Redefining Events in *art*

Tom Junk

art Users' Meeting

17 June 2016

Many Thanks

- Many thanks to the team who put it all together

I'm 35-ton centric, but have a bit on LArIAT's slicer

- *artist* help: initial version. Marc Paterno, Chris Green, Kyle Knoepfel
- DUNE team: Karl Warburton. Advice, channel maps etc: Tom Junk, Michelle Stancari, Tingjun Yang, Mike Wallbank
- LBNE discussions about reformatting data: Jonathan Davies, Jeff Hartnell, Alec Habig, Brett Viren, Maxim Potekhin

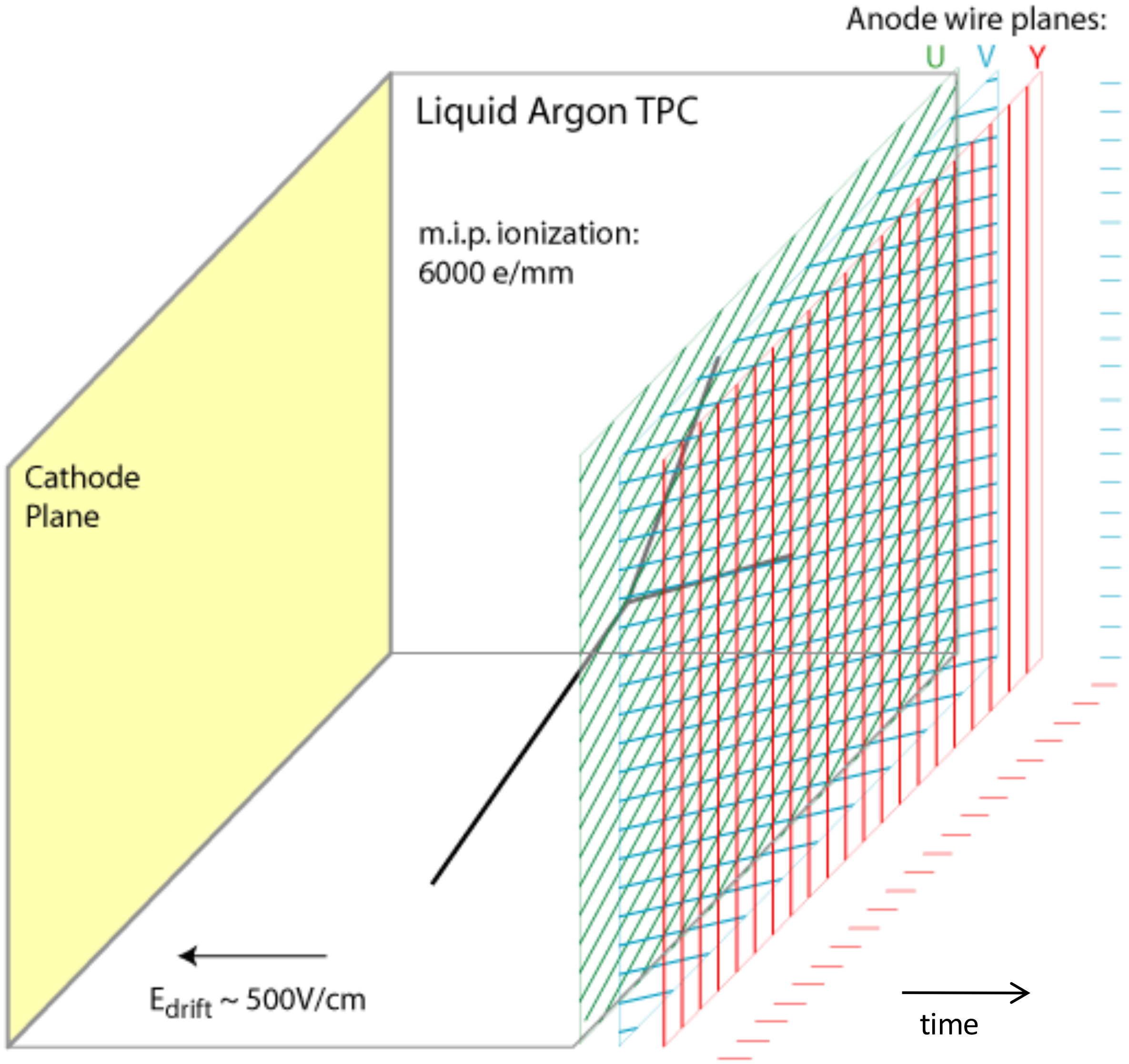
- LArIAT's event slicer: Johnny Ho, Jason St. John, Kyle Knoepfel, among others.

