EPICS HYPERARCHIVER: INITIAL TESTS AT ESSBILBAO

M. del Campo*, ESS-Bilbao, Zamudio, Spain,
J.Jugo, University of the Basque Country, Leioa, Spain,
M. Giacchini, L. Giovannini, INFN/LNL, Legnaro, Italy

Abstract

The aim of this work is to present the results obtained after different tests performed regarding data storage for an Ion Source, by means of an EPICS control system at ESS-Bilbao (Spain). As a first approach, data was recorded on a MySQL database, using a traditional EPICS RDB Channel Archiver instance, maintained at ORNL SNS (USA). Nevertheless, initial results shown the need of an evolution towards a high performance scalable database. Therefore, current tests are focused on the customization and usage of a HyperArchiver instance, developed at INFN/LNL (Italy) as an extended version of the already mentioned RDB Channel Archiver, but using Hypertable as its main database. Hypertable is a distributed, high performance non relational database, released under GNU licence and focused on large-scale data-intensive tasks. At ESS-Bilbao, a slightly modified version of the HyperArchiver developed at INFN/LNL (Italy) was used, due to the necessity of an improvement on the management of array variables. Regarding data retrieval and visualization, a python GUI developed at ESS-Bilbao was used, in opposition to the traditional CSS data browser, trying to make data retrieval as fast and simple as possible. Hypertable is therefore presented as a high performance alternative to the broadly extended MySQL database for any EPICS control system.

INTRODUCTION

ITUR, the ESS-Bilbao's front-end test stand for ion sources [1] is the first operational system at ESS-Bilbao and, as a result, a tests stand for control systems, data acquisition, logging and analysis that will be required in the future at the linear accelerator. The Experimental Physics and Industrial Control System (EPICS) [2] has been chosen as control middleware, based on the experience of a large number of physics projects in North America, Europe and Asia. Following this lead, EPICS Channel Archiver tool was also selected to deal with data storage.

Since its original design, the EPICS Channel Archiver has undertaken several significant changes. The version publicly known as an EPICS extension [3] was developed at the Spallation Neutron Source [4] and its main feature was the usage of a relational database instead of indexed files, as the original Channel Archiver did. This version of Channel Archiver supports both MySQL and Oracle RDB and is entirely written in Java, using Java Database Connectivity (JDBC) for database access. This tool significantly improves data access and retrieval, in comparison to the original indexed file, but both Oracle and MySQL present drawbacks, like the expensive costs of Oracle or the fact that MySQL is not distributed, which can become a great limitation, considering that the RDB Channel Archiver stores all samples in one single sample table.

At ESS-Bilbao, RDB Archiver with MySQL implementation was used for the first tests with ITUR's control system, which is characterized by the use of several PVs of type array (RDB Archiver manages arrays considering every item of every array as a tuple of one single table at the MySQL database); but soon the size limitation became a problem. RDB Archiver with a distributed cluster of Oracle's database was immediately dismissed due to the costs of licenses. Therefore, the need of a distributed, high performance and large-scale dataset oriented database emerged. Different column-oriented databases were studied, such as Hypertable.

Hypertable appeared as a suitable alternative to MySQL, because it is a non relational, high performance and distributed database released under GNU license, which focuses on management of large data sets with maximum scalability and reliability. Furthermore, people from INFN/LNL in Legnaro (Italy) [6] had developed a customized version of RDB Archiver based on Hypertable [5] which was proved to have a very good performance on data insertion and retrieval [7]. It is called HyperArchiver and it is actually a hybrid version, as samples (bulk data) are stored in Hypertable database, but the basic static data of EPICS PVs is still recorded in RDB Archiver's MySQL database. Nevertheless, nothing changes from the user's point of view, because launching of both tools, EngineConfigImport and ArchiveEngine, stays the same, with identical calls and the whole communication with Hypertable and MySQL remains invisible to the final user.

SYSTEM DESCRIPTION

Several tests have been performed with HyperArchiver attending different IOC types, from very simple ones to more complex ones, by ESS-Bilbao Control & Diagnostics group at the University of Basque Country laboratory facilities (Leioa, Spain), with two main goals:

- Evaluate the main characteristics of HyperArchiver, like its installation complexity, performance and suitability on different Linux distributions and machine features.

* mcampo@essbibao.org

- Collect performance data in collaboration with INFN/LNL in Legnaro (Italy).

