ISBN: 978-3-95450-208-0 doi:10.18429/JACoW-IPAC2019-TUZZPLM3

THE EPICS SOFTWARE FRAMEWORK MOVES FROM CONTROLS TO PHYSICS

Greg White, Murali Shankar (SLAC, Menlo Park, California), Andrew Nicholas Johnson, Mark Lloyd Rivers, Guobao Shen, Sinisa Veseli (ANL, Argonne, Ilinois), Kunal Shroff (BNL, Upton, Long Island, New York), Matej Sekoranja (Cosylab, Ljubljana), Tom Cobb (DLS, Oxfordshire),Timo Korhonen (ESS, Lund), David Gareth Hickin (EuXFEL, Schenefeld), Heinz Junkes (FHI, Berlin), Martin Gregor Konrad (FRIB, East Lansing), Ralph Lange (ITER Organization, St. Paul lez Durance), Steven M. Hartman, Kay-Uwe Kasemir, Matthew Pearson, Klemen Vodopivec (ORNL, Oak Ridge, Tennessee), Leo Bob Dalesio, Michael Davidsaver (Osprey DCS LLC, Ocean City), Dirk Zimoch (PSI, Villigen PSI), Martin Richard Kraimer (Self Employment)

Abstract

The Experimental Physics and Industrial Control System (EPICS), is an open-source software framework for high-performance distributed control, and is at the heart of many of the world's large accelerators and telescopes. Recently, EPICS has undergone a major revision, with the aim of better computing supporting for the next generation of machines and analytical tools. Many new data types, such as matrices, tables, images, and statistical descriptions, plus users' own data types, now supplement the simple scalar and waveform types of the classic EPICS. New computational architectures for scientific computing have been added for high-performance data processing services and pipelining. Python and Java bindings have enabled powerful new user interfaces. The result has been that controls are now being integrated with modelling and simulation, machine learning, enterprise databases, and experiment DAQs. We introduce this new EPICS (version 7) from the perspective of accelerator physics, and review early adoption cases in accelerators around the world.

EPICS BASE AND EPICS VERSION 7

The Experimental Physics and Industrial Control System (EPICS), is an open-source software framework for high-performance Supervisory Control and Data Acquisition (SCADA). It has been co-developed, and used, over three decades, by a large international collaboration of accelerator laboratories, telescopes, and scientific enterprises.

Recently, due to the evolving needs for fast online data analysis, accelerator performance tuning, Machine Learning, and detector and experiment optimization, a significant upgrade of EPICS, version 7, has been developed.

The base software of EPICS is, and remains, software for supervisory control, closed loop feedback, archiving, alarm management, timing, and other aspects of front-end processors and device facing hardware. Often hosted in some commodity Linux or Windows PC, or sometimes in an embedded system processor such as RTEMS or vxWork, this software and its host in an EPICS control system, are collectively known as the IOC (Input / Output Controller). IOCs are optimized for low-latency I/O. They control and/or monitor a collection of devices such as actuators (magnets, klystrons etc) and measurement diagnostics. Each IOC node contains a memory resident real-time database.

The IOC database is a set of "smart" records, which are interconnected in a *data flow* pattern. They're smart in that their field values may come directly from hardware, or a result of processing that was dependent on the type of record. The records may contain "device support" code, to interface the processing to physical devices through device drivers. IOC code is optimized for throughput. Much more information can be found on the EPICS base at [1].

