Data sharing by means of multiple fog robot servers

A B Klimenko¹, D Y Ivanov¹

¹ Southern Federal University, 2, Chehovast., Taganrog, 347928, Russia
E-mail: anna_klimenko@mail.ru

Abstract. Fog robotics is an entirely new direction in the robotic field, inspired by the fog-computing concept. Some fog architectures have been developed for robots groups and robot swarms, yet, to the best of our knowledge, there are no developed mechanisms of data sharing and replication in such structures. So, they are in the focus of this paper. The distributed ledger-based architecture for the fog robot servers is considered and described, as well as some models have been developed to estimate the time needed for data sharing. Simulation results show the expediency of consensus methods usage for distributed ledger-based.

1. Introduction
The concept of fog robotics was presented and described in this decade. This concept, no doubt, was inspired by fog computing, which can be and has been spread to robotics, and the robot swarms or groups direction in particular. The first studies in the field of fog robotics were presented in the following works [1-3].

These works are an introduction to the fog robotics and contain significant terms and architectures. The following works continue the development of fog robotics [4-6], and some others.

As is proposed in [3], fog robotics is an extension of cloud robotics, as well as fog-computing is. Robots become an edge of the network, and some data preprocessing and storage can be conducted on the robot fog servers. A common scheme of the fog robotics was proposed in the paper [1].

Some other schemes, proposed in the field of fog robotics, describe the extension of the fog robotics concept and include the following schemes: robot interaction with unique fog robot server, the interaction between robots and fog robot server and interaction between robots and multiple fog robot servers.

Among this architectures, the one with multiple fog robot servers is of considerable interest. The reason for this is that robots, solving typical tasks of the robot swarm, sometimes have to exchange the data, or preprocessed data as it takes place in the tasks of objects search and identification [7,8]. However, if a particular robot is bounded to some particular fog robot servers, there emerges the problem of system latency, as far as the data read and write operations must proceed on a remote fog server, located more than in one hop from the robot.

The promising way to avoid this situation is seen as a fog servers data replication organization, and the perfect solution in terms of robot information access is to implement the data storage as a distributed ledger. Nevertheless, the Distributed Ledger Technology [9], in general, consists of two main elements: the data storage structure and the consensus method. As far as there is a great number
of storage types and consensuses, the proposed choice must be modelled, at least, to estimate the pros and cons of the methods selected.

In this paper, we propose a distributed-ledger based scheme of robot interaction. To choose an appropriate distributed ledger technology, we model and estimate some of them in terms of system latency and analyze the simulation results.

2. The cloud and fog robotics
As was mentioned earlier, the proposed fog robotics extends the cloud robotics with the possibility to store, share and preprocess the information, received from multiple robots - in other words, to implement all features proposed by the fog computing.

As regards the application of the cloud robotics, the following fields are topical:
- Google's self-driving cars (Waymo);
- Cloud medical robots [10];
- Assistive robots;
- Industrial robots [11].

Also, such well-known projects as RoboEarth, Rapyuta, MyRobots, COALAS, Rocos use cloud technology. However, despite multiple benefits, cloud robotics has some limitations of usage:
- Controlling the motion of a robot which relies heavily on (real-time) sensors and feedback of the controller cannot benefit much from the Cloud.
- The tasks involving real-time execution require onboard processing.
- Cloud-based applications can get slow or unavailable because of high-latency responses or network hitch. If a robot relies too much on the Cloud, a fault in the network could leave it "brainless."

Fog-computing, solving similar problems for the Internet of Things, reduces the system response time and makes it possible to process some real-time data. Fog Robotics uses a range of resources for computing and storage that are onboard a robot and throughout the network. The Edge of the network consists of a large number of geo-distributed devices owned and operated by various administrative entities.

Resources at the Edge are heterogeneous and implemented in various sizes, e.g. light-weight microservers, networking devices such as gateway routers, switches and access points. These devices communicate with the sensors in a Peer to Peer (P2P) manner or form a cluster depending upon the location or type of the device. Fog Robotics enable[4]:
- sharing of data and distributed learning with the use of resources in close proximity instead of exclusively relying on Cloud resources,
- security and privacy of data by restricting its access within a trusted infrastructure, and
- resource allocation for load balancing between the Cloud and the Edge [12].

