

Development of Web Interfaces for Analysis Codes

M. EMOTO, T. WATANABE, H. FUNABA, S. MURAKAMI¹⁾, Y. NAGAYAMA
and K. KAWAHATA¹⁾

National Institute for Fusion Science, Toki, Gifu 509-5292, Japan

¹⁾*Kyoto University, Kyoto, 606-8501, Japan*

(Received 4 December 2006 / Accepted 16 April 2007)

Several codes have been developed to analyze plasma physics. However, most of them are developed to run on supercomputers. Therefore, users who typically use personal computers (PCs) find it difficult to use these codes. In order to facilitate the widespread use of these codes, a user-friendly interface is required. The authors propose Web interfaces for these codes. To demonstrate the usefulness of this approach, the authors developed Web interfaces for two analysis codes. One of them is for FIT developed by Murakami. This code is used to analyze the NBI heat deposition, etc. Because it requires electron density profiles, electron temperatures, and ion temperatures as polynomial expressions, those unfamiliar with the experiments find it difficult to use this code, especially visitors from other institutes. The second one is for visualizing the lines of force in the LHD (large helical device) developed by Watanabe. This code is used to analyze the interference caused by the lines of force resulting from the various structures installed in the vacuum vessel of the LHD. This code runs on PCs; however, it requires that the necessary parameters be edited manually. Using these Web interfaces, users can execute these codes interactively.

© 2007 The Japan Society of Plasma Science and Nuclear Fusion Research

Keywords: Web interface, GUI, analysis, FIT, database

DOI: 10.1585/pfr.2.S1130

1. Introduction

To analyze the various aspects of plasma physics, several analyzing codes such as VMEC [1], HINT [2], and TASK [3] have been developed. Most of them are written in FORTRAN and run on supercomputers. Therefore, several researchers find it difficult to use these codes as they typically use personal computers (PCs). For example, a majority of these codes are command-line applications that have to be invoked through terminal emulators, while PC users typically use GUI-based applications with mouse operations. Furthermore, the necessary parameters are usually provided as text files and users have to write them manually. This often results in mistakes.

In order to facilitate the widespread use of the existing codes, a user-friendly interface is required. Therefore, the authors have developed Web interfaces for these codes. Using these interfaces, researchers can provide the necessary parameters interactively to the codes through familiar Web browsers. To demonstrate the usefulness of this approach, the authors have developed two systems. The first one is an interface for the FIT [4] code and the second one facilitates the visualization of the lines of force in the LHD (large helical device) [5].

The FIT code was developed by Murakami to analyze NBI heat distributions. It runs on supercomputers and requires a text file containing the distributions of electron temperatures, electron densities, and ion temperatures as

polynomial expressions of minor plasma radius. This approach poses difficulties for users who are not a part of the project because they do not know where the data is stored or how to read it. Therefore, for such users, a more user-friendly interface is required to execute the code.

The LHD code was developed by Watanabe to analyze the interference caused by the lines of force resulting from the various structures installed in the LHD vacuum vessel. Unlike FIT, this code runs on PCs, but it still requires the necessary parameters to be provided through a text file. Furthermore, it utilizes a considerable amount of computer resources and requires a large amount of time to perform the calculations, depending on the conditions. Therefore, a batch system is required to allocate the appropriate computer resources to execute the code. To facilitate the management of such batch jobs, a user-friendly interface is required.

2. System Overview

Fig. 1 shows the overview of the Web interface for the FIT code, Fig. 2 shows the visualization of the lines of force, and Figs. 3 and 4 show their web interfaces. Both systems use a three-tier model that comprises a user interface component, web servers, and calculation component. The role of the user interface component is to provide a graphical interface to edit the necessary information interactively. This component is written in HTML and JavaScript. Therefore, users can use popular web browsers

author's e-mail: emoto.masahiko@nifs.ac.jp

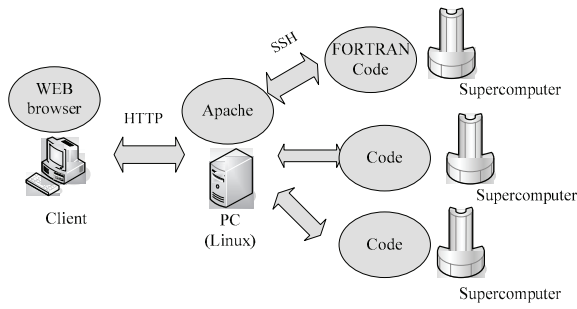


Fig. 1 Overview of the Web interface for the FIT code.