Different Definitions of "Event"

- "Event" is a highly overloaded word
- Not really a source of confusion but it does require extra words to explain what we are talking about

- Physics: A point in (x,y,z,t) space, whether or not anything happened there.
- Physics: An interaction such as a collision or a particle traversing a detector
- Experimental HEP: A triggered readout of a detector. Defined by the trigger and DAQ. May contain data from multiple interactions (pileup "vertices" just to avoid re-use of "event")
- From the *art* workbook: art::Event is the smallest unit of information that *art* can process.
- Statistics: A set of experimental outcomes to which a probability is assigned. Your entire experimental data set is an "event".

Single-Phase LArTPC Charge Deposition



3D charged particle tracks and showers projected onto 2D time vs. wire views

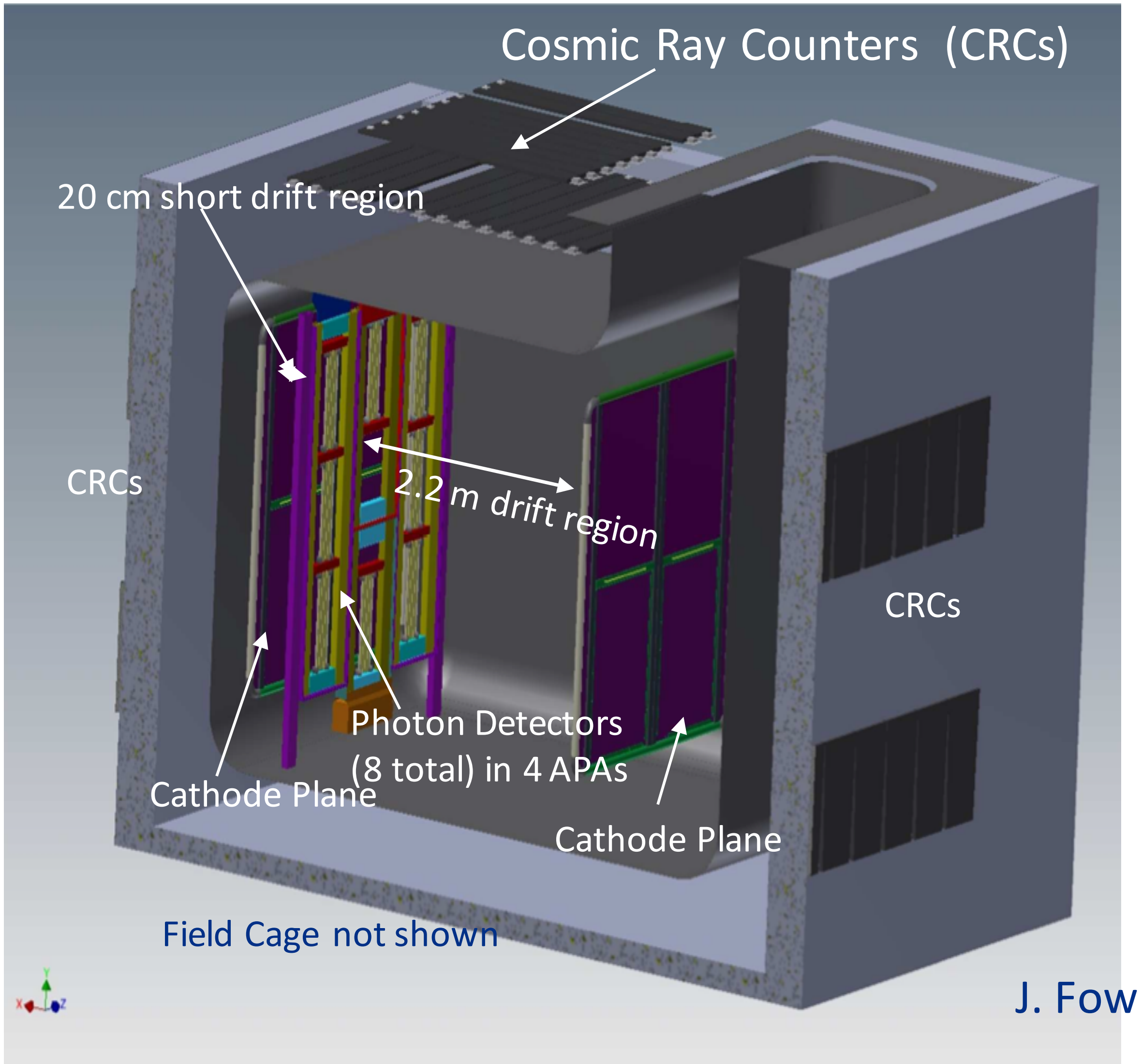
An Unusual LArTPC – DUNE 35-ton Prototype