Thus, considering the multiple fog robot servers architecture, the following problem becomes topical: how to provide robots with the information, which is accessible on the remote fog server (i.e., more than 1 hop distance from the robot). An obvious and quite new solution is to organize the fog data storages as a distributed ledger, assuming that the fog server provides enough computational complexity.

3. Distributed ledger-based robot fog servers
The distributed ledger is a consensus of replicated, shared, and synchronized digital data geographically spread across multiple sites, countries, or institutions [13].

Nowadays, the distributed ledger is widely used in cryptocurrencies, as well as is an intensively growing technology [9,13,14]. Some interesting distributed ledger applications are proposed in the field of robotics [15]. In the latter work, some paramount ideas of distributed ledger application in robotics are presented and described, including the ledger-based data exchange between robots in the robot swarm.
We assume that the application of the distributed ledger is expedient for data sharing among the fog robot servers. The common architecture and model are presented below in Fig. 1.

Consider the piece of information, which the robot sends for processing, as a transaction. Each transaction sent is stored in the local storage of the fog server, so the information exchange with the Cloud is reduced. Also, each transaction has to be replicated through the fog servers, according to the distributed ledger definition. Transaction replication can be conducted in the way of gossip methods (i.e. randomized rumour spreading) with the time estimation $O \ln (N)$ exchanges, where $N$ is the number of fog servers [16]. Meanwhile, to organize full data replication, the consensus mechanism and the data storage type have to be chosen and estimated.

There is a wide range of data storage types and consensus types. Data can be stored as:
- SQL or NoSQL data storage;
- Blockchain structures;
- Blocklattice structure;
- Hashgraph;
- DAG.

It is expedient to notice that structures like block-lattice, hashgraph, DAG focus on the multiple micro p2p transactions and partially have specific consensus methods. However, it is not needed for fog robot servers organizing. In our case, data exchange is provided by fog servers, so it is their responsibility to disseminate data through the network. So, it seems more expedient to use data storage or blockchain. As to consensus type, the following types were developed:
- Proof-based algorithms;
- Vote-based algorithms.

The first group, proof-based, includes such well-known consensus types as Proof of Work, Proof of Stake, Delayed Proof of Work, Delegated Proof of Stake, Proof-of-Authority, Proof of weight (Proof of reputation, Proof of Space), Proof of elapsed time, Proof-of-Capacity, Proof-of-history, Proof of Stake Velocity, Proof of Importance, Proof of Burn, Proof of Identity, Proof of Activity, Proof of Time, Proof of Existence, Proof of believability and others.

The second group includes Practical Byzantine Fault Tolerance (PBFT), Paxos, Zab, Viewstamped Replication, Raft.

All of those consensus types have their pros and cons in case of the fog server data sharing usage. The examples are presented in Table 1.
Table 1. Pros and cons of consensus types in case of the fog server data sharing usage.

|                         | Advantages                                                                 | Disadvantages                                                                 |
|-------------------------|----------------------------------------------------------------------------|------------------------------------------------------------------------------|
| **Proof-based consensus** | Distributed consensus for untrusted networks (as PoW, for example), the leader is elected by the proof-procedure, which must provide a uniform distribution of the chances to become a leader. Also, Byzantine fault tolerance is provided. | The transaction is not finite; the time of block processing can be unacceptable. |
| **Vote-based consensus** | Relatively fast and straightforward. Some of these protocols are not -byzantine fault tolerant. Transactions are finite (as in PBFT). A leader can be selected via simple round-robin as well as Paxos, conduct a redundant data exchange. At the same time, proof-based consensus generates just transactions and blocks dissemination without any unnecessary procedures of "pre-prepare"."prepare" like it is implemented in PBFT. |                                                                                   |

So, the choice of the consensus type is quite an inquisitive and sophisticated procedure. In essence, the decision must be made with the analysis of future system requirements, assumed network features. However, in this paper, we model and simulate the processes of data sharing for the following cases:

- No fog data sharing, i.e., to get some data from the robot of the other fog servers, the robot requests to those servers;
- Simple non-Byzantine fault-tolerant solution, i.e., as it is implemented in View Stamped replication protocol [17, 18]. View Stamped Replication is a crash-tolerant consensus and provides the common system fault-tolerance;
- Proof-based consensus type, excluding Proof of Work. In this paper, we consider the proof of stake [19, 20] to be an example of proof-based consensus, though the model conducted can describe any of such consensuses.

