Coping with BIG DATA image formats: integration of CBF, NeXus and HDF5

Herbert J. Bernstein[a], Jonathan M. Sloan[b], Graeme Winter[b], Tobias S. Richter[b], NeXus International Advisory Committee[c], Committee on the Maintenance of the CIF Standard[d]

[a]*Department of Mathematics and Computer Science, Dowling College, Oakdale, NY 11769*

[b]*Diamond Light Source, Harwell Science and Innovation Campus, OX11 0DE (UK)*

[c]*http://wiki.NeXusformat.org/NIAC*

[d]*http://www.iucr.org/resources/cif/comcifs*

Correspondence email: yayahjb@gmail.com

Introduction

This is an update to a July 2013 report of the same title (Bernstein, Sloan *et al.* 2013).

The BIG DATA demands of the new generation of X-ray pixel array detectors necessitate the use of new storage technologies as we meet the limitations of existing file systems. In addition, the modular nature of these detectors provides the possibility of more complex detector arrays, which in turn requires a complex description of the detector geometry (for example, see the companion article "XFEL Detectors and ImageCIF", this issue). Taken together these give an opportunity to combine the best of CBF/imgCIF (the Crystallographic Binary File), NeXus (a common data framework for neutron, X-ray and muon science) and HDF5 (Hierarchical Data Format, version 5, the high-performance data format used by NeXus) for the management of such data at synchrotrons. Discussions are in progress between COMCIFS (the IUCr Committee for the Maintenance of the CIF Standard) and NIAC (the NeXus International Advisory Committee) on an integrated ontology. A proof-of-concept API based on CBFlib and the HDF5 API is being developed in a collaboration among Dowling College, Brookhaven National Laboratory and Diamond Light Source. A preliminary mapping and a combined API are under development. Releases of CBFlib since CBFlib 0.9.2.12 can store arbitrary CBF files in HDF5 and recover them, support use of all CBFlib compressions in HDF5 files, and can convert sets of miniCBF files to a single NeXus file.

Data Rates, Formats and High Performance X-ray Detectors

Table 1. Typical Sustained Data Rates

Detector	Raw Image Size (MB)	Frame Rate (Hz)	Compressed Rate (Gb/sec)	USB Disk Data Rate (%)
ADSC Q315 (2x2 binned)	18	0.37	.013	7
Pilatus 2 6M	24	10	.48	240
Pilatus 2 Fast 6M	24	25	1.2	600
CSPAD	4.6	120	4.4	2208
Pilatus 3 6M	24	100	4.8	2400
Eiger 16M	72	125	18	9000

CCD X-ray detectors provide images at a moderate data rate of one every few to several seconds (see *e.g.* http://www.adsc-xray.com/Q4techspecs.html). Current higher performance X-ray detectors, such as the DECTRIS Pilatus, are capable of collecting six-megapixel images at 10 - 25 frames per second (Trueb *et al.* 2012), while the newest Pilatus3 6M instruments can operate at 100 frames per second (see https://www.dectris.com/pilatus3_specifications.html). The coming next generation of high performance X-ray detectors for MX such as the DECTRIS Eiger will be capable of collecting 16+ megapixel images at more than 125 frames per second (Willmott 2011) (Johnson *et al.* 2012). The ADSC DMPAD is also expected to produce 900 fine-sliced images in steps of two-tenths of a degree at 125 frames per second (Hamlin, Hontz and Nielsen 2012). The Cornell-SLAC pixel array detector (CSPAD) for XFELs produces 120 2.3 megapixel frames per second, using 2 bytes per pixel (Hart, Boutet *et al.* 2012). Note that gain-corrected CSPAD images use 8 bytes per pixel. Table 1 shows typical sustained data rates for detectors used for MX at NSLS, DLS, etc. compared to uncompressed XFEL rates (likely to decrease with suitable compression) and expected rates from Eiger, expressed in terms of the typical data rate for an inexpensive USB disk of 25 MB/sec = 200 Mb/sec. A data management system designed for very large numbers of files as well as for very large data volumes and data rates is needed (Fig. 1). Efficient recording of metadata coordinated with the data is also needed, and database access to information about images and experimental runs is needed. For MX, these data rates, data volumes, numbers of distinct images and numbers of distinct experiments argue for a very organized, high performance infrastructure. HDF5 and NeXus provide the necessary organization of the raw data, and CBF provides the necessary organization of the associated metadata for subsequent processing as well as contributing useful compression algorithms.

