PaN-data ODI

Deliverable D8.3

Draft: D8.3: Implementation of pNeXus and MPI I/O on parallel file systems (Month 21)

Grant Agreement Number	RI-283556
Project Title	PaN-data Open Data Infrastructure
Title of Deliverable	D8.3: Implementation of pNexus and MPI I/O on parallel file systems (Month 21) - Prototype Report
Deliverable Number	D8.3
Lead Beneficiary	STFC
Deliverable Dissemination Level	Public
Deliverable Nature	Report
Contractual Delivery Date	01 Jul 2013 (Month 21)
Actual Delivery Date	02 Sep 2013

The PaN-data ODI project is partly funded by the European Commissionunder the 7th Framework Programme, Information Society Technologies, Research Infrastructures.

Abstract

Implementation of pNeXus and MPI I/O on parallel file systems (Month 21) - Prototype Report

Keyword list
PaN-data ODI, Scalability

Document approval
Approved for submission to EC by all partners on 25.09.2013

Revision history

Issue	Author(s)	Date	Description
1.0	Bill Pulford	27 Aug 2013	Complete version for discussion
1.1	Diamond co-workers	31 Aug 2013	
1.2	Rudolf Dimper - ESRF	5 Sep 2013	Comments and suggested improvements
1.3	Frank Schluenzen - DESY	21 Sep 2013	Significant Revisions

Acknowledgements:

Tobias Richter (DLS), Jonathan Sloan (DLS), Mark Basham (DLS), Graeme Winter (DLS), Herbert J. Bernstein

Additionally all within the PaNData project particularly Work Package 5 for CFLib and the Nexus International Advisory Committee.

Table of contents

1. Introduction

The PANData ODI project sets out to optimize coordination between research groups working at one or more different large experimental facilities across Europe and with the potential of expanding its scope across the scientific world. There are a number of components to the project such as common authentication, application software and federated searchable data storage systems. This report relates to a joint research activity, Work Package 8 **Scalability,** that concerns standardization of file formats and research to identify supporting data storage architectures to optimize speeds and data storage capacity.

The timeline for this workpackage:

- D8.1: Definition of pHDF5 capable Nexus implementation – Software – Report Delivered Aug 2012
- D8.2: Evaluation of Parallel file systems and MPI I/O implementations - Report Delivered Aug 2012
- D8.3: Implementation of pNeXus and MPI I/O on parallel file systems - **This report**
    - Note that the second part (implementation on parallel file systems) is covered in report D8.5
    - Note that there is no D8.4 in the original work package
- D8.5: Examination of Distributed parallel file systems - To be delivered
- D8.6: Demonstrate capabilities on selected applications (Month 21 June 2013)
    - A demonstration application is distributed and is in daily use by many users and at a number of European facilities see DAWNScience.
- D8.7: Evaluation of coupling of prototype to multi-core architectures (Month 27 March 2013) - Report - Work continuing in the community.

The PaNData Europe project selected HDF5 as the most suited underlying standard data format and considered NeXus to be the starting point for a specification of file structures and application definitions to be integrated. The principal requirement was to build on this and implement an architecture that permits close collaboration between scientific users at geographically distributed sites. The focal point of the Work Package 8 was to implement an architecture that, in ideal cases, supports the concept that any data files should contain sufficient descriptive metadata that no further parameters are needed to perform data evaluation and analysis using NeXus enabled applications.

The following topics are covered in this report.

1. Implementation details of the underlying file format, the Hierarchical Data Format or HDF5, based on practical experience.
    - More advanced issues concerning the provision of high performance including parallel access is discussed on the companion report PANData D8.5
2. Definition of the NeXus metadata model describing the contents of the NeXus file and conditions or configurations during an experiment, particularly with respect to parameters essential for performing the continuing scientific data analysis.

- Many of the examples in the report are based on the Dawn Science collaboration between the ESRF, EMBL Grenoble, Isencia (A Belgian software house) and Diamond Light Source[1]. This continues to be a very valuable tool not only for data analysis but also for its ability to explore the contents of NEXUS files and ICAT repositories in one application.

3. Implementation details of HDF5 routines and architecture to implement a native interface to superimpose the NeXus metadata model.
    - This report is complementary to report D5.2 Virtual Laboratories in that it provides further practical experience and tools that have been developed
4. Implementation of a plugin framework that permits a standard application to interact with heterogeneous file formats through the NeXus applications programming interface.
    - Diamond has implemented a transparent link between NeXuS and ImageCIF[2].

