QUEST Database for Tokamak Big Data

Makoto HASEGAWA, Aki HIGASHIJIMA, Ichiro NIIYA, Kazuaki HANADA, Hiroshi IDEI, Takeshi IDO, Ryuya IKEZOE, Takumi ONCHI, Kengo KURODA and Daisuke SAKURAI[1)]

Research Institute for Applied Mechanics, Kyushu University, 6-1 Kasuga-koen, Kasuga, Fukuoka 816-8580, Japan
[1)]*Pan-Omics Data-Driven Research Innovation Center, Kyushu University, 6-1 Kasuga-koen, Kasuga, Fukuoka 816-8580, Japan*

(Received 21 November 2022 / Accepted 19 April 2023)

A database has been developed to provide a highly efficient data analysis environment by registering various types of data obtained in the QUEST (Q-shu University Experiment with Steady-State Spherical Tokamak) experiments and providing the data to researchers through standard SQL query output. In conventional experiments, measurements are usually obtained using several instruments, and the results are often stored separately in proprietary formats. An analysis to collect distributed files and convert them into appropriate formats individually was necessary, which led to a decrease in analysis efficiency. In order to build a system to centrally store and provide data, we constructed new servers, each for registering data in a database and for extracting data from the database and visualizing the data. As a result, various real-time monitoring data can be registered in the database, and data retrieval for analysis can be performed easily with a unified user interface. The construction of this database has made it possible to easily retrieve real-time monitoring data and a wide variety of table data, enabling analysis from a new perspective, and improving analysis efficiency.

© *2023 The Japan Society of Plasma Science and Nuclear Fusion Research*

Keywords: database, big data, SQL, continuous measurement, user interface

DOI: 10.1585/pfr.18.1305048

The QUEST (Q-shu University Experiment with Steady-State Spherical Tokamak) project aims to maintain plasma for a long time and promotes research on long-term current drive of spherical tokamak plasmas and plasma-wall interactions with the QUEST device [1–3]. A major feature of the QUEST device is that the plasma-facing wall temperature can be actively varied in order to control the absorption and desorption of particles [4]. This has been found to be effective in controlling the particle balance, even in a high-temperature environment [5, 6]. The project is unique as researchers can directly design and optimize the types of experiments performed in the facility. This has opened up an opportunity to access a rich data set, to investigate past events, and even to design different scheduling strategies for experiments. In such projects, data, including logs, tend to be gathered from various equipment having their unique data access methods and formats. Over the years, each research project will have independently accumulated data relevant to the project. While it was considered practical at the time, currently, the substantial QUEST data accumulation has generated a need to discuss linkages in past experiments and events retrospectively. One example is analyzing patterns of gas leaks measured by mass spectrometers to ensure efficient and safe tokamak operation [7]. Many data loggers, including those attached to mass spectrometers, typically obtain data and output files at periodic intervals, leading to a large number of data files over time. This can lead to file loss, e.g., when moving data to a data server. Additionally, real-time monitoring data retrieval for analysis is inefficient due to the large number of files. Furthermore, measuring parameters such as ambient temperature and cooling water flow rate is important for the plant healthiness, and these may be used for estimating the cause of an abnormality or error in a measurement instrument. Therefore, it is important to obtain various types of data from different instruments and store them in a unified manner.

To this end, we report how we developed the *QUEST database*, which is an exhaustive collection of data from past experiments and events in QUEST since 2008. In some tokamaks, the experimental physics and industrial control system (EPICS) [8] is adopted as the control, data access and communication (CODAC) system [9–11]. Other devices also build data acquisition and provision systems according to the features of the device [12, 13]. Here, a relational database system is chosen to unify the data access process, which is beneficial due to an abundance of time-tested packages for reading records through SQL-based communication. Since most physicists are first-time database users, we built an easy-to-use interface so that novice users can access data easily using their data analysis scripts and visualize the data using a web interface, without the need to write a code. We believe that this report will benefit other tokamak projects by showcasing the development of the database.