Today for MX alone Diamond Light Source employs one Pilatus 2M, three Pilatus 6M fast and one Pilatus 3 6M, giving a combined data rate of over 1 GB/sec and over 200 files/sec, creating the need to manage hundreds of thousands of images each day. For the Advanced Beamlines for Biological Investigations with X-rays (ABBIX) that are being built for NSLS-II (Hendrickson 2012), just two of the beam lines, the Frontier Macromolecular Crystallography (FMX) beamline and the Automated Macromolecular Crystallography (AMX) beamline (Schneider *et al.* 2012), are expected to produce an aggregate of more than 94 terabytes per operational half day, 660 terabytes per week or 38 petabytes per year. The anticipated beamline flux is 10^{13} photons per second for FMX and 2×10^{13} photons per second for AMX, approximately 50 times the NSLS X25 and X29 fluxes. One subtle effect of these high fluxes is that there will be more photons per pixel in images, making them more difficult to compress.

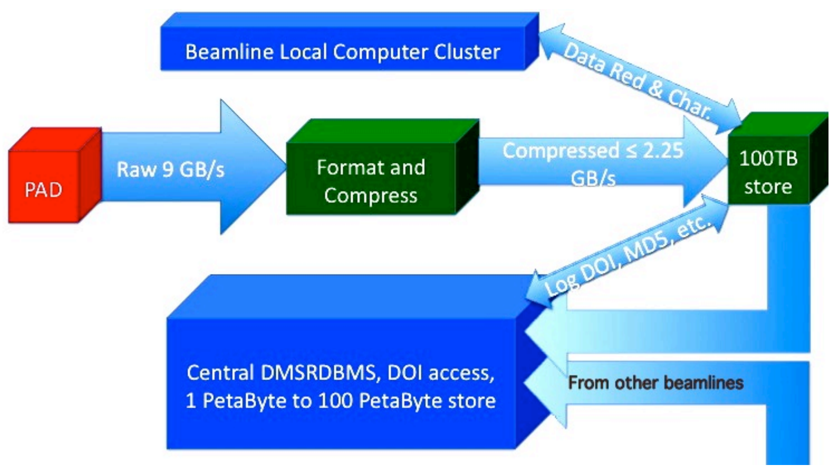


Fig. 1. Major data flows from the beamline advanced pixel array detectors (PADs) to the data management system relational database (DMSRDBMS). In order to be manageable, the raw 9 gigabyte per second data flow from each PAD needs to be compressed locally by at least 4:1, before going into a beamline 100 terabyte store for beamline local computer cluster access for up to a week for data reduction and characterization. The required bandwidth of the pipes from the beamlines to the DMSRDBMS depends on the compression used. If no further compression is used, 2.25 gigabyte (18 gigabit) per second per beamline network connections are required. If a combined lossless/lossy compression is used, then 225 - 450 megabyte (1.8 to 3.6 gigabit) per second per beamline network connections will suffice. The flows for transfers to user home institutions are not shown.

A final issue, in addition to the actual recording of data, is that of automated processing. At Diamond Light Source and elsewhere there has been a push towards the automated analysis of diffraction data, as interactively processing diffraction data at the current rate of typically 20 data sets per hour per beamline is unsustainable. This however places an increased strain on the file system, as typically the same data are read as many as six times in order to be processed, resulting in over 1000 file access operations per second. With the storage of many frames per file, as planned for NeXus, the rate of file operations would decrease substantially.

