DEVELOPMENT OF CONTEMPORARY LOG MANAGEMENT SOLUTION FOR THE SOFTWARE INFRASTRUCTURE OF THE BM@N EXPERIMENT

I. Romanov[a,1], *A. Chebotov*[a,], *K. Gertsenberger*[a,]

[a,]Joint Institute for Nuclear Research, Dubna, Moscow region, 141980 Russia

In the course of developing a software ecosystem of the BM@N experiment, a multitude of systems and services has been implemented, including data processing and information systems. An integral component of the maintenance procedure is monitoring the operation of the systems of the BM@N software infrastructure and requests to them, which are logged to the corresponding files. The paper presents the implementation of a contemporary solution for the log management of the BM@N software complex. The log management solution has been implemented based on a high-performance and cost-effective technology stack of ClickHouse and Datadog Vector. The employment of the Metabase business intelligence platform and Grafana real-time monitoring tool further facilitates the analysis of the collected logs. As a result, the integration of these solutions enhances the resilience of the BM@N software infrastructure and facilitates security auditing.

PACS: 07.05.Bx; 07.05.Kf

INTRODUCTION

BM@N (Baryonic Matter at Nuclotron) is the first experiment operating at the NICA (Nuclotron based Ion Collider fAcility) accelerator complex [1]. The experiment studies the interaction of relativistic heavy ion beams with fixed targets. A set of specialized software [2] has been developed for the purpose of analyzing the data obtained during the experiment. In addition, a collection of information systems has been implemented to provide necessary information on the experiment for BM@N data processing.

The software infrastructure upon which the systems operate must ensure their security, availability, and capabilities for efficient maintenance. One of the responsibilities inherent to maintenance is the investigation of incidents based on the analysis of the log data. Manual analysis of log files is an ineffective method for timely detection of security issues and errors in the operation of the systems. And since the number of the supported systems increases, the manual approach becomes impractical. Additionally, it limits the ability to perform comprehensive log analysis, thereby preventing one from making informed decisions based on log data. To solve the issues, the first version of the log management system has been developed, which automates a collection of logs from monitored BM@N systems and provides a unified storage and tools for their analysis.

[1] E-mail: iromanov@jinr.ru

REQUIREMENTS FOR THE SOLUTION AND USED TOOLS

The main idea of the log management system is to combine within one solution such capabilities as real-time monitoring functions for timely detection of emerging issues and comprehensive analysis of accumulated log data. Therefore, the system must ensure the collection and processing of the log data in real-time with the subsequent storage, provide tools for monitoring and analytics, as well as mechanisms for reporting incidents. It is important that the system is readily scalable, for example, in the case of the introduction of new systems as a part of the BM@N software complex.

Software tools utilized for the implementation of the log management system must be distributed under an open source license, provide high performance for productive log management, and be cost-effective regarding resource consumption. Moreover, they should be proven by large organizations and exhibit a favorable operating history, as well as the ability to easily scale in order to meet evolving needs.

SELECTED TECHNOLOGY STACK

The ELK stack [3] is a common implementation of log management systems. It comprises Elasticsearch for log storage, Logstash for data collection, and Kibana for data analysis. The popularity of the stack can be attributed to the effectiveness of its search system and extensive range of analytical functions. However, Elasticsearch has significant resource requirements, especially disk space and RAM, making it quite expensive to use for implementing the log management system of the BM@N experiment. Furthermore, it is possible that such a search system may prove superfluous for our needs. The ClickHouse DBMS [4] is currently gaining prominence as a log storage solution. It is a column-oriented database management system (DBMS) designed to facilitate high-performance analytics and processing of large data volumes in real-time. ClickHouse has lower resource consumption, which renders it a more cost-effective alternative to Elasticsearch[1], while still ensuring the provision of sufficient search capabilities as required. In view of these considerations, it was decided to utilize ClickHouse for the storage of the logs.

