Enhancement of Real-time EPICS IOC PV Management for the Data Archiving System

Jae-Ha Kim

Korea Multi-purpose Accelerator Complex, Korea Atomic Energy Research Institute, Gyeongju 38180, Korea

(Received 24 January 2015, in final form 10 June 2015)

The operation of a 100-MeV linear proton accelerator, the major driving values and experimental data need to be archived. According to the experimental conditions, different data are required. Functions that can add new data and delete data in real time need to be implemented. In an experimental physics and industrial control system (EPICS) input output controller (IOC), the value of process variables (PVs) are matched with the driving values and data. The PV values are archived in text file format by using the channel archiver. There is no need to create a database (DB) server, just a need for large hard disk. Through the web, the archived data can be loaded, and new PV values can be archived without stopping the archive engine. The details of the implementation of a data archiving system with channel archiver are presented, and some preliminary results are reported.

PACS numbers: 29.17.+W

Keywords: KOMAC, Channel archiver

DOI: 10.3938/jkps.67.1372

I. INTRODUCTION

In the KOrea Multi-purpose Accelerator Complex (KOMAC) for operating a 100 MeV linear proton accelerator, more than 4000 machine parameters and statuses are used. These parameters and statuses need to be recorded and evaluated in order to improve the linac operation of the KOMAC. According to the experimental conditions, new machine parameters need to be archived and some that are no longer needed should be deleted. For example, the RF power values are archived every 10 seconds. However, experimental data must be archived every a second. A reconfiguration of the archive engine takes at least 10 minute and is needed after each experiment. Functions that can add new data and delete data in real time without a reconfiguration need to be implemented. When new data must be archived in the existing data-archiving system configured by using a MySQL database and control system studio (CSS), the archive engine should be stopped to reconfigure [1]. Therefore, we need an additional data-archiving system which has functions that can add data and which can manage data through Web browser. The existing data-archiving system archives data which need to be archived over a long period of time. The data necessary for experiments are archived using an additional archiving system. The additional data archiving system is configured with the Experimental Physics and Industrial Control System (EPICS) base, extensions, archive viewer and channel archive on LINUX [2]. Through the channel archiver, new data are added on the web page. The Archive viewer, a user interface tool, can plot data and save data in text format if clients are connected to network.

II. DATA ARCHIVING

The EPICS mechanism was chosen to manage the KO-MAC proton linear accelerator. A data archive system based on EPICS is needed to archive the machine parameters and data. Therefore, the CSS, which is related with EPICS, was chosen for KOMAC. An archive engine, a CSS tool can take PV value from EPICS IOC and store data in a database. An archive engine is configured by using an XML file showing what to archive and how to do so. JDBC libraries for a MySQL database are included in the CSS archive system. A relational database (RDB) with MySQL was created for the KOMAC database. Figure 1 shows the logical data archiving system of KOMAC. The databases using MCCS are redundant. RDB1 is the main database where the archive engine usually writes data. RDB2 is synchronized with RDB1. RDB2 checks the heartbeat of RDB1 every second. If RDB1 causes an error, MCCS recognizes the error signal and causes an interrupt. RDB2 is to replace...
Enhancement of Real-time EPICS IOC PV Management

Fig. 1. (Color online) Software architecture of the KO-MAC EPICS data archive system.

Fig. 2. (Color online) Software architecture of the EPICS data archive system based on the channel archiver. RDB1. RDB2 takes all data from RDB1 and becomes the main database.

The additional data-archiving system is configured to the EPICS base, extensions, archive viewer and channel archiver. The archive engine in the channel archiver is configured with an XML file. The data are written to a hard disk in index and data-file formats. When the archive engine is running, the archive engine will create a built-in web server. Clients can check the engine’s status and configuration. On the “Config” page of the web server, data can be added without stopping the archive engine. Daily restart and monitoring-engine conditions are available using the Archive Daemon configuration. When by configuring the Archive Daemon and using a cron tab command the network data server is run, other machines can read and access the archived data by using the hypertext transfer protocol (HTTP) web server configuration [3]. The additional data archiving system is shown Fig. 3. Data can be manipulated, plotted and saved, for example, by using an archive viewer.

III. IMPLEMENTATION

We made a test environment that consisted of an EPICS IOC, a data archiving system and an EPICS client on a network for testing the data-archiving system. One hundred PVs matched with the data, were created in the EPICS IOC. EPICS base R3.14.10, extensions, the channel archiver, and the archive viewer were installed on Centos 6.3, a Linux operating system (OS). The archive engine was configured to store 100 PVs every second. Data are written and updated every day. The directory, named as the date of that day is created and data are archived in the directory by using Archive Daemon. Crontab command executes Archive Daemon every day at midnight. The HTTP web server was configured to recognize the data storage path and the archive data server in a common gateway interface (CGI) format. An archive viewer based on JAVA and JDK1.7 was installed. The web page of the archive data server was configured to run the archive viewer by clicking on the “Archive” text. Through the “Config” page of the engine web server, an additional dataset was added to test the additional functions.

The client PC needs to install a JRE or JDK to access archived data through the archive viewer. By typing in the internet protocol (IP) of the data archiving system, the webpage of the data archiving system is accessible.

IV. CONCLUSION

PVs are taken from the data, and the channel archiver scans the PV values every second with channel access.
Data are stored in a specified directory. The channel archiver archives data for three days. The data can be archived at 220 MB per day. If the number of PVs reaches about 4000, the archived data capacity will be 8.8 GB per day. A 1-TB hard disk can be used for about 120 days. The archive viewer is shown in Fig. 3. Through the archive viewer, we can load the required data, plot the data and export the data in a spreadsheet format and much more.

The new data-archiving system for KOMAC is shown in Fig. 4. The major parameters and data for a long observation period are stored in the data-archiving system based on CSS. Some parameters and data that are required during the experiment are recorded by the channel archiver, and can reduce the loss of data and manage the data more efficiently. Before the new data archiving system can be applied to the KOMAC data-archiving system, a central processing unit (CPU) load test with more than 4000 PVs is needed.

ACKNOWLEDGMENTS

This work is supported by the Ministry of Education, Science and Technology of the Korean Government.

REFERENCES

[1] MySQL reference manual. Oracle Corporation. URL: http://www.dev.mysql.com.
[2] Experimental Physics and Industrial Control System (EPICS). URL: http://www.aps.anl.gov/epics.
[3] Apache Server Foundation. URL: http://httpd.apache.org.