HDF5 and NeXus

The Hierarchical Data Format Version 5 (HDF5) is a self-describing file format with a robust, well documented API routinely handling multi-gigabyte files of data. It has a diverse user community covering a wide range of disciplines and is fully supported (Dougherty *et al.* 2009). HDF5 is particularly well suited to the management of very large volumes of complex scientific data and has been adopted as the primary data format in a wide range of disciplines (http://www.hdfgroup.org/HDF5/users5.html) and provides the "inner workings" of important frameworks, such as NetCDF (Rew *et al.* 2004) and NeXus. To avoid confusion we use the term *format* to describe the logical organization of data on a storage medium. An *ontology* is a dictionary of terms that may include descriptions of the relationships between terms. An ontology can be realized in one or more formats. We are therefore dealing with the HDF5 *format*, the NeXus *ontology*, a CBF *format*, and an imgCIF *ontology*. The HDF5 *format*, XML *format* and NeXus *ontology* together form the NeXus data transfer *framework*. The CBF *format*, CIF *format* and imgCIF *ontology* form the imgCIF data transfer *framework*.

HDF5 is tree-oriented, which is a very powerful and useful characteristic allowing file-system-like nesting of groups of data within groups of data, in order for information to be easily, reliably and efficiently searched. However, tables are more useful for loading information into a relational database management system (Codd 1970).

NeXus (Filges 2001) (Könnecke 2006) is a tree-oriented ontology for use wth HDF5 (and XML and HDF4) of importance in managing neutron and X-ray data. NeXus adds rules for storing data in files and a dictionary of documented names to HDF-5 in order to make HDF-5 applicable to the problem domain of synchrotron, neutron and muon scattering. NeXus is a convenient thin layer over HDF5 that is widely used at many physics research centers, including at synchrotrons. Together NeXus and HDF5 provide a portable, extensible and efficient framework for the storage and management of data.

Jan 2013 DECTRIS Eiger Workshop and Followup

The attendees at the January 2013 DECTRIS Workshop agreed on the use of an HDF5-based NeXus framework for the DECTRIS Eiger pixel array detector. The workshop charged Herbert J. Bernstein with following up on mapping additional terms to the new format. Tobias Richter, Jonathan Sloan and Herbert J. Bernstein have worked on a CBF-NeXus concordance and supporting software based on CBFlib and HDF5 with the cooperation of Bob Sweet, Graeme Winter and Mark Koennecke. Discussions with NIAC were held and then discussions with COMCIFS were held prior to ECM 28 in August 2013. There was general agreement that it was a good idea to have CIF and NeXus interoperate. COMCIFS and NIAC have agreed to start on a single crystal monochromatic macromolecular crystallography experiment NeXus application definition. An application definition in NeXus is a specification of the required metadata and data for that application. Significant progress has been made on the application definition and a draft will be available in Spring 2014.

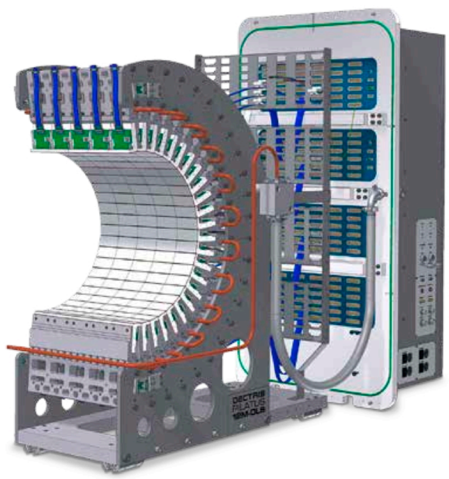


Fig. 2. Curved DECTRIS detector for DLS beamline I23, an example of a detector with a complex geometry best described using the imgCIF/CBF ontology. Prior to this work, NeXus could not support a detector such as this. Now it will be possible.

Mapping from NeXus to CBF

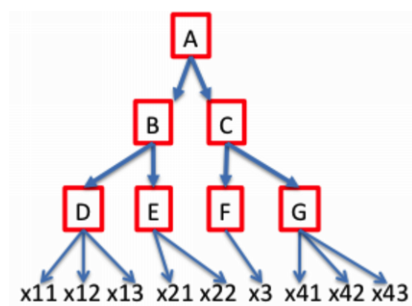
All NeXus base classes now have proposed slots in CIF categories. Handling of the DECTRIS-proposed Eiger HDF5 format is in the concordance. This concordance will require some relaxation of current NeXus name practices. "CBF_" prefixes are being used as an interim solution. Most of these prefixes are expected to be removed in the final version.

