Reducing Delays and Optimizing Model in Fog Computing-Based SCADA Systems

Rashid Alakbarov
Institute of Information Technology, Ministry of Science and Education Republic of Azerbaijan
Baku, Azerbaijan
t.direktor_muavini@iit.science.az

Mammad Hashimov
Institute of Information Technology, Ministry of Science and Education Republic of Azerbaijan
Baku, Azerbaijan
mamedhashimov@gmail.com

***Abstract*—SCADA systems are of great importance in monitoring and safely managing operations in the oil and gas industry. However, these systems face network challenges due to delays in communication channels. Fog Computing technology provides an effective solution to handle these challenges. Processing data at the network edge, explicitly, at points close to the source, increases the response speed of SCADA systems, reduces network load, and enhances security. This article examines the architecture of Fog Computing-based SCADA systems and their implementation possibilities in the oil and gas industry. The proposed mathematical model provides optimal deployment of sensor data in fog devices in Fog Computing-based SCADA systems, upsurges the response speed of the system, lessens network load and balances computing resources.**

Keywords—SCADA systems, FOG computing, Fog Computing-Based SCADA System, Objective Function

INTRODUCTION

SCADA (Supervisory Control and Data Acquisition) systems collect data from various sensors and devices, send this data to a central server or cloud, and ultimately provide system management. Traditional SCADA systems are built on a local infrastructure and rely on physical components such as servers, network equipment, and software. In these systems, data is stored on local servers and SCADA operators can access this data only through an internal network. Setting up such systems usually requires a large initial investment and ongoing technical support. Security is mainly provided by local firewalls and physical security measures. Traditional SCADA systems are ideal for local and stable environments.

Cloud-based SCADA systems operate over the internet and data is stored on cloud servers such as AWS (Amazon Web Services), Microsoft Azure, Google Cloud. These systems can be accessed from anywhere and on any device, which facilitates remote monitoring and management. The cloud-based approach makes it easier and more flexible to scale and update systems. Additionally, it offers a more accessible and cost-effective solution with a monthly service fee model. Cloud-based SCADA is more suitable for modern enterprises with agility, mobile monitoring and analytics needs [1].

However, SCADA systems should operate in real time and analyze data instantly. If the data is first sent to the central server, this can cause delays, decreasing the speed of operations. Fog computing systems have been recently used to overcome the mentioned problems [2]. Fog computing systems refer to a computing model that involves processing data not entirely in the cloud, but on the network edge, on local devices and network devices (i.e., adjacent to sensors). Systems used in the oil and gas industry include thousands of sensors. Each sensor collects data, such as temperature, pressure, gas leakage, etc. Sending all this data directly to the cloud would produce a lot of latency and traffic. Instead, they use nodes located in fog computing systems. These devices are a server or smart device close to the sensors that first processes the data locally, reacts immediately if an abnormality is detected, and only transmits important data to the cloud [3].

Fog computing systems are based on the principle of processing and storing data close to the devices and users. In this approach, data is processed and stored on local devices and networks before being sent to cloud servers. This approach creates opportunities for real-time data analysis and processing. By storing data on local networks before being sent to the cloud, the latency and bandwidth required for data transmission are minimized. Moreover, storing and processing data locally also provides high security, as data is sent to the cloud less often and is better protected on the local network [4].

FOG COMPUTING-BASED SCADA SYSTEMS

SCADA systems used in the automation and monitoring of industrial processes have played an indispensable role in critical areas such as manufacturing, energy, transportation, water supply, and particularly the oil and gas industry for many years. However, since traditional SCADA systems are based on a centralized architecture, they rely on central servers or cloud platforms to process, store, and respond to large amounts of data. This can lead to system delays, data security risks, and overload of network resources. Lately, Fog Computing technology has begun to be integrated into SCADA systems in order to overcome these shortcomings.

Fog Computing is a distributed computing model that provides processing close to the point of data collection, i.e., sensors and control devices, that is, at the "edge of the network". Through this model, data is processed in real time, preliminary analysis is performed, and only important and valuable data is transmitted to the cloud environment. Accordingly, both the system response speed increases and data security and network performance are improved. The advantages of Fog Computing-based SCADA systems are presented below [5, 6].

