Monitoring of the SPD data management system

Мониторинг системы управления данными SPD

A. Konak[a,], A. Petrosyan[a]*

А.С. Конак[a,], А.Ш. Петросян[a]*

[a] Joint Institute for Nuclear Research, Dubna, Russia

[a] Объединенный институт ядерных исследований, Дубна, Россия

[*] e-mail: konak@jinr.ru

Active work continues on the creation of the SPD (Spin Physics Detector) facility at the NICA accelerator complex, which is located at the Joint Institute for Nuclear Research (JINR). Since the detector or facility will collect a large amount of data, data processing and storage will be carried out in a distributed computing environment. Despite the fact that the detector will not be launched soon, active work is already underway on the infrastructure in which the accumulated data will be stored and processed. The data management system Rucio, a standard solution for data management in the field of high energy physics, is used as a management tool. The system has already been launched, integrated into an offline data processing system, and manages terabytes of data. There are also already active users. And there is a need to monitor the state of the system, its operability, as well as user activity. The report presents the experience of designing a monitoring system for the Rucio data management system.

Продолжается активная работа по созданию установки SPD (Spin Physics Detector) на ускорительном комплексе NICA, который находится в Объединенном институте ядерных исследований (ОИЯИ). Поскольку установка будет собирать большой объем данных, обработка и хранение данных будут осуществляться в распределенной вычислительной среде. Несмотря на то, что детектор будет запущен еще не скоро, уже ведутся активные работы над инфраструктурой, в которой будут храниться и обрабатываться накопленные данные. В качестве инструмента управления используется система управления данными Rucio, стандартное решение для управления данными в области физики высоких энергий. Система уже запущена, интегрирована в автономную систему обработки данных и управляет терабайтами данных. Также уже есть активные пользователи. Существует необходимость в мониторинге состояния системы, ее работоспособности, а также активности пользователей. В статье представлен опыт проектирования системы мониторинга для системы управления данными Rucio.

INTRODUCTION

The SPD (Spin Physics Detector) experiment is one of the key projects of the NICA collider aimed at studying the spin structure of nucleons and other fundamental properties of matter [1]. The successful implementation of the scientific experimental program will generate a large amount of data that requires reliable, scalable, and efficient infrastructure for storage, processing and management.

Rucio [2] has been chosen to manage data in a distributed storage environment. The deployed data management system demonstrates stable and efficient operation. The critical interactions between the workload management system PanDA [3], the data management system Rucio, and the data transfer service FTS3 (File Transfer Service 3) are configured and functioning perfectly. This level of maturity and integration leads us to the next challenge: the need for a comprehensive monitoring system for the Rucio itself.

This article describes the design, implementation, and features of a new monitoring system developed to provide full visibility and control over all aspects of the distributed data system in the SPD experiment.

METRIC TYPES

Rucio collects a bunch of metrics that can be used to monitor different activities, but does not provide a ready-made monitoring solution. It was necessary to: design a monitoring system, deploy all the necessary infrastructure, configure each component, study information provided by Rucio and finally build visualization panels.

Rucio provides different types of information for monitoring:

- **Metrics** provided by Rucio-server and Rucio-daemons. Metric is a measurable numerical characteristic that shows how the system works.
- **Messages** provided by Rucio-server. Generated during replication, deletion, transfer submission processes and stored in the Rucio-DB (Rucio database). Messages consist of information about replication, deletion or transfer submission operations, e.g. scope, name, size, status (successful/failure), duration, etc in JSON (Java Script Object Notation) format.
- **Traces** provided by Rucio-server. Trace is information about the execution of a single transaction in JSON format. Rucio gets GET, PUT traces from Workload Management System – PanDA, and UPLOAD, DOWNLOAD, TOUCH traces from rucio-clients.

COMPONENTS OF THE MONITORING SYSTEM

In order to organize monitoring for the Rucio data management system, it is necessary to deliver all types of information for monitoring to the storage systems of this type of information. Then build graphs by taking information from these systems.

Drawing upon Rucio documentation and the experience of colleagues, an integrated monitoring architecture was designed, leveraging Graphite, OpenSearch, RabbitMQ, and Grafana. (fig. 1). Each component was deployed in a Docker container based on JINR cloud computing infrastructure [4], configured and tested.

