ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-THPV006

DESIGN OF REAL-TIME ALARM SYSTEM FOR CAFe

N.Xie,Y.H.Gua, R.Wang, B.J.Wang, IMP, LAN Zhou 730000, P.R. China

Abstract

In accelerator control, the alarm system is a very important real-time monitoring and control system. In order to find specific failures of accelerator-related equipment in time, improve the high availability of the equipment, and ensure the long-term operation of the accelerator. An accelerator alarm system based on Kafka was designed and built on the CAFe. The system uses Phoebus for architecture deployment. Kafka is used as the streaming platform of the alarm system, which effectively improves the throughput of the system and realizes real-time alarms.In order to realize the function of remote monitoring of data in the central control room, CS-Studio is used to draw the opi interface to deploy to the enterprise WeChat platform to realize remote data monitoring. This system greatly improves the response speed of fault handling and saves a lot of valuable time for accelerator fault handling.

INTRODUCTION

China Initiative Accelerator Driven System (CiADS) plays an important role in the safety of spent fuel handling. This is a global challenge that has not yet been resolved by our country and the international nuclear energy community. Chinese ADS Front-end Demo Linac (CAFe) as a CiADS prototype. Figure 1 is the layout of the CAFe superconducting linear accelerator, which consists of nine parts. CAFe as a prototype of CiADS.Its research purpose is to develop clean, efficient and safe nuclear fission energy, and to solve the future energy supply. Therefore, the stable operation of the CAFe equipment is particularly important for the debugging and stable operation of the beam experiment.

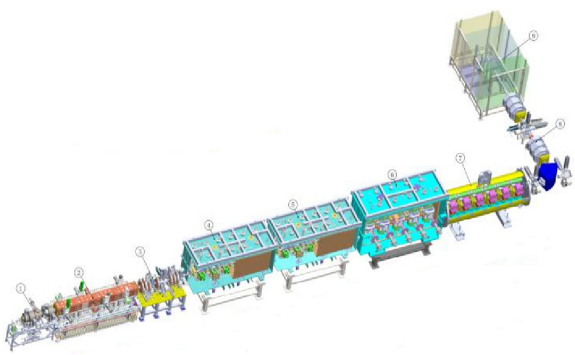

Figure 1: Layout of CAFe Superconducting Linear Accelerator.

SYSTEM STRUCTURE

Figure 2 is the overall framework of the alarm system, which is mainly composed of the control layer, the alarm service layer, the Kafka layer and the application layer.

Control Layer

IOC (input output controller) is the executor of control tasks[1]. It is used to obtain the data of the monitoring equipment and define the alarm threshold of the equipment. The sending of alarm information is based on the setting of the threshold. Each record in the IOC has multiple fields, “SEVR”, “ACKT” “STAT” and “ACKS” are related fields of the alarm and can be customized.

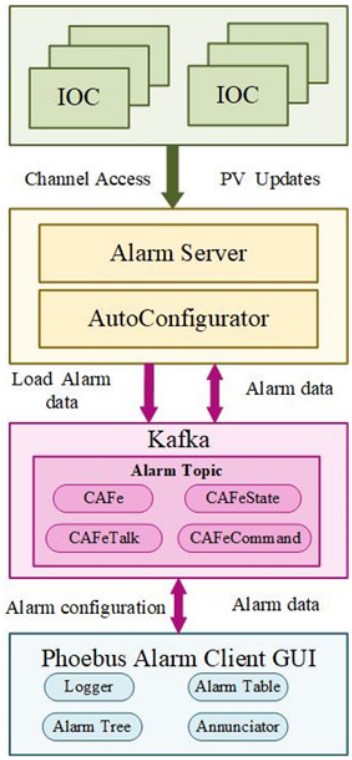

Figure 2: Framework diagram of an alarm system based on Kafka.

Alarm Service Layer

Alarm Server connects all PVs and obtains the records that need to be monitored through the CA protocol and configuration files. At the same time, Alarm Server collects the command information from the upper application “Acknowledge” in the topic "CAFeCommand" in Kafka to confirm the alarm. When the alarm status changes in the records, the Alarm Server will generate a new alarm to update the content in the "AcceleratorState". AutoConfigurator obtains the recorded configuration information and delivers alarm configuration information to the topic “CAFe”.

Kafka Layer

The Kafka layer generates 4 themes: Accelerator,AcceleratorState, AcceleratorCommand, and AcceleratorTalk. Alarm Server obtains current alarm configuration information, real-time alarm status, Acknowledge commands, and voice alarm records from the above 4 topics respectively.

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-THPV006

Application Layer

Phoebus subscribes to Kafka Topic to realize visualization of alarm information. It can display Alarm Tree, Alarm Table, Alarm Area Panel, Logger and Annunciator user interface. The staff in the central control room can know whether the accelerator is operating normally according to the alarm interface.