For this project, organizing data and metadata according to the conventions of the IUCr Crystallographic Information File (Hall *et al.* 1991) using imgCIF (Bernstein, Hammersley, 2005) and its open source supporting software CBFlib (Ellis, Bernstein 2005) provides a database-friendly tabular structure. The imgCIF ontology provides the metadata needed for the analysis of diffraction images and is supported by all the major detector manufacturers. This aspect is particularly important for instruments with complex geometries, e.g., the Pilatus 12M being constructed by DECTRIS for the long wavelength beamline I23 at Diamond Light Source (Fig. 2).

The embedding of CIF tables in HDF5 files was demonstrated at the "HDF5 as hyperspectral data analysis format" workshop in January 2010 (Götz *et al.* 2010). The workshop recommendation was, in part, "Adopt as much as possible from imgCIF and sasCIF".

Tables are easily embedded into trees. Going in the other direction is more difficult. There is serious effort required to make general trees into tables suitable for use in a relational database management system, involving a process known as "normalization"(Codd 1972). See Fig. 3.

One of the tasks of this project is to extend the imgCIF ontology to ensure workable database access to metadata in the HDF5 tree that has not already been normalized into CIF categories. For example, Digital Object Identifiers (DOIs) and SHA2 or SHA3 checksums from multiple experiments will need to be brought forward into a common table for post-experiment forensic validation.



Node	Parent
A	.
B	A
C	A
D	B
E	B
F	C
G	C

Node	Child
A	B
A	C

Node	Child
B	D
B	E

Node	Child
C	F
C	G

Node	Data
D	x11
D	x12
D	x13

Node	Data
E	x21
E	x22

Node	Data
F	x3

Node	Data
G	x41
G	x42
G	x43

Fig. 3. Example of a tree mapped into tables for database access identifying the links between parents and children in both directions as well as data. This is about the simplest example we can provide of a tree that demonstrates the need for "normalization" in the conversion from trees to tables.

CBF and Database Access

The Crystallographic Binary File (CBF) format is a complementary format to the Crystallographic Information File (CIF), supporting efficient storage of large quantities of experimental data in a self-describing binary format with a sophisticated description of the experimental geometry. For large PAD images, the raw binary CBF format is heavily used both within laboratories and for interchange among collaborating groups. When dealing with large numbers of independent experiments producing large numbers of CBF/imgCIF image files, HDF5 provides the virtual file system needed to manage the massive data flow. While it is feasible to simply encapsulate the CBF/imgCIF image files as opaque objects within an HDF5-based data-management system, active management of the data can be done more efficiently when the imgCIF tags are made visible in the HDF5 tree, a capability demonstrated in 2010.

With the anticipated throughput of NSLS II beamlines, and the current capabilities of MX beamlines equipped with pixel array detectors, the management of data and the possibility of interrogating the data files for experimental information becomes critical.

Compression Method	Compression Ratio	Relative Time
external bzip2 compression	20.4:1	5.6
HDF5 CBFlib canonical compression	15.7:1	3.9
HDF5 CBFlib nibble offset compression	11.5:1	2.9
HDF5 CBFlib packed V2 compression	11.0:1	2.8
HDF5 zip compression	9.7:1	2.4
external LZ4 compression (C1 one pass)	8.7:1	2.2
HDF5 CBFlib packed compression	8.6:1	2.2
external LZ4 compression (C0 two passes)	5.2:1	1.3
HDF5 CBFlib byte offset compression	4.0:1	1.0

Table 2. Tests on fifty image files from DLS with low to moderate pixel density. Total size 1.2 gigabytes. Relative times are shown. Multiply by 3 for the number of cores needed to keep up with the compression workload. The fifty files are from a set of 900 images recorded by Graeme Winter at Diamond Light Source beamline I02 as part of routine development work, and come from a crystal of DNA (TCGGCGCCGA) bound to a large ligand (lambda-[Ru(TAP)2(dppz)]2+). The aggregate of fifty was chosen to produce an uncompressed file small enough (1.2 GB) to be acceptable at user sites. Larger aggregates could be used for sites able to accommodate larger files.

