ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA181

WEB BASED VISUALIZATION TOOLS FOR EPICS EMBEDDED SYSTEMS: AN APPLICATION TO BELLE2

G. Tortone, A. Anastasio, V. Izzo, INFN Napoli, 80126 Naples, Italy
A. Aloisio[1], F. Di Capua[1], R. Giordano[1],
Dipartimento di Fisica Università di Napoli Federico II, 80126 Naples, Italy
F. Ameli, INFN Roma, 00100 Rome, Italy
P. Branchini, INFN Roma 3, 00100 Rome, Italy
[1]also at INFN Napoli, 80126 Naples, Italy

Abstract

Common EPICS visualization tools include standalone Graphical User Interface or archiving applications that are not suitable to create custom web dashboards from IOC published PVs. The solution proposed in this work is a data publishing architecture based on three open-source components: Collectd: a very popular data collection daemon with a specialized plugin developed to fetch EPICS PVs, InfluxDB: a Time Series DataBase (TSDB) that provides an high performance datastore written specifically for time series data, Grafana: a web application for time series analytics and visualization able to query data from different datasources. A description of this architecture will be provided showing flexibility and user friendliness of such developed solution. As a case study, we show the environment developed and deployed in the Belle2 experiment at KEK Laboratory (Tsukuba, Japan) to monitor data from the endcap calorimeter during the installation phase.

INTRODUCTION

EPICS process variables (PVs) represent a named set of data associated with an Input Output Controller (IOC) that publish status or measurements of a given instrument.

Users are interested to display PVs using user-friendly Graphical User Interfaces (GUI) in order to highlight critical trends and to handle alarm status.

Control System Studio (CSS) [1] is a very useful tool for EPICS PVs visualization. CSS allow users to easy design their own Operator Interface (OPI) using “widgets” that allow display of a single PV in different graphics format (text, gauge, bar, plot, ...).

A similar approach to EPICS PVs visualization is performed by Qt-based GUI systems [2] where users can choose to design an OPI using C++ classes or in “code-free” fashion by drag-and-drop a set of controls on a panel.

Another approach is to extend commercial scientific applications with EPICS Channel Access API in order to visualize and process PVs data. Matlab is widely used for this kind of tasks. With Matlab Channel Access plugin (MCA) [3] it is possible to retrieve, visualize and process EPICS PVs on a Matlab user designed GUI [4].

Although these solutions for EPICS PVs visualization represent a viable compromise between software coding and “drag-and-drop” GUI design, they don't provide a solution to display a PVs dashboard using Internet protocols but rely on a desktop application. It is desirable for some use cases to apply the same vision of Internet of Things (IoT) to EPICS PVs in order to access to dashboard and control panels by a simple web browser on a desktop PC as well as on a smartphone or tablet device.

ARCHITECTURE

The proposed data publishing architecture is based on a common pattern of data collection. Each produced data fragment (PV) is collected by any EPICS node and sent by network to a central server that gathers all data and store them in a database. A web application runs on the central server and displays collected data.

In order to overcome either network policies enforced on incoming traffic of each scientific institution and the frequently adoption of Network Address Translation (NAT) of local IP addresses, we have selected HTTP as transport protocol and organized the data flow to “push” data from inside a LAN to outside on a central server. With this option, the PVs published by an EPICS IOC running on a machine with a private IP network and Internet connectivity can be collected and displayed on an arbitrary platform, without any issue.

Data Collection

The data collection task is performed by Collectd [5] an open-source software. Collectd is a daemon which collects system and application performance metrics periodically and provides mechanism to store the values in a variety of ways.

Collectd community provides plugins to monitor system resources availability, performance and offers a wide set of interface to the most used database services (MySQL, Postgresql, RRD).

Plugins are classified as “input plugins” if they send values in Collectd logic or “output plugins” if they send values outside Collectd [Fig. 1].

A specialized input plugin [6] for Collectd has been developed in order to read EPICS PVs. This plugin uses Python API of Collectd and PCASpy [7] library for EPICS Channel Access.

The “write_http” output plugin of Collectd is used to push read EPICS PV towards a Python Tornado [8]

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA181

HTTPS server that handle the data flow to a central database.

Tornado is a Python web framework that uses non-blocking network I/O. It can scale to huge number of open connections making it ideal for applications that require a long-lived connection to each user.

A Tornado web application has been developed enabling some security enhancements like HTTP+SSL for communication encryption and client side (Collectd) authentication. Data gathered from Tornado web application are stored on a local database.

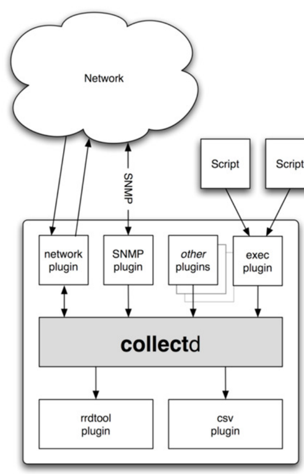

Figure 1: Collectd data flow

Data Storage

EPICS PVs collected over time are time series that can be efficiently stored and queried in a Time Series DataBase (TSDB). TSDB is optimized to handle time series data as arrays of numbers indexed by time. Tables contained in TSDB are not linked as in Relational Data Base System (RDBMS) and for such a reason often TSDB is referring as a NoSQL [9], semi-structured or “non relational” database system.