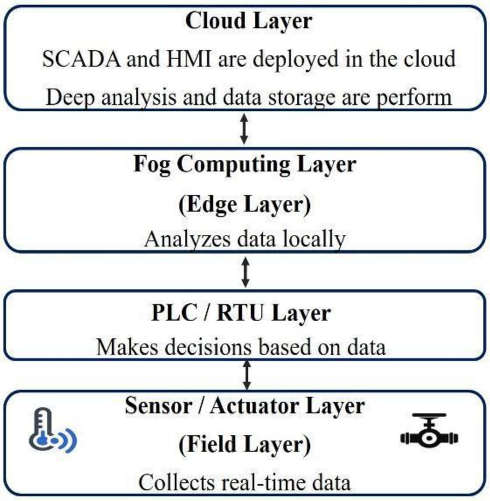
- *Low latency:* Due to the data processing on local networks, latency is significantly reduced. Data can be analyzed immediately and decisions can be made that have an instantaneous impact on operations. This option is specifically imperative for systems that are managed in real time.

- *Bandwidth utilization:* The data collected in SCADA systems can be very large, and sending this data to the cloud overloads the bandwidth. Fog computing processes the data locally, sending only the important data to the cloud. This allows for more efficient use of bandwidth.
- *Security and data protection*: While sending data to the cloud poses additional security risks, fog computing ensures data to be processed locally, which intensifies data protection. Since data sharing between users and devices occurs on local networks, the risks of data leakage are reduced.
- *Resource optimization:* Fog computing ensures resource optimization, as the data is processed locally and only the important results are sent to the central system. This ensures more efficient use of server and other network resources.
- *Seamless integration:* Fog computing can be easily integrated into existing SCADA systems. This allows companies to implement new technologies without replacing their existing infrastructure. This approach can be the only and effective way to improve SCADA system performance.

FOG COMPUTING-BASED SCADA SYSTEM ARCHITECTURE

Traditional SCADA systems have been used for decades to automate and monitor industrial processes. However, since the classic structure of these systems is based on centralized data processing and remote control, they face numerous shortcomings. Issues such as delays, network overload, limitations in real-time decision-making and data security come to the fore. These problems are especially important in critical areas such as energy, transportation, oil and gas, and water supply. Along with the development of new technologies, the Fog Computing approach has begun to be integrated into SCADA systems to address these difficulties. The Fog Computing architecture ensures data to be processed locally, close to the sensors, and only the important results are transmitted to central or cloud systems. This approach brings significant advantages to SCADA systems in terms of low latency, local decision-making, efficient use of bandwidth, and security. Fog Computing-based SCADA systems are built on four main functional layers. This architecture creates both a flexible and reliable industrial control environment by regulating the data flow between different technological layers. The main layers of this architecture and their functions are presented below (fig. 1) [7]:

- *Sensor / Actuator Layer (Field Layer)* - collects real-time data (temperature, pressure, vibration, humidity, level, etc.) through physical sensors and executes commands via actuators. It initially transmits data to Programmable Logic Controllers (PLCs) or Remote Terminal Unit (RTUs).
- *PLC / RTU Layer* - Here, PLC devices make decisions according to programmed logic based on the data collected, and the RTU collects data remotely and sends it to the upper layers. This layer ensures fast and local response of the system.
- *Fog Computing Layer (Edge Layer)* - Data is analyzed at the local level through Fog nodes, edge gateways, and micro-servers. Real-time analysis, data filtering, anomaly detection, and local alarms are performed at this stage. The system can operate effectively even without going to the cloud.
- *Cloud Layer (including SCADA/HMI)* **-** At this layer, SCADA systems control blocks and HMI (Human-Machine Interface) interfaces are deployed in the cloud. Here, operators control the system in real time through web-based panels or mobile applications, send control signals, and all visual monitoring functions are performed.



Fog-based SCADA system architecture

When integrating fog computing into SCADA systems, two components are mainly used: local network devices (edge devices) and fog computing nodes. These components are combined to collect and analyze data, and establish communication between different parts of SCADA systems [8]:

- *Local network devices (Edge Devices)* - perform the initial collection and pre-processing of data. In SCADA systems, this can consist of sensors, actuators, PLCs (Programmable Logic Controllers), and other devices. These devices collect data, perform initial analysis, and send only the important information to local fog devices. This stage is very important as it is the first stage in the data processing, and it reduces the volume of data and transmission time.
- *Fog level (Fog Nodes)* - these devices are more powerful and perform more extensive data analysis. Fog nodes are involved in the control and analysis stages of the SCADA system, process data from various devices in a more extensive way, and then send important data to central servers or the cloud. In this way, operations can be managed faster and more effectively.

