CONSTRUCTION OF NEW DATA ARCHIVE SYSTEM IN RIKEN RI BEAM FACTORY

M. Komiyama[#], N. Fukunishi, RIKEN Nishina Center, Wako, Saitama, Japan
A. Uchiyama, SHI Accelerator Service, Ltd., Shinagawa, Tokyo, Japan

Abstract

RIKEN Radioactive Isotope Beam Factory (RIBF) is a cyclotron-based next-generation radioactive beam facility. RIBF uses two types of control systems; an experimental physics and industrial control system (EPICS)-based system and a group of several non-EPICS-based systems. For each control system there is a corresponding data archiving system in operation, and in order to unify these two data archives, since October 2009 we have been developing a new archiving system. This new data archive system, named RIBF Control Data Archive System (RIBFCAS), is required, at intervals between 1–60 s, to collect and store more than 3000 data generated by EPICS controlled devices through EPICS input/output controllers (IOCs). In addition, RIBFCAS must be able to combine the existing non-EPICS-based systems. To fulfill these requirements, a Java-based platform has been created, and client applications are based on Adobe AIR runtime. The RIBFCAS hardware system, therefore, consists of an application server, a database server and client-PCs. Presently, we have succeeded in stably acquiring approximately 3000 data from 22 IOCs every 10 s. Moreover, incorporation of the non-EPICS-based data archive system into RIBFCAS is now in progress.

OVERVIEW OF RIBF CONTROL SYSTEM

With an aim to explore unknown properties of unstable nuclei, the RIKEN Radioactive Isotope Beam Factory (RIBF) began operation in 2006 as the first of several next-generation RI beam facilities planned worldwide. RIBF is a cyclotron-based in-flight facility, and RIBF accelerators can supply RI beams at energies hundreds of MeV/nucleon over the entire range of atomic masses [1].

As shown in Fig. 1, the RIBF control system has a degree of complexity. The primary components of RIBF accelerators, such as magnet power supplies, beam diagnostic devices and vacuum systems, are controlled by an experimental physics and industrial control system (EPICS)-based system [2]. Additionally, there are several standalone control systems not integrated into the EPICS-based system, for example, those for radio frequency (RF) systems, the cooling water system and various long-standing ion sources. These systems operate independently of the remainder of the RIBF control system.

The EPICS-based system utilizes a number of device controllers, such as a Versa Module European (VME) control board, numerous types of programmable logic controllers (PLCs), general-purpose interface bus (GP-IB) controllers, PIC network interface card kit (PICNIC) [3], computer automated measurement and control (CAMAC) in-house controllers and in-house controllers based on network interfaces,. To integrate these device controllers within the EPICS framework, EPICS input/output controllers (IOCs) running “iocCore” software are necessary. “iocCore” is a set of EPICS routines that define process variables and implement real-time control algorithms. In our system, almost all EPICS driver/device supports for controllers can also be executed on Linux-based IOCs.

CAMAC in-house controllers, Device Interface Modules (DIMs), are managed by using the network crate controller CC/NET, a commercial product of Toyo Corporation [4, 5]. Because CC/NET is a Linux-based single-board computer, we can execute EPICS base software on it, and therefore the CC/NET itself becomes an IOC. In addition to CC/NET, a PLC-CPU module newly developed by Yokogawa Electric Corporation, F3RP61-2L (hereinafter, RP) [6], also works as an IOC; a soft real-time Linux environment. In contrast, the network-based in-house controller, Network-DIM (N-DIM) [7], the GP-IB controllers, PICNIC and the PLCs do not contain general operating systems (OS) on which to execute EPICS base software. To manage these controllers in the EPICS framework, they must be connected to additional IOCs through Ethernet connections to convert the communication protocols between EPICS channel access (CA) and the N-DIM or GP-IB controllers or PICNIC or PLCs. Linux-based IOCs, small single-board computers (ALIX) [8] are used to convert the protocols. The exception to the above is for NIO, a commercial control board from Hitachi Zosen Corporation, which is instead controlled by a device/driver support originating from VxWorks.

