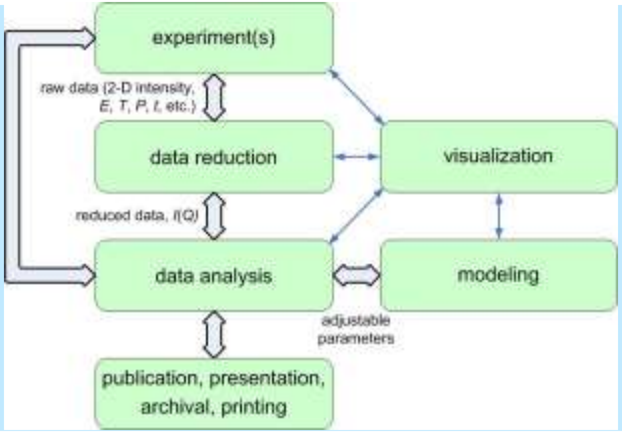
Bluesky Satellite workshop at
2018 EPICS Collaboration Meeting

Bluesky (*et al.*) at the APS



measurement workflow

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

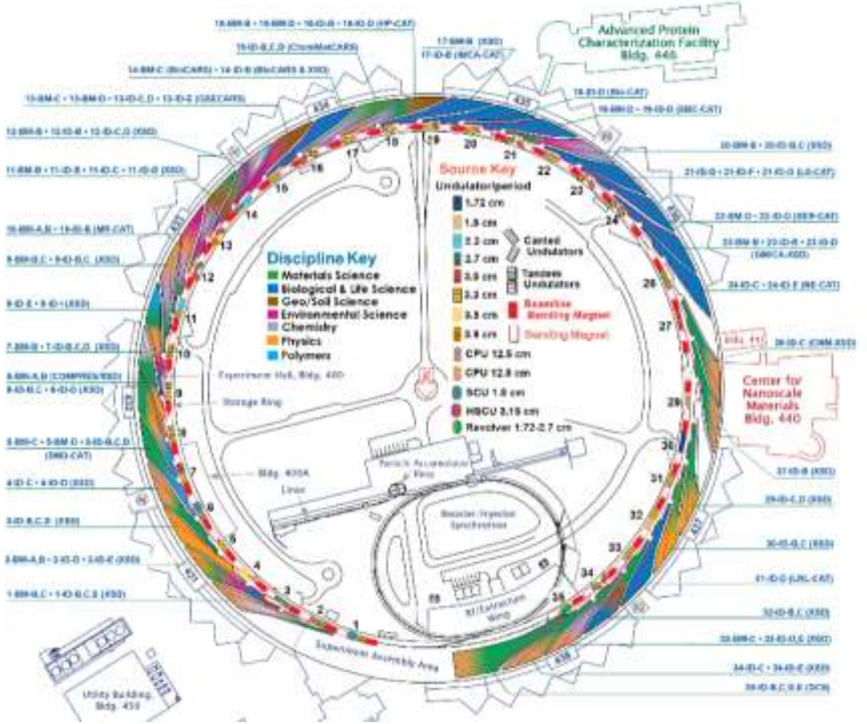
2018-06-11

ABCD

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

facility	db host	URL for GitHub organization
bcda	otz	https://github.com/BCDA-APS/use_bluesky
2-BM	arcturus.xray	https://github.com/APS-2BM-MIC/ipython-user2bmb
3-ID	dy.xray	https://github.com/APS-3ID-IXN/ipython-s3blue
USAXS at 9-ID-C	usaxsserver.xray	(*) https://github.com/APS-USAXS/ipython-usaxs
12-ID-B	eggplant.xray	https://github.com/APS-12IDB-GISAXS/ipython-s12idb
29-ID	groggy.xray	https://github.com/APS-29ID-IEX/ipython-29id
32-ID-C	32idcws.xray	https://github.com/APS-32IDC-MIC/ipython-32idc

