Unified MuCh unpacker and Rootification of raw data

V. Singhal* and S. Chattopadhyay
[1] *VECC, 1/AF Bidhan Nagar, Kolkata-700 064, India*

Introduction

Unpacking is the first step of reconstruction which converts detector specific raw data into a common ROOT readable format which is termed as "rootification". This article describes the development related to the MuCh (muon chamber) unpacking software. mCBM is the phase-0 version of the CBM and has been taking data since 2018 with different colliding species at different interaction rates. During these years, 'unpackers' for the different subsystems in mCBM have been developed differently by each subsystem. This leads to incoherence between the subsystems and restrictions towards running in parallel with different subsystem. For a stronger integration of the unpackers, a common architecture has been introduced by separation between the framework and the algorithm part. Algorithmic part has also been subdivided into the algorithm and the configuration. Input of the unpacker is the raw hit messages gathered from data acquisition (DAQ) and output is the ROOT readable digis. 'Digi' is the smallest unit of the information which represent hit message of each detector pad. This digi must be stored using only the standard digi classes in order to be compatible with a reconstruction chain already running on the simulated data. To make unpacking process as light as possible and to run in real-time, the monitoring of the raw data has been separated out and moved to a separate class. Monitoring is performed only if an instance of it is added to the unpacking.

Implementation

During the 2021 mCBM campaign, a unified unpacking scheme has been developed [1] by separation between the framework bound part and the pure algorithmic part. It also provides stronger integration of the different subsystem's unpackers. Following the common architecture, a new unpacker has been developed for the MuCh subsystem. According to the above description three classes have been developed i.e *CbmMuchUnpackAlgo, CbmMuchUnpackConfig* and *CbmMuchUnpackMonitor* under the CbmRoot directory structure *reco/detectors/much/unpack*. The developed unpacker rootifies the mCBM raw data and the same will be used for CBM also. It uses the standard output digi container, *CbmMuchDigi* vector, to store digi information. It has been kept in the same format as for the CBM and the same digi container is used for simulation then reconstruction.

MuCh Digi and unpacking

MuCh Digi object stores only 3 basic information: a) time-stamp of the digi, b) unique address of the digi following *CbmMuchAddress* scheme, and c) charge of the hit in terms of ADC value. Time-stamp of the digi is computed based on the data format of hit message of Sts-MuCh-Xyter (SMX) chip and microslice format. Time-stamp of the digi is the relative time to the start time of the particular TimeSlice, instead of the absolute UNIX time. A TimeSlice is a container of raw data of all the subsystems for a fixed time interval and contains n number of microslices where n is according to optimal data rate flow. The spatial position information from the CRI (common readout interface) based data acquisition readout chain and the hit message of Sts-MuCh-Xyter (SMX) chip, are extracted and packed in a unique 32-bit integer value following *CbmMuchAddress* scheme during the unpacking. This spatial address value, relative hit time and ADC information are packed in the *CbmMuchDigi* container. Note: In the unpacking

*Electronic address: vikas@vecc.gov.in

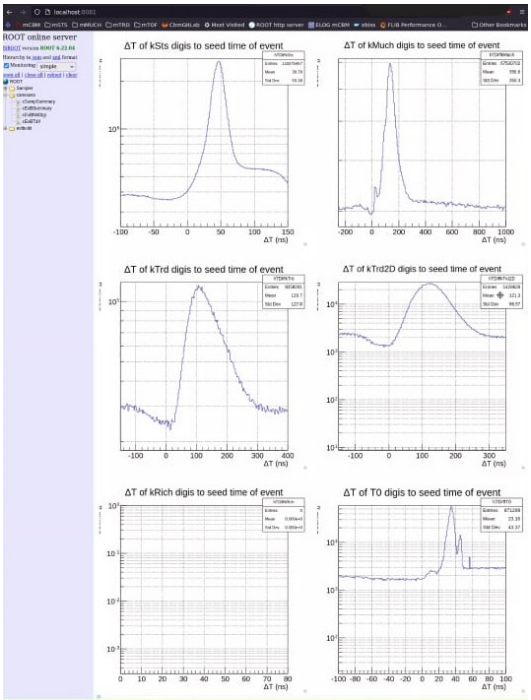

FIG. 1: Generated online time-correlation plots between different subsystems in the mCBM setup. Top right plot shows the time-correlation of the mMuCh detectors (both GEM and RPC combined) using this unpacker.

process, raw DAQ related information are not present in the unpacked rootified file, therefore the ascii parameter file *mMuchPar.par* is important with respect to mapping of CRI to MuCh Detector position, Number of readout board (GBTx) connected to CRI, Number of FEBs connected to GBTx, FEB to channel mapping etc. This information is used for generating *CbmMuchAddress* for each digi. During 2022 data taking at mCBM, in the mMUCH setup, one layer of real-size RPC detector has been installed along with 2 real-size GEM modules. The channel number mapping information for the detectors has also been incorporated in the MuCh parameter file. Hit message from SMX chip contains e-link number and based on this, FEB id is generated in the *MuchUnpackConfig* class, which is used for generating spatial address information.

Conclusion

To summarize, a unified MuCh unpacker has been developed in which the unpacking approach has been revisited and translation of raw data into digis has been performed and the same is used for data taking with mCBM during March-April 2022. One of the main goals, a lighter structure which is easier to handle and to adjust towards parallel processing/multi-threading, has been achieved by unpacking data in a common process, in particular for MuCh using the standard output digi vector *CbmMuchDigi*, and also the digi vector is stored in root file. Using this unpacker algorithm, an Message Queue (MQ) device has been developed which is compatible for online usage and the first attempt has been made for performing time correlation between different subsystems via MQ based online event building. Fig. 1 is an online plot during data taking where an event builder device is running directly on the raw data stream using unified unpackers of different subsystems thereby demonstrating the validation of the implementation of the unpacker.

Acknowledgments

We thank P.-A. Loizeau and V. Friese for help during the implementation of the unpacker. We also thank entire mCBM team for consistent support for data taking during the mCBM campaign.

References

[1] P.-A. Loizeau *et al.*, “A new unified architecture for unpacking, applied to mCBM data“, CBM Progress Report 2021.