Two-sided anode planes U, V, Collection

Long and short drift volumes

No beam – just cosmic rays. Approx. 1 per 3 ms long drift time.

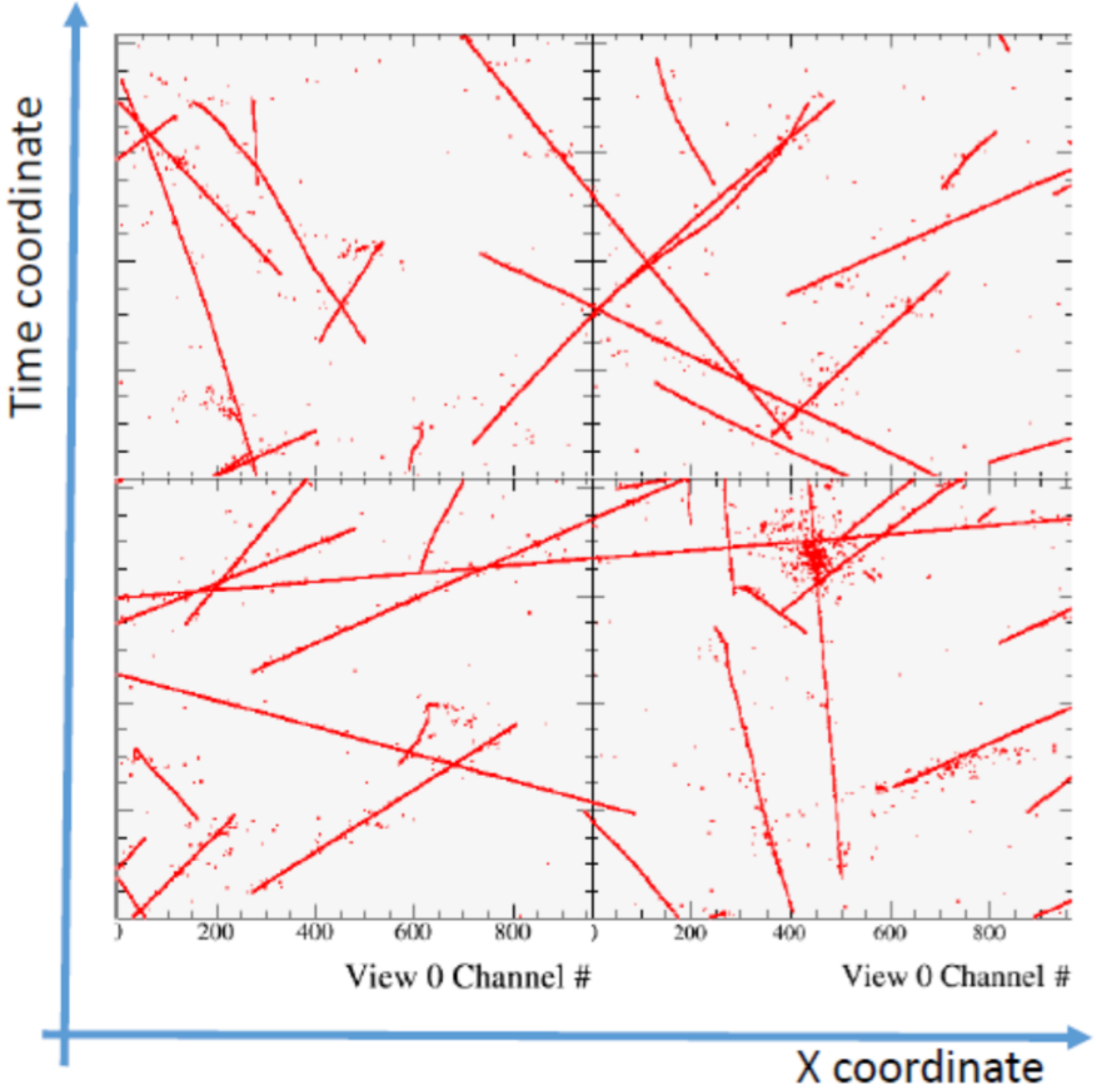
External scintillator counters for triggering



