Fog-computing concept usage as means to enhance information and control system reliability

To cite this article: E V Melnik et al 2018 J. Phys.: Conf. Ser. 1015 032175

View the article online for updates and enhancements.

Related content
- New Technologies for Smart Grid Operation: Reliability principles
  S T Mak
- Distributed Information and Control system reliability enhancement by fog-computing concept application
  E V Melnik, A B Klimenko and D Ya Ivanov
- Workload analyse of assembling process
  L D Ghenghea
Fog-computing concept usage as means to enhance information and control system reliability

E V Melnik¹, A B Klímenko², D Ya Ivanov²

¹ Southern Scientific Center of the Russian Academy of Science, 41 Chekhov St., Rostov-on-Don, 344006, Russia
² Scientific research institute of multiprocessor computer systems of Southern Federal University, 2 Chekhov St., GSP-284, Taganrog, 347928, Russia

E-mail: anna_klmenko@mail.ru

Abstract. This paper focuses on the reliability issue of information and control systems (ICS). The authors propose using the elements of the fog-computing concept to enhance the reliability function. The key idea of fog-computing is to shift computations to the fog-layer of the network, and thus to decrease the workload of the communication environment and data processing components. As for ICS, workload also can be distributed among sensors, actuators and network infrastructure facilities near the sources of data. The authors simulated typical workload distribution situations for the “traditional” ICS architecture and for the one with fog-computing concept elements usage. The paper contains some models, selected simulation results and conclusion about the prospects of the fog-computing as a means to enhance ICS reliability.

1. Introduction

Information and control systems (ICS) obviously have been widely used and in demand for recent decades. A huge amount of mechatronic complexes operate under the ICS monitoring and control: from aircraft and spacecraft to gas and oil production, not excluding hazardous industries equipped with robotics, where a man is not supposed to be present [1,2]. Among the mechatronic complexes, the special class is to be emphasized. It includes objects, which are critical to failures. For such system, failure leads to the financial losses, ecocatastrophes and casualties, so the question of system dependability is one of the main issues of contemporary science and engineering.

There are a great number of papers devoted to the ICS dependability, considered from different points of view [3–10].

Within the current paper, let us focus on the reliability issue, which is the probability of system functioning through the determined time period. Moreover, let us presuppose the fog-computing concept to be a means of enhancing the ICS reliability, while other works devoted to fog-concept usage consider the issues of system latency and communication network load.

The current paper contains:
- the fog-computing concept brief overview;
- the ICS traditional architecture;
- the simplified model description of ICS with the “traditional” computational process and with the fog-computing concept usage;
• the mathematical models of system component workload and the models of component reliability;
• the selected simulation results, where the advantages of fog-computing application for the reliable ICS are presented;
• conclusion and future work prospects.

2. Fog-computing concept
The fog-computing concept is quite new. As is known, the Marketing team at Cisco introduced it in January, 2014 [11]. So, despite the large area of new works devoted to the fog-computing and Internet of Things (IoT), this brief overview of the fog-computing concept is based on Cisco’s published documents [12].

The key idea of the fog-computing is to set the data processing process as near as possible to data sources and thus the following advantages are reached:
• the system latency reduces;
• the network traffic reduces.
Cisco recommends considering fog-computing when:
• data are collected at the extreme edge (vehicles, factory floors, railways, etc.);
• lots of things across a large geographic area generate data;
• it is necessary to analyze and act on the data in less than a second.

Fog-nodes are usually controllers, communicators, different network infrastructure facilities, etc. The general architecture of the fog-computing concept is presented in fig. 1.

Figure 1. The fog-computing concept model

According to the Cisco’s declaration, the fog nodes (fig.1) closest to the network edge ingest the data from IoT devices. Then - and this is crucial - the fog IoT application directs different types of data to the optimal place for an analysis.

The most time-sensitive data are analyzed on the fog node closest to the things generating the data. In a Cisco Smart Grid distribution network, for example, the most time-sensitive requirement is to verify the fact that protection and control loops are operating properly. Therefore, the fog nodes closest to the grid sensors can look for signs of problems and then prevent them by sending control commands to actuators.

Data that can wait seconds or minutes for action are passed along to an aggregation node for analysis and action. In the Smart Grid example, each substation might have its own aggregation node that reports the operational status of each downstream feeder and lateral. Data that are less time sensitive are sent to the cloud for historical analysis, big data analytics, and long-term storage. For example, each of thousands or hundreds of thousands of fog nodes might send periodic summaries of grid data to the cloud for historical analysis and storage.

The following prospective examples of ICS based on fog-computing are known [13]:

---

---
• E-health systems [14,15];
• Food chain monitoring [16];
• Energy management [17];
• Mobile Application Support [18].

