Research on remote data access and visualization for long pulse experiment based on web technologies

Mengqi Fan1, Yu Gu1, Tengfei Cao1, Xiaodan Zhang1,*

1Department of Computer Technology and Applications, Qinghai University, Xining, 810016, China

*Corresponding author e-mail: xdzhang@qhu.edu.cn

Abstract. Model Drive System plus (MDSplus) is an efficient and convenient data storage and management software designed by MIT specifically for pulse experiments. After years of development, MDSplus has become a data management system that integrates data acquisition, storage and processing of complex scientific data. However, with the development of the experimental pulse, long pulse experiment has now become the main mode of pulse-type experiment. Compared with the short pulse experiment, the data acquisition under the long pulse experiment speed is faster, the experiment time is longer, and the amount of data is larger. In view of the fact that most of the existing visualization systems are based on client/server (C/S) architecture, which can only display local visualization. Therefore, In order to enrich the way for experimenters to obtain data, this paper designs a visual platform based on browser/server (B/S) that using the WAMP (Windows+apache+MySQL+php) architecture to build the server. The bootstrap framework is used to build the front-end. Before the visualization data transmission, the original data is processed by segmentation technology to reduce server load and improve transmission utilization so that experimenters can query and obtain the required data remotely through the platform.

1. Introduction

Based on the operating times of the system, the pulse-type experiment can be divided into long pulse mode and short pulse mode. The pulse-type experiment is to complete the steady-state operation of the long pulse experiment under the constraint of the magnetic field, and realize the nuclear fusion to solve the energy crisis problem. At present, the typical pulse experimental devices include EAST (Experimental and Advanced Superconducting Tokamak) of the Institute of Plasma Physics of the Chinese Academy of Sciences, JT60 in Japan, KSTAR[1] in South Korea and ITER of the International Thermonuclear Fusion Reactor, etc. To better meet the needs of long-pulse steady-state experiments, these experimental research institutions have applied MDSplus to their experimental researches. The existing visualization system includes the BOY (Best Operator interface, Yet) operation interface developed by CCS, which is designed by the ITER nuclear fusion experimental device based on the Linux-based ITER CODAC core system[2]; The web scope visualization software, which is developed by EAST device based on jscope[3] and Java-applet technology and the data acquisition; human-machine interaction display system J-TEXT[4] was developed based on EPICS (Experimental Physics and Industrial Control System). By comparing these visualization systems, most of these visualization systems are oriented towards long-pulse data.

It is especially important for researchers to be able to view and visually analyze any segment of experimental data after the experiment completed. However, as the experimental data increases, it...
becomes more difficult for terminals to obtain data at any time. In order to better display the data, it is compressed and segmented in advance. The visualization platform basing on the B/S architecture which is combined with charts.js component and combining MySQL relational database and MDSplus helps experimenters access the platform through a web browser without installing a client.

In the preliminary data processing and acquisition, in order to reduce the data transmission load and save the required space for data storage, the collected raw data are generally processed for data compression, using the open source LZO algorithm[6] to compress the data. After processing, the data from long pulse experiments is compressed into a file in ‘*.lzo’ format and then the file is segmented according to the experimenter’s requirements. When compressing data, in order to facilitate the subsequent data segmentation processing, ‘*.lzo’ format data needs to include two parts: the header file and the original data which the header file is used to describe the raw data information. When segmenting the data, based on the previously compressed complete data, the experimenters select the required period to segment the data[5]. Since the focus of this design is on remote visualization of data, the system architecture and experimental details of the pre-data segmentation processing are not described in detail here. The architecture of data processing and acquisition is shown in Fig 1.

![Fig 1 Architecture of data processing and acquisition systems](image)

This design will introduce the implementation of the platform from the following aspects of remote data visualization platform construction and data transmission. The reliability of the visualization platform is verified by using the given test pulse data as the test object. The rest of the paper is as follows: first introduce the architecture of the whole visualization platform, then introduce the detailed design of the visualization platform, and finally summarize and improve the platform.

2. Architecture design
The visualization platform mainly provides remote data access and visualization services for experiments, so the platform plays a very important role in the data analysis of experiments. Theoretically, the overall process of the platform can be described as follows: the experimenters fill in the relevant information of the data to be displayed in the visualization platform, and then send the access request to the web server, the web server sends a request to the database server for accessing the database after unifying the request. When the database receives the request from the web server, it
processes the request and sends the returned result to the web server. The web server transforms the data received from the database into HTML and presents it on the browser, and finally completes the whole visualization display.