author's e-mail: hasegawa@triam.kyushu-u.ac.jp

© 2023 The Japan Society of Plasma Science and Nuclear Fusion Research

A network configuration diagram for the database is shown in Fig. 1. There are four networks in the advanced fusion research center, where the QUEST device is located: a demilitarized zone (DMZ), an experiment network, a control network, and an office network. The main measurement instruments and control equipment belong to the experiment network or the control network. The database is network-attached storage (NAS) with MariaDB [14] installed, has a capacity of 100 terabytes, and is located in the DMZ. Data obtained from the control or experiment network are registered in the NAS via a firewall. Two Ubuntu-based servers are also installed to add functions to use this NAS. One server is a *registration server* placed in the control network to acquire data from the measurement instruments and register the same in the database. The other server is a *visualization server* placed in the DMZ to extract data from the database and display the data graphically.

In the MariaDB, multiple databases are created according to the measurement contents, and each database has multiple data tables. For example, in order to investigate the temperature distribution of the plasma-facing wall and the vacuum vessel in detail, more than 330 thermocouples were installed in the QUEST instrument [6]. More than 15 data loggers are used to measure the temperature, and a database for storing this information was constructed, which contains a data table for each logger. Access control to the database is performed by user authentication, and access privileges for each user are set for each database.

Several types of tables are registered in the database, and there are three typical tables. Type A table is for real-time monitoring data, and comprises time-series data that are not directly related to the plasma discharge sequence. The data, such as the plasma-facing wall and the vacuum vessel temperatures, the cooling water flow rate, and the experimental power consumption, are measured all day long. This corresponds to the table with the primary key as the datetime type variable supported in a relational database, usually not requiring high-speed (faster than millisecond-order) sampling. Most data of this table type are measured every few seconds and are automatically registered in real time. Type B table is for data that is directly dependent on the shot number and elapsed time since the start of the sequence, such as coil current, plasma current, and plasma heating power. This corresponds to a table with the primary keys are the shot number and the elapsed time. Since this type of data generally requires high speed (faster than millisecond-order) sampling, real-time registration is difficult, and data is collectively registered after the end of the discharge sequence. Type C table is for data determined for each discharge, such as the start time of the discharge sequence, the discharge duration time, and the discharge settings. The primary key for this table is the shot number, which is registered in the database immediately before the sequence for setting parameters, or after the end for obtaining the values.

Several methods for registering data in the database are supported in QUEST experiments. Typical methods are shown in Fig. 2. Measurement instruments, that can be programmed using a language such as Python, register data directly in the database by using the Python rich library (type III). For other measurement instruments, the registration server provides services to register the data in the database. One method involves a service in which the server constantly monitors the status of measurement instruments and the details of data creation; the service reads the data via protocols such as FTP and Samba, and registers the data in a database (type I). For this database type, almost no settings are required on the measurement instrument, but it is necessary to create a dedicated operation script for each instrument on the server. Another method

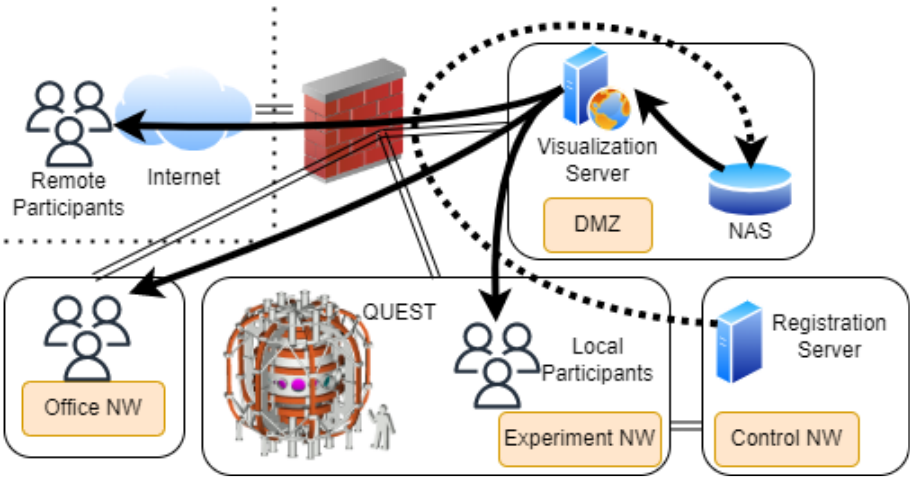

Fig. 1 Data flow to users. Data registration is performed from the control network via a firewall (dashed line), and data extraction is performed in a demilitarized zone (solid line).