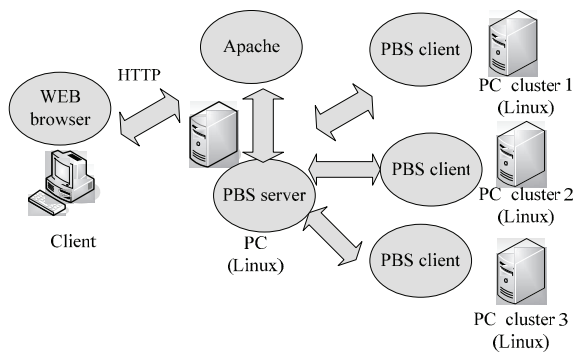


Fig. 2 Overview of the visualization system for the lines of force in the LHD.

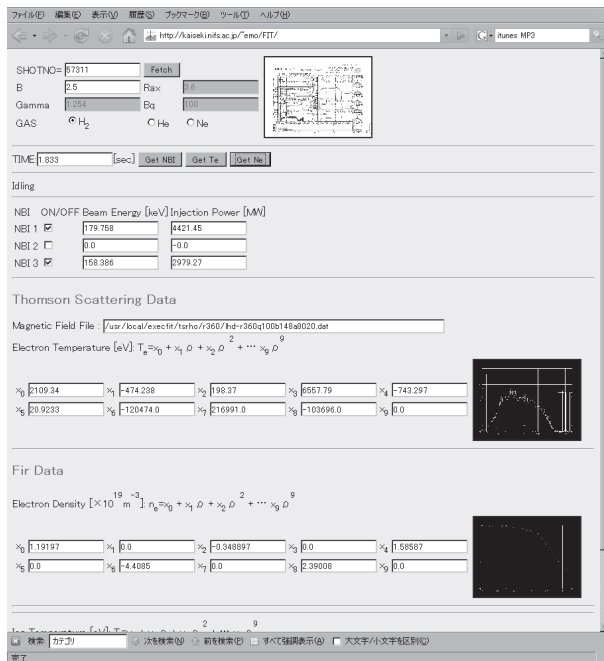


Fig. 3 Web interface for the FIT code.

such as Internet Explorer, Mozilla Firefox, etc. The calculation component is implemented as CGI (Common Gateway Interface) programs on an Apache web server. This model was chosen for the flexibility it provides. Because

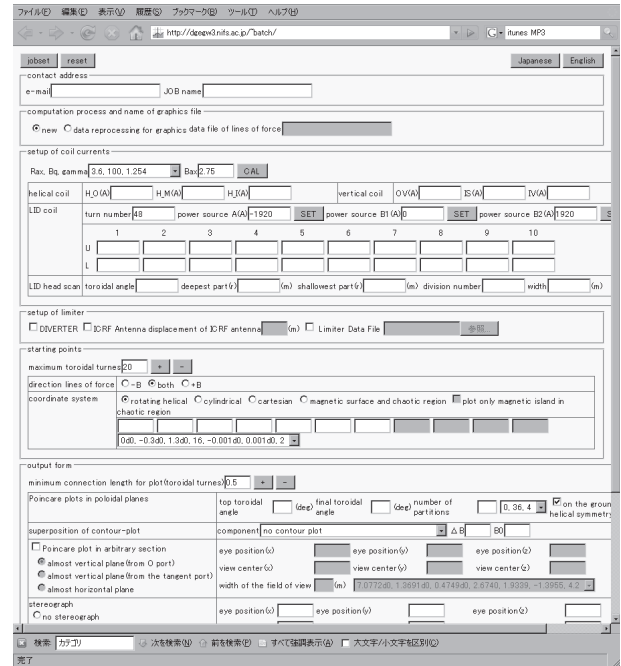


Fig. 4 Web interface for the visualization of the lines of force.

each tier is loosely connected, it is relatively easy to replace a module without modifying the existing code considerably. Thus, the system is easy to maintain.

