HDF5 Experience in DUNE

Barnali Chowdhury

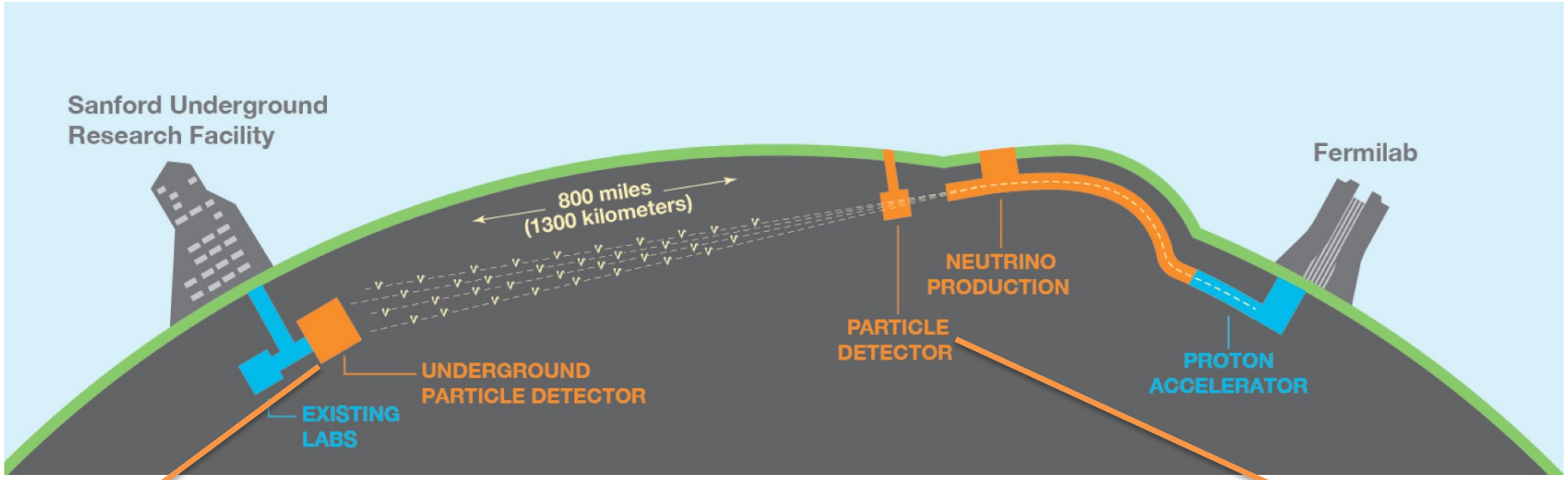
On behalf of the DUNE Collaboration

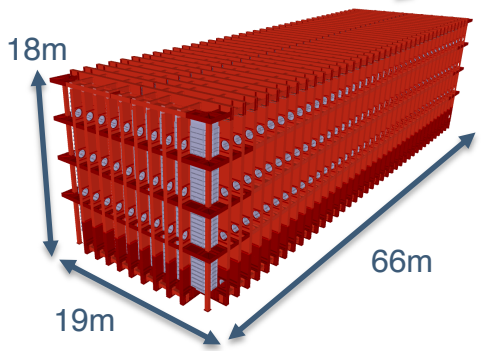
INTERNATIONAL CONFERENCE ON COMPUTING IN HIGH ENERGY & NUCLEAR PHYSICS

May 8th, 2023

Main Physics Goals of DUNE

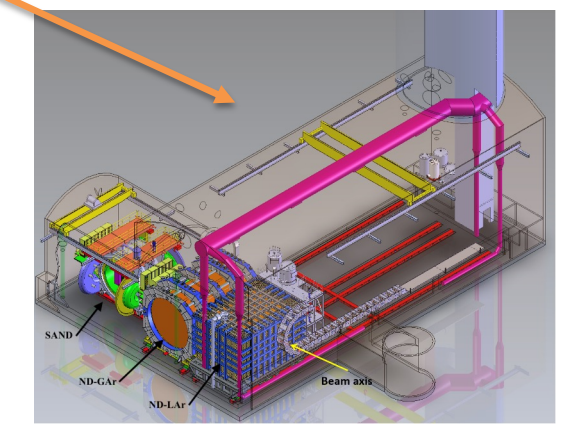
(picture not to scale)