4. Data sharing models and time estimations

Consider the first case of data sharing, when the situation is shown in Fig. 2 takes place. One can see that without data replication a robot writes the data into the storage of a local fog server, while if there is a need to get some data from other servers, the robot has to send a request to the remote server. We estimate the situations from the time angle.

The following equations describe the time needed for reading and writing operations without data replication.

\[
\begin{align*}
T_{\text{write}}^l &= \frac{N \cdot w}{v} ; \\
T_{\text{read}}^l &= \frac{N w_r}{v} (H + 1) + \frac{N w_a}{v} (H + 1) ;
\end{align*}
\]

Where
- \( N \) is the number of robots linked to the local fog server;
- \( w \) is the size of the data block to be written;
- \( w_r \) is the size of the data block to conduct a request for data;
- \( w_a \) is the size of the data block, which is sent to the robot as an answer to the request made;
- \( H \) is the number of hops between the local robot fog server and the remote one.
Figure 2. The read operation from the remote fog server.

The following describes a simple View Stamped Replication protocol operational phase (the leader change procedure is out of this paper’s focus).

According to the VR protocol, all data replication is conducted by the leader. So, to write some information to the ledger, the piece of data must be sent to the leader node, and after that, the leader node replicates all transactions to its followers.

So, a simple write operation with the replication process takes the following time:

$$T_{write}^2 = \frac{N \cdot W}{v} (H_l + 1) + D \frac{N_w}{v}$$

Where

- $H_l$ – is the distance between the local fog server and the leader node;
- $D$ – is the network diameter.

As all transactions are finite according to the VR, $T_{write}^2$ is the time when all transactions are sent to the followers and so replicated. However, while reading, all robots read from the local server.

The third case of our time estimations is the time estimation of reading and writing operations of some proof-based algorithms. Here every write operation is supposed to be finalized when and only when the block is added to the ledger. So, to estimate the time of write operation, we have to take into account the time, which is needed for consensus reaching between the participants of the process.

$$T_{write}^3 = \frac{N \cdot W}{v} + \ln(N_{fog}) \frac{N_w}{v} + T_{cons} + \ln(N_{fog})W_{block};$$

$$T_{read}^3 = \frac{Nw_r}{v} + \frac{Nw_a}{v}$$

Where

- $N_{fog}$ is the number of consensus procedure participants;
- $W_{block}$ is the size of data block.

It is useful to mention that $T_{cons}$ in case of PoS consensus can be estimated as $O(N_{fog})$ because the consensus procedure consists in the choice of the node with the biggest “stake.” Moreover, the read procedure is conducted just for the local server because of the fully replicated the data.
5. Simulation results
Our first simulation is made for writing operation with the growth of data block to be written. The results are presented in Fig. 3.

![Figure 3](image)

**Figure 3.** The growth of the data size to be written.

One can see that the graph of the simple local write is beneath the lines of Viewstamped Replication write and Proof-of-Stake-like consensus write. This is because all simple write operations are made on the local server. The tendency is similar for the network diameter growth, as is shown in Fig. 4.

![Figure 4](image)

**Figure 4.** The number of nodes and network diameter growth.

However, there is a difference between the graphs in Fig. 3 and Fig. 4. One can see that with the node number growth there is a threshold when the application of the VR-based consensus is inexpedient, as well as there is a threshold of the data volume to be written in the previous figure. So, summarizing, the View Stamped-based consensus is expedient when the data volumes to be written relatively small, as well as the network diameter.
As for reading operations (Fig.5), one can see the dramatic growth of time estimations for the read from the remote fog servers; meanwhile, in case of distributed ledger technologies, all read operations are conducted by means of local fog servers.

6. Conclusion
As the fog robotics is a growing theoretical and applied field, the questions of the inter-robots data sharing are topical. However, to the best of our knowledge, there are no solutions for the fog robotics of how robots can share information through fog servers. In this paper, the distributed ledger is proposed as a way to organize the data sharing in question and to spread data through the network. Besides, some estimations are made based on the elaborated models.

Summing up, the following conclusions can be made:
- The full replication of the data through the fog nodes is of a high need in case of multiple read operations. In this case, the write operations are slower than write operations to the local server. However, in the case of the reading operation, the group of robots reads data from the local fog server, decreasing the time of system response.
- There is a threshold of network diameter and the data volumes to be written, when the VR protocol is more efficient than the distributed proof-based consensus, the write choice of the consensus type depends on the robot group parameters and their operational configurations.

