The comparative estimation of workload relocation approaches in the fog- and edge-computing environments

A B Klimenko¹,² and I B Safronenkova²

¹ Scientific research institute of multiprocessor computer systems of Southern Federal University, 2 Chekhov Street, Taganrog, 347928, Russia
² Federal Research Centre, The Southern Scientific Centre of the Russian Academy of Sciences, 41 Chekhov Street, Rostov-on-Don, 344006, Russia

* E-mail: anna_klimenko@mail.ru

Abstract. In the current paper the issues of the workload relocation in the fog- and edge-computing environments are in-question. The workload relocation problem is closely connected to the scheduling problems, yet, outside the cloud there is almost unlimited number of nodes to place the computational tasks. So the search space for the optimization problem grows, and the time of the workload relocation degrades. We emphasize the techniques to limit the set of candidate nodes for the tasks distribution. In the paper two approaches are proposed and considered in terms of time consumption. The simple models are developed with the following simulation. Also the distributed-ledger-based modifications of the techniques are proposed and examined.

1. Introduction

A lot of contemporary information systems functions in the cloud environments nowadays. Yet the instant growth of data and the number of end-point devices has made the “cloud” infrastructure insufficient. So, the new fog-computing concept was introduced and applied in numerous fields. Also the so-called edge-computing emerged, enhancing the computations to the edge of the network [1-3]. Since there are mechanisms [4] to shift the workload to and from the cloud, some particular information and control systems (ICS) demand an individual approach to the computations relocation, e.g., in case of special system requirements. So, the relocation of the workload must be modeled, estimated and passed to the fog- and edge- middleware to outline the desired nodes of the new workload location.

The workload relocation procedure is easy for implementation within the cloud due to its relative stability and fixed number of computational resources. Having a fixed set of computational nodes it is easy to solve a multicriteria and multiobjective scheduling problems. Yet, there is no stability outside the cloud, and the number of fog- and edge-devices may be numerous, the same way the scheduling problem search space is. So, the key issue to relocate the workload outside the cloud is to make some boundaries on the computational nodes set. It is obvious that the less redundant the set of nodes is, the less time is needed to distribute the computational workload.

In this paper we deal with some approaches to the workload relocation, including the problem of making boundaries for the search space of scheduling problem. The workload relocation itself, connected closely to the classical scheduling problems [5-7], has some differences, for example:
• the connections of the computational tasks to be relocated with the tasks, which stay on the nodes previously selected;
• connections between the tasks to be relocated, some particular objective functions to be optimized and, possibly, some predefined algorithmic structures.

The knowledge of the systems state is quite useful, when the nodes decide where to relocate the computational workload, so the issue of the knowledge representation and propagation is in-question too.

This paper is a continuation of the previous study, presented in [8].

The following sections of this paper contain:

• the formulation of the workload relocation problem;
• the description of the local device group-based (LDG) and ontology-based approaches to the workload relocation;
• the comparison of cases with the lack of overall knowledge of the network state and those with the distributed-ledger-based knowledge representation.

2. A workload relocation problem
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 is shown in figure 1.

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

Also, such subtasks relocation presupposes the optimization problem solving 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 studies [9, 10]).

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 \( P = \{ <j,p_j> \text{list} \} \), where \( j \) is a node identifier, \( p_j \) – node performance, \( \text{list} \) – matrix of communication channels bandwidth among the incident network nodes.

Then consider the subtasks of \( G' \) has to be relocated, while subtasks of \( G'' \) continues to perform. There are some dataflows between \( G' \) and \( G'' \), which, relating to \( G' \) can be described by the following set of tuples:

\[
\text{Flow}_{\text{in}} = \{ <id_{out},id_{in},w_{out\_in}> \} - \text{is a tuple set, which describes the data amounts to be transmitted between the node of } G'' \text{ id}_{out} \text{ and the node of } G' \text{ id}_{in}, \text{from } G'' \text{ to } G'.
\]

\[
\text{Flow}_{\text{out}} = \{ <id_{in},id_{out},w_{in\_out}> \} - \text{is a tuple set, which describes the data amounts to be transmitted between the node of } G'' \text{ id}_{out} \text{ and the node of } G' \text{ id}_{in}, \text{from } G' \text{ to } G''.
\]