Phoebus

Phoebus is a new version of Control System Studio, which eliminates the dependence on Eclipse RCP and SWT, which is conducive to the development of control system users. Phoebus is the graphical interface of EPICS control system. It provides a simple and easy-to-operate alarm interface for the central control room. The alarm module components are divided into server and client. The Alarm Server component on the server side monitors the status changes of the PV in the IOC, and sends data to Kafka based on the previous configuration and status of the PV. The server subscribes to the topic in Kafka, and displays the alarm interface on the Phoebus architecture: Alarm Tree, Alarm Table, Alarm Area Panel, Alarm Logger, and Annunciator. The functions are shown in Table 1.

Table 1: Alarm

Application	Function
Alarm Tree	Display hierarchical alarm data flow, record the type and severity of alarms
Alarm Table	Display real-time alarm information of process variables in tabular form
Alarm Area Panel	Display the name of the top alarm layer, which can be used to display the basic alarm status indication interface
Alarm Logger	Record all statuses associated with the service archive and the alarm server
Annunciator	Display the alarm time, alarm security level and the description of the alarm PV and send out the alarm voice

HARDWARE

Figure 3 is a diagram of the hardware architecture of the control system. The operator interface uses Linux operating system to run Phoebus to realize alarm visualization and complete the human-computer interaction process. IOC provides a physical interface for the device and communicates directly with the underlying device[2].

IOC uses records to limit the description of the controlled device, so that the value obtained by the controlled device is divided into alarm levels.LAN realizes the communication between OPI and IOC. IOC and OPI adopt CA (Channel Access) channel access mechanism。CA includes client CAC (CA Client) and server CAS (CA Server), which provide application interface libraries for OPI (Operator Interface) and IOC respectively[3]

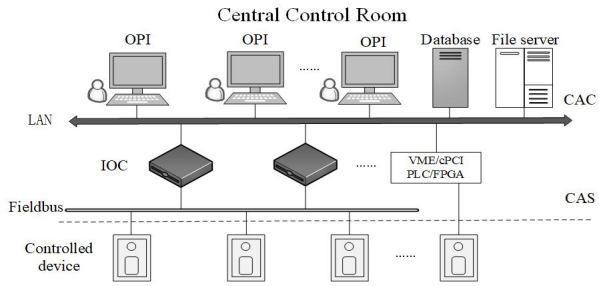

Figure 3: Hardware architecture of control system.

REAL-TIME REMOTE INTERFACE

Remote Real-time Interface Deployment

In order to realize the function of remotely accessing the interface of the central control room, the enterprise WeChat is adopted. At the same time, in order to ensure data security, the server and enterprise WeChat account are deployed to the intranet and the extranet respectively. The deployment process is shown in Figure 4. Intranet: The opi interface drawn by CS-Studio is deployed in the Tomcat application server on the intranet[4]. Write the configuration file config.php to connect to the enterprise WeChat, which mainly includes CorpId, TxlSecret and AppsConfig. Extranet: Deploy enterprise WeChat account, and log in to the enterprise WeChat account through the administrator's authorization for staff in the office. Enterprise WeChat obtains real-time data from the central control room through the firewall. The staff of the institute can view the real-time remote interface anytime and anywhere to understand the experimental situation.

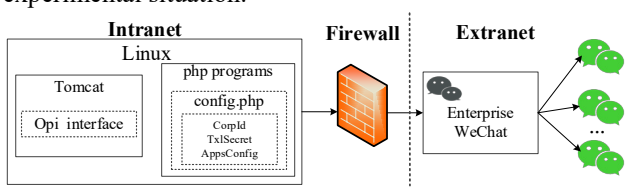

Figure 4: Remote real-time interface deployment.

Real-time Mobile Terminal Alarm Interface

Figure 5 shows the remote real-time data interface. The value of the alarm can be marked in red by setting the control rules of the OPI interface. The user group can view the relevant on-site real-time interface through the enterprise WeChat, which is helpful for the staff to understand the on-site operation in time and deal with the failure in time.

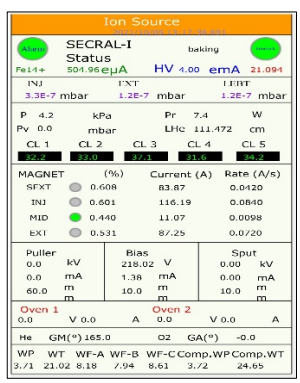

Figure 5: Real-time remote interface.

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-THPV006

CENTRAL CONTROL ROOM ALARM INTERFACE DISPLAY

Alarm

The alarm interface of Phoebus is shown in Figure 6. The alarm interface mainly includes four parts: Alarm Tree , Alarm Area Panel and Alarm Table[5].

