DEVELOPMENT STATUS OF DATA ACQUISITION SYSTEM FOR LIPAC

H. Takahashi #, T. Kojima, K. Narita,
Japan Atomic Energy Agency (JAEA), Rokkasho, Aomori, Japan
H. Sakaki, JAEA, Kizugawa, Kyoto, Japan
S. Komukai, Gitec Co. Ltd, Hachinohe, Aomori, Japan

Abstract

The Control System for Linear IFMIF Prototype Accelerator (LIPAc) consists of six subsystems; Central Control System (CCS), Local Area Network (LAN), Personnel Protection System (PPS), Machine Protection System (MPS), Timing System (TS) and Local Control Systems (LCS). The LIPAc provides deuteron beam with beam power more than 1 MW, and this control system is required the high reliability and usability to perform various operation modes for a beam commissioning phase [1]. To realize the usability operation, we started developing of Data Acquisition System (DAC), which is one of the important functions of CCS.

In this article, the development status of DAC for the LIPAc is presented.

INTRODUCTION

The LIPAc consists of a part of sub-systems for the real IFMIF Accelerator (in the future), and the LIPAc purposes the engineering validations of these components. Therefore, for the validations of each sub-system and the activity of the real IFMIF Accelerator design, it is very important to record the commissioning data of these systems.

The Control system of the LIPAc constructs the remote control, monitoring and data acquisition by using EPICS. To archive the LIPAc operation's data, we started developing Data Acquisition System based on Relational Database (RDB). The first design for the DAC of the LIPAc control system is configured, I) to use PostgreSQL for RDB and II) to improve a processing data rate, two or more RDBs are installed. In addition, III) an independent RDB for data retrieval (user interface) is performed in the DAC. In this way, several RDBs for the DAC can behave only one RDB against users.

BACKGROUND

The LIPAc consists of a 100keV injector equipped with following 9 subsystems, that is an electron-cyclotron-resonance type ion source (ECR-IS), a low energy beam transport line (LEBT), 5MeV RFQ, the medium energy beam transport line (MEBT), 9MeV eight half-wave-resonator type superconducting linacs (SRF linac), a high energy beam transport line (HEBT), beam diagnostic system, a 1.2 MW CW beam-dump (BD), and RF subsystems [2]. These subsystems are developed and delivered by F4E (CEA, INFN, CIEMAT). And these will be installed at Rokkasho site in Japan from 2013, except SRF Linac that will be installed in summer 2016. At the beam commissioning, there are following 4 operation modes, "Injector (only)", "Injector + RFQ", "Injector + RFQ + SRF Linac" and "LIPAc (Injector to BD)", will be planned from the spring of 2013 to the middle of 2017.

On the operation's data acquisition since 2013, it should be considered in the management and the search because the data have important implications for the design study of the real IFMIF Accelerator.

Therefore, we started design to the DAC for the LIPAc from spring 2011, and also built the test bench to validate our design.

DATA ACQUISITION SYSTEM

Target Performance

In the LIPAc, we use the EPICS based CCS and LCS. Total number of EPICS record will be assumed to finally be about 10,000. Then, we designed to the target performance of the DAC is possible to acquire the 10,000 EPICS records per second in the first phase.

Requirement

However, the number of archived EPICS records will be growing as the commissioning is progress. Then, the DAC has to be constructed i) flexibility; it is able to respond appropriately to the growth of archived data. In addition, all operation data of the LIPAc is important to design the real IFMIF Accelerator in the future. So, data has to be managed with versatile-data-format which is not performed in only the special environment. Therefore, the DAC is also required ii) general versatility; it should utilize a standard-data-format and an environment in a database system.

Design

The DAC configuration is shown in Fig.1. To realize a requirement "ii) general", we decided the DAC is configured using PostgreSQL which is a common. In addition, to utilize a simple data format, one EPICS record data is inserted to one data table which is not perform the data compression and aggregation. In this way, it is possible for users to utilize the data inserted in RDB by the generalized "sql command".

By the reasons of ,

a) the DAC is not needed large scale because the LIPAc mainly consists of six subsystems,

takahashi.hiroki@jaea.go.jp

b) it is important to make the DAC management low cost. So, we decided to use PostgreSQL which is not a commercial RDB (free RDB).

Next, to realize "i) flexibility", we designed that the DAC is configured with several hardware (Slave PC) to collect data and it is easy to add the Slave PCs. However, this configuration makes a problem that users have to understand some information to search, for example, which EPICS record data is archived in which Slave PC. To resolve this problem, PC to work user interface for data search is added in the DAC. This interface PC (Master PC) has a function to behave like as all data is inserted in its own RDB. Using a Master PC, users need not understand the information of the DAC configuration, and it is possible for users to utilize the DAC as like only one RDB.