2. Meetings and Workshops

- NOBUGS Meetings and Workshops – Diamond Light Source, 24-28 September 2012[3]
- Eiger Detector Workshop – SLS, 23-25 January 2013
- Computing Workshop – DESY, 4-6 March 2013[4]
- PaNdata ODI – LUND, 12-14 March 2013

3. Definition the NeXus metadata model

The number of scientific disciplines with NeXus compatible metadata models sufficient to support data analysis has been increased since report D8.1. The details and current status of the implementation is as follows:

1. Tomography and Imaging - Scientific metadata sufficient for reconstruction
2. Non Crystalline Diffraction (NCD) - Scientific metadata sufficient for data analysis
3. Small Angle X-ray Scattering (SAXS) - Scientific metadata sufficient for data analysis
4. Angle Resolved Photoelectron Emission Spectroscopy (ARPES) - Not fully tested but scientific metadata should be sufficient for data analysis.
5. Spectroscopy - Scientific metadata partially sufficient for subsequent data analysis
6. Macromolecular (MX) and Small Molecule Crystallography (SX) - Full automation fully available for some time but extra functionality could be provided by the CBF -> NeXus filter and NeXus software applications.

[1] See http://www.dawnsci.org/ for details.
[2] https://sites.google.com/site/nexuscbf/mapping-draft/Concordance_Summary_21May13.pdf
[3] http://nobugs2012.org/
[4] https://indico.desy.de//event/7333

Notes:

- The example screen shots in this section are provided by the DAWN science application. DAWN is a versatile, extendible data processing framework, a user driven joint development of PaNdata partners DLS and ESRF and as such is an in-kind contribution to the project. DAWN includes an ICAT data file explorer and a NeXus file browser to demonstrate the interfaces with ICAT (PANData work package 4) and the metadata provided by the data acquisition process. The integrated workflow engine also permits demonstration of automatic data processing pipelines. Due to DAWNs extensive capabilities, the tool will serve as the prime demonstrator in the following sections as well as others for related tasks of WP5.
- The latest NeXus metadata models for Tomography, NCD and ARPES can be downloaded from http://www.dawnsci.org/downloads/nexus-metadata-model

3.1. Graphical User Interfaces for Tomography, NCD and ARPES.

3.1.1. Tomography and imaging beamlines

Notes:

- The tomography and imaging data collection is now capable of streaming the individual images of a 3D dataset into a HDF5 container and of storing the data into the PaNdata and NeXus International Advisory Committee (NIAC)[5] accredited Tomography NeXus format.
- Full reconstruction to a Tiff stack can be realized through the NeXus formatted dataset. Utilizing a GPU cluster the full reconstruction of a tomography scan be realized in about 30 mins. We are actively developing more advanced tools to accelerate the process even further.
- A specific GUI is now available to make this process easier and more automatic than before.