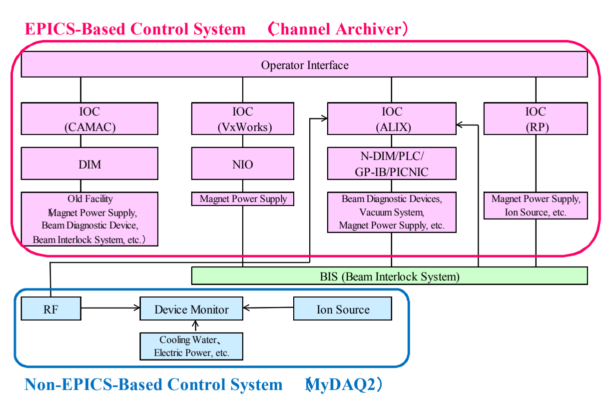

Figure 1: Schematic of RIBF Control System.

misaki@riken.jp

DIMs control approximately 370 magnet power supplies, and 100 beam diagnostic devices and vacuum systems. N-DIMs control around 250 beam diagnostic devices and vacuum systems. NIO boards control about 420 magnet power supplies. RPs control approximately 50 magnet power supplies and newly developed 28 GHz electron cyclotron resonance ion sources [9]. PLCs control the vacuum systems of cyclotrons and certain beam diagnostic devices. Although RIBF has many PLCs, EPICS controls only a proportion of these; Other PLCs have their own control systems and are independent of EPICS.

Two data archive systems have been used to record the various parameters of RIBF during operation. The pair "Channel Archiver" and "Archive Viewer", developed by the EPICS collaboration [10], is used as a data archiver for the EPICS control system. A new version of Channel Archiver, "RDB Channel Archiver", was released in 2010 to store data and configurations in a relational database (RDB); however, our system still uses the old release.

The second achieve system, "MyDAQ2" developed by the SPring-8 control group [11], is used to acquire and store non-EPICS-based systems parameters. Initially, each non-EPICS-based system individually managed their own data, and MyDAQ2 was introduced in order to manage all non-EPICS-based data in a cohesive manner. MyDAQ2 obtains data from each device through an Ethernet and stores them in a MySQL database such that users can retrieve stored data by using an associated Web application.

Although coexistence of two data archives covering different sections of the accelerator system does not cause significant problems in daily use, unifying the two database systems is favored. However, it is difficult to apply Channel Archiver to our non-EPICS-based systems. On the contrary, while it may be technically feasible to apply MyDAQ2 to our EPICS system, handling vast amounts of data is highly demanding. MyDAQ2 was developed for users who perform experiments at one of the SPring-8 beam lines, not the entire Spring-8 control system, and as a result the interface of MyDAQ2 is suitable for handling smaller amounts of data.

A drawback of the coexistence of two data archives is that data processing is necessary to investigate correlations in the stored data. For example, suppose we notice a gradual decrease in beam intensity, we are then required to investigate the correlation between operation parameters, such as temperature of cyclotron magnets and beam intensity. Since beam intensity data are stored in the Channel Archiver and magnet temperature data are stored in MyDAQ2, data processing is needed for comparison between the data. With a unified system, performing correlation analyses becomes easier and may be helpful for a more stable operation of RIBF. Furthermore, if the unified system can be used not only to display retrieved data but also to monitor real-time data, we can immediately fix an observed instability.

For the reasons indicated above, we decided to construct a new data archive system. This new system must meet the following requirements: acquire all parameters essential for the stable operation of RIBF accelerators at the required sampling rates, retrieve stored data and monitor real-time data.

