Selecting a Monitoring Technology for a Control System of Distributed Oil Production Facilities

Mykhaylo Lobur, Mykola Malyar*

Lviv Polytechnic National University, 12 Stepan Bandera Street, Lviv, 79013, Ukraine

Received: April 15, 2024. Revised: May 24, 2024. Accepted: June 03, 2024.

Abstract

The article proposes the structure of a SCADA system for monitoring and control of oil production facilities that are distributed over a large area. The main emphasis is on the selection of technology that will enable effective monitoring of the equipment of each oil well. Factors such as reliability, ease of use, availability of protection against third-party interference, as well as availability and accessibility of an open-source software code were taken into account. After reviewing the most common software platforms, a system based on Prometheus and Grafana was selected. It is a combination of the Prometheus time series database server and the Grafana information visualization and analysis system. The important factors that determined the choice of this platform were the availability of the open source code and a large library of ready-made templates for displaying the well parameters in real time. An example of the created visualization window of the dynamometer card of the well, built on the basis of the experimentally recorded data, is presented.

Keywords: SCADA; monitoring; control system; visualization; Prometheus; Grafana; oil pumping unit.

1. Introduction

Automated control systems for technological processes combine various objects and devices, both local and remote, in a single complex enabling efficient control and programming of the operation of both the whole system and its elements, using SCADA systems [1]-[6]. In the contemporary world, automation systems based on programmable logic controllers are widely used. The major advantages they offer are a possibility of partial or complete elimination of the human factor, decreased need for human staff, reduced energy and financial expenditures, and, as a result, an increased production efficiency. Programmable logic controllers are used for collecting and analyzing data from primary sensors, for comparison and processing of parameters using preset algorithms, and for issuing commands to actuators.

In many industries and in the housing sector there are objects located at large distances from each other. Usually, their control is implemented manually, which is inefficient and requires additional time and efforts. Besides, this approach can cause failures and emergencies due to the absence of continuous real-time control. The examples of such objects are water supply stations, oil-producing units and other similar objects [5]-[7].

For real-time control and monitoring of these objects, SCADA systems are created with a single control center, where all means of communication (both wired and wireless) interact and exchange information and control signals. Monitoring is an integral part of the effective operation of such systems. The huge amount of data and the high speed of their processing make early detection of problems, optimization of resources and ensuring the reliability of the SCADA system important. At the same time, the choice of an appropriate approach to monitoring is critically important, since it depends on the quality, reliability and efficiency of the facilities distributed over large areas.

* Corresponding author. Email address: mykola.a.maliar@lpnu.ua

This paper should be cited as: M. Lobur, M. Malyar. Selecting a monitoring technology for a control system of distributed oil production facilities. *Energy Engineering and Control Systems*, 2024, Vol. 10, No. 1, pp. 28 – 34. https://doi.org/10.23939/jeecs2024.01.028

The goal of the paper is to analyze the existing monitoring technologies and to select the optimal SCADA control system for distributed oil production facilities.

2. Formulation of the research problem

The specific features of oil production facilities, such as their geographical distribution on large areas, limited amount of storable data and complexity of surveillance and maintenance imply certain priority requirements for their monitoring and control systems [8]:

- Reliability. As oil production facilities can be located in remote and hard-to-reach areas, control systems must be highly reliable to ensure stable operation even in severe operating conditions.
- Low cost. Taking into account the large number of oil production facilities, control systems should be affordable and cost-effective.
- Protection against unauthorized access and damage. Since oil production facilities are scattered across large areas, including hard-to-reach places, there is a risk of third-party interference in the operation of the equipment or its theft. Therefore, control systems must be equipped with protection measures to prevent unauthorized access.

In general, for effective control and monitoring of oil production facilities, it is important to take into account the above-listed features during the design and installation of SCADA systems for them to meet these priority requirements.

One of the important functions of SCADA systems is the optimization of resources, early detection of problems in the technological process and ensuring the reliable operation of the equipment. For this, SCADA systems must ensure high-speed processing of large volumes of collected data. Therefore, it is very important to select the right approach to the monitoring system and the principles of its design, since the efficiency of the entire SCADA system relies on it.