Once the EPICS tool proved to be reliable and suitable for ESS-Bilbao requirements, a replica of ITUR's control system was deployed at Leioa's laboratory facilities in order to simulate ITUR's behaviour and study HyperArchiver's performance. The whole system will be now described.

HARDWARE DESCRIPTION

The control system is based on a NI PXIe-1082 [9] running on LabVIEW Real Time and providing an EPICS server. It is accessed through a local area network to a Dell Precision T7400 tower workstation [8], with 2 Intel®Xeon®QuadCore processors e5430 @ 2.66GHz (8 core) and 8GB of RAM, with 6 different virtual machines deployed. One of them is the VMWare Linux virtual machine with 1 processor, 1GB RAM and 100GB of hard disk which was created to host the HyperArchiver instance.

SOFTWARE DESCRIPTION

An EPICS server is deployed by means of the LabVIEW Datalogging and Supervisory Control (DSC) Module [10]. Regarding data storage, both a MySQL 5.1.52 and a Hypertable 0.9.5.0 database servers are installed, in order to be used by HyperArchiver, everything on the already mentioned Dell workstation, inside a Fedora 14 32 bits. It is worth to mention, though, that the same system was tested on different Linux distributions, from Ubuntu 32 and 64 bits to Scientific Linux 6.0, always with satisfying results. Nevertheless, Fedora was chosen as the final distribution at the time the tests were carried out, to thoroughly reproduce the existent configuration at ITUR's control system.

HyperArchiver Customization

One of the main problems at ITUR's control and diagnostics system is the need to work with array PVs for waveform storage, which was something that had not been tested during HyperArchiver development at INFN/LNL (Italy), mostly focused on single PVs. Therefore, some development work was performed at ESS-Bilbao in order to:

- Upgrade from 0.9.3.3 to 0.9.5.0 Hypertable version (Hypertable changed to a general structure based on namespaces from version 0.9.4.3).
- Improve the management of large arrays and the way they are stored inside Hypertable.
- Provide the user the possibility to automatically create the EngineConfigImport configuration file directly from an EPICS' database file.

LabVIEW Interface

ESS-Bilbao's front-end test stand for ion sources (ITUR), located at Zamudio (Spain), has a control and diagnostic general control system based on LabVIEW. Therefore a simple interface to HyperArchiver was also developed in Labview, which allows not only to launch and stop the Archiver executables files (in order to import the engine configuration and to run the engine itself), but also to automatically check and modify the configuration files. This avoids the user a lot of tedious work, but most of all it makes very easy the integration of HyperArchiver in the general control system.

Data Visualization

As far as data visualization is regarded, a graphical user interface (GUI) was developed in python at ESS-Bilbao in order to have at disposal a simple data visualization tool, lighter than CSS and, as it was proved by the tests' results, faster than CSS Data Browser, which had already been used by people at INFN/LNL. It was also interesting for ESS-Bilbao to test and measure performances on python communication with Hypertable, as it could be used for future development tools.

RESULTS

Up to the moment, several tests have been carried out at ESS-Bilbao, using the new HyperArchiver tool and performance results were collected in both, data insertion and data retrieval, and analysed with the collaboration of INFN/LNL in Legnaro (Italy). The values here presented were obtained simulating a working period at ITUR, were 61 PVs were recorded every 1 second, in order to check the performance of Hypertable at the worst case in ITUR. Other tests were also carried out with sampling rates of 0.5 seconds and satisfying results. However, in order to keep it more realistic (not all PVs are scanned on a regular basis, but recorded only when they change; and not always the maximum scanning rate will be required), a maximum rate of 1 second was eventually selected.

Out of the 61 PVs, 7 were arrays: 1 of them made of 8K items and the other 6 of 4K items. That makes a total of 32054 samples/second, for the ArchiverEngine to deal with. Nevertheless, as far as performance measurements are regarded, array PVs are considered as single PVs, in order to measure the consumption of time accessing the database (each array PV is recorded as one tuple of the Hypertable database). Performance measurements were done after 1 hour, 5 hours, 1 day and 5 days running and the results are published in this document.

Data Insertion

Insertion rates might seem a little bit low, compared to the latest results from INFN/LNL, presented at the EPICS collaboration meeting in June 2010 by L. Giovannini and M. Giacchini [7], which arrived to 24 Ksamples/s at insertion. However, it is important to note that they are still competent values compared to Oracle or MySQL (always accordingly to INFN/LNL tests) working with regular PVs.