But, as was mentioned above, the fog-computing concept is used as a means to reduce system latency and to offload the network, while the authors consider the fog-computing concept as a means to enhance the system component reliability function.

3. Information and control system architecture and reliability issues
The traditional ICS computational process architecture can be described as follows: there is a set of sensors, which in particular cases can generate huge amounts of data, a set of actuators, which receives some control impacts, and computational environment, where the data from sensors are collected and processed, so some control decisions can be made. Computational environment in contemporary ICS is distributed and includes some computational elements (CE). Network-centric ICSs considered in [19] are organized in the same manner.

ICS reliability is an important issue. As reliability is a component of the “dependability” term, issues of reliability, fault-tolerance and, as a consequence, reservation methods are considered altogether frequently. The examples of the “dependable” ICS components are considered in [20,21]. In general, the structural reservation is in use, including sliding reserve.

Another one technique to improve system dependability is connected with the performance redundancy usage and system reconfiguration as a means to deliver fault-tolerance [22–27]. These works are of interest because of possibility to avoid the multiple reserve elements usage. Besides, there is a proof for some system configurations, where the performance redundancy provides better reliability than the structural one.

Within this paper, let us focus on the latter technique: according to it, the workload can be distributed among the CEs, such as to maximize the overall system reliability function. But, considering the system reliability models too cluttered and sophisticated, let us present the simplified models of the workload distribution to prove the fog-computing prospective as a means to enhance the system reliability.

Here, in the figures below (fig.2-3), the simplified models of ICS computational process are presented. The first model shows the major computational process in traditional ICSs, and the second model gives the example of how it can work with the fog-computing concept application.
Figure 3. Computational process in fog-computing-based ICS

So, the workload of the system can be presented as shown in fig.4.

Figure 4. Workload distribution for traditional ICS and ICS based on fog-computing concept

4. Models of workload distribution
Let us consider traditional ICS architecture: within it, sensors send data to the computational environment, and actuators receive some control data. Within the model presented the following parameters must be determined:

- $C_{in}$ – the number of data transfers from sensors to computational environment,
- $V_{in}$ – amount of data per information transfer to the computational environment,
- $C_{out}$ – the number of data transfers from the computational environment to the actuators,
- $V_{out}$ – amount of data per information transfer from the computational environment.

Let us assume sensors and actuators as edge computational units (ECU). Every ECU has its own computational workload $x_1$, while tasks performed by the computational environment can be estimated by workload $x_2$. Also assuming splitting data to packages for the transfer, the package size is $w$ [e.g. bytes]. The computational workload generated by the package transfer to the computational environment is $\xi_1$, and computational workload generated by the package transfer from the computational environment is $\eta_1$. For the computational environment workloads generated by package sending and receiving is $\xi_2$ and $\eta_2$ correspondingly.
So, the workload for one ECU can be described in the following manner:

\[ L = x_1 + \frac{C_{in} \cdot V_{in} \cdot \xi_1}{w} + \frac{C_{out} \cdot V_{out} \cdot \eta_1}{w} \]  

(1)

When computational environment receives, processes and sends data, workload can be estimated as follows:

\[ L_{env} = x_2 + M \frac{C_{in} \cdot V_{in} \cdot \xi_2}{w} + M \frac{C_{out} \cdot V_{out} \cdot \eta_2}{w} \]

(2)

where \( M \) is the number of ECUs.

The models given below describe the workload of the elements of ICS developed with fog-computing concept usage.

As a part of workload is shifted unto the fog-layer, as well as a part of information exchanges.

Let us assume \( X \) as a part of workload shifted to the fog-layer, and \( \Delta \) is the workload to be performed in computational environment. The number and order of the information exchanges in the fog-layer depends on the particular monitoring and control task planning and implementation, hence it is useful to model and estimate two cases – when every ECU sends and receives data \( C_{in}+C_{out} \) times, and when every ECU sends and receives data once per monitoring and control task.

Then, assuming the facilities of communicational environment as participants of a computational process, the ECU workload can be described by the following equations (\( K \) is the number of communicators, controllers, etc.).

The ECU workload with an upper boundary of the information exchange number is as follows:

\[ L_{\gamma} = \frac{X}{M+K} + \frac{C_{in} \cdot V_{in} \cdot \xi_1}{w} + \frac{C_{out} \cdot V_{out} \cdot \eta_1}{w} \]  

(3)

The ECU workload with one information exchange per task is:

\[ L_{\lambda} = \frac{X}{M+K} + \frac{V_{in} \cdot \xi_1}{w} + \frac{V_{out} \cdot \eta_1}{w} \]  

(4)

Assuming that the computational process can be planned so as there are only two information exchanges between computational environment and the fog-layer (sending preprocessed data and receiving the control data), the following equation describes the workload for the computational environment:

\[ L_{\text{env}} = \Delta + \frac{V_{in} \cdot \eta_2}{w} + \frac{V_{out} \cdot \xi_2}{w} \]  

(5)

Having the workloads described, it is possible to embed these estimates in the reliability function of system elements.

