DOI: 10.17199/NOBUGS2016.16

"Manyo-Lib" Object-Oriented Data Analysis Framework for Neutron Scattering

Jiro Suzuki[1], Yasuhiro Inamura[2], Takayoshi Ito[3], Takeshi Nakatani[2] and Toshiya Otomo[1]

[1] High Energy Accelerator Research Organization (KEK), 1 Oho, Tsukuba-city, Ibaraki, Japan.

[2]Japan Atomic Energy Agency, 2-4 Shirakata, Tokai-mura, Naka-gun, Ibaraki, Japan.

[3]Comprehensive Research Organization for Science and Society, 162-1 Shirakata, Tokai-mura, Naka-gun, Ibaraki, Japan.

jiro.suzuki@kek.jp

Abstract. "Manyo Library" (ML) is a standard framework for developing data analysis software for neutron scattering experiments. The data analysis software based on ML has been installed on the 16 instruments in Materials and Life Science Experimental Facility of J-PARC. ML is a C++ class library which was designed so as to handle large scale data in high-performance and has common and generic analysis functionalities for neutron scattering experiments. Users can perform data reduction and analysis on the C++ library through the Python interface. Data containers for one, two and three dimensional histograms are prepared in ML. NeXus format files can be created on ML from the data containers with NeXus-C-API. In 2016 a new interface between the data containers and NeXus file was developed with HDF5 library on ML, and data input/output (I/O) rate between the present and new interfaces is compared. The I/O rate of the new interface is two or four times faster than that of the present interface.

1. Introduction

Japan Proton Accelerator Research Complex (J-PARC) is a proton accelerator complex, and Materials and Life Science Experimental Facility (MLF) is a user facility providing pulsed neutron and muon sources for experiments in J-PARC. Twenty-one instruments for neutron scattering experiments have been installed in MLF, but requirements of data reduction and analysis software for each instrument are different among many instruments and scientists. Thus the MLF data-analysis environment group decided to provide a software framework for developing data-reduction and analysis software which can be applied to each aim of instrument and scientist.

Our basic concept of analysis environment is to provide a framework which has common and generic analysis functionalities for neutron scattering experiments [1, 2]. The framework, "Manyo Library" (ML), is a standard framework for developing data analysis software. ML is working on the 16 beam lines which are shown in Figure 1. Because ML should be maintained and improved by many scientists, it is developed on object-oriented methodology. Origin of the name ML is "Manyo-shu" which is a Japanese classical and one of the most famous and oldest anthology edited in 7th and 8th centuries in Japan. The poems were composed by all sorts of people: emperors, warriors, fisherman, etc. We think that ML can contribute to developing data analysis software utilized by all sorts of scientists using neutron scattering instruments.

ML is a C++ framework and class library which was designed so as to handle large scale data in high-performance. The class library is wrapped by a Python user interface created by Simplified Wrapper and Interface Generator (SWIG). Users can perform data reduction and analysis

on the C++ library through the Python interface. Figure 2 shows a screen shot of the graphical user interface working on the chopper spectrometer, BL01, and the data-analysis software based on ML is called from the interface. ML is a software component in the software framework in MLF/J-PARC[3]. The framework is available on various operating systems: Linux, MacOSX and Windows, and the binary install packages have been provided.

Design of data containers is one of key issues of the framework; efficient and simple data containers are required, because it determines system performance including development efficiency of application software on the framework. The design concept was discussed and introduced in detail in the previous papers [1, 2].

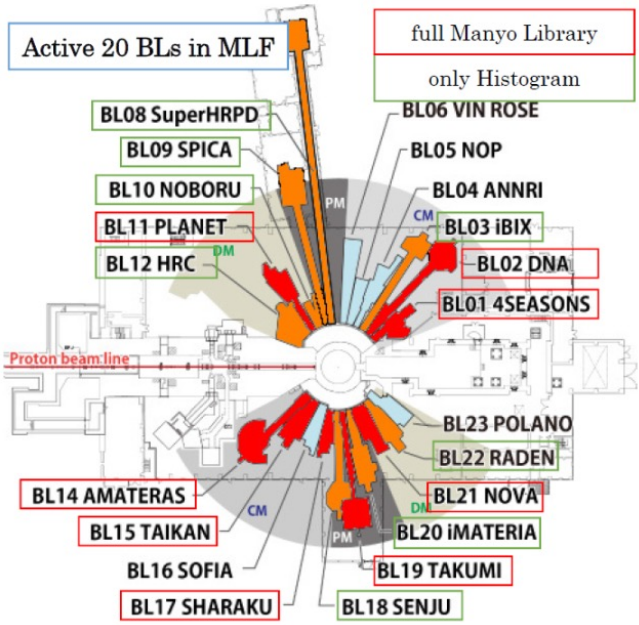

Figure 1: Floor layout of MLF and ML are working on the 16 beam lines indicated by orange and green boxes.