- naming variant since this instrument has moved to several beam lines

Typical ipython layout

Branch: master ▾ ipython-user2bmb / profile_2bmb / startup /

Create new file | Upload files | Find file | History

prjemian comments — Latest commit cd13eac 3 days ago

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

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

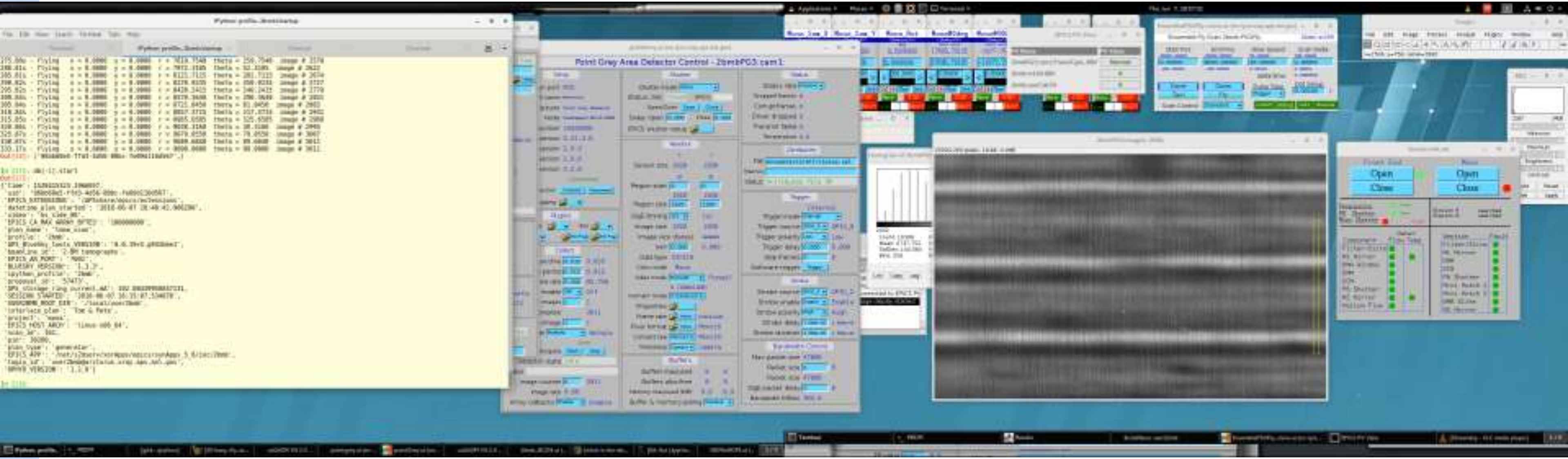
APS Bluesky tools

- 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

- 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
  - Docs: 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

    https://github.com/APS-USAXS/ipython-usaxs/blob/master/profile_bluesky/startup/29-axis_tuning.py

Example Bluesky session



various GUIs

ipython console

caQtDM

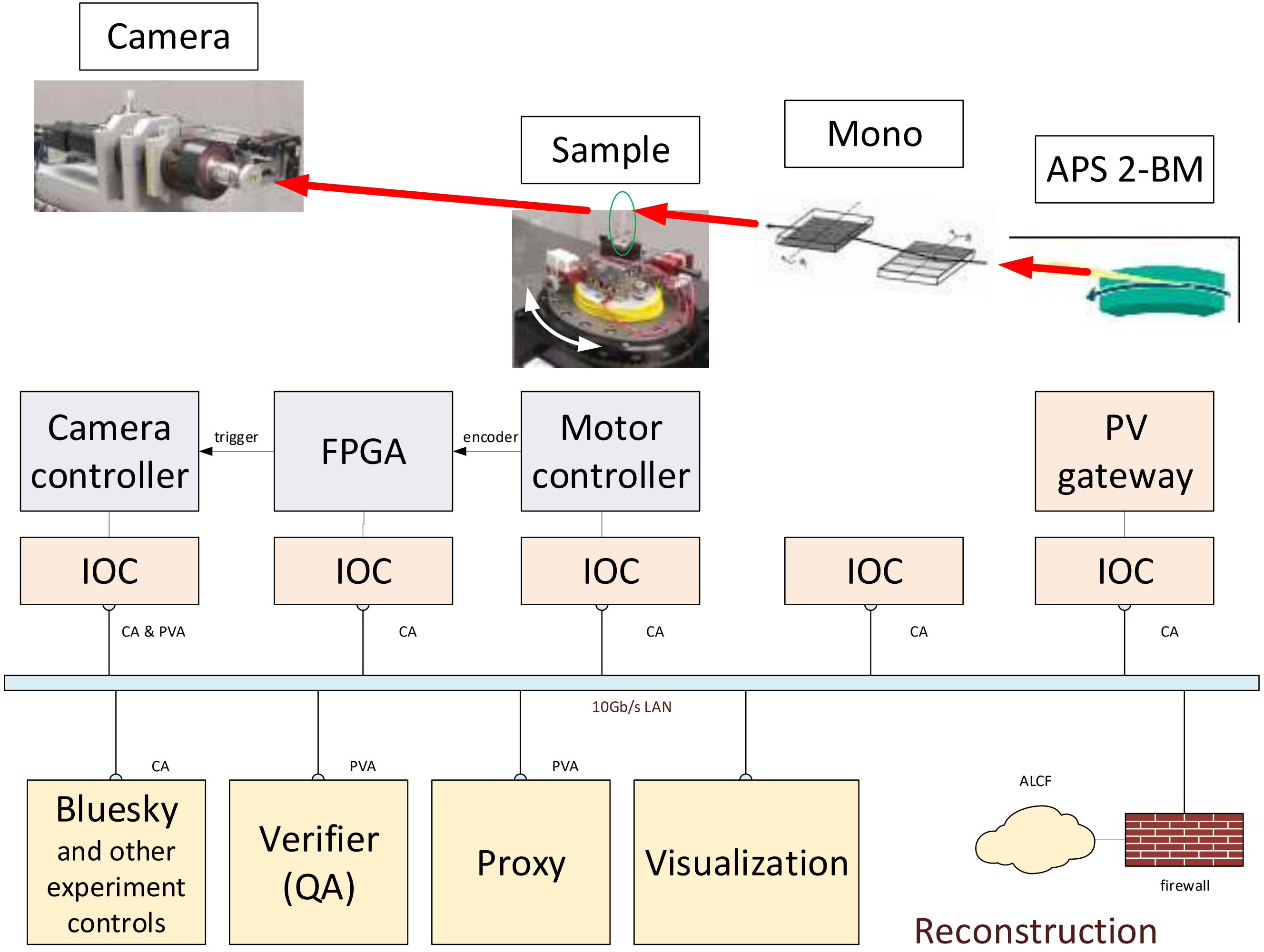
MEDM

PyQt

ImageJ

text editor (minimized)

2-BM-B tomo



Camera
Sample
Mono
APS 2-BM
Camera controller
trigger
FPGA
encoder
Motor controller
PV gateway
IOC
IOC
IOC
IOC
IOC
CA & PVA
CA
CA
CA
CA
10Gb/s LAN
CA
PVA
PVA
Bluesky and other experiment controls
Verifier (QA)
Proxy
Visualization
ALCF
firewall
Reconstruction

