ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-WEPV049

CONTROLS DATA ARCHIVING AT THE ISIS NEUTRON AND MUON SOURCE FOR IN-DEPTH ANALYSIS AND ML APPLICATIONS

I. D. Finch*, G. D. Howells, A. Saoulis, ISIS Neutron and Muon Source,
Rutherford Appleton Laboratory, Didcot, United Kingdom

Abstract

The ISIS Neutron and Muon Source accelerators are currently operated using Vsystem control software. Archiving of controls data is necessary for immediate fault finding, to facilitate analysis of long-term trends, and to provide training datasets for machine learning applications. While Vsystem has built-in logging and data archiving tools, in recent years we have greatly expanded the range and quantity of data archived using an open-source software stack including MQTT as a messaging system, Telegraf as a metrics collection agent, and the InfluxDB time-series database as a storage backend.

Now that ISIS has begun the transition from Vsystem to EPICS this software stack will need to be replaced or adapted. To explore the practicality of adaptation, a new Telegraf plugin allowing direct collection of EPICS data has been developed. We describe the current Vsystem-based controls data archiving solution in use at ISIS, future plans for EPICS, and our plans for the transition while maintaining continuity of data.

INTRODUCTION

The ISIS Neutron and Muon Source accelerators [1, 2] are currently controlled using the Vista Control System's software product Vsystem [3] (often colloquially called Vista), while the majority of beamlines and associated instruments [4, 5] are controlled using the Experimental Physics and Industrial Control System (EPICS) [6].

Vsystem and EPICS both originated in work done in the 1980s to create a control system for the Ground Test Accelerator Control System at Los Alamos National Laboratory [7]. While EPICS became an open-source project developed as a collaboration between multiple accelerator organisations, Vista is a closed-source commercial product with paid support. Both are distributed control systems with common features such as databases, and channels (Vsystem) or process variables (EPICS).

At ISIS Vsystem is deployed on four Itanium servers running the OpenVMS operating system. (Vsystem can be deployed on MS Windows, Linux, and OpenVMS, and operated in a hybrid configuration using any combination of these operating systems.) With the announced discontinuation of the Itanium processor architecture [8] a transition to a different processor architecture is required.

A decision has been made to transition the ISIS accelerators control system from Vsystem on OpenVMS / Itanium to EPICS on Linux / x86. This is planned as a gradual transition while ISIS is operating instead of an all-at-once conversion [9]. There will thus be a period in which both Vsystem and EPICS must operate in concert, and an early transition period in which EPICS operates only a small proportion of hardware.

This paper describes the logging functionality included with Vsystem, the software stack developed and deployed at ISIS to improve on this system, and the way in which it has been applied to machine learning and other applications at the facility. Software tools designed to allow EPICS to be used as a data source for this logging framework during the control system transition are then described.

VSYSTEM LOGGING AT ISIS

Vsystem Built-in Logging

Vsystem has an integrated logging subsystem called Vlogger [10] which consists of a Vlogger service that archives data from Vsystem databases to file, Vtrend for visualizing archived data, and utilities to manage and extract data from the generated archive files (including an SQL-like query language allowing CSV export).

At ISIS Vlogger is used to maintain short-term archives of data sampled every 30 seconds for 7 days, and longer-term data sampled every 30 minutes for 7 weeks. Both short- and longer-term archives are circular buffers. Permanent copies of the longer-term data are made at the end of each user cycle. The ISIS archive of this longer-term lower time resolution data begins in 2003, covered key channels by 2005, and subsequently steadily increased in scope.

The main limitations of this Vlogger system at ISIS are that:

- It must be run on a licensed Vsystem installation, with users therefore requiring access to and familiarity with our operations-critical OpenVMS server infrastructure.
- Channels to be logged must be specified in advance.
- It lacks a modern web-based query and visualisation interface.

For these and other reasons a parallel logging system for Vsystem was developed at ISIS.

InfluxDB Logging of Vsystem

Vsystem has an extensive and well-documented API [11] which can be used to access the Vsystem databases and their channels. Specifically, an event based callback API can be used to monitor for changes to live databases or channels.

Python code called vista_mqtt was developed to use the event based callback API to forward the changes in value and alarm states of all channels to an MQTT broker. MQTT is a publish/subscribe messaging standard and was chosen

* ivan.finch@stfc.ac.uk

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-WEPV049

for its simplicity and for its availability on OpenVMS. At the time the code was written the latest standard was 3.1.1 [12]; due to limitations in the client software there are no plans to move to MQTT 5. The MQTT broker selected was Eclipse Mosquitto [13], again chosen for its simplicity and ease of deployment. One important result of this architecture is that Mosquitto and all systems downstream (e.g. Telegraf, InfluxDB) of it may be run on the transition endpoint Linux operating system.

