Data suppression and compression SW in DUNE detector simulation and beyond

DUNE FD DAQ

David Adams
BNL

October 17, 2016

Introduction

DUNE TPC streaming data rates are large

- 10 kt: (384k chan)x(12 bit)x(2 MHz) = 1.2 TB/s = 100 PB/day
- ProtoDUNE: (16k chan)x(12 bit)x(2 MHz) = 50 GB/s = 4.0 PB/day

How to reduce these rates?

- Trigger (only read out data where something happens)
  - Can be external (beam spill, scintillator paddles, …) or on TPC data
  - Keep full detector or 1+ APAs for 1+ drift lengths
  - E.g. (DocDB 1086) ProtoDune triggers at 25Hz and keeps 5 ms/trigger
    - ➔ 1.2 GB/s = 110 TB/day (beam duty = 0.213)
- Suppression: drop ticks with uninteresting data
  - E.g. below threshold and far from interesting data
  - DocDB 1086 does not envision this for protoDUNE DAQ
- Compression: encode data with fewer bits than original format
  - DUNE uses larsoft Huffman encoding (lossless)
  - DocDB 1086 assumes this gives factor of four for protoDUNE
    - ➔ 0.3 GB/s = 27 TB/day

SW organization

Want to develop suppression and compression algorithms

- Use DUNE real and simulated data to assess performance
- Like algorithms to run in a framework where data is easily accessed

Proposal: Package algorithms as TSI tools

- TSI = Tool-Service-Interface
- Provide ctor from FCL configuration
- Inherit from an art service interface
  - Define interfaces for suppression and compression
- Add CPP macros that enable art to discover the tool at run time
- Limitation: at present, each interface can only map to one tool configuration at run time
  - Tricky to run multiple suppression algorithms in the same job
  - Typically not a problem but is sometimes limiting
  - Have asked art/larsoft to remove this restriction
    - art Service → art Tool

DetSim

Detector simulation was rewritten last winter

- Algorithmic code moved from modules to TSI tools
- Includes interfaces and implementations for suppression and compression
  - Interfaces in dunetpc/dune/DuneInterface
  - Service implementations in dunetpc/dune/DetSim/Service
  - Module in dunetpc/dune/DetSim/Module
- New module SimWireDune
  - Has switch to turn suppression on or off
    - If on, service interface AdcSuppressService is invoked
  - Compression is always invoked via AdcCompressionService
    - But implementation has the option to leave data in input format
  - Possible to add noise via ChannelNoiseService
    - Important for suppression/compression studies
  - Run-time FCL configuration of a job determines with compress and suppression algorithms are used and the configuration of each

Suppression interface

Transient data struct used in suppression interface

- Class AdcCountSelection holds the data for one channel
- Contents are shown in table

Type	Name	Meaning
vector<short>	counts	TPC data—one short for each tick
unsigned int	channel	Channel number
float	pedestal	Pedestal assumed for the suppression
vector<bool>	filter	Indicates if each tick is to be retained

Suppression interface is AdcSuppressService

- Method: int filter(AdcCountSelection&)
  - For use by caller; not implementer
- Alternative where the components of the above struct are explicit.
  - Implementer is required to provide this one
  - First method calls this

Suppression implementations

Three existing suppression implementations are listed below

- All are in dunetpc/dune/DetSim/Service
- These are the ones I know—there may be others

FixedZeroSuppressService

- No parameters
- Keeps everything
  - Placeholder to test no suppression
  - Not needed with DetSim which now has switch to turn off suppression

Legacy35tZeroSuppressService

- Params: AdcThreshold, TickRange, SuppressStickyBits, MinTickGap
- If tick *i* is above threshold, [*i*-TickRange, *i*+TickRange] are retained
  - Algorithm from the old DetSim module
- In addition, would-be gap of less than MinTickGap is retained
- SuppressStickyBits triggers special handling of 35t stuck bits

Suppression implementations (2)

