identifier;

$p_j$  – PDPD performance;

$\{R_{jk}\}$  – the list of distances between PDPDs. Here the distance is the number of transitory network segments.

$\{F_{jk}\}$  – the list of system PDPD computational resources. It must be refreshed periodically through the system operating stage;

$u_j$  – available computational resource of the PDPD  $j$ ;

$s_j$  – the cost of PDPD  $j$ ;

$v$  – data transmission speed, in order to simplify the problem model, it is identical for all network segments.

The next step for our DCF problem model formalization is the description of the PDPTs.

The input data for a PDPT can be some variations of data, which must be pre-processed by the device community.

For instance, it can be an image batch, or a video batch, or huge amount of numbers, generated by sensor. We assume that the types of such data is predefined, so the software components can be constant too. The programs of the data preprocessing are described formally as a set of graphs, and each graph can be performed in a distributed manner.

Lets PDPT  $i$  to be described by a graph  $g_i$ , where the vertexes are the code fragments with computational complexity  $a_k$ , ribs are the data flows between code fragments,  $w_{dk}$  – the amount of data which must be transferred between code fragments  $k$  and  $d$ . Here the auxiliary variable must be defined:  $f_{dk}=1$ , if data is transferred through the network, and  $f_{dk}=0$ , if code fragments are performed on the same device.

The set of PDPTs to be preprocessed is  $G=\{g_i\}$ . Every PDPT has its  $t_i$ , time of the preprocessing beginning, and  $T_i$ , planned completion time.

The key idea of the fog-computing is the system latency minimization by the computation distribution among the neighbor devices. So, while device community forming, it is expedient to choose devices with minimum distance between them.

It must be noted, that at the moment  $t_i$  PDPD  $j$  can have insufficient computational resources  $u_{ij}$ . This computational resource can be insufficient to solve any code fragment  $a_k$  of the PDPT  $i$ , and so the task must be given to another PDPDs.

There are two main ways of such situation development:

- $j$ -identifier is connected to the PDPT description and the PDPT result. It is translated to the CE after the PDPT solving. In this case PDPT can migrate everywhere from the initial device, but, obviously, the community must not be too far. If it is so, all pros of the fog computing concept usage are eliminated.
- If PDPD has not got enough computational resources, it can give its PDPT to the community, but always waits the result and send it to the CE itself. In this case the distance between initial PDPD and community formed must be as short as possible too, as was mentioned in previous case.

Generalizing, we presuppose, that in all cases it is expedient to form the device community somewhere nearby the initial PDPD. So we formed the additional optimization criterion: the distance between the community participants.

Situations, presented above, illustrated in Figs.2,3,4.

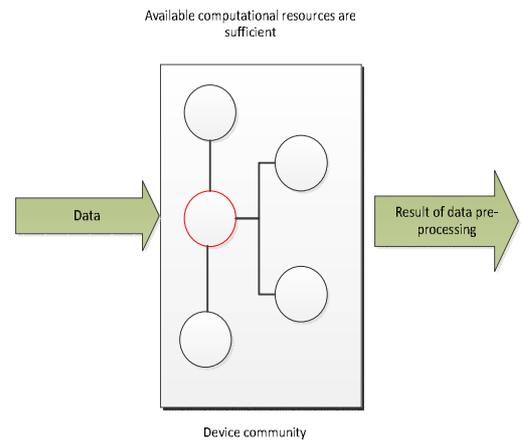


Fig. 2. The community forming with available resources

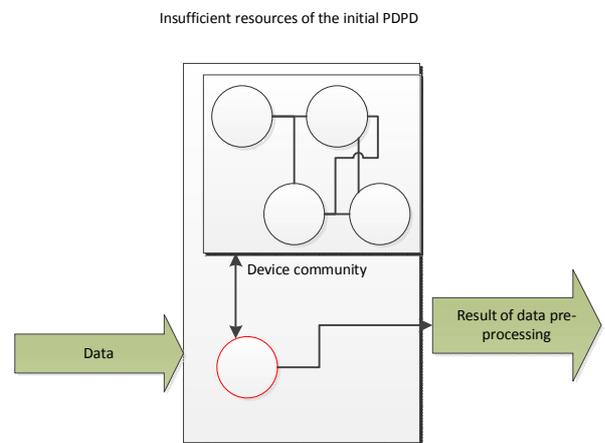


Fig. 3. The community forming with insufficient resources

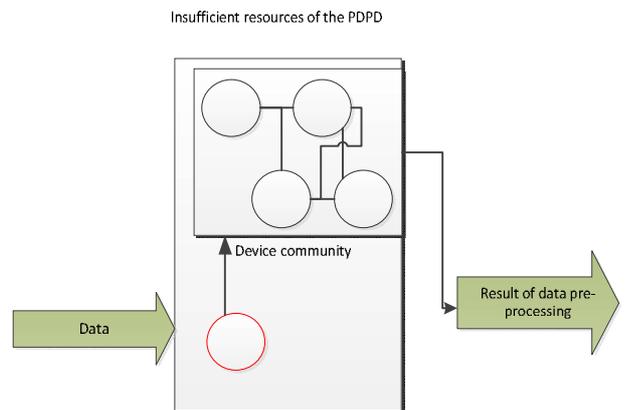


Fig. 4. The community forming with insufficient resources and task PDPT migration

The identifier of the initial PDPD, which must preprocess PDPT,  $J_{cur}$ .

