Diamond II

Dr Nick Devenish

GridPP49/Swift-HEP05 -29th March 2023

Data processing challenges in Synchrotron Macromolecular Crystallography at Diamond

Dr Nick Devenish

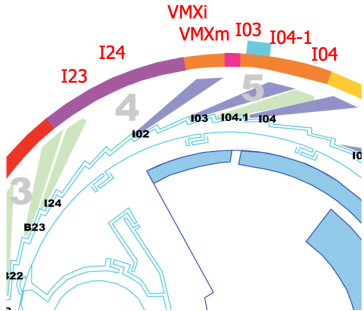
GridPP49/Swift-HEP05 -29th March 2023

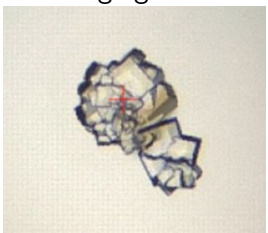
Diamond Light Source

- UK national synchrotron
- Co-funded by STFC and Wellcome trust
- 3GeV electron ring
- First phase completed in 2007
- Industrial (>100 companies) and Academic (>12,000) users

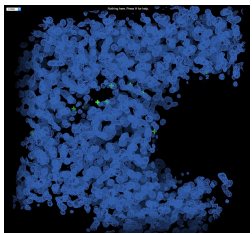
MX - Macromolecular Crystallography

- X-ray crystallography imaging, specifically targeting biochemical molecules - proteins, viruses, nucleic acids
- Typical crystal unit cells ~100 Å
- 7 beamlines dedicated to MX at Diamond
- ~40% DLS papers come from MX
- Multifaceted approach X-ray diffraction, microscopy, spectroscopy, fluorescence, IR imaging etc..



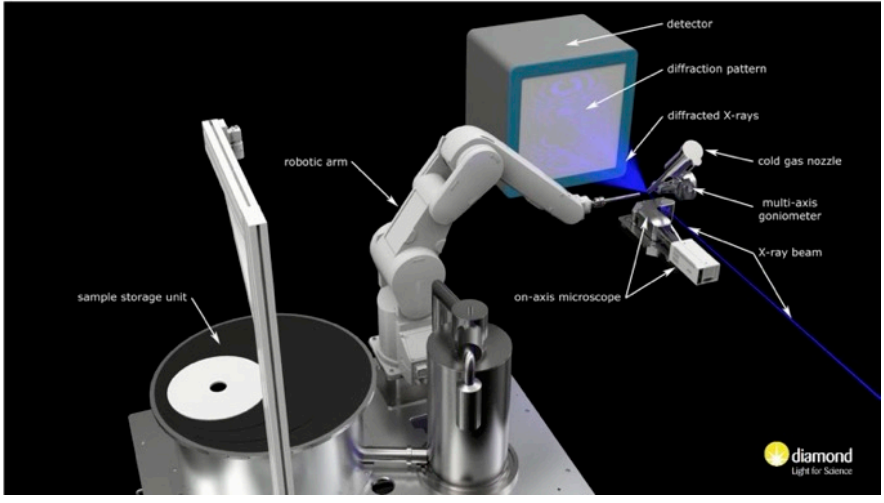


->



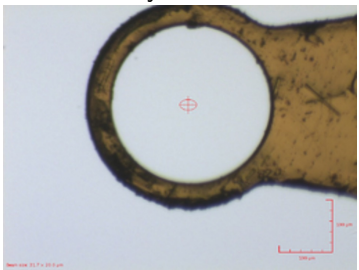
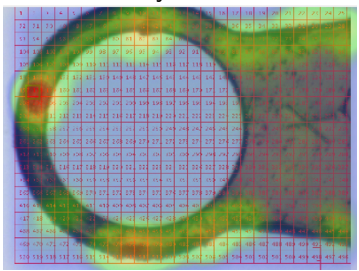
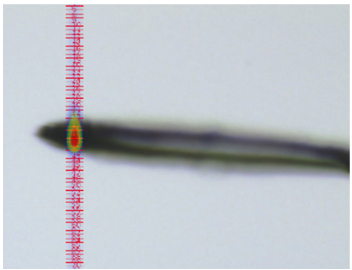
Collecting and Processing MX Data - Mounting

- Robot arm mounts samples onto goniometer



Collecting and Processing MX Data - Centring

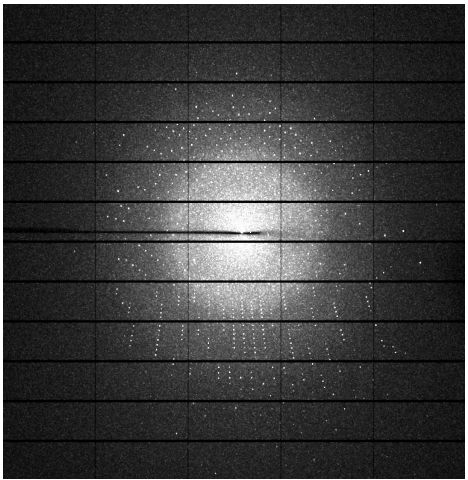
- Pin is first centered with the On-axis camera visually
- A 2D X-ray scan is done to find the crystal in the loop
- A 1D X-ray scan is done to center the crystal in 3D

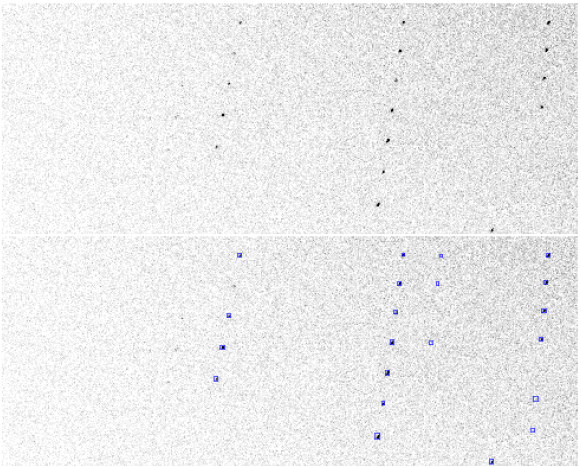




- Sample is then rotated 360° with continuous exposure

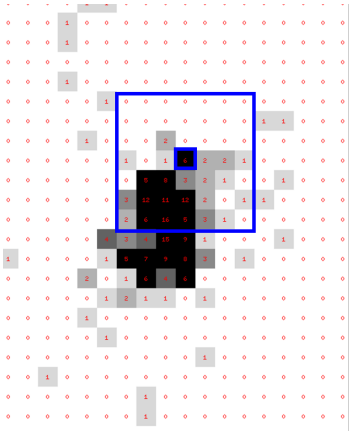
Collecting and Processing MX Data - Spotfinding

Spotfinding is used to classify images from scan to find the crystal





Spotfinding - Algorithm



- Rolling kernel of 7x7, centered over every pixel on the image
- Mean, variance calculated within each window. Pixel is classified as "Strong" if it satisfies:

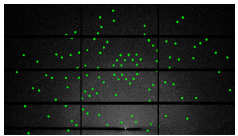
$$\frac{\sigma^2}{\mu} > G\left[1 + \sigma_b\left(\frac{2}{N-1}\right)^{1/2}\right]$$

$$c_i > \mu + \sigma_s\left(G\mu\right)^{1/2}$$

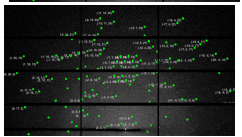
- A Connected Components algorithm is used to group contiguous pixels into "reflections"
- On CPU, this is more efficiently calculated using Summed Area Tables.

Collecting and Processing MX Data - The Rest

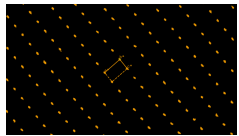
Once we've taken the experimental dataset, it goes through "offline" processing:

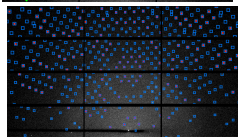


- Spot finding gives us the location of every "strong" reflection on every (consecutive) image



- Indexing transforms these spots into reciprocal space, runs an FFT to extract the crystal lattice and orientation, identifying true reflections





- Using the knowledge of the lattice, predict where every weak or absent reflection should be, and record the measured intensities

After this, we do various corrections for beam, absorption effects, then pass to downstream processing (which eventually gives the user electron density maps of their sample).

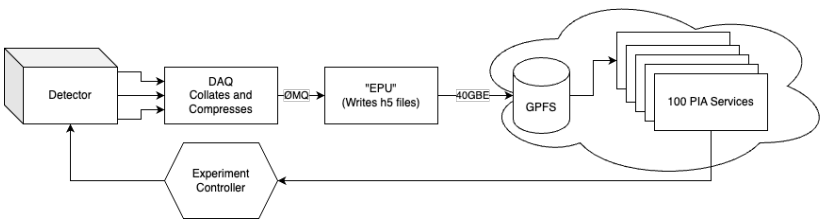
The Problem (i) - Data Rates - number go up

(pre-photon-counting detectors excluded)

