ISBN: 978-3-95450-237-0 ISSN: 2673-5512 doi:10.18429/JACoW-PCaPAC2022-FR021

EPICS IOC AND PVs INFORMATION MANAGEMENT SYSTEM FOR SHINE

Huihui Lv, Yingbing Yan†, Qingru Mi, Guanghua Chen
Shanghai Advanced Research Institute, Chinese Academy of Sciences, Shanghai, P.R. China

Abstract

For Shanghai HIgh repetition rate XFEL aNd Extreme light facility (SHINE), EPICS is adopted as standard to build the complex control system which involves hundreds of IOCs and millions of records. Most of IOCs run in industrial control computers as soft IOC. However, with the large amount of PVs, the need has emerged to develop remote tools to monitor all the channels and IOCs. One application backed by MySQL is designed, running periodically to interact with EPICS system where to take data from run-time databases via Channel Access and pvAccess. We embed scripting codes into the IOC startup script to realize that as soon as the IOC starts up, the information of the server's address, the IOC installation path, and all the records maintained by the IOC will be automatically pushed to the application and then stored in the database. With this necessary information, we could access all the active IOCs and PVs. Another web application is designed to render servers, IOCs and PVs data in MySQL on the web to give us an overall running status of the control system. We also fully consider the modularity and portability for the applications to apply and extend in none-SHINE environments.

INTRODUCTION

Motivated by the successful operation of X-ray FEL facilities worldwide and the great breakthroughs in atomic, molecular, and optical physics, condensed matter physics, matter in extreme conditions, chemistry and biology, the first hard X-ray FEL light source in China, the so called Shanghai HIgh repetition rate XFEL aNd Extreme light facility (SHINE), is under construction. SHINE will utilize a photocathode electron gun combined with the superconducting Linac to produce 8 GeV FEL quality electron beams up to 1 MHz repetition rate [1]. Hundreds of IOCs, located in Shaft #1, Shaft #2 and Shaft #3, will directly or indirectly control almost every equipment distributed in tunnels of the injector, the superconducting linac, and three FEL undulator lines, with the total length of approximately 3.1 kilometers long(as shown in Figure 1). Shaft #1 is a building at the end of the injector, Shaft #2 is at the end of linac, and Shaft #3 is at the end of undulator lines.

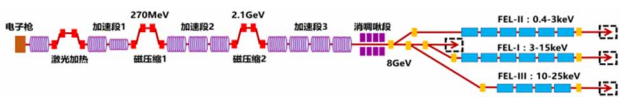

Figure 1: SHINE Accelerator Layout.

The control system involves heterogeneous equipment and interfaces with a number of different subsystems. Several IOCs will be deployed in one server, in which hundreds of PVs perform real-world I/O and local control tasks. But we suffer from a lack of high level tools to centrally manage all the IOCs and PVs and help us to verify which PVs are running on a specified IOC, and how many IOCs are running on a specific server. For an IOC, we also need to know which PVs are active. In this case, we develop the application to obtain all the PVs from EPICS system and store them in the MySQL database. In order to search through data easily, we also develop the web application to represent the information of the database. Details are described in the following sections.

ARCHITECTURE OVERVIEW

The data flow view of the architecture as shown in Figure 2 can be divided into four functional modules, namely database, import service, PV service and web service. The database is a general storage container for the properties and real-time values of servers, IOCs, EPICS records and PVs. The import service aims to import the initial information such as record types, field name lists and field types into the relational database from a spreadsheet. PV service fetches PVs from EPICS run-time databases via Channel Access and pvAccess. At the same time, it retrieves the running information of servers such as the available free memory, CPU load, etc. Then it saves all the data into the database. Web service is mainly used to render the data on the web to give us an overall running status of EPICS control system for SHINE.