One of the 4 FD modules

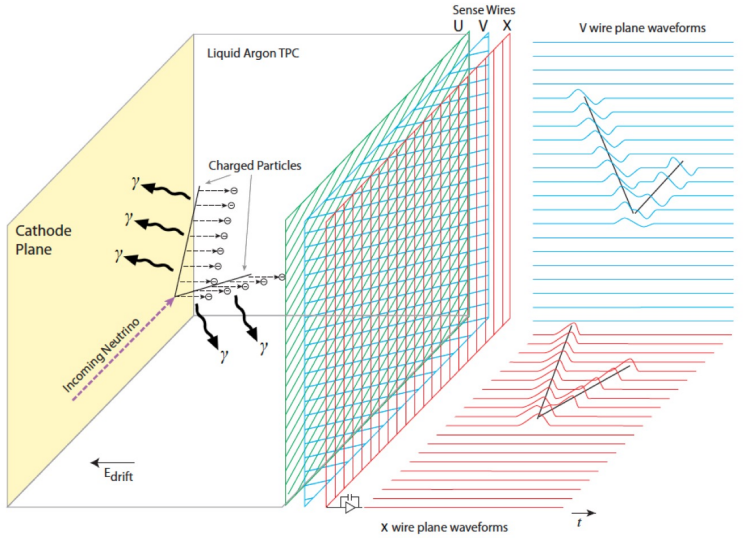
- Determine neutrino mass ordering, observe and measure CP violation (if it is present) in the neutrino sector.
- Sensitivity to neutrinos from astrophysical sources (solar, atmospheric, supernova burst) and BSM physics.
- Far Detector (FD) located 1.5 km underground and designed to be 4 10 kT Liquid Argon TPC at Sanford Underground Research Facility.
- Near Detector (ND) located 550m from proton target, 215ft deep, on-site at Fermilab.

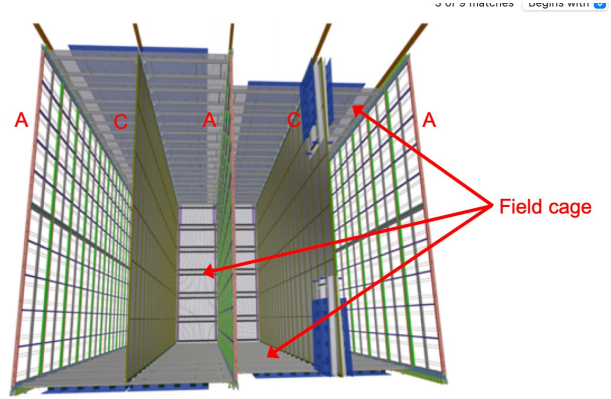


DUNE ND on-axis configuration

LArTPC (Liquid Argon Time Projection Chamber) Operation

- Fully active interaction medium of liquid argon
- Use scintillation and ionization to find 3D position of particles and interactions
- Drift charges recorded by 3 readout wire planes, with different orientations.
- One APA (Anode Plane Assembly) has 2560 read out wire channels
- The first FD module, “FD-Horizontal Drift” (HD), will read 150 APAs = 384,000 channels, 14-bit ADC values @ 2MHz
- The second FD module, “FD-Vertical Drift” (VD), will read 491,520 channels, 14-bit ADC values @ 2MHz
- TPC Raw data volumes will be dominated by the digitized waveforms from these channels.





Cartoon of DUNE FD SP module, showing the alternating anode (A) and cathode (C) planes that divide the LArTPC into four separate drift volumes.

DUNE Raw Data Volumes

- Prototyping is critical for DUNE
  - ProtoDUNE I was constructed and operated between 2018 – 2020 at CERN
  - ProtoDUNE II currently under construction at CERN
- Rate of beam data coming from ProtoDUNE is similar to rate of events from the full Far Detector.
- ~ 25 MB of uncompressed raw data from a single APA

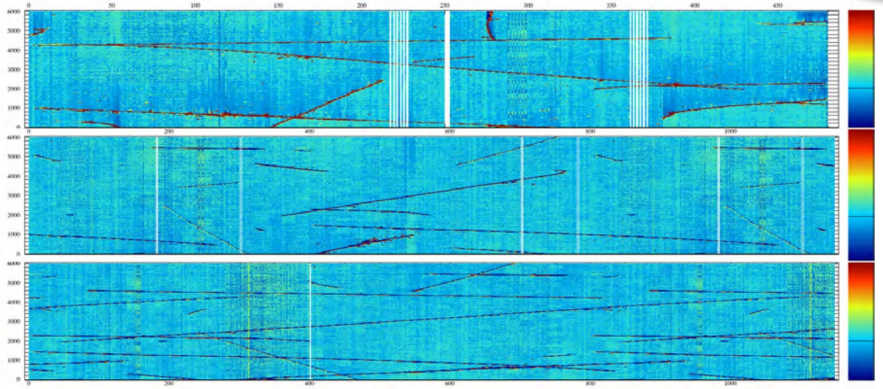
ProtoDUNE I Raw Data Volume

Quantity	Value	Explanation
Number of APAs	6	
Number of channels/APA	2,560	
Readout time	3 ms	
# of time slices	6000	
Single APA readout	23 MB	Uncompressed estimate
Full detector readout	178 MB	Uncompressed real
Full detector readout	70 MB	Compressed real
Effective compression factor	2.5	