"ElementContainer" (EC) is a simple and fundamental data container in ML, and a one-dimensional histogram with its error values can be stored in an EC, where any number of vector<double> objects can be handled with their names. Basic mathematical arithmetic-operators and statistical operators provided in ML can operate ECs with error-propagation. Data containers for two- and three-dimensional histograms, "ElementContainerArray" (ECA) and "ElementContaienrMatrix" (ECM), are prepared, and raw data and analysis data are stored in the EC family on ML.

Figure 3 shows the structure of ECM. The number of vector objects in an EC is Nv. Any number of EC and ECA objects can be stored in an ECA and ECM, and the number of EC and ECA in a container is N_{EC} and N_{ECA}, respectively. The procedures of memory allocations and their garbage collections have been prepared in the containers. Users can build data analysis software in Python environment by using the operators prepared in the C++ library; ML provides a rapid prototyping environment for data analysis software without coding in C++ language.

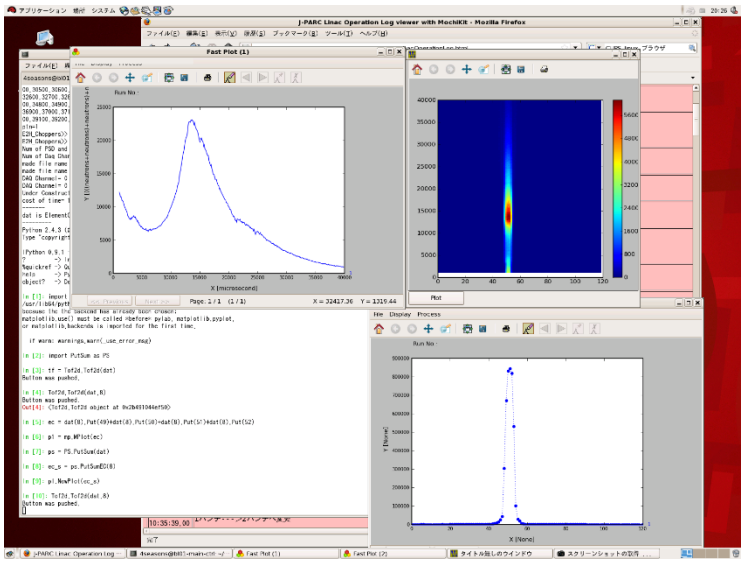

Figure 2: Screen shot of graphical user interface of the chopper- spectrometer in MLF.

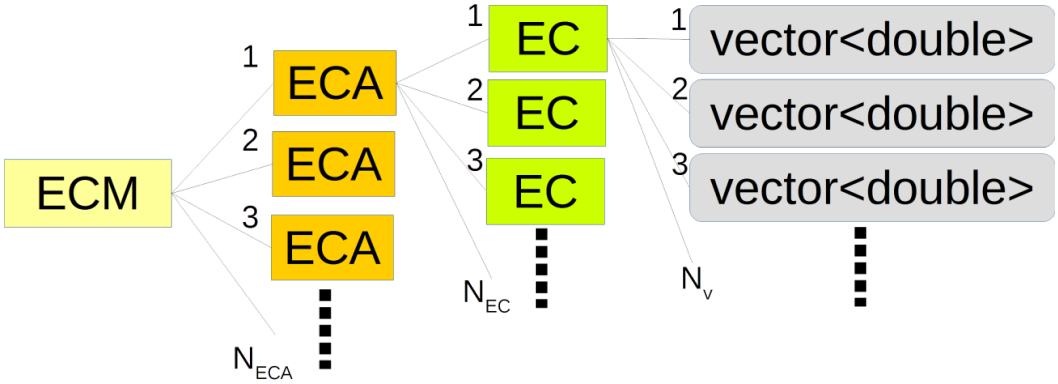

Figure 3: Structure of the EC family defined in ML. The number of vectors, EC and ECA in an EC, ECA and ECM are N_v, N_{EC} and N_{ECA}, respectively.

