

UPGRADE OF THE HISTORICAL DATA QUERY AND ANALYSIS SYSTEM FOR HLS-II

Z. Xie, Y. Song, Z. Shao, C. Li, K. Xuan, G. Wang, G. Liu*

National Synchrotron Radiation Laboratory, University of Science and Technology of China, Hefei, Anhui, China

Abstract

The current historical data query and analysis system for the Hefei Light Source II (HLS-II) was developed with Apache Struts2. However, Apache Struts2 need to be fixed from time to time to avoid being attacked. Therefore, a new system based on Spring Boot and Vue.js is developed. Meanwhile, the performance of the system is optimized, and the radiation monitor module is added. This paper will detail the system architecture and software implementation.

INTRODUCTION

The Hefei Light Source II (HLS-II) is a vacuum ultraviolet (VUV) synchrotron light source, it is comprised of an 800 MeV linac and an 800 MeV storage ring and a transport line connecting the linac and storage ring. The control system of the HLS-II is a distributed system based on EPICS. The archiver system was set up with Relational Database (RDB) Channel Archiver and Oracle database in 2014. Accordingly the historical data query and analysis system (HDQAS) was developed with Apache Struts2 [1].

However, due to the vulnerability Apache Struts2 need to be fixed from time to time to avoid being attacked. Besides, there are some aspects to be optimized in the current HDQAS, e.g. query speed. Therefore, a new HDQAS is developed with Spring Boot and Vue.js [2,3]. It optimizes the speed of query, data download and user interface. Besides, the radiation monitor module is integrated into the new HDQAS.

SYSTEM ARCHITECTURE

The new HDQAS is developed under the architecture of separated front-end and back-end. As shown in Fig. 1, browsers request static resource like HTML, CSS, JavaScript files from the front-end server, and get real-time data, historical data and analysis results from the back-end server. Nginx server is used as a proxy server to distribute users' requests and restrict IP address who can access the system.

In the back-end server, there are 4 modules: real-time status, radiation monitor, historical data query and data analysis. The data sources consist of the Oracle database and IOCs/CA Server. The Oracle database is the data source for the historical data query and the data analysis modules. The historical data of the HLS-II is saved in the Oracle database via RDB Channel Archiver, and retrieved from the Oracle database with Java Database Connectivity (JDBC) by the back-end server. Browsers could get historical data and analysis results

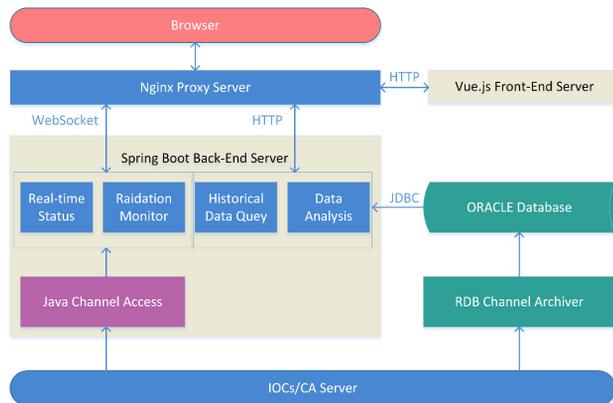


Figure 1: Architecture of the new historical data query and analysis system.

from the back-end server through HTTP connections, while they get real-time data through Websocket connections with the back-end server [4]. Java Channel Access (JCA), a pure Java Channel Access client implementation [5], is used to get real-time data from the IOCs/CA server for the real-time status and the radiation monitor modules.

SOFTWARE IMPLEMENTATION

Axios is a JavaScript HTTP communication module, it is used to send HTTP requests from the front-end server to the back-end server. Highcharts, as an open-source JavaScript chart library, is used to draw line chart and pie chart on the web page.

According to the Web-GUI, the new HDQAS mainly includes 4 modules: home, historical data, data analysis and radiation monitor, as shown in Fig. 2.

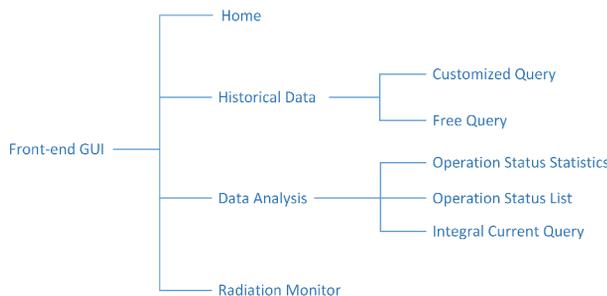


Figure 2: Components of the Web-GUI.