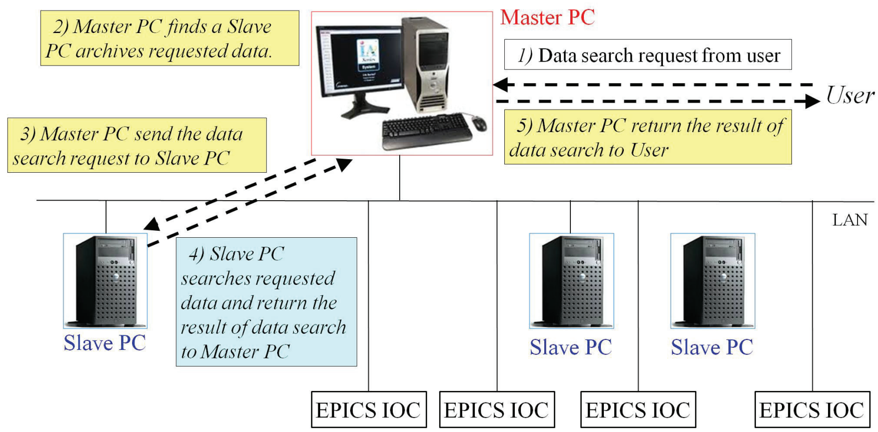

Figure 1: Configuration of Data Acquisition System.

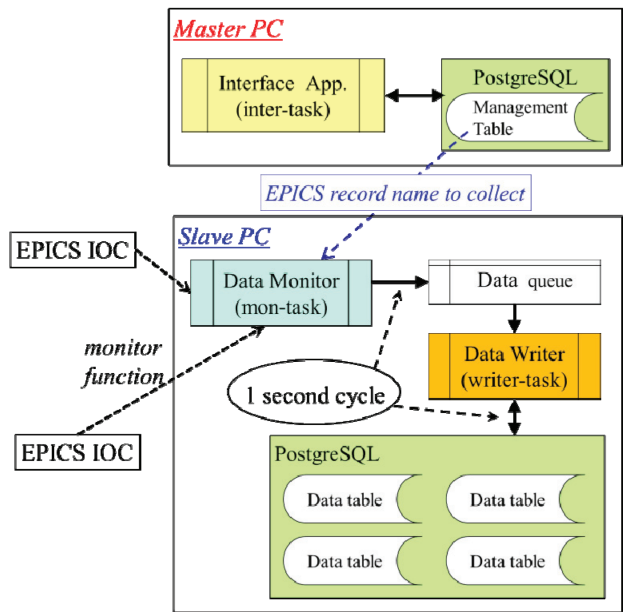

Figure 2: Software Configuration of DAC.

SOFTWARE

In the DAC, the archived data means measurement data (ai record with EPICS), status data (mbbi record with EPICS) and waveform data (waveform record with EPICS). At normal operation that is a steady state operation, "status data" is relatively constant data. However, in the transient state of subsystem, like the machine trouble happens, the subsystem starts up and shuts down, and so on. Therefore, it is important for the status data of time stamp and the transient value when the status changes. It is better that the status data is collected by using EPICS record monitor function, because the event is sent to a Slave PC only, likely the subsystem status changes. Thus, in case of the status data archive, the load of a Slave PC assumes to be high only transiently. On the other hand, the measurement data is needed to be archived not only the steady state, but also the transient state. The load of a Slave PC assumes to be constantly high and the amount of data becomes large.

The waveform data archive is difficult to estimate the load of a Slave PC. Because this archive greatly depends on the data size and the data archive cycle. Here, we design and study only the method for data archive.

Therefore, we design and test about the data acquisition of measurement data (ai record), mainly.

The Software configuration is shown in Fig. 2.

The Master PC

The Master PC works as the interface between users and Slave PC. This PC has a "Management Table" to manage the relation each EPICS record and each Slave PCs. This table is inserted in IP address of Slave PC, EPICS record name, start time of data archive and so on. The sample of master table is shown Table. 1.

The "Interface App." task (inter-task) works on the Master PC (shown in Fig. 2). When this task received the data search request (EPICS record name, start time and end time for searching) from user, this task gets IP

Table 1: Sample of Management Table in Master PC

recordname	recordtype	effective	starttime	IP address	dbname	schemaname	tablename
TEST:REC01	ai	0	2012/02/23 00:00:00	192.1.189.47	SlaveDB01	Schema01	table010223
TEST:REC02	ai	1	2012/02/24 00:00:00	192.1.189.47	SlaveDB01	Schema01	table02
TEST:REC03	ai	1	2012/02/24 00:00:00	192.1.189.47	SlaveDB01	Schema01	table03
TEST:REC04	ai	1	2012/02/24 00:00:00	192.1.189.47	SlaveDB01	Schema01	table04
TEST:REC05	ai	1	2012/02/24 00:00:00	192.1.189.47	SlaveDB01	Schema01	table05
TEST:REC01	ai	1	2012/02/24 00:00:00	192.1.189.47	SlaveDB01	Schema01	table010224

address of Slave PC which archives the requested EPICS record data from the master table. Next, this task sends "sql command" to search the requested data to a Slave PC. A Slave PC searches and returns the result of data search to the Master PC. Finally, inter-task sends the data from a Slave PC to user.