This so-called “EPICS base” software, and the software extensions built on top of it such as user level display managers, archiving and logging systems, detector frameworks etc, have proven very successful for the control aspects of scientific instruments. It provides excellent low level I/O, DAQ, optimal control, and user interfaces for many accelerators.

```
$ eget QUAD:LI24:900:TWISS
    energy 5.00512
    psix 37.7625
    alphax 13.6562
    betax -2.78671
    etax -0.00698294
    etaxp 0.00107115
    psiy 31.9488
    alphay 116.762
    betay 5.2592
    etay 0
    etayp 0
    z 2438.72

$ eget -s XCOR:IN20:491:RMAT -a b BPMS:IN20:525
0.669544 0.694654 0        0        0        0
-0.57085 0.901274 0        0        0        0
0        0        1.333434 0.966896 0        0
0        0        0.358411 1.009578 0        0
0        0        0        0        1        0
0        0        0        0        0        1
```

Figure 1: The Courant-Snyder parameters of a given quadrupole (a structure of named fields), and the transfer matrix (a to b) between a corrector and a beam position diagnostic (a PV subject to arguments), illustrate two examples of physics oriented quantities. Examples from SLAC.

ISBN: 978-3-95450-208-0 doi:10.18429/JACoW-IPAC2019-TUZZPLM3

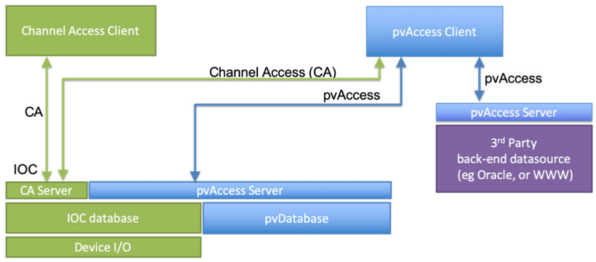

Figure 2: The basic architecture of EPICS version 7, showing the classic "base" components of EPICS in green, and components added by version 4 in blue. A new protocol, pvAccess, transports potentially complex data types encoded by pvData. pvAccess connects clients to IOCs, which may include a new database type, pvDatabase to support complex data processing services (and is the basis of the high-performance detector streaming applications described here), and to middleware services that can themselves connect to enterprise data stores, the web, etc. Note that hardware device I/O remains under the IOC database.

EPICS Version 7 for New Controls Problems

Modern control systems of large instruments call for more science to be done in the control system itself, than is possible with the EPICS base software alone. High output detectors call for pipelined data processing; physics applications deal with systems of process variables and their values rather than one process variable (PV) at a time; process variables may be statistically aggregated, structured, tabular, or multi-modal. Recently, key work in accelerator tuning and prognostics by Machine Learning or multi-parametric regression analysis, require acquisition and collation of "big" control system data, and significant metadata. For instance, experiment data must be identified with a given accelerator bunch, images with a codec, or timing related metadata to give context when reviewing archives.

EPICS 7 addresses these emerging requirements.

Core Modules of EPICS Version 7

Apart from EPICS base, the two core modules of EPICS version 7 are "pvData" and "pvAccess."

The pvData module is the high-performance structured data backbone of EPICS version 7. It enables the dynamic creation and management of structured data types and arrays. Although types can be created on the fly, EPICS version 7 defines a standard set of data types oriented towards scientific controls data (called the EPICS "Normative Types" [2]). The types there defined include matrices, key-value sets, tables, histograms, continua, and images, among others both generic and application specific.

pvAccess is the network protocol of EPICS 7. In addition to the expected get, put, and notification of change (so-called "monitor") methods one would expect, pvAccess supplies Remote Procedure Call (RPC) methods, both synchronous and asynchronous, where the PV value is computed with respect to user supplied arguments (see Figure 1). PvAccess and pvData include memory management and efficient encoding and deserialization, to minimise copy and wire transactions for even large complex data I/O. For instance, in the case of a PV defined by a complex structure, a client can subscribe only to the fields of the structure of interest to that client [3].

Both BNL and ORNL have thoroughly investigated pvAccess network performance. Their findings are that pvAccess' network efficiency is constant, well-behaved, and capable of delivering at or near 95% of the nominal maximum bit rate on 1 Gb/s or 10 Gb/s Ethernet, with no CPU saturation [4].

A number of EPICS "Display Managers" now, or will imminently, support EPICS 7. CS-Studio has done so for some time. PyDM and EPICS Qt both have support, including the Normative Types, in beta.