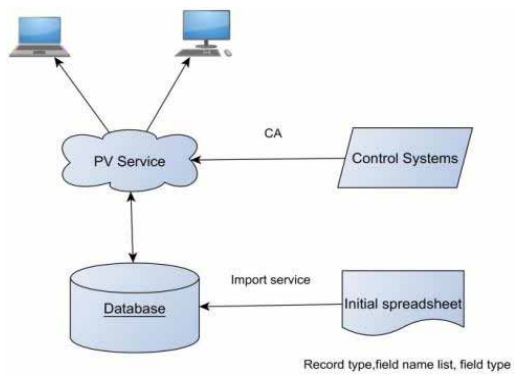

Figure 2: System Architecture.

Database Schema

The database saves the properties as Name/Value pairs. Details are illustrated in Figure 3. More specifically, it is needed to store the information of server, IOC, record and PV. For the server, the database is not only to store static attributes, like environment variables related to EPICS, but also to store the runtime information, for instance, when the server is started, when the server is scanned, CPU load,

yanyingbing@zjlab.org.cn

ISBN: 978-3-95450-237-0 ISSN: 2673-5512 doi:10.18429/JACoW-PCaPAC2022-FR021

etc. These are handled by tables of server, server_env, epics_env, and server_runtime, respectively. Generally speaking, there are several environment variables set up for the server to run EPICS system, such as EPICS_BASE, EPICS_EXTENSIONS, EPICS_HOST_ARCH, EPICS_CA_AUTO_ADDR_LIST,EPICS_CA_ADDR_LIST and so on. They are stored as Name/Value pairs as shown in table server_env. Similarly, others environment variables, such as EPICS_TIMEZONE, EPICS_CA_CLNT_CNT, EPICS_CA_CONN_TMO, EPICS_TS_NTP_INET, EPICS_VERSION, stored in epics_env table are also designed as Name/Value pairs. The running information, like available free memory, CPU load, buffers are archived with timestamp when the server is scanned.

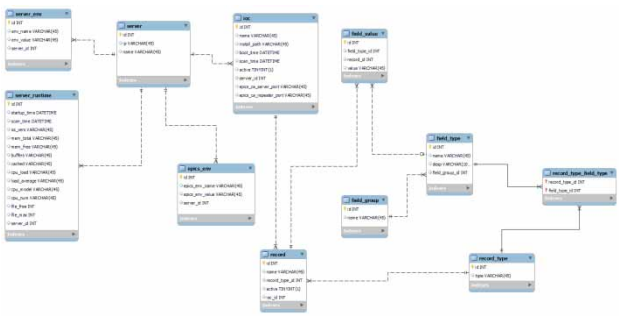

Figure 3: Database Schema.

The ioc table is to store the installation directory, startup time, active/inactive state and the UDP port to which an IOC is listening (5064, 5066...). What's more, a foreign key is inserted into the table that represents the server for One-to-Many relationship between them. The other six tables are used to store records and PVs. However, their relationship is a little complicated. A diversity of data from hardware are obtained by IOC as different record types, such as ai, bi, mbbi, mbbo and so on. Moreover, each record contains dozens of fields for different purposes, for example, INP field used to obtain the record's input, EGU and DESC used to present meaningful data to the operator. These data are stored in tables of record_type and field_type. Similarly, one field will be used in many record types. For instance, the DESC filed is utilized in multiple records. A many-to-many relationship occurs here, which is handled by a joining table that sits between the two tables. Its purpose is to store a record for each of the combinations of these two tables. It might seem like a bit of work to create, but it provides a much better data structure. A specific record which belongs to a typical record type involves a fixed set of field types. The values of fields, the so-called PVs are stored in field_value table. Field type/value is designed as Name/Value pairs for storing, that could simplify the overall complexity of the database schema.

Import Service

We create a spreadsheet with custom CSV layout to store the field types included by each type of record, that allows for easy import into the database. The tabs of ai and bi are shown in Figure 4. A single tab is initialed for each type of record. There are 15 record types we use frequently for SHINE, called ai, ao, bi, bo, longin, longout, mbbi, mbbo, stringin, stringout, waveform, subArray, compress, calc, calcout. Types vary from one facility to another.