DEVELOPMENT OF NEW DATA ARCHIVE SYSTEM

Outline

Since October 2009, we have been developing a new data archive system, RIBF Control Data Archive System (RIBFCAS), which covers both EPICS-based and non-EPICS-based systems. An outline of RIBFCAS is as follows:

1) Adoption of an open-source database was considered preferable.
2) For the EPICS-based system, a Java-encoded data acquisition program was newly developed to communicate with IOCs.
3) For the non-EPICS-based system, RIBFCAS works together with MyDAQ2 instead of developing new interfaces with the standalone control systems.
4) The client program has an ability to show both past and real-time data.
5) We plan to retain stored data for a minimum of five years. Data from up to two years previously can be searched immediately from client PCs at all times; however, older data are retrieved from a backup.

The RIBFCAS data acquisition system has to be capable of collecting more than 3000 data values at 1–60 s intervals, depending on the device type. System performance is highly reliant on the choice of database. Accounting for query performance, especially for large tables, we concluded that PostgreSQL 8.4.1 satisfied our fundamental requirements, because it supports a basic table partitioning function.

To achieve 2), a CA library must be used in RIBFCAS data acquisition program. The performance of this CA library is also important when storing the large data discussed above. A number of CA libraries were tested in order to find the most suitable for our intended purpose, and the results are discussed in the proceeding section.

For 3), MyDAQ2 data are merged with the data from the EPICS-based system by using a client application. We have created a CGI program to retrieve data from MyDAQ2 and export them in extensible markup language (XML) format to the RIBFCAS client application.

In realizing 4), a client application was developed on Adobe AIR. The advantage of using this platform is that client applications can be executed in multiple Adobe AIR runtime environments regardless of the file format.

Performance Test of CA Libraries

We evaluated data acquisition speeds for three CA libraries: Java Channel Access (JCA), Java Channel Access Light Library (JCAL) and an adapter library for JCA application programming interface (API) (JCAL–JCA). JCA is supported by the EPICS collaboration [12], and JCAL and JCA–JCAL were developed by the J-PARC control group [13]. Performance of the libraries was examined by writing programs with each such that all data could be processed for three chosen IOC types, having approximately 200 parameters apiece. One process of the test programs attempted to obtain data every 1 s and a test was concluded by obtaining five successive failures or successes. The JCA program was implemented for a single thread, whereas the JCAL program was implemented for multiple threads, since JCAL was developed as a thread-safe library in order to improve the stability of JCA when used in multi-thread technology. We measured the data acquisition time during each test and the results are summarized in Table 1.

The JCAL and JCA–JCAL programs successfully processed all data within 1 s, where the average time to acquire a datum was approximately 2 ms. There are no significant differences between the performances of JCAL and JCA–JCAL. When compared with these two programs, a longer data acquisition time was found for the JCA program due to the introduction of a delay time in order to establish a safe connection between the program and an IOC. If the JCA program attempts to process data from any channel of an IOC following a JCA-API call to establish channel connection, the program receives signals indicating an unconnected status and returns a failure message. To avoid this scenario, it was necessary to wait for 40 ms after each connection was established. For the case of the CC/NET IOC, this delay was increased 100 ms. As a consequence, we selected JCAL as the CA library for RIBFCAS.

Table 1: Performance of EPICS Java Library

IOC	Number of Parameters	JCA			JCA–JCAL			JCAL		
		Execution Time (ms)	Average (ms)		Execution Time (ms)	Average (ms)		Execution Time (ms)	Average (ms)	
CC/NET	187	49830	43	Fail	5102	2	Pass	5035	2	Pass
ALIX	198	46986	42	Pass	5104	2	Pass	5042	2	Pass
VME	192	45540	42	Pass	5101	1	Pass	5037	2	Pass