Vsystem as implemented at ISIS is primarily a pull-based architecture for reading, i.e. any interfaces to external systems and hardware are polled periodically. (Any writes to external systems or hardware are performed immediately.) Most polling is conducted at a 2 second cadence, though a small number of systems are polled faster or slower. The fastest cadence is 0.5 seconds for diagnostic profile and beam line monitors, some interlocks, etc. Monitored changes to values or alarm states are forwarded by vista_mqtt via MQTT immediately. In the case of reads from hardware this is at most at the polling frequency.

Telegraf [14, 15], currently run on the same Linux virtual machine as Mosquitto, subscribes to all MQTT topics published by vista_mqtt and writes the data to a single InfluxDB version 1.x time-series database [16, 17]. An upgrade to Influxdb version 2 is planned. The earliest retained data in this system dates to 2018 and the system was expanded to log changes in the value and alarm states of all single value Vsystem channels in 2019. This data is recorded at the full-time resolution. Data is interactively queried and visualised using the web-based Chronograf [18] or Grafana [19] tools.

The ISIS deployment of Vsystem currently manages more than 33,000 channels, of which approximately 1,500 may trigger alarm states. Together with the described update cadence this establishes a maximum data rate that the MQTT systems must support. In practice the data rate is significantly lower, as most values do not change at every polling interval. The current archiving rate is approximately 20 GB per user cycle (~70 days), and data rates are much lower during shutdowns.

The deployed Python-based monitoring code has proven reliable but has been found to be resource intensive on Itanium-servers which cannot be easily upgraded or expanded. Under high load this has caused system responsiveness issues and slowed down sampling of channel data which has affected the throughput of the logging system. For this reason an alternate C-based system is being developed to replicate the existing Python-based functionality, and to add monitoring of changes to other channel fields to facilitate the ongoing EPICS transition [20].

Other uses of vista_mqtt, Mosquitto, and InfluxDB

Vsystem's alarm viewer is called Valarm [21] and while reliable it is inflexible. It is a Motif-based X Window application which precludes being easily run on modern mobile devices, and as a closed source product it is difficult to integrate with locally developed software such as ISIS's FLD [22] fault diagnostics tool. Vista_mqtt publishes all changes to the alarm states of Vsystem channels and this has been used to develop a web-based alarm viewer called Rona which will address identified shortcomings.

The ability to set the value of Vsystem channel values and other fields via MQTT has been added to mqtt_vista, making it a general remote access interface to Vsystem. Although not originally designed for the purpose, this has allowed it to be leveraged to form an important component of the planned EPICS transition [9, 20].

Since the Mosquitto broker and InfluxDB time-series databases are now supported as part of normal ISIS operations, other internal groups have made use of these services. Some low-power RF functions in the synchrotron magnets are transferred via MQTT and integrated beam loss monitor and intensity data is stored in InfluxDB, though neither pass through Vsystem.

Support for Machine Learning

The relatively high time resolution logging of all inputs to the ISIS accelerators control system has been used to enable machine learning studies. One example is the real-time detection of temperature anomalies in the ISIS Target 1 methane moderator [23].

ISIS ACCELERATORS CONTROL SYSTEM TRANSITION TO EPICS

The planned transition from Vsystem to EPICS at the ISIS accelerators will be a gradual process, and initially EPICS will be a small subset of the accelerators control system. For this reason it is attractive to initially integrate EPICS into the existing logging infrastructure rather than to implement a "native" EPICS solution such as the EPICS Archiver Appliance [24]. Moving too soon to an EPICS based system would split data across two systems, making access and analysis more difficult.

In the medium-term an evaluation of EPICS logging systems will be conducted and, if necessary, a strategy will be developed to migrate our existing data holdings.

TELEGRAF AND EPICS

Telegraf is already used to aggregate and write the stream of Vsystem value changes from MQTT to InfluxDB. To support integrating EPICS into the existing logging system at ISIS, plugins have been developed for reading EPICS process variables (PVs) using Telegraf.

Telegraf is one of many metrics collection agents (others include Logstash, Metricbeat, statsd) used to collect metrics and telemetry from local and remote systems, process and aggregate that data, and then dispatch to other systems for storage or other purposes such as alert notification. Telegraf was originally selected in 2018 as it forms part of the TICK (Telegraf, InfluxDB, Chronograf, Kapacitor) stack [25] and InfluxDB had been chosen as our time-series data storage backend. In 2020 a more in-depth analysis of alternate metrics and telemetry frameworks was conducted, including a deployment of the full ELK (Elasticsearch, Logstash, Kibana) stack [26]. Our conclusion was that for our requirements there was little to

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-WEPV049

distinguish between the metric collection agents, and consequently little reason to switch from Telegraf given our existing expertise with the software.