Far Detector Raw Data Volume Estimate

Quantity	Value	Explanation
Far Detector Horizontal Drift		
APAs per module	150	DAQ spec.
TPC channels	384,000	DAQ spec.
TPC channel count per APA	2560	DAQ spec.
TPC ADC sampling time	512 ns	DAQ spec.
TPC ADC dynamic range	14 bits	DAQ spec.
FD module trigger record window	2.6 ms	DAQ spec.
Extended FD module trigger record window	100 s	DAQ spec.
Size of uncompressed trigger record	3.8 GB	DAQ spec.
Size of uncompressed extended trigger record	140 TB	DAQ spec.

Quantity	Value	Explanation
Far Detector Vertical Drift		
CRPs per module	160	DAQ spec.
TPC channels	491,520	DAQ spec.
TPC channel count per CRP	3,072	DAQ spec.
TPC ADC sampling time	512 ns	DAQ spec.
TPC ADC dynamic range	14 bits	DAQ spec.
VD module trigger record window	4.25 ms	DAQ spec.
Extended FD module trigger record window	100 s	DAQ spec.
Size of uncompressed trigger record	8 GB	DAQ spec.



Event display from 1 APA's worth of Raw Data is shown for ProtoDUNE-I trigger record collected in October 2018. ~25 MB of data.

Data Representation

- Issues in ProtoDUNE I
  - ROOT format was used for raw data storage in ProtoDUNE-I
  - Struggled to handle events with large memory footprint stored in ROOT tree
- DAQ decided to explore the use of HDF5 as a promising format for large, streaming raw datasets
  - Technical reasons – no need for ROOT data model support at raw data level, lower overhead, higher performance.
  - Multiple threads can write to the same file.
  - HDF5 is used commonly in ML applications and in HPC workflows.
- HDF5 will be used as raw data file format in ProtoDUNE(s)-II.
  - Have successfully taken data in HDF5 from HD, VD Cold Box testing. ND-LAr Module0 data is in HDF5 format.
- DUNE will support multiple data representation. Currently
  - HDF5 is an ingest format from the DAQ
  - HDF5 is an output file format for analysis Ntuples
  - ROOT serves as the mainline processing chain format
- All compression of raw data must be lossless due to the way charge is shared across wires/planes and ROIs are formed
  - We are trying to detect very small signals near the noise threshold
  - Need to preserve sensitivity to small energy depositions (~few MeV scale interactions)

Requirements to write in HDF5 from DAQ

- Raw data files from DAQ now being written in HDF5 format
- We write data fragments from detector to datasets without modification
  - Datasets must have granularity that is meaningful and manageable for offline data processing
- Data must be self-describing
  - Files should contain a 'manifest' of what is in them, and the information needed to know how to navigate and read them
- Tools for reading data must be backward compatible
  - Data formats and layout are versioned
- hdf5libs: library for writing and reading files in the DAQ software stack
  - https://github.com/DUNE-DAQ/hdf5libs
  - Makes use of HighFive, a C++ header-only product

HDF5 Data Format

- Key features of HDF5 that are currently being used
  - **Datasets**: DAQ data fragments are being written as datasets
  - **Groups**: DAQ data fragments can be organized into groups and subgroups to reflect some organization in the file
  - **Attributes**: additional "metadata" can be attached as attributes (at file, group, or dataset level)
- Data are in files with unique filenames
  - Each piece of data has four identifier numbers for
    - Run Number
    - Subrun Number
    - Trigger Record
    - Sequence ID
- Data fragments, within DAQ, are configured to have sizes that make network transfers efficient
  - Those sizes are reflected in the fragments seen by offline in HDF5 file
  - An important part of efficient data transfers is having block sizes that are sufficiently large. (transfers of many small blocks are not efficient)

Current Layout of HDF5 Data

- File Level Attributes are at the top level and under root group
- Within that, data is grouped as "TriggerRecord". A Trigger Record corresponds to 1 trigger decision.
- TriggerRecord consists of
  - TriggerRecord Header +
  - A set of Fragments
- Each Fragment corresponds to a part of the detector and consists of
  - Fragment Header +
  - Fragment Payload ->The Raw (unmodified) data from the detector electronics.

