NEW DATA ARCHIVE SYSTEM FOR SPES PROJECT BASED ON EPICS RDB ARCHIVER WITH PostgreSQL BACKEND

M. Montis, M. Giacchini, S. Fantinel, INFN/LNL, Legnaro, Italy
M. Bellato, INFN sez. Padova, Padova, Italy

Abstract

SPES project[1] is a ISOL facility under construction at INFN, Laboratori Nazionali di Legnaro, which requires the integration between the accelerator systems actually used and the new line composed by the primary beam and the ISOL target. EPICS[2] has been chosen as main Control System framework for the project; as consequence, a migration from the actual control system to the new one is mandatory in order to reuse the actual system for the new facility. One of the first implementation realized for this purpose is the Archiver System, an important service required for experiments. Comparing information and experiences provided by other Laboratories, an EPICS Archive System based on PostgreSQL is implemented to provide this service. Preliminary tests are done with a dedicated hardware and following the project requirements. After these tests, the system is going to be moved in production, where it will be integrated with the first subsystem upgraded to EPICS. Dedicated customizations are made to the application for providing a simple user experience in managing and interact with the archiver system.

INTRODUCTION

In a complex and extended installation like an accelerator, where different sub-systems work simultaneously and share information each other, the requirement of having available all the data under control, both online and offline, is mandatory: it results useful for production (such as post-analysis processes) and maintenance. The Channel Archiver is an archiving application realized for EPICS based control systems where a dedicated machine properly configured can archive any kind of process variable available in the control system network through the transparent communication protocol based on TCP/IP standard provided by EPICS, the EPICS Channel Access.

In the principal laboratory where EPICS is used as main control system framework, the original Channel Access Archiver, designed in 2006, is largely used. However in these few years different new solution based on Database backend are growing and starting to be a new standard for data archive service.

In this scenario, the EPICS RDB (Relational DataBase) Archiver with PostgreSQL Database has been chosen as main archive service for the SPES Project.

THE EPICS RDB ARCHIVER

During initial development, the test bench realized for the archiver system was composed by a single server equipped with all the hardware and the software required for a stand-alone test bench. In a next step, the hardware involved in the study case was upgraded and extended, in order to provide a full machine ready for production. The last set of tests has done using a part of the real control system environment under upgrade at LNL[3][4].

Hardware

The machine used to tests and deploy the archive service is a server equipped with 2 esa-core processor working in Hyper Treading, 32GB RAM and 2TB disk space, in order to have the maximum resources available both for development and production steps. It also has redundant power supply as required in this kind of environment.

Preliminary tests were performed on this host, using it as EPICS server (running an IOC) and EPICS client (the archiver application) due to analyse the correct configuration of the client tool. Later, extending tests to a real environment where different EPICS server provides different Process Variables (PVs), the hardware used for this represented in Figure 1.

Software

The focus followed in the software definition was having a machine with the minimum amount of applications and services required to perform this task, due to minimize the maintenance.

Following the guidelines adopted by LNL related to production hosts, RHEL compilant Linux was chosen as Operative System for the Archive machine. Over this OS, EPICS base was installed for providing the environment required to execute tests. At the same time, this machine was used to compile and develop the Archive tools (Archive Config Tool and Archive Engine) used to realize the final service; as consequence, a minimal graphic interface was installed for working with the Eclipse IDE and the EPICS CSS source code.