involves a service in which the measurement instrument sends plain text to the server, which the server registers in the database (type II). Since this method only involves standard text communication, it can be implemented for a relatively large number of instruments.

Users in experiment and office networks can access the database directly via the DMZ. Although the database system allows users to use SQL commands or a scripting language such as Python to obtain QUEST data, we also provide data access through a JupyterHub [15] instance with a pre-configured Jupyter notebook installed on the visualization server (Fig. 3). The latter allows researchers and technicians to check fundamental data using a button-based graphical user interface without writing scripts (Fig. 4). This displays graphs based on input parameters such as the required data type, the shot number, and the elapsed time since the sequence started, and allows users to freely change the graph layout and select other data to be displayed by editing the Javascript object notation (JSON) based configuration file.

The advantages of registering data in the database and performing analysis are described using a Type A table for a mass spectrometer as an example. Conventionally, mass spectrometer data are output as data files of approximately 2,000 numbers. Additionally, the capacity of text-based

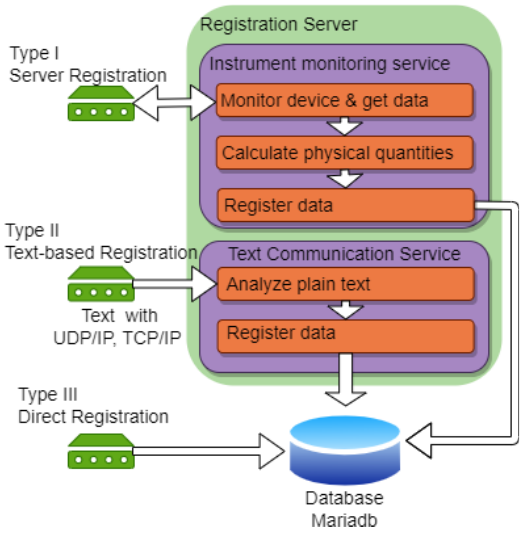

Fig. 2 Data registration types for the QUEST database.

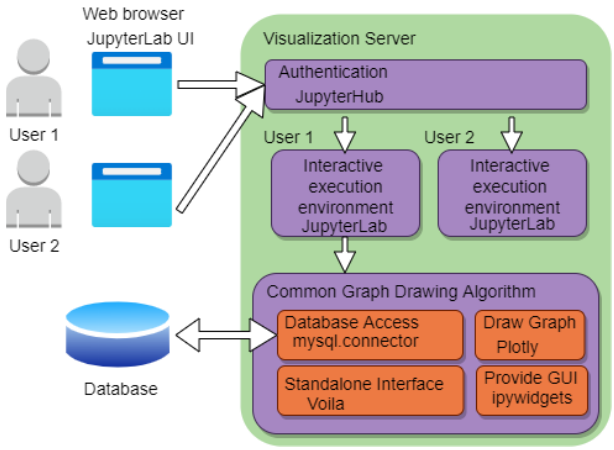

Fig. 3 Configuration of the visualization server for viewing data.

