HDF5 Experience in DUNE

Barnali Chowdhury[1,*]
On Behalf of the DUNE Collaboration

[1]Argonne National Laboratory (US)[**]

Abstract. The Deep Underground Neutrino Experiment (DUNE) has so far represented data using a combination of custom data formats and those based on ROOT I/O. Recently, DUNE has begun using the Hierarchical Data Format (HDF5) for some of its data storage applications. HDF5 provides high-performance, low-overhead I/O in DUNE's data acquisition (DAQ) environment. DUNE will use HDF5 to record raw data from the ProtoDUNE-II Horizontal Drift (HD), ProtoDUNE-II Vertical Drift (VD) and a number of test stands. Dedicated I/O modules have been developed to read the HDF5 data from these detectors into the offline framework for reconstruction directly and via XRootD. HDF5 is also very commonly used on High Performance Computers (HPCs) and is well-suited for use in AI/ML applications. The DUNE software stack contains modules that export data from an offline job in HDF5 format, so that they can be processed by external AI/ML software. The collaboration is also developing strategies to incorporate HDF5 in the detector simulation chains.

1 Introduction

The Deep Underground Neutrino Experiment (DUNE), hosted by the U.S. Department of Energy's Fermilab, is expected to begin operations in the late 2020s. The primary physics goals of the experiment include 1) studying neutrino oscillations using a beam of neutrinos from Fermilab in Illinois to the Sanford Underground Research Facility (SURF) in Lead, South Dakota, 2) detecting and measuring the ν_e flux from supernova bursts in or near our galaxy, and 3) searches for physics beyond the Standard Model. DUNE will consist of a modular far detector (FD) located at SURF in South Dakota, USA, and a near detector (ND) located on site at Fermilab in Illinois. The DUNE detectors will be exposed to the world's most intense neutrino beam originating at Fermilab.

2 LArTPC (Liquid Argon Time Projection Chamber) Operation in DUNE Detectors

The DUNE FD will consist of four LArTPC detector modules. The four identically sized modules will be installed approximately 1.5 km underground. Each module will

*e-mail: bchowdhury@anl.gov

**Argonne National Laboratory's work was supported by the U.S. Department of Energy, Office of Science, under contract https://www.anl.gov/prime-contractDE-AC02-06CH11357

be contained in a cryostat that holds $14 \times 12 \times 58\ m^3$ volumes of liquid argon (LAr). The first two detector modules are described in Refs. [1–4]. Schematics of those modules are shown in Figure 1 [5]. The general operating principle of the LArTPC is illustrated in Figure 2 [6].

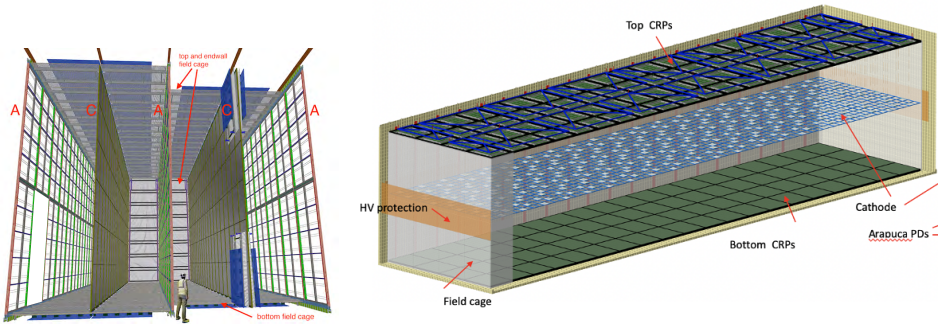

Figure 1. (Left) Schematic of 10 kt DUNE FD horizontal drift (HD) module, showing the alternating 58 m long (into the page), 12 m high anode (A) and cathode (C) planes, as well as the field cage that surrounds the drift regions between the anode and cathode planes. The modular anode and cathode planes are constructed of units called anode plane assemblies (APAs) and cathode plane assemblies (CPAs); the blank area on the left side was added to show the profile of a single APA. (Right) A 10 kt DUNE FD vertical drift (VD) module with CRP lining the top and bottom planes and ARAPUCA photon detectors mounted on the central cathode plane. Picture taken from [5].