Detector	Beamline	Date	Resolution	Rate	Data Rate
DECTRIS Pilatus3 6M	I24	2009	6 Mpx	100 Hz	1.2 GBps
DECTRIS Pilatus 2M	I04-1	2010	2 Mpx	100 Hz	0.4 GBps
DECTRIS Pilatus 6M	I03	2011	6 Mpx	25 Hz	0.3 GBps
DECTRIS Pilatus 6M-F	I04	2013	6 Mpx	140 Hz	1.7 GBps
DECTRIS Pilatus3 6M	I03	2014	6 Mpx	100 Hz	1.2 GBps
DECTRIS Pilatus3 6M-F	I04-1	2015	6 Mpx	25 Hz	0.3 GBps
DECTRIS Eiger2 XE 16M	I04	2019	16 Mpx	560 Hz	17.6 GBps
DECTRIS Eiger2 XE 16M	I03	2019	16 Mpx	550 Hz	17.6 GBps
DECTRIS Eiger2 X 9M	VMXm	2020	4 Mpx	245 Hz	4.4 GBps
DECTRIS Eiger2 X 4M	VMXi	2021	4 Mpx	560 Hz	4.48 GBps
DECTRIS Eiger2 XE 9M	I04-1	2021	9 Mpx	550 Hz	9.9 GBps
DECTRIS Eiger2 X 9M	I24	2021	9 Mpx	245 Hz	4.4 GBps
Jungfrau 1M		2023	1 Mpx	2200 Hz	4.4 GBps
Jungfrau 9M		2024	9 Mpx	2200 Hz	39.6 GBps

The Problem (ii) - Throughput, (iii) - Latency

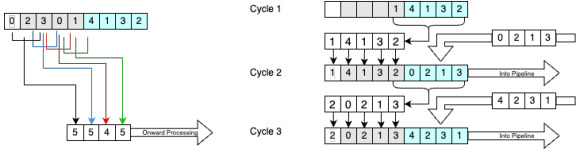
- These data rates are *burst* rates. ~3600 images over ~8 seconds
- Beamline throughput is getting higher
  - Serial (without rotation) collections getting more common. Continuous >25,000 images
  - Automated collection: Time spent waiting for results affects throughput.
- Future beamlines push this: MX-Bridge/Diamond II plans for new sample every 30s
- Competition on shared resources for running spotfinding at ~1s/16Mpx image



- We want to compute efficiently and locally

Solution Exploration (i): Intel OneAPI for FPGA

- Started project developing on Intel OneAPI - a SYCL HLS implementation that can target (Ex-Altera) FPGA
- Pitched as free software targeting specific card implementations
- Structured like CUDA; Pure C++, no exposure to device uploads or bitstreams. Good reporting.
- FPGA is fast! Can guarantee processing as fast as you can transfer data into the card (250fps uncompressed on PCIE3)
- Developed fully pipelined version of algorithm



But...

- Purchased 2x (Stratix) D5005 cards January 2022... and promptly discontinued
- All replacement devices require software licensing. Not a showstopper, but -
- Encountering several compiler bugs and showstoppers
  - Months of work to track down, with Intel engineers telling us was our code
  - "DevCloud" FPGA nodes just outright broken or offline
  - Really wondered how many people actually using
- Massive churn: *Every release* required non-significant but annoying changes. Became significant over 2+ releases. All documentation constantly moving.
- Cards we bought still "Supported" but cannot install drivers and toolkit on same machine

Solution Exploration (ii): CUDA

- Feeling burnt by abstraction, looked to CUDA direct
- Previously advised that CUDA wouldn't be a good fit, seems to be wrong
- Documentation and support is *fantastic*
- Quickly built naive-approach solution appears to work well

```
                Copy:   1.7 ms
              Kernel:   3.5 ms
           Copy Back:   0.8 ms
CPU post-processing:   11.3 ms
                       ========
               Total:  15.6 ms (1.3 GBps)
```

- On a 32-Core workstation, can feed Turing GPU fast enough to sustain 10.42 GBps image processing
- Live testing on I03 beamline next run on Pascal P100 GPUs

Solution Exploration (iii): (AMD) Xilinx

- Longer-term we still want FPGA - energy usage, and throughput
- Purchasing new "Jungfrau" detectors from PSI
  - 1 Mpx on-site now. 2.2 KHz/4.4 GBps
  - 9 Mpx in near future. 2.2 KHz/39.6 GBps (>8x PCIE 5)
- Don't come with DAQ/control systems, MX have to write our own
- Contacts have reported better experience with Xilinx

- Bought dev boards, need to build expertise
- The system isn't abstracted away by their HLS solution
- Ongoing project to evaluate and skillup

People

MX Operations

- Graeme Winter
- Richard Gildea
- Irakli Sikharulidze
- Ben Williams

DIALS Team

- James Beilsten-Edmands

Beamlines

- Neil Patterson

DLS Scientific Computing

- James Thorne
- Murray Collier

Ral PPD

- Stewart Martin-Haugh
- Dave Newbold
- Thomas Williams
- David Sankey
- Rob Halsall
- Sam Harper

We have Questions:

- Are we heading into a haunted forest of bad decisions?
- Is our Intel experience atypical?
- How easy have you found it to onboard new developers? Non-EE Physicists?
- How hard is it to otherwise *train* people?

Questions?

Comments?

Comments disguised as Questions?

I want them all! Please find me.