Dune35tZeroSuppressService

- Many parameters
- This is a simulation of suppression algorithm that was developed for the 35t DAQ
  - I don't think this was ever used in 35t running
  - Noise levels were too high
- For details, see
  - https://cdcvs.fnal.gov/redmine/projects/35ton/wiki/Data_compression_and_zero_suppression

Compression interface

Compression interface is AdcCompressService

- Service method has arguments shown in table

Type	**Name**	**Meaning**
vector<short>&	counts	TPC data—one short for each tick
const vector<bool>&	keep	Indicates if each tick is to be retained
short	offset	Pedestal assumed for the suppression
raw::Compress_t	comp	Larsoft enum specifying the comp. alg.

Compression implementations

Two existing suppression implementations are listed below

- Both are in dunetpc/dune/DetSim/Service
- These are the ones I know—there may be others

ReplaceCompressService

- Params: Zero
- Replaces suppressed ticks with the indicated value

LarsoftHuffmanCompressService

- Params: UseBlock, UseHuffman, LogLevel
- Calls the usual larsoft compression in RawData/raw.h
- If UseBlock is set, data is put in block format
  - Suppressed ticks are dropped; size of each block is added to the data
  - The larsoft code is rewritten to use tick flags instead of a threshold
  - If flag is not set, the above service is used to replace bits with zero
- If UseHuffman is set, larsoft Huffman encoding is applied
  - https://cdcvs.fnal.gov/redmine/projects/35ton/wiki/Data_compression_and_zero_suppression

Future development

Add suppression and compression algorithms

- Presumably a major topic for this group/meeting
- Art/larsoft will find service by name as long as the containing package is set up (ups setup command)
- I propose the DAQ group or individuals have their own repository
    - Or subdirectory in dunetpc
- Redmine can be used to track these developments

Documentation

- It would be nice to have a place where these services are all listed
    - With some description
- Note the existing services have documentation in their headers
- Tests in …/Service/test/test_MyService.cxx provide example of use
    - Not yet (but should be) provided for all service implementations

Future development (2)

Multi-channel suppression

- Should add or replace suppression interface with one that takes multiple channels
- Retain ticks in neighboring channels when ticks are above threshold

Merge ADC channel struct with that used in DataPrep?

- DataPrep is the first step in processing raw data
- There data is also passed around in channel structs
  - AdcChannelData has more info than AdcCountSelection
- Switch to that struct in the suppression interface?
- And for compression?
- Drop methods with explicit list of struct members?

Merge suppression with signal finding in DataPrep

- Both determine which ticks to retain
- Enough to switch to ADCChannelData?
- AdcSuppressSignalFindingService calls AdcSuppressService

Future development (3)

Add modules to call suppression and compression

- I.e. in addition to DetSim
- Some possibilities
  - Add suppression to DataPrep
  - Dedicated module that reads raw data, applies suppression and compression, and writes new raw data
- These allow suppression and compression of real (non-MC) data

Work outside art framework (event loop)

- E.g. dedicated main program or Root macro
- Should be able to use art canvas to read in event data
- No output in larsoft format?
- But can write histograms or trees

Summary

DetSim was rewritten last year

- Algorithmic code including suppression and compression moved to TSI services
- Propose future suppression and compression algorithms also be provided as such services
  - Easy to plug in to current DetSim, future modules and non-art environment
- Expect some evolution in the service interfaces
  - As described here
  - To meet the requirements of this group

I would like to be involved

- Development of infrastructure
- Aid in migration when interfaces change
- Development of algorithms
  - In particular, ROI-based zero suppression

Extra: Getting started

Dunetpc releases are described at

- https://lbne.bnl.gov/wiki/DUNE_LAr_Software_Releases
- Gives the current release and available platforms
  - Current dunetpc/larsoft release is v06_11_00
  - Updated approximately weekly
- Instructions for installation on your local machine
  - Including mac laptop
  - Note installation only replaces packages that have changed and should be much faster the second time
- Instructions for setting up to run or build and run
- Questions or problems, please let me know