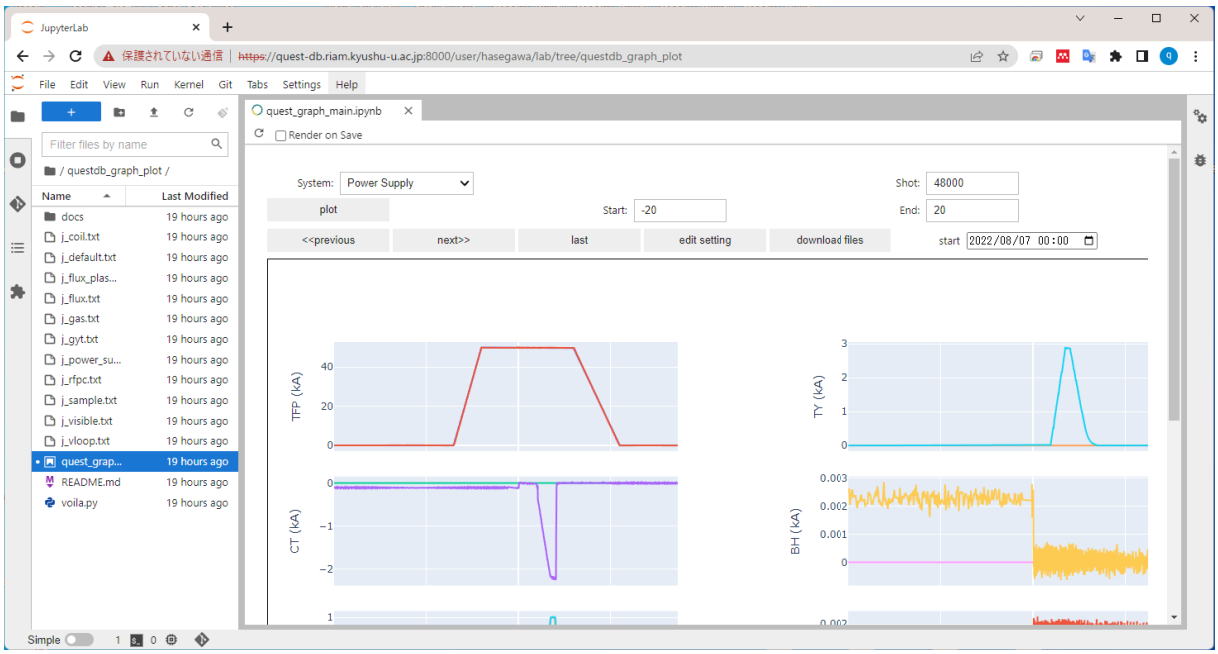

Fig. 4 Screenshot of graph drawing script running on Jupyter notebook.

data was approximately 60 gigabytes. When these data were registered in the database, the table consisted of one datetime type column data and approximately 50 columns of floating-point real-value data of single precision for each mass number, and approximately 250 million rows. The size of this table was reduced to approximately 20 gigabytes because the text-based data were stored in a binary format of floating-point real-value of single precision. Taking an example of the H_2O (mass number 18) partial pressure at 8 am for 6,000 days, extracting this data from the database took approximately 7 s. Although the extraction time depends greatly on the extraction conditions and the environment of the terminal and network, this was faster than extracting from 60-gigabyte text data files that took more than an hour to read. Using a database in this manner made it easier to analyze long-term fluctuations in data rather than reading a bunch of files. Examples of the results are shown in Fig. 5. Although the H_2O partial pressure depended on the status of the cryopump, it was observed that decrease over several days after the vacuum chamber was vented to the atmosphere in the QUEST device.

In the case of a Type B table, in which the primary keys are the shot number and elapsed time, time-series data for a certain shot number are often required. Previously, a text file containing fundamental parameters, such as the coil current and plasma heating power and plasma current, with more than 180 columns was output for each shot. Each file had a size of approximately 200 megabytes, and reading this file using the Python Pandas module took approximately 4.1 s. However, retrieving only two columns such as the time and plasma current from the database took approximately 1.1 s, which was a considerably shorter time frame. Another advantage of using SQL commands is