Sample 'h5dump –H' output

```
GROUP "/" {
   # pre-v3.2.0 Atributes
   . . .
   ATTRIBUTE "source_id_geo_id_map" {}
   GROUP "TriggerRecord00042.0000" {
      ATTRIBUTE "fragment_type_source_id_map" {
      ATTRIBUTE "record_header_source_id" {
      ATTRIBUTE "source_id_path_map" {
      ATTRIBUTE "subdetector_source_id_map" {
      GROUP "RawData" {
         DATASET "Detector_Readout_0x00000000_ProtoWIB"
         DATASET "Detector_Readout_0x00000001_ProtoWIB"
         DATASET "Detector_Readout_0x00000002_ProtoWIB"
         DATASET "Detector_Readout_0x00000003_ProtoWIB"
         DATASET "HW_Signals_Interface_0x00000000_Hardware_Signal"
         DATASET "TR_Builder_0x00000000_TriggerRecordHeader"
         DATASET "Trigger_0x00000000_SW_Trigger_Primitive"
         DATASET "Trigger_0x00000001_SW_Trigger_Primitive"
         DATASET "Trigger_0x00000002_SW_Trigger_Primitive"
         DATASET "Trigger_0x00000003_SW_Trigger_Primitive"
         DATASET "Trigger_0x00000004_Trigger_Activity"
         DATASET "Trigger_0x00000005_Trigger_Activity"
         DATASET "Trigger_0x00000006_Trigger_Candidate"
```

File Level Attribute

Group Level Attributes

Subgroup

Offline Read-in Software for HDF5

- DUNE uses larsoft/art, an event processing framework supported by Fermilab SCD for offline data processing. Most DUNE *art* jobs use art ROOT files as input.

- *art*'s ROOT input source is quite full-featured
  - Delayed reading: Input source does not read actual data, i.e., delayed until some "get" method is called

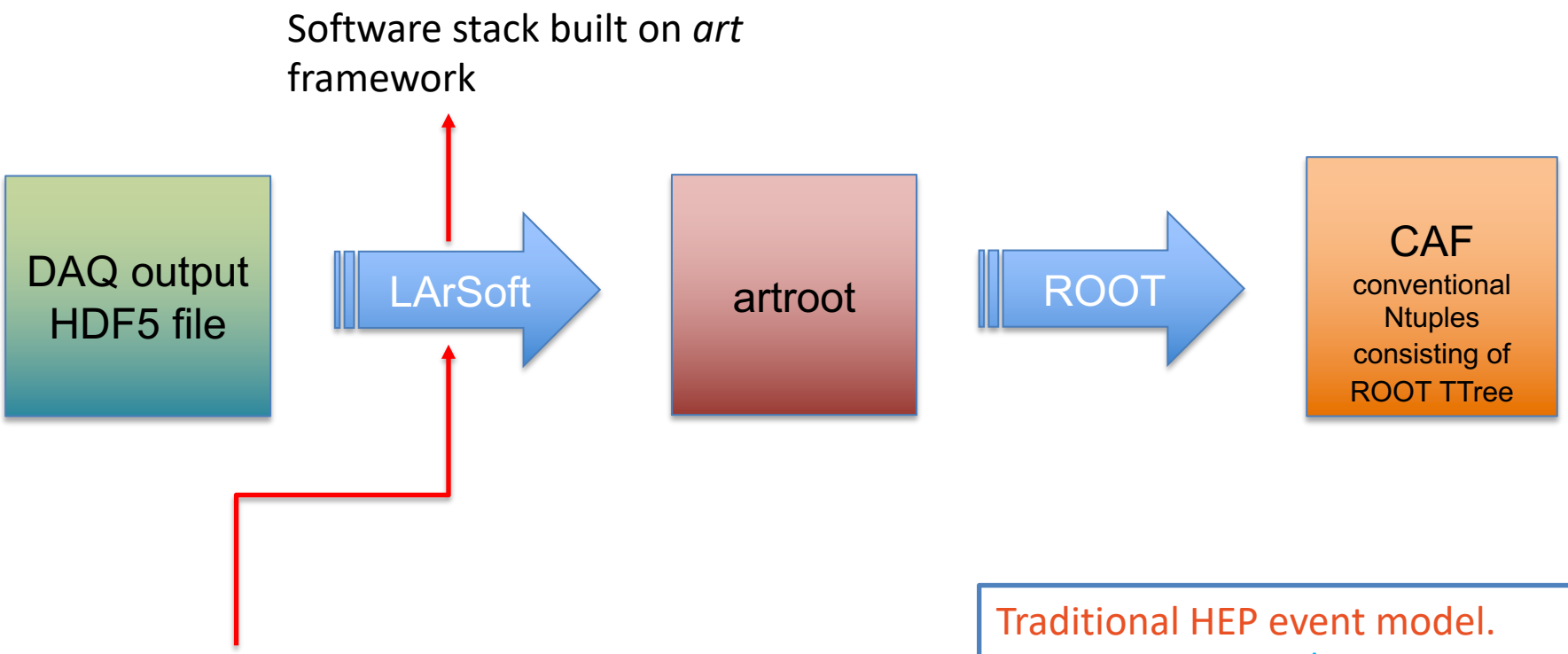
- To read HDF5 data offline with *art*,
  - Delayed Reading: Implemented for HDF5. Put proxies in the art event which lets downstream tools access the input file and deserialize it.
  - Input source just opens and closes the file(s) and leaves a file handle in the *art* event memory.
  - Decoder tools do the actual I/O and transform data into useable formats.