Name	Description	group
VAL	Value Field	Read Write and Convert
INP		
DTYP		
LINR		
RVAL		
ROFF		
EGUF		
EGUL		
AOFF		
ASLO		
. . .		

Name	Description	group
OMSL	Output Mode Select	Desired Output
DOL	Desired Output Location (an Input Link)	Desired Output
OIF	Out Full or Incremental	Desired Output
VAL	Desired Output	Desired Output
OUT	Convert and Write	Read Write and Convert
DTYP	Device Type	Read Write and Convert
LINR	Type of Conversion	Read Write and Convert
RVAL	Raw Value	Read Write and Convert
ROFF	Raw Value Offset	Read Write and Convert
EGUF	Engineering Units Full	Read Write and Convert
. . .	. . .	. . .

Figure 4: CSV layout.

Import Service aims to import the data in the spreadsheet, as well as their relationships into MySQL database at one time. Apache POI [2] helps us to easily read and manipulate the excel files in our application by writing the Java code. Of course, duplicate records in the spreadsheet will be checked and consolidated into single records in the database.

PV Service

PV Service accesses EPICS Channel Access channels using ca library to get PVs asynchronously. Ca is a pure Java Channel Access client implementation developed by PSI [3]. As shown in Figure 5, PV Service is responsible to store PVs into the database as well. A channel is established for each of several fields involved in a record, and a channel group is formed by multiple channels, which communicates with IOC asynchronously. Meanwhile, SSH protocol is used to connect to IOC servers to execute shell commands for getting running status of the IOC server, such as CPU load and memory usage. JSch [4], as a pure Java implementation of SSH2, allows us to integrate these functions into our Java programs.

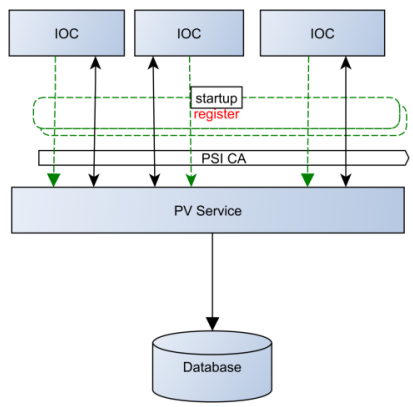

Figure 5: PV Service Architecture.

To be specific, the operation process of PV Service is divided into three stages: register, scanning and updating. First of all, necessary scripts (Figure 6) are embedded into the startup shell (st.cmd) in order to obtain the information of the IOC, such as the installation path, the startup time, the record lists, and so on. All the information is automatically written to a text file, which is then delivered to the

ISBN: 978-3-95450-237-0 ISSN: 2673-5512 doi:10.18429/JACoW-PCaPAC2022-FR021

server running PV Service, making use of the combination technologies of rsync+ssh+lsync.

Second, the program parses the text file for the information of the installation path, IP address, epics ca server port, epics ca repeater port and lists of the record for an IOC. With the IP address, we can access the server to get the memory usage information: mem total, mem free, buffers and cached, the CPU usage information: CPU model, CPU number, as well as the information of the file descriptors: file free, file max. The IOC related information is stored to ioc table, while the record related information is inserted into record table.

```
epicsEnvSet("PVINFOFILE", "/home/iocusr/pvinfo/${IOCNAME}.pvinfo")

system "echo IOCName:${IOCNAME} > ${PVINFOFILE} "
system "echo path:${TOP} >>    ${PVINFOFILE}"
system "echo -n Date: >> ${PVINFOFILE}"
date >> ${PVINFOFILE}

system "echo env_params: >>${PVINFOFILE}"
epicsPrtEnvParams >> ${PVINFOFILE}

system "echo record:    >>    ${PVINFOFILE}"
dbl >> ${PVINFOFILE}
```

Figure 6: Part of Scripts.

Finally, the Java scheduled task is set up to obtain PVs for all the fields in a record via CA protocol and store the data in the database. Therefore, the data are always in the latest state.

