Recent Advancements and Deployments of EPICS Version 4

ICALEPCS 2015, Melbourne, Australia.

Timo Korhonen, for the EPICS Version 4 working group

Talk and paper prepared by Greg White
EPICS Version 4 Working Group

GitHub (source code management) https://github.com/epics-base/
Sourceforge (documentation, admin, downloads) http://epics-pvdata.sourceforge.net
Talk Outline

• Version 4 Additions to EPICS
• Deployments
• User Feedback and Conclusions
• Recent Work
Talk Outline

• Version 4 Additions to EPICS
• Deployments
• User Feedback and Conclusions
• Recent Work
EPICS Version 4 in a Nutshell

- New Protocol, “pvAccess”
- Structured data
- Introspection interface, “pvData”
- Dynamic typing
- Standard Scientific Types
- RPC and putGet added
- New smart database
- All APIs in C++ and Java
- Python and Matlab
- High Performance
- High Reliability

$ eget -s XCOR:LI24:900:TWISS
non-normative type
structure
double energy 5.00512
double psix 37.7625
double alphax 13.6562
double betax -2.78671
double etax -0.00698294
double etaxp 0.00107115
double psiy 31.9488
double alphay 116.762
double betay 5.2592
double etay 0
double etayp 0

Figure: pvAccess method “ eget”, which is for service data, getting PV of a structure of optics parameters. In this case a standard “Normative Type” type was not used, so the raw structure is displayed by eget
Version 4 Additions to EPICS

The EPICS V4 “Normative Types”


5. General Normative Types
   1. NTScalar
   2. NTScalarArray
   3. NTEnum
   4. NTMatrix
   5. NTURI
   6. NTNameValue
   7. NTTable
   8. NTAtribute

6. Specific Normative Types
   1. NTMultiChannel
   2. NTNDArray
   3. NTCentum
   4. NTHistogram
   5. NTAgregate

$ eget -s XCOR:LI24:900:RMAT
  0.0727485  0.0289316     0     0     0.0652488  0.00125391
  0.0578214  0.0391775     0     0   -0.027185  -0.000192344
  0  0  0.00943029  1.14291  0     0
  0  0   -0.0013367   -0.0348832  0     0
 -0.000370971 -0.000283933  0     0   -0.0182387  -0.000198345
  0.10031  0.018722    0     0   -10.5721   -0.179568

$ eget pva://mccas0.slac.stanford.edu:39633/QUAD:LTU1:880:RMAT?type=design

$ eget -s LCLS:ELEMENTS

<table>
<thead>
<tr>
<th>ELEMENT</th>
<th>ELEMENT_TYPE</th>
<th>EPICS_DEVICE_NAME</th>
<th>S_DISPLAY</th>
<th>OBSTRUCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATHODE</td>
<td>MAD</td>
<td>CATH:IN20:111</td>
<td>2014.7</td>
<td>N</td>
</tr>
<tr>
<td>SOLBK</td>
<td>MAD</td>
<td>SOLN:IN20:111</td>
<td>2014.7</td>
<td>N</td>
</tr>
<tr>
<td>CQ01</td>
<td>MAD</td>
<td>QUAD:IN20:121</td>
<td>2014.9</td>
<td>N</td>
</tr>
<tr>
<td>SOL1</td>
<td>MAD</td>
<td>SOLN:IN20:121</td>
<td>2014.9</td>
<td>N</td>
</tr>
<tr>
<td>XC00</td>
<td>MAD</td>
<td>XCOR:IN20:121</td>
<td>2014.9</td>
<td>N</td>
</tr>
</tbody>
</table>

Figure: An extract of the Table of Contents of the Normative Types Specification document, together with examples of 4 selected types
What does an EPICS V4 PV for structured data look like?