The home module presents the real-time data of operation status and the line chart of the latest 24-hours data of beam current and beam life.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2019). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

* gffiu@ustc.edu.cn

The historical data module is comprised of 2 submodules: customized query and free query. In the customized query submodule, users can select records to query according to the configured tree. In the free query module, users can query historical data in self-define groups.

The data analysis module consists of 3 submodules: operation status statistics, operation status list and integral current query. The operation status statistics submodule presents the statistics of operation status of the HLS-II in the form of pie chart. The operation status list submodule lists the operation status of the HLS-II in chronological order. The integral current query submodule shows the daily integral current in the line chart.

The radiation monitor module presents the real-time radiation dose data and the positions of measurement.

In the new HDQAS, many efforts are made to optimize performance, e.g. the speed of query and calculation is improved significantly in the customized query submodule. Besides, the new module, radiation monitor, is added to it. These two modules will be described in detail in the following sections.

Customized Query

The customized query page is shown in Fig. 3. The configured tree is on the left, it's generated by parsing xml configuration files. Users can find specific records easily in it. The line chart of data of the selected records is presented in the center. Users can zoom out and zoom in the line chart or export it as JPEG, PNG, SVG or PDF file. The records of the selected group are listed below the chart. The time range selector is used to set the time range for querying data. There are 4 function modules: draw line chart, view statistics, view data and download data, and the corresponding buttons are at the bottom of the page. There are also two switches: logarithmic and beam current. The "Logarithmic" switch is used to change the type of the vertical axis of the chart from linear to logarithmic scale. The "Beam Current" switch is used to add beam current data in the chart for analysis.

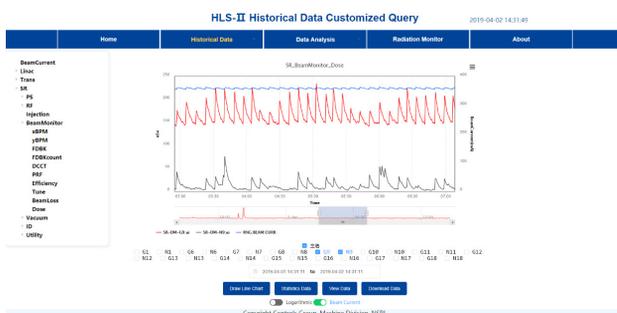


Figure 3: The customized query page.

In the draw line chart function, browsers request historical data from the back-end server and render it to the line chart. As shown in the center of Figure 3, the red line is for the vacuum pressure and the blue line is for the beam current. When the amount of data is very large, it's hard to render it to the chart, and the data transmission spends much time. In

order to solve these problems, the back-end server calculates the amount of data to query at first. When the amount of data is less than 4000, the back-end server queries all data from the Oracle database. When it's larger than 4000, the back-end server queries data by the appropriate interval in chronological order to ensure the amount is between 2000 and 4000.

In the statistics data function, the statistics like mean, deviation, root mean square, max, min of selected records, are shown in a table. In the current system, the calculation is done by Java program. But when the amount of data is large, memory leaks may occur and calculation speed is low. In the new HDQAS, we use SQL statements to calculate the statistics of the raw data in the Oracle database to avoid memory leaks and speed up calculations.

In the view data function, raw data is listed in pagination. To avoid memory leaks, the back-end server uses "ROWNUM" in SQL statements to retrieve single-page historical data from the Oracle database instead of all historical data. It also improves query speed greatly.

The download data function is the most troublesome in the current system. All data from the Oracle database is saved in the memory at first, then the back-end server writes it into a file and sends the file to the browser. When the amount of data is large, the download time is very long and the download thread might suspend due to memory leaks, but no information is given to users. In the new HDQAS, the download process is optimized to solve these problems. The data is divided into many parts in chronological order, then the back-end server retrieves data from the Oracle database one part after another and writes it to the file. Fig. 4 is the flow chart of the download process in the new HDQAS. The back-end server calculates the amount of total data in the Oracle database by using SQL statements, then estimates the size of the file and the time required accordingly. As shown in Fig. 5(a), the dialog is popped up to let user decide whether to download. After the download process is started, browsers show the progress of the download progress as shown in Fig. 5(b).