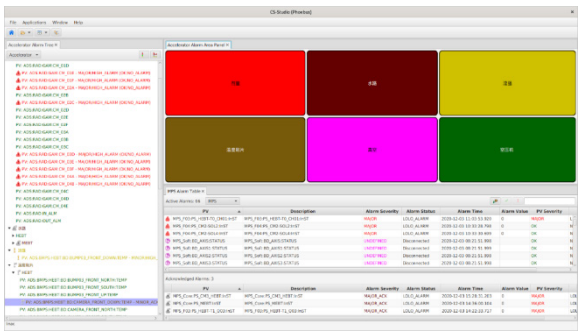

Figure 6: The alarm interface of Phoebus.

Alarm Tree

The alarm tree is mainly used to configure the alarm system, add PV and define the detailed information of the alarm. The tree-like hierarchical structure is used to record the PV that needs to be monitored. The tree structure mainly includes three parts: root, branch and leaf. Root represents the top-level domain name, branch represents the subsystem under the top-level domain name, and leaf represents the monitored process variable[6]. As shown in Figure 7, there are 6 top-level areas, namely dosage, waterway, flow intensity, temperature patch, vacuum and air compressor. Subsystems can be divided under each area, such as HEBT and MEBT in the waterway, and PV can be classified under the sub-system. The top-level domain name shows the most serious alarm status among all PVs under the domain name. The hierarchical view allows the operator to quickly find the affected subsystem or area.

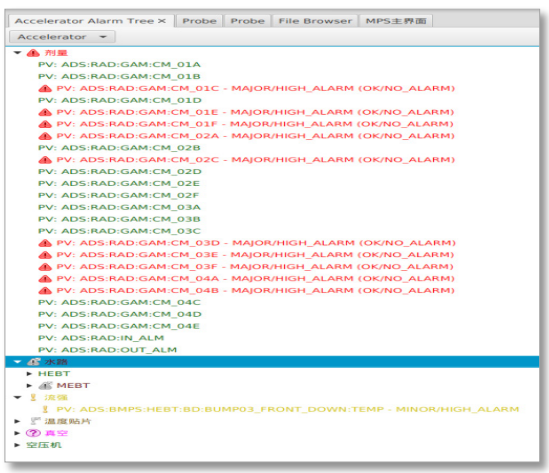

Figure 7: The interface of the Alarm Tree.

Alarm Area Panel

The Alarm Area Panel displays the most serious alarm level, and the panel color is colored according to the highest alarm in the area. Different colors represent different levels of alarms, and each alarm domain corresponds to the top-level alarm of the Alarm Tree. As shown in Figure 8, the red is the MAJOR alarm state, the red is the MAJOR alarm state and the operator confirms that the alarm is received, the yellow is the MINOR alarm state, the yellow is the MINOR alarm state and the operator confirms that the alarm is received, the purple is the INVALID disconnection state, and the green is the normal state.

Figure 8: The interface of the Alarm Area Panel.

Alarm Table

As shown in Figure 9, the Alarm Table is empty under normal conditions. When an alarm occurs, the alarm PV will be listed in the table, and the user can sort by PV name, description, alarm time and alarm level.

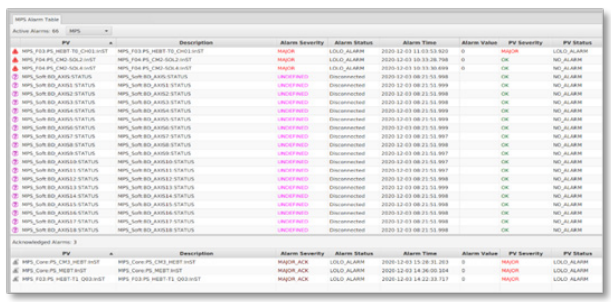

Figure 9: The interface of the Alarm Table.

Alarm Log Table

The Alarm Log Table records historical alarms and is updated every 10 s, which can be used to query historical alarm information, as shown in Figure 10.

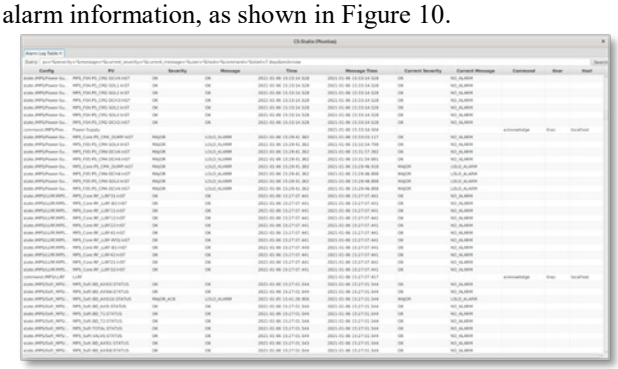

Figure 10: The interface of the Alarm Log Table.

