dunetpc - Bug #23811

Crashes and memory errors in protoDUNE SP TPC data upacking

12/28/2019 03:43 PM - David Adams

Status:	Closed	**Start date:**	12/28/2019
Priority:	Normal	**Due date:**	
Assignee:	David Adams	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

I have been trying to run dataprep on typically 3000-event samples to estimate signal strength. I had a lot of crashes in the early data and each time I had a crash, I added the event to the tool skip list in the event filter until I was able to process the remaining events in the first 3000 events (actually 1-2999). I attach the list of filtered events. I also reject events where the reported clock (timestamp) for the TPC data was very from that for the trigger (issue #23807).

With this approach, I was able to process all but one of the samples in the attached samples list. I process each APA separately and gave up on APA 3 for run 5581 because I was getting a crash on different events from job to job.

I could declare victory here but I fear that the crashes are a symptom of memory corruption whose effect may cross event boundaries or affect downstrem processing here and in full data production.

History

#1 - 12/28/2019 03:59 PM - David Adams

- File memcheck_error_events.fcl added

I ran valgrind memcheck on the the first 2999 events in run 4517. I processed groups of 100 events, restarting and adding events to a skip list if valgrind reported an error in the unpacking. I did not run any dataprep tools for these jobs--just the unpacking. I only unpacked the data for APA 3.

The configuration holding the skipped (i.e. memcheck error) events is attached. One event (575) did not have an error in its group of 100 an error was found when the preceding group continued into that group.

Many but not all of the events identified here appear in the above list used to avoid crashes.

#2 - 12/28/2019 04:32 PM - David Adams

I looked at the first memcheck error event, event 148, by skipping the first 14 events in np04_raw_run004517_0001_dl1.root. I see the problem occurs in proto-dune-dam-lib/dam/source/cc/src/TpcTrimmedRange.hh
in the function locate at line 366:

```
   auto  wibTs = wf[wfOff].getTimestamp ();
```

The index wfOff is evaluated earlier and found to be 4166 and so the same size as the array, i.e. it is one past the valid data. There is test earlier:

```
   if (wfOff > nTotFrames)
   {
      // ------------
      // Try guessing
      // ------------
      .
      .
      .
```

that would catch this if ">" were replaced with ">=". I also wonder if we should give up rather than attempting a guess. Also should we again check the condition after the guess?

It is not clear to me how we should proceed. In #23807, Tom outlined a rather cumbersome procedure by which I could modify a copy of proto_dune_dam_lib and try it out. Different or additional packaging would make this process much easier and could provide means to get any changes into the DUNE code base. I have tried to include the author (JJ) here.

#3 - 12/29/2019 06:58 PM - David Adams

- File memcheck_error_events_v2.dat added

After updating dunetpc, I reran the above jobs looking for valgrind errors in unpacking of APA3 for run 4517. I attach the log listing the events with memcheck errors and the error type and location. All but two were the error and location discussed in the previous entry and, when I processed those two, they reverted to that error and location.

The errors reported here and above are the first found by valgrind which is configured to stop execution after the first error is encountered.

There is considerable overlap but the events with errors here differ from those above.

1. We should fix this memcheck error and then check and see if there are others.

2. The change in events causing errors and the change in the error when events are reprocessed make me wonder whether the results of the unpacking are reproducible when errors are not discovered or crashes do not occur.

#4 - 12/31/2019 02:48 PM - David Adams

- File memerrApa1v2.log added

I ran valgrind on the same events for APA 1. There is much overlap with the events that failed for APA3 but also some differences. My summary from that job is attached.

#5 - 12/31/2019 02:49 PM - David Adams

I made a mod in my local copy of proto_dune_dam_lib to avoid the above memory error and am rerunning valgrind on the 3000 run 4517 events. The change is described in #23807.

#6 - 12/31/2019 04:02 PM - David Adams

- File diff.log added

I attach the diff log (git diff) for my changes to proto_dune_dam_lib.