As mentioned above, the visualization platform mainly includes two modules: data transmission and data visualization. The former is mainly used to obtain and store data into the database remotely, while the latter is mainly used for the visualization of data. As shown in the figure, the whole visualization platform is divided into two parts: web server and data server. The web server includes two parts: the web side and the MySQL relational database; The data server is MDSPlus database. The web server sends the request of calling data to the data server through the MDSip protocol. The data server packages the data in MDSPlus into packets and transmits them to the web server through the TCP/IP protocol to complete the response. Using two databases, on the one hand, the logical structure of data in MDSPlus is a tree structure, it is not convenient for experimenters to compare the data, but storing the data in MySQL can analyze the data more intuitively. The other is to better connect with the visualization platform. The structure of the whole platform is shown in Fig 2.

### 3. Platform detailed design

The functions of the visualization platform introduced in this paper can be summarized into the following two parts: data transmission and data visualization. Data transmission is to transfer of data from MDSplus to MySQL after pre-segmentation processing, and data visualization refers to the visual display of the data in MySQL.

#### 3.1 Data transmission

Data transmission mainly includes the following two parts: data acquisition based on the MDSip protocol and data transmission based on the TCP/IP protocol. Fig 3 describes the data transmission process of this platform. Firstly, the data is stored in MDSPlus. The experimenter sends the request to get the data remotely, and transmits the request to MDSPlus through the MDSip protocol. MDSPlus transmits the data to MySQL through the TCP/IP protocol.

The MDSplus database uses a tree structure to save the collected data information in three types of files: *.data, *.character, and *.tree. The tree structure can not only unify the collected data format, but also clearly expresses the logical relationship between nodes. After the tree is built and the data is stored, the experimenters who want to access the data remotely need to install the MDSip [7]protocol on the data side, and MDSplus obtains data through the MDSip protocol. MDSip is a protocol based the TCP/IP...
protocol which can be used for remote expression evaluation and is mainly used for reading and writing data to MDSplus. It can connect to the remote server, send the data buffer to the remote server, and receive the data buffer from the remote server. When connecting to a database remotely, MdsConnect is used to specify the connection string IP and port number for connecting to the remote server. MdsValue is used to send expressions and optional parameters and get feedback. MdsClose is used to close the connection with the remote server, so as to achieve the purpose of remote connection.

Fig 3 Flowchart of visualization platform data transmission

MDSip has an authentication mechanism. When a client connects to a remote server, MDSip sends the user’s IP address and port number to the server, and the administrator of the server configures the MDSip server to accept or reject the connection according to the information of the client and the IP address of the connected client system.

MDSplus provides access interfaces such as c, java, python, etc. When obtaining data, the python language is used to obtain the long pulse experimental data in the MDSplus database. Due to the characteristics of MySQL database which supports multi-user, multi-threaded, concurrent access, stability and efficiency, this relational database is selected to store the data. By getting the corresponding meta information in the file such as ExpNum, DatabaseName, SignalName, etc., and transferring the obtained data to MySQL database via the TCP / IP protocol[8], it constitutes the data source for the front-end visualization of the platform.

The TCP/IP protocol that data transmission relies on is divided into three parts: the TCP header, the data header and the experimental data. Fig 4 shows the composition of the TCP/IP protocol: the TCP header mainly consists of four parts: source address, destination address, data size, and CRC checksum; the data header is used to store the information of the collected data, including ExpNum, DatabaseName, SignalName, etc.; the experimental data is the data compressed by the open source lzo algorithm.

Fig 4 Components of the TCP/IP protocol

When transmitting data by TCP/IP protocol, the source and destination addresses in the TCP header are first parsed to obtain the addresses of the sender and receiver, and then connected by 3-Way
Handshake [9]. In order to verify that the TCP message segment, that is, the TCP header and data are not lost in transmission between the sender and receiver, it is necessary to perform a CRC checksum on the TCP message segment, which is calculated by the sender and then verified by the receiver. After calculating the sum of the header and data parts of the TCP message segment, the sender gets the checksum then re-loads it into the message for transmission. When receiving the message, the receiver calculates the checksum by the same algorithm. If it matches, it accepts the message; if not, it proves that there is frame loss, discards the erroneous message and triggers the retransmission mechanism to notify the sender to resend it.

The received data is placed in MySQL’s mdsplus_data information table. The table structure is shown in Tab 1, which mainly includes: ServerIP, EXPNum, DatabaseName, SignalName.

| Parameter       | Type         | Description                       |
|-----------------|--------------|-----------------------------------|
| id              | int(10)      | Record the number of data         |
| ExpNum          | varchar(128) | The number of experiments         |
| DatabaseName    | varchar(128) | The name of the database          |
| SignalName      | varchar(128) | Acquisition channel name          |
| DateSegment     | text         | Collected data                    |
| ServerIP        | varchar(128) | Server IP address                 |

### 3.2 Data visualization

#### 3.2.1 Login module

There are two kinds of user groups for this platform: administrator and ordinary user, and the permission level of each type is different. The information about users is stored in MySQL’s user information table, the table structure is shown in Tab 2, mainly including: id, username, password, email, occupation, work, status, user_type.