And, above all, HyperArchiver has proved to be the most reliable tool dealing with array PVs tested so far at ESS-Bilbao. Previous tests with MySQL and RDBArchiver had to be interrupted, because it didn't manage to deal with big array PVs. As it was previously mentioned, it is important to take into account the fact that each array PV is actually made of at least 4K items, which changes the actual meaning of the performance values.

Table 1: Data insertion

	Time	Volume	Throughput
1 hour	59.958 s	219.6 KPV	3.662 KPV/s
5 hours	6min 56.53 s	1.098 MPV	2.636 KPV/s
1 day	34min 52.31 s	5.270 MPV	2.041 KPV/s
5 days	218min 31.7 s	26.352 MPV	2.009 KPVs/s

Data Retrieval

Data retrieval was made with a python GUI developed at ESS-Bilbao. Performance data was obtained as an average of different retrievals, all of them of 3600 samples datasets (1 hour period). It might seem weird the fact that after 5 days working, data retrieval rates are better than on short periods. This can be explained by the intensive use of the computer at the beginning of the tests. Furthermore, it is in any case an indicator of the good performance that can be achieved with python and Hypertable, even when the database reaches several GB of data.

Table 2: Data retrieval with Python GUI

	Time	Volume	Throughput
1 hour	3.26 ms	3.6 KPV	1.104 MPV/s
5 hours	27.32 ms	3.6 KPV	131.758 KPV/s
1 day	44.45 ms	3.6 KPV	80.980 KPV/s
5 days	3.106 ms	3.6 KPV	1.159 MPV/s

CONCLUSIONS

Accordingly to the results, HyperArchiver has proved to be a very powerful tool, with high performance and reliability, overcoming data size limitations. In addition, the use of a simple python GUI instead of CSS data browser shown improvements on data retrieval times. From the collaboration with INFN/LNL (Italy), retrieval values with CSS were of 58 KSamples/s, which has been significantly improved by the results here published. Moreover, Hypertable is a distributed storage system, which allows scalable and robust solutions to be built on low cost hardware and very easily expanded, which is a very important feature for a growing facility like ESS-Bilbao.

Array management's performance was a key factor to be measured, as no tests had been made before by any application using HyperArchiver. The possibility to store waveforms was important for ITUR's control and diagnostics system, and therefore special attention was paid to HyperArchiver's behaviour which, together with some slight modifications, proved to be flexible enough to stand array PVs with no problems at all, keeping competent performance and reliability.

Likewise, the developed LabVIEW interface shown the lack of difficulty to integrate HyperArchiver (or any other instance of ArchiveEngine) inside a more general control system like the one at ITUR.

NEXT STEPS

By the time this paper was written, a replica of the HyperArchiver instance used for the tests was deployed and installed at the real ITUR, located in ESS-Bilbao's facilities at Zamudio (Spain), on a more powerful machine. Further and more extensive results will be gathered from its working periods.

In the meanwhile, it is also planned to analyse the impact of the system load (others processes running, machine features, etc.) on the performance of HyperArchiver (and therefore to explain the improvements on the results after 5days of execution), in order to define the minimal requirements for any future HyperArchiver installation.

The great amount of possibilities available from the use of Hypertable as main storage database will also be explored and exploited, from its distributed architecture to its multi-dimensional data model, trying to achieve the maximum performance possible.

REFERENCES

[1] I. Bustinduy, D. Fernandez, D. de Cos, J. Feuchtwanger, S. Lawrie, D.Faircloth, C. Plostinar, A. Lechtford, F.J. Bermejo, V. Etxebarria, J. Portilla, J. Jugo, J. Lucas, S. Jolly, M. Eguiraun, J. Alonso, R. Enparantza, M. Larranaga, "First simulation tests for the Bilbao accelerator ion source test stand", Proceedings of IPAC'10, Kyoto, japan, p.4211-4213

[2] http://www.aps.anl.gov/epics/.

[3] http://sourceforge.net/apps/trac/epicschanarch/wiki.

[4] http://ics- web.sns.ornl.gov/kasemir/archiver/.

[5] http://www.lnl.infn.it/ epics/joomla/archiver.html.

[6] http://www.lnl.infn.it/.

[7] http://www.lnl.infn.it/ epics/HyperArchiverAix2010.pdf.

[8] http://www.dell.com/us/dfb/p/precision-t7400/pd.

[9] http://sine.ni.com/nips/cds/view/p/lang/en/nid/207346.

[10] http://www.ni.com/labview/labviewdsc/.

[11] http://css.desy.de/content/index eng.html.