Experimental Physics and Industrial Control System

EPICS

Kay Kasemir
kasemirk@ornl.gov

Sept. 2014
What EPICS looks like

http://xkcd.com/730/
Australian Square Kilometre Array Pathfinder (ASKAP), Oct. 5, 2012

ASKAP User Interface

First EPICS/CSS operation with beam, Jan. 8, 2013
EPICS is not...

One product that you
• Install
• Run
• Done!
EPICS is a Collaboration

- ~1989: Started between
  - LANL Ground Test Accelerator
  - ANL Advanced Photon Source

- Until 2004: License agreement required
  - LANL registered >150

- Now:
  - SNS, ANL/APS, SLAC, LANL, JLAB/CEBAF, LBNL, Fermilab D0, Keck & Gemini Telescopes … in the USA
  - Canadian Light Source; DESY, BESSY, .. in Germany; PSI/SLS in Switzerland; Ganil, SA CLAY in France; Diamond light source in England; KEK-K, J-Parc in Japan; IHEP in China; NSRRC in Taiwan; PLS in South Korea; Australian Synchroton, …

- Yearly collaboration meetings
  - One each in US and elsewhere
  - 100+ people attended in 2004, 05, 06.

- 'Tech-Talk' email reflector usually provides responses within a few hours
  - http://www.aps.anl.gov/epics
EPICS Meetings

Santa Fe 2004

Tzukuba, 2000

Berlin, 1998
EPICS is a Toolkit

... for distributed control systems.

- Front-end: “Input/Output Controller” (IOC)
- Protocol: Channel Access
- Clients: Operator displays, alarm system, ...

- Mostly Portable: vxWorks, RTEMS, Linux, OS X, Windows
Distributed

- Operator interface
- Services: Archive, ...
- Front-end IOCs
- I/O, PLCs, ..
What an IOC does

• Runtime ‘Database’
  – Executes records

• Known set of ‘Records’
  – Read analog value
  – Write analog value
  – Perform computation
  – Control motor

• Configuration
  – SCAN=1 second
  – INP=..what to read..

• Serve all via Channel Access
Example: Basic Temperature Control

Task:
1. Read temperature
2. Open/close switch as needed
3. Repeat
Simplistic Code

Sensor temp = connectToSensor(...);
Switch switch = connectToSwitch(...);
Loop:
    if (temp.value() < 25)
        switch.close();
    else
        switch.open();
sleep(1.0);
What we omitted

- **Error** checking
- Code comments
- Apply some **smoothing** to the temperature reading to filter noise.
- Send current temperature and switch state to **network** clients (operator display). With **units**.
- Attach a **time stamp** to the data, so that network clients can see for example when the switch was last opened.
- Send **alarm** when the temperature is too low or high.
- Allow **runtime changes** of the threshold from the remote operator interface.
- Allow runtime changes to the **scan rate**.
- Maybe allow runtime changes to the **device address**?
- What if we have more than one fishtank?
EPICS 'Database' for Fishtank

Takes getting used to, but handles what we omitted.
Some Detail on EPICS 'Records'

```c
record(ai, temp) {
    field(DESC, "Read Temperature")
    field(SCAN, "1 second")
    field(DTYP, "XYZ ADC")
    field(INP, "#C1 S4")
    field(PREC, "1")
    field(LINR, "type3degC")
    field(EGU, "Celsius")
    field(HOPR, "100")
    field(LOPR, "0")
    field(SMOO, "0.5")
    field(HIGH, "15")
    field(HSV, "MAJOR")
}

record(calcout, check) {
    field(DESC, "Control Heater")
    field(CALC, "A<10")
    field(INPA, "temp CP MS")
    field(OUT, "switch")
    field(OOPT, "On Change")
}

record(bo, switch) {
    field(DESC, "Heater switch")
    field(DTYP, "XYZ DAC")
    field(OUT, "#C1 S3")
    field(ZNAM, "Open")
    field(ONAM, "Closed")
    field(IVOA, "Set output to IVOV")
    field(IVOV, "0")
}
```

**Programming Configuration**

- "SCAN=1 second" instead of start thread, delay until next multiple of 1 second, lock required resources, ...

- "SMOO=0.5" configures the smoothing algorithm.

- Almost any field in any record is accessible via network at runtime
  - Change scan rate, smoothing, ...

Managed by UT-Battelle for the Department of Energy
**IOC Database**

- A single record can handle the scanning, signal conditioning, alarming of a temperature, pressure, or similar analog reading.

- Combined with binary and computational records, it can express the *data flow* logic for a front-end computer
  - Avoiding the pitfalls of real-time, multithreaded and networked programming.

- Can have thousands of records in one IOC.

- kHz-rate processing with record chains is doable
  - Of course limited by CPU. Not 1000nds of kHz rate-records…
How fast?

- Can be fast or slow, it depends how you use it!
- Use the correct tool for the job; Database, custom code (IOC) or custom code (client)
- Ultimately speed depends upon hardware
- Some benchmarks*:

<table>
<thead>
<tr>
<th>Machine</th>
<th>OS</th>
<th>CPU</th>
<th>Speed</th>
<th>Rec/sec</th>
<th>%CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVME167</td>
<td>vxWorks</td>
<td>68040</td>
<td>33MHz</td>
<td>3,000</td>
<td>25</td>
</tr>
<tr>
<td>MVME2306</td>
<td>vxWorks</td>
<td>PPC604</td>
<td>300MHz</td>
<td>20,000</td>
<td>20</td>
</tr>
<tr>
<td>MVME5100</td>
<td>vxWorks</td>
<td>PPC750</td>
<td>450MHz</td>
<td>100,000</td>
<td>25</td>
</tr>
<tr>
<td>PC</td>
<td>Linux</td>
<td>PII</td>
<td>233MHz</td>
<td>10,000</td>
<td>27</td>
</tr>
<tr>
<td>PC</td>
<td>Linux</td>
<td>P4</td>
<td>2.4GHz</td>
<td>100,000</td>
<td>18</td>
</tr>
</tbody>
</table>

* Extrapolated from benchmark figures courtesy of Steve Hunt (PSI) and L.Hoff, (BNL)

From Andrew Johnson (APS) EPICS Intro slide
Example Client: Operator Displays

Created in Editor
- No coding nor compilation

Networked
- Open/close
- Multiple OPIs
EPICS Vocabulary

• **EPICS Base**
  Code for IOC, database support, basic records, channel access

• **IOC**
  Input Output Controller, the front-end software
  – **Hard IOC**
    Using real-time OS in VME crate
  – **Soft IOC**
    IOC software running on Linux, …, typically communicating with networked I/O

• **Database**
  Executes the EPICS Records

• **Record**
  EPICS processing block

• **Device support**
  Code that connects records to hardware Driver

• **Driver**
  Code that talks to hardware. May be unaware of EPICS

• **Channel Access**
  EPICS network protocol. Exposes Channels aka Process Variables
EPICS Summary

• Control System Toolkit
  – Distributed, multi-platform, open source, extensible, …

• Not fancy, but “works”
  – Accelerators,
    Beam lines,
    Telescopes,
    Fusion experiments,
  …