ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA033

DEVELOPMENT OF STATUS ANALYSIS SYSTEM BASED ON ELK STACK AT J-PARC MLF

K. Moriyama, T. Nakatani[1], Y. Yasu[2], H. Ohshita[2] and T. Seya[2]
Comprehenshive Research Organization for Science and Society (CROSS), Tokai, Ibaraki, Japan
[1]Japan Atomic Energy Agency (JAEA), Tokai, Ibaraki, Japan
[2]High Energy Accelerator Research Organization (KEK), Tokai, Ibaraki, Japan

Abstract

In recent neutron scattering experiments, large quantities of various types of data, including raw data, metadata, logs and metrics have been generated by the system, apparatus and devices. At J-PARC MLF, it is possible to conduct many experiments under various conditions within short time by using high-intensity neutron beams, high-performance neutron instruments, and various sample environments. In this experimental environment, it is essential to perform efficient and effective data analysis. Additionally, since it has been almost nine years from the start of operation in MLF, the rate of occurrence of failures is rising due to ageing of devices. Given that such failure can lead to loss of precious beam time, failure or its signs should be detected early. The MLF status analysis system based on Elasticsearh, Logstash, and Kibana (ELK) Stack, which is one of the web-based framework that is being rapidly adopted for big data analysis, collects various data from neutron instruments. It offers insight to decision-makers in terms of data analysis and experimentation as well as instrument maintenance, by facilitating flexible user-based analysis and visualization. In this paper, we present an overview and the development status of our status analysis system.

INTRODUCTION

J-PARC MLF

The Materials and Life Science Experimental Facility (MLF) at the Japan Proton Accelerator Research Complex (J-PARC) is an experimental facility for neutron scattering, providing domestic and international users from a wide variety of research fields with one of the highest intensity pulsed neutron beam in the world since 2008. Currently 21 neutron instruments are operational in this facility. Each instrument is equipped with a large-area neutron detectors and a wide variety of purpose-built sample environment equipment.

Status Analysis with Log Information

Since neutron instruments include a wide variety of systems, apparatus, and devices, large amounts of diverse data, including raw data, metadata, logs, and server metrics, are generated. These data contain useful information from the viewpoint of instrument operation and data analysis, such as operating state, physical values, and neutron detection. However, there is not much opportunity to positively utilize these logs. This is because these logs are usually generated and stored in different system and have different data structure. Moreover, there is no convenient tool to analyze these logs. Therefore, there is a need for as tool that can provide useful insights into actions required to prevent failure of system and devices, as well as for performing data analysis, especially, since it has been almost nine years from the start of operation of MLF, and system and device failures occur frequently owing to age-related degradation. It has been challenging to develop such a tool until now. However, the rapid growth of big data analysis frameworks of late has made it possible to develop such a tool.

Big Data Approach

Figure 1 shows an overview of our status analysis system. We employed the ELK Stack [1], an open-source software framework developed by Elastic Inc. for big data analysis, to develop the system. This system collects log information with a wide variety of structures from various types of systems and devices installed in the neutron instruments in real-time. Moreover, it can be used to flexibly and easily analyze and visualize log information via a web-based interface. Furthermore, we plan to combine the system with a machine learning scheme to facilitate anomaly detection and advanced status analysis.

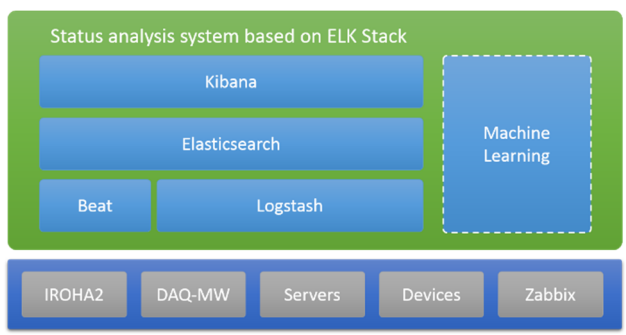

Figure 1: Overview of the status analysis system.

ELK STACK

The ELK Stack consists of three main components, namely, Elasticsearch, Logstash, and Kibana, and a subcomponent called Beat. Figure 2 shows the architecture of ELK Stack.