The Rucio developers recommend using Graphite to store metrics and this particular solution was chosen. During the work with Graphite, no problems were observed and it was simply integrated with Rucio to store metrics.

OpenSearch was chosen to store messages and traces. It is an open source document-based search and analysis platform based on ElasticSearch. It provides high performance for full-text search for our monitoring system without licensing restrictions, which makes it an ideal solution for scientific projects requiring complete openness.

Queues in RabbitMQ are used for temporary storage of messages and traces before delivery to long-term storage in OpenSearch. RabbitMQ is a high-performance open source message broker. The Rucio developers recommend using ActiveMQ, but RabbitMQ offers better performance and stability. It is optimized to handle large streams of small messages.

Grafana is used to build visualisation panels. It is an open source platform for visualization and analysis of metrics, focused on working with time series. It supports multiple backends simultaneously (Graphite, OpenSearch, databases) and rich possibilities for creating custom dashboards. For our monitoring system, Grafana provides more specialized and flexible tools for analyzing time series and creating interactive dashboards.

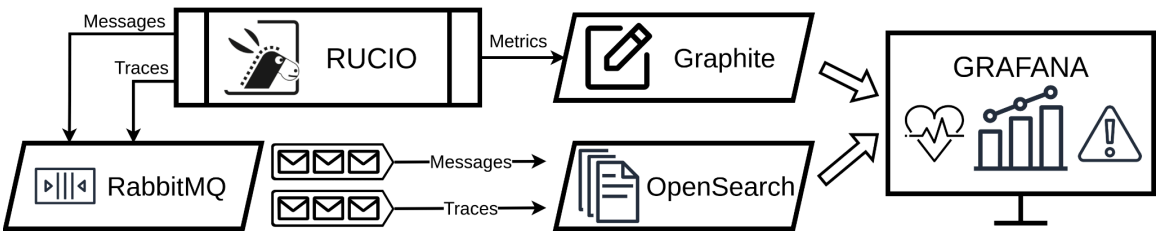

Figure 1 – Components of monitoring system

COLLECTING OF METRICS FOR MONITORING

Metrics are being sended to Graphite automatically after appearing. The lifetime of metrics can be presented by three steps.

1) **Metrics generation**. The Rucio code is instrumented using a StatsD library for collecting metrics. The metrics are collected in a memory buffer and sent in batches to the StatsD daemon on Rucio-server side.
2) **Aggregation and forwarding**. The StatsD daemon pre-aggregates the received metrics and converts them for transmission to Graphite. Finally daemon sends a ready-to-send metric to Graphite.
3) **Receiving and storage**. On the Graphite side Carbon daemon receives metrics and writes them to disk in an efficient format in Whisper DB. Whisper is a fixed-size, time-series database format for storing numerical data over time.

Messages and traces temporarily stored in RabbitMQ. A Hermes (rucio-daemon) sends messages to RabbitMQ. Messages are taken from the Rucio-DB and sent in batches of ten thousand. Traces are automatically sent to RabbitMQ after the Rucio-server receives them.

One unit of information in OpenSearch is a document. Documents are stored in Indices. Index is a logical space for similar documents in OpenSearch. There are different indices in OpenSearch in our setup for different event_type:

- for messages there are indices for transfer submission process (rucio_transfer_queued, rucio_transfer_submitted, rucio_transfer_submission_failed), transfer process (rucio_transfer_done, rucio_transfer_failed), and for deletion process (rucio_deletion_failed, rucio_deletion_done, rucio_deletion_not_found).
- For traces there are different indices for information provided by PanDA (rucio_trace_put, rucio_trace_get), and for information provided by rucio-clients (rucio_trace_upload, rucio_trace_download, rucio_trace_touch).

Messages and traces were being delivered to OpenSearch for long term storage with special consumers. Consumer takes message or trace from the queue, preprocesses document and distributes it to the target index in OpenSearch by event_type of message or trace.

Load test of message consumer, when the Rucio-daemon sends messages to the queue and the message consumer acknowledges messages at the same time, showed that the consumer can handle such a load. Incoming messages have time to be read out and do not accumulate in the queue. For the load trace of trace consumer, 550 thousand traces were queued, and after that the consumer was started. The consumer is able to cope with half a million traces in two minutes, which is quite enough to ensure fast delivery of information from Rucio to the monitoring system during production.