For the Archiver system, source code related to SNS CSS version 3.1.5 has been used. As suggested Ruizhe

ISBN 978-3-95450-146-5

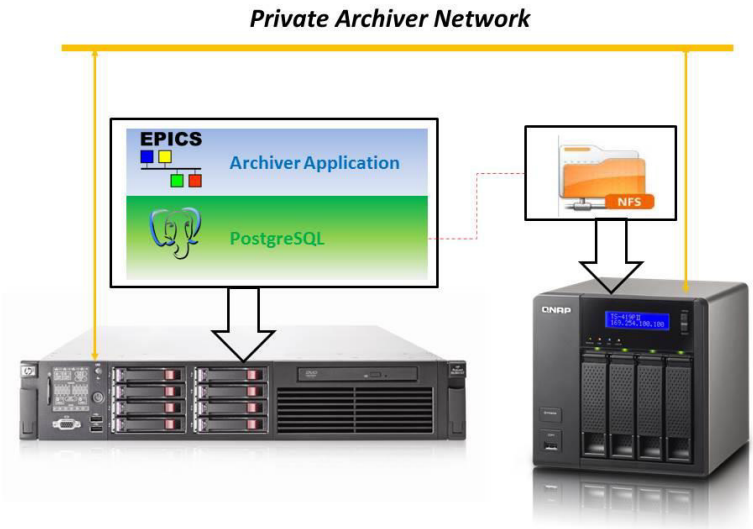

Figure 1: EPICS RDB Archiver schema: hardware and software involved to realize archive service.

Ma in his document[5], also source code stored in CSS Sourceforge repository[6] is required to compile properly part of the Archiver application. All the applications required have been compiled for Linux targets.

ARCHIVER APPLICATION AND CONFIGURATIONS

The EPICS RDB Archiver System results composed by different pieces (tools and applications) required to manage the communication between the PostgreSQL Database and the EPICS environment:

- *ArchiveConfigTool* let developer to automatize the Database configuration with all the information required to define a proper Archive Engine. For the SPES Project, different configuration XML files, used to describe Engines structure, are defined. Not particular configurations are required for this tool up to now.
- *ArchiveEngine* is the main application which defined the mechanisms of data storing and retrieving. For this application different Engine's profiles have been defined in order to customize and optimize host's resources. In this moment, all the experience acquired during the test phase was used to define the performance required for the ALPI-PIAVE diagnostics system. At the same time, because this system is still used as complex test bench, these configurations are not completed and frozen.

Because of the ArchiveEngine is a command line application which runs in foreground by default, a dedicated script was developed to realize a standalone service. This solution, merged with a dedicated network monitoring service, let system administrators supervise easily the status of the archiver system, minimizing the maintenance. In addition, for administrators and end users, a dedicated main page, available through Apache webserver running on the archiver host, provides all the minimal information and links related to the Archiver service.

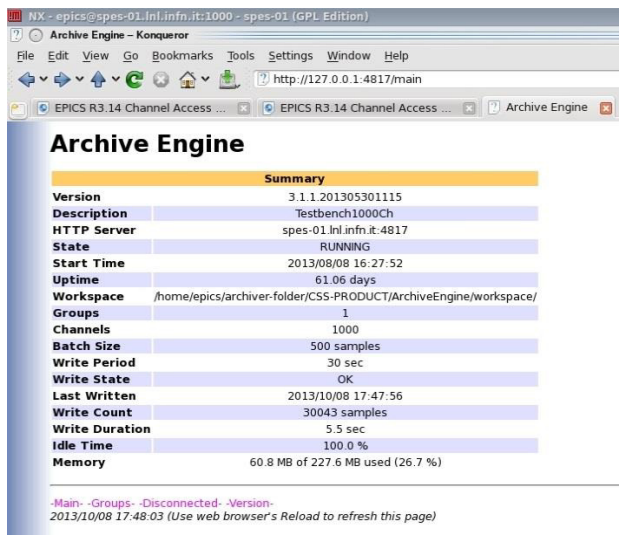

Figure 2: Archiver Engine used during tests: in this case a soft IOC running 1000 records with SCAN 1 second.

To start and stop archiving operations, a dedicated softIOC is prepared to provide the minimum set of records needed to manage the data storage (Figure 3): one binary record placed into the Control Panel related to a dedicated sub-system let user to enable or disable data acquisition, in order to save disk space when experiments are not executed. This binary input record controls several records placed in each Engine Group, starting and stopping them. This mechanism with double record (one main enable + one group enable) gives to users

ISBN 978-3-95450-146-5

(developers and maintainers) a high degree of freedom in customizing the archiver environment for particular operations.

