Java Based Data Monitoring and Management System for LHD

M. Emoto, M. Shoji, S. Yamaguchi, National Institute for Fusion Science, Toki 509-5292, Japan
J. Kariya, Yamaguchi University, Ube 755-8611, Japan
H. Okumura, Matsusaka University, Matsusaka 515-8511, Japan
Y. Teramachi, University of Industrial Technology, Sagamihara 229-1196, Japan

Abstract

We developed a Java based experiment monitoring and management system DODAS (Databased Open Data Acquisition System) for LHD(Large Helical Device) at NIFS(National Institute for Fusion Science). This system is based on the two key concepts “database” and “open”.

“Database” means that system maintenance facility and experiment monitoring facility are integrated in order to guarantee correct experimental conditions.

“Open” means that the realtime data are shared among distributed computers on the network for load sharing and incorporating new data processing ideas.

1 INTRODUCTION

This system is developed for the LHD in NIFS. It can monitor many slow and real-time data such as such as temperatures of the vacuum vessel and the divertors, the currents and voltages of the superconducting magnets, the pressures and flow rate of the liquid helium and so on. It also gathers many fast sampling data with the slow data in the same time in same ADCs (Analog-Degital Converter). The number of ADC can be expanded easily and is available up to 4000 channels.

This system is an extended version of the prototype system [1]. In the present state, the system is continuously recording 512 channels of data in 1Hz and records 64 channels of data in 1kHz while the experiment. The number of ADC channels will be doubled in the future.

There are so many measuring data, the most important things are how to maintain correct experiment conditions and how to many members attend to the experiment and monitor the data.

Therefor we developed the system on the keyword “database” and “open” by Java language. We adopted Java because it is adequate for developing distributed open system and rapid, reliable programming.

2 DATABASE

The information and controls of the system are centralized to the database. The functions of the database are shown bellow by the menu titles. They are grouped to 3 parts.

2.1 Data Management Menu

- User Registration
- Purchase
- Data Acquisition Front
- LHD Component
- Location of Sensor Registration
- Measuring Item
- Shield Box
- Power Supply for Amplifier in shield box
- Power Supply Tag for Amplifier
- Power Supply Tag Assign for Amplifier
- Power Supply for Sensor
- Power Supply Tag for Sensor
- Power Supply Tag Assign for Sensor
- Amplifier
- Amplifier Tag
- Amplifier Tag Assign
- Sensor Purchase
- Sensor Conversion Table
- Virtual Tag Registration

2.2 Experiment Set up Menu

- Tag Monitoring
- Experimental Registration
- Connection Search
- ADC Control
- System Configuration Maintenance
- Quench Set up

2.3 Experiment Control Menu

- Experiment Selection
- Experiment Note
- Experiment Member Registration
- Experiment Status Control
- Slow Sampling
- Fast Sampling
- Realtime Monitor
- Slow Data Search
- Fast Data Search

2.4 Features of the database

As shown in the above menu, we want to manage everything in the database as far as possible. Now it keeps

information of the components such as amplifiers, sensors, connections between them, and others to maintain the LHD status.

The tags are designed to represent everything and attached to the components of the LHD. For example, the menu “Virtual Tag Assignment” means to assign tag to the high-level measurement quantity that is derived by computing other measured quantities. If a tag is used in a document, we can know that from which quantity the quantity was derived and with which sensor, amplifier and ADC it was measured.

In the “Experiment Note” menu, the experiment members write memos in the applet window, which usually was written in the personal notes. These memos are time stamped, related to the experiment and saved in the database. The members share the collected and time-threaded memos written by all members and know whole aspect of the experiment.

In the “Tag Monitoring” menu, we can set the upper and lower bounds of tag values. The system monitors those values and when it finds out of range value, it sends e-mail and telephones alarm to the staff.

3 OPEN SYSTEM

The system collects data and provides them in the form of strip charts applet [Fig. 2] to the members in realtime. Because there are so many data, the system has not enough performance to provide simultaneously every data. Therefore we must introduce some mechanism of load sharing.