[5] http://wiki.nexusformat.org/NIAC

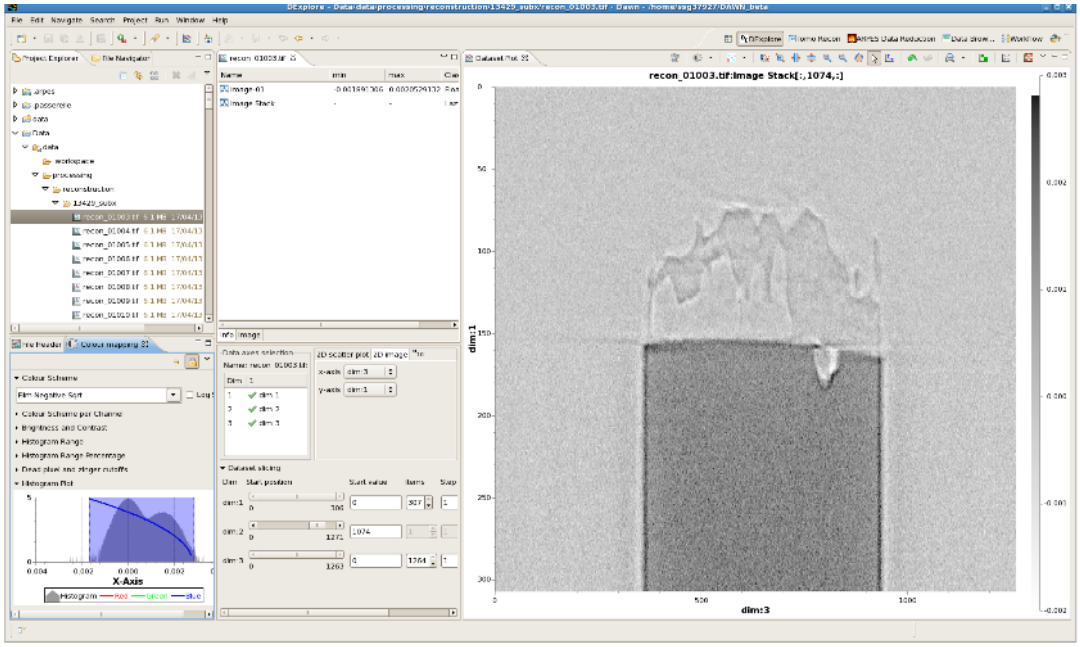

Figure 1: A screen shot of the calibration for Tomographic Reconstruction providing a high resolution image viewer, basic graphics and the relevant files sourced from the ICAT.

3.1.2. Non Crystalline Diffraction (NCD) and small angle scattering (SAXS)

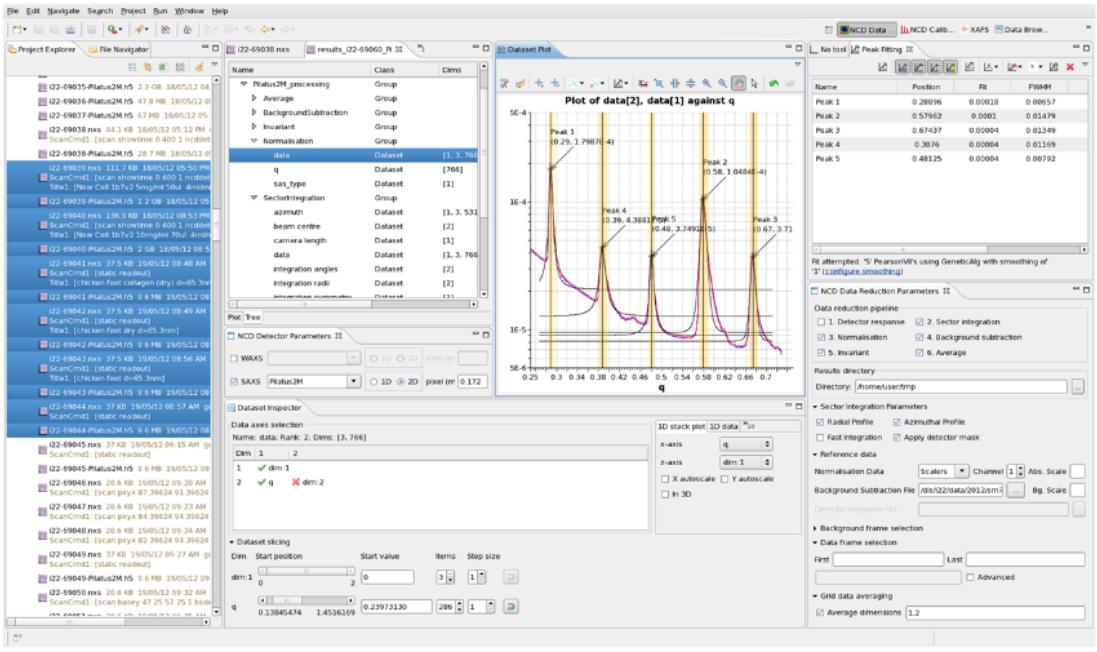

Figure 2: A screen shot of the calibration for Non Crystalline Diffraction. The basic details in this case are as in Figure 1 above but in addition there is a NeXus file browser to highlight the individual items in the current file.

