A workload distribution problem model and online constraint forming technique for the control systems in the fog-computing environment

E V Melnik¹, A B Klimenko²

¹ Southern Scientific Center of Russian Academy of Sciences, 41, Chehova st., Rostov-on-Don, 344006, Russia
² Scientific research institute of multiprocessor computer systems of Southern Federal University, 2, Chehova st., Taganrog, 347928, Russia

E-mail: anna_klimenko@mail.ru

Abstract. The paper deals with the workload distribution problem in the fog-computing environments. This problem is topical due to the wide range of networked control systems functioning on the basis of cloud and fog-computing infrastructure. The latter presupposes the techniques of workload relocation between the cloud, fog, and, in particular, the endpoint devices. As control systems have their own requirements in terms of latency, reliability, fault-tolerance, etc., the specific techniques must be developed to deliver these particular features using the fog-computing infrastructure. In the current paper a problem model of workload relocation is presented, the problem of constraints forming is considered and the new constraint forming technique is proposed.

1. Introduction

The most part of contemporary information and control systems (ICS) are the networked ones and they use all the existing network infrastructure [1]. The large number of ICSs uses the “cloud” as a fault-tolerant, scalable platform to ensure the service quality [2]. Moreover, the concept of Industry 4.0 is rather cloud-oriented to enhance the efficiency of industry management process [3,4]. There are some examples of successful usage of clouds in the area of networked control systems: Experion Elevate (Honeywell Process Solutions, iSpace, Battery Monitoring system (ADAC lab) and many others. Actually, numerous papers, devoted to the cloud and the industrial control systems, including the networked ones, were written and considered in the first decade of the third millennium, and then the new computational concept emerged, which was called “fog-computing” [5-7].

Fog-computing enhances the cloud paradigm, and allows one to preprocess data on the so-called fog-nodes. Edge computing goes quite farther and allows one to process the sensor data on the edge, endpoint devices. There is a wide range of papers devoted to the fog- and edge-computing paradigms, they can be observed in detail in [5-10]. But the important distinction between the classic “cloud” and the “Fog” is that the computational tasks can be offloaded from the “cloud”, which is located in some datacenter, to the fog, which consists of numerous nodes, geographically distributed and interconnected by the highly heterogeneous communication environment. One more issue of the “fog” is the relational variability of the node set. The instability of nodes increases on the edge of the network.
The problem of workload distribution among the fog-nodes is a topical one. The problem of workload distribution between the cloud and the fog is considered in [11-12]. At the same time, very little attention is paid to the “offloading” procedure, when a part of computational subtasks has to be shifted to the fog-or edge devices from cloud and vice versa [13]. Besides, particular systems demand particular criteria optimization during the workload distribution, for example, the device reliability enhancement is considered in [14]. The key issue of the workload distribution problem is that the problem relates to the combinatorial optimization problem class with multiple constraints and multiple criteria. Outside the cloud, there is no exact information about the set of the nodes, which determines the problem constraints. In other words, the search space of the problem seems to be unconstrained, yet it is a serious issue for solving the combinatorial optimization problem.

In the current paper a new workload distribution problem model adapted to the fog-environment is presented, as well as online technique for the constraint forming is proposed. It allows solving the workload distribution problem within the limited search space and providing acceptable results in terms of quality and rate of operation.

2. The workload distribution problem from the “offloading” point of view
The offloading computational model [13] presupposes in terms of the fog-computing the workload shift from one location to another. It is related to the urgent need to offload some devices, or, to the reconfiguration procedure, when some computational tasks must be accomplished on the functioning devices. So, let us consider the subtask graph $G$ given, with subtasks computational complexities $x_i$ and data portions $w_i$ to be transmitted between subtasks. Graph $G$ is split into two subgraphs, $G'$ and $G''$. $G'$ is considered to be bound to the computational units (CUs) of the network segment $P'$, while $G''$ is still performed on the network segment $P''$, as shown in Figure 1.

![Figure 1. Subtasks distribution for the offloading computational model.](image)

Also, such subtasks relocation presupposes the solution of the optimization problem with some criteria and constraints, partially formed by system requirements of the ICS. Further we will consider the main criterion of the tasks relocation to be a reliability function of the system (the connection between the reliability function and the workload is presented in the study [15]).

Let us consider $G$ is a graph-based description of the task set, $G = \{<i, x_i, w_i>\}$, where $i$ is a subtask unique identifier, $x_i$ – task computational complexity, $w_i$ – the data size to be transmitted to the communication environment by subtask $i$.

The subtasks of $G$ are bound to the nodes of CU set $P$, where $P$ is described by a graph structure $P = \{<j, p_j> list\}$, where $j$ is a node identifier, $p_j$ – node performance, list – matrix of communication channels bandwidth among the incident network nodes.