The data acquisition machine, in which ADC interface board is equipped, sends realtime data in multicast packets. Any machine on the segment can be a client of the data server and it receives these packets and processes the data. The acquisition machine itself is one of the clients. It receives raw data and compute high level quantities. Another client is a data saver and it periodically creates saved data files. The web server can access these files by NFS (Network File System).

Provided that 1) on the 10BASE ethernet dedicated to the realtime data transfer, 2) running the receiver program in the realtime scheduling class on a SPARCstation 2, the experiment [2] showed that the program can receive 800 packets/sec with 1040 bytes length packet. The transfer time from the sender to the receiver is 3msec.

It is a useful way to use a network as a realtime data bus. If we use a gigabit ethernet and high performance machines, we may send 10kHz realtime data.

4 DEVELOPMENT BY JAVA

We developed the system by Java language, because Java is adequate to develop reliable distributed systems. Due to the security of JDK1.1, the Web server, RMI daemon and database server run on the same machine. The programs on the data acquisition machine directly control the ADC device, so they are written in C language. In order not to lost realtime data, the data acquisition machine is separated from the Web server and only fixed jobs run on it. As shown in Fig. 1, device control messages such a “start ADC” are sent from applet to RMI daemon by RMI (Remote Method Invocation), and sent to C language interface by JNI (Java Native Interface), then sent to the Data acquisition machine by RPC (Remote Procedure Call).

5 PLAN

- Developing data client programs.

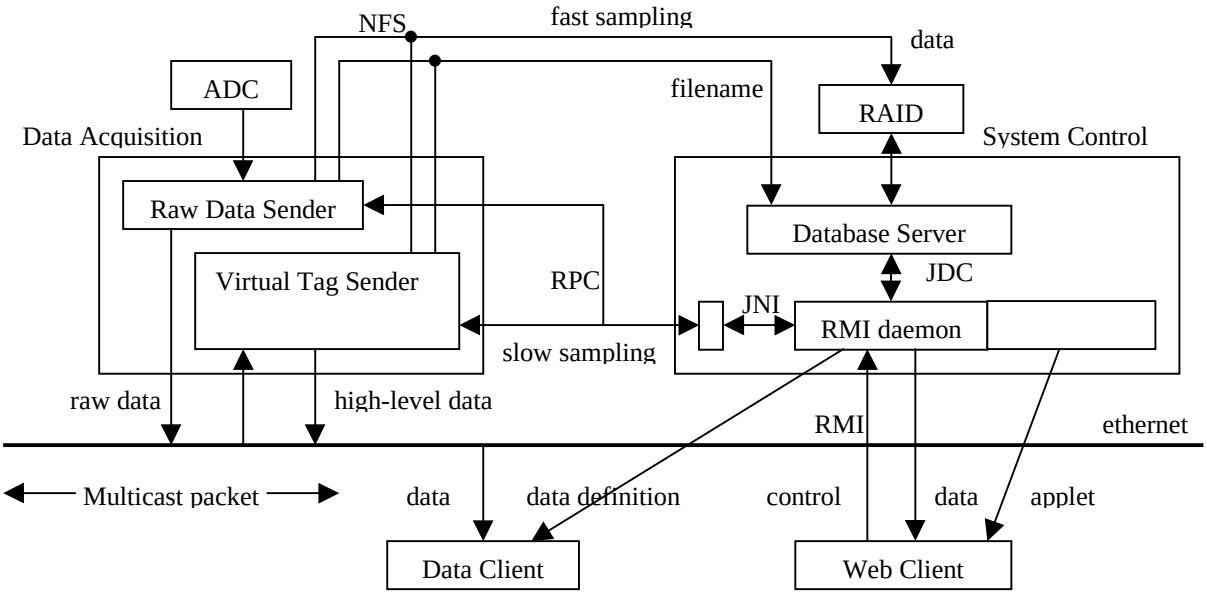

Figure1: System configuration

We install the data client programs to the distributed machines on the network data bus, so that all of data can be shown at the same time.

- Adaptive setting of bounds of tag values according to the stage of LHD control.

We can know the unexpected statuses more early, so we can do finer control of LHD.