#7 - 01/01/2020 11:28 AM - Thomas Junk

I ran the decoder on run 4517 event 148 with the fragment file writer turned on and also in valgrind, to verify that I have the right event with an invalid read. Outputs are in
/dune/data/users/trj/forjj/run4517_event_148_rce_fragments

The output of valgrind for that one event unpacking is in the file larvalgrind.output.

I ran PdReaderTest on each of the fragment files, each of which contains one RCE fragment, in valgrind, but didn't see any invalid reads. The output is in allpdrtvalgrind.output. One fragment file, rce148_162.fragment, did turn up the message Have TpcStream data type: TpcDamaged while the others are TpcNormal.

In the unpacking run, the invalid read message got printed out after processing rce148_156.fragment.

The saving of fragment files is a bit clumsy due to the fact that each fragment must be opened multiple times in order to see if it contains the desired APA's data. Only once is the waveform data unpacked though.

#8 - 01/01/2020 11:30 AM - Thomas Junk

If memory serves, the TpcDamaged flag may be set for FEMB302's data always, but I could be misremembering.

#9 - 01/01/2020 12:20 PM - David Adams

I reran valgrind on the 2999 events in run 4517 for both APA1 and APA3 with my updated proto_dune_dam_lib. There are no reported memory errors. I also reran the signal strength jobs for all APAs on those same events and saw no crashes with no events skipped. So it looks like the patch resolves the problem reported here. I will reprocess all the runs where there previously were crashes.

#10 - 01/01/2020 12:37 PM - David Adams

- File diff2.log added

The mod I made to proto_dune_dam_lib logs a warning message (search for WARNING) each time TpcTrimmedRange::locate receives an invalid (out or range) index. It writes different messages for the case where the guess produces a valid or invalid index. I have not seen any of the former yet. The latter was not showing the original index and I have made another mod to add that information. I attach here a new diff file. I will run the signal strength jobs with this mod.

Here is a grep on the log for the first few events in run 4517 APA 1:

```
TpcTrimmedRange::Location::locate: WARNING: Skipping invalid frame index: 6144 --> 6144 >= 6144
TpcTrimmedRange::Location::locate: WARNING: Skipping invalid frame index: 37074 --> 37074 >= 1024
TpcTrimmedRange::Location::locate: WARNING: Skipping invalid frame index: 43074 --> 43074 >= 1024
TpcTrimmedRange::Location::locate: WARNING: Skipping invalid frame index: 6144 --> 6148 >= 6144
```

```
TpcTrimmedRange::Location::locate: WARNING: Skipping invalid frame index: 7041 --> 269185 >= 1024
TpcTrimmedRange::Location::locate: WARNING: Skipping invalid frame index: 13041 --> 275185 >= 1024
TpcTrimmedRange::Location::locate: WARNING: Skipping invalid frame index: 59687 --> 190759 >= 1024
TpcTrimmedRange::Location::locate: WARNING: Skipping invalid frame index: 2802 --> 2802 >= 1024
TpcTrimmedRange::Location::locate: WARNING: Skipping invalid frame index: 8802 --> 8802 >= 1024
```

The first number is received index, the second is the one after the guess and the last is the size of frame array. We see here that index is sometimes larger than the array size (not always equal) and the guess sometimes is a different value but, so far, still out of range.

#11 - 01/02/2020 08:16 AM - David Adams

- File runsumv2.txt added

I have started reprocessing all the signal strength samples (at present 138 run-triggers, most with 1-3k events). I attach my log check for the first of these with no event skips. Seven of the 46 jobs still crash, five of those in APA 2. I will investigate those.

#12 - 01/02/2020 09:16 AM - David Adams

I checked eight crashes and the cause was always the same: the noise removal tool aborts when it receives a channel with ADC length of zero. There are also messages logged by UndershootCorr.