- Existing Input source and decoder tools for each detector/prototype
  - Vertical Drift Coldbox, Horizontal Drift Coldbox (December 2021 and new May 2022 versions) ICEBERG, Coming: ProtoDUNE-2-HD and VD, FD MC
  - Supports per-APA reading

Streaming HDF5 with XRootD

- HDF5 data is streamable with XRootD using the dynamic library.
- Dynamic library uses LD_PRELOAD to load an XRootD POSIX I/O library which replaces all the low-level methods: fopen, fread, fwrite, fclose, etc. Can be used with many applications, but not all.
- The XRootD POSIX library, when preloaded, allows an art job to run and read an HDF5 file. Also, “h5dump” is tested and works.
- 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 requires adjusting a set of Virtual File Layer I/O methods. This work is in progress.

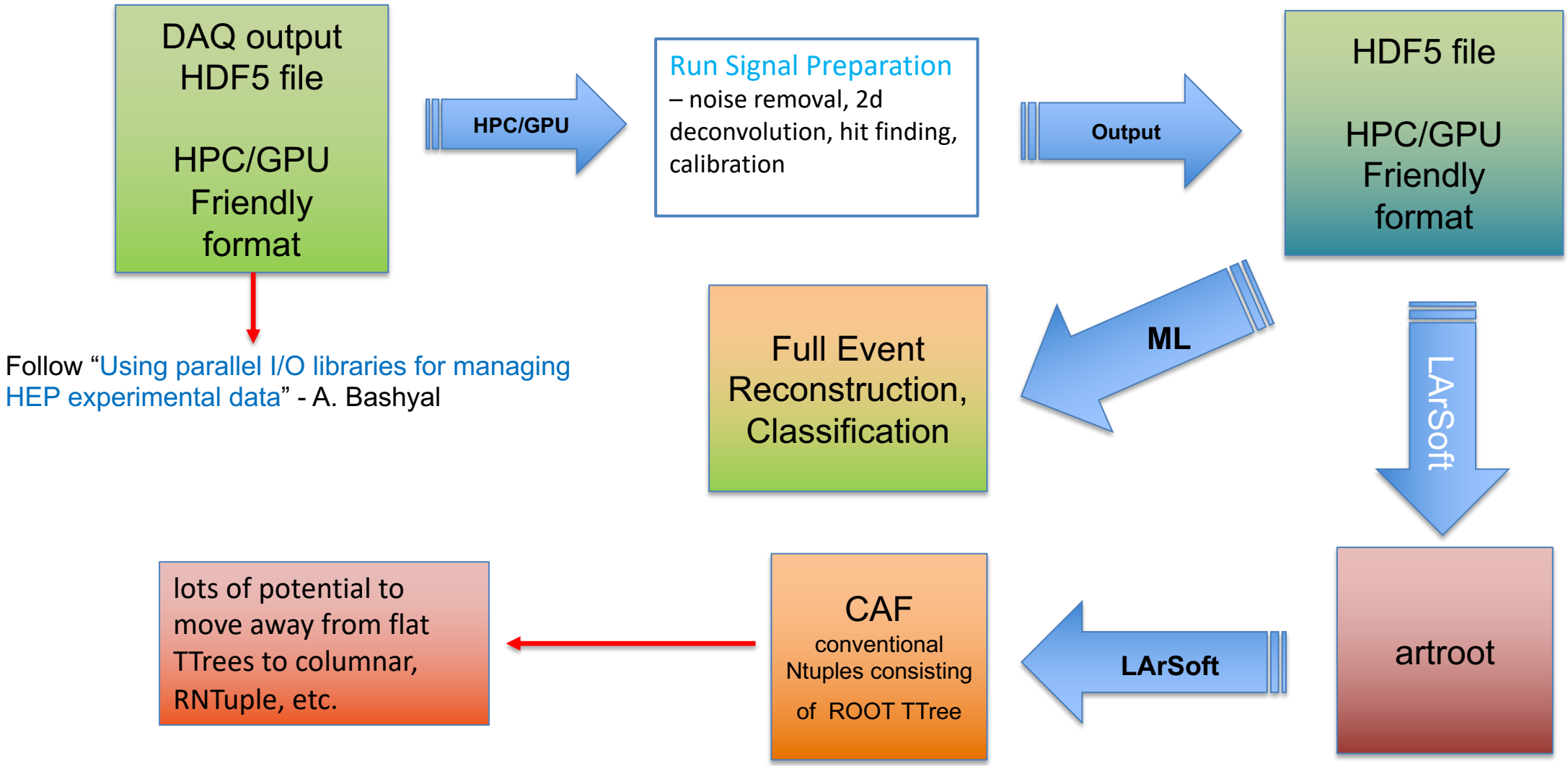
HDF5 Workflow in Current ProtoDUNE II Data Model