Telegraf has a large number of input plugins (218 at the time of writing) used to collect data from a variety of sources and formats. EPICS is not currently among the supported data sources. There are two ways of adding the capability to handle a new source or format to Telegraf. The first is to develop a new plugin [27] for Telegraf in the Go language [28] in which Telegraf is written. The second, added in Telegraf 1.14 [29], is the use of the execd input plugin [30, 31] to collect data from an externally run program.

EPICS has two protocols used to transfer data. The Channel Access (CA) protocol [32] is the most widely deployed protocol, being supported in EPICS with revisions since at least v3.11 [33]. In EPICS v4 a new protocol called PVaccess (PVA) [34] was developed and since EPICS v7.0.1 both CA and PVA protocols have been supported by the main release series.

For the ISIS accelerator control system the intention is to use PVA where possible, and CA where necessary or convenient. This establishes a requirement to support both protocols with the existing logging software stack.

EPICS Telegraf Plugins

The CA Plugin for Telegraf developed at ISIS was based on the goca package [35]. The goca package uses Go's cgo system [36] (a binding to C libraries) to interface with the standard EPICS CA libraries. A minor modification to the package's handleEvent Go code was required to enable it to handle values of types other than double.

The CA Plugin for Telegraf is a simple development of the example Telegraf plugin and is only ~100 lines long. The channels to be monitored are simply defined in the Telegraf configuration file.

The cgo system used for the CA Plugin for Telegraf described above does not have a direct interface to C++ (or Java). There are therefore two methods available to develop a PVA plugin for Telegraf, either develop a supported interface in Go (e.g., via a shim to C) or develop a native Go implementation of the PVA protocol. Both would require significant development effort.

Instead the Telegraf execd input plugin [30] was used. This plugin allows an external program to be run as a long-running daemon and monitors the daemon's output on stdout. The execd input plugin will accept metrics in any of Telegraf's compatible input data formats [37], which Telegraf then aggregates and forwards to downstream applications.

A program called pvxs-archiver was developed in C++ using the pvxs module [38] for use with the Telegraf execd input plugin. It monitors a list of PVs listed in a configuration file and writes all changes to stdout in InfluxDB line protocol.

The CA Plugin and pvxs-archiver each handle a single EPICS protocol. Both are used to store the PVs they monitor in InfluxDB alongside any Vsystem data.

Docker and Telegraf Plugin Development

At ISIS Docker [39] containers are extensively used for development and testing [40]. Both EPICS Telegraf plugins were developed in similar Docker environments, utilising Docker Compose to instantiate containers with Telegraf running the plugin under development and InfluxDB acting as storage. Either a test IOC within a container environment or external test IOCs provided test input. This allowed a self-contained testing environment from IOC to storage backend, independent of any live systems.

Limitations

Telegraf is a metrics collection agent designed to accept timestamped metric values and timestamped log messages. It expects metrics values to be single-valued and has no direct support for more complex data structures such as 2D images. The current ISIS accelerators control system only needs to support single and 1-D array values. The latter is supported via conversion to strings which are stored in that format in InfluxDB.

CONCLUSIONS

Since 2018 the ISIS Neutron and Muon accelerators control system has greatly increased its capacity for logging Vsystem data, with a new software stack developed and deployed for this purpose. Amongst other uses the additional data collected has been used for machine learning, including detection of anomalies. As part of the transition to EPICS new plugins for Telegraf have been developed to allow designated EPICS PVs to be logged using this software stack.

REFERENCES

[1] *A Practical Guide to the ISIS Neutron and Muon Source*, Science and Technology Facilities Council, 2021, https://www.isis.stfc.ac.uk/Pages/A%20Practical%20Guide%20to%20the%20ISIS%20Neutron%20and%20Muon%20Source.pdf

[2] J. W. G. Thomason, "The ISIS Spallation Neutron and Muon Source—The first thirty-three years", *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 917, pp. 61-67, February 2019. https://doi.org/10.1016/j.nima.2018.11.129

[3] Vista Control Systems, Inc., http://www.vista-control.com

[4] F. A. Akeroyd *et al.*, "IBEX - an EPICS based control system for the ISIS pulsed neutron and muon source", in *Proc. 22nd meeting of the International Collaboration on Advanced Neutron Sources (ICANS XXII)*, Oxford, United Kingdom, Mar. 2019. doi :10.1088/1742-6596/1021/1/012019

[5] K. V. L. Baker *et al.*, "IBEX: Beamline Control at ISIS Pulsed Neutron and Muon Source", in *Proc. 17th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'19)*, New York, NY, USA, Aug. 2019. doi:10.18429/JACoW-ICALEPCS2019-MOCPL01