Acknowledgments
The reported study was funded by the RFBR, according to the research project No.19-07-00907.

References
[1] Gudi S, Ojha S, Clark J, Johnston B and Williams M-A 2017 Fog Robotics: An Introduction. IEEE/RSJ International Conference on Intelligent Robots and Systems (Electronic Materials URL: http://hdl.handle.net/10453/120748) pp 1-2
[2] Fadelli I 2018 Fog Robotics: A New Approach to Achieve Efficient and Fluent Human-Robot Interaction EE World online (Electronic Materials URL: https://techxplore.com/news/2018-11-fog-robotics-approach-efficient-fluent.html ) pp 1-3
[3] Chand GUDI S L K, Johnston B and Williams M-A 2019 Fog Robotics: A Summary, Challenges and Future Scope arXiv Prepr pp 1-10
[4] Tanwani A K, Mor N, Kubiatowicz J, Gonzalez J E and Goldberg K 2019 A Fog Robotics
Approach to Deep Robot Learning: Application to Object Recognition and Grasp Planning in Surface Decluttering  Proc. of 2019 International Conference on Robotics and Automation (ICRA) pp 4559–4566

[5] Tian N, Tanwani A K, Chen J, Ma M, Zhang R, Huang B, Goldberg K and Sojoudi S 2019 A fog robotic system for dynamic visual servoing  Proc. of 2019 International Conference on Robotics and Automation (ICRA) pp 1982–1988

[6] Abdulkareem K H, Mohammed M A, Gunasekaran S S, Al-Mhiqani M N, Mutlag A A, Mostafa S A, Ali N S and Ibrahim D A 2019 A Review of Fog Computing and Machine Learning: Concepts, Applications, Challenges, and Open Issues  Proc. of IEEE Access 7 153123–153140

[7] Martinez-Martin E and Del Pobil A P 2017 Object Detection and Recognition for Assistive Robots: Experimentation and Implementation  Proc. of IEEE Robot. Autom. Mag. vol 24 123–138

[8] Liu Q, Zhang C, Song Y and Pang B 2018 Real-time Object Recognition Based on NAO Humanoid Robot  Proc. of 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) pp 644–650

[9] Maull R, Godsiff P, Mulligan C, Brown A and Kewell B 2017 Distributed ledger technology: Applications and implications Strateg. Challeng. 26 481–9

[10] Cao H, Chen R, Gu Y and Xu H 2017 Cloud-Assisted Tracking Medical Mobile Robot for Indoor Elderly  Proc. of 2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC) pp 927–930

[11] Derbel N, Ghomam J and Zhu Q 2019 New Developments and Advances in Robot Control  Proc. of Studies in Systems, Decision and Control vol 175 (Springer) pp 1-359

[12] Song D, Tanwani A K, Goldberg K and Siciliano B 2019 Networked-, Cloud- and Fog-Robotics  Networked Robots (Springer) chapter 24 pp 1–35

[13] Walport M and others 2016 Distributed ledger technology: Beyond blockchain  A report by the Government Chief Scientific Adviser sets out the future of distributed ledger technology (UK Government Office for Science) vol 1 pp 1-88

[14] Burkhardt D, Werling M and Lasi H 2018 Distributed ledger  Proc. of 2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC) pp 1–9

[15] Ferrer E C 2018 The blockchain: a new framework for robotic swarm systems  Proc. of the Future Technologies Conference pp 1037–58

[16] Decker C and Wattenhofer R 2013 Information propagation in the Bitcoin network  Proc. of 13th IEEE International Conference on Peer-to-Peer Computing, IEEE P2P 2013 pp 1-10

[17] Oki B M and Liskov B H 1988 Viewstamped replication: A new primary copy method to support highly-available distributed systems Proceedings of the seventh annual  Proc. of ACM Symposium on Principles of distributed computing pp 8–17

[18] Liskov B 2010 From Viewstamped Replication to Byzantine Fault Tolerance  Replication Part of the Lecture Notes in Computer Science book series (Springer) vol 5959 pp 121–49

[19] King S and Nadal S 2012 Ppcoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake (Electronic Materials URL: www.peercoin.net/whitepapers/peercoin-paper.pdf) pp 1-6

[20] Li W, Andreina S, Bohli J-M and Karame G 2017 Securing Proof-of-Stake Blockchain Protocols  Data Privacy Management, Cryptocurrencies and Blockchain Technology (Springer) vol 10436 pp 297–315