Proposal to Lengthen raw::RawDigit's fSamples

Tom Junk

LArSoft Coordination Meeting

5 November 2019

raw::RawDigit's fSamples is too small for DUNE

Currently (and since ArgoNeuT days), raw::RawDigit's private member fSamples is an unsigned short.

The maximum number storable is $2^{16} – 1 = 65535$.

The member function raw::RawDigit:Samples() also returns an unsigned short.

Oddly enough, raw::RawDigit::NADC() returns a size_t. NADC is the compressed size, while Samples() is the uncompressed size. Seems backwards!

ProtoDUNE-SP studies have been collecting data with much longer readout windows. I've seen 200,000 recently. Not even supernova readout, just coldbox tests and noise studies. Doubtless there will be more of this.

The Consequences of Overflowing fSamples

- Surprisingly it hasn't been a total showstopper.
- ADCs().size() and NADC() can be used to determine correct size of the compressed waveform.
- If Compress() == raw::kNone, then there's no ambiguity.
- But if the digits are compressed, then one must call raw::Uncompress, but one needs to reserve sufficient memory for it to work. raw::Uncompress does not do this (and rightfully so).
- If one uses raw::RawDigit::Samples() to reserve a buffer for uncompressed raw digits, it will not be big enough, and one gets a segfault.
- Workaround: don't Uncompress() non-compressed data and use the vector's size() to get the real size.
- Lose some functionality here.

The Proposal

- Upgrade fSamples from unsigned short to ULong64_t.
- Maximum possible value is $2^{64} - 1$
- long int is also an option for saving four bytes per channel, with a maximum of around 4 billion.
- Still much less memory usage than the waveforms. And it compresses very well – all channels are supposed to have the same number of samples
- Use ROOT's data product schema evolution for backwards compatibility

- I tried this a couple of years ago in my own test release.
- I ran afoul of a line in 35t code like this one
  auto nsamples = digit.Samples();
  and the type of nsamples changed from an unsigned short to a ULong64_t, which clashed with other code in the method.
- We can fix this, or even simply delete the 35t code.
- There might be similar things in other experiments' code that assumes the type of raw::RawDigit::Samples().

Rolling it out

- It's at the bottom of the apple cart, so doubtless things will break when we change the size of fSamples.
- But they should be simple to fix.

- I fixed the dune problem in a test release (was a clash of TMath::max(type1,type2)) and built
  - lardataobj
  - dunetpc
  - lareventdisplay
- I was able to create a file with new raw::RawDigits and display them with the evd, and also use the same evd to display an older file with the short sample count.
- Scanning the fSamples TBranch in a rootfile produces odd results – all zeroes with my new test release on a file containing old raw::RawDigits, random large numbers when scanning a new one. Maybe need to update the dictionaries.