OPTIMIZATION MODEL OF DATA DISTRIBUTION FOR FOG COMPUTING-BASED SCADA SYSTEMS

In fog-based SCADA systems, optimal data processing and effective distribution of data (tasks) to devices are of great importance. In these systems, computing power, memory capacity, and response time of each fog device are limited. Therefore, mathematical optimization models are applied to plan tasks in accordance with these resources. The proposed model aims to assign tasks entering the system to fog devices in such a way that the total response time (i.e., the sum of loading and execution times) is minimized. Moreover, restrictions such as the execution of each task on only one device, the memory capacity of fog devices not being exceeded, and the execution of only one task on one device at a time are taken into account. The mathematical model presented in this section formally expresses the optimization objective taking these conditions into account. Along with the model, the definition of variables, the construction of the objective function, and the applied constraints are explained in sequence.

Objective Function Z: It aims to minimize the sum of the times of transferring (loading) to fog devices and executing tasks:

$$min\ Z = \sum_{i=1}^{n} \sum_{j=1}^{m} (L_{ij} + C_{ij}) \cdot x_{ij}$$

here:

- L_{ij}— loading time for sending the i-th task to the j-th fog device,
- C_{ij} — execution time for processing the i-th task on the j-th fog device,
- x_{ij}— binary variable:
- Here x_{ij} is a binary variable:
- $x_{ij} = \{1, if\ the\ i-th\ task\ is\ sent\ to\ the\ j-th\ fog\ device.\ 0,\ otherwise$

Within the following conditions:

Memory Limit. Since the memory capacity of each fog device is limited, the total memory usage of the tasks placed in the memory of each device should not be greater than a certain limit (M_j).

$$\sum x_{ij} \cdot S_{ij} < M_{j,} \quad \forall i \in \{1, 2, \dots, n\}, \quad \forall j = \in \{1, 2, \dots, m\}$$

where:

S_{ij} — the amount of memory used by the i-th task on the j-th fog device,

M_j— the total amount of memory on the j-th fog device.

Task placement: Each task can be executed on only one device. This condition is represented by $\sum x_{ij} = 1$

$$\sum x_{ij} = 1, \quad \forall i \in \{1, 2, \dots, n\} \quad , \quad \forall j = \in \{1, 2, \dots, m\}$$

$$x_{ij} \in \{0, 1\}, \quad \forall i \in \{1, 2, \dots, n\}, \quad \forall j = \in \{1, 2, \dots, m\}$$

The proposed model creates the following possibilities in the SCADA system by minimizing the sum of the loading and execution times of tasks (data):

- *Optimization in terms of time.* Time losses in both the transmission (loading) and processing (execution) stages of tasks are taken into account and the total delay is minimized. This is important for real-time systems.
- *Optimal distribution of tasks.* The model ensures balancing of the computational load by directing each task to the most suitable fog device. Consequently, situations such as overloading of some devices and leaving others vacant are eliminated.
- *Efficient use of fog device resources.* The overall performance of the system increases as the resource requirements (e.g., memory and computing power) of the tasks to be executed are distributed according to the capabilities of each device.
- *Reduced network load.* Due to the loading time component, the model encourages sending data to fog devices as close as possible, which saves overall traffic on the network.

Since the decision variable in the proposed model is $x_{ij} \in \{0, 1\}$, this is a Binary Integer Linear Programming (BILP) task. Algorithms (Branch and Bound, Genetic Algorithm, Heuristics) are used to solve such problems.

Due to these capabilities, the presented model plays an important role in building intelligent, efficient and flexible fog-based control systems.

EXPERIMENT

This problem is a 0-1 binary integer programming model for the optimal distribution of tasks to fog devices. The goal is to minimize the sum of the task transfer (loading) and execution times. Let's explain the problem step by step below. Let's set up an experiment on the model for 4 tasks (data from sensors) and 4 fog devices (nodes). The goal is characterized as follows:

- Assign tasks to devices,
- Each task must go to only one node,
- Each node receives only one task.
- There are 4 tasks (data) and 4 nodes, and each can only be used once, which means 4! = 24 (permutations)

The objective function must be minimized.

$$min\ Z = \sum_{i=1}^{n} \sum_{j=1}^{m} (L_{ij} + C_{ij}) \cdot x_{ij}$$

TABLE I. DATA LOADING TIME (L_IJ)

Task/ Node	Node 1	Node 2	Node 3	Node 4
Task 1	3	2	4	6
Task 2	5	3	2	4
Task 3	6	4	3	2
Task 3	4	5	3	1

TABLE II. EXECUTION TIME OF THE DATA (C_IJ)

Task/ Node	Node 1	Node 2	Node 3	Node 4
Task 1	7	5	6	8
Task 2	4	6	3	7
Task 3	5	7	6	4
Task 3	6	4	7	5

TABLE III. SUM OF LOADING AND EXECUTION TIMES (L + C)

Task/ Node	Node 1	Node 2	Node 3	Node 4
Task 1	10	7	10	14
Task 2	9	9	5	11
Task 3	11	11	9	6
Task 3	10	9	10	6

The task (L + C)-node correspondence and the Z-value of the objective function of the full 24 variants are shown in Table 4.