J. Fowler

Annotated picture from M. Stancari

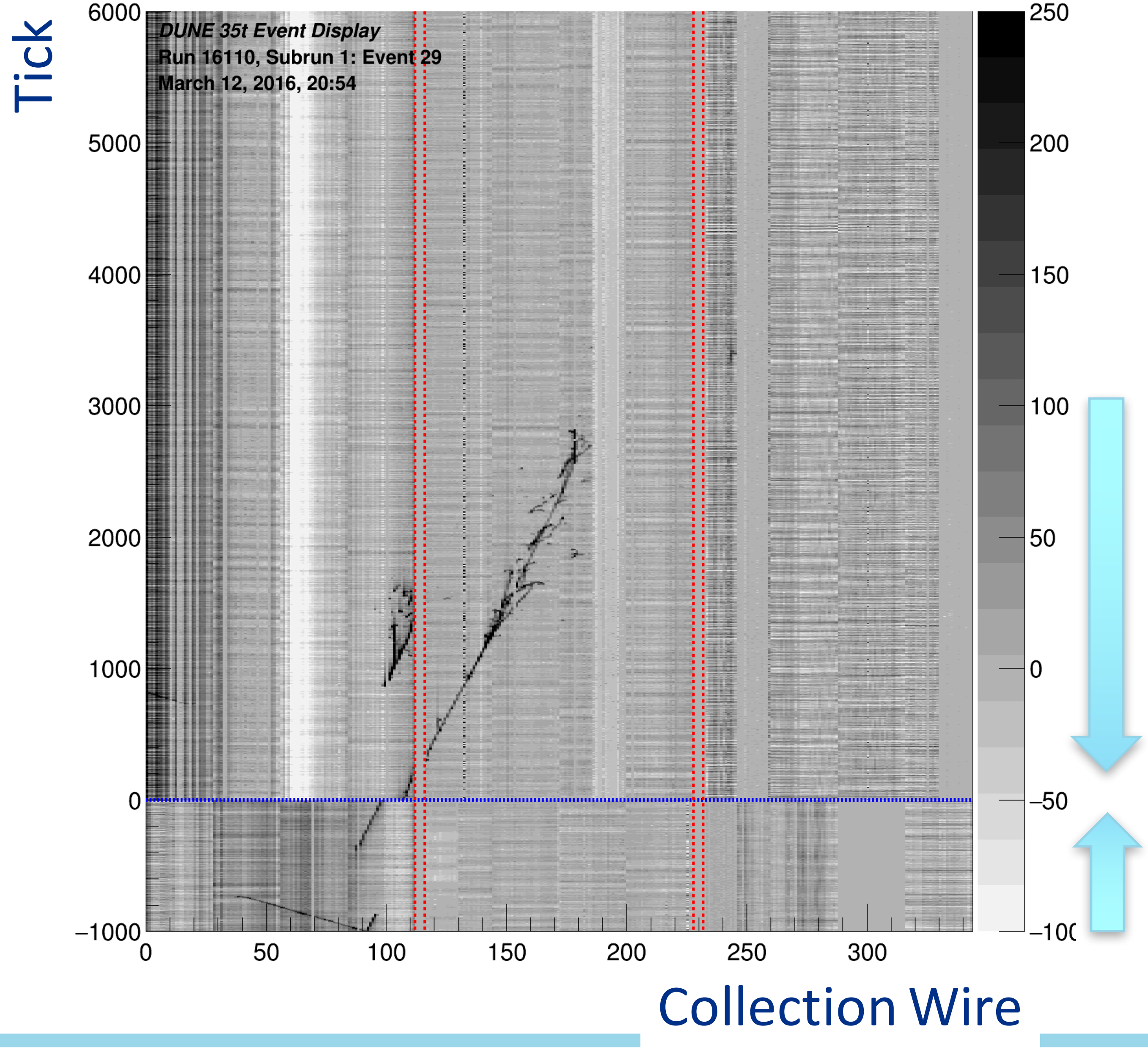
A Larger Detector – WA105



Simulated event from Elisabetta Pennacchio, Dec. 2015

Data rolls by on a conveyor belt

An Online Event Display from 35t



Triggering and Data Acquisition

- We would like to collect data from interesting interactions and store them for later use.