2.1 ProtoDUNEs: Far Detector Prototypes

From 2018-2021, the DUNE collaboration built and operated two prototype detectors at CERN, single-phase (horizontal drift) and dual phase (DP) (liquid and gas, vertical drift), called ProtoDUNE-SP and ProtoDUNE-DP respectively. Each prototype represented ∼1/25 scale test of a DUNE FD module using identical components in size to those of the full-scale module. A second generation of prototypes, known as ProtoDUNE-II, are currently under construction at CERN, with two detectors, Horizontal Drift (HD) and Vertical Drift (VD), using the same LArTPC technology as the first two FD modules [7] for DUNE.

A modular anode plane (shown in Figure 1) is a collection of anode plane assembly (APA)s stacked on top of each other. Each APA consists of an aluminum frame with three layers (U, V and W) of active readout wire channels (displayed in Figure 2). The electronic readout system of ProtoDUNE-SP serviced 6 APAs and a Photon Detection System (PDS). An individual APA records 2560 readout channels. ProtoDUNE-DP recorded signals using two charge-readout planes (CRPs) during the 2019 run. CRPs perform charge readout using perforated PCB anodes with finely 440 segmented strip electrodes [7].

The first FD module, called HD, will read 150 APAs or 384,000 channels with 14-bit ADC values at about 2 MHz whereas the second FD module, called VD, will read 160 CRPs or 491,520 channels at the same rate as HD for every readout window [5].

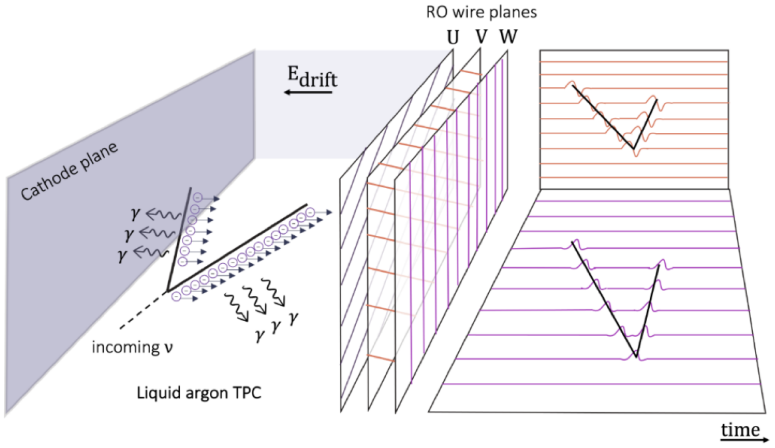

Figure 2. General operating principle of LArTPC illustrating the signal formation in three wire planes.

3 Raw Data Volume Estimates

All of the ProtoDUNE and DUNE LArTPC detectors have electron drift times from cathode to anode of the order of a few ms, which sets the time scale for which data is collected after a decision to trigger. Data, composed primarily of the digitized signals from the detector electronics, is collected and organized into "trigger records" (TRs) by the data acquisition (DAQ) system. Trigger decisions can be issued in response to various criteria, including signals from beam electronics and analyses of the raw data itself to determine if there is interesting activity in the detector.

ProtoDUNE-SP collected raw data over a period of six weeks in 2018. The total data size for a single TR was 180 MB. Lossless compression for the readout data was implemented in the DAQ, resulting in a final compressed trigger record size of about 75 MB. In total 850 TB of raw test beam data were written, along with 1 PB of commissioning and cosmic data [5]. ProtoDUNE-II intends to collect 20 M events for HD with expected data size of 140 MB/TR and 10 M events for VD with the data size of 110 MB/TR.

For FD data volumes, we use our ProtoDUNE-SP experience and assume that raw data sizes scale with the number of anode plane assemblies. Overall, the data volumes (dominated by calibration samples) are estimated to be 9.4 PB/year and 16 PB/year for the FD HD and FD VD modules respectively [5]. When two additional FD modules are added, the total data volumes are expected to be significantly large. However, the DAQ has a requirement to limit raw data across all modules and modes of operation to 30 PB/year [5]. The size of individual uncompressed trigger records will be three orders of magnitude larger than those from ATLAS/CMS, 3.8 GB for FD HD and 8 GB for FD VD, for cosmic ray and beam induced readouts.

4 Data Representation in Hierarchical Data Format (HDF5)

ProtoDUNE SP was operated between 2018 and 2020 and the data were represented through a combination of custom and ROOT [8] I/O based data formats. Issues with event (TR) sizes started to appear in long readout window noise runs and in simulation of beam events. Reconstruction struggled to handle events with large memory footprint stored in a ROOT [8] tree.