REFERENCES

[1] S. Yamaguchi et al., J. Plasma Fusion Res. 73, 335 (1997). (in Japanese)

[2] J. Kariya et al., J. Plasma Fusion Res. 74, 67 (1998). (in Japanese)

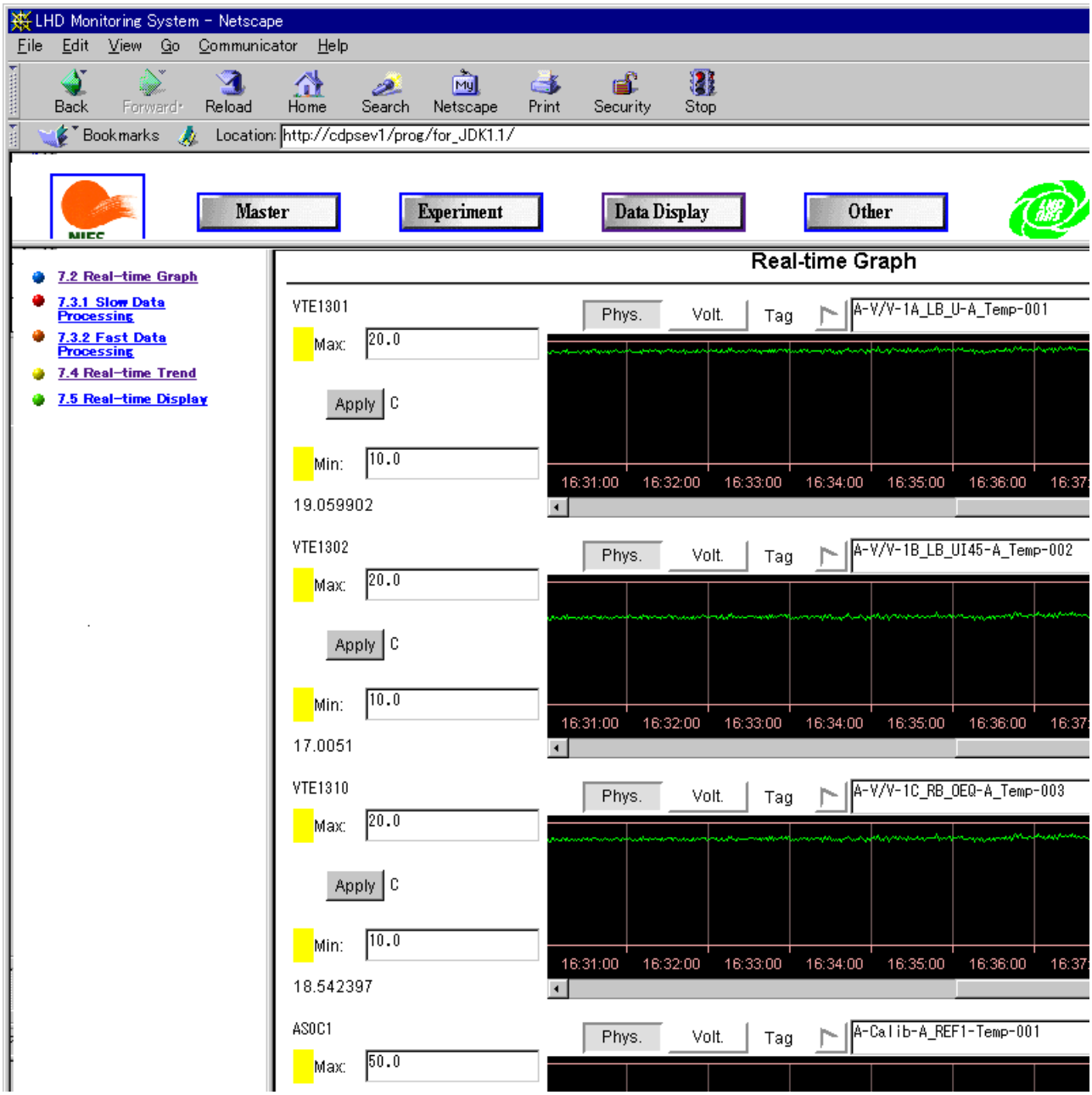

Figure2: Real-time Graph