The foreign companies that develop control systems for oil producing units include Lufkin Automation, eProduction Solutions, ABB, Automation Electronics, DrSCADA Automation, R&M Energy Systems, International Automation Resources, SPOC Automation and others.

The structure of such systems can be in general presented as shown in Fig.1 [4].

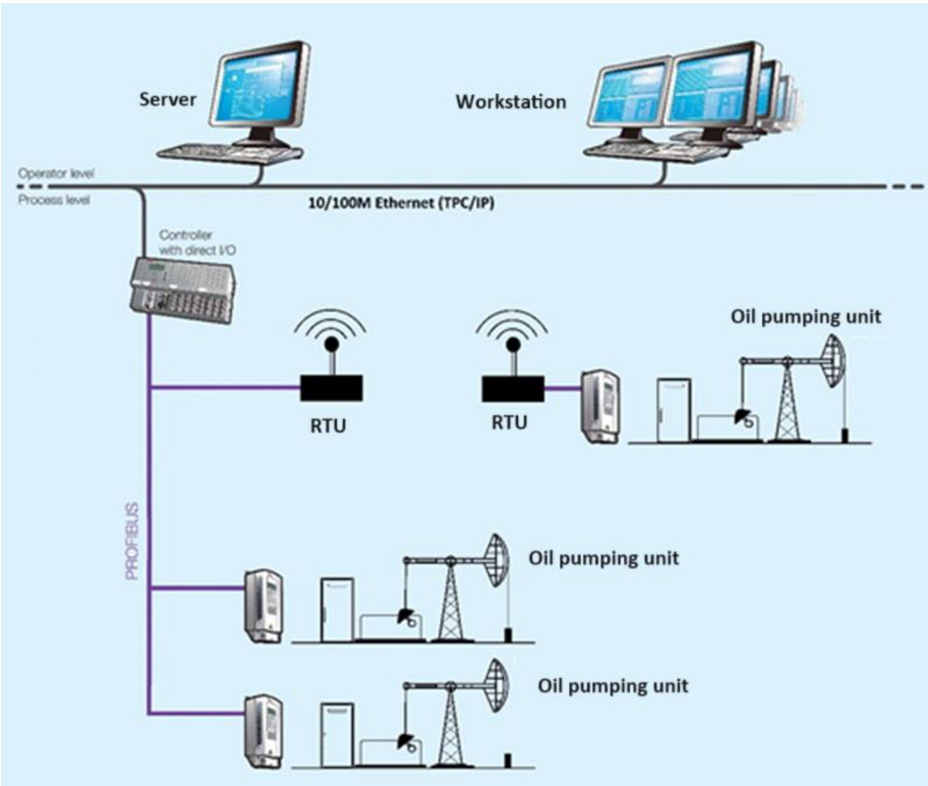

Fig.1. Structure of SCADA system.

The main elements of such SCADA systems are:

- means of collecting primary data from field devices;
- means of control and registration of signals about emergency situations;
- means of storing information with an option of its processing;
- means of processing primary information;
- means of visualization of information using graphs, histograms, etc., depicted in Fig. 1 as Workstation.

3. Features of software for monitoring and visualization

From the above, it can be concluded that monitoring is the main basic function on which the operation of control systems of oil production facilities is based. Today, the vast majority of SCADA systems are implemented on MS Windows platforms using Java. This language is popular due to the possibility of its integration with the native code of drivers of programmable controllers. The main framework for the development of the server and the client part is the Spring Framework. Several popular platforms have been developed on its basis for organizing secure information exchange between network elements. Each of them has its advantages and limitations, which are important to consider when selecting the optimal tool for monitoring and analyzing the condition of oil production equipment. Let us consider each of them.

A REST (Representational State Transfer) API [10], [11] is one of the main architectural styles for developing network applications based on the HTTP protocol. Applying a REST API to monitor distributed objects using Spring Boot and Java can be an efficient approach, especially when dealing with structured data.