The calculation part of the FIT code system consists of two components—a PC application and the analyzing code residing in a supercomputer. The main calculation is performed by the supercomputer while the PC application provides the necessary information to the users. The calculation component of the visualization system also comprises two sub-components—a batch server and several batch agents. The batch server sends a job to one of the batch agents, and the main calculation code is executed by the agent application.

The other basic information required to execute the analysis codes, such as magnetic field, coil current, gas, etc., are managed by a relational database. The information regarding experiments is recorded automatically in the database during the experiments, and the analyzed data are recorded by the batch programs at night. Users can search for data by specifying various physical conditions, for example, magnetic field = 3.6, electron density > 1.0¹⁹ [m⁻³], etc. Usually, users are required to write SQL (structured query language) statements to perform such complicated searches. SQL is a common language used to perform searches in relational databases; however, those unfamiliar with SQL can use the proposed Web interfaces to search through the data (Fig. 5). The proposed system enables users to compile SQL statements interactively to perform searches easily. By adding these SQL statements to their programs, users can search through the database without any knowledge of SQL.

The subsequent sections provide detailed information

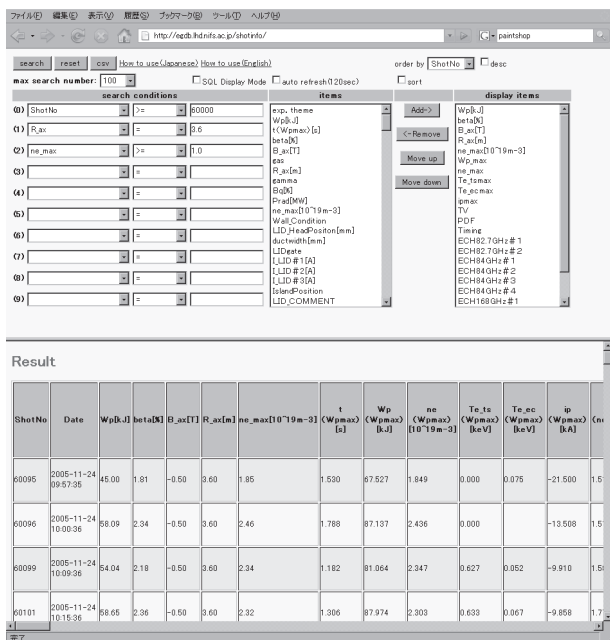


Fig. 5 Experimental database.

regarding the proposed system.

2.1 User interface

Essentially, both systems consist of Web interface and CGI components. Users provide the necessary parameters through the Web interface and call the CGI programs to run the analyzing codes. A basic HTML interface can do this task. However, it takes time to execute the analyzing codes and the Web browser has to postpone displaying the results page until it receives the results from the CGI component. Therefore, a user has to wait until the next page loads; this disrupts the flow of the site. This is because the CGI program is called “synchronously” from the browser and the browser postpones the loading of a page until it receives the content. To solve this problem, the authors adopted AJAX (asynchronous JavaScript and XML) [6]; the Web interface is written in both HTML and JavaScript. The JavaScript code calls the CGI programs and the browser displays the page “asynchronously.” Instead of waiting for the entire content to arrive, the browser shows a part of the content; when the JavaScript code receives the rest of the content from the CGI program, it redraws the page. This technique is used to retrieve graphs from the server. The graph shown in Fig. 3 is plotted by the server and it takes several seconds to perform this task. However, because the browser retrieves the graph asynchronously using AJAX, the user can input other parameters on the same page while the graph is being plotted at the server end (Fig. 3).

2.2 Connection between PC and supercomputer

The PC CGI program uses SSH (secure shell) to submit a job to execute the FIT code residing in the super-

computer. There exist other remote access methods such as telnet and rsh (remote shell). However, because they send messages to remote servers without encrypting them, user accounts and passwords might be eavesdropped upon while they are being sent to the server. Unlike these programs, SSH is an encrypted protocol that establishes secure communications.

Because currently there is no facility available to notify the user when a batch program ends, the PC program checks the status of the queue at intervals of 1 s to determine whether the entire batch program has finished executing. When the program detects the end of the batch program, it sends the results to the web server using SFTP (secure file transfer protocol).

