

§4. Web Interfaces to Data Visualization for LHD and Other Experiment

Emoto, M., Yoshida, M., Iwata, C., Nagayama, Y., Inagaki, S. (Kyushu Univ.)

The author have been developing remote manipulation functions of various applications for LHD experiment such as visualization and retrieving of acquired data in order to facilitate remote participation of LHD experiment [1]. To realize these functions, we adopted web interfaces. This is because the Web browser is the most common application of current operating system, and the users don't need to install other programs to use the applications. This feature is very useful especially for those who are not familiar with the system such as visitors.

To realize Web interface for data visualization, the author adopted AJAX (Asynchronous JavaScript plus XML)[2]. AJAX is a technique to realize interactive web application using JavaScript without installing other software such as Java Virtual Machines. However, AJAX must use both JavaScript and HTML at the same time, and it worsens the readability of source codes and becomes hard to maintain the source codes. To solve this problem, the authors adopted Ruby on Rails [3] as an application frame work. Fig 1. shows the web interface to data visualization tool. The user uses an ordinal web browser to use this application, using AJAX interface the user can customize the way visualize the data interactively. Fig. 2 shows the overview of this system. When the Ruby on Rails server receives the request from the client, it invokes the visualization programs. The visualization programs used here is originally an X-Window System based program to display graphs of the raw data developed by Inagaki. The authors modified this program to produce PNG and PDF graphic images, and the graphic images are stored into the relational databases. The relational database is also used to store the user settings information. This information is reused by other users, and providing the setting beforehand, the user can use the setting as templates. Using this scheme, the authors did not modify the existing visualization tools very much, and we could have reduced the development processes.

The disadvantage to use AJAX is that the CPU load centralized at the Web server because most of the transaction is done by the web server. It worsens the response of the program especially when the many users use the application at the same time. To improve the response, the author is planning to use distributed processing system. Fig 3 shows the overview of the distributed system. The visualization process, which is the most CPU power demanding process, runs on the other computers. When the server receives the request from the client, the Rails server runs the visualization process using SSH, and it stores the calculation results into the database. Using this approach, the author can make the system scalable just by increasing the number of servers.

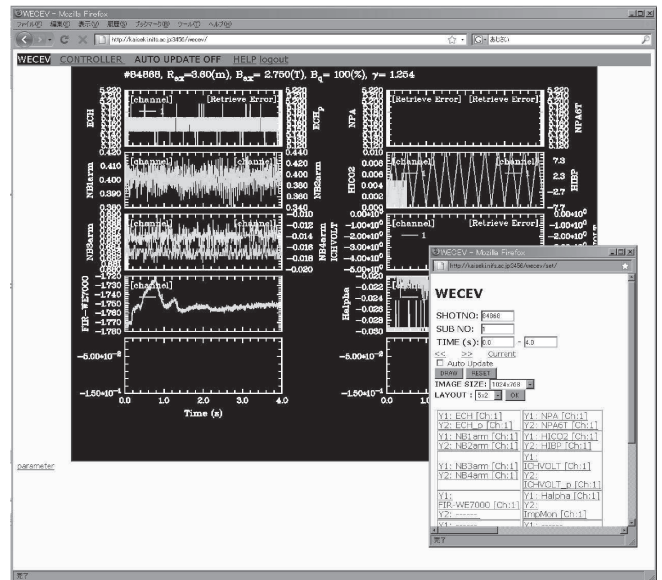


Fig. 1 Web Interface to the Data Visualization Tool

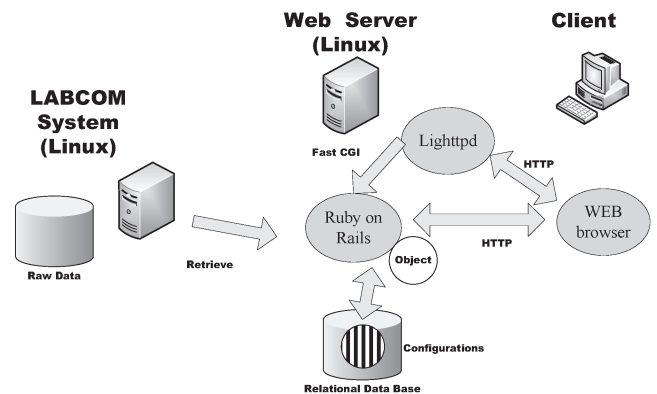


Fig.2 System Overview

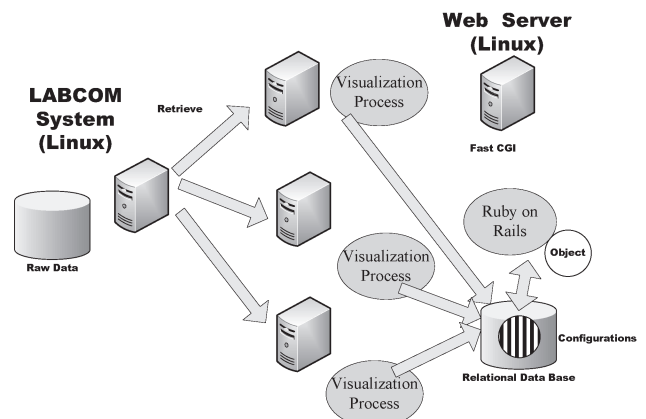


Fig.3 Distribute System (Future Plan)

- 1) IAEA Technical Meeting, 2007, Inuyama, Japan
- 2) <http://www.adaptivepath.com/publications/essays/archives/000385.php>.
- 3) <http://www.rubyonrails.org/>