In the first failure, the (modified) unpacking reports 260 invalid frames, dataprep reports 128 channels have clock=0 and UndershootCorr reports 48 channels with no data. The latter two are consistent because UndershooCorr is only run for collection channels (48 of 128 channels in each FEMB).

Tom, is it deliberate that the unpacking tool returns channels with length zero instead of not returning those channels? I am not arguing either way but I will have the dataprep module broadcast a warning if this is not the expected behavior. In any case, I propose to have the dataprep module (or service) skip such channels.

#13 - 01/02/2020 10:47 AM - Thomas Junk

There are fcl-steerable options, EnforceSameTickCount and EnforceFullTickCount, with the latter
depending on the value of FullTickCount. If EnforceSameTickCount is true, and some channels have different numbers of ticks in their waveforms than others, then an empty collection of raw digits and RDTimestamps is returned. If EnforceFullTickCount is true and any channel has a waveform that differs in length from the desired number of ticks, empty collections of raw digits and RDTimestamps are returned. Both of these are set to false, as FEMB302 dashed our hopes of being this insistent on data quality. The tool enforces this on each call separately (so an APA at a time in the way you normally call it), and the module enforces it on an event-by-event basis.

There are no other checks on the ticks in the decoder, so a null-length waveform will get through just fine. The median and RMS are returned as zero. I can modify this behavior if so desired.

#14 - 01/02/2020 10:54 AM - Thomas Junk

I noticed another downstream issue. There are fcl parameters for limiting the range of channels the decoder tool will unpack data for. If I unpack just channels 0 to 200, then there is a calculation of the median of a waveform in WireCell that fails with a vector out of bounds exception. If I unpack a whole APA, it's fine. Could be the median noise removal in WireCell is not that forgiving if channels are missing.

#15 - 01/02/2020 11:20 AM - David Adams

Tom: I see you created #23820 to address the last issue. Let's get some feedback from them.

I already committed a mod to the dataprep module that skips over the empty channels. It also excludes them from the clock consistency calculation and so will likely reduce the number of warning messages emitted by that module.

I have added Christoph to this report. Could he or Tingjun, let us know if I have broken reco? If need be, I can add the option to write out empty wires for missing channels.

#16 - 01/03/2020 07:50 AM - Christoph Alt

The datareco_protoDUNE-SP CI test runs normally with the current head of dunetpc develop.

#17 - 01/03/2020 09:20 AM - David Adams

Thanks Christoph. Most likely the event (events?) use in the CI test does not have the problematic fetures that cause problems here but it is good to know no problem was introduced there.

#18 - 01/03/2020 10:25 AM - David Adams

With my last set of mods, I was able to process 22 samples, most with six APAs, without any crashes before I reached APA 3 in run 5581 which aborted in the noise removal because it has channels (most likely one FEMB) with 1325 ticks instead of the expected 6000. The code in the noise removal uses a fixed length FFT and aborts when the received length is not within half to twice the fixed length. Without that abort, there is invalid memory access.

Here is a snippet from the log:

```
TpcTrimmedRange::Location::locate: WARNING: Skipping invalid frame index: 6961 --> 11261 >= 2048
TpcTrimmedRange::Location::locate: WARNING: Skipping invalid frame index: 5261 --> 5261 >= 2048
TpcTrimmedRange::Location::locate: WARNING: Skipping invalid frame index: 6961 --> 11261 >= 2048
TpcTrimmedRange::Location::locate: WARNING: Skipping invalid frame index: 5261 --> 5261 >= 2048
TpcTrimmedRange::Location::locate: WARNING: Skipping invalid frame index: 6961 --> 11261 >= 2048
DataPrepByApaModule::produce:   APA 3 digit count from tool: 2560
DataPrepByApaModule::produce:   APA 3 stats count from tool: 1
DataPrepByApaModule::produce:   APA 3 clock count from tool: 2560
DataPrepByApaModule::produce: Raw data read status: 2
DataPrepByApaModule::produce: WARNING: Channel clocks for APA 3 are not consistent.
DataPrepByApaModule::produce: WARNING:     Clock     ticks   count
DataPrepByApaModule::produce: WARNING: -99999999    -4e+06     128 Channel clock is zero.
DataPrepByApaModule::produce: WARNING:    -12518   -500.72    2304
DataPrepByApaModule::produce: WARNING:   No ADC samples:       128
DataPrepByApaModule::produce:               APA 3 # input digits: 2560
DataPrepByApaModule::produce:          APA 3 # channels selected: 432
DataPrepByApaModule::produce:           APA 3 # channels skipped: 2128 (48 empty)
DataPrepByApaModule::produce:   APA 3 # channels to be processed: 432
ToolBasedRawDigitPrepService:prepare: Processing 432 channels with 10 tools.
```