Includes algorithms for
Signal Preparation – noise removal, 2d deconvolution, hit finding, calibration
WireCell, Pandora etc.

Traditional HEP event model.
Slightly different I/O – CPU time compared with LHC because of larger ProtoDUNE event size.

HDF5 Workflow in Potential ProtoDUNE II Data Model



Writing Simulation using HDF5

- Existing mechanism for writing DAQ-formatted raw digits for the FD simulation.
  - It uses the DAQ HDF5 file (and data) format
  - Memory Efficiency: all raw digits from an event is stored in memory first, and then serialized out to an HDF5 file, making it memory inefficient. Work in progress.
  - Successful first test – FD-HD HDF5 writer 3.2 GB HDF5 file with 2 trigger records in it.

- Convert GEANT4 Root file into HDF5
  - Takes the output root file of neutrino event generator and GEANT
  - Stand alone Python code reads the branches, for example, trajectories, vertices, energy deposition etc. from the root tree
  - Converts into HDF5 file

Summary & Future Work

- DUNE has made good progress on using HDF5 online and offline.
- We have been able to write and read in all data produced by the DAQ(s).
- Streaming with XRootD works but could be made more robust.

Future Work

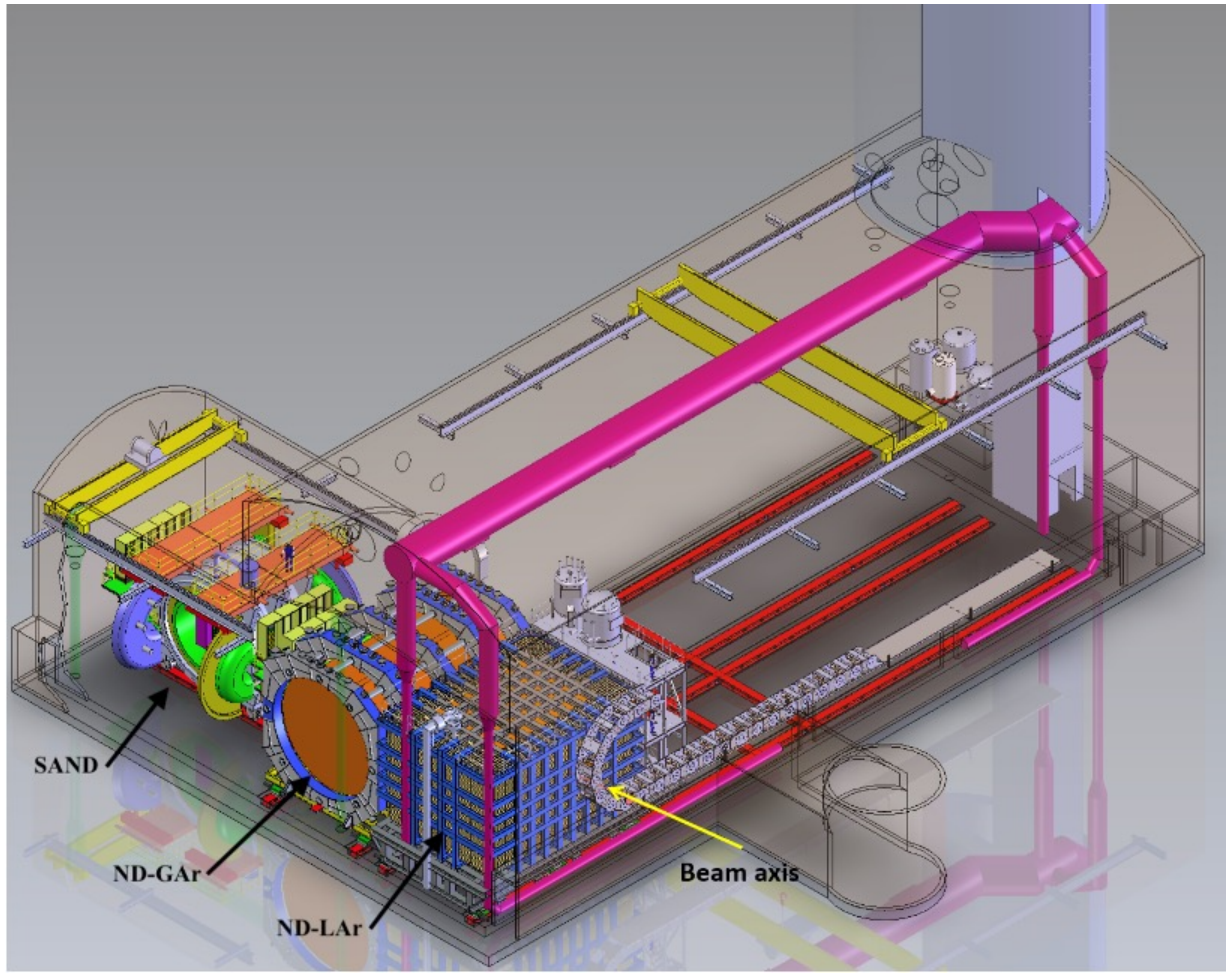
- Use HDF5 to export data from *art* jobs to external ML tools
- Stream raw digits out from WireCell one APA at a time.
- DUNE has collaborated with HEP-CCE for
  - GPU Friendly HDF5 data model
  - for parallel I/O, this allows us to take DUNE data model and evaluate it
- Envision collaborating with HEP-CCE on HDF5 streaming

Thank You

DUNE Collaboration

Back up Slides

DUNE Near Detector Design



Near Detector - 3 sub-detectors serving different purposes

ND-LArTPC: Highly segmented Liquid Argon Time Projection Chamber

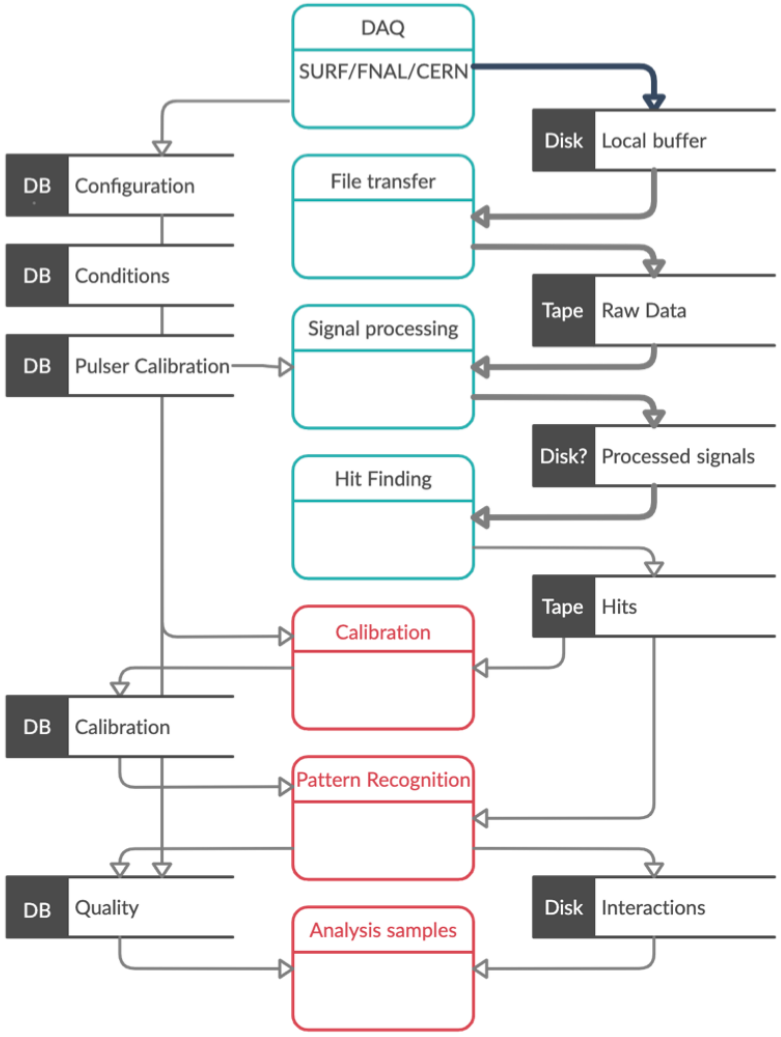
TMS: Muon Spectrometer (Phase 1)