Web Service

The web service is built with JavaServer Faces [5] technology, together with PrimeFaces UI framework. It is deployed in Glassfish container and serves pages viewed in the web browser hierarchically. There are three panels illustrated in Figures 7, 8 and 9 to display the information of the server, ioc and record, respectively. When the web page appears, it shows a simple drop-down menu that allows you to select a server on the left side. Then the relevant information will be demonstrated on the right side, coupled with all the IOCs deployed in the server. The server's information includes the IP address, the startup time, OS version, CPU model, CPU cores, the available memory size, the unused memory size, CPU load, etc.

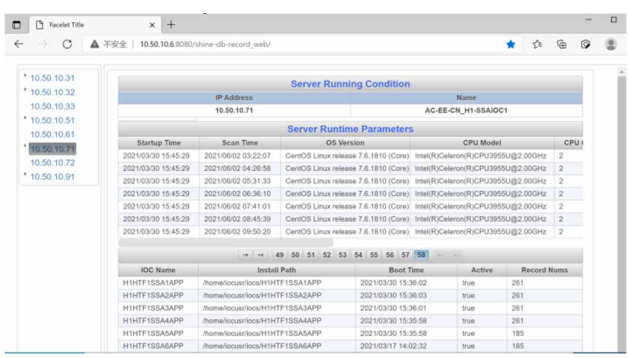

Figure 7: Display interface of the server.

The IOC panel as shown in Figure 8 will display not only the running information of the IOC, but also all the records included in the IOC. The information of the IOC includes the installation path, the startup time, epics ca server port, epics ca repeater port, running status, the number of connected and unconnected records. The record panel shown in Figure 9 displays all the fields' information.

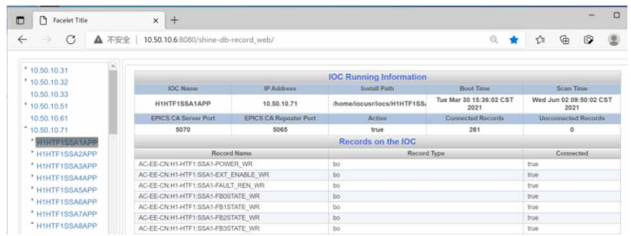

Figure 8: Display interface of the IOC.

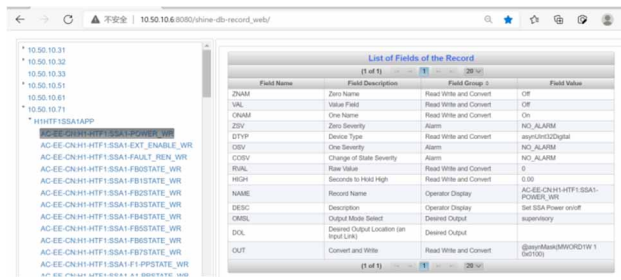

Figure 9: Display interface of the record.

CONCLUSION

We succeed in developing EPICS IOC and PVs information management system for SHINE. MySQL database is utilized to store all the information, of which the running data from the control system is fetched via PSI CA interfaces and the running states of servers are obtained via SSH protocol. The web service allows web pages to render for the information in MySQL database. The system has been operating stably for more than one year in the cryomodule test facility for SHINE [6]. With this system, we can easily and fully grasp the holistic control system operation.

REFERENCES

[1] Kai Li and Haixiao Deng, "Systematic design and three-dimensional simulation of X-ray FEL oscillator for Shanghai coherent light facility", *Nucl. Instrum. Methods*, vol. 895, pp. 40–47, 2018. doi:10.1016/j.nima.2018.03.072

[2] Apache POI, https://poi.apache.org/

[3] CA Library, https://github.com/channelaccess/ca

[4] JSch, http://www.jcraft.com/jsch/

[5] JSF, https://www.oracle.com/java/technologies/javaserverfaces.html

[6] H. Y. Wang, G. H. Chen, J. F. Chen, *et al.*, " Control System of Cryomodule Test Facilities for SHINE", in *Proc. ICALEPCS'21*, Shanghai, China, Oct. 2021, pp. 353-357. doi:10.18429/JACoW-ICALEPCS2021-TUBR04