Then, the formal description of the criterion of the distance between community devices can be presented as follows:

$$F_1 = \max_k R_{J_{cur,k}} \rightarrow \min \quad (1)$$

The criterion of the data flows between code fragments, transferred through the network, can be presented as:

$$F_2 = \sum_{k=1}^M w_{dk} \cdot f_{dk} \rightarrow \min, \quad (2)$$

where M is the number of devices in community.

3. Community cost criterion:

$$F_3 = \sum_{j=1}^M s_j \rightarrow \min, \quad (3)$$

where  $s_j$  is the cost of PDPD  $j$  from the community.

4. PDPD load intense:

$$F_4 = (1 - u_j) \rightarrow \min, \forall j \subseteq M. \quad (4)$$

The first constraint is a completion time constraint:

$$\forall k, j \max(t_i + \frac{a_k}{p_j u_j}) \leq T_i. \quad (5)$$

Besides this, every subtask  $a_k$  can not be solved earlier than its predecessor in the graph  $g_i$ :

$$\forall k, j, t_i + \frac{a_i}{p_j u_j} + \frac{w_{lk}}{v} < t_i + \frac{a_k}{p_j u_j} \quad (6)$$

As the available resources are known in the system, each computational resource related to the PDPD takes one subtask  $a_k$  to perform.

So, the final DCF problem can be formulated as following: with given PDPT parameters  $g_i$ , PDPT planned time completion  $T_i$ , PDPD with initial data identifier  $j$  and available computational resources  $u_j$ , the subset of PDPD and subtask  $a_k$  of  $g_i$  distribution among the PDPD community must be formed so that constraints (5,6) are satisfied and the vector objective function  $F=(F_1, F_2, F_3, F_4)(1,2,3,4)$  is minimized.

### III. THE PROBLEM SOLVING TECHNIQUES BRIEF OVERVIEW

The DCF problem is a multicriteria optimization problem with constraints. It is useful to mention, that in case of one criterion – the community cost – there are well-known and widely used solving techniques[17], including heuristic

scheduling algorithm for the heterogeneous computational systems [17].

But the problem, formulated in the scope of this paper is a multicriteria one, and all criteria in general case are expedient. So, the following, also well-known techniques can be used for DCF problem solving:

- scalarization of the problem (should be done with the expert support);
- Transformation of the multicriteria problem to the one-criterion. In this case selected criteria are transformed to the constraints. Changing the constraints, the simulation can be done, and the best solution are chosen.
- The search of the Pareto front.

When the DCF problem is in general form, and there are no any preferences related to the particular area of ICS usage, it is rather useful to solve a DCF problem as a multicriteria problem to get all possible solutions. But, of course, this method has the highest computational overheads.

As is shown in studies [13,14,18], first two methods can be applied to the operational stage of the ICS through the parallel asynchronous search. Simulated annealing technique with the “quenching” temperature scheme allows to get quite viable solutions in an acceptable time periods, as is shown in Fig.5.

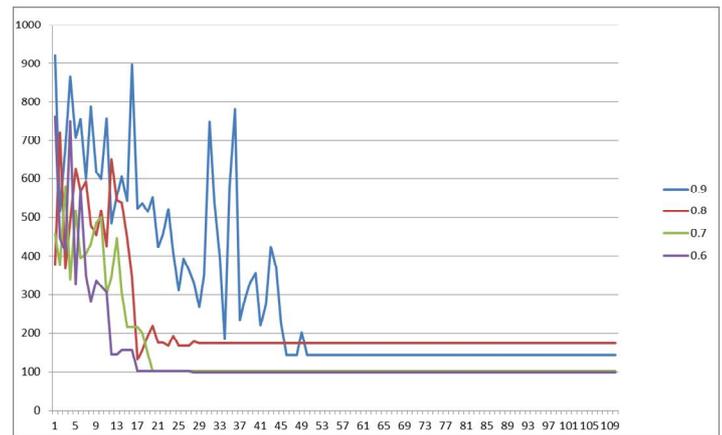


Fig. 5. The simulated annealing search example for the one-criterion problem of task distribution among the computational nodes. At the right side of the picture the quenching ratios are given.

It is seen in Fig.5, that simulated annealing with fast temperature decreasing does get acceptable solutions in 10-25 algorithm iterations.

### IV. CONCLUSION AND FUTURE WORK

The key issues of the contemporary geographically distributed ICSs are:

- System latency, due to the transitory network fragments and increase of data amount;
- Low reliability of the computational environment due to the high computational load;
- Network system low reliability due to the high load;
- Low system scalability.

In the current paper the fog-computing (or edge-computing) concept usage is proposed to preprocess sensor data before its transition to the computational environment. The generalized DCF problem is formulated, the optimization criteria are proposed and described. The brief analysis of the problem solving techniques is presented. Based on the previous experience in the similar problem solving the following conclusion is made: the DCF problem can be solved successfully during the ICS operational stage with the parallel simulated annealing search techniques.