Radiation Monitor

Users who work in the experiment hall need to conveniently know the radiation situation. Therefore, the radiation dose monitor module is added to the new HDQAS.

As shown in Fig. 6, the map of the experiment hall is on the right, and the blue points represent positions of the radiation dose measurement. The radiation dose data of each measurement point is shown in the left table. The data of operation status is shown on top of the table. All this data is from the IOCs/CA server and refreshes every second.

There are also links for the latest 24-hours dose history in the table of Fig. 6. The latest 24-hours gamma and neutron dose of DM10 is shown in Fig. 7. The red line is for the gamma dose, and the black is for the neutron. The "Beam Current" switch is used to add beam current data in the chart. The default max values of vertical axes of radiation dose are

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2019). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI

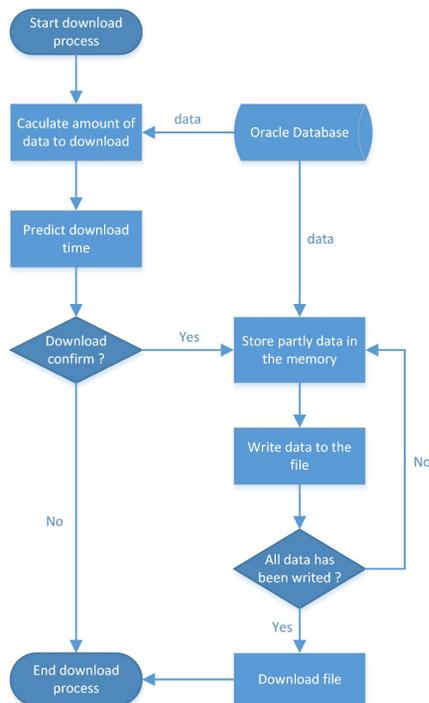


Figure 4: The flow chart of download process.



Figure 5: The tips and progress bar for download.

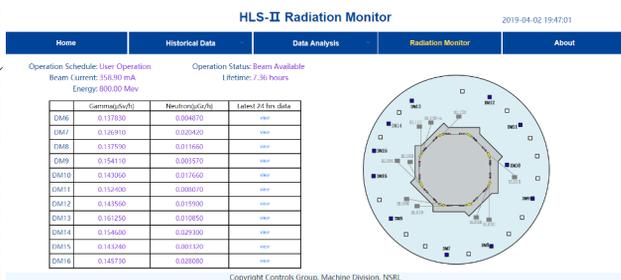


Figure 6: The radiation monitor page.

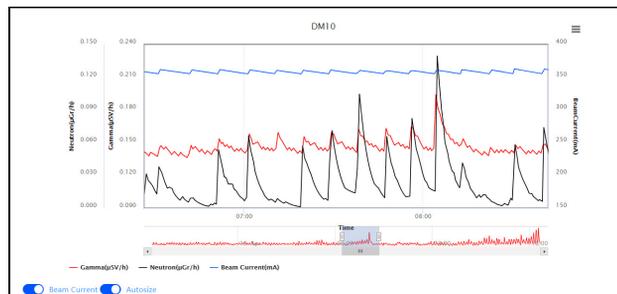


Figure 7: The 24-hours dose history presentation page.

fixed, and users can use the “AutoSize” switch to adjust the range of these axes to the data.

CONCLUSION

The new HDQAS is developed based on Spring Boot and Vue.js. It fixes memory leaks when querying or downloading massive data, and optimizes the performance and user interfaces. The radiation monitor module is added. The system is under commissioning from March 2019. It works well, and users’ feedback is good. The new system will be put into operation in the near future.

REFERENCES

- [1] P. Wen. The Historical Data Query and Analysis System for the HLS-II[D]. University of Science and Technology of China, 2016: 15-16.
- [2] Spring Boot, <https://spring.io/projects/spring-boot>
- [3] Vue.js, <https://vuejs.org/index.html>
- [4] V. Pimentel, B.G. Nickerson, "Communicating and displaying real-time data with websocket[J]". *IEEE Internet Computing*, vol. 16, no. 4, pp. 45-53, 2012.
- [5] Java Channel Access <https://github.com/channelaccess/ca>