

Q Frames: Coincident Events in IceTray

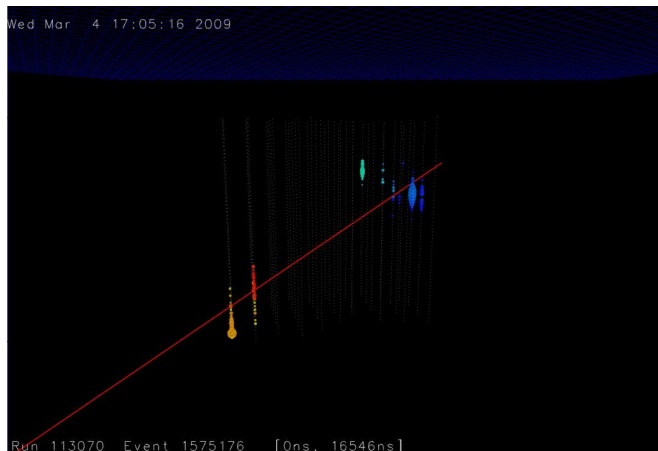
Nathan Whitehorn, Naoko Kurahashi, Jakob van Santen

University of Wisconsin - Madison

September 24, 2011



Coincident Muons



Problem

A single DAQ readout can contain many physics events not easily separable just by time. How to handle this in software? (Similar problem: long events, like ANTARES DAQ or monopole trigger)

Options for Handling Coincident Events

- ▶ Shove many I3Particles into a frame
 - ▶ Identifying reconstruction domains complicated
 - ▶ N hard-coded processing modules
 - ▶ Calculating cut variables difficult

Options for Handling Coincident Events

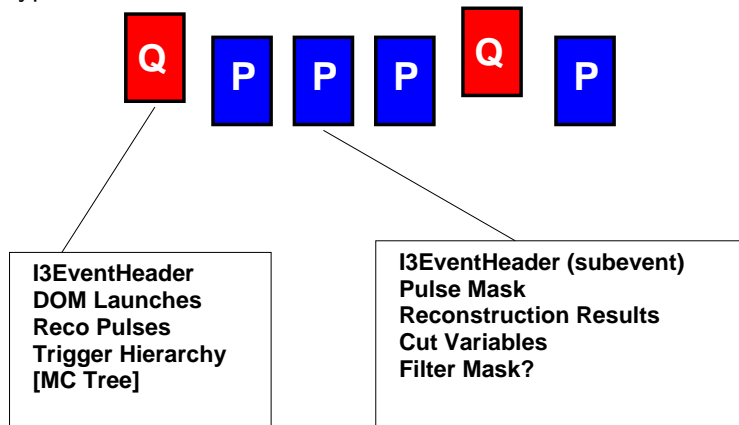
- ▶ ~~Shove many 13Particles into a frame~~
 - ▶ Identifying reconstruction domains complicated
 - ▶ N hard-coded processing modules
 - ▶ Calculating cut variables difficult
- ▶ Duplicate frames with pulse subset
 - ▶ Massive waste of disk space
 - ▶ Event IDs ill-defined

Options for Handling Coincident Events

- ▶ ~~Shove many 13Particles into a frame~~
 - ▶ Identifying reconstruction domains complicated
 - ▶ N hard-coded processing modules
 - ▶ Calculating cut variables difficult
- ▶ ~~Duplicate frames with pulse subset~~
 - ▶ Massive waste of disk space
 - ▶ Event IDs ill-defined
- ▶ Reimagine physics frames as a different object from DAQ readouts
 - ▶ Can [ab]use IceTray mix-in mechanism to inherit PMT readouts, etc.
 - ▶ Mostly software and user transparent
 - ▶ Breaks a small amount of software

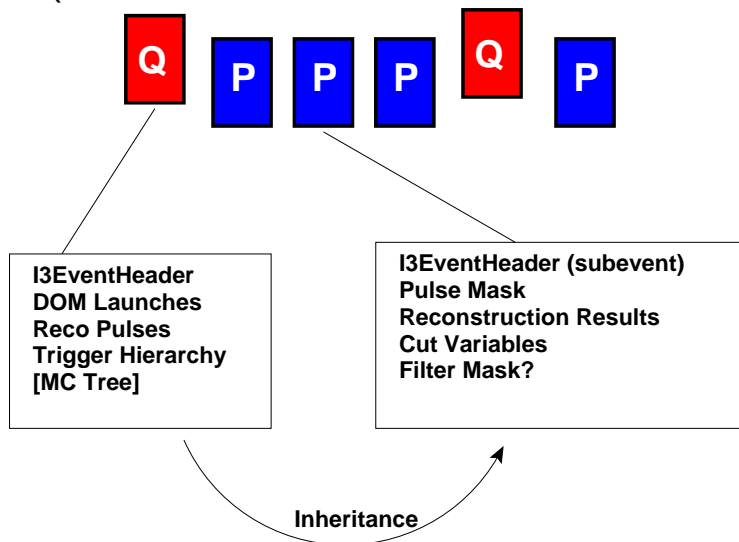
Anatomy of an Event

Event information currently stored in P frames is split into two types: Q and P



Software Transparency

Using the Ictray frame mix-ins (already used for G/C/D data), events processing Physics (P) frames have access to the contents of Q frames



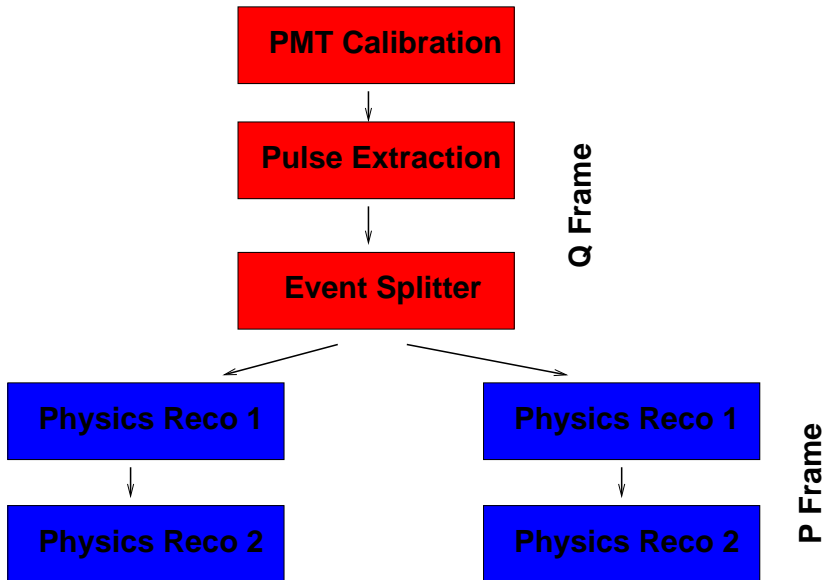
Processing Data in Q+P

- ▶ I3EventHeader now has a subevent stream (string) and ID (integer)
 - ▶ Stream: instance name of the splitter
 - ▶ ID: subevent sequence number from each splitter from parent Q frame
- ▶ Splitter (e.g. I3NullSplitter) required after low-level processing (payload parsing, feature extraction)

Important Note

Any analysis script working at a higher level than calibration and waveform processing requires no modifications!

IceCube Processing Chain



Streams

Each subevent is assigned a subevent stream that is the instance name of the splitter module

- ▶ Multiple splitters (different algorithