Bluesky (et al.) at the APS

**Pete Jemian**
BCDA group
X-ray Science Division
Advanced Photon Source
Argonne National Laboratory

2018-06-11
Topics

Bluesky (et al.) at the APS

- APS beam line control environment
- Bluesky software installation
- User operating environment
- Results from 2018-06-07 beam time
- Conclusion
APS beam line control environment

- APS is very diverse
- More than 60 beam lines in operation
- More than half are facility-managed
- EPICS used at most, not all, beam lines
- Data acquisition code is diverse
  - Multiple tools used together, segmented decisions, deep investment
- Data retention policies are diverse
- Yet: Facility is operating and publishing

- New software should be compelling and provide what is not already possible (or easy). Be easy to use. And, most important, without any flaws. A tall order.
- APS-U upgrade offers ripe opportunity to advance the data acquisition code suite
- Early demos of Bluesky capabilities are most persuasive
Bluesky software installation

- **Database**
  - One *mongodb* server for each sector or beamline
  - Will monitor disk usage
  - Q: *Any advantage to coordinate these servers?*
- **Common Python software managed by BCDA support group**
  - Common read-only installation for all beam lines
    (updated via nightly rsync same as other beam line control software)
  - Local installation for exceptional needs
  - Don’t rely on virtual environments
  - Install additional tools as needed
  - Keep public HISTORY.txt file of all updates
- **Instrument-specific software**
  - Default ipython profile
  - Jupyter notebooks to document or build tutorials
GitHub use

- Use GitHub organizations to provide version control for beam line configurations.

- Naming convention
  - Create a GitHub organization with name like: APS-SSS-GGG
    - SSS: sector, beam –line, and station (such as 2BM)
    - GGG: operating group (such as MIC for the Microscopy group)
  - Within each organization, create a repository: ipython-username
    - ipython: the text ipython
    - username: instrument account, such as instruser

- Consistent naming makes similar work easier to locate
  - facilitates sharing of common code
- Similar to pattern established by NSLS-II DAMA team

https://github.com/BCDA-APS/use_bluesky/wiki
GitHub APS beam line organizations

<table>
<thead>
<tr>
<th>facility</th>
<th>db host</th>
<th>URL for GitHub organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>bcda</td>
<td>otz</td>
<td><a href="https://github.com/BCDA-APS/use_bluesky">https://github.com/BCDA-APS/use_bluesky</a></td>
</tr>
<tr>
<td>2-BM</td>
<td>arcturus.xray</td>
<td><a href="https://github.com/APS-2BM-MIC/ipython-user2bmb">https://github.com/APS-2BM-MIC/ipython-user2bmb</a></td>
</tr>
<tr>
<td>3-ID</td>
<td>dy.xray</td>
<td><a href="https://github.com/APS-3ID-IXN/ipython-s3blue">https://github.com/APS-3ID-IXN/ipython-s3blue</a></td>
</tr>
<tr>
<td>USAXS at 9-ID-C</td>
<td>usaxsserver.xray</td>
<td>(*) <a href="https://github.com/APS-USAXS/ipython-usaxs">https://github.com/APS-USAXS/ipython-usaxs</a></td>
</tr>
<tr>
<td>12-ID-B</td>
<td>eggplant.xray</td>
<td><a href="https://github.com/APS-12IDB-GISAXS/ipython-s12idb">https://github.com/APS-12IDB-GISAXS/ipython-s12idb</a></td>
</tr>
<tr>
<td>29-ID</td>
<td>groggy.xray</td>
<td><a href="https://github.com/APS-29ID-IXE/ipython-29id">https://github.com/APS-29ID-IXE/ipython-29id</a></td>
</tr>
</tbody>
</table>

- naming variant since this instrument has moved to several beam lines
# Typical ipython layout

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipynb_checkpoints</td>
<td>this demo should go</td>
<td>20 days</td>
</tr>
<tr>
<td>00-0-checks.py</td>
<td>STY: whitespace</td>
<td>4 days</td>
</tr>
<tr>
<td>00-startup.py</td>
<td>working fine diagnostics off now</td>
<td>3 days</td>
</tr>
<tr>
<td>01-databroker.py</td>
<td>#16 initial setup</td>
<td>a month</td>
</tr>
<tr>
<td>02-pyepics.py</td>
<td>#16 initial setup</td>
<td>a month</td>
</tr>
<tr>
<td>10-imports.py</td>
<td>#22 new working code</td>
<td>3 days</td>
</tr>
<tr>
<td>11-motors.py</td>
<td>#17 - Wahoo - first working plan, no HDF yet</td>
<td>4 days</td>
</tr>
<tr>
<td>15-custom-devices.py</td>
<td>weeds</td>
<td>3 days</td>
</tr>
<tr>
<td>20-signals.py</td>
<td>fixes #22</td>
<td>3 days</td>
</tr>
<tr>
<td>25-PG3-grasshopper.py</td>
<td>fixes #26</td>
<td>3 days</td>
</tr>
<tr>
<td>30-busy_fly_scan.py</td>
<td>fixes #26</td>
<td>3 days</td>
</tr>
<tr>
<td>45-interruptions.py</td>
<td>reset stop that bit after MONA requests us to stop</td>
<td>3 days</td>
</tr>
<tr>
<td>60-handler.py</td>
<td>comments</td>
<td>3 days</td>
</tr>
<tr>
<td>60-metadata.py</td>
<td>good default value</td>
<td>20 days</td>
</tr>
<tr>
<td>README</td>
<td>#16 initial setup</td>
<td>a month</td>
</tr>
</tbody>
</table>
User operating environment

- Challenging
  - Deploying new ipython profiles
  - Keeping existing ipython profiles consistent with updates

- Using common tools for new deployments
  - [https://github.com/BCDA-APS/use_bluesky](https://github.com/BCDA-APS/use_bluesky)
**APS Bluesky tools**

- **Starter script:** `use_bluesky.sh [profile]` for interactive use
  - Runs Python software and correct ipython profile
  - [https://github.com/BCDA-APS/use_bluesky/tree/master/bin](https://github.com/BCDA-APS/use_bluesky/tree/master/bin)

- **Common code for APS:**
  - **Caveat:** Much of this existing code needs to be update for ophyd v1.0
  - **Code:** [https://github.com/BCDA-APS/APS_BlueSky_tools](https://github.com/BCDA-APS/APS_BlueSky_tools)
  - **Docs:** [http://aps-bluesky-tools.readthedocs.io](http://aps-bluesky-tools.readthedocs.io)
  - **Devices:** shutters, attenuation filters, APS info (e.g., SR current)
  - **Callbacks:** write scan data to SPEC file
  - **Plans:** `TuneAxis` so each motor can *know* how to be tuned
Example Bluesky session

ipython console

various GUIs
caQtDM  MEDM  PyQt  ImageJ

text editor (minimized)
Camera

Sample

Mono

APS 2-BM

camera control

FPGA

motor controller

PV gateway

Bluesky and other experiment controls

Verifier (QA)

Proxy

Visualization

Reconstruction
2018-06-07 beam time at 2-BM-B

MONA project
Monitor, Optimize, Navigate and Analyse experimental conditions and progress on-the-fly

Interlaced Fly Scan Tomography with real-time data streaming to QA, reconstruction, and visualization

- Bluesky directs the measurement
- Motor controller triggers camera via FPGA
- Images as EPICS 7 PVaccess structures
- Images also written to local HDF5 file (one file)
- QA code can stop experiment if data bad
- Reconstruction code on remote cluster (ALCF)
- Sinogram visualization

24 rotations, 12.5s per full rotation
10 ms per image, 1920x1200, 16-bit
95.6 ms & 2.8695° between images
30°/s, 3011 images

PointGrey Grasshopper3, USB
Aerotech Ensemble motor controller
softGlueZynq FPGA
APS Fly scans in Bluesky

- Only core components shown
  - Data typically recorded externally
  - Each busy record calls one or more sseq records which perform sequence of data acquisition steps
- Fly scans are often hardware-assisted and unique to each instrument
- Bluesky must interface to existing code
- Awkward to implement as ophyd Flyer (data collected externally)
- We’re still learning
Console session, 16x

- 30s preparation phase
- 300s fly scan, 3011 frames
- ~120s finish writing HDF5 data
Reconstruction, 16x

- 300s fly scan, 3011 frames
- 1 sinogram shown
MONA team acknowledgements

- APS MONA team
  - Doga Gürsöy, project lead
  - Tekin Bicer
  - Barbara Frosik
  - Arthur Glowacki
  - Pete Jemian
  - Nicholas Schwarz

- APS support
  - Fred Carter
  - Mark Engbretson
  - Patricia Fernandez
  - Troy Lutes
  - Tim Mooney
  - Kevin Peterson
  - Giampiero Sciutto
  - Roger Sersted
  - Ron Sluiter
  - Stefan Vogt

- 2-BM-B Tomo Instrument
  - Xianghui Xiao
  - Pavel Shevchenko
  - Francesco de Carlo

- Argonne ALCF
  - William Scullin

- LBL
  - Harinarayan Krishnan
  - Dinesh Kumar
  - Ronald Pandolfi
  - Dilworth Parkinson
  - James Sethian

- NSLS-II DAMA
  - Dan Allan
  - Tom Caswell
  - Julien Lhermitte
  - Maksim Rakitin
Conclusion
Thank you for your attention