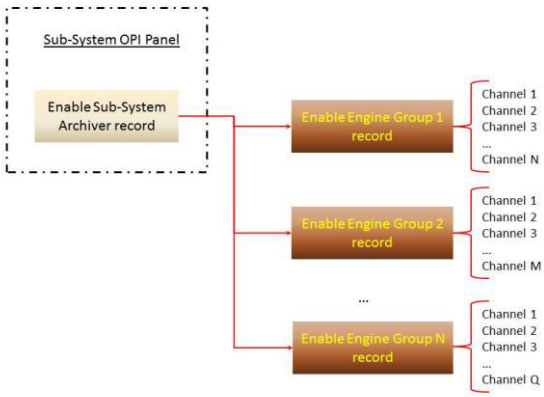

Figure 3: Database schema used to enable and disable data archiving without stopping the archive service.

TESTS PERFORMED

To prove the efficiency of the Archiver system, several tests have been performed using different configuration.

Table 1: Test Performed

Number of channels	Number of IOC	Parameters observed	Test period [months]
1000	1	• DB partition • Write/Read Time • Data loss	3
2000	1	• Write Buffer • Write Time • Data loss	1
310	9	• Write Buffer • Write Time • Network config • Time Synchronization	2 (still under test)

Analysing the results obtained with the first test and comparing them with the information provided by M. Konrad[7], a database partitioning based on weekly sub-tables is coherent with end user's request and doesn't compromise its complexity. At the same time, the high volume of data managed by the service doesn't affect it, providing a reasonable write time period: during the test with 2000 channels having SCAN=1s, the archiver set with:

- Write Period: 30 s
- Batch size: 60000 samples

writes the entire buffer in 11.4s, without data loss. Moving the tests to a real scenario, PostgreSQL Database's network interface was properly set to communicate only with a well-defined set of machines available in the control system network, in order to avoid unauthorized intrusion. The engine's configuration parameters found during first tests are still under test in this second part but, up to now, feedbacks confirms the behaviour expected. More detailed results will be available in the next period.

CONCLUSION

The archiver system is a mandatory service required for the SPES project and a brand new feature for the actual facility. A big amount of time was spent to study and analyse its performances and behaviour in order to find a good configuration (hardware and software) for the production phase.

First tests performed with the ALPI-PIAVE diagnostics system confirm the good approach adopted during the first stage and let developer to define a starting point for integrate all the remaining sub-systems composing the ALPI-PIAVE facility. At the same time, custom configurations will be improved for providing a better experience for system-administrators and end-users.

ACKNOWLEDGMENTS

This works leveraged of years of experience on EPICS use from good engineers of other laboratories around the world: great acknowledgments to them.

REFERENCES

[1] SPES Project website: web.infn.it/spes/
[2] EPICS website: http://www.aps.anl.gov/epics/
[3] B. Liu et al., "Upgrade of Beam Diagnostics System of ALPI-PIAVE Accelerator's Complex at LNL", WPO018, PCAPAC'14, Karlsruhe, Germany (2014)
[4] M. Giacchini et al., "Magnet Power Supply Control Mockup for the SPES Project", WPO016, PCAPAC'14, Karlsruhe, Germany (2014)
[5] Implementation and testing of RDB archiver with MySQL: http://www.illinoisacceleratorinstitute.org/2011%20Program/student _papers/Ruizhe _Ma pdf
[6] CSS Sourceforge website: http://sourceforge net/projects/cs-studio/
[7] M. Konrad et al., " Control System Studio Archiver with PostgreSQL Back-End: Optimizing Performance and Reliability for a Production Environment", WEPD03, PCAPAC'12, Kolkata, India

ISBN 978-3-95450-146-5