The pvAccess PV name

The EPICS V4 data type identifier “NTNDArray”, defined in the Normative Types Specification document

The raw image data

The image meta data; giving how to interpret the data in the value, and other information

Figure: A screenshot of the output of the EPICS V4 “pvget” command, showing data of a PV which encapsulates all the data of an areaDetector NDArray (from B. Martin’s AD work later in talk).
EPICS in the nominal usage: An EPICS client communicates over Channel Access (CA) protocol to an Input/Output Controller (IOC) Channel Access server (module rSrv in an IOC)
EPICS Version 4 is an extension of V3

V4 IOC == V3 IOC + pvAccess Server

Use Case: Network efficient acquisition of archived meta data

Presently, only 1 PV per pvAccess channel. But plan is to get/monitor a group of PVs through one pvAccess channel.
EPICS Version 4 includes CA

The pvAccess API includes Channel Access support, so one client lib does both.
A new smart database, “pvDatabase” can be used for data assembly and processing.

Examples; the SNS and NSLS-II beamline experiment high performance data acquisition and processing (later in talk).
EPICS Version 4 middleware support

RPC and Service Oriented Architecture (SOA)

Channel Access Client

pvAccess Client

CA

pvAccess

CA Server

pvAccess Server

CA

pvAccess

pvAccess Server

pvDatabase

Examples: the SLAC/ESS model and infrastructure system; BNL/FRIB configurations; and NSLS-II experiment data support (later in talk)

BYO back-end datasource

IOC database

Device I/O
Talk Outline

• Version 4 Additions to EPICS
• Deployments
• User Feedback and Conclusions
• Recent Work
SNS uses EPICS V4 for high throughput event readout, of structured PV data.

 pvAccess streams event data at 80 Mbytes / sec

The V4 data structure includes an array of pixels and a corresponding array of times of flight for each recorded neutron event. Additional fields record accelerator pulse information and detector diagnostic information.

nED provides the driver interface to the detector electronics and streams experiment data using pvAccess.

ADnED, a pvAccess client, generates online histograms and counting statistics from the nED data stream and serves them using the CA protocol to clients including CS-Studio.

Kay Kasemir, Steven Hartman, ORNL
**SNS’s use of EPICS V4 for transport of beamline neutron event data**

**SNS Conclusions:**

Five beam lines currently using EPICS V4

Plans to extend to all experiment beam lines.

Additionally, A pvaPy-based V4 client is used for detector calibration and diagnostics.

EPICS V4 meets the performance requirements for all existing SNS instruments

Demonstrated at data rates of 10M events per second

Excellent reliability.

ADnED plot of a diffraction pattern from neutron scattering of a single-crystal sample at SNS CORELLI

Fragment of the SNS V4 structure used for streaming experiment data at the 60 Hz rate of the pulsed neutron source

Kay Kasemir, Steven Hartman, ORNL
NSLS-II areaDetector EPICS V4 support

Problem: Modern detector rates
• Eiger 1M: 1030x1065 @ 3 kHz
• Eiger 4M: 2070x2167 @ 750 Hz
• Eiger 9M: 3110x3269 @ 238 Hz
• Eiger 16M: 4150x4371 @ 133 Hz
• All these detector configurations saturate a 10 Gbps link
• Other non-EPICS methods tried and failed (HTTP-chunking).

NSLS-II v4 Solution:
V4 server is an areaDetector plugin, NDPluginPva. V4 client is areaDetector driver.

Architecture tested with SimDetector datasource:

Bruno Martins, BNL (following work by James Rowland and Dave Hickin at Diamond)
NSLS-II V4 areaDetector Performance Test

Test simDetector datasource 5K x 5K @ 50Hz ~= 10 Gb/s over 10Gig Ethernet. Non-blocking callbacks. AD ImageMode: Multiple. NumImages:10000

Transfer bandwidth: EPICS V4 & practical limit:

Conclusions: EPICS V4 based areaDetector pipeline has high throughput, few frames lost, with no CPU saturation. Network bandwidth is close the practical maximum.