3.1.3. Angle Resolved Photoelectron Emission Spectroscopy (ARPES)

Notes:

- ARPES allows direct resolution in energy-momentum space and hence imaging of the electronic structures of materials

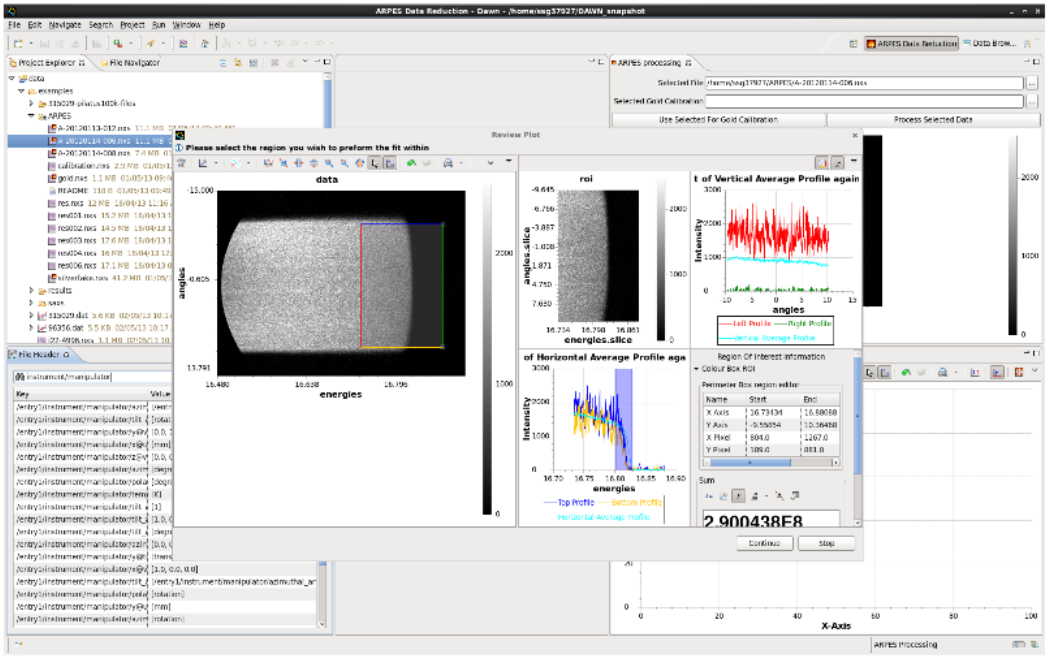

Figure 3: A screen shot of ARPES data reduction. A further example of the increasingly broad possibilities enabled by the appropriate use of ICAT and NeXus files.

4. The Implementation details of Hierarchical Data Format (HDF5)

This work involved an analysis of requirements for the NeXus library as specified in report D5.1 and D5.2 and the current state of deployment of NeXus at Diamond and DESY.

1. The vital functionality added to the NeXus definition is the addition of file links. The principal advantage conferred is that a NeXus file does not need to store all data but can have active links to component files.
    a. The essential architecture is for a head NeXuS file storing all appropriate metadata.
    b. This head file is a directory used by the NeXuS libraries to access the component data whether internally or in external files.
    c. Separating the NeXus metadata from the raw scientific data is sometimes essential to achieve the desired performance (like in the case of the Dectris Eiger detector) since it decouples the i/o from the metadata header – which is bound to be single threaded – from multi-threaded recording of the detector images. The separation has however also certain disadvantages since the corresponding files will appear as independent entities on the file system level. Object stores like CEPH[6] or dCache[7] might help to overcome this limitation by declaring a NeXus metadata header plus the raw data as an object on the file system level. Investigation of object stores is currently an on-going effort at ESRF and DESY.
    d. The development of transparent object-links inside the NeXus header also offers the possibility to use the native pHDF5 implementation effectively as a pNeXus implementation. It works both with the mpi io as well as the pHDF5 API offering a tremendous acceleration on appropriate parallel file systems.
2. A significant number of beamlines have been enabled writing NeXus files including NCD, Spectroscopy and Tomography. New beamlines at DLS write NeXus by default
3. The metadata derived from the experiments gained substantially from the close coordination with the NeXuS International Advisory Committee (NIAC), particularly with respect to the inclusion of linked files in the NeXus metadata header..
4. Analysis tools are now available from both Diamond, ISIS and DESY; these read and write NeXus format files. Examples: DAWN science (Diamond/ESRF/EMBL), DPDAK (DESY), Mantid (ISIS).
5. A coordination of metadata understanding has been reached between NeXus metadata and imgCIF/CBF (IUCR standard) enabling possibility of scientific applications to be modified to use either format. An example Diamond has 17 beamlines that write NeXuS files and 7 that write CBF (primarily crystallography).

[6] http://ceph.com/
[7] http://www.dcache.org/

6. There is a continuing dialogue between the collaborating facilities and the Nexus International Advisory Committee to add the Metadata definitions found to be necessary from experience in many scientific disciplines.