[6] EPICS Control System, https://epics-controls.org

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-WEPV049

[7] L. Dalesio, A. Johnson, and K.-U. Kasemir, "The EPICS Collaboration Turns 30," in *Proc. 17th International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'19)*, New York, NY, USA, Aug. 2020. doi:10.18429/JACoW-ICALEPCS2019-MOCPR02

[8] "Select Intel Itanium Processors and Intel Scalable Memory Buffer, PCN 116733-00, Product Discontinuance, End of Life", Intel, 30 January 2019. http://qdms.intel.com/dm/i.aspx/F65EEA26-13FB-4580-972B-46B75E0AB322/PCN116733-00.pdf

[9] I. D. Finch, "Evaluating Vista and EPICS with regard to future controls systems development at ISIS", in *Proc. 17th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'19)*, New York, NY, USA, Aug. 2019. doi: 10.18429/JACoW-ICALEPCS2019-MOPHA042

[10] *Vlogger (Vsystem User's Guide V4.3)*, Los Alamos, New Mexico, Vista Controls Systems Inc., Jan. 2014.

[11] *Vaccess Reference (Vsystem User's Guide V4.3)*, Los Alamos, New Mexico, Vista Control Systems, Inc., Jan. 2014.

[12] QTT Version 3.1.1, http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html

[13] Eclipse Mosquitto: MQTT broker, https://mosquitto.org

[14] Telegraf: metrics agent, https://www.influxdata.com/time-series-platform/telegraf/

[15] Telegraf 1.20 documentation, https://docs.influxdata.com/telegraf/v1.20/

[16] InfluxDB: time-series database, https://www.influxdata.com/products/influxdb/

[17] InfluxDB 1.8 documentation, https://docs.influxdata.com/influxdb/v1.8/

[18] Chronograf, https://www.influxdata.com/time-series-platform/chronograf/

[19] Grafana, https://grafana.com/

[20] K. R. L. Baker, I. D. Finch, G. D. Howells, A. Saoulis, and M. Romanovschi, "pvecho: design of a Vista/EPICS bridge for the ISIS control system transition", presented at 18th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'21), Shanghai, China, Oct. 2021, paper MOPV019, this conference.

[21] *Valarm (Vsystem User's Guide v4.3)*, Los Alamos, New Mexico, Vista Control Systems, Inc., Jan. 2014.

[22] A. Yaqoob, J. Brower, G. Owen, R. Titmarsh, and B. Mannix, "FLD 2.2 – First Line Diagnosis", presented at the *6th Accelerator Reliability Workshop (ARW2017)*, Versailles, France, Oct. 2017.

[23] A. Saoulis, K. R. L. Baker, M. Romanovschi, S. Lilley, and R. Burridge, "Machine learning for anomaly detection in continuous signals", presented at 18th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'21), Shanghai, China, Oct. 2021, paper FRBL01, this conference.

[24] EPICS Archiver Appliance, https://slacmshankar.github.io/epicsarchiver_docs

[25] Introduction to InfluxData's InfluxDB and TICK Stack, https://www.influxdata.com/blog/introduction-to-influxdatas-influxdb-and-tick-stack/

[26] Elastic Stack, https://www.elastic.co/what-is/elk-stack

[27] Input plugins [for Telegraf], https://github.com/influxdata/telegraf/blob/release-1.20/docs/INPUTS.md

[28] Go, https://golang.org

[29] New Plugins in v1.14 release notes, https://docs.influxdata.com/telegraf/v1.20/about_the_project/release-notes-changelog/#v114-2020-03-26

[30] execd input plugin [for Telegraf], https://github.com/influxdata/telegraf/blob/release-1.20/plugins/inputs/execd/README.md.

[31] External plugins [for Telegraf], https://docs.influxdata.com/telegraf/v1.20/external_plugins/

[32] Channel Access Protocol Specification v1.6, https://docs.epics-controls.org/en/latest/specs/ca_protocol.html

[33] EPICS Base Release v3.11, https://epics.anl.gov/base/R3-11.php

[34] pvAccess Protocol Specification, https://github.com/epics-base/pvAccessCPP/wiki/protocol

[35] M. Gibbs, goca: A Go interface to EPICS Channel Access, https://github.com/mattgibbs/goca

[36] cgo package documentation, https://pkg.go.dev/cmd/cgo

[37] Input Data Formats [for Telegraf], https://github.com/influxdata/telegraf/blob/master/docs/DATA_FORMATS_INPUT.md

[38] M. Davidsaver, PVXS, https://mdavidsaver.github.io/pvxs/overview.html

[39] Docker, https://www.docker.com/

[40] G. D. Howells and I. D. Finch, "Containerised Control Systems Development at ISIS and Potential Use in an EPICS System", presented at the 18th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'21), Shanghai, China, Oct. 2021, paper WEPV050, this conference.