Consider the problem of the workload distribution as follows: having \( G'' \) linked to \( P'' \), and \( \text{Flow}_{\text{in}}, \text{Flow}_{\text{out}} \), one needs \( G' \) to be located within the \( P' \) so as the total time of \( G \) completion is less than given \( T \), with the objective function optimization (as was mentioned earlier, we consider the
nodes reliability functions as the objective ones). The solution of the problem determined is a linking of subtasks \( G' \) to the nodes \( P' \), described as matrix \( A \):

\[
A = \begin{bmatrix}
<t^0_{ij}, u_{ij}>
& \cdots \\
& \vdots \\
&t^m_{ij}, u_{NM}
\end{bmatrix},
\]

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

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

For the further model development the following parameters must be considered:
- \( L_p(A) \) – node workload, which is generated by the subtask binding to the node;
- \( L_{dist}(A, Flow\_in, Flow\_out) \) – node workload, which is generated by data exchange between \( G' \) and \( G'' \);
- \( L_p(A, Flow\_in, Flow\_out) \) – node workload, which is generated by the data transition through the node;
- \( D_{lk} \) – the list of ribs of graph \( P \), which determines the route between nodes \( l \) and \( k \);
- \( ListD_{lk} \) – the matrix, which describes the network channels bandwidth between nodes \( l \) and \( k \).

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 \( \lambda = \lambda_0 \cdot 2^{AT/10} \), and \( \Delta T = k L \), 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:

\[
F_j = e^{L_p j 2^{AT/10}}.
\]

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:
- \( L_p(A) \), \( L_{dist}(A, Flow\_in, Flow\_out) \), \( L_{tr}(A) \).

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

\[
L_j = L_p(A) + L_{dist}(A, Flow\_in, Flow\_out) + L_{tr}(A).
\]

Adding the overall device workload equation to the objective function for a device is formed.

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

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

\[
G = G' \cup G'' \ \forall i \in G : \frac{x_i}{p_j u_{ij}} + t_{dist}(i) < T,
\]

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

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

\[
t_{dist}(i) = \tilde{\zeta}(A, G, P).
\]
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 $D_{lk}$ and $ListD_{lk}$.

3. **Local device group-based and ontology-based approaches to the workload relocation**

At least two techniques are proposed [8] to set boundaries for the search space in the fog-computing environment. The first technique is based on the forming of local device groups, the second is based on ontologies usage.

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;
- solve the subtasks of the integral task G.

LDG-based technique can be described by the following steps:

- Leader election is conducted among the nodes, which perform the overall computational task;
- The leader asks its local group for information about resources to place the additional workload;
- If the answer is positive, the set of nodes is fixed (and the boundaries of the search space are set). Then the leader models the distribution of the computational subtasks to be relocated through the nodes selected, solving the scheduling optimization problem.
- If the answer is negative (that is, there are no nodes in the local group with sufficient resources), then the local group is extended in the following way: each node of the local group retransmits the request of the initial node to its local group. This procedure is iterative and is repeated until the nodes with resources are found.
- If the scheduling problem has been solved successfully, and the result meets all problem constraints, the computational subtasks are binded to the fixed nodes set. If there is no acceptable solution, the procedure is repeated from the step 4 with the extension of the local group.

The technique described above is universal and none of the nodes possesses the knowledge about the resource state of its neighbours. The obvious advantage of this technique is that the nearest neighbours are examined first, which is expedient in case of computational graph partition. So, the distance between the connected tasks with information exchange would not be too long in terms of network hops. Yet, due to its iterative nature, it can consume unacceptable time to find appropriate nodes set.

To estimate roughly the time needed to apply this technique, consider the following parameters:

- $N_i$ – the number of nodes, on which the computational tasks are performed initially;
- $N_t$ – the number of tasks to be relocated;
- $L_i$ – the size of the local group on the request $i$;
- $I_g$ – the overall iteration number;
- $S$ – the number of iterations to form the local group with the possibility to solve the problem.