Due to the challenge of reading and processing the large trigger records and large datasets DUNE far detectors will produce, and the array-like structure of the LArTPC readouts, DUNE is considering exploring the effectiveness of HDF5 [9] as raw data storage in DUNE workflows and to evaluate its usability. In addition to HDF5, DUNE will support multiple data formats and I/O layers as required by data management and physics analyses.

The strength of HDF5 [9] is that its tabular structure allows very efficient "columnar" analysis of data (i.e. analysis of one or more variables from across a dataset). The format also has native support for parallel data access, and in particular for parallel reading and writing of compressed data [10]. This support extends to highly parallel file systems, such as those found in leadership computing facilities, and has been tuned to scale extremely well using the MPI protocol in these environments. For the larger DUNE event data, which can be highly compressed, this support for parallel I/O is advantageous, especially for the storage of raw data. As a result of these features, we expect to support the HDF5 format as part of our computing model in the future. For ProtoDUNE II, DAQ has already adopted HDF5 for storing raw data files due to its lower overhead and ability to write from multiple processes to a single file etc. DUNE DAQ has successfully written data from ProtoDUNE-II HD coldbox (APAs in coldbox with cold N2) and ProtoDUNE-II VD coldbox (CRP in coldbox with LAr).

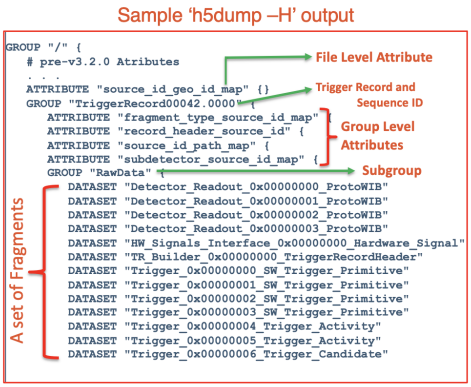

Figure 3. Sample output from a single ProtoDUNE II HD Coldbox HDF5 file.

5 Requirements to write HDF5 Data Formats from DAQ in ProtoDUNE II

We follow some key principles in terms of writing data fragments.

- *Granularity:* Datasets must have granularity that is meaningful and manageable for offline data processing.
- *Self-describing:* Files containing datasets should manifest what is in them and the information needed to know how to navigate and read them.
- *Backward Compatibility:* Tools for reading data should be backward compatible i.e. data formats and layout are versioned.

HDF5 provides a hierarchical data model similar to the directories, sub-directories and files in modern file systems. For ProtoDUNE-II data, this model is a "trigger record based" approach that divides data from trigger records into datasets within a file. Some of the key features of HDF5 data format are

- *Dataset:* DAQ data fragments are written as datasets.
- *Group:* DAQ data fragments can be organized into groups and subgroups to reflect organization based on sub-detector type or region in the file.
- *Attribute:* : Additional "metadata" can be attached as attributes at the file, group, or dataset level. The configuration information for the file layout is written as the attribute.

There exists a library *hdflibs* [11] for writing and reading files in the DAQ software stack. The library contains the classes used for interfacing between DAQ data applications (writers and readers) and the HighFive library [12], a modern header-only C++11 friendly interface, that provides C++ wrappers around HDF5 C API.

6 Navigating a ProtoDUNE-II HDF5 File

To navigate through an HDF5 file, each piece of data is recognized by 4 identifiers, 1) Run no., 2) Subrun no., 3) Trigger Record, and 4) Sequence ID. Figure 3 displays the general structure a ProtoDUNE-II coldbox HDF5 file and shows the current layout of how data is organized in files. Stored data are first "grouped" according to "TriggerRecord" (TR). A TR corresponds to one trigger decision. For ProtoDUNE-II, there are typically multiple TRs per HDF5 file. Each of those TRs contains a TR header and a set of data fragments. Each data fragment is represented by an HDF5 dataset, and has header, and data payload containing "raw waveform data" from different parts of the detector electronics. Each of those fragments can be read into memory in the offline and examined separately from all other fragments.

Trigger records can also be split into consecutive 'sequences' of fixed time, allowing for much longer data requests (up to 100 seconds for supernova burst (SNB) triggers) to be split in a way that is more easily handled in offline processing. These split trigger records appear in a similar structure as un-split ones. A supernova TR, collected by an FD module, yields 140-180 TB of uncompressed data, a potential challenge for DAQ [7]. Data from this TR will be spread across many files. Our goal is to design the HDF5 framework to be flexible and to adapt to changing technologies and evolution to incorporate future requirements.

7 Offline Read-in Software for HDF5

LArSoft/art [13] is DUNE's long-standing offline data processing framework for LArTPC data. Most DUNE art jobs use art ROOT files as an input because of *art*'s well-featured ROOT input source. However, with emerging new data formats, we needed to integrate HDF5 data processing with *art*. To read and process HDF5 data with offline software, *art* was equipped with similar functionalities as ROOT, for example, "delayed reading" (i.e. on demand reading). Delayed reading is performed by HDF5 "input source", that opens and closes the file(s) and leaves a file handle in the *art* event memory. This allows the framework to access individual datasets in the HDF5 file without having to read a whole trigger record from the file into memory while retaining *art*'s input and module execution scheduling. The HDF5 *art* input source is then coupled by decoder tools that perform the actual I/O, deserialize data, and extract the run, subrun and other relevant data products. Although all of the detectors share similar technology and DAQ systems, each requires a custom decoder tool due to differences in detector read-in requirements. Because of these implemented features the *LArSoft/art* algorithms can read and process portions of a trigger record at a time and support per-APA reading.

DUNE's large-scale offline data uses streaming via XrootD to take advantage of CPU at sites without large data stores. We have very recently demonstrated streaming an HDF5 file via XRootD in test mode using a dynamic library. Dynamic library uses LD_PRELOAD to load an XRootD POSIX I/O library. The XRootD POSIX library, when preloaded, allows an *art* job to run and read an HDF5 file. In contrast, ROOT directly links with XRootD libraries and does not rely on LD_PRELOAD. This is more robust, but to use the static library with HDF5 will require development of a Virtual File I/O interface for HDF5 that calls XRootD methods [14]. This work is in progress. Caching strategies for efficient streaming are also under crucial development.

DUNE is also developing DAQ-formatted raw digits for detector simulation. The existing mechanism uses the waterfall model i.e. all raw digits are stored in the memory first, and then serialized out to an HDF5 file. We have successfully simulated two "full FD-HD raw, non-ZS" trigger records in a 3.2 GB HDF5 file. There's an ongoing effort to stream raw digits out from WireCell [15] one APA at a time instead of waiting for all raw digits from all APAs to be produced and written. DUNE is exploring various other ways to simulate using HDF5, for example, converting GEANT4 Root file into HDF5 by using a stand-alone python code.

8 HDF5 Workflow in Current ProtoDUNE II Data Model

Figure 4 represents the current HDF5 workflow of the data analysis model. The output of DAQ is an HDF5 file that directly serves as input for traditional LArSoft algorithms, such as data preparation, WireCell, Pandora, and others. LArSoft produces *art* ROOT files. The complex data structures within those *art* ROOT files are then analyzed by ROOT-based programs. One of the applications produces CAF files [16], which are conventional Ntuples consisting of ROOT TTrees for final physics analyses.

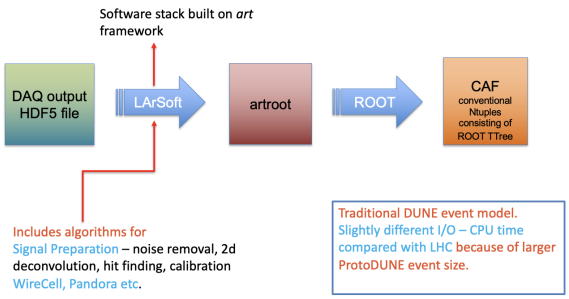

Figure 4. Existing ProtoDUNE II data model with HDF5 workflow.

9 Future Directions

With the growing computational needs of DUNE, next-generation efforts will likely rely on high-performance computing (HPC) clusters. The storage technologies like HDF5 perform better in an HPC environment. This section presents the future outlook of the DUNE computing and the role of HPCs. In Figure 5 we explore an improved data analysis model in light of utilizing HDF5 in an HPC environment. This model expects the DAQ to write raw HDF5 files in HPC friendly format. The DAQ output then experiences a "Signal Preparation" stage i.e. noise removal, 2D deconvolution, calibration etc., which again maintains an HPC/GPU friendly format. At this stage users will have two options. First, one can send those processed HDF5 files directly to Machine Learning (ML) algorithms for full event reconstruction/classification and further physics analyses. Such an effort will require the adaptation of our existing LArSoft algorithms to a new system but there will be large payoffs in terms of portability and increased efficiency in using HPC resources. Alternatively, one may potentially run traditional LArSoft algorithms for the reconstruction and analysis purposes in the absence of ML. .

