WebScan
(or Goodbye scanSee!)
Visualising ssCan data in a web browser

David Vine

Advanced Photon Source
Argonne National Lab

APS Technical Working Group
15 August 2013
Displaying sscan data in a browser

- Overview
- The PV Archiver
- Implementation
- WebScan
1 Displaying sscan data in a browser
   • Overview
   • The PV Archiver
   • Implementation
   • WebScan
Why put scans on the web?

- Cross platform (Mac, *nix, Windoze, SunOS)
- Location independent - beamline, guesthouse, home (via VPN)
- Single instance to maintain
- Searchable archive of scan data (currently 7 days stored)
What is it?

- Matt Newville’s EPICS PV Archiver
  - Python + PyEpics
  - LAMP server

- WebScan
  - Javascript + Google Chart Tools + ASP
Displaying sscan data in a browser
- Overview
- The PV Archiver
- Implementation
- WebScan
The EPICS PV Archiver

cars9.uchicago.edu/~newville/Epics/PVArchiver/
## The PV Archiver

### Monitoring
- **Approximately 2500 PVs, 7 beamlines, 4 sectors**

### MIC Beamlines Status:

**Storage Ring**
- **SR Operating Status**
- **SR Current [mA]**
- **SR Lifetime [h]**
- **Shutter Permit**

**Shutter Summary**
- 2-ID: A, B, D, E
- 8-BM: A, B
- 26-ID: A, B, C
- 34-ID: A, B

**Scan Summary**
- 2-ID-B Scan
- 2-ID-E Step Scan
- 2-ID-E Fly Scan
- 2-ID-D Scan
- 2-ID-DF Scan
- 8-BM-B Scan
- 34-ID-C Scan

**2-ID: U3.3 Undulator**
- **Energy [keV], Gap [mm], Harmonic**

**2ID: U5.5 Undulator**
- **Energy [keV], Gap [mm], Harmonic**

**26ID: U3.3 Undulator**
- **Energy [keV], Gap [mm], Harmonic**

**34ID: U3.3 Undulator**
- **Energy [keV], Gap [mm], Harmonic**

---

**Delivered Beam**
- 102.221
- 8.263
- PERMIT

**WebScan**
- 1, OFF, OFF, OFF
- ON, ON, OPEN, OPEN
- 1, open

**Status**
- IDLE
- IDLE
- WAIT:DECTRS
- IDLE
- WAIT: MOTORS

**Settings/Admin Help**

---

[ APS Storage Ring Status | APS Facility Page | APS OAG Data ]
Outline

1. Displaying sscan data in a browser
   - Overview
   - The PV Archiver
   - Implementation
   - WebScan
1 Displaying sscan data in a browser
   • Overview
   • The PV Archiver
   • Implementation
   • WebScan
Design philosophy

- Different paradigm to Archiver
  - not a replacement for mda files
- Store scan data in MYSQL DB
  - delete entries > 7 days old
- DB atomicity - scan rows
- Cache values to local variable
- Serialize and write arrays to DB
- Using callbacks:
  - no need for handshaking (AAWAIT, AWAIT)
  - sscan performance is not slowed
- Assumes that slow axis→ fast axis goes Scan2→Scan1→ScanH
Features

Current:
- Realtime line-by-line scan updates
- Realtime images
- Select & deselect detectors at any time during scan
- Oblivious to IOC reboot & scan progress
- Monitor scans in multiple IOCs simultaneously

Planned:
- Click through scan history
- Interactive web pages
Collecting SScan Data

- **Scalar**
  - `ioc:scan1.DnnCV`
  - Less network overhead

- **Array**
  - `ioc:scan1.DnnCA`
  - Independent of IOC reboot
  - Ensures all data collected
Collecting SScan Data

- **Scalar**
  - `ioc:scan1.DnnCV`
  - Less network overhead

- **Array**
  - `ioc:scan1.DnnCA`
  - Independent of IOC reboot
  - Ensures all data collected
How does it work

Scan
- Record

DnnCA

PV
- PV
- PV

Channel Access

Scan

Cacher

Archiver

MYSQL DB

MYSQL Queries

Mod Python

cgi-bin python scripts to generate webpages

Apache Server

Browser

Browser

Browser
Scan Process - Pseudocode

**main loop**

Connect to database

Connect to PVs & add callback

while True:
    Process queue

Update scan database

**on callback**

Cache new value
Append pv to queue

**Process queue**

while not queue == empty:
    pvname, value = queue.pop()
    if pvname ends with "2.EXSC":
        if value == 1: # 2D Scan Begins
            ...
        else if pvname ends with "CA":
            ...
    add pvname to scan DB queue
Scan Database

<table>
<thead>
<tr>
<th>id</th>
<th>pvname</th>
<th>value</th>
<th>row</th>
<th>scan_id</th>
<th>ts</th>
</tr>
</thead>
<tbody>
<tr>
<td>90439</td>
<td>djv:ScanDim.VAL</td>
<td>F2\n</td>
<td>0</td>
<td>/home/david/data/mda/djv_0091</td>
<td>1376323793.51191</td>
</tr>
<tr>
<td>82864</td>
<td>djv:scan1.EXSC</td>
<td>l0\n</td>
<td>0</td>
<td>/home/david/data/mda/djv_0057</td>
<td>1376323952.78413</td>
</tr>
</tbody>
</table>

Query for single row

MYSQL: select value from scan where pvname="djv:scan1.D01CA" and row=0 and scan_id="xxx";
Python: self.scan.select('value', 'pvname="djv:scan.D01CA" and row=0 and scan_id="xxx"')
Browser Side Interaction

- Continuous updating with ASP and Google Chart Tools
- Chart tools requires connection to internet to Google JavaScript API (data NOT sent to Google)
- Page uses ≈100% of 1 processor
Future

- Add some more advanced features of scan see - derivative, FWHM
- More interactive pages with ASP
- Migrate to Django