Monitoring via the REST API involves pulling metrics and collecting data from distributed objects by making HTTP requests to their remote resources. The main idea is that each object to be monitored exports its metrics via an HTTP endpoint. This data is then collected on a central server for further processing and analysis.

The advantages of the REST API include simplicity of implementation, flexibility and easy integration with other services. The REST API allows you to quickly develop and configure the collection and transfer of metrics from the distributed objects. Using the REST API to pull metrics is a simple and fast way to get data from objects. Software-based queries can receive metrics from different sources and collect them on a central server for analysis. However, this approach may be less efficient for multi-parameter systems, as it requires individual programming for each type of metric. This is the case of oil production facilities. In addition, the lack of ready-to-use tools for data visualization and analysis can limit the system's ability to respond to changes in the state in a timely manner.

Apache Kafka is another popular distributed platform for exchanging messages between different system components [12]-[14]. Apache Kafka can be used to asynchronously monitor and process data from distributed objects. Monitoring via Apache Kafka can be implemented as follows. Each object generates metrics and sends them to the Apache Kafka server. This can be implemented using the Kafka Producer API. On the central server or monitoring agent, a Kafka Consumer is configured to receive data. This agent asynchronously retrieves data from Kafka and stores it for further analysis. Apache Kafka allows an easy scaling of the system by adding additional brokers (servers) and distributed data storage, making it efficient for systems with a large number of distributed objects. There can be either one or several Consumers, as well as Producers (for sending queries) (Fig.2).

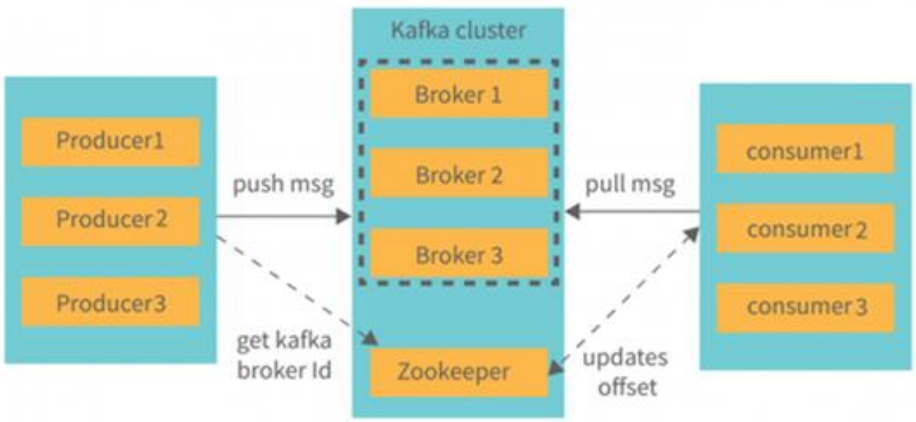

Fig.2. Base components of Kafka.

Apache Kafka requires significant computing and network resources to run efficiently, especially when processing large data streams. As can be seen from the above, Apache Kafka is commonly used to exchange streaming data between different objects. To provide full functionality, such as monitoring and control, additional tools and platforms must be used, which complicates system deployment, configuration, and optimization.

ELK Stack is a combination of three powerful tools: Elasticsearch, Logstash and Kibana, which together provide the ability to collect, store, analyze and visualize a large amount of system data (logs) [15], [16].

Using ELK Stack to collect and analyze logs of distributed objects can help to effectively detect problems, analyze them and respond to events in a timely manner (Fig. 3). Logstash allows collecting data from different objects. Elasticsearch is used to store the data, and Kibana provides the ability to process and visualize the data. Graphs, charts, and dashboards can be created to analyze the system's state.

Fig.3. Structure of ELK Stack.

However, the ELK Stack is not very suitable for objects control, since its main function is to collect and analyze data. For control tasks, ELK Stack requires integration with additional tools or extensions. This, in turn, will lead to increased requirements to computing resources and will complicate setting up and configuring such a system. Another complication with the ELK Stack configuration is the fact that since 2021, new versions of Elasticsearch and Kibana have ceased to be released as open source, which significantly reduced the users' options.