4.1. Architecture for simple acquisitions.

The original model for NeXus was to provide a single file able to contain all data and associated metadata. These data items could originate from a number of different sources and be stored with associated metadata indicating such parameters as dimensionality and units that would be sufficient for further data evaluation and analysis.

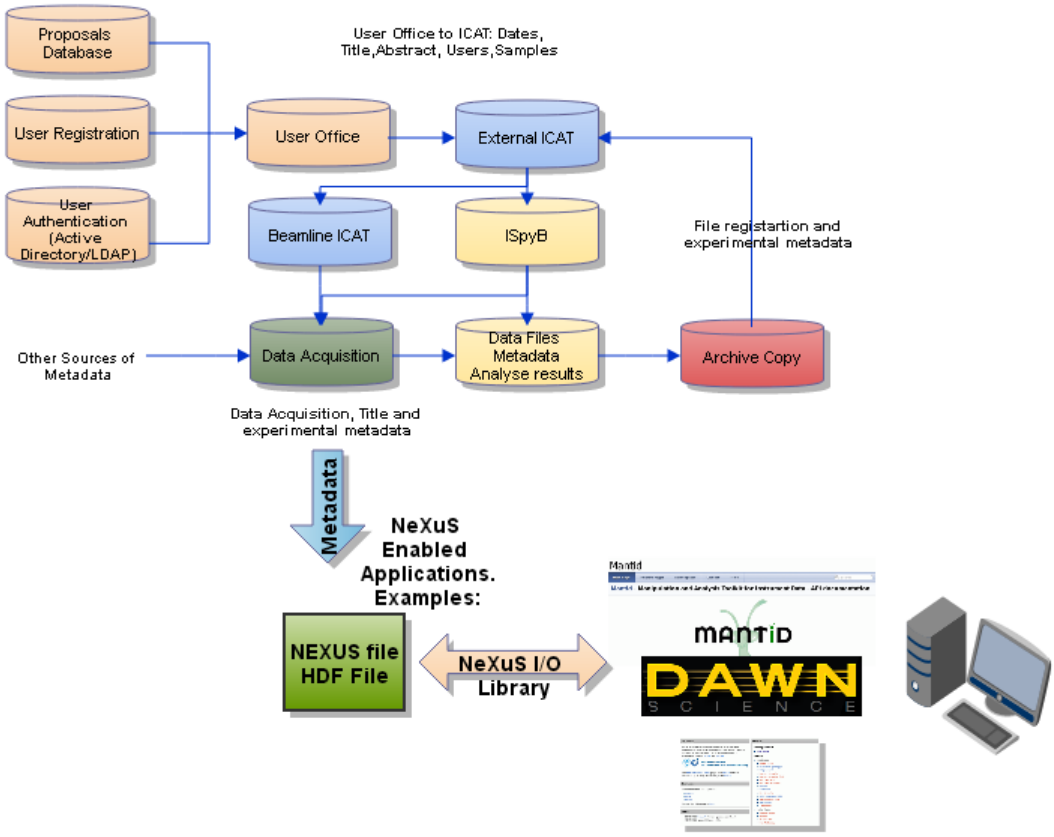

Figure 4 - A schematic representation of the processes contributing to the NeXus files generated in the small to medium data volume data acquisition process. Both the actual data and associated metadata are stored in the same file ensuring that the file administration is as simple as possible.

4.2. Architecture for heterogeneous high data rate acquisitions.

The above model remains the simplest to administer and can be used effectively for data acquisition resulting in NeXus files for small to medium data volume scientific investigations. However particularly for X-Ray synchrotrons and probably Free Electron Lasers this standard model becomes difficult to implement practically; contributing issues include:

- The data components originate from multiple detectors and other sources potentially at high data rates.

- These component data streams may have manufacturer specific formats such as TIFF.
- The detectors may be controlled by computers of diverse architectures.
- The data volumes may be very large; a tomographic dataset may easily exceed 120Gb.
- In these circumstances simultaneous data acquisition and evaluation would be very challenging in the one file model.