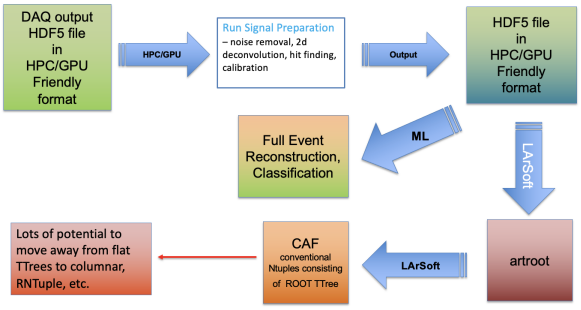

Figure 5. HDF5 Workflow of Potential ProtoDUNE II Data Model in HPC environment.

10 Summary

We have successfully implemented HDF5 as an intermediate raw data storage in the ProtoDUNE II workflow in preparation for DUNE. The existing framework is

now able to write and read-in all HDF5 datasets produced by DAQ(s). DUNE is also working on streaming HDF5 data with XRootD. There already exists a mechanism that allows writing DAQ-formatted raw digits for the FD simulation. DUNE computing is actively engaged with multiple other computing organizations, such as HEP-CCE, for exploring HPC/GPU friendly HDF5 data model, parallel I/O etc. with the intent of both drawing from and contributing to the community knowledge of computing solutions.

11 Acknowledgement

This document was prepared by the DUNE collaboration using the resources of the Fermi National Accelerator Laboratory (Fermilab), a U.S. Department of Energy, Office of Science, HEP User Facility. Fermilab is managed by Fermi Research Alliance, LLC (FRA), acting under Contract No. DE-AC02-07CH11359. This work was supported by CNPq, FAPERJ, FAPEG and FAPESP, Brazil; CFI, IPP and NSERC, Canada; CERN; MŠMT, Czech Republic; ERDF, H2020-EU and MSCA, European Union; CNRS/IN2P3 and CEA, France; INFN, Italy; FCT, Portugal; NRF, South Korea; CAM, Fundación "La Caixa", Junta de Andalucía-FEDER, MICINN, and Xunta de Galicia, Spain; SERI and SNSF, Switzerland; TÜBİTAK, Turkey; The Royal Society and UKRI/STFC, United Kingdom; DOE and NSF, United States of America.

References

[1] B. Abi et al. (DUNE) (2020), 2002.03005
[2] B. Abi et al. (DUNE), JINST **15**, T08008 (2020), 2002.02967
[3] B. Abi et al. (DUNE) (2020), 2002.03008
[4] B. Abi et al. (DUNE) (2020), 2002.03010
[5] A. Abed Abud et al. (DUNE) (2022), 2210.15665
[6] R. Acciarri et al. (MicroBooNE), JINST **12**, P02017 (2017), 1612.05824
[7] B. Abi et al. (DUNE), JINST **15**, T08009 (2020), 2002.03008
[8] R. Brun, F. Rademakers, *ROOT - An Object Oriented Data Analysis Framework* (2024), "https://root.cern.ch"
[9] The HDF Group, *Hierarchical data format version 5* (2000-2010), "http://www.hdfgroup.org/HDF5"
[10] A. Bashyal, P. Van Gemmeren, S. Sehrish, K. Knoepfel, S. Byna, Q. Kang (EP-CCE IOS Group), *Data Storage for HEP Experiments in the Era of High-Performance Computing*, in *Snowmass 2021* (2022), 2203.07885
[11] The DUNE-DAQ Group, *DUNE-DAQ hdflibs repository* (2020), "https://github.com/DUNE-DAQ/hdf5libs"
[12] Blue Brain Project, *HighFive - Header-only C++ HDF5 interface* (2015-2022), "https://github.com/BlueBrain/HighFive"
[13] LArSoft Project, Fermilab, *Liquid Argon Software (LArSoft) project* (2009), "https://github.com/LArSoft"
[14] Quincey Koziol, *HDF5 Virtual File Layer* (1999), "https://support.hdfgroup.org/HDF5/doc/TechNotes/VFL.html"
[15] DUNE Wirecell Projct, *Wire Cell Toolkit Configuration Data* (2017), "https://github.com/WireCell/wire-cell-data"

[16] C. Backhouse, *The CAFAna framework for neutrino analysis*, in *Snowmass 2021* (2022), 2203.13768