SAND: scintillator-based tracking and active argon target for on-axis beam monitoring

DUNE-PRISM: Movement of LAr + TMS transverse to the beam

LAr TPC Data Processing

- Initial offline processing stage, labeled “TPC signal processing”
  Noise reduction, Deconvolution etc.
- Hit finding and Deconvolution
  x5 (ProtoDUNE) -100 (Far Detector) data reduction
  Takes 30 sec/APA
  Do it 1-2 times over expt. Lifetime
- Pattern recognition (Tensorflow, Pandora, WireCell)
  Takes ~30-50 sec/APA now
- Analysis sample creation and use
  multiple iterations



LAr TPC Data Volumes

In General

The first far detector module will consist of 150 **Anode Plane Assemblies (APAs)** which have 3 planes of wires with 0.5 cm spacing. Total of **2,560** wires per APA

Each wire is read out by 12/14-bit ADC's every 0.5 microsecond for 3-6 msec. Total of **6-12k** samples/wire/readout.

Around 40-80 MB/readout/APA uncompressed with overheads → **6GB/module/readout**

15-20 MB compressed/APA → **2-3 GB/module/readout**

Read it out ~5,000 times/day for cosmic rays/calibration → **3-4PB/year/module (compressed)**

(x 4 modules x stuff happens x decade) = ….

DUNE Far Detector raw data

- Far Detector front end is capable of producing incredible data rates
  - -- Reduction through trigger, zero suppression, and compression data
  - -- DAQ constrained to 30 PB/year from far detector
  - -- Beam trigger, calibrations, supernova time-extended trigger records
  - -- ~2029: Beam/cosmic ray event in1FD module--150APA~6GB at<0.1Hz

Process	Rate/module	size/instance	size/module/year
Beam event	41/day	3.8 GB	30 TB/year
Cosmic rays	4,500/day	3.8 GB	6.2 PB/year
Supernova trigger	1/month	140 TB	1.7 PB/year
Solar neutrinos	10,000/year	$\leq$3.8 GB	35 TB/year
Calibrations	2/year	750 TB	1.5 PB/year
Total			9.4 PB/year

- DAQ has chosen HDF5 for the raw data output format for ProtoDUNE II operations
  - -- Currently in discussion/design phase of the serialization of raw data for output
  - -- HDF5 datasets consist of headers + binary fragments defined by Front End Electronics Readout

Process	Rate/module	event size	size/module/year
Beam event	41/day	8 GB	63 TB/year
Cosmic rays	4,500/day	8 GB	12.5 PB/year
Supernova trigger	1/month	180 TB	2 PB/year
Solar neutrinos	10,000/year	46 TB/year	
Calibrations	2/year		1.5 PB/year
Total			16 PB/year

ND-LAr Software Development

ProtoDUNE-ND (ArgonCube 2x2)

- A modularized LArTPC demonstrator in the Fermilab NuMI Beam
- Smaller but complete version of ND-LAr module (0.7×0.7×1.4 m)

ArgonCube2x2 current simulation workflow

- GENIE → GEANT → Edep-sim (root output) → HDF5 → larnd-sim → ndlar_flow files → convert to root → Pandora/ML reco → CAF

- Still work under development for “ndlar_flow” validations, “reconstruction” and “CAF”
- Generated and simulated 20k neutrino events in LArsoft (used by ProtoDUNE and MicroBooNE)
- GENIE → GEANT → SimDump (similar to edep-sim)