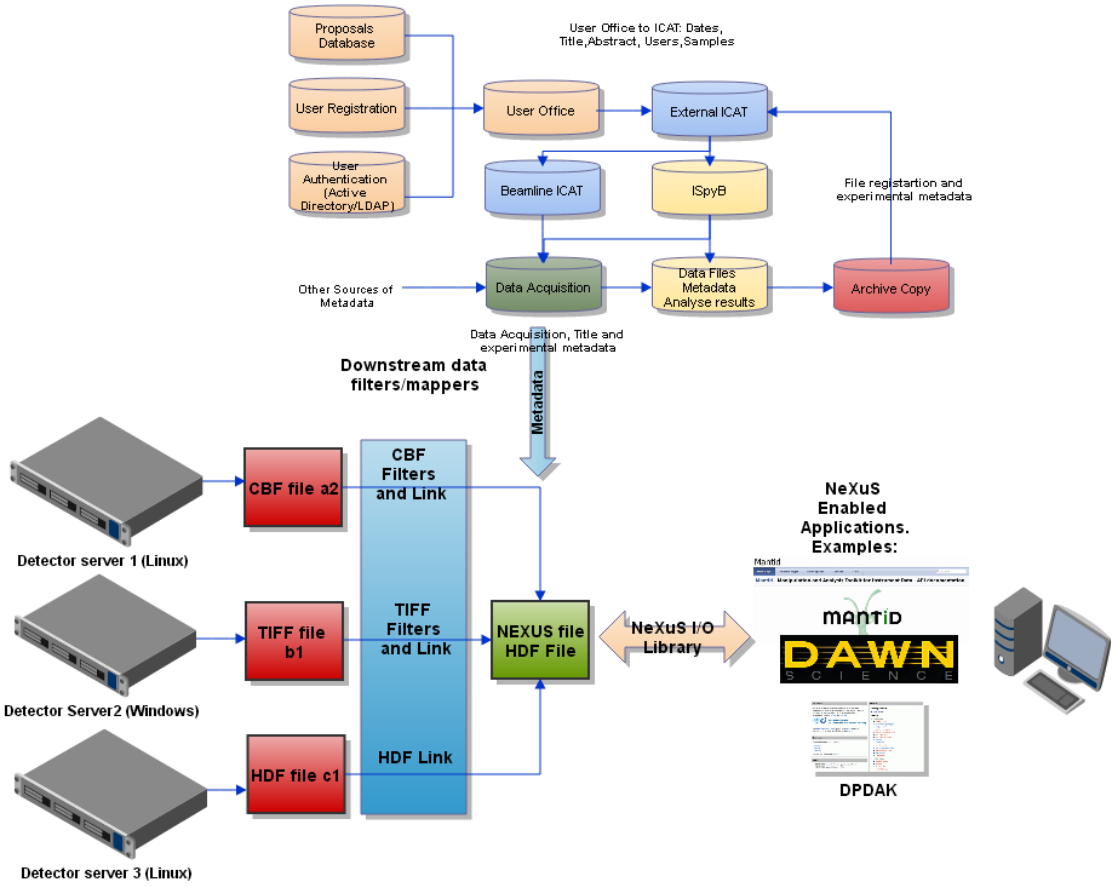

Figure 5 - A schematic representation of the processes contributing to the NeXus files generated in a large volume or very high speed acquisition process. The actual data are stored in the most efficient way possible subject to the timing and data volume constraints of the experimental process. The associated metadata are stored in one file leading to the actual data from an experiment existing in a number of files but all accessed through the NeXus metadata file. This allows the same applications to be used for both Figures 4 and 5 acquisition strategies.

5. Heterogeneous file format bridge - NeXuS to CBF

This project demonstrates an important strategy to expand the functionality of the NeXuS/HDF architecture particularly since it allows the integration of Small Molecule and Macromolecular Crystallography. The talk by H.Bernstein recently presented at the ECM28 in Warwick provides a nice overview of the efforts and use-cases for a high-level interoperability between CIF/CBF, NeXus and HDF5[8].

5.1. Completed and Delivered

5.1.1. CBF → NeXus mapping

This is completed and available on sourceforge.[9]

5.1.2. MiniCBF → NeXus mapping and conversion

MiniCBF files contain a unique header listing information which may not directly correspond with CBF definitions. This has been directly mapped for almost all parameters of a Pilatus version 1.2 header; which makes the conversion faster, less memory-intensive and potentially of higher quality – as an intermediate data format is skipped. A *minicbf2nexus* program has been released as part of CBFlib version 0.9.2.12. This has several important features:

- A choice of zlib compression or no compression for the main image dataset.
- The ability to convert some sample Pilatus version 1.2 miniCBF files, with some extra data provided by the user as the miniCBF file does not contain all of the information required to describe an experiment fully.
- The ability to pack the resulting NeXus data from any number of miniCBF files (within the limits of your machine, the libraries and file formats) into a single NX_entry group to allow them to be read by other tools. The output NeXus file is compatible with *Dawn*, so will almost certainly be widely readable.