Prometheus and Grafana are the two popular open source platforms widely used for data monitoring and visualization [17]-[19]. The advantage of Prometheus is the flexibility of collecting data from different devices, which it sends queries to, using a certain algorithm. Object control applying Prometheus can be implemented as follows. Metrics Exporters and MySQL Exporter components are used to do periodic scraping of objects in order to collect data. The list of object addresses is recorded in the Prometheus configuration file. All this data is recorded in the time series database and subsequently used to analyze the state of the object or the system as a whole. Based on the analysis, alerts are generated, which are sent using the Alertmanager component to specified recipients.

In its turn, with the help of Grafana, an interface is created for visualizing data obtained from Prometheus in the form of graphs, diagrams, etc. It should also be noted that Grafana has a large set of ready-to-use dashboards that can be customized for a specific task. In our opinion, such a combination (Fig. 4) is most suitable for real-time monitoring of the parameters of distributed objects in order to respond instantly to changes in the equipment operation. In addition, the output code of these software products is open source, which significantly facilitates the possibility of modification and integration into the existing systems.

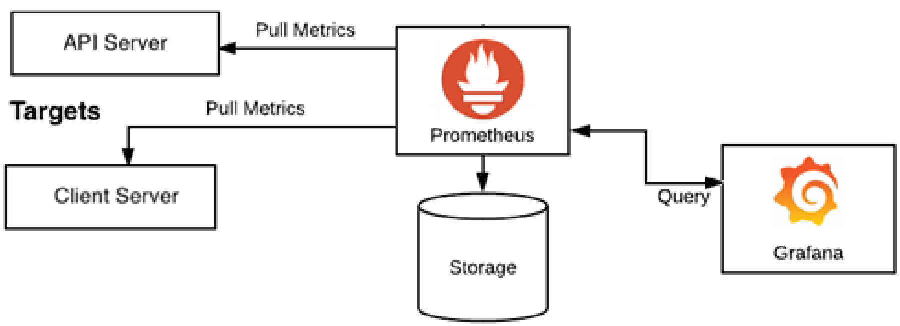

Fig.4. Structure of the monitoring and visualization system based on Prometheus and Grafana.

4. Results

As mentioned above, one of the functions of the SCADA system is obtaining information. For oil production facilities, such information includes real-time monitoring of the technological process and the state of the equipment, logical and computational operations, functions to display the retrospective of the technological process. The important components of the control system for oil pumping units are the sensor of force in the polished rod and walking beam cap sensor. Using the data they output, dynamometer cards are plotted, which show the dependence of the force on the displacement, or the angle of rotation of the crank. The dynamometer card is one of the main characteristics based on which the state of both the surface and underground equipment of the well is assessed.

Monitoring implementation using Prometheus and Grafana involves configuration of Prometheus via HTTP endpoint. Grafana also requires the use of the Micrometer library to configure the data in a user-friendly format. For a start, to simplify the customization of the visualization interface, we used experimentally recorded dynamometer cards that had been previously entered in the database. Grafana enables creating your own dashboards based on the PromQL language, but there is also a library of ready-made dashboard templates. In our case, we used the Dashboard Kubernetes Resource Requests template [20] (Fig. 5). This template has been modified in order to output the recorded dynamometer card of the well.

Fig.5. Dashboard Kubernetes Resource Requests template.

For instance, Fig. 6 shows an example of the output window of the recorded dynamometer cards for a selected well created in Grafana. Information from the well is displayed using the green color. The yellow color is used to display data from an inactive well, i.e. one that is not working at the moment. Since the load of the oil pumping unit is periodic in nature, the window provides information about the duration of the cycle. It should also be noted that in this dynamometer card, the time coordinate is converted into the rotation angle in degrees (0-360°), which corresponds to one cycle.