Then we consider the subtasks of $G'$ to be relocated, while subtasks of $G''$ continue to perform. There are some dataflows between $G'$ and $G''$, which, relating to $G'$ can be described by the following set of tuples:

$Flow_{in} = \{<id_{out}, id_{in}, w_{out,in}>\}$ is a tuple set, which describes the data amounts to be transmitted between the node of $G''$ id_{out} and the node of $G'$ id_{in} from $G''$ to $G'$. 
Flow_out = \{<id_{in}, id_{out}, w_{in, out}>\} – a tuple set, which describes the data amounts to be transmitted between the node of G’’ id_{out} and the node of G’ id_{in} from G’ to G’’.

We consider the problem of the workload distribution as follows: having G’’ linked to P’’, and Flow_in, Flow_out, one needs G’ to be located within the P’ so as the total time of G completion is less then given time T, with the objective function optimization (as was mentioned earlier, we consider the nodes reliability functions as the objective ones). The solution of the problem described above is a linking of subtasks G’ to the nodes P’, expressed as matrix A:

A = \begin{bmatrix} <t_{0}^{ij}, u_{ij}> \\ \vdots \\ <t_{0}^{NM}, u_{NM}> \end{bmatrix},

where \( t_{0}^{ij} \) – the time moment of the subtask \( i \) computations beginning by the node \( j \),
\( u_{ij} \) – the fraction of total performance \( p_{i} \), given by node \( j \) for task \( i \) accomplishment.

This problem model allows one to link more than one task per node at the same time, enhancing the classical scheduling problem formulation [16].

For further model development the following parameters must be considered:
\( L_{p}(A) \) – node workload, which is generated by the subtask binding to the node;
\( L_{\text{dist}}(A, \text{Flow}_{\text{in}}, \text{Flow}_{\text{out}}) \) – node workload, which is generated by data exchange between G’ and G’’;
\( L_{a}(A, \text{Flow}_{\text{in}}, \text{Flow}_{\text{out}}) \) – node workload, which is generated by the data transition through the node;
\( D_{n} \) – the list of ribs of graph P, which determines the route between nodes \( l \) and \( k \);
\( \text{ListD}_{n} \) – the matrix, which describes the network channels bandwidth between nodes \( l \) and \( k \).

We consider the objective function as a set of particular nodes reliability function values \( F_{j} \):

\( F_{j} = e^{-\lambda_{j}t}, \)

where \( \lambda_{j} \) - node \( j \) failure rate,
\( t \) – elapsed time of device operation.

As \( \lambda = \overline{\lambda}_{j} \cdot 2^{\Delta T/\lambda_{0}}, \) and \( \Delta T = kL \), where \( L \) is a device workload, \( k \) is a ratio and depends on the device type, the dependency between the reliability function and the workload will be as follows:

\( F_{j} = e^{-\lambda_{j}2^{\Delta T/\lambda_{0}}} \).

Device workload depends on the computational tasks distribution through the set of devices, which is described by matrix A. Also the following parameters must be included into the problem model:
\( L_{p}(A), L_{\text{dist}}(A, \text{Flow}_{\text{in}}, \text{Flow}_{\text{out}}), L_{a}(A) \).

So, the overall workload of device \( j \) will be as follows:

\( L_{j} = L_{p}(A) + L_{\text{dist}}(A, \text{Flow}_{\text{in}}, \text{Flow}_{\text{out}}) + L_{a}(A) \).

In case of a set of devices the scalar objective function transforms into the vector one.

The major constraint for this problem is the G completion time T, in other words:

for \( G = G' \cup G'' \)

\( \forall i \in G: \frac{x_{i}}{P \mu_{i}} + t_{\text{dist}}(i) < T, \)

where \( t_{\text{dist}}(i) \) - the maximum time of data delivery from subtask \( i \) to the subtasks-receivers of the data.

As the model considers dataflow routes, \( t_{\text{dist}}(i) \) is calculated with the following function:

\( t_{\text{dist}}(i) = \xi(A, G, P), \)
More precisely, the data delivery is calculated on the basis of full information about the subtasks binding to the computational nodes and with the participating of the parameters $D_a$ and $ListD_a$.

Yet the problem with this model is that there is no constraints, which could describe the initial set of the nodes, considered for the subtasks distribution. The authors of this paper propose an algorithmic online technique, which forms the device constraints for this problem.

3. An online constraint forming technique for the workload distribution problem

Let us consider the term “local device group” (LDG) as a set of devices, which are:
- interconnected by the high-velocity communicational channels without any transitional nodes;
- participate in solving of computational task $G$.

Then, we consider that some computational subtasks must be relocated somewhere (Figure 2), for example, from the “cloud” to the fog-layer, or from the edge devices to the fog-layer, or within the fog-layer. As the operations are quite equal in all the three cases, we will describe the technique proposed by the “cloud-offload” example.