Many more warnings from TpcTrimmedRange are not shown. It appears two FEMBs have problems: one with no samples that is now dropped and one with clocks (i.e. timestamps) set to zero and presumably also has the short data.

From the event display at https://internal.dunescience.org/people/dladams/protodune/data/dqmw/run005581, we can see the new problem is in FEMB 307 (which shares a WIB with 302 that has the old problem). It is clear from that display that the data for that FEMB is of very little value for this event.

For now, I am going add this event to the skip list and reprocess the sample.

#19 - 01/03/2020 10:29 AM - David Adams

I have created a separate ticket, #23822, for the problem with missing or inconsistent clocks.

#20 - 01/03/2020 11:23 AM - David Adams

I notice that in run 5581 APA 6 (Felix readout), TPC data is often missing, i.e. the unpacking tool returns fewer than 2560 channels. Sometimes there is no data at all and other times, FEMBs are missing. I processed the first 3000 events and find data for only 1400 - 1850 events depending on FEMB/WIB. Often or always when no data is found, the unpacking broadcasts an message indicating a duplicate channel. Often or always when part of the data is present, there are no special messages or warnings.

This is handled by signal strength analysis which counts events for each channel and so should not cause problems there but I would at least like to see a warning from dataprep. I will add that.

Tom, does the analysis event filter reject all events missing data?

#21 - 01/03/2020 01:05 PM - Thomas Junk

If a channel number in offline channel notation is seen twice in an event by the decoder tool, it will return empty collections of RawDigits and RDTimestamps for that call, if this parameter is set (and it is by default):

tools.pdsp_decoder:EnforceNoDuplicateChannels: true

you can set that parameter to false and and it will allow data with duplicates. No guarantees of what anything downstream will do with such data. When decoding just one APA at a time, only duplicates within that call can be checked, so global duplicates may still exist.

The FEMB filter by default requires all FEMB's on the beam-right side (Rack, or Saleve) to contribute data. It has another mode which is not default which requires all FEMBs in ProtoDUNE-SP to contribute data. It does this by looking for the presence of appropriate RDTimestamps.

#22 - 01/03/2020 03:01 PM - David Adams

When you say "seen twice", are you also checking for an overlap in tick range after correcting for the timestamp? And, if so, do you check if the duplicated samples differ?

Or maybe one of the candidates is part of a corrupt block that should be discarded before checking?

#23 - 01/03/2020 03:06 PM - Thomas Junk

No check is made on the timestamp when a duplicate channel is flagged.

#24 - 01/03/2020 04:00 PM - David Adams

Run 5581 continues to abort with the same problem in other events: 771, 812, 817, 819.