According to the [28], failure rate of an element depends on its temperature:

\[ \lambda = \lambda_0 \cdot 2^{\Delta T / T_{0}} \]

(6)

where \( \Delta T \) is the temperature increase.

As temperature depends on the workload, the equation (6) can be presented as follows:

\[ \lambda = \lambda_0 \cdot 2^{\frac{D}{kT_{0}}} \]

(7)

where \( k \) is the ratio between workload \( D \) and the temperature increase.

The reliability function is described by the following equation:

\[ P(t) = e^{-\lambda t} \]

(8)

where \( t \) is the time of element functioning.
So, embedding equations (1-5) to the (8), the reliability function values can be estimated for elements of the traditional ICS architecture and ICS based on the fog-computing concept.

The next section of the paper contains the selected simulation results and comparison between two ICS architectures from the reliability point of view.

5. Simulation results

The authors simulated reliability functions for the parameter values: \( C_{\text{in}} = 10 \ldots 1000; \ C_{\text{out}} = 10; \ V_{\text{in}} = 200; \ V_{\text{out}} = 50; \ w = 50. \)

The workload dependency of \( C_{\text{in}} \) for the traditional ICS architecture is given in fig. 5.

![Figure 5](image5.png)

**Figure 5.** The workload increase for ECU and CE in the traditional ICS architecture.

It is shown that the computational environment workload grows much faster than the ECU one. Obviously, the scalability of such system is quite poor; the computational environment can be overloaded, as well as the communication one.

In fig. 6, the reliability function plots of the previously considered parameter values are presented.

![Figure 6](image6.png)

**Figure 6.** The reliability functions for the traditional ICS architecture

So, it is seen that the reliability function trend is of unsatisfactorily low quality: just after 200 h the reliability function value is less than 0.945. Of course, it depends on the parameter values chosen, but the trend of reliability function degradation is obvious.

Then let us consider workloads and reliability function plots of the ICS based on the fog-computing concept. The simulation results are presented in fig. 7.
Here trend $L_f$ relates to the full information exchange (eq. 3), $L_{f1}$ relates to the one-message exchange (eq. 4), and $L_{\text{env}}$ is the result of computational environment workload modelling. It is seen that the computational environment workload here is much more promising than the one shown in fig.6. Hence, the reliability function is supposed to be of higher quality.

The reliability function plots are shown in fig. 8 for one-message information exchange.

As a result of simulation, let us present the comparative plots of the reliability functions for traditional and fog-computing-based architectures (fig. 9).
This paper is prepared within the government task implementation GT SSC RAS 2018, № 01201354238

[1] Kalyaev I A and Melnik E V 2011 Decentralized systems of computer control (Rostov n/D: Publishing SSC RAS)

[2] Zhang Y and Jiang J 2008 Bibliographical review on reconfigurable fault-tolerant control systems Annu. Rev. Control32 229–52

[3] Laprie J-C 1985 Dependable Computing and Fault-Tolerance: Concepts and Terminology

Conclusion
In this paper, the problem of ICS reliability is considered. A new approach to enhance the system reliability is presented: it is based on the new computing paradigm – fog-computing. The keystone of fog-computing is to shift the computations to the sources of data. As the present-day sensors, actuators, controllers and other communicational facilities have network interfaces, as well as data storages, memory and computational capacities, a part of monitoring and control tasks can be distributed among the elements of the fog-layer.

The authors considered simplified ICS structures to synthesize models of element workload and their reliability functions. The simulation results allow one to make a conclusion that the fog-computing concept is rather prospective in the ICS area: the application of the fog-concept leads to the system element reliability function quality growth. Also it would be true to say that fog computing application to the reliable ICS development needs more research, modelling and simulation.

Acknowledgments
This paper is prepared within the government task implementation GT SSC RAS 2018, № 01201354238

Figure 9. The comparison of the reliability functions for traditional and fog-computing-based ICS architectures.

It is seen, that for the particular parameter values, the quality of the reliability function is much better for the ICS based on fog-computing than for the traditional one. Within this paper let us suppose that if the reliability function of each element is better, the overall system reliability is better too, but, of course, this is a preliminary assumption. The problem of system reliability modelling is much more complex and sophisticated in case of possible failures. Besides this, for the fog-computing based systems, the authors have to note that the number of units participating in computations grows, and this impacts the overall reliability.