The fog-computing concept usage for the geographically distributed ICSs seems to be expedient and encouraging. It decreases system latency, improves system reliability, dependability, and scalability, and also decreases the energy consumption due to the communication network load optimization.

As for future work, the authors of this paper distinguish at least two ways of research and development:

- to formulate the mathematical model of DCF problem more precisely with emphasize on the peculiarities of ICS usage areas;
- to evaluate pros and cons of the PDPT distribution methods.

#### ACKNOWLEDGMENT

The reported study was funded by SSC RAS projects 02562014-0008, 0256-2015-0080 within the task 007-01114-16 PR.

#### REFERENCES

- [1] I. A. Klyuchev et al. Programmnoye obespecheniye vstroennih vychislitel'nykh sistem. SPb. SPbGU ITMO. 2009, 213 p.
- [2] J Zhang, Y., Jiang, J.: Bibliographical review on reconfigurable fault-tolerant control systems. *Annual Reviews in Control*, 2008, V.32(2), pp. 229–252.
- [3] Cerstani D., Godary-Dejean K. Fault-tolerance in control architectures for mobile robots: fantasy or reality? In proc. Of 7<sup>th</sup> national conference on Control Architectures of Robots, 2012, Nancy, France.
- [4] Korovin I.S. Osobennosti vnedreniya sistemy podderzhki prinyatiya resheniy po control'yu sostoyaniya UETsN na osnove neuronnoy seti: prakticheskiye rezultaty, perspektivy razvitiya // *Izvestiya UFU. Technicheskiye nauki*. 2006, vol .71, № 16.
- [5] Melnik E.V. et al. A Novel Approach to Fault Tolerant Information and Control System Design // *5-th Int. Conf. Informatics, Electron. Vis. Dhaka, Bangladesh: University of Dhaka*, 2016.
- [6] Klimenko A.B., Ivanov D.I., Melnik E. V. The configuration generation problem for the informational and control systems with the performance redundancy // *Ind. Eng. Appl. Manuf. (ICEAM), Int. Conf.* 2016, pp. 1–5.
- [7] Melnik, E.V. Simulation options for redundancy in distributed information and control systems with a decentralized organization / E.V.Melnik, G.V.Gorelova // *Proceedings of SFU. Ser. Technical science*. 2013, №3, pp.184-193.
- [8] Yi S. et al. Fog computing: Platform and applications // *Proc. - 3rd Work. Hot Top. Web Syst. Technol. HotWeb 2015*. 2016, pp. 73–78.
- [9] More P. Review of implementing fog computing // *Int. J. Res. Eng. Technol.* 2015, Vol 4, № 6, pp. 2319–2322.
- [10] Stojmenovic I., Wen S. The Fog Computing Paradigm: Scenarios and Security Issues // *Proc. 2014 Fed. Conf. Comput. Sci. Inf. Syst.* 2014, Vol. 2, pp. 1–8.
- [11] <https://www.automationworld.com/fog-computing-vs-edge-computing-whats-difference>
- [12] Raspredeleonnaya informatsionno-upravlyauschaya Sistema na osnove intellektualnih datchikov: p. m. 89257 Russian Federation : MPK G06F15/00 G05B13/00 / V.Kotov, E.Melnik, I. Scherbinin, I. Korovin; zayavitel I patentoobladatel FGAOU VPO «UFU». – No2009134431/22; zayavl. 14.09.2009; opubl. 27.11.2009, bul. № 33, p. 2 c.
- [13] Melnik E., Korobkin V., Klimenko A. System Reconfiguration Using Multiagent Cooperative Principles // *Proc. of the First International Scientific Conference “Intelligent Information Technologies for Industry” (IITI'16)*. 2016, V. 451 (Series “Advances in Intelligent Systems and Computing”), pp. 385 – 394.
- [14] Klimenko A., Klimenko V., Melnik E. The Parallel Simulated Annealing-Based Reconfiguration Algorithm for the Real Time Distributed Control Fault-Tolerance Providing // *Application of Information and Communication Technologies – AICT'2015 (IEEE Catalog Number CFP1556H-PRT)*. Rostov-on-Don, Russia, 14 – 16 Oct. 2015. Rostov-on-Don, 2015, pp. 277 – 280.
- [15] Melnik E, Klimenko A, Korovin I 2016 A recovery method for the robotic decentralized control system with performance redundancy. *Proceedings of First International Conference, ICR 2016* , pp.9-17
- [16] E.Melnik, A.Klimenko, V.Korobkin. Reconfigurable Distributed Information and Control System Multiagent Management Approach. // *Proc. of the Second International Scientific Conference “Intelligent Information Technologies for Industry” (IITI'17)*. 2017.
- [17] A.B. Barskiy. Parallelniye informatsionniye tehnologii. Binom. Laboratoriya Znaniy, 2013.
- [18] A. Klimenko, E.Melnik, D.Ivanov. The configuration generation problem for the informational and control systems with the performance redundancy. *Proceedings of Second International Conference on Industrial Engineering, Applications and Manufacturing, ICEAM -2016*.