DEVELOPMENT OF DATA LOGGING AND DISPLAY SYSTEM, MYDAQ2

T. Hirono, T. Matsushita, T. Ohata, A. Yamashita, JASRI/Spring-8, Hyogo, Japan

Abstract

MyDAQ [1] is a simple data logging and display system using a relational database. It is "a chart recorder on the web" logging data like temperature or pulse motor status. MyDAQ receives data from data taking PCs through the network, stores them to the database with timestamp and displays them as a time-chart on browser. Client users are only required to add small TCP socket sequence to their data taking program. We upgraded MyDAQ to MyDAQ2. The main features of the upgrades were (1)support of binary and text data, (2)asynchronous network communication, (3)data management functions, (4)user friendly data viewer and (5)inclusion of an installer. We adopted MyDAQ2 as one of user interfaces of the newly introducing common data storage in SPring-8 beamlines.

INTRODUCTION

We have developed a simple data logging and display system using a relational database, MyDAQ, since 2002 [1]. Many data acquisition system was developed using MyDAQ, such as a temperature monitor system of monochromator or a flow logger system of liquid nitrogen tanks. The MyDAQs were used as a part of rather small data acquisition system that log data less than 1000 points in about 0.1Hz, not as large as the data acquisition system of the whole accelerators in SPring-8.

The success of the MyDAQ is due to its simple data-input interface and web browsing feature. The data-input interface is a TCP socket that receives data in string message. Any data-taking programs written in any language can use MyDAQ by adding small socket communication sequence without SQL programming. Users of MyDAQ can get benefits of the relational database without skills of SQL or table managements of databases.

However, MyDAQ was so simple that some features were lacked to apply to complicated data acquisition systems. More stability is required also. We upgraded MyDAQ and released as MyDAQ2. In this paper, we shortly review design concept of MyDAQ2 that inhabited from MyDAQ in section 2 and shows the new features in section 3. We present a newly introduced data storage system as an application of MyDAQ2 in section 4. We summarize our study in section 5.

DESIGN OF MYDAQ2

The main design concept of MyDAQ2 is basically same as the previous MyDAQ. Figure 1 shows the structure of MyDAQ2. MyDAQ2 is consisted of a TCP socket server, a relational database, a plot application and an http server with Common Gateway Interfaces, CGIs. The TCP socket server and CGI are written in an object-oriented script language, Python [2].

The users send their data via network to the TCP port of MyDAQ2 server. The data must be formatted in string message with a delimiter and separators like "create/ my_temperature/fi\r" or "put/my_temperature/12.3_0\r". A signal named "my_temperature" with a float and an integer was created by the first message. The second message asks MyDAQ2 to save value of "my_temperature" as 12.3 and 0.

In MyDAQ2, one signal corresponds to one database table. The socket server creates a table named "my_temperature" that has columns of one float and one integer with columns of UNIX-time and microseconds by the create-message in the above example. The TCP socket server receives the message and sends back a message with "ok" like "my_tempatrue/put/mydaq/ok\r" to notify the user that MyDAQ2 server received the user's message and save them into database successfully. The TCP socket server also translates the message into SQL and save the data into relational database with a timestamp.

The user access the http server of MyDAQ2 to view the data. The http server displays a html page with a list of signals, which is automatically created by a CGI script. The users can get a time chart by following a link of the html page and specify a period of time. CGIs of MyDAQ2 select data from the relational database make chart image with the plot application and create html page with the time chart. The data can be displayed in tab-separated text format, which allow the user further analysis of the data.

Table 1: Application list of MyDAQ2

Application	Application Name	Version*
Http server	Apache [3]	2.0
Plot application	gnuplot [4]	4.0
Relational database	MySQL [5]	5.0
Script Engine	Python	2.5
	Twisted [6]	2.5
	MySQL-python [7]	1.2

*A version we have tested.