So, speaking of the final system reliability model, the authors have to consider all types of facilities participating in a computational process and, taking it into consideration, maximize the overall system reliability by means of workload distribution. Such pre-formulated task seems to be multiobjective with constraints (the number of objective functions is up to the G*H, where G is the number of computational process participants, H is the number of discrete time periods), so the next issue of the problem solving is to find methods, which can get viable solutions in the acceptable time.
Proc. Int. Symp. Fault-Tolerant Comput. III 2–11

[4] Laprie J C 1992 Dependability: Basic Concepts and Terminology Dependability: Basic Concepts and Terminology pp 3–245

[5] Avizienis a, Laprie J C and Randell B 2001 Fundamental Concepts of Dependability Tech. Rep. Ser. Newcastle Upon Tyne Comput. Sci. 1145 7–12

[6] Garlan D, Cheng S-W and Schmerl B 2003 Increasing system dependability through architecture-based self-repair Architecting dependable systems (Springer) pp 61–89

[7] Reis G A, Chang J, Vachharajani N, Rangan R and August D I 2005 SWIFT: Software implemented fault tolerance Proceedings of the 2005 International Symposium on Code Generation and Optimization, CGO 2005 vol 2005 pp 243–54

[8] Daňhel M, Kubátová H and Dobiáš R 2013 Predictive analysis of mission critical systems dependability 16th Euromicro Conf. Digit. Syst. Des. DSD 2013 561–6

[9] Sanislav T, Merza K, Mois G and Miclea L 2016 Cyber-physical system dependability enhancement through data mining 2016 20th IEEE International Conference on Automation, Quality and Testing, Robotics, AQTR 2016 - Proceedings pp 1–5

[10] Kreutz D, Ramos F M V and Verissimo P 2013 Towards secure and dependable software-defined networks Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking - HotSDN ’13 p 55

[11] Maroto F 2016 What is the role of Fog Computing in the IOT reference model Big Data Made Simple

[12] Cisco Systems 2016 Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are www.Cisco.com 6

[13] Hosseinpour F, Westerlund T and Meng Y 2016 A Review on Fog Computing Systems Int. J. Adv. Comput. Technol. Hannu Tenhunen 8(5) 48–61

[14] 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, PerCom Workshops 2015 pp 518–23

[15] Gia T N, Jiang M, Rahmani A-M, Westerlund T, Liljeberg P and Tenhunen H 2015 Fog computing in healthcare internet of things: A case study on ecg feature extraction Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on pp 356–63

[16] Chen R Y 2016 Fog computing-based intelligent inference performance evaluation system integrated internet of thing in food cold chain 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2015 pp 879–86

[17] Al Faruque M A and Vatanparvar K 2016 Energy Management-as-a-Service over Fog Computing Platform IEEE Internet Things J. 3 161–9

[18] Hassan M A, Xiao M, Wei Q and Chen S 2015 Help your mobile applications with fog computing 2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking - Workshops, SECON Workshops 2015 pp 49–54

[19] Melnik E V, Klimenko A B and Korobkin V V 2017 The method providing fault-tolerance for information and control systems of the industrial mechatronic objects IOP Conference Series: Materials Science and Engineering vol 177p 12004

[20] Kieckhafer R M, Walter C J, Finn A M and Thambidurai P M 1988 The MAFT Architecture for Distributed Fault Tolerance IEEE Trans. Comput. 37 398–405

[21] Harper R E and Lala J H 1990 Fault-Tolerant Parallel Processor Guid. Control Dyn. 14 554–63

[22] Melnik E V, Klimenko A B, Schaefer G and Korovin I S 2016 A Novel Approach to Fault Tolerant Information and Control System Design 5-th International Conference on Informatics, Electronics & Vision (Dhaka, Bangladesh: University of Dhaka)

[23] Korovin I S, Melnik E V and Klimenko A B 2016 A recovery method for the robotic decentralized control system with performance redundancy International Conference on
Interactive Collaborative Robotics vol 9812(Springer)pp 9–17

[24] Melnik E, Korobkin V and Klimenko A 2016 System reconfiguration using multiagent cooperative principles Proceedings of the First International Scientific Conference “Intelligent Information Technologies for Industry”(IITI’16) pp 385–94

[25] Melnik E V, Klimenko A B and Ivanov D. Y. 2017 The Model of Device Community Forming Problem for the Geographically-Distributed Information and Control Systems Using Fog-computing Concept Proceedings of the IV International research conference "Information technologies in Science, Management, Social sphere and Medicine" pp 132–6

[26] Klimenko A B, Ivanov D I and Melnik E V 2016 The configuration generation problem for the informational and control systems with the performance redundancy Industrial Engineering, Applications and Manufacturing (ICIEAM), International Conference on pp 1–5

[27] Melnik E and Klimenko A 2017 Agent-Based Approach to Distributed Information and Control System Reconfiguration Proceedings of the Computational Methods in Systems and Software pp 196–205

[28] Strogonov A 2002 Dolgovechnost’ integral’nyh skhem i proizvodstvennye metody ee prognozirovaniya ChipNews. 44