Elasticsearch

Elasticsearch is a document store based on a distributed document-oriented database with a full-text search engine

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA033

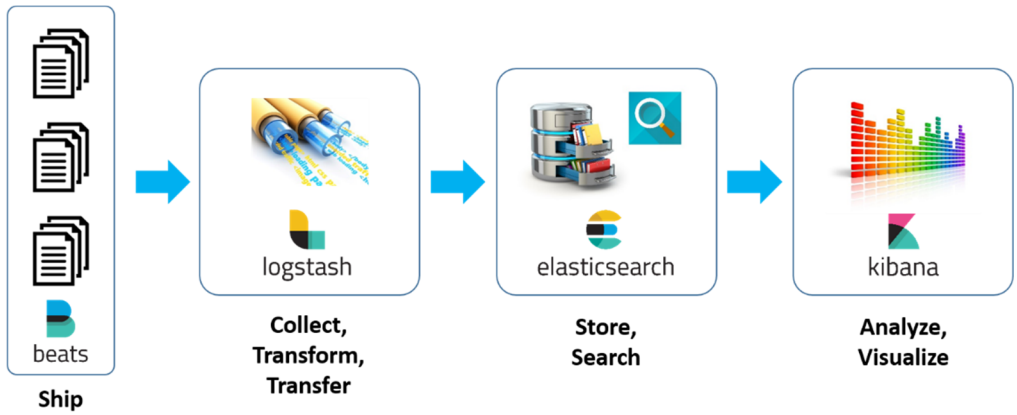

Figure 2: Architecture of ELK Stack.

built on top of Apache Lucene [2], an open-source Java library for full-text search. Because full-text search engine extracts data strongly related to retrieval conditions, Elasticsearch performs data retrieval more flexibly and quickly compare to a traditional RDBMS, retrieving only the data that are exactly matched with the retrieval conditions. This flexibility is achieved by using document-oriented architecture.

Elasticsearch indexes each log data entry as a JSON [3] document, a flat list of key-value pairs, without explicitly providing a schema for multitenancy. However, such a schema can be defined whenever required.

Moreover, Elasticsearch was designed as a cluster system for horizontal scalability and reliability, and can easily introduce additional nodes in response to the amount of data and queries to be handled. Each indexed document is subdivided into multiple pieces, called shards, and replicas. The number of shards and replicas can be configured as necessary. The distribution of these shards and replicas on nodes improves reliability.

Elasticsearch provides a RESTful API using JSON documents over HTTP. Therefore, any handling such as document operation, parameter settings and status confirmation can be easily performed via an HTTP interface from external applications.

Logstash

Logstash is an integrated tool written in Ruby for log collection, parsing and shipping. Logstash can centrally collect log events from various sources, analyze and transform them, and then ship the processed data to arbitrary data stashes such as Elasticsearch. This series of processing steps is performed in a pipeline composed of elemental plugin components: input, filter and output. These plugin components have a wide array of processing functions, such as plug-in and library types.

Kibana

Kibana is a powerful analytics and visualization front-end tool, implemented in JavaScript. Kibana facilitates flexible data search by using the full-text search engine of Elasticsearch. It can be used to freely select and visualize various parameters of searched data, leading to the discovery of a correlation with multiple parameters and data. Moreover, it can detect the causes and signs of system failure. By using the dashboard function, arbitrary graphs can be organized in accordance with objectives.

Beat

Beat is a light weight log shipper running on servers as an agent. It consists of a few components for wire data, log file, server metrics, and windows events. Beat can collect log with various structures. Beat monitors log event and send it Logstash or Elasticsearch.

LOG GENERATION

Neutron instruments at MLF consist of a wide variety of apparatus and systems such as beam control apparatus, data acquisition (DAQ) system and sample environmental devices. Thus, many logs of various types are generated in neutron experiments. As shown in Figure 1, there are some components generating log information pertaining to our neutron instruments.

IROHA2

Many neutron instruments employ a standard instrument-control framework called IROHA2 [4, 5], which integrates DAQ and device control. IROHA2 supports about 20 types of devices, so almost all standard devices used in MLF, such as beam chopper and slit, goniometer, temperature controller, magnet and DAQ system, can be controlled using IROHA2. Experiment, which is a series of device control and DAQ, is performed through the web-based interface of IROHA2. IROHA2 generates various logs in CSV format related to experiments, including the control and status of devices and DAQ, as follows:

- **Surveillance log:** timestamp, log level, and physical values are recorded at fixed time.
- **Measurement log:** timestamp and physical values are recorded at fixed time during an experiment.
- **Operation log:** timestamp, log level and messages are recorded each time any operation is performed.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA033

DAQ-Middleware

DAQ-Middleware (DAQ-MW) is a data-acquisition software framework employed in many neutron instruments at MLF [6]. This software framework is a distributed system based on component architecture, so that it can be flexibly adapted to various types of detector and data rate. Because DAQ-MW has a web-based interface, DAQ can be performed in a web browser or in IROHA2. DAQ-MW generates an operation log. In addition, it is possible to generate logs including those of neutron-detection events as histogram data by using the monitor component of DAQ-MW.