- Not every interaction is interesting.

- Sometimes they pile up too fast
- Deadtime when we cannot read out fast enough.
- External counters or photon detectors or analysis of the TPC data can provide triggers

Original Motivations for DUNE Splitter

- Early plans – run untriggered, continuous readout. Need large data reduction from zero suppression.
- Pre-time service code – assumed that interactions happened at t=0.
- Convert ticks to *x* needs a t_0
- Need to know t_0 in order to apply fiducial cuts
  - Stopping muon analysis: throughgoing cosmics can look like stopping muons if t_0 is wrong.
- There is not a complete symmetry in $(x-v_{drift}t)$.
  - electron lifetime
  - charge diffusion
- Idea was to center the interaction of interest at the same time for all events in an analysis

artdaq and Events

- Earlier example: LongBo input Source (M. Stancari) read in a custom-formatted input file and handed data to *art* to package in events. Lots of flexibility in reading a flat file.

- 35-ton used artdaq – events are defined by the DAQ.
  - More information tied to events in an *art*-formatted file than just a flat file.
    - event counts
    - event indexes
    - TBranch indexes and members.
  - More difficult to rearrange information than if the data were recorded in a flatter format

Steps Needed for the Splitter

- A replacement for the RootInput Source
  - Opens and closes files
  - Reads data from appropriate branches (Data and MC are different!)
  - Reformats data from DAQ to offline format (Data only)
    - artdaq::Fragments to raw::RawDigit
    - artdaq::Fragments to raw::OpDetWaveform
    - artdaq::Fragments to raw::ExternalTrigger
  - Check data integrity – all pieces of the detector reporting?
  - Runs a trigger algorithm on the reformatted data
  - Compares timestamps and shifts data into appropriate containers and fills in the EventPrincipal
  - Different subdetector components have different formats for their timestamps
  - Number of ticks for output event is configurable
  - Assigns a new event number (worry if > 2^32)

dunetpc/dune/daqinput35t/SplitterInput_source.cc

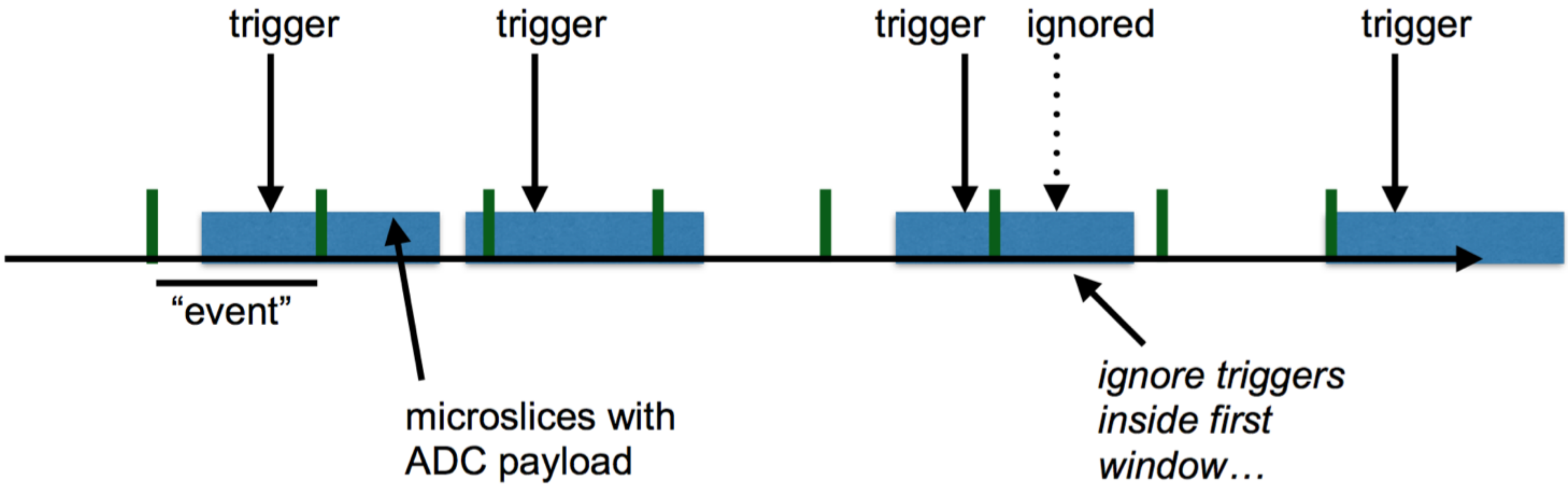
Stitching Events Too!