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

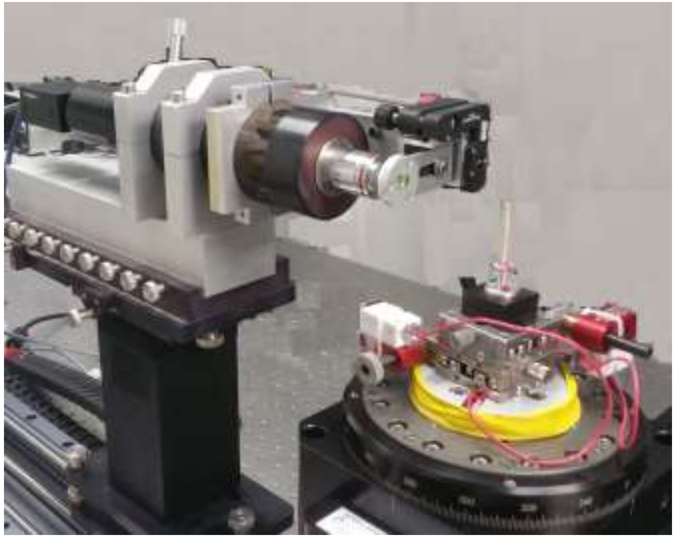
MONA project

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

Argonne NATIONAL LABORATORY

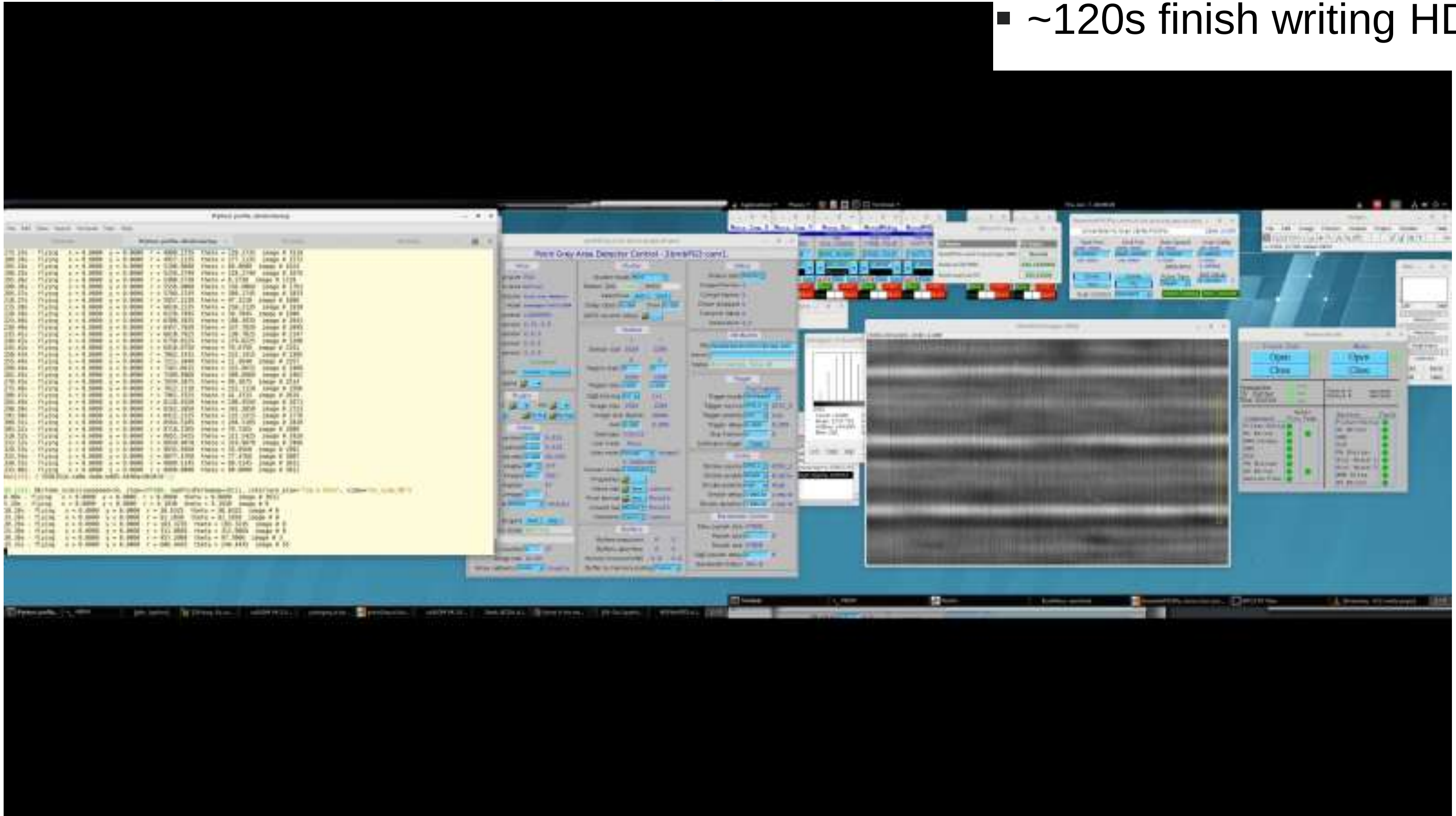
APS Fly scans in Bluesky

- 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

```python
class TaxiFlyScanDevice(Device):
    """
    BlueSky Device for APS taxi & fly scans

    Some EPICS fly scans at APS are triggered by a pair of
    EPICS busy records. The busy record is set, which triggers
    the external controls to do the fly scan and then reset
    the busy record.

    The first busy is called taxi and is responsible for
    preparing the hardware to fly.
    The second busy performs the actual fly scan.
    In a third (optional) phase, data is collected
    from hardware and written to a file.
    """
    taxi = Component(EpicsSignal, "taxi", put_complete=True)
    fly = Component(EpicsSignal, "fly", put_complete=True)

    def plan(self):
        #logger.info("before taxi")
        yield from bps.mv(self.taxi, self.taxi.enum_strs[1])
        #logger.info("after taxi")

        #logger.info("before fly")
        yield from bps.mv(self.fly, self.fly.enum_strs[1])
        #logger.info("after fly")
```