Then, a rough time estimation will be as follows:

$$T_{LDG} = \alpha N_1 + \left[ \sum_{i=1}^{S} v L_i + N_1 L_i \right] I_g,$$

where $\alpha$ is a ratio between the time of leader election procedure and the number of nodes,

$v$ is a ratio between the time of requests handling and the number of nodes to be examined;

$\gamma$ is a ratio between the size of the scheduling problem and the time of its solving.
Another technique for creating boundaries in the fog environment is the ontological-based approach, proposed in [8, 11]. There is a considerable assumption in this technique: it is presupposed that the structures of computational tasks to be relocated are defined and described, and the ways of task graph partitioning are determined with the finite number of possible combinations. Then, having the ontology of algorithm type, algorithm graph partition and the initial place of the tasks to be performed, we can classify the subgraph of tasks to be relocated according to the ontology, and, using the heuristic rules, to select the nodes, which are the most appropriate for the subtasks assigning. The example of ontology is presented in figure 2.

Figure 2. The example of parallel algorithm and graph partition ontology.

To locate some workload according to the ontology-based technique the following steps must be done:

- the leader is elected among the nodes, on which the subtasks are located initially;
- the request to the fog-nodes is formed and transmitted through the whole fog domain;
- all nodes send the information about the state of their resources;
- processing the overall information about the accessible fog nodes, and with the usage of heuristic rules, the sets of nodes are formed;
- the leader models the tasks distribution for the set of pre-selected nodes.

While the set of pre-selected nodes can be quite large (larger than the LDG in the technique previously described), we assume that the scheduling problem has a solution.

To estimate the time of this technique application consider the parameters:

- $D$ – the number of nodes in the fog-domain;
- $D_s$ – the number of pre-selected nodes after the ontological analysis procedure.

So, the time estimation for the ontology-based technique of the workload relocation will be as follows:

$$T_o = \alpha N_i + k \ln D + \eta D + \beta D + n_i D_s \gamma,$$

where $k \ln D$ - is the time needed to examine the nodes in the network using the gossip-based algorithm;

$\eta D$ - is the time to receive the answers from nodes;

$\beta D$ is the time needed for ontological analysis;

$N_i D_s \gamma$ - is the time for modeling of $N_i$ tasks distribution through the $D_s$ nodes.
4. The overall knowledge of system resources

The techniques considered in the previous section are similar: both of them presuppose that none of the nodes has the relevant knowledge about the resources of other nodes. It generates the need to send requests and receive the answers from nodes with the information about the available resources. If the node possessed the overall knowledge about the resource state of the system, it would make the procedure of forming scheduling problem boundaries less time-consuming. The way to provide such overall knowledge of the system state is similar to the distributed ledger transactions propagation through the network [12, 13]. Assuming every resource utilization change to be a transaction, while every device has its own copy of the ledger, we receive the situation, when no search complicated procedures to pre-select nodes needed. Then, the time of LDG-based technique with the overall system knowledge would be as follows:

\[ T_{LDG} = \alpha N_1 + \Phi + N_1 C_{LDG} \]

where \( C_{LDG} \) - is the set of nodes preselected on the basis of a ledger analysis;
\( \Phi \) - is the time needed to examine the ledger copy to find the appropriate nodes.

The time of the ontology-based technique can be estimated as is shown in expression below:

\[ T_o = \alpha N_1 + \Phi + \psi C_o + N_1 C_f \]

where \( \psi C_o \) - is the time needed to conduct the ontological analysis of the preselected nodes;
\( C_f \) - is the resulting set of nodes.

Obviously, such approach is promising in terms of time consumption because the ledger is analyzed on the leader node without any data transmissions through the network. Yet, there is a need for instant data exchange within the network for ledger updates, which can generate data exchange of a high redundancy.

5. Modeling and simulation results

To examine the techniques, described in the section above, consider the following cases:

- LDG-based technique: the case with the variable number of local group devices and the fixed number of iterations with requests for resources (Li). The last request is successful, the resources are sufficient and the nodes can participate in workload distribution. The number of “unsuccessful” overall iterations is fixed.
- LDG-based technique: the case with the fixed number of local group devices and the fixed number of iterations with requests for resources. The number of “unsuccessful” overall iterations is variable.
- The ontology-based technique: case with the variable D.

The results of comparative analysis are shown in figure 3.