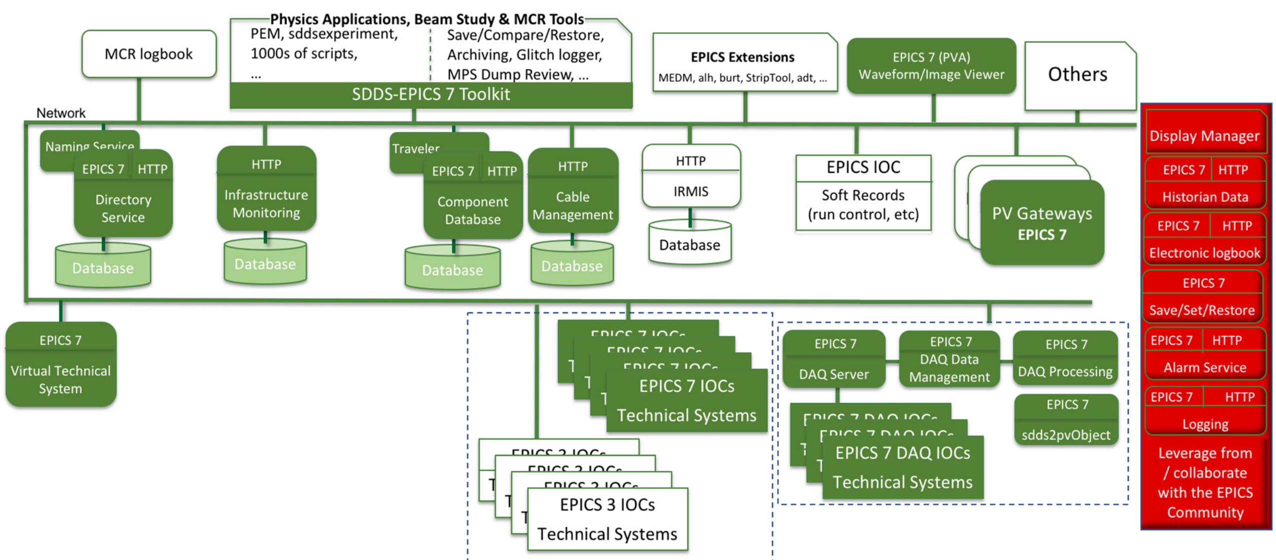

Figure 3: The control architecture plan of APS, showing the typical usage model enabled by EPICS now - a three tier system of front end computers (IOCs), middleware services, and intelligent clients for both display and machine analysis. Notably at APS, EPICS 7 is used to publish the results of the SDDS physics analyses package on the network.

ISBN: 978-3-95450-208-0 doi:10.18429/JACoW-IPAC2019-TUZZPLM3

New Data Processing Database and Usage

The new "pvDatabase" module of EPICS 7 implements a framework for a memory resident database of records defined in terms of pvData structures (Fig. 2). Like the IOC database of classic EPICS, the records of pvDatabase can process on I/O events; unlike the IOC the records may be of any structure the engineer wishes, and may pull in data from any pvAccess-ible data source, plus Channel Access.

areaDetector is an important extension to EPICS for 2-D detectors. These are an essential component of accelerator systems, where they are used both for particle and x-ray beam diagnostics, and for x-ray beamline experiments at storage rings and FELs. areaDetector is a C++ framework for controlling such detectors in an EPICS control system. areaDetector integrates directly to the EPICS 7 normative type NTNDArray. This is a major improvement over the previous EPICS Channel Access implementation, since the NTNDArray contains all of the metadata such as image dimensions, data type, timestamp, and user-defined attributes as a single atomic object on the network. A recently developed multi-threaded plugin (NDPluginCodec) compresses image data using Blosc, LZ4, Bitshuffle/LZ4 or JPEG codecs. The consequent NTNDArray network objects served by pvAccess are also compressed in this scenario, which can dramatically reduce the required network bandwidth between the pvAccess client and server. Frame rate speedup > x10 is expected when coupled with HDF5 direct chunk write.