5.1.3. CBFlib compression filters

CBFlib compressions have been implemented as a dynamically loaded plug-in built as part of the CBFlib library. These will be used in several places to allow the NeXus output of any data conversions to be used by existing software, such as *XDS*[10], which relies on the use of a specific compression scheme for the main dataset.

5.1.4. HDF5 abstraction layer

Within CBFlib an abstraction layer for calling HDF5 functions with a CBFlib-based error-code handling method has been implemented and documented. This allows some common tasks to be

[8] http://www.iucr.org/__data/assets/pdf_file/0007/80269/HJB_Managing_Crystallographic_Data.pdf
[9] http://sourceforge.net/projects/cbflib/
[10] http://xds.mpimf-heidelberg.mpg.de/

performed quickly and reliably, and is available to users of CBFlib as a generally useful way for them to interact with HDF5 files. It does not attempt to expose the full HDF5 API. An early version of this has been released with CBFlib 0.9.2.12. This has a few limitations compared to the development version but the majority of the functionality is present.

5.1.5. Tests

The *minicbf2nexus* program has a minimal test that verifies that it can correctly convert some arbitrary sample data; this conforms to the same style as the majority of the CBFlib tests.

5.2. Current Work

5.2.1. The implementation of precision-based floating point comparisons.

This will be open to further discussion but represents the current model in this implementation.

Most floating point comparisons use a very small constant number, or a small fraction of the numbers being compared, to ensure they are 'close enough'. Unfortunately, both of these methods can break down near zero: the first will use a 'delta' bigger than the numbers being compared; the second can result in an exact comparison whether there is a flush-to-zero chasm or not, and won't be at all obvious unless you happen to be an expert with floating point numbers.

The IEEE 754 standard was designed in a way that ensures the floating point number representation, when treated as a sign-magnitude integer, is strictly monotonically increasing. This allows the difference between any two floating point numbers – measured in the number of representable numbers between them – to be very easily extracted. This is a signed number, so requires one more bit than may be available. The larger number is trivially found by floating point comparisons so the maximum amount of data is available by finding the unsigned number of representable numbers between two given floating point numbers. This is a relatively new and/or unusual method of comparing numbers, but has applications in ensuring floating-point-based networked physics engines give precisely the same result on any given platform. This is one of the hardest parts of a notoriously difficult problem, which clearly shows that this technique is effective. For applications in scientific analysis this means that the errors due to precision-limited number formats are correctly represented and can be used to provide more correct and often tighter bounds on the errors that are introduced by intermediate calculations. With iterative methods (particularly slowly converging methods) this can be incredibly important – it could feasibly make the difference between being able to claim a discovery or not

This has been implemented, along with a test program to verify that it works, but not yet released.

5.2.2. minicbf2nexus improvements

CBFlib compression support has been implemented, but it may or may not currently work as there is an issue with the filter plug-ins in HDF5 1.8.11-pre2. As this uses a very new feature in the HDF5 library some issues like this are to be expected. This program has a few other limitations, such as where to write the data within the NeXus file and not being able to read more than one image from a single miniCBF file, amongst others. These are being addressed.

5.2.3. CBF → NeXus conversion

This will use parts of the miniCBF → NeXus conversion program as a simple example and make use of the HDF5 abstraction layer to allow the full CBF → NeXus conversion to be implemented relatively rapidly.

5.2.4. HDF5 abstraction layer

This now includes a more general data comparison method allowing a parameter to be passed to a custom comparison function. This allows a much wider range of comparison functions to be used, including some which rival the abilities of C++ function objects. More checks are performed on the input data in some functions to avoid causing errors in the HDF5 library.

The HDF5 abstraction layer comes with a recently written set of fairly comprehensive tests which attempt to check pre-conditions, side effects and post-conditions for almost all of the functions it contains. This prints some information to help trace the error if anything fails, so should ease maintenance.