#hirono@spring8.or.jp

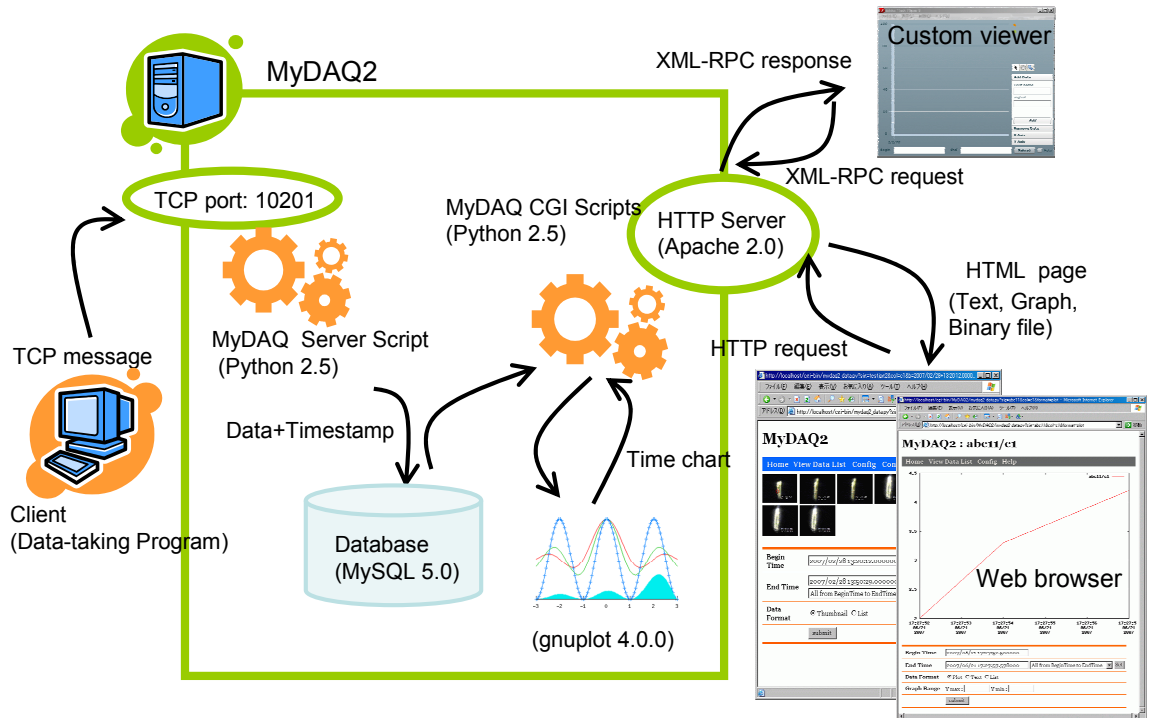

Figure 1: The structure of MyDAQ2. MyDAQ2 is consisted of an http server (Apache), relational database (MySQL) graph application (gnuplot) and scripts written in Python. The users of MyDAQ2 input data by sending ASCII message and browse data on web.

MyDAQ can be used with user's data-taking program on any OS with any programming language, which supports socket communication, because its interface are a TCP socket and web browser. The user only add small sequence into the user's program, because raw TCP socket communication can be implemented with simple API comparing to other protocol such as RPC. The applications that MyDAQ2 consist of are given in Table1. We adopted newer versions of the applications by upgrading MyDAQ.

UPGRADED FEATURES

There are 5 main features that were upgraded that enable MyDAQ2 to apply various data acquisition system.

Support of Binary and Text Data

We added two data types that MyDAQ2 can handle, which are binary and text. MyDAQ2 is able to handle floats, integers, binary and text.

Since the protocol between the user's client and the TCP socket sever of MyDAQ2 is string message with delimiter, we designed a protocol for binary data as base-64 encoded data with Multipurpose Internet Mail Extensions, MIME, header. The TCP socket server saves binary data as-is sent, base-64 encoded with MIME header. The CGIs of the http server provide the binary data as a file.

MyDAQ2 supports text includes 2-byte characters. A code of 2-byte characters differs according to OS. We designed a protocol for text data as base-64 encoded data with the character code name as a header. The TCP socket server decodes base-64, change code of characters into UTF-8, encode base-64, put MIME "text/plain" and save into the database.

This addition of data type enables enhanced use of MyDAQ2. For example, a user can save jpeg images with comment on them. MyDAQ2 can be used as an e-logbook.

Asynchronous Network Communication

We refactored the TCP socket server using Twisted[6]. Twisted is a library that supports asynchronous network communication and database access. Twisted reduced the implementation of complicated error handlings of TCP communication, which shorten the period of development of MyDAQ2. And also, stability of the TCP server was increased.

Data Management Functions

MyDAQ2 is able to "group" the signals. This feature makes the user easy to access data on web. The user can create group and associate signals with the created group through web.

Column name can be also customized by the users. For example, the users can name the first float of "my_temperature" in Section 2 as "Temperature [K]" and the second integer as "Status of Thermometer". The customization of the name of column are also done thorough the web.

We added three management tables to the relational database. These tables manage signal and group associations, group name and column name.