Bruno Martins, BNL (following work by James Rowland and Dave Hickin at Diamond)
The V4 PV “Eiger1M:pva1:Image” of type NTNDArray

Figures: NSLS-II CS-Studio screenshots showing an EPICS V4 PV of the type for areaDetector images (NTNDArray) displayed using a CS-Studio “formula.”
NSLS-II use EPICS V4 for Beamline Data Management

Experimental data Logical View

Implemented as EPICS V4 services

Figure: Services with thin, configurable, interfaces allow a small system of services to satisfy diverse requirements of many beamline experiments

Arman Arkilic BNL, Michael Davidsaver then at, BNL
An EPICS V4 server mediates all experiment data

“DataBroker” gives access to all data, from all services, over pvAccess or HTTP.

Arman Arkilic BNL, Michael Davidsaver then at, BNL
EPICS V4 Normative Type (NTTable)
Examples from NSLS-II metaDataStore

NSLS-II beamline “run-start” metadata

```
epics:nt/NTTable:1.0
string[] labels []
structure value
string _id [553ce3af7368e3176b472061]
string animal []
double arman []
boolean beamline_config []
string beamlineconfig_id []
string beamline_id [xf23id]
boolean config []
string config.beamline_id []
boolean config.custom []
string config.group []
string config.owner []
string config.project []
double config.scan_id []
double config.time []
string config.uid []
boolean custom []
string group []
boolean jupiter [false]
string mood []
string plotx [pgm_energy]
boolean ploty []
string project []
boolean sample []
double scan_id [10637]
boolean threading []
double time [1.43005e+09]
boolean time_as_datetime []
string uid [f9a83f88-2d14-469c-9bce-7607e3dbfc83]
string user []
```
BNL and FRIB use EPICS V4 for PV configuration management

The MASAR app (Machine Snapshot, Archiving, and Retrieval) allows a user to take snapshots of systems of CA PVs, save them in a database, view them, and restore them to IOCs. Whole machine configurations can be delivered to clients as a single set using EPICS V4.
MASAR Architecture

Deployments of EPICS Version 4: BNL’s CA PV Configurations save/restore system

Figure: MASAR server side delivers CA PV configurations using EPICS V4, to various client types.

Guobao Shen, FRIB
SLAC and ESS collaboration on EPICS V4 for beam dynamics modelling and infrastructure data

Figure: EPICS V4 modelling service giving orbit response matrices and Twiss parameters for given devices. These are the basis of 95% of emittance minimization applications – feedback, steering, bumps, etc
SLAC and ESS collaboration on EPICS V4 for beam dynamics modelling and infrastructure data

Directory Service (based in EPICS V4 channelFinder) examples:

# The names of PVs, by device name pattern:
$ eget -s ds -a name=XCOR:LI21:135:%
name
  XCOR:LI21:135:ABORT
  XCOR:LI21:135:ACCESS
  XCOR:LI21:135:ALLFUNCGO
  XCOR:LI21:135:BACT
  XCOR:LI21:135:BACTFO

# Regular expression (restrict to sectors LI25-LI29)
eget -s ds -a regex='XCOR:LI2[5-9]:.*:BDES'

# Device names of the instruments in the laser heater
$ eget -s ds -a etype INST -a tag LSRHTR -a show dname

# A recent search for invalid data in corrector PVs
$ eget -tTs ds -a name %COR:LTU%:%:%DES | \\
eget -p ca -f - | grep nan
  XCOR:LTU1:558:BDES nan
  XCOR:LTU1:558:IDES nan