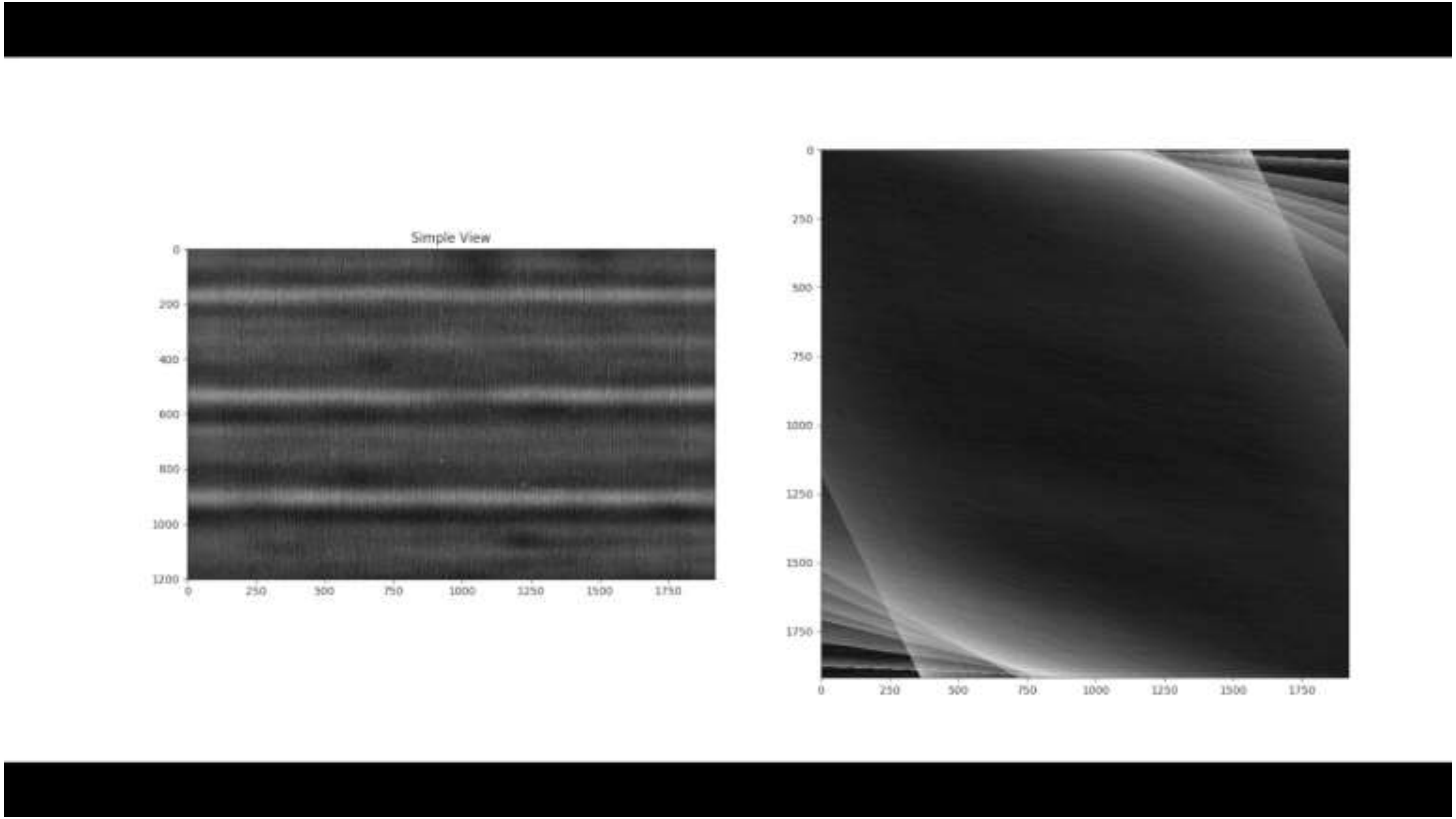
Console session, 16x

- 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 Gürsŏy, project lead
  - Tekin Bicer
  - Barbara Frosik
  - Arthur Glowacki
  - Pete Jemian
  - Nicholas Schwarz
- 2-BM-B Tomo Instrument
  - Xianghui Xiao
  - Pavel Shevchenko
  - Francesco de Carlo
- APS support
  - Fred Carter
  - Mark Engbretson
  - Patricia Fernandez
  - Troy Lutes
  - Tim Mooney
  - Kevin Peterson
  - Giampiero Sciutto
  - Roger Sersted
  - Ron Sluiter
  - Stefan Vogt
- 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

SAS2018

http://sas2018.anl.gov