User Friendly Data Viewer

In addition to time chart, we developed some CGIs as data viewer, such as list of links to file for binary data, thumbnail for image data and spectrum plots for wave form data. We also developed a data viewer which customized to a specific data, which is a status report for pulse motor controller.

MyDAQ2 has a CGI that act as a XML-RPC server. The XML-RPC server responds to user's request that specify signal and range of time period. It returns data in XML format instead of a time chart. The XML-RPC server enables users to develop their customized data viewer. The users are also able to develop data viewer that shows data from several MyDAQ servers.

Appearance of html page was also revised by using cascading style sheets.

Inclusion of an Installer

We included an installer to MyDAQ2. The installer installs the CGI scripts and the TCP socket server of MyDAQ2 and setups tables and account of the relational database. The user can install MyDAQ2 by standardized installation command of Python, which is "python setup.py install".

A DATA STORAGE SYSTEM WITH MYDAQ2

In SPring-8, beamline common-use data storage has been prepared. MyDAQ2 will be one of interface application for the data storage. Since MyDAQ2 is assumed to used as a part of rather small data acquisition system in each beamline, we required to provide low performance but multiple and "separated" MyDAQ.

The architecture we designed was shown in Figure 2. We introduced a blade servers system, in which virtualized OSs run. MyDAQ2 was installed in the virtualized OSs. The data storage was mounted via NFS. We provide one MyDAQ to one group of people who share same data. The users use assigned MyDAQ2 on virtualized OS, and the data of people in other group would not be mixed.

This architecture provides several MyDAQ2s on one blade server. We can provide MyDAQ2 on users demand by adding virtualized OS.

FUTURE PLAN

MyDAQ2 is basically designed as a system that used by one user or small group. However, we upgraded MyDAQ2 to able to apply larger and complicated data acquisition system used by many users. A data access control may be required in future. MyDAQ2 dose not have features for data access control itself. The data access is only controlled by the http server or a firewall of OS. We start designing the next MyDAQ2 with fine-turn able access control.

We are also preparing a web site for MyDAQ2 to support the users to install MyDAQ2 and develop data-inputting client programs [8].

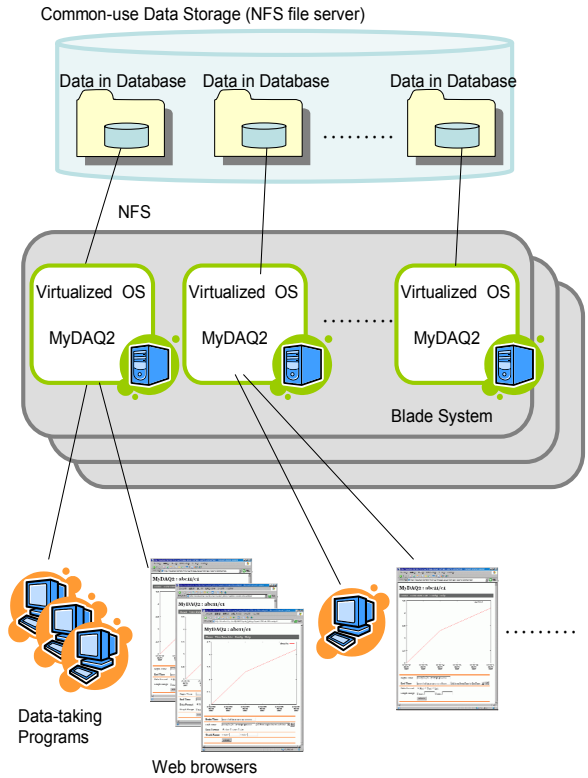

Figure 2: Architecture of MyDAQ systems with common-use data storage.

CONCLUSION

We concluded our development as flows,

- We upgraded MyDAQ to MyDAQ2. MyDAQ2 also has a SQL-free TCP socket input interface and data browser on web.
- The upgrade expanded the variety of data acquisition system that MyDAQ is able to apply to.
- MyDAQ2 was successfully upgraded and worked well as expected. We will provide MyDAQ2 as one of interface common-use data storage for beamlines.

REFERENCES

[1] A. Yamashita and T. Ohata, "MyDAQ, a Simple Data Logging and Display Server", Proc. of PCaPAC'05, Hayama, Japan, 2005.
[2] http://www.python.org/.
[3] http://www.apache.org/.
[4] http://www.gnuplot.info/.
[5] http://www.mysql.com/.
[6] http://sourceforge.net/projects/mysql-python/.
[7] http://twistedmatrix.com/trac/.
[8] http://acc-web.spring8.or.jp/~hirono/.