After that I got though four samples but saw crashes in APA3 and 4 in run 6509. I see valgrind error when I unpack event 1588:

```
DataPrepByApaModule::produce: Fetching digits and clocks for APA 3.
==13923== Invalid write of size 2
==13923==    at 0x2209ED3A: adcs_decode (TpcCompressed.cc:590)
==13923==    by 0x2209ED3A: pdd::access::chan_decode(short*, unsigned long const*, int, int, int, int, int, bo
ol) [clone .constprop.5] (TpcCompressed.cc:464)
==13923==    by 0x2209F125: pdd::access::TpcCompressed::decompress(short*, int, int) (TpcCompressed.cc:286)
==13923==    by 0x2209B071: extractAdcs (TpcStreamUnpack.cc:507)
==13923==    by 0x2209B071: getMultiChannelDataBase(short*, pdd::access::TpcStream const*, int, int) (TpcStrea
mUnpack.cc:653)
==13923==    by 0x2209BC1A: TpcStreamUnpack::getMultiChannelData(short*) const (TpcStreamUnpack.cc:894)
==13923==    by 0x368E4AE6: PDSPTPCDataInterface::_process_RCE_AUX(art::Event&, artdaq::Fragment const&, std::
vector<raw::RawDigit, std::allocator<raw::RawDigit> >&, std::vector<raw::RDTimeStamp, std::allocator<raw::RDTi
meStamp> >&, std::vector<int, std::allocator<int> >&) (PDSPTPCDataInterface_tool.cc:465)
==13923==    by 0x368E5FAB: PDSPTPCDataInterface::_rceProcContNCFrags(art::Handle<std::vector<artdaq::Fragment
, std::allocator<artdaq::Fragment> > >, unsigned long&, bool, art::Event&, std::vector<raw::RawDigit, std::all
ocator<raw::RawDigit> >&, std::vector<raw::RDTimeStamp, std::allocator<raw::RDTimeStamp> >&, std::vector<int,
std::allocator<int> >&) (PDSPTPCDataInterface_tool.cc:277)
==13923==    by 0x368E695A: PDSPTPCDataInterface::_processRCE(art::Event&, std::__cxx11::basic_string<char, st
d::char_traits<char>, std::allocator<char> >, std::vector<raw::RawDigit, std::allocator<raw::RawDigit> >&, std
::vector<raw::RDTimeStamp, std::allocator<raw::RDTimeStamp> >&, std::vector<int, std::allocator<int> >&) (PDSP
TPCDataInterface_tool.cc:216)
==13923==    by 0x368E6A87: PDSPTPCDataInterface::retrieveDataAPAListWithLabels(art::Event&, std::__cxx11::bas
ic_string<char, std::char_traits<char>, std::allocator<char> >, std::vector<raw::RawDigit, std::allocator<raw:
:RawDigit> >&, std::vector<raw::RDTimeStamp, std::allocator<raw::RDTimeStamp> >&, std::vector<raw::RDStatus, s
td::allocator<raw::RDStatus> >&, std::vector<int, std::allocator<int> >&) (PDSPTPCDataInterface_tool.cc:136)
.
.
.
```

I will investigate...

#25 - 01/03/2020 04:52 PM - Thomas Junk

We still haven't heard back from JJ about how to deploy fixes. Some options, in increasing level of effort for David and Tom:

1) JJ adds your fixes in to the repo in github. He may be eager for others to maintain it however.
2) JJ adds you and me to the list of people who can push to proto-dune-dam-lib in github and we take over maintenance
3) We create a new repo like JJ's owned by us and make/deploy fixes. JJ should put a notice in proto-dune-dam-lib that it is no longer maintained and give a link to the new one.
4) We copy proto-dune-dam-lib code into dune-raw-data and take over maintenance. Same notice in proto-dune-dam-lib

Option 4 has the benefit of easing co-development of this code and code in the regular dune offline stack. It will require some rearrangement and possibly change #includes elsewhere.

#26 - 01/04/2020 09:00 AM - David Adams

Tom:

You and I did hear privately from JJ and he seemed to lean toward the latter solutions. I am inclined to prefer 3 because dune-raw-data mixes RCE and Felix and because we might someday replace JJ's code entirely.