![Figure 3. The comparison of ontology-based and LDG-based techniques.](image-url)
For the ontology-based technique the case with the varying D is shown. For the LDG-based technique the three cases are considered: when the number of local group nodes increases (Tldg_growth), the number of local group nodes is of a uniform distribution law (Tldg_uniform) and the number of nodes in the local group decreases with the iterations (Tldg_decrease). One can see that with the local group growth (in case of unsuccessful requests for resources) the time of the overall workload distribution procedure is the worst. Yet, if the resource requests in the LDG-based technique are successful, the time for those cases is better than the time of ontology-based technique. It must be mentioned, that there is no guarantee of the lack of the local group growth, while the ontology-based technique gives the stable increase of time with the fog node number growth.

With the usage of distributed ledger-based technique to inform the nodes about the system state, the diagrams show the following results in figure 4.

In the figure above the following cases are considered:

- For the ontology-based technique the number of fog-nodes to be examined increases;
- For LFG-based technique the number of unsuccessful iterations increases;
- For the ledger-based techniques modifications the volume of the ledger grows.

One can see that with the parameters set the ontology-based technique shows the best results, yet they degrade fast with the growth of fog nodes number (see figure 5).
So, with the rough time estimations the main conclusion can be made: the workload relocation technologies application is connected closely to the network particularities. So the choice of the technique must be conducted in relation with network functioning conditions, or there can be used a hybrid technique.

6. Conclusion

In this paper the topical problem of the workload relocation in the fog- and edge-environments is considered. The main issue of the workload relocation problem solving is that without any boundaries set onto the computational nodes the search space is almost unlimited. The two approaches to the workload relocation and techniques of the boundaries forming are considered, modelled and analyzed in the current paper. Also we have compared these approaches with the possible their modifications with the distributed ledger technology application. The general results of the research are as follows:

- LDG-based technique has shown the worst results in the condition of the local group growth and the increase of unsuccessful resource request iteration number;
- with the ledger volume increase there is a threshold, when the distributed ledger is inexpedient for usage despite the lack of iterations and the need to examine of the large number of nodes.

6. Acknowledgments. The paper has been prepared within the RFBR project 18-29-03229 and RAS presidium fundamental research №7 «New designs in the prospective directions of the energetics, mechanics and robotics», № gr.project AAAA-A18-118011290099-9.

References
[1] 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
[2] Stojmenovic I, Wen S 2014 The Fog Computing Paradigm: Scenarios and Security Issues Proc. 2014 Fed. Conf. Comput. Sci. Inf. Syst. (September 7-10, Warsaw) vol 2 ed M Ganzha, L Maciaszek and et al (New York City: IEEE) pp 1–8
[3] Yi S, Li C and Li Q 2015 A Survey of Fog Computing: Concepts, Applications and Issues Proc. of the 2015 Workshop on Mobile Big Data (June 22-25, Hangzhou) (New York: ACM) pp 37–42
[4] Moysiadis V, Sarigiannidis P; Moscholios I 2018 Towards Distributed Data Management in Fog Computing Wireless Communications and Mobile Computing 7597686
[5] Pinedo M L 2008 Scheduling: Theory, Algorithms, and Systems 3rd edn. (New York: Springer)
[6] Pereira J 2015 Procedures for the bin packing problem with precedence constraints. European Journal of Operational Research 250(3) 794–806
[7] Sha L et al. 2004 Real time scheduling theory: A historical perspective Real-Time Systems 28 101–105
[8] Klimenko A, Safronenkova I 2019 An Ontology-Based Approach to the Workload Distribution Problem Solving in Fog-Computing Environment. Proc. Computer Science On-line Conf.(April 24-27, Zlin) vol 985 ed R. Silhavý (Cham: Springer )
[9] Strogonov S A 2006 Individual reliability forecasting of IC chip with the help of ARIMA models J. Components & Technologies 10 44–49
[10] Mylov G V, Medvedev A M, Semenov P V and Konstantinov P N 2016 The Scientific Basis of Printed Board Designing (Moscow: Telekom)
[11] Klimenko A B, Safronenkova I B 2018 Ontology based workload allocation problem solving in fog computing environment Izvestiya sfedu. Engineering sciences 8 83-94
[12] Maull R, Godsiff P, Mulligan C, Brown A, and Kewell B 2017 Distributed ledger technology: Applications and implications Strategic Change 26(5) 481-489

[13] Natarajan H, Krause S and Gradstein H 2017 Distributed Ledger Technology and Blockchain. FinTech 1