Zabbix

Zabbix is an OSS for integrated monitoring of servers and network equipment [7]. This software monitors life-and-death state and process, and detects failure at the threshold-base. Many servers at MLF are monitored using Zabbix. Zabbix has an RDBMS as its backend. It is possible to collect log information related to server operation by using Zabbix API.

Servers

At MLF, the computer systems of neutron instruments of many components such as DAQ system, instrument control system IROHA2, and analysis system, adopt Linux as the operating platform. Linux creates various logs including system log, access log, and application log.

Devices

There are some devices out of the control of IROHA2; for example, the devices brought at neutron instrument by facility user for temporal use. If the device has a TCP/IP connected, it is possible to acquire the log data by using Logstash.

STATUS ANALYSIS SYSTEM

System Structure

In the status analysis system, each main component of the ELK stack runs on different physical servers. Table 1 shows the specifications of the servers.

Elasticsearch has a cluster configuration composed of 2 physical servers because indexing and searching process can be memory- and CPU-resource-intensive depending on the amount of log events to be processed. Especially, this problem becomes sever when performing the indexing of a huge amount of documents simultaneously. We were often confronted with service outage of Elasticsearch in the early stages of development. In the Elasticsearch cluster, the master node distributes loads and document within nodes. Figure 3 shows the CPU loads of two cluster nodes monitored by Zabbix when indexing a large number of log events. The upper figure shows the master node. It is found that the load is balanced equally across nodes.

Both Logstash and Kibana run on a single server. In addition, Beat runs on the server generating logs to be shipped to Logstash.

Test for Log Collection

We applied the status analysis system to SENJU, a single-crystal neutron diffractometer installed at BL18 in MLF.

Table 1: Server Specifications

Server	Specifications	Unit
Elasticsearch	Dell PowerEdge R410	2
OS	Scientific Linux 6.9	
CPU	Intel Xeon E5620, 2.40GHz	
Memory	12GB	
HDD	1TB SATA 7.2k rpm x4	
Elasticsearch	5.6.1	
Logstash, Kibana	Dell PowerEdge R320	Each 1
OS	Scientific Linux 6.9	
CPU	Intel Xeon E5-2420 v2, 2.20GHz	
Memory	16GB	
HDD	1TB SATA 7.2k rpm x4	
Logstash, Kibana	5.6.1	
Filebeat	5.6.1	

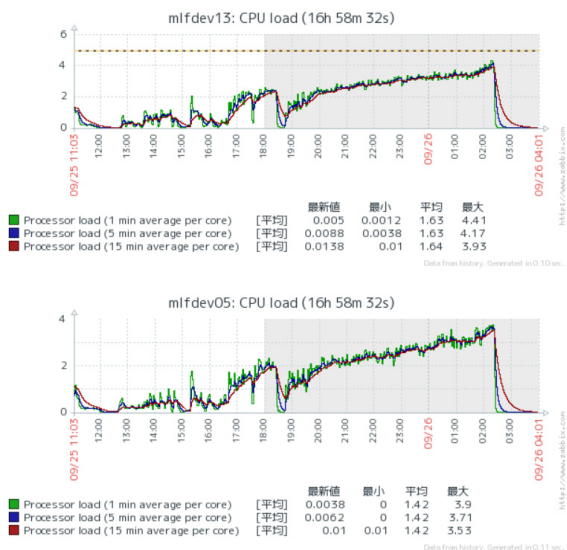

Figure 3: CPU load of cluster nodes in Elasticsearch. Upper figure shows the master node.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA033

We collected logs generated by IROHA2. Currently, eight devices, including beam chopper, temperature controller, goniometer, and magnet, are controlled by IROHA2 at SENJU. We ran Filebeat, a subcomponent of Beat for log file, on the IROHA2 server. Filebeat monitors the logs generated by IROHA2 and transfers them as log events to Logstash. When shipping log events, Filebeat adds the key-value pair to identify the log type of IROHA2 from among surveillance, measurement, and operation, and the device type in the shipping document. We adopted the index name like "(log-type)-YYYY.MM.dd" in Elasticsearch, which is the default type in Logstash. The number of primary shards and replica shards were 1 and 0, respectively, and the document schema was not defined. This is the simplest condition for testing and brief performance evaluation from the viewpoint of indexing. Of course, the number of shards should be optimized eventually for distributed processing and for improving availability. Moreover, defining the schema could influence the indexing performance.