5.3. Future Work

5.3.1. NeXus → CBF mapping

This is mostly the inverse of the CBF → NeXus mapping, so much of this will already be done. A few extra pieces of data defined within a NeXus file may still need to be converted to CBF data before this is complete. The primary use for this would be to verify that the CBF → NeXus conversion is being performed correctly on arbitrary input data, as opposed to a carefully constructed input file designed to exercise each function.

5.3.2. NeXus → MiniCBF mapping

The miniCBF format is aimed primarily at allowing data to be read from detectors at high rates, and allowing it to later be converted to a full CBF-formatted file before processing. Taking into account the very limited amount of data available from a miniCBF file would strongly suggest that this conversion is of little importance beyond checking for errors; but this can be done without resorting to defining and implementing this conversion.

Please note that an accordance agreement has been reached and is attached to this report.

6. Conclusions and recommendations

The work reported on is complementary to that in report D5.2 but concentrates on:

1. The practical implementations of software libraries to read and write HDF files.
2. The definition of metadata to support particular scientific fields.
3. A mechanism that enables the direct and transparent incorporation of additional widely used data formats into a NeXus based data evaluation and analysis framework.

a. This is very important to exploit NeXus/HDF5 tools to operate on data of crystallographic origin and facilitate cross disciplinary data evaluation and analysis.

4. The continued elaboration of tools such as DAWN[1] o use NeXus access file sets; the key principle being that the metadata in one NeXus file should be sufficient to perform most common operations.

6.1. Requirements:

- Specifically the NeXus format file generated should be able to be interchangable between software applications running at the different facilities
- There must be standardization of science specific metadata to be stored in the data files so that any analysis can be performed with little or ideally no recourse to other external information.
    - Currently agreed standards

    a.

        i. Tomography and Imaging
        ii. Non Crystalline Diffraction (NCD) and small angle scattering (SAXS)
        iii. Angle Resolved Photoelectron Emission Spectroscopy (ARPES) - Directly resolves in energy-momentum space and hence image the electronic structures of materials
        iv. Small Molecule Crystallography (SAX)
        v. Macromolecular Crystallography (MX)
- This requires the availability of standard Input/Output libraries supporting as wide a range of software languages as possible
    - Languages to be supported
        i. C and C++
        ii. Java
        iii. Python
- NeXus enabled analysis software needs to widely used and available across collaborations
    - Examples
        i. DAWN - Dawnscience collaboration, Diamond Light Source, ESRF,
        ii. DPDAK – PNI-HDRI, DESY
        iii. Mantid - Mainly Neutron research, ISIS,SNS
        iv. Matlab, IDL, Origin - capacity to read HDF5 and hence NeXus files, but would require development of NeXus file handlers to make use of the incorporated metadata. After consultation with Mathworks (Matlab) and EXELIS (IDL) both will support HDF5 1.8.11 in the upcoming releases permitting the use of the new filter mechanisms (developments being co-funded by PaNdata partners)..

The principal commonality across all applications is the NeXus interface appropriately populated with metadata describing the acquisition process and agreed sufficient information to permit further analysis requirements. In essence the provision of this interface isolates the software applications from the actual structure of data lying behind.

6.2. NeXuS software library locations

- http://www.h5py.org/ - h5python, a version of python optimized to access HDF5 files and allow the use of additional tools such as numpy.
- http://sourceforge.net/p/cbflib/code-0/349/tree/ - cbflib -> NeXus
- http://www.opengda.org/ - contains a NeXuS data writer
- https://code.google.com/p/pni-libraries/ - a high performance library for directly reading and writing NeXuS files.
- http://cars9.uchicago.edu/software/epics/areaDetector.html - EPICS area detector - Software to provide a standard interface to area detectors from the EPICS controls system.
    - The detailed control of the detector is delegated to plugins within the EPICS area detector architecture; the plugins are normally written in C or C++.
        - The parallel hdf5writer is currently tailored to EPICS/Diamond requirements, however this is only superficial. The intention would be to abstract it out and publish it on our external website.

          The main issue is to abstract the TCP protocol from detector system to phdf5writer.
    - Given the plugin code it should be relatively straightforward to integrate into the LIMA architecture(Lima.blissgarden.org/applications/tango/doc/index.html) - (LIMA DEVELOPERS please comment)

6.3. The operating systems used.

- Linux – Normally RedHat Enterprise version 5 or latterly RedHat Enterprise version 6 (These are very similar to the versions Centos 5 and 6[11] that are freely available)
- Windows – Normally Windows server 2008 R2 (64bit)

[11] http://www.centos.org/