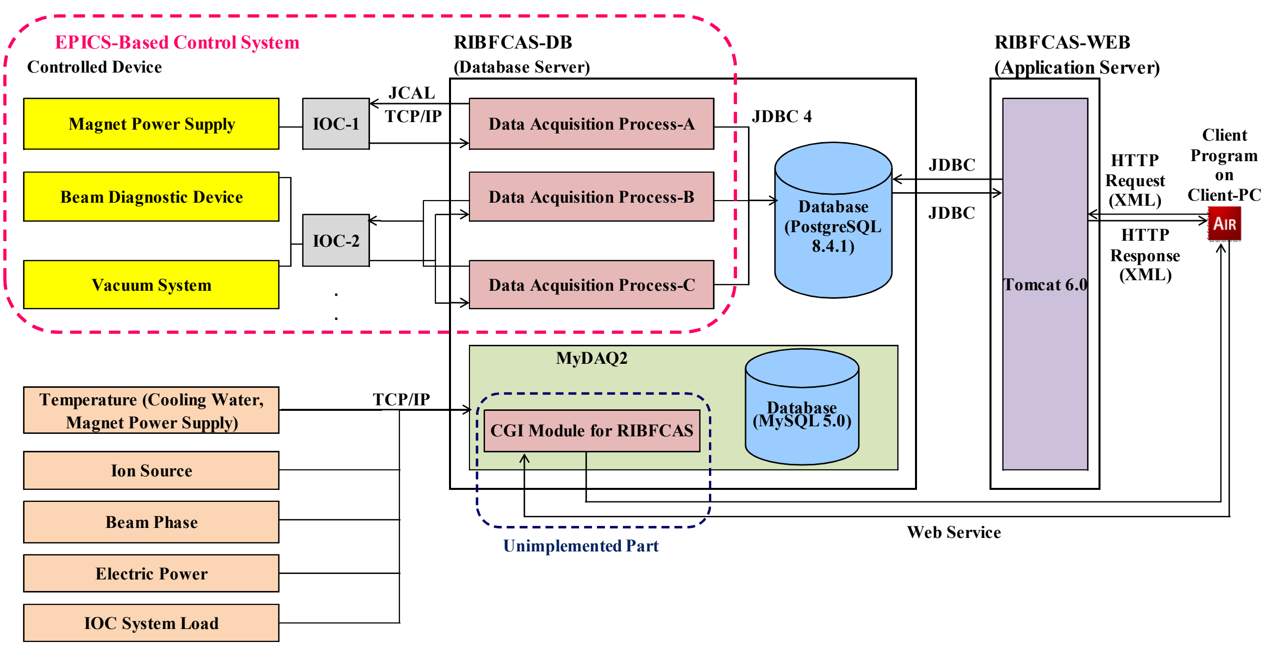

Figure 2: RIBFCAS Schematic Overview.

Structure of RIBFCAS

Figure 2 shows the structure of RIBFCAS. RIBFCAS consists of a database server, an application server, and client PCs. These are all connected to the EPICS local area network. The database server (RIBFCAS-DB) executes a JCAL-based data acquisition program to collect EPICS data. In addition, MyDAQ2 currently runs on RIBFCAS-DB and the MySQL database is also managed on this server. The application server (RIBFCAS-WEB) implements Tomcat version 6.0, which is a Web application server for the Java environment. Tomcat executes tasks in response to requests from client PCs, and returns retrieved data to client PCs in XML format. On client PCs, applications are executed on Adobe AIR runtime.

Table 2 shows the hardware specification of these servers and PCs. However, the performance of the existing database server is not sufficient to fulfill our requirements. Hence, we plan to upgrade the server.

The data acquisition program has the following features:

- The data collection program is divided into several processes and multi-thread technology is used to execute these processes simultaneously. This program structure gives the flexibility to divide existing processes into segments or to add new threads for devices introduced to RIBF accelerator complex in future upgrades.

- A list containing a large amount of information collected by RIBFCAS is managed within the XML framework in the program, making it easier to change device data.
- The system has an organized multilayer structure; components such as the database, data collection processes and data collection logic in the database server, Web service and client services are isolated from each other.

The main RIBFCAS system was completed in FY2010 and performance tests commenced in FY2011.