Compression

There are long-standing issues about compression in crystallography. High-speed, high-compression-ratio compression is a critical issue for the next generation of detectors. Some compressions raise license issues. Some popular ones are slow or inefficient or both. Some can be handled in processing programs such as XDS if license and programming language issues can be addressed. Low pixel density fine-slicing with clean backgrounds makes some compressions more effective.

CBFlib provides useful compressions. See Table 2. A plugin module has been written to allow HDF5 to read and write CBFlib compressions. Starting with CBFlib release 0.9.2.11, that module is included. HDF5 1.8.11 and later is required. For general documentation on HDF5 dynamically loaded filters, see

http://www.hdfgroup.org/HDF5/doc/Advanced/DynamicallyLoadedFilters/HDF5DynamicallyLoadedFilters.pdf

The filter has been registered with the HDF5 group as 32006, and cbf.h includes the symbolic name for the filter CBF_H5Z_FILTER_CBF. The source and header of the CBFlib filter plugin are cbf_hdf5_filter.c and cbf_hdf5_filter.h, respectively, in the CBFlib kit. To use the filter in C applications, you will need to include cbf_hdf5_filter.h in the application and have the cbflib.so library in the search path used by HDF5 1.8.11.

Each compressed image in the HDF5 file is in the same format as the MIME-headed compressed images in the corresponding CBF, so the Fortran image search logic used in XDS can be used directly on these files.

Where to Find Software and Documentation

Draft imgCIF/CBF version 1.7 dictionary that now includes information on going from CBF to NeXus: https://www.sites.google.com/site/nexuscbf/home/cbf-dictionary

PDF summary of the concordance: https://www.sites.google.com/site/nexuscbf/mapping-draft

CBFlib kit: http://downloads.sf.net/cbflib/CBFlib-0.9.3.3.tar.gz

that includes both Jonathan Sloan's utilities to convert sets of minicbfs into a single NeXus file and a plugin filter that supports the full set of CBFlib compressions in HDF5.

Conclusion

The essential first steps in the integration of CBF, NeXus and HDF5 have been taken. There is work still to be done in applying this work at beam lines and in data processing software. Collaborators are most welcome.

Acknowledgements

Our thanks to James Hester, Chair of COMCIFS, and Mark Koennecke, Chair of NIAC, for supporting and encouraging this effort and to all participants in COMCIFS and NIAC for helpful comments, and also to James Hester for contributing the section on ontologies and formats, a critical issue in this work. Our thanks to Nick Sauter and Aaron Brewster for extending this work to CSPAD. Our thanks for years of supporting efforts, at the BNL PXRR Group: Robert M. Sweet, Dieter Schneider, Howard Robinson, John Skinner, Matt Cowan, Leonid Flaks, Richard Buono; at DLS: Alun Ashton, Bill Pulford; and at the Dowling College ARCiB Lab Group: Mojgan Asadi, Kostandina Bardhi, Maria Karakasheva, Ming Li, Limone Rosa

Our thanks to DECTRIS, BIOIHDF, and the HDF Group

Our thanks to Frances C. Bernstein

This work was funded in part by NIGMS, DOE, NSF, PaNdata ODI (EU 7th Framework Programme)

References

Bernstein HJ and Hammersley AP (2005). "Specification of the Crystallographic Binary File (CBF/imgCIF)". In: S. R. Hall and B. McMahon, Eds., International Tables For Crystallography, Chap. 2.3, pp. 37 - 43, International Union of Crystallography, Springer, Dordrecht, NL.

Bernstein HJ, Sloan JM, Winter G, Richter TS, NeXus International Advisory Committee and Committee on the Maintenance of the CIF Standard (2013). "Coping with BIG DATA Image Formats: Integration of CBF, NeXus and HDF5." poster, American Crystallographic Association, 2013 Annual Meeting. Honolulu, HI.