#27 - 01/04/2020 11:32 AM - David Adams

Back to the last crash. The problem is in the decompression in proto_dune_dam_lib. TpcStreamUnpack::extractAdcs(...) loops over packets calling TpcCompressed::decompress(...) for each returning the number of decoded samples which is subtracted from the # remaining ticks. The latter returns uint_32 but this is received as int. In the problematic case, the returned value is negative. The crash occurs on the next (second) packet in the loop. The loop has test that causes it to break when the # remaining ticks is <= 0.

I will try changing the receiving type to unsigned int and see if that prevents the crash.

#28 - 01/05/2020 01:11 PM - David Adams

- File old-adcraw_tpp0z_run005581_evt000831.png added

I again started reprocessing all the signal strength samples. All was OK up to run 5581 event 831 which had an abort in APA 3 with the too few samples problem. More concerning, the attached event display show the missing data is not just shifted (or is shifted a very long way).

I again looked at the TpcTrimmedRange class. Previously the method locate(...) was modified to zero member data and return nonzero indicating error. However the ctor was not checking those return codes and went ahead and used their results to calculate the number of ticks. I modified that so that the # ticks is set to zero if either call to locate fails. I further modified extractAdcs to return immediately when the # ticks is zero instead of repeatedly calling TpcTrimmedRange and generating a cascade of error messages. Presumably the decoder tool should also be modified not to call the unpacking after the # ticks is found to be zero.

With this change, I find FEMB 302 has no data for this event, i.e. the desired behavior.

#29 - 01/05/2020 05:30 PM - David Adams

With my mods, the locate method in TpdTrimmedRange prints a message each time it is cannot find the timestamp in the data (corrupt data). With my nticks=0 fix in the preceding message, the corrupt data is ignored but we get a slew of these error messages. I believe this is because the decoding tool fetches the timestamp for each channel. Tom, could we skip those calls for the channels that have no data, i.e. nticks = 0?

#30 - 01/05/2020 06:09 PM - Thomas Junk

Okay, calls skipped in decoder tool and module if n_ticks == 0 for RCE or FELIX in the versions pushed to develop just now.

#31 - 01/05/2020 06:36 PM - David Adams

That did the trick--now there are three warnings for each APA in the event. Before APA 3 had an additional 350+ warnings. Thanks for the quick fix.

#32 - 01/05/2020 07:58 PM - David Adams

TpcTrimmedRange can fail searching for the first sample, last sample or both. I have changed the error logging so that with Tom's last mod, we get one message for each bad fragment even when both searches fail. The event above now logs two warnings instead of three. I restart the signal strength jobs with these changes.

#33 - 01/06/2020 01:08 PM - Thomas Junk

Very good. I made a fork of the github repository slaclab/proto-dune-dam-lib and named it dune/dunepdsprce. It is now in the collection of DUNE repositories

`https://github.com/DUNE`

and permissions are set by default so that members of the DUNE team in github can write to it. I added David and JJ as collaborators on the list. I see Larry Ruckman (Ruck314) updated the License.txt to increment the copyright date. I can switch over the Jenkins script to point to this version once fixes are pushed and a new tag is made.

#34 - 01/06/2020 05:13 PM - David Adams

- File adcraw_tpp0v_run005581_evt000823.png added

- File adcraw_tpp0u_run005581_evt000823.png added

Again my reprocessing in order of event number was fine until I reached APA 3 in run 5581. I again run into the problem of too short data (much less than the nominal 600 ticks) for FEMB 302 leading to a crash in the noise removal. This time it is event 823 that causes the problem.

I attach event displays for the u and v planes. Both show a FEMB with too short data but the early track in the u plane looks fine while a late track in the v-plane shows a big offset. One explanation is that data in the middle of the tick range has been dropped.

I will filter this event out and see if others fail.

#35 - 01/06/2020 06:42 PM - Thomas Junk

I had a check in the decoder module that computed the median number of ticks in all channels, and dropped the ones that did not have the median number of ticks. It's a bit gentler than dropping an entire event if the numbers of ticks are not the same. I also had some generosity for FEMB302, allowing it to have up to 10% fewer ticks. I didn't include this check in the tool, as the tool is meant to be called one APA at a time, and one could still have a mismatched number of ticks undetected. But if the failure mode is just one or two FEMBs, then this check could still be useful, and not require peeking in all of the fragments, like the module does. Would you like this?