| Parameter       | Type         | Description                       |
|-----------------|--------------|-----------------------------------|
| id              | int(10)      | Record the number of data         |
| username        | varchar(32)  | Ordinary user’s name              |
| password        | varchar(128) | Ordinary user’s password          |
| email           | varchar(128) | Ordinary user’s email             |
| occupation      | varchar(128) | Ordinary user’s occupation        |
| work unit       | varchar(128) | Ordinary user’s work unit         |
| status          | int(10)      | Ordinary user’s audit status      |
| user_type       | int(10)      | Administrator or ordinary user    |

In the login page, only authorized and legitimate users can enter the platform, which can effectively prevent users from illegal login. A server authentication mechanism is used in the login screen, that is, an authentication file is added on the server side, which is controlled by the session variable in PHP. This is a variable for global service, which is stored in the server and cannot be accessed by the client. By setting the value of the session variable, it can determine whether the user has a legitimate identity. When the browser loads the login interface, the platform first gets the authentication file on the server and detects the values of username and user_type through $_SESSION[], if the username is ‘admin’ and the value of user_type is ‘1’, it proves that the user is an administrator, after entering the correct password, it can jump to the audit interface to audit the registration records of ordinary users; if it matches the username stored in MySQL, and the login password is correct, then the ordinary users can login successfully. The flow chart of the login interface is shown in Fig 5.
3.2.2 Data processing and visualization module

The remote connection between MySQL and the web side is achieved by “$conn = new mysqli($hn, $un, $pw, $db)”, When getting the data, the fopen is used to open the txt file, and the fgets is used to get the data and store it in MySQL. “$_SESSION['form_data'] = array()” is used to create an array to store the values currently entered by the experimenter. After filling the parameter ‘ServerIP’, the experimenters click the search button to submit to the array. The Back-End of the platform matches all the EXPNum, DatabaseName and SignalName stored under this ServerIP in MySQL by trim($_POST['']). If it matches the value stored in MySQL, the corresponding parameters under this ServerIP will be automatically displayed for the experimenter to select. If it fills incorrectly then no response will be made.

Next, trim($_POST['']) is used to get the following parameters selected by the experimenter: EXPNum, DatabaseName, SignalName, StartTime, EndTime and part. The maximum time interval that the experimenter can select for visual presentation is limited to 1000 microseconds. If the value entered does not match the value stored in MySQL or the time exceeds the limit, it indicates a wrong input and prompts re-enter it. If the input is correct, it will be stored into array() for the next update and clear functions to retrieve the currently submitted value.

When updating the data, the data in the array is printed on the update page, when the changes are completed, the experimenters click the update button to update the array’s values, and pass the updated array to the visualization interface for display. When emptying the data, that is, empty the array, and the visualization interface is emptied immediately.

When visualizing the data, due to the visualization is at most in four different areas, the values in the array are divided into four different arrays by the part. The new arrays define three variables: values, labels, and titles. The labels are used to store each time point of the period; The values are used to store the values corresponding to each time point and the titles are used to store the EXPNum, DatabaseName, and SignalName of the values. In order to prevent data confusion during partitioning, the grouped values are compared with MySQL again. Chart.js is used to visually display the values within each part, and a new function is defined to take up the partitioned array. Charts.js calls the function so that the horizontal coordinates are the values of ‘labels’ and the vertical coordinates are the values of ‘value’, and the information of each point is displayed as the value of ‘title’.

Fig 5 Flowchart of login interface
3.2.3 Supporting technology

In order to facilitate experimenters to visualize the collected data, considering security and stability, this platform mainly uses B/S architecture[10]under the environment of Windows+apache+php, with bootstrap [11] framework and chart.js to make data visualization more convenient and efficient. At present most browsers support the bootstrap framework, in which the responsive CSS design can be adapted to different terminals, providing a unified solution for developers to create interfaces. Chart.js [12] is relatively lightweight and does not rely on other libraries. It is an open source visualization library based on JavaScript, which is also compatible with most browsers.

We choose the B/S architecture because it relies on the TCP/IP transport protocol to connect the web network and the LAN, solving the problem of heterogeneous systems. In the B/S architecture, a "thin client" is used, which improves the openness of the system and eliminates the limitation of the number of users that can access it. Apache server [13] is chosen as the "bridge" to the web server, and only the corresponding modules are loaded. The server calls the php program, interprets the language directly, and passes the interpreted language to the client. The client reads the file and automatically interprets it, converting it to html format and sending it to the web server, then the web server sends the interpreted file back to the client and finally pushes it forward to the web page.