2. Data File Format

The size of data files handled on ML is increasing with increasing the intensity of proton beam producing pulsed neutron beam in MLF. At now we should concern how to increase the data input/output (I/O) rate on the framework. Histogram data handled on ML can be stored in two data format, (1) a binary format produced by the boost C++ serialization library (serialized format) and (2) the NeXus format [4] by the NeXus-C-API. The points in the issue are: (1) Data files in the serialized format cannot read on the other operating systems. For example, histogram data are stored in MLF with a Linux system, they cannot be read in their home laboratory with a Windows system. Because the data I/O rate is very efficient, temporary files during data analysis on ML are usually written in the serialized format. (2) Data files written in NeXus format can be read by the other operating systems, but the I/O rate for NeXus files on ML is smaller than that for the serialized files. NeXus file format is based on HDF file format [5], and defined and supported by the NeXus international advisory committee. Data files in NeXus format are produced in many neutron, muon and X-ray facilities. If data are stored in NeXus format, they can be read and shared by scientists in the world. We decided that standard data file format in MLF is NeXus, and we should resolve the issue of file I/O rate.

A new and simple data file structure was defined with keeping NeXus format and a new interface between the EC family and NeXus file was developed by using the HDF5 C-API library [4] on ML. The data structure is improved and simplified from the present structure, where the number of groups and the depth of the hierarchical structure in the file structure are decreased.

3. Result and Discussion

In the first half of 2016 we finished prototyping of the new interface on ML. The rates of reading and writing ECMs from/to NeXus files with the present and new interfaces were obtained and shown in Table 1. The file size on the disk are 4.0 and 1.3G bytes at (N_v,N_{EC},N_{ECA})=(3,100,300) and (3,10000,1), respectively. The structures and dimensions of the former and latter data sets are the same as those of three-dimensional histogram data produced by the chopper- spectrometer and the powder diffractometer in MLF. Each EC holds a one dimensional histogram data obtained by a pixel of two dimensional detector. One dimensional histogram can be described by a set of three vector objects: histogram bin, intensity and their error values. In this case the size of each vector stored in an EC is 4000 and the dimension of two dimensional detector are described by N_{EC} and N_{ECA}. The measurements were performed on an Ubuntu Linux (64bit) system with an Intel Core i5-4460 3.2GHz processor and 16 GB memory. The version of HDF5 library is 1.8.16. The I/O rates with the new interface are about 1.6 and 4 times larger than those with the present interface.

N_v	N_{EC}	N_{ECA}	Tw	Tr
3	100	300	24.7(37.9)	8.33(33.2)
3	10000	1	10.39(16.9)	2.76(11.46)

Table 1: Comparison between new and present interfaces in the efficiency of reading and writing NeXus files from/to ECMs. Tw and Tr are the writing and reading times in seconds with the interface developed in this study, and the numbers in the parentheses are those with the present interface. Nv, N_{EC} and N_{ECA} indicate the dimensions of ECM.

The I/O rate on ML can be increased by using the new interface, and we will develop and integrate it into ML. We also decided that the present interface should be kept in ML, because the data files in the file archiver in MLF should be read. Because the new file format developed in this study is in NeXus format, the files produced with the new interface can be shared by the other facilities in the world by using NeXus-API or HDF5 library.

References

[1] Suzuki J, Murakami K, Manabe A, Kawabata S, Otomo T, Furusaka M 2004 Nucl. Instr. and Meth. A 534 175-179

[2] Suzuki J, Nakatani T, Ohhara T, Inamura Y, Yonemura M, Morishima, Aoyagi T, Manabe A and Otomo T 2009 Nucl. Instr. and Meth. A 600 123-125

[3] Nakatani T, Inamura Y, Ito T, Harjo S, Kajimoto R, Arai M, Ohhara T, Nakagawa H, Aoyagi T, Otomo T, Suzuki J, Morishima T, Muto S, Kadono R, Torii S, Yasu Y, Hosoya T, Proceedings of ICALEPCS2009, Kobe, Japan https://accelconf.web.cern.ch/accelconf/icalepcs2009/papers/thc005.pdf

[4] http://www.nexusformat.org/

[5] https://www.hdfgroup.org/