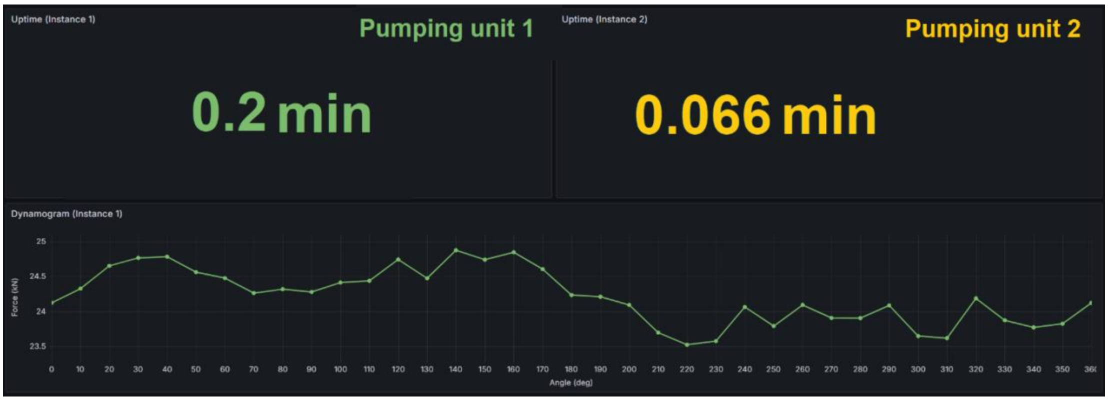

Fig.6. Output window of dynamometer cards recorded in Grafana.

5. Conclusion

Modern SCADA systems are quite intricate software and technical complexes, which are getting smarter and ensure complex control of technological processes. Due to this, developers do not only need to have an in-depth understanding of production flows and communication protocols, but also to be competent in the area of development of in-built systems, data analysis, cloud technologies, etc.

The effectiveness of real-time object control and prevention of emergencies will depend on the extent to which the SCADA system will meet the requirements of multitasking, data processing speed, and user interface friendliness.

Based on the analysis of software products, the Prometheus and Grafana platforms were selected to create a monitoring and data visualization interface. Factors such as data collection and analysis capacities, compatibility with the existing systems, scalability, performance, ease of use, availability of extensions and plugins, cost of licenses and support, and integration with other tools and services were taken into account during the selection process. As an example, a fragment of the software implementation of the visualization of the recorded parameters of the oil well is presented. Further implementation of the monitoring system on the Prometheus and Grafana platform will make it possible to integrate it into the general SCADA system for the control of all wells of the entire oil field.

References

[1] K. Stouffer, J. Falco, K. Kent, T. Grance, R. Ross (2006) Guide to supervisory control and data acquisition (SCADA) and industrial control systems security. NIST Spec. Publ., 800.

[2] Shahzad, A., Musa, S., Aborujilah, A., Irfan, M. (2014) The SCADA review: system components, architecture, protocols and future security trends. American Journal of Applied Sciences, 11, 1418-1425. DOI:10.3844/ajassp.2014.1418.1425

[3] A.Setiawan, Sugeng, K.Koesoema, S.Bakhri, J.Aditya (2019) The SCADA system using PLC and HMI to improve the effectiveness and efficiency of production processes. IOP Conference Series: Materials Science and Engineering. 550. 012008. DOI:10.1088/1757-899X/550/1/012008.

[4] K. Chkara, H. Seghiouer (2022) Criteria to Implement a Supervision System in the Petroleum Industry: A Case Study in a Terminal Storage Facility", Advances in Science, Technology and Engineering Systems Journal, vol. 5, no. 5, 29-38.

[5] Rashad O, Attallah O, Morsi I. A smart PLC-SCADA framework for monitoring petroleum products terminals in industry 4.0 via machine learning. Measurement and Control (2022). no.55(7-8), 830-848. DOI:10.1177/00202940221103305

[6] Al-Fadhli M., A. Zaher (2018) A smart SCADA system for oil refineries. *International Conference on Computing Sciences and Engineering (ICCSE)*, Kuwait City, Mar. 2018, 1-6. DOI: 10.1109/ICCSE1.2018.8373996

[7] R. Pandit, J.Wang (2024) A comprehensive review on enhancing wind turbine applications with advanced SCADA data analytics and practical insights. IET Renew. Power Gener. vol.18, 722-742 (2024). DOI:10.1049/rpg2.12920.