2.3 Multiple users problem

A majority of the analysis codes that run on PCs are developed to be executed by a single user; programmers do not assume that their programs will be executed by multiple users simultaneously. For example, the polynomial fitting code used in the system uses a fixed configuration file that contains the names of the data files and users have to modify this configuration file to tell the program what data file to use. However, the proposed system can be used by several users simultaneously. The abovementioned scheme would not work in a multi-user environment because another user might change the configuration file while the program is still running. To resolve this problem, there are two options: 1) create a new configuration file each time the program is invoked or 2) provide the necessary parameters via other mediums such as command-line options. Because the latter approach would require lesser modification to the existing system than the former, the author adopted the latter approach and modified the program such that it obtains parameters via command-line options.

2.4 Batch job management

The execution time required to visualize the lines of force depends on the number of grids—it increases with the number of meshes and can exceed one day. Because the user cannot wait until the program finishes, a batch system is required. The author adopted a PBS (portable batch system) [7] to manage batch programs. The batch jobs generated by the web server are queued in the PBS server and the PBS manager requests one of the calculation servers to execute the jobs. This system determines which PC from a cluster is chosen to execute the job in a round robin manner—the first PC server found to be free is assigned the job. However, the selection algorithm can be changed to utilize CPU power more efficiently. Because the PCs are shared by other users for various purposes, the batch job could result in performance problems for those users. However, because the system is scalable, it is relatively easy to remedy the performance problem by simply adding new PCs to the existing system.

3. Future Work

3.1 Interfaces between programs

The authors are planning to integrate other analysis codes into this system to enhance its usability. However, the problem in performing this integration is that these codes are not always programmed to run by multiple users and they often use a fixed-name file to read/write data. To resolve this problem, the authors modified the codes such that they obtain parameter from command-line arguments. However, this modification cannot work for some codes because some computer languages do not possess the functionality to obtain command-line arguments.

Another approach to receive parameters from external sources is to use environment variables. In this case, programs can use environment variables such as PARAMETER_FILE to determine which file they should use. However, whether these systems can use environment variables or not depends on their implementations, and some systems do not possess this functionality.

Therefore, the authors propose to replace system calls. When a program uses a file, it eventually uses the “open” system call. Therefore, the authors propose to provide an environment wherein the original system call “open” will be replaced by another function. This environment will contain a file mapping table; when a program tries to open a specific file present in the table, it will use a different file listed in the table that corresponds to the specified file. The advantage of this method is that the original programs do not need to be modified. Furthermore, since the current system is implemented on the Linux operating system, which allows for remapping system calls, it would be easy to implement this scheme.

3.2 Enhancing the database

In order to perform searches based on various parameters, the authors have been expanding the database to store as much data as possible related to the LHD experiment. The database currently provides the configuration and analyzed data. However, a few plasma analysis programs require further analyzed physical data such as magnetic surface, heat distribution, density distribution, etc. These are calculated from the basic physical data and they might differ from one researcher to another depending on their models. To study the physical phenomena of plasma dynamics in detail, researchers must use the same data. Therefore, the authors have developed a system to provide the basic analyzed data such as vacuum magnetic surface for each plasma discharge experiment and also plan to develop a system capable of providing data such as electron temperature and density profiles.

4. Conclusion

The authors have developed two systems to demonstrate the usefulness of Web interfaces for analysis codes. Using this approach, the authors plan to develop a user-friendly interface for other existing analyzing codes.

- [1] S.P. Hirshmann *et al.*, *Comp. Phys. Comm.* **43**, 143 (1986).
- [2] K. Harafuji *et al.*, *J. Comp. Phys.* **81**, 169 (1989).
- [3] A. Fukuyama *et al.*, *Proc. of 20th IAEA Fusion Energy Conf.* (2004) IAEA-CSP-25/TH/P2-3.
- [4] S. Murakami *et al.*, *Fusion Eng. Des.* **26**, 209 (1995).
- [5] T. Watanabe *et al.*, NIFS Report NIFS-TECH-14 (2006).
- [6] J.J. Garrett, Ajax: A New Approach to Web Applications, <http://www.adaptivepath.com/publications/essays/archives/000385.php>.
- [7] <http://www.openpbs.org/>