Dashboard and Graph for Status Analysis

We prepared a number of graphs and dashboards with Kibana, and we intended to utilize the status analysis of the IROHA2 log for the following three use cases considering instrument operation and decision-making such as experimental planning during and data analysis after experiment:

Analysis for experimental planning Mainly for analysis of surveillance logs of IROHA2 during experiment, it is possible to monitor the experiment progress. In addition, correlation analysis of sample device parameters and neutron detection is possible, which supports decision making for experimental planning.

Analysis for experimental data analysis Intended mainly for analysis of the measurement log of IROHA2 after experiment, it is possible to search, analyze and download parameters of the sample environment required for experimental data analysis.

Analysis for instrument operation Intended mainly for integrated analysis of operation log and surveillance log of IROHA2, it can provide insights into the operating status, including error and warning occurrence, and device usage condition. The result of this analysis can be used to detect failure and its signs, as well as for improvement of the experimental environment in the future.

Figure 4 shows images of dashboard for these status analysis. These graphs and dashboards can be easily and quickly customized according to the intend use of users.

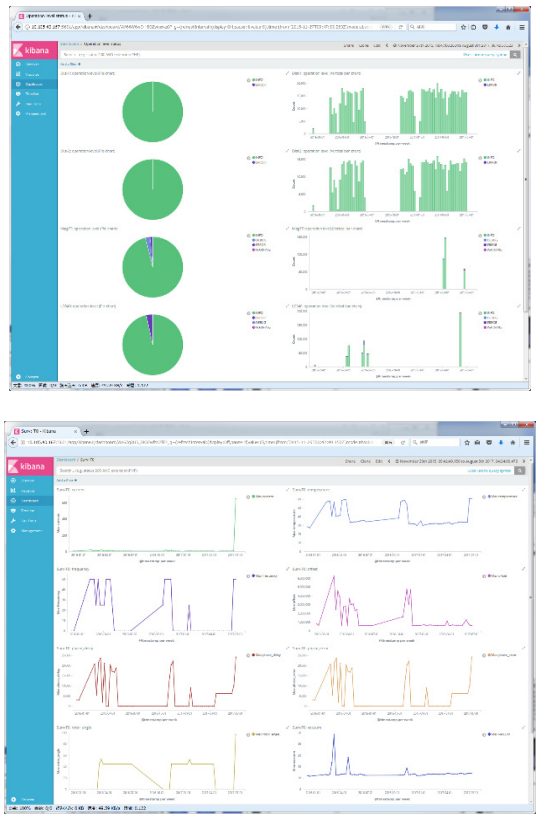

Figure 4: Dashboards for status analysis.

CONCLUSION AND FUTURE PLAN

We have developed a status analysis system based on the ELK stack to integrally analyze log information generated in neutron experiments. By adopting this system for the neutron instruments at MLF, it is possible to easily and flexibly analyze log information, and gain insights into data analysis and experimental planning, as well as instrument operation.

To start full-scale operation, we are plan to advance the following developments.

- Log collection from DAQ-MW, devices, and servers.
- Performance evaluation based on a practical data rate.
- System optimization in terms of cluster configuration, server resources, and indexing parameters in Elasticsearch.
- Establishment of more practical and efficient schemes of status analysis.
- Introduction of an online machine-learning scheme for anomaly detection and advanced status analysis.

REFERENCES

[1] ELK Stack, https://www.elastic.co/products
[2] Apache Lucene, https://lucene.apache.org
[3] JSON, http://www.json.org
[4] T. Nakatani *et al.*, "The Implementation of the Software Framework in J-PARC/MLF", in Proc. ICALEPCS'09, Kobe, Japan, (2009) p.673.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA033

[5] T. Nakatani *et al.*, "IROHA2: Standard instrument control software framework in MLF", in Proc. of the New Opportunities for Better User Group Software NO-BUGS2016, Copenhagen, Denmark, (2009) p.76.
[6] Y. Yasu *et al.*, "Development of DAQ-Middleware", in Proc. of the 17th International Conference of Computing in High Energy and Nuclear Physics CHEP09, Prague, Czech Republic, (2009) p.022025.
[7] Zabbix, http://www.json.org