Although ClickHouse does not provide a ready-made stack for implementing a log management system, numerous well-known tools for analytics and log collection are compatible with it. This enables users to select the most suitable technology stack for their given tasks. For the BM@N software infrastructure, the following solutions are selected, as they implement necessary functions for the effective log management and meet the stated requirements. Vector [5] serves as a tool for the collection, processing, and routing of logs and metrics, while Metabase [6] functions as a business intelligence platform for in-depth log analysis. In addition, the Grafana [7] service enables real-time monitoring and visualization of data analysis results.

[1] Based on the following benchmark: `https://github.com/ClickHouse/examples/tree/main/ClickHouse_vs_ElasticSearch/DataAnalytics`

The design of the log management system depends on a selected Vector topology, which comes in three types, such as *distributed*, *centralized*, and *stream-based*. The *distributed* topology was rejected on the grounds that Vector could have a negative impact on the systems it monitored. This is due to the fact that each of its instances, in addition to the collection and routing of logs, also performs processing logic, which may be complex and consequently require an extra amount of resources. In turn, the *centralized* topology permits log management solutions to separate log collection and processing operations, thus reducing the impact on monitored systems. *Vector as an agent* is utilized for the sole purpose of collecting logs. As a result, in the BM@N experiment the logs are processed by *Vector as a central service* and are routed to the storage. Nevertheless, both *distributed* and *centralized* topologies are susceptible to loss of logs, which arises from the loss of buffered data in the event of a Vector instance failure. The issue is resolved through the utilization of the *stream-based* topology, but this solution is quite expensive due to the necessity of employing a stream-based service. In view of the above, the *centralized* topology has been selected for the BM@N log management system as it represents the most acceptable compromise.

It is obvious that *Vector as a central service* is a bottleneck and its failure leads to the inoperability of the log management system as a whole. The issue has been resolved for the BM@N experiment by utilizing multiple instances of *Vector as a central service*, which are managed by a load balancer. The final design of the log management system comprises a set of *Vector as an agent* for the collection of the logs, *Vector as a central service* for log processing, ClickHouse for the log storage, and both Grafana and Metabase for analytics and visualization of the log data.

IMPLEMENTATION OF THE LOG MANAGEMENT SYSTEM

The log management system was implemented in a staged manner. The initial stage of the implementation involved integrating the system with the primary components of the BM@N software infrastructure (see Fig. 1), including the single reverse proxy and Keycloak identity provider. Subsequently, the system was integrated with the BM@N software hosted within the infrastructure. The single reverse proxy serves as a single point of access to all the BM@N software components. This feature enables the centralized collection of access logs from the BM@N systems in a unified format. Moreover, since a new system is deployed in the infrastructure, the access logs are automatically collected from it. During the integration with the single reverse proxy, *Vector as an agent* was deployed on the server, where the *agent* monitors log files generated by the single reverse proxy and routes them to *Vector as a central service*. A special script was added to parse the received logs and convert to the required format for the storage. A table was created in ClickHouse for the purpose of storing the logs. Following this, a set of dashboards were implemented in Metabase and Grafana to visualize the analyzed log data.