[8] Boyko V.S. (2004). Development and operation of oil fields. RealPrint: Kyiv, 695.

[9] Malyar A. (2016). Study of stationary modes of sucker rod pumping unit operation. Przeglad elektrotechniczny, 12, 255-259. DOI:10.15199/48.2016.12.65.

[10] L. Richardson, M. Amundsen (2013. RESTful Web APIs. Publisher: O'Reilly Media,Inc., 373.

[11] M. Amundsen (2022) RESTful Web API Patterns and Practices Cookbook. Publisher: O'Reilly Media, Inc., 468.

[12] Vohra D. (2016) Using Apache Kafka. In: Pro Docker. Apress, Berkeley, CA. 185-194. DOI:10.1007/978-1-4842-1830-3_12

[13] Wu Z. Shang, K. Wolter (2019) Performance Prediction for the Apache Kafka Messaging System, *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, Zhangjiajie, China, 2019, 154-161. DOI: 10.1109/HPCC/SmartCity/DSS.2019.00036.

[14] Tejas V., Dr. Kiran V. (2020) Development of Kafka Messaging System and its Performance Test Framework using Prometheus, International Journal of Recent Technology and Engineering (IJRTE), 9(1), 1622-1626. DOI: 10.35940/ijrte.A2516.059120

[15] Gurpreet S. Sachdeva (2017) Practical Elk Stack: Build Actionable Insights and Business Metrics Using the Combined Power of Elasticsearch, Logstash, and Kibana. Publisher: APress, 302.

[16] Tong Z. (2015) Elasticsearch: The Definitive Guide. O'Reilly Media, Inc. 686.

[17] M. Yudha Erian Saputra, Noprianto, S. Noor Arief, V. Nur Wijayaningrum and Y. W. Syaifudin (2024) Real-Time Server Monitoring and Notification System with Prometheus, Grafana, and Telegram Integration, *ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS)*, Manama, Bahrain, 1808-1813. DOI: 10.1109/ICETSIS61505.2024.10459488.

[18] Vivek Basavegowda Ramu, Ajay Reddy Yeruva (2023) End-to-End Observability with Grafana: A comprehensive guide to observability and performance visualization with Grafana (English Edition). Publisher: BPB Publications, 710.

[19] Kumar S., Saravanan C. (2021) A Comprehensive Study on Data Visualization Tool-Grafana. Journal of Emerging Technologies and Innovative Research (JETIR), 8(5), f908. Available: http://www.jetir.org/papers/JETIR2105788.pdf.

[20] Kubernetes Resource Requests. Grafana Labs, https://grafana.com/grafana/dashboards/7187-kubernetes-resource-requests (accessed on 2024-02-28)

Вибір технології моніторингу для системи керування розподіленими об'єктами нафтовидобутку

Михайло Лобур, Микола Маляр

Національний університет «Львівська політехніка», вул. Степана Бандери, 12, Львів, 79013, Україна

Анотація

В статті запропоновано структуру SCADA системи для моніторингу та керування об'єктами нафтовидобутку, які розосереджені на великій площі. Головний акцент в статті зроблено на виборі технології, яка дасть змогу здійснювати ефективний моніторинг за обладнанням кожної нафтової свердловини. Під час вибору бралися до уваги такі фактори, як надійність, зручність у користуванні, наявність засобів захисту від стороннього втручання, а також відкритість та доступність програмного коду. У результаті огляду найбільш поширених програмних платформ, було вибрано систему на основі Prometheus і Grafana. Це є поєднання сервера бази даних часових рядів Prometheus з системою візуалізації та аналізу інформації Grafana. Важливими факторами при виборі цієї платформи було наявність відкритого коду та великої бібліотеки готових шаблонів для відображення параметрів свердловини в реальному часі. Продемонстровано приклад створеного вікна візуалізації динамограми свердловини, яка побудована на основі експериментально знятих даних.

Ключові слова: SCADA; моніторинг; система керування; візуалізація; Prometheus; Grafana; нафтовидобувна установка.