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-THPV006

Annunciator

When the PV alarm information that exceeds the threshold in the topic "AcceleratorState" is sent to the Alarm Server, the topic "AcceleratorTalk" will send the alarm to the Annunciator.The Annunciator interface will display the alarm time, alarm security level and the description of the alarm PV, as shown in Figure 11. By default, the central control room receives the alarm voice once/15 s, and the voice is the content of the description, and it broadcasts until the alarm PV returns to the normal threshold.

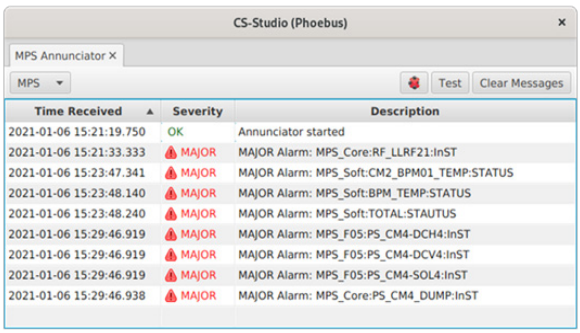

Figure 11: The interface of the Annunciator.

COMPARISON OF THROUGHPUT PERFORMANCE BETWEEN Kafka AND ActiveMQ

Kafka and ActiveMQ are streaming media platforms, both of which can process streaming data. The old version of the alarm system uses ActiveMQ. The alarm system of this report uses Kafka as an intermediate message plug-in. As shown in Figure 12 and Figure 13, JMeter is used to test the throughput of Kafka and ActiveMQ, both of which use a single thread to obtain 1000 data for multiple tests. The experiment shows that Kafka throughput is 6188119/minute, ActiveMQ throughput is 739153/minute. Kafka's performance far exceeds ActiveMQ. Kafka has high throughput and is more efficient for processing real-time data streams from accelerators. Moreover, Kafka can be used not only as a message intermediate plug-in, but also as a database cache data. There is no need to install the database separately, which simplifies the installation process. Therefore, it is more effective to use Kafka as the streaming platform of the alarm system.

Figure 12: The throughput of Kafka.

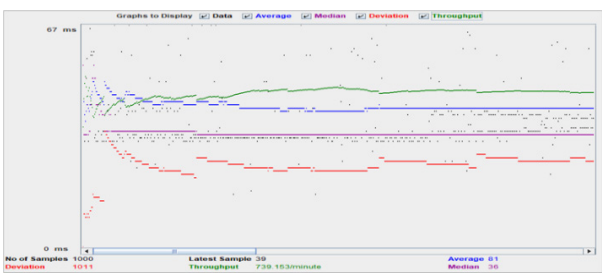

Figure 13: The throughput of ActiveMQ.

CONCLUSION

CAFe alarm system was built based on Kafka streaming media platform. Phoebus adds the PV to be monitored and displays the running status. Phoebus provides an interface for stable monitoring of PV status and real-time alarm function. Using Kafka can effectively increase the data throughput rate and ensure the high availability of the server. And use the remote real-time interface to improve the cross-regional alarm function.The entire control system has been deployed and put into operation in the central control room. The interface is simple and easy to operate, works well and is easy to maintain, which saves troubleshooting time and increases the maintainability of CAFe .

REFERENCES

[1] Xu Lunming, Wen Pengquan, Wang Jigang, Xuan Ke, Li Chuan, and Liu Gongfa, "The alarm system of Hefei light source based on BEAST", *Nuclear Electronics and Detection Technology*, vol. 35, 2015, pp. 417-421.

[2] C. Jianjun, Y. Youjin, Z. Wei, *et al.*, "Upgrade of Control System for 320 kV Heavy Ion Multidisciplinary Research Facility", *Atomic Energy Science and Technology*, cvl. 53, 2019, pp. 1612-1616. doi:10.7538/yzk.2019.youxian.0159

[3] Hao Peng-Hai, Xu Cheng-Long, and Liu Yi-Tian, "Monitoring and Alarm System for Power Grid Cloud Platform Based on Kafka and Kubernetes", *Computer Systems & Applications*, vol. 29(8), pp. 121-126, 2020. doi:10.15888/j.cnki.csa.007611

[4] Y. L. Zhang, K. J. Xue, F. Q. Guo, *et al.*, "Public Account Based Alarm Information Pub/Sub Pattern System for CSNS Accelerator Facility", *Nuclear Electronics & Detection Technology*, 2019.

[5] K.-U. Kasemir, X. H. Chen, and E. Danilova, "The Best Ever Alarm System Toolkit", in *Proc. 12th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'09)*, Kobe, Japan, Oct. 2009, paper TUA001, pp. 46-48.

[6] K. S. White and K.-U. Kasemir, "Alarms Philosophy", presented at 12th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'09), Kobe, Japan, Oct. 2009, paper TUP022, unpublished.