Table 2: Hardware Specification of RIBFCAS

	RIBFCAS-DB	RIBFCAS-WEB
CPU	Intel Xeon (Quad Core) 2.33 GHz	Intel Core2Duo 2.20 GHz
Memory	16 GB	1 GB
Hard disk drive	120 GB (RAID10, SAS)	80 GB
OS	CentOS 5.2 (32 bit)	CentOS 5.3 (64 bit)

OPERATIONAL STATUS

A trial for the EPICS-based section of RIBFCAS began on 26 April 2011, according to the operating schedule of RIBF. The program acquires values for approximately 3000 various component parameters of RIBF from 22 IOCs by using nine separate processes. During the test period, RIBFCAS has been storing data of the vacuum systems of cyclotrons and beam transport lines, the magnet power supplies and the beam diagnostic devices, once every 10 s. Although the present data sampling rate of 10 s is sufficient to monitor the stability of the systems, it is also possible to optimize the time interval for acquisition depending on device type.

Thus, we have succeeded in continuous data acquisition without significant problems occurring. The stored data has been constantly growing and reached approximately 76 GB on 13 July. During this period, unexpected system interruptions have transpired on several occasions, with the data collection process terminated due to limitations in memory usage of the 32-bit OS. The cause of this phenomenon might be unresolved small memory leaks in the data acquisition program. To avoid these interruptions resulting from memory usage limitations, we have adopted a practice of restarting the relevant processes routinely and automatically. Note that this phenomenon was not observed for CC/NET.

The client application has also been shown to work well. The client program successfully monitors real-time data and can also retrieve data for any period stored in the database. In both cases, a user can select any amount of any data for viewing. The client program displays data over a 24 h interval for a parameter within 10 s, and data over 1 h within 0.3 s. Even more, only 0.05 s is required to recall 24 h data of a parameter once it has been referred to. Monitored or retrieved data are displayed as a time plot in the client application and obtained plots can also be saved as text or JPEG files.

FUTURE PLAN

Although investigation of the source of the memory leakage in the data acquisition program must be pursued, RIBFCAS has started to collect stably data from the EPICS-based system. As a next step, we plan to develop an additional client application that processes the data stored by MyDAQ2. This is an important stage in unifying the two existing databases. Another problem to be solved is management of the vast amounts of data. As already stated, the EPICS-based section of RIBFCAS collects 76 GB of data every 2.5 months, which corresponds to 400 GB of stored data every year. Selection of appropriate hardware to handle such data is now in progress. We plan to complete the development of RIBFCAS in FY2011.

ACKNOWLEDGMENTS

We would thank to Mr. T. Etoh and Mr. Y. Otaka in Swadok Co., Ltd. for their coding the RIBFCAS programs.

REFERENCES

[1] N. Fukunishi, et al., “Operating Experience with the RIKEN Radioactive Isotope Beam Factory”, PAC09, Vancouver, Canada, May 2009, MO3GRI01, p.60 (2009)
[2] M. Komiyama, et al., “Upgrading the Control System of RIKEN RI Beam Factory for New Injector”, ICALEPCS2009, Kobe, Japan, October 2009, TUP084, p.275 (2009)
[3] http://www.tristate.ne.jp/picnic-e.htm
[4] http://wwwonline.kek.jp/~yasu/Conference/rt2003/CAMAC/RT2003-CAMAC.pdf
[5] http://www-online.kek.jp/~yasu/Parallel-CAMAC/
[6] http://www.yokogawa.co.jp/rtos/Products/rtos-prdcpu9-ja.htm
[7] M. Fujimaki, et al., RIKEN Accel. Prog. Rep., 37, p. 279 (2004)
[8] http://pcengines.ch/alix.htm
[9] N. Nakagawa et al., Rev. Sci. Instrum. 79 02A327 (2008)
[10] http://www.aps.anl.gov/epics/extensions/index.php
[11] T. Hirono, et al., “Development of Data Logging and Display System, MyDAQ2”, PCaPAC08, Ljubljana, Slovenia, October 2008, TUY01, p. 55 (2008)
[12] http://epics-jca.sourceforge.net/jca/
[13] H. Sako, et al., “Development of High-level Application Framework with a Script Language JCE for Accelerator Beam Commissioning”, ICALEPCS2009, Kobe, Japan, October 2009, THP091, p. 842 (2009)