Live Eiger Analysis with HDF5 & SWMR

Graeme Winter

2021 European HDF5 Users Group



Acknowledgements - many folks at Diamond

Controls: Emma Dixon, James O'Hea, Gary Yendell

Data Acquisition: Chris Kelley, Paul Coe, Neil Smith

Data Analysis: Richard Gildea, Markus Gerstel

Infrastructure: Dave Bond, Rich Lear, Frederik Ferner, James Thorn

Beamlines: Ralf Flaig, David Aragao, Neil Paterson, Dave Hall, Danny Axford, Robin Owen

































Challenge : giving live feedback (Previous gen)



- Nice:
- Can scale CPU •
- Can analyze during acquisition ullet

Nasty:

GPFS dislikes 1000's of small files •

Eiger: default configuration



- OMQ allows you to do what you like
- File writer gives you finished data

Diamond configuration (late '18)



Nice:

- Can acquire 500+ Hz / 16 M pixel for quite a long time Nasty:
- Hard to analyze data during collection
- Data processing at end limited to one node (HDF5 1.8)

User feedback...

- Not favourable
- Per-image analysis for grid scans at end of collection big problem for big grid scans
- Retrograde step compared with previous detector systems
- "560 Hz Eiger 2XE slower than 100 Hz Pilatus"
- This feedback was not wrong







"the hack" - qualities

Nice

- Can acquire data "flat out" but briefly e.g. 1,500 2,000 frames
- Can use CBF parallel infrastructure
- Can analyze during acquisition

Nasty

- Extra load (~ doubled) on file system
- Limited to 2,000 frames or so no good for SFX

Key point: it *WORKED* and this hack was in production for 2 years

Meanwhile...

- HDF1.10 became commonplace in the wild
- Features added to HDF5 1.10.5 to check chunk size
- 1.10.5 features in h5py 3.0+ (rather recent) get_chunk_info_by_coord
- Push for SSX with Eiger

... time to revisit live analysis

SWMR, VDS, HDF5

SWMR, VDS

- SWMR single writer, multiple reader -> treat the HDF5 file as a directory, add "files" to it (i.e. add chunks to a dataset)
- VDS virtual data set define /entry/data/data to *look* like a real data set but it points into data_000001.h5 etc.

Using SWMR and VDS

- Master file defines VDS -> we know what images will appear
- Data files written as SWMR from ODIN
- Extension to "file watcher" to
 - Watch for appearance of data_00000N.h5
 - Watch data_00000N.h5 for appearance of frames (i.e. size > 0)
 - Trigger analysis on logical frame (i.e. map from real chunk ID in data_00000N.h5 to N00j)

Diamond configuration



• No real limits

Nasty

• No real limits

Example case i04 Eiger 2XE 16M

• 86x73 grid scan – 6,278 x full 16M frames @ 1.99ms exposure

Scan took 22.3s

Real-world ~280 Hz

• 1 image per trigger on position readback



You're going to need a bigger-boat cluster

800 cores... not a whole cluster but...

Processing 6,278 images < 22s -> can keep up in real-time

09:54:21.715 - filewatcher started watching **09:54:31.657** - filewatcher found all images

09:54:21.811 - PIA started on first image **09:54:43.685** - PIA on last image

However...

SWMR does not fix everything



Discussion

- HDF5 better solution in long term than e.g. dumping CBF files for i04 example 6,278 frames were 108GB in CBF format, 10GB in HDF5
- Far fewer files (~3 orders of magnitude) happy GPFS
- Can easily keep up acquisition as fast as detectors can do
- Removed limiting factors on grid scan size -> need to provision more analysis resources
- All of the software (ODIN, dials etc.) open source so this should all be available
- Making the spot finding more compute efficient on our to-to list

Comments

- Removing limitations on what we can do with the detector highlights other issues e.g. processing resources etc.
- Displaying analysis for O(10k) and greater grid scans will require thought
- SWMR, VDS and HDF5 *appear* to be stable in production on MX beamlines at Diamond
- We can only do this because we have Eiger 2 XE, ODIN, 40 Gb network, GPFS, infiniband in cluster, ... the software is only part of the solution
- Given the compute resources no obvious blocker for ~ real time analysis