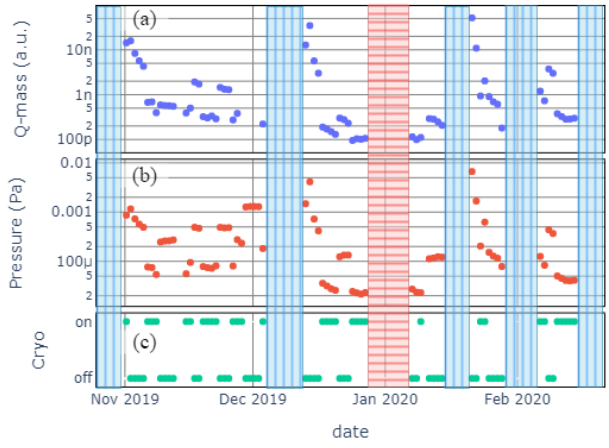

Fig. 5 Real-time monitoring data retrieved from databases. (a) H_2O partial pressure obtained by mass spectrometry, (b) vacuum vessel pressure, and (c) cryopump status. The vacuum vessel was vented during the periods indicated by bands with vertical lines, and the vacuum was maintained by sealing the vessel during the periods indicated by horizontal lines.

that particular data can be specified, such as shot numbers where the plasma current exceeded 100 kA.

The QUEST project is replacing to its web-based monitoring interface with a frontend to our database (see Fig. 4). While the previous interface showed only some fundamental data, users are now not restricted with regard to the available measurement data. The database is a convenient and fast way to access measurement results, such as the temperature of the vacuum vessel, mass spectrometer data in a vacuum, the coil current used to confine plasma, and the fuel supply to the plasma. After the initial phase of the database launch, the QUEST project has conducted workshops to gather feedback from QUEST members and is currently working with more members on integrating data previously held by individual researchers.

The study was supported by the NIFS Collaboration Research Program (NIFS20KUTR150, NIFS19KUTR136).

[1] K. Hanada *et al.*, "Steady-State Operation Scenario and the First Experimental Result on QUEST", Plasma Fusion Res. **5**, S1007 (2010).
[2] H. Idei *et al.*, "Electron heating of over-dense plasma with dual-frequency electron cyclotron waves in fully non-inductive plasma ramp-up on the QUEST spherical tokamak", Nucl. Fusion **60**, 016803 (2020).
[3] T. Onchi *et al.*, "Non-inductive plasma current ramp-up through oblique injection of harmonic electron cyclotron waves on the QUEST spherical tokamak", Phys. Plasma **28**, 022505 (2021).
[4] M. Hasegawa *et al.*, "Modification of plasma control system and hot-wall temperature control system for long-duration plasma sustainment in QUEST", Fusion Eng. Des. **129**, 202 (2018).
[5] K. Hanada *et al.*, "Overview of recent progress on steady state operation of all-metal plasma facing wall device QUEST", Nucl. Mater. Energy **27**, 101013 (2021).
[6] M. Hasegawa *et al.*, "Extension of Operation Region for Steady State Operation on QUEST by Integrated Control with Hot Walls", Plasma Fusion Res. **16**, 2402034 (2021).
[7] M. Hasegawa, D. Sakurai *et al.*, "Towards automated gas leak detection through cluster analysis of mass spectrometer data", Fusion Eng. Des. **180**, 113199 (2022).
[8] https://epics.anl.gov/
[9] J. Waterhouse *et al.*, "Introduction of ITER CODAC relevant technologies on JET and MAST", Fusion Eng. Des. **161**, 111858 (2020).
[10] R. Castro *et al.*, "Soft real-time EPICS extensions for fast control: A case study applied to a TCV equilibrium algorithm", Fusion Eng. Des. **89**, 638 (2014).
[11] W. Lee *et al.*, "New control systems at KSTAR compatible with ITER standard technologies", Fusion Eng. Des. **129**, 94 (2018).
[12] R. Yamazaki *et al.*, "Development of JT-60SA experiment database system", Fusion Eng. Des. **155**, 111555 (2020).
[13] H. Nakanishi *et al.*, "Real-Time Data Streaming and Storing Structure for the LHD's Fusion Plasma Experiments", IEEE Trans. Nucl. Sci. **63**, 222 (2016).
[14] https://mariadb.org/
[15] https://jupyter.org/