4. Visual platform testing

In order to test whether the platform can be used properly, two different sets of data are selected to test the platform in user login, add update delete data, etc. The two cases are:

I. The same EXPNum, the same DatabaseName, the same SignalName in different periods of time;
II. The different EXPNum, different DatabaseName, different SignalName in the same period of time.

In the first case, we selected the values from 0 to 200 microseconds and 300 to 600 microseconds in the same EXPNum3072, the same DatabaseNameDAQ05, and the same SignalNameCH00 for comparison, and the comparison results are shown in Fig 6.

In the second case, we selected different EXPNum3072 and 3075, different DatabaseNameDAQ02 and DAQ05, and different SignalNameCH00 and CH01 in the same time period from 0 to 200 microseconds for comparison, and the comparison results are shown in Fig 7.
To test whether the platform is able to visualize when the values entered by the experimenter are wrong, we entered two sets of wrong values. After subtracting the entered EndTime and StartTime values more than 1000, we click add button will clear the present value and it will prompt "The maximum time interval is 1000 microseconds". When the experimenter selects a combination of EXPNum, DatabaseName, SignalName that does not match the database, the current selection is cleared and prompted with "Input error, please re-enter". The results of this fault tolerance test are shown in Fig 8 and Fig 9.

![Data](image)

**Fig 7** Test results of different EXPNum different, DatabaseName different SignalName and same time period

**Fig 8** Input time overrun
5. Conclusion

The visualization platform is based on the long pulse experiments, and the research is conducted in long-term pulse data storing, analyzing and visualizing. Comparing the previously existing data visualization systems, this article summarizes its features and data processing modes, and completes the framework and web platform design. The platform is based on the high-efficiency storage capability of MDSplus and MySQL, as well as the B/S architecture to realize the online display of client web pages, providing a platform for experimenters to display the data selected randomly. As the experiment proceeds, the storage of long pulse data becomes more and more difficult, the collected data also increases, and the structure of the data becomes more and more complex, so the visualization platform should also be developed more deeply. However, with the development of computer technology, a new generation of visualization terminals will eventually be developed to support the fusion long pulse experiments.

Acknowledgment

The authors gratefully acknowledge the supports of the Scientific research cooperation project (IPP-ZC-19071806), the National Natural Science Function of Qinghai Province (2020-ZJ-943Q) and the National Natural Science Foundation of China (61572370).

Reference

[1] M Kwon, I.S Choi, J.W Choi, et al. The control system of KSTAR. 2004, 71(1):17-21.
[2] Wei Zheng, Yuxing Wang, Ming Zhang, Zhou Yang, Yuan Pan. Designing CODAC system for tokamaks using web technology[J]. Fusion Engineering and Design, 2019, 146(Pt B).
[3] G. Manduchi, T. Fredian, J. Stillerman. A new web-based tool for data visualization in MDSplus[J]. Fusion Engineering and Design, 2014, 89(5).
[4] Wei Zheng, Kuanhong Wan, Zhi Chen, Feiran Hu, Qiang Liu. J-TEXT WebScope: An efficient data access and visualization system for long pulse fusion experiment[J]. Fusion Engineering and Design, 2016, 112.
[5] X. D. Zhang, Y. Zhang, S. R. WEI, and W. D. XUE, “Data Segmentation Processing Software for Long Pulse Data[C]. Advanced Science and Industry Research Center. Proceedings of 2018 3rd International Conference on Automation, Mechanical and Electrical Engineering (AMEE 2018). Advanced Science and Industry Research Center: Science and Engineering Research Center, 2018:611-615.
[6] LZO [Online] http://www.oberhumer.com/opensource/lzo/lzofaq.php
[7] MDSplus Home Page. Available from: http://www.mdsplus.org/
[8] TCP / IP protocol [online]https://blog.csdn.net/weixin_44198965/article/details/90083126
[9] Fu-Hau Hsu, Yan-Ling Hwang, Cheng-Yu Tsai, Wei-Tai Cai, Chia-Hao Lee, Kai Wei Chang. TRAP: A Three-Way Handshake Server for TCP Connection Establishment[J]. Multidisciplinary Digital Publishing Institute, 2016, 6(11).
[10] B/S architecture[Online] https://baike.baidu.com/item/B%2FS%E7%BB%93%E6%9E%84/4868588?fromtitle=b%2Fs&fromid=219020&fr=aladdin

[11] Long Zhang. Research and Application of Icon Fonts in Web Front-end Technology[A]. Wuhan Zhicheng Times Cultural Development Co., Ltd..Proceedings of the 3rd International Conference on Computer Engineering, Information Science,2019:5.

[12] Chart.js[Online] https://www.chartjs.org/

[13] Pushpa Singh,Narendra Singh. Analysis of Free and Open Source Software (FOSS) Product in Web Based Client-Server Architecture[J]. International Journal of Open Source Software and Processes (IJOSSP),2018,9(3).