#36 - 01/06/2020 07:21 PM - David Adams

I would prefer the decoder tool make decisions for each fragment independently. I think that is what it does now.

If we have to compare fragments, we can do that at a higher level, e.g. the decoder or dataprep modules.

#37 - 01/06/2020 07:56 PM - David Adams

- File adcraw_tpp0z_run005581_evt000815.png added

I aborted at another too-few-ticks event, 815, again in FEMB302. I attach the z-plane display. It again looks like ticks missing in the middle.

#38 - 01/07/2020 05:07 PM - Thomas Junk

I looked at the PdReaderTest example for DataFragment::isTpcNormal(), isTpcDamaged(), and isTpcEmpty(). These are per-fragment bits and not per-FEMB bits (there are two FEMB streams per RCE fragment). I unpacked a "healthy" event and the fragment from FEMB 302 comes marked TpcDamaged, as I remembered it. So if it's FEMB 302 data that's coming up short then the isTpcDamaged() check won't be useful as it is always labeled thus. It might help with other FEMBs however.

I'll go back and re-implement the median tick count check and skip channels that do not have the median, as implemented for the decoder module.

#39 - 01/07/2020 07:25 PM - Thomas Junk

I pushed to develop a modification to the PDSPDataInterface_tool that checks the numbers of ticks on each channel decoded for the APA requested, and if it differs from the median, remove the raw digits and corresponding RDTimestamps. This skip is not done for FEMB302 unless n_ticks < 0.9*median ticks. Maybe that can be tightened up.

I tried it out on a couple of clean events in run 5177 and it decoded fine. I tried it out on run 5581 event 815 however and FEMB 302's data was absent anyway, perhaps it failed another test. I ran the decoder module on it and it flagged FEMB 302 as having a crazy large n_ticks.

See if it helps.

#40 - 01/08/2020 09:21 AM - David Adams

I modified the dataprep module to loop over channels and find the most common tick count and then added an option to to keep only channels within a fcl-specified fractional range of that count. I set the range to 0.5% and started reprocessing the signal strength samples again. So far, I have processed the first 32 samples including the troublesome run 5581 without any crash or abort.

Tom, I would like to have a decoder tool (or configuration) that unpacks as much as possible with the option to drop or flag fragments that appear to be bad. That decision should be based only on data from the same fragment. I assume your last mod is based on data from multiple fragments and so I would like the option to disable this check.

The problem you see in event 815 should be fixed by the patches I have for proto_dune_dam_lib. I will work on moving those into the new repository.

#41 - 01/08/2020 09:47 AM - David Adams

I checked the new dunepdsprce out and looked at the README but it is not clear to me how I should build the package. Will this build in mrb or should I do the same as I did with proto_dune_dam_lib? Thanks.

#42 - 01/08/2020 09:54 AM - Thomas Junk

Yes, the dune/dunepdsprce repo in GitHub is just a fork of slaclab/proto-dune-dam-lib. One would replace the line in the Jenkins build script which clones slaclab/proto-dune-dam-lib and replace that with the dune/dunepdsprce. You probably already substituted that line to point to your own copy of the source. It will take some effort to redo the makefiles to use mrb. So I'm following option 3 and not 4 at the moment. In the future we can do some work to put it in mrb but we may lose some features (avx2). And the makefiles used to have some switches in them to get things to work on macOS but I think some of that's been fixed in the code so it might be more easily ported.

#43 - 01/08/2020 10:05 AM - Thomas Junk

An issue with unpacks as much as possible with the option to drop or flag fragments that appear to be bad is that attempting to unpack a fragment that appears to be bad often creates a segmentation fault. The small-fragment check staved off a lot of crashes, as does the isOkay test. I can make the test on consistency of tick counts optional.