VISUALISATION PANELS

The collected metrics, messages and traces are aggregated into a series of specialized visualization panels in Grafana, providing a comprehensive operational overview of the Rucio data management system. These dashboards are categorized by their functional purpose as follows:

Core System & Daemons Performance. This category focuses on the health and efficiency of the Rucio infrastructure itself. Metrics obtained from the Rucio server and daemons are used as source of information for this category.

- Transfer Submission Monitoring: Tracks the volume and success rate of transfer submissions, including the number of successful/failed requests (fig. 2) and the time spent per submission.

- Deletion Monitoring: Provides insight into the data deletion process, monitoring the number of deletion requests, unlocked replicas, the protocols used, and the average duration from daemon request to storage system response.
- Daemons Activity Dashboard: Offers a unified view of all running Rucio daemons, confirming their continuous operation, activity levels, and absence of critical errors.

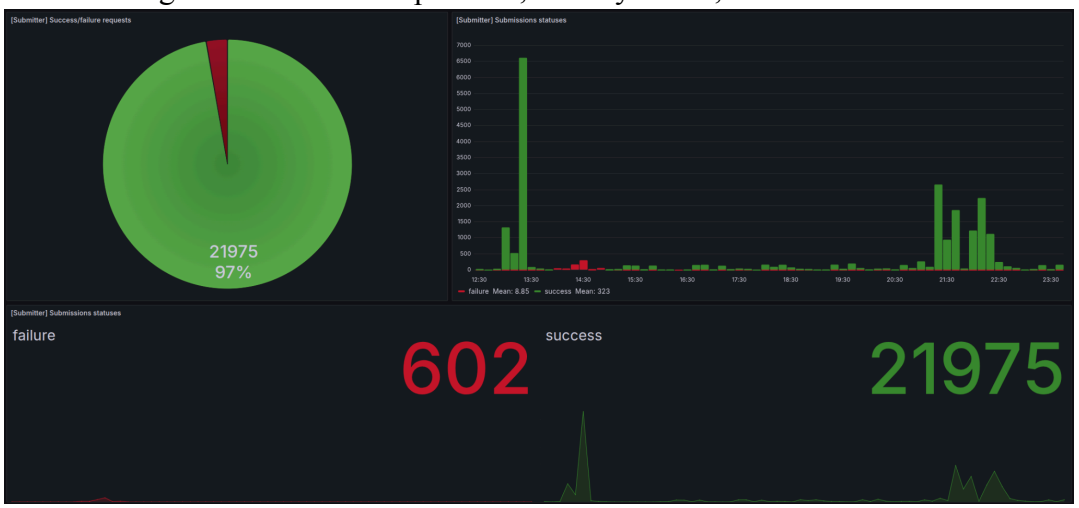

Figure 2 – Transfer submission dashboard

User and Production Activity. These panels offer visibility into how the system is utilized by users and production workflows.

- User Activity Monitoring: Tracks authentication and authorization events (with metrics) in a depersonalized manner, ensuring no information security issues while providing data on system access patterns. With trace data (e.g., upload, download) these dashboards monitor users data flow. They display number of traces, client states (OK/error), error types and counts, activity per Scope/Dataset/Local Site, file sizes uploaded/downloaded with rucio-client, and statistics about using protocols, such as number of activities and average duration of transaction.
- Production Workflow Traces: Based on trace data (e.g., get, put operations), these dashboards monitor the data flow from production jobs. They display the number of traces, jobs per Panda Queue/Remote Site, task ID distribution, client states (OK/error), error types and counts, activity per Scope/Dataset/compute node, and average file sizes used in tasks. Figure 3 shows an example of this dashboard.