InfluxDB [10] has been selected as preferred TSDB implementation for such data publishing architecture. Its main features include: native support for time-centric functions in SQL-like query language, data tagging to allow flexible data querying, native downsampling for historical time series with continuous queries, retention policies to efficiently auto-expire stale data.

Data Visualization

Grafana [11] dashboard framework has been selected as web visualization tool. It allow users to create, using web application on a common Internet browser, their own control dashboards defining plots that fetching data from InfluxDB data store. Grafana also provides a rapid deployment due to embedded web server and it allows different user roles with the definition of custom authentication and authorization policies.

Figure 2 shows the data publishing architecture and the complete data flow.

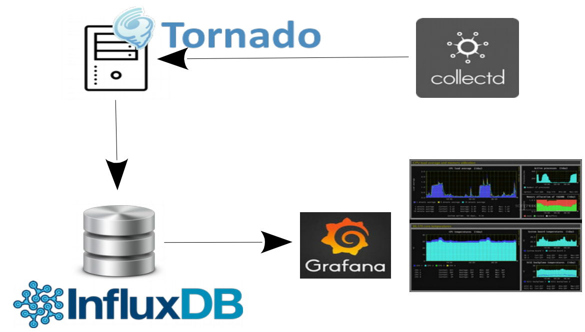

Figure 2: Data publishing architecture for EPICS PVs.

Testing and Use Case

The suggested data publishing architecture has been tested in the field in Belle2 experiment.

THE BELLE2 EXPERIMENT

The BelleII detector [12] is currently under construction at the SuperKEKB electron-positron collider at the KEK Laboratory (Tsukuba, Japan). As a major upgrade of the forerunner Belle experiment at the KEKB collider, the BelleII detector has been improved to make measurements of CP-violating asymmetries in rare B meson decays, to achieve precision determination of CKM parameters, and to perform sensitive searches for lepton flavour violation and lepton number violation in rare and forbidden B and D decays.

The BelleII Electromagnetic Calorimeter (ECL) is divided in a barrel and two anular endcap regions, named forward and backward following the asymmetric design of the low-energy (positron, 4 GeV) and high-energy (electron, 7 GeV) collider rings. CsI(Tl) was chosen as the scintillation crystal material in all regions, due to its high light output at an affordable cost. Light yield does change with temperature [13] and crystals are strongly hygroscopic and they can be severely damaged by humidity [14]. Forward and backward endcaps are made of 2112 CsI(Tl) crystals, arranged in 16+16 sectors. Each sector is equipped with three thermistors and an active humidity probe for a total of 128 analog channels.

Temperature and relative humidity in the two BelleII ECL endcaps are monitored by a uSOP-based network [15]. uSOP is a Single Board Computer (SBC) based on ARM processor and Linux operating system [16] that makes it possible to develop and deploy easily various control system frameworks (EPICS, Tango) supporting a variety of different buses (I2C, SPI, UART, JTAG), ADC, General Purpose and specialized digital IO.

A specialized EPICS IOC has been developed to interface uSOP with selected ADC through SPI bus. Temperature and humidity values are published on BelleII

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA181

EPICS network and, through Collectd daemon, are sent to a central server that gathers EPICS PVs.

Additional metrics are also collected on each uSOP board and pushed to the central server: CPU load, RAM and eMMC usage and network throughput.

Figure 3 shows the Grafana dashboard related to uSOP system metrics monitoring. Through a selection control, it is possible to select different uSOP boards and time period.

Figure 4 shows the dashboard of EPICS PVs gathered by central server. Multiple sectors of ECL endcap and different timeperiod can be selected in order to display historical plots of gathered metrics.

Every dashboard displays data in "live mode" and when a new sample is available in InfluxDB for a selected metric it is inserted in the plot without manually requesting a web page refresh.

Figure 3: uSOP dashboard for system metrics monitoring.