#44 - 01/08/2020 10:42 AM - David Adams

The changes I made in (now) dunpdsprce are to avoid crashes when you first ask for nticks for a fragment. It should return 0 instead of crashing and or having memory errors.

I am trying to create a script to build and install dunepdsprce in my local development area. Do you know where we set and retrieve the version in JJ's code?

#45 - 01/08/2020 12:10 PM - David Adams

To answer my own question, I found a line in Makefile where the version is set. I change that to v1.1.1.

#46 - 01/08/2020 01:02 PM - David Adams

I pushed my mods to dunepdsprce. In addition to the fixes described above, I added a directory dunebuild with a modified copy of your Jenkins build script and a wrapper buildMrb that uses it to build and install dunepdsprce in a mrb development area using the current mrb configuration.

I tested on run 5581, event 815 and it runs successfully losing two FEMBs in APA 3 with two error messages. One is the new code from you. Please tell me when I can disable that check.

Please try out the new code and, if you happy with it, make a Jenkins build, bump the version in dune_raw_data and build that package.

Thanks. --da

#47 - 01/08/2020 02:06 PM - Thomas Junk

Just now I pushed an update to PDSPDataInterface_tool.cc that makes the new requirement on all channels having the same tick count as the median except FEMB302 optional, and set the default so the new requirement is disabled, i.e. it will pass through readouts with varying numbers of ticks.

Tingjun had pointed out the CI tests were failing due to a lack of e19 dune_raw_data and lbne_raw_data products and I released new ones just now (missed this). I can make another dune_raw_data after tagging and building dunepdsprce.

#48 - 01/08/2020 02:20 PM - Thomas Junk

As for the version number in JJ's makefile, that's for naming the .so's. The build script uses the version to check a tag out of github. It's probably good to make them the same just to avoid confusion. Though we do not use the version-number-labeled .so's since UPS has another solution -- putting all the .so's in versioned directories, rather than putting them all in the same directory which is what JJ did, with symlinks in the unversioned so to the latest. Thanks for upgrading to v1.1.1 but we need to tag and build. I suppose I should check it out and build it first just to see how it works.

#49 - 01/08/2020 05:02 PM - Thomas Junk

David committed his changes to the new repository at GitHub: dune/dunepdsprce.

dunepdsprce v1_1_1 is now available on supported platforms with e17, e19, c2 and c7 compilers.
dune_raw_data v1_17_43 is now available with e19 and c7 compilers, and depends on the new dunepdsprce.

I tested it with a clean event in run 5177 with e17:debug and also
run 5581, event 815 and got expected behavior. I did see a strange segfault in the c2:prof build but could have built it wrong interactively.

#50 - 02/22/2021 08:12 AM - David Adams

- Status changed from New to Closed

I suppose it is OK to close this now.

#51 - 02/22/2021 10:19 AM - Thomas Junk

Yes, fixed long ago. Thanks.

Files

crash_events.fcl	3.02 KB	12/28/2019	David Adams
runs.log	5.93 KB	12/28/2019	David Adams
memcheck_error_events.fcl	635 Bytes	12/28/2019	David Adams
memcheck_error_events_v2.dat	4.16 KB	12/30/2019	David Adams
memerrApa1v2.log	3.67 KB	12/31/2019	David Adams
diff.log	2.41 KB	12/31/2019	David Adams
diff2.log	2.46 KB	01/01/2020	David Adams
runsumv2.txt	4.13 KB	01/02/2020	David Adams
old-adcraw_tpp0z_run005581_evt000831.png	906 KB	01/05/2020	David Adams
adcraw_tpp0u_run005581_evt000823.png	1.68 MB	01/06/2020	David Adams
adcraw_tpp0v_run005581_evt000823.png	1.68 MB	01/06/2020	David Adams
adcraw_tpp0z_run005581_evt000815.png	925 KB	01/07/2020	David Adams