Oracle Database example

$ eget -s LCLS:ELEMENTS

<table>
<thead>
<tr>
<th>ELEMENT</th>
<th>ELEMENT_TYPE</th>
<th>EPICS_DEVICE_NAME</th>
<th>S_DISPLAY</th>
<th>OBSTRUCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATHODE</td>
<td>MAD</td>
<td>CATH:IN20:111</td>
<td>2014.7</td>
<td>N</td>
</tr>
<tr>
<td>SOL1BK</td>
<td>MAD</td>
<td>SOLN:IN20:111</td>
<td>2014.7</td>
<td>N</td>
</tr>
<tr>
<td>CQ01</td>
<td>MAD</td>
<td>QUAD:IN20:121</td>
<td>2014.9</td>
<td>N</td>
</tr>
<tr>
<td>SOL1</td>
<td>MAD</td>
<td>SOLN:IN20:121</td>
<td>2014.9</td>
<td>N</td>
</tr>
<tr>
<td>XC00</td>
<td>MAD</td>
<td>XCOR:IN20:121</td>
<td>2014.9</td>
<td>N</td>
</tr>
</tbody>
</table>

... (many rows snipped)

Figure: Access to Oracle gives device infrastructure, magnet calibrations, drawing names, etc. Will be used in LCLS-II for cryogenic plant system hierarchy etc.

Greg White, Murali Shankar SLAC; Ivo List, ESS
Talk Outline

• Version 4 Additions to EPICS
• Deployments
• User Feedback and Conclusions
• Recent Work
User Feedback – what’s good:

• Performance is excellent
• Reliability needs have been met or exceeded
• Easy programming and scripting, once you’ve got started
• Complex data and RPC enables one, simple, high performance, infrastructure across the whole controls and online scientific system. Utility of this effect previously overlooked, but in practice seen to be key
• Normative Types enable systems of narrowly defined services to be applied generally to many experiment user problems
• Streaming supports big online data processing. Beats tested alternatives in ease of use and performance.
User Feedback – what’s bad

It’s difficult to get started!

We are trying to address that: see especially the new Developer’s Guide: http://epics-pvdata.sourceforge.net/informative/developerGuide/developerGuide.html

But, you know, point taken!
Talk Outline

• Version 4 Additions to EPICS
• Deployments
• User Feedback and Conclusions
• Recent Work
Recent Additions to EPICS Version 4

- High performance array management; enforced copy-on-write semantics and zero-copy. Used by HP areaDetector projects
- Union data types
- Bound and unbound arrays
- Codec based transport, pvAccess can be replaced by zeroMQ for instance
- Security plugin
- Pipelining. Used by HP areaDetector
- New Database, pvDatabase
- Simplified APIs. New easy to use API for synchronous operations
- Easy to use wrappers for introspection interfaces of Normative Types
- Python API
- Developers Guide being written
References

- The EPICS V4 website (packaged downloads, documentation etc), [http://epics-pvdata.sourceforge.net](http://epics-pvdata.sourceforge.net)
- EPICS V4 sourcecode repos, [https://github.com/epics-base/](https://github.com/epics-base/)
- EPICS V4 EVALUATION FOR SNS NEUTRON DATA, K.U. Kasemir, G.S. Guyotte, M.R.Pearson, ORNL, Oak Ridge, TN37831, USA, contribution WEPGF105 of these proceedings
- areaDetector EPICSv4 modules, B. Martins, talk at spring 2015 EPICS Meeting (at Michigan State), [https://indico.fnal.gov/contributionDisplay.py?contribId=81&sessionId=11&confId=9718](https://indico.fnal.gov/contributionDisplay.py?contribId=81&sessionId=11&confId=9718)
- areaDetector's ADCore on github, B. Martins, [http://github.com/areaDetector/ADCore](http://github.com/areaDetector/ADCore)
- NSLS-II Data Management Framework, A. Arkilic, talk at spring 2015 EPICS Meeting (at Michigan State), [https://indico.fnal.gov/materialDisplay.py?contribId=80&sessionId=5&materialId=slides&confId=9718](https://indico.fnal.gov/materialDisplay.py?contribId=80&sessionId=5&materialId=slides&confId=9718)