Codd EF (1970). "A relational model of data for large shared data banks". Communications of the ACM, Vol. 13, No. 6, pp. 377 - 387.

Codd EF (1972). Courant Computer Science Symposium 6, Chap. "Further Normalization of the Data Base Relational Model", pp. 33 - 64. Prentice-Hall.

Dougherty MT, Folk MJ, Bernstein HJ, Bernstein FC, Eliceiri KW, Benger W, Zadok E and Best C (2009). “Unifying Biological Image Formats with HDF5”. Communications of the ACM, Vol. 52, No. 10, pp. 42 – 47.

Ellis PJ and Bernstein HJ (2005). Definition and Exchange of Crystallographic Data, International Tables For Crystallography, Chap. “CBFlib: an ANSI C library for manipulating image data”, pp. 544 --556. International Union of Crystallography, Springer, Dordrecht, NL.

Filges U (2001). “The new NeXus API based on HDF5". In: VITESS Workshop Berlin, 25 - 27 June 2001.

Götz A, Solé V, Madonna C and Maydew AF (2010). “ELISA VEDAC Workshop Report, Workshop Title: HDF5 as hyperspectral data exchange and analysis format, Grenoble, January 11th - January 13th, 2010”. http://vedac.esrf.eu/public-discussions/hdf5-workshop/workshop-report.

Hall SR, Allen FH and Brown ID (1991). “The Crystallographic Information File (CIF): a new standard archive file for crystallography”. Acta Crystallographica Section A: Foundations of Crystallography, Vol. 47, No. 6, pp. 655 – 685.

Hamlin RC, Hontz T, and Nielsen C (2012). “The New Dual Mode Pixel Array Detector" in: Meeting of the American Crystallographic Association, Boston, MA, 28 July - 1 August 2012, American Crystallographic Association, Abstract 11.01.1151.

Hart P, Boutet S, Carini G, Dubrovin M, Duda B, Fritz D, Haller G, Herbst R, Herrmann S, Kenney C, Kurita N, Lemke H, Messerschmidt M, Nordby M, Pines J, Schafer D, Swift M, Weaver M, Williams G, Zhu D, Van Bakel N and Morse J (2012). "The CSPAD megapixel x-ray camera at LCLS." Proceedings of SPIE 8504: 85040C-85040C-85011.

Hendrickson WA (2012). “NSLS-II - Status of the Life Sciences Program”. Tech. Rep., Brookhaven National Laboratory, X6A Science Advisory Committee, February 2012. http://protein.nsls.bnl.gov/mediawiki/images/e/e3/Hendrickson_2012.pdf.

Johnson I, Bergamaschi A, Buitenhuis J, Dinapoli R, Greiffenberg D, Henrich B, Ikonen T, Meier G, Menzel A, Mozzanica A, Radicci V, Satapathy DK, Schmitt B and Shi X (2012). “Capturing dynamics with Eiger, a fast-framing X-ray detector”. Journal of Synchrotron Radiation, Vol. 19, No. 6, pp. 19, 1001 -1005.

Könnecke, M (2006). “The State of the NeXus data Format”, Physica B: Condensed Matter Vol. 385–386, Part 2, 15 November 2006, 1343–1345, Proceedings of the Eighth International Conference on Neutron Scattering.

Rew R, Ucar B and Hartnett E (2004). “Merging netCDF and HDF5”. In: 20th Int. Conf. on Interactive Information and Processing Systems.

Schneider DK, Sweet RM and Skinner J (2012). “Projection of MX ABBIX needs at AMX and FMX for Data Acquisition, Data Processing, Software, Data Archiving and Networking". February 2012. Private Communication.

Trueb P, Sobott BA, Schnyder R, Loeliger T, Schneebeli M, Kobas M, Rassool RP, Peake DL and Broennimann C (2012). “Improved count rate corrections for highest data quality with PILATUS detectors”. Journal of Synchrotron Radiation, Vol. 19, No. 3, pp. 347 - 351.

Willmott P (2011). An Introduction to Synchrotron Radiation: Techniques and Applications. John Wiley and Sons, Chichester, UK. page 6.