TABLE IV. COST OF THE OBJECTIVE FUNCTION ACROSS OPTIONS

Variant	Task 1	Task 2	Task 3	Task 4	Z-value
1	N1	N2	N3	N4	34
2	N1	N2	N4	N3	35
3	N1	N3	N2	N4	32
4	N1	N3	N4	N2	30
5	N1	N4	N2	N3	42
6	N1	N4	N3	N2	39
7	N2	N1	N3	N4	31
8	N2	N1	N4	N3	32
9	N2	N3	N1	N4	29
10	**N2**	**N3**	**N4**	**N1**	**28**
11	N2	N4	N1	N3	39
12	N2	N4	N3	N1	37
13	N3	N1	N2	N4	36
14	N3	N1	N4	N2	34
15	N3	N2	N1	N4	36
16	N3	N2	N4	N1	35
17	N3	N4	N1	N2	41
18	N3	N4	N2	N1	42
19	N4	N1	N2	N3	44
20	N4	N1	N3	N2	41
21	N4	N2	N1	N3	44
22	N4	N2	N3	N1	42
23	N4	N3	N1	N2	39
24	N4	N3	N2	N1	40

The lowest value is obtained in option 10 (Z=28). Therefore, we need to place the data from the sensors in the nodes corresponding to option 10 so that the value of the objective function is minimal.

Calculation of the objective function (Z) for option 10:

$$Z = (L_{1,2} + C_{1,2}) + (L_{2,3} + C_{2,3}) + (L_{3,4} + C_{3,4}) + (L_{4,1} + C_{4,1}) = 7 + 5 + 6 + 10 = 28$$

CONCLUSION

The article proposed a model that ensures optimal placement of sensor data to fog devices in Fog Computing-based SCADA systems. The model aimed to minimize the overall loading and execution time and effectively distribute tasks (data) taking into account resource constraints. This approach increases the system's response speed, reduces network load, and balances computing resources, ultimately creating a more reliable and efficient control system.

ACKNOWLEDGMENT

This work is supported by the Science Foundation of the State Oil Company of Azerbaijan Republic (SOCAR) (Contract No. 01LR-EF/2024).

REFERENCES

[1] R. G. Alakbarov and M.A. Hashimov. "Migration İssues Of Scada Systems To The Cloud Computing Environment (Review)," SOCAR Proceedings No.3, 2020, pp. 155-164.

[2] R. J. Tom and S. Sankaranarayanan. "IoT based SCADA Integrated with Fog for Power Distribution Automation," 12th Iberian Conference on Information Systems and Technologies (CISTI), 21-24 June 2017, pp. 1-4.

[3] M. A. Hashimov. "Issues of the use of fog technologies in the IoT environment," Problems of Information Technology, No.2, 2020, pp. 80-90.

[4] R. G. Alakbarov and M.A. Hashimov. "Fog Computing Application In Oil And Gas Industry And Analysis Of Cybersecurity Problems," The 8th International Conference on Control and Optimization with Industrial Applications, COIA 2022, pp. 63-65.

[5] R. Das and M. M. Inuwa. "A review on fog computing: Issues, characteristics, challenges, and potential applications," Telematics and Informatics Reports, Vol. 10, June 2023, 100049.

[6] Hazra, P. Rana, M. Adhikari and T. Amgoth. "Fog computing for next-generation Internet of Things: Fundamental, state-of-the-art and research challenges," Computer Science Review, Vol. 48, May 2023, 100549.

[7] M. D. Stojanović, S. V. Boštjanĉiĉ Rakas and J. D. Marković-PetroviC, "Scada systems ın the cloud and fog envıronments: mıgratıon scenarıos and securıty ıssues," Electronics and Energetics, 32(3), 2019, pp. 345–358.

[8] V. Hurbungs, V. Bassoo and T. P. Fowdur. "Fog and edge computing: concepts, tools and focus areas," International Journal of Information Technology, Vol. 13, 2021, pp. 511-522.