![Figure 2. The scheme of the “offloading” strategy.](image)

As $G'$ has to be brought to the fog, there must be a node in the $P''$, which is responsible for the relocation. Consider the LDG, which solves the $G''$. A leader must be elected for the further actions. It is done, for example, by “round-robin” discipline, or by one of the “leader election” algorithms. Having been elected, the leader sends the request for resources to its LDG, as is shown in Figure 3.

![Figure 3. The elected leader sends the requests to its LDG.](image)

If devices answer the request with agreement to allocate the resources needed for the subtasks to be relocated, then the workload distribution problem is solved within the set of those devices. If the solution is acceptable, the subtasks are linked to the devices.

If the number of resources is insufficient, or the solution is not acceptable, the devices send the request to their LDG, as is shown in Figure 4.
This procedure is iterative and eventually forms the set of nodes, which are to solve the computational subtasks. After the constraints are formed, the problem of workload distribution will be solved.

The generic scheme of the constraints forming technique usage is presented in Figure 5.

**Figure 4.** The LDG enhancement procedure.

![Figure 4](image)

**Figure 5.** The solution of the workload distribution problem with the constraints forming technique.

### 4. Conclusion

This paper deals with the problem of workload distribution in the networked control systems which use the fog-computing infrastructure. So, the appropriate optimization problem must be solved.

The problem under consideration is that of combinatorial optimization, so the search space must be constrained.

Outside the cloud, or drifting from the edge of the network to the fog-layer, we deal with almost unlimited set of nodes which obviously must have some boundaries.

In this paper the model of the optimization problem and the technique of the online constraint forming are proposed. The model pays attention to the workload of the system, which is generated by the transitory data flow. The technique proposes to mark algorithmically nodes for the computational task possible relocation and it makes it possible to solve the optimization problem in the limited search space.
5. Acknowledgments
The reported study was funded by the RFBR project 18-29-03229 and the GZ SSC RAS N GR project AAAA-A19-11901190173-6.

References
[1] Ge X, Yang F and Han Q L 2017 Distributed networked control systems: A brief overview Information Scence 380 117-131.
[2] Hu P, Dhelim S, Ning H and Qiu T 2017 Survey on fog computing: architecture, key technologies, applications and open issues J. Netw. Comput. Appl. 98 27–42
[3] Givehchi O, Trsek H and Jasperneite J 2013 Cloud computing for industrial automation systems - A comprehensive overview IEEE International Conference on Emerging Technologies and Factory Automation, ETFA
[4] Givehchi O and Jasperneite J 2013 Industrial Automation Services as part of the Cloud: First experiences Proceedings of the Jahreskolloquium in der Automation-Komma, Magdeburg
[5] Cisco. Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are Retrieved from: https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf
[6] Bonomi F, Milito R, Zhu J and Addepalli S 2012 Fog Computing and Its Role in the Internet of Things Proc. first Ed. MCC Work. Mob. cloud Comput. 13–15
[7] Stojmenovic I, Wen S. 2014 The Fog Computing Paradigm: Scenarios and Security Issues Proc. 2014 Fed. Conf. Comput. Sci. Inf. Syst. 2 1–8.
[8] Cao Y, Chen S, Hou P and Brown D 2015 FAST: A fog computing assisted distributed analytics system to monitor fall for stroke mitigation Proceedings of the 2015 IEEE International Conference on Networking, Architecture and Storage, NAS 2015 2–11
[9] Aazam M and Huh E N 2015 E-HAMC: Leveraging Fog computing for emergency alert service 2015 IEEE International Conference on Pervasive Computing and Communication Workshops 2015 518–523
[10] Inaltekin H, Gorlatova M and Mung C 2017 Virtualized Control over Fog: Interplay Between Reliability and Latency Retrieved from: https://arxiv.org/pdf/1712.00100.pdf
[11] Abedin S F, Alam M G R, Tran N H and Hong C S 2015 A Fog based system model for cooperative IoT node pairing using matching theory 17th Asia-Pacific Network Operations and Management Symposium: Managing a Very Connected World, APNOMS 2015 309-314
[12] Agarwal S, Yadav S and Yadav A K 2016 An Efficient Architecture and Algorithm for Resource Provisioning in Fog Computing Int. J. Inf. Eng. Electron. Bus. 8(1) 48-61
[13] Moysiadi V, Sarigiannidis P, Moscholios I 2018 Towards Distributed Data Management in Fog Computing Wireless Communications and Mobile Computing
[14] Melnik E, Korovin I and Klimenko A 2017 Improving dependability of reconfigurable robotic control system Second International Conference ICR 10459 144–52
[15] Melnik E and Klimenko A 2018 Agent-Based Approach to Distributed Information and Control System Reconfiguration Proceedings of the Computational Methods in Systems and Software 661
[16] Sha L, Abdelzaher T, Årzén K E, Cervin A, Baker T, Burns A, Buttazzo G, Caccamo M, Lehoczky J and Mok A K 2004 Real time scheduling theory: A historical perspective Real-Time Syst. 28 (2-3) 101-155