RECENT WORK

The pvAccess server in an IOC (qSrv) can now aggregate a number of IOC records into a single pvAccess PV. For example, DESY utilized this to acquire all the cavity temperatures of a cryo module as one PV.

A number of Python packages for EPICS 7 are now available. Both "pvaPy" and "p4p" packages interface Python to EPICS 7 data; p4p allows rapid development of EPICS 7 services too [5].

Pva2pva, being developed at SLAC and ESS, is an EPICS 7 "gateway" - essentially a smart application level network router for filtering and user authentication of control data over large networks.

A plug-in architecture for implementing security schemes has been added, along with the plugin for Channel Access security.

The Beam Synchronous Data Service at SLAC, collects and collates all bunch-by-bunch data across the whole LCLS machine, even at full rate (120 Hz), and stores it for offline analysis. Notably, the bunch metadata – e.g. cathode charge, beam switch requests, etc., are also stored. This enables Machine Learning and multi-parametric regression analysis on every bunch, everywhere in the accelerator. Performance tests for the 1 MHz LCLS upgrade are under way. The system uses the C++ service support in EPICS 7, publishing a table Normative Type PV, with a python p4p client daemon listening to the table PV outside the control network, and writing HDF5 files.

SELECTED DEPLOYMENTS

The primary uses of EPICS version 7 have so far been found to be in high-performance detector data fan out, and in middleware data services.

EPICS 7 is used extensively at SNS in data acquisition and processing of neutron scattering experiments [4]. Detector image pixel data from an experiment is combined with meta-data such as probe pulse id and charge, by a pvDatabase, into a pvData structure, which then fully describes an experiment observation event. The event data is then streamed by pvAccess to clients for processing and analysis. The primary client – effectively an EPICS 7 plugin for areaDetector – includes statistical processing and publication of selected fields as histograms by Channel Access, for consumption by existing GUI tools. 21 beamlines at SNS have so far been commissioned on this system. It has worked reliably giving 10M events/s (with throughput of 80 MB/s).

Diamond and BNL have developed an areaDetector pipeline based on EPICS version 7. "Malcolm," from Diamond, makes it easy to program the hardware of a synchrotron experiment's hardware using block structures and utilizes the multimodal nature of EPICS 7 PVs. The BNL framework is distributed with areaDetector.

APS similarly publishes physics analysis from SDDS (Self Describing Data Sets) over EPICS 7 (Fig. 3). SLAC is using EPICS version 7 in services for a host of middleware, including beam dynamics modelling, device infrastructure, magnet polynomials and archive data.

SLAC has also performed extensive and successful IOC field testing and is now rolling out 7 to all IOCs.

CONCLUSIONS

EPICS version 7 can be easily integrated into an EPICS installation to supplement existing device support, DCS and SCADA. It effectively extends EPICS to support structured scientific data and complex processing. Early users have found it particularly useful for high-throughput detector data. pipelined processing, and integration of beam dynamics and accelerator enterprise data, into the controls system.

REFERENCES

[1] EPICS (new) homepage, `https://epics-controls.org`

[2] G. White, L. Dalesio, M. Rivers, M. Kramer, D. Hickin, "EPICS Normative Types", `http://tinyurl.com/l3pypbc`

[3] M. Sekoranja *et al.*, "pvAccess Protocol Specification", `http://epics-pvdata.sourceforge.net/pvAccess_Protocol_Specification.html`

[4] K.-U. Kasemir, G. S. Guyotte, M. Pearson, "EPICS V4 Evaluation for SNS Neutron Data", in *Proc. ICALEPCS'15*, Melbourne, Australia, Oct. 2015, pp. 947-949. `doi:10.18429/JACoW-ICALEPCS2015-WEPGF105`

[5] M. Davidsaver, "pvAccess for Python (p4p)", `https://mdavidsaver.github.io/p4p/`