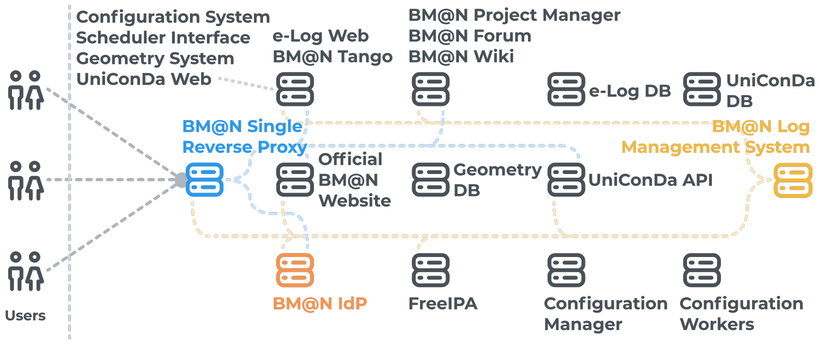

Fig. 1. The design of the software infrastructure for the BM@N experiment

In the software infrastructure of the BM@N experiment, Keycloak is utilized as an OpenID Connect identity provider, assuming responsibility for the authentication and authorization in all the BM@N software systems. This enables the centralized collection of data related to user logins, as well as the additional ability to automatically collect logs upon the deployment of a new software system within the infrastructure. The process of integrating with Keycloak was similar to that of integration with the single reverse proxy. The consolidation of log data from the identity provider and the single reverse proxy into the unified storage provides a powerful tool for security auditing, allowing for more effective management of access and monitoring of potential threats. The operation of servers with the BM@N software systems is monitored by means of the *server monitoring agents*. Currently, the *agent* is a concept that solely gathers data from the */var/log/auth.log* file (or */var/log/secure*, depending on the operating system), which contains information regarding user authentication, as this task was the most pressing. The collection of the data enables the monitoring of unauthorized login attempts and suspicious activity on the servers. Furthermore, the placement of the data in a separate storage location serves to prevent the occurrence of malicious data wipes.

The integration of the log management system with the software systems of the BM@N experiment has been implemented in the same way as with the single reverse proxy and the identity provider. To facilitate the integration of new software systems in the future, a set of documentation has been created. It includes configuration templates and manuals for setting up the Vector solution, scripts for creating necessary tables in ClickHouse, as well as manuals for creating the corresponding dashboards in Metabase and Grafana. This enables more efficient integration and reduces deployment time.

CONCLUSIONS

The developed log management system is an important element of the BM@N software infrastructure. It improves the reliability of the infrastructure components and the hosted systems used for data processing in the experiment via timely detection and investigation of malicious incidents. The system is based on productive and cost-effective solutions, including ClickHouse, Vector, Metabase and Grafana. The software solutions ensure the efficient operation of the system with low resource consumption. The system design enables easy scalability, allowing the introduction of new data sources represented by *Vector as an agent*, for instance, when hosting new BM@N software systems. Moreover, it is possible to add extra analytical tools as needed. Software solutions facilitate user behavior analysis, infrastructure component evaluation, and data-driven decision-making. The implementation of the log management system thus provides the foundation for further enhancement and optimization of the BM@N ecosystem and contributes to the improvement of its security and quality of the services provided.

CONFLICT OF INTEREST

The authors of this work declare that they have no conflicts of interest.

FUNDING

This work was supported by ongoing institutional funding. No additional grants to carry out or direct this particular research were obtained.

REFERENCES

1. *NICA Collaboration.* NICA White paper. Searching for a QCD Mixed Phase at the Nuclotron-Based Ion Collider Facility. — 2014.

2. *Gertsenberger K., Alexandrov I., Filozova I., Alexandrov E., Moshkin A., Chebotov A., Mineev M., Pryahina D., Shestakova G., Yakovlev A., Nozik A., Klimai P.* Development of Information Systems for Online and Offline Data Processing in the NICA Experiments // Physics of Particles and Nuclei. — 2021. — Jul. — V. 52, no. 4. — P. 801–807.

3. *Balashov N., Balashova M., Knigin S., Kutovskiy N.* Using ELK Stack for Event Log Acquisition and Analysis // Modern Information Technologies and IT-Education. — 2021. — V. 17, no. 1. — P. 61–68.

4. *ClickHouse, Inc.* Fast Open-Source OLAP DBMS — ClickHouse. — URL: `https://clickhouse.com/` (online; accessed: 18.11.2024).

5. *Datadog, Inc.* Vector | A lightweight, ultra-fast tool for building observability pipelines. — URL: `https://vector.dev/` (online; accessed: 18.11.2024).

6. *Metabase, Inc.* Metabase | Business Intelligence, Dashboards, and Data Visualization. — URL: `https://www.metabase.com/` (online; accessed: 18.11.2024).

7. *Grafana Labs.* Grafana: The open observability platform. — URL: `https://grafana.com/` (online; accessed: 18.11.2024).