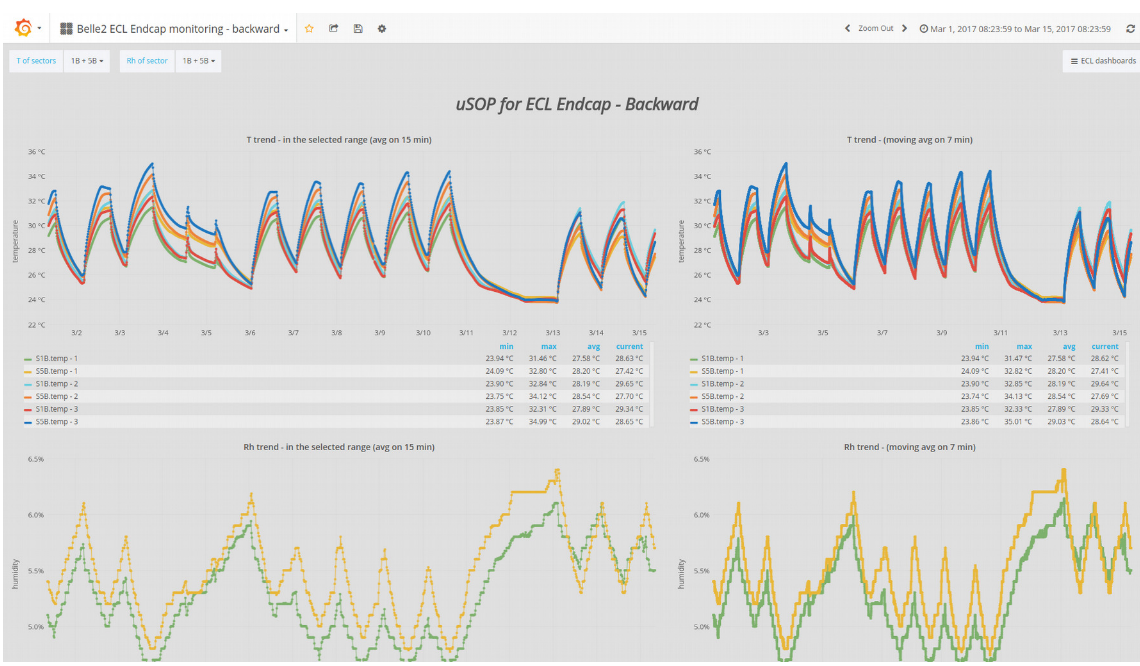

Figure 4: Belle2 ECL backward endcap monitoring.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA181

CONCLUSIONS

The data publishing architecture shown in this paper makes it possible to design interactively, using a web browser, custom web dashboards without knowing complex languages or configuration file formats. The adoption of high scalable HTTP web framework (Tornado) and high performance TSDB (InfluxDB) provides a framework suitable for concurrent usage from multiple users.

This architecture has been adopted for the monitoring of the ECL forward and backward endcaps during the installation phase and it has proven to be flexible and easy to use, maintain and adapt to different use cases. Once completed the installation, the monitoring software infrastructure will adopt as main visualization tool the Control System Studio (CSS).

REFERENCES

[1] K. Kasemir, "Control System Studio Applications", in *Proc. 11th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'07)*, Knoxville, USA, 2007, paper ROBP02, pp. 692-694.

[2] A.Rhyder *et al.*, "Qt based GUI system for EPICS control systems", in *Proc. 9th Int. Workshop on Personal Computer and Particle Accelerator Controls (PcaPAC2012)*, Kolkata, India, 2012, paper WECC03, pp. 10-11.

[3] MCA, Matlab Channel Access plugin - homepage, `http://ics-web1.sns.ornl.gov/~kasemir/mca`

[4] E.Tikhomolov, "Processing and visualization of EPICS data with Matlab applications", in *Proc. 11th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'07)*, Knoxville, USA, 2007, paper ROBP02, pp. 692-694.

[5] Collectd – the system statistics collection daemon, `http://www.collectd.org`

[6] Collectd plugin for EPICS PVs, `https://github.com/gtortone/b2-eclmon`

[7] Portable Channel Access Server in Python, `https://github.com/paulscherrerinstitute/pcaspy`

[8] Tornado Python framework, `http://www.tornadoweb.org`

[9] J. Han *et al.*, "Survey on NoSQL database." *in Proc. 6th International Conference on Pervasive Computing and Applications (ICPCA)*, Porth Elizabeth, South Africa, 2011, p. 363-366.

[10] InfluxDB, `http://www.influxdata.com`

[11] Grafana, `http://www.grafana.org`

[12] T. Abe *et al.*, "Belle II Technical Design Report", KEK, Tsukuba, Japan, Report 2010-1, Nov. 2010, `https://arxiv.org/abs/1011.0352`

[13] M. Kobayashi, P. Carlson, S. Berglund, "Temperature Dependence of CsI(TI) scintillation Yield For Cosmic Muons, 5 and 1.25 MeV y-Rays", *Nuclear Instruments and Methods*, vol. A281, pp. 192-196, 1989.

[14] P.Yang, C. D. Harmon, F. P. Doty, and J.A. Ohlhausen, "Effect of Humidity on Scintillation Performance in Na and Tl Activated CsI Crystals", *IEEE Transactions on Nuclear Science*, vol. 61, pp. 1024-1031, 2014.

[15] A.Aloisio *et al.*, "Monitoring complex detectors: the uSOP approach in the Belle II experiment", *Journal of Instrumentation*, vol. 12, 2017

[16] A. Aloisio *et al.*, "uSOP a microprocessor-based service -oriented platform for control and monitoring", *IEEE Transactions on Nuclear Science*, vol. 64, pp. 1185-1190, 2017.