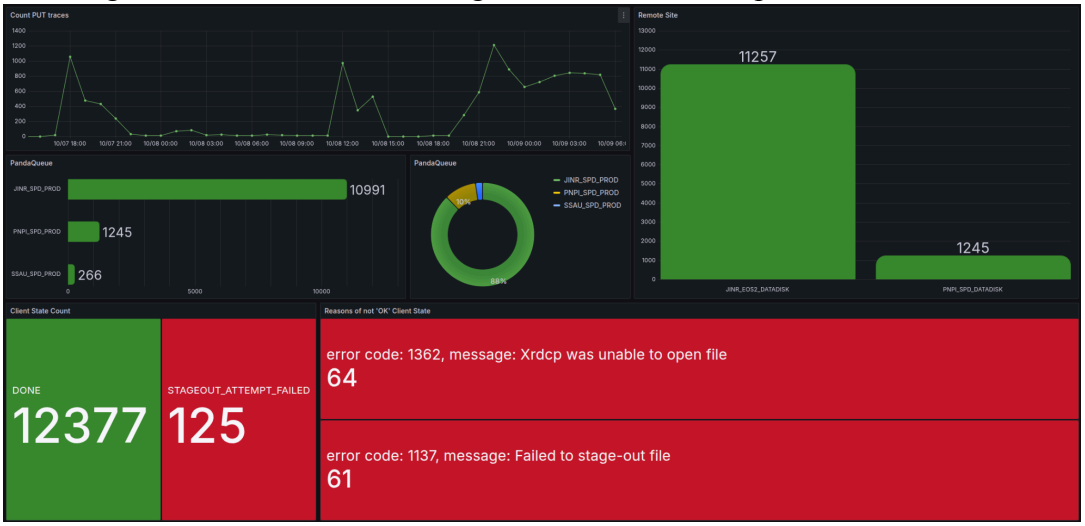

Figure 3 – Production process dashboard

Data Lifecycle Operations. This set of panels delivers detailed analytics on the core data management operations: transfer and deletion. Messages are used for this category.

- Transfer Submission Analysis: Focuses on the submission queue, showing counts of successful and failed submissions, broken down by Scope and Dataset.
- Transfer Execution Analysis: Monitors the actual file transfers, reporting the count of success/failure, the total volume of bytes transferred (overall and per Scope/Dataset), average transfer duration, average file size, and a matrix of successful transfers and data volume between RSEs.
- Deletion Execution Analysis: Tracks the outcome of deletion requests, including counts of success, not_found, and failure outcomes, the total volume of data deleted, average deletion duration, average deleted file size, and deletion statistics per RSE. Figure 4 shows an example of the deletion process dashboard.



Figure 4 – Deletion process dashboard

CONCLUSIONS

A comprehensive and robust monitoring system for the Rucio Data Management system has been successfully designed and deployed. This system is built upon a sophisticated pipeline for the collection, processing, and visualization of Rucio's internal metrics, traces, and messaging data. Its implementation provides real-time visibility into the entire data lifecycle, the core health of Rucio servers and daemons to the intricate details of user and production activities.

The operationalization of this monitoring framework has fundamentally enhanced our operational reliability. It enables the proactive detection of anomalies and the rapid diagnosis of a wide range of typical errors, significantly reducing the mean time to resolution (MTTR) and ensuring the high availability and performance of the data management infrastructure. This is a critical milestone for ensuring the smooth execution of the SPD experiment's data processing workflows.

Looking forward, the evolution of the monitoring system will focus on two primary directions. First, it is planned to continuously refine and expand the existing dashboards and alerting mechanisms based on operational experience and evolving requirements. Second, a key strategic priority is the finalization and in-depth analysis of user activity monitoring, aiming to derive deeper insights into usage patterns and further optimize the user experience. The ongoing development of this system will be pivotal in supporting the escalating data challenges of the experiment.

REFERENCES

1. Abazov V.M. et al. Technical Design Report of the Spin Physics Detector at NICA // Natural Sci. Rev. — 2024. — V. 1, no. 1. — doi.org/10.54546/NaturalSciRev.100101.
2. Barisits M. et al. Rucio: Scientific data management // Comput. Softw. Big Sci. — 2019. — V. 3.
3. Maeno T. et al. PanDA: Production and Distributed Analysis System // Comput. Softw. Big Sci. — 2024. — V. 8, no. 4 — doi.org/10.1007/s41781-024-00114-3.
4. N. Balashov, I. Kuprikov, N. Kutovskiy et al. Changes and Challenges at the JINR and Its Member States Cloud Infrastructures // Physics of Particles and Nuclei — 2024. — V. 55, P 366-370. — doi.org/10.1134/S1063779624030092.