- Chris Green observed that if we wanted to be able to divide events into smaller ones, there would be nothing preventing us from stitching events together.

- Previous discussions in LBNE S&C focused on the need for this.
  - Interactions of interest could cross a record boundary
  - Even if not, we would like time before and after the interaction of interest in order to reconstruct and filter cosmic overlays.
  - An idea that didn't get much traction – duplicate data near record boundaries so each record has extra non-fiducial data on the ends.

35-ton DAQ Format Forces Stitching, Even for Single Events

Matt Graham, SLAC


trigger
trigger
trigger
ignored
trigger
"event"
microslices with ADC payload
ignore triggers inside first window…

Someone may ask…why do we do it this way??? This is crazy! You want me to stitch events together?

> ***answer:*** we don't want to run triggered…we want to take most of the data waveform extracted. Yes, you will have to stitch events together (or duplicate data).

Issues Raised by Stitching

- When combining data from different DAQ events to make an offline event:
  - Timestamps must match what is expected so that data can be laid end to end. No missing ticks.
  - The same portions of the detector must contribute data to all input events, for all ticks.
    - In 35t, some RCE's were flaky
    - For much of the run, one of the APA's was not read out at all.
    - LArIAT – different subdetectors may contribute for different amounts of time.
  - No attempt to stitch events in different files.
  - Input source needs to be able to open and close files, and flush its buffers when moving to a new file.
- Events are sorted using art's index branch – tree indexes can give events out of order.

The Channel Map

- Mapping online channels to offline channels has to be done somewhere
- Initial attempts had wrong channel assignments
  - Checked with data – tracks zig-zagging around in ways easily explained by wrong channel ID's
- Geometry services and helpers: ChannelMapAPAAlg for DUNE encodes the offline channel sorter. Could have encoded it there.
- Wanted an upstream map fixer so that all plots, even of just charge vs. channel, could be made without looking it up at each step.

Split Data for Analysis Use

- Reformatting raw data is slow (possibly some low-hanging fruit here)

- Need to reformat the data on readin so the same trigger algorithms and split/stitch algorithms can be applied in data and MC.

- We didn't have time or inclination to devise many trigger algorithms.
  - Settled on external counters.
  - Code is flexible to allow other algorithms (photon detectors, things that depend on TPC data, or just chop events into uniform pieces)
  - Re-slicing with a different trigger means run and event numbers aren't constant for the same data.

- Pre-split data samples created and registered in SAM as a separate data tier: "sliced".

- Need to re-split data when channel map is updated. Or other problems found.

- Helps reproducibility to store sliced data.

Pushing DAQ work offline

- Talk of skipping the artdaq event builder step in order to increase readout speed. BoardReaders write disk directly.
- Multiple streams easily defined if data are not funneled through an event builder.
- Offline input source can build events.
- Ugh – difficult to do DQM or tell online if data are corrupt.

- Similarly – sequential events in data may not be in the same file, due to parallel writes.
- This is okay, as long as we don't have to stitch offline. Otherwise, we may need an event sorting step

Splitting MC

- Data are already in the right format.
- MC events are independent – same initial timestamp so you cannot stitch them.
- 35-ton MC events simulate long DAQ events
- Need to split MC truth information too.
  - MCParticles
  - MCTruth
  - sim::SimChannel

LArIAT's Slicer

- LArIAT front-end units send data packets on each trigger
- Except for the wire chamber (momentum measurement of beam particles) – saves up all triggered packets and sends it all at the end of a spill
- artdaq wraps all data from a spill (plus a bit of pre and post-spill cosmic data) into four LariatFragments, each of which gets its own art::Event
- All four of these input fragments must be read in before offline event assembly can proceed, similar to 35t's offline event building.

Many thanks to Jason St. John, Johnny Ho, Kyle Knoepfel, and others.

Summary

- Event redefinition is something experiments need to do.
- There is no completely general procedure of what it means to redefine events – intimiately tied to the online and offline data products
- art's flexibility of using alternate input sources is very welcome by the experiments.
- Examples are very useful. We are grateful for the help of the artists in getting started, and customization was straightforward.