Slave PC

A Slave PC mainly works to collect data and archive data. In the test bench, "Data Monitor" task (mon-task) and "Data Writer" task (writer-task) run on this PC (shown in Fig. 2). The mon-task makes links between EPICS record, and receive the data from EPICS IOCs. The writer-task inserts the collected data to RDB.

The mon-task gets the EPICS's record names which this task should collect from the master table in the Master PC (this is interface PC for users), when this task starts up. Next, this task creates the tables to insert the collected data. The table is created every EPICS record. (If one table exists corresponding EPICS record, this task will not create a table.) The table structure of Slave PC is shown in Fig. 3.

Object Name		Creation Date	2012/02/24	Last Update Date	2012/02/24
Schema Name	AcceleDB	Composer		Authorizer	
System Name		DBMS	PostgreSQL		
Subsystem Name		Record Length			
Summary	Data Table on Slave PC (for ai record)				

Field Definition

No.	Field Name	Type	Length	NOT NULL	Default Value	Summary
1	starttime	timestamp	8			Start time
2	runtime	timestamp	8	NOT NULL		Run time
3	value	float8	8			Value
4	scantime	timestamp	8			Scan time

Index Definition

Index Name	Primary Key	Unique Key	Objective Field
(Object Name).pkey	TRUE	TRUE	runtime

Figure 3: Table Structure of Slave PC.

The mon-task received the measurement data (ai record) using EPICS monitor function with EPICS IOCs. And, this task buffers the received data by the event of monitor function and outputs the data queues to common memory every 1 second. In addition, at the measurement data, the cyclic data archive is necessary. However, the event is not output from IOCs when the measurement data does not change. Then, this task outputs the last data queue every 1 second if the event of one measurement data is not received. On the other hand, the writer-task inserts the data (data queues) on common memory to each data table on RDB every 1 second.

Here, basically, "Data Monitor" is able to be also utilized for data acquisition of "status data (mbbi)" and "waveform data (waveform)".

And, these data search on a Slave PC and the Master PC is performed using PostgreSQL standard functions.

TEST BENCH

The performances of data archive and data search were tested at the test bench of DAC. Test bench was configured one Master PC, three Slave PCs and some EPICS IOCs. The specifications of hardware for this test bench are shown in Table 2.

Table 2: Specifications of Master PC and Slave PC

Model	DELL PRECISION T5400
OS	Red Hat Enterprise Linux 5 Desktop (64bit)
Memory Size	7.8 GB
Swap Size	9.7 GB
PostgresSQL	Version 9.1

Result of a Slave PC Lifetime Test

First, we test a Slave PC which archives 3,000 ai records / 1sec. However, unfortunately the data queue was over flow and all data is not inserted to RDB after about 77 hours run. Next, the amount of data is changed from 3,000 to 2,000 ai records / 1 sec. In this parameter, the data queue was not over flow and a Slave PC continued to archive all data over 1 month.

Result of the Master PC Performance Test

The Master PC works like as one RDB. The Master PC is able to receive the request to search max 5 EPICS records. The Master PC sends requests to each target Slave PC, and receives each reply. Master PC successes to return these requested data to user as array data.

CONCLUSION

In this article, we confirmed that Slave PC have the performance to archive 2,000 ai record / 1sec. From this result, it is possible that the target performance of the DAC (10,000 records / 1sec) is achieved by the DAC is configured using 5 Slave PCs. It shows that this DAC has the performance corresponding to increase the archive data. And, in reaction to searching data from user, it is cleared that the DAC works like as one RDB with the Master PC. In this result, we shows the proposed the DAC has the enough performances of data archive and data search for the LIPAc,

At present, we have just started confirming that "Data Monitor" is able to be also utilized for data acquisition of "status data (mbbi)" and "waveform data (waveform)". Next, we plan the improvement of the Master PC and table format to archive the data for a long period. In addition, we will develop the CSS plug-in to display the archived data in this DAC.

REFERENCES

[1] H Takahashi et al., "Overview of the control system for the IFMIF/EVEDA Accelerator" Proceedings of the 6th Annual Meeting of Particle Accelerator Society of Japan, Tokai, Japan

[2] A. Mosnier et al., "The Accelerator Prototype of the IFMIF/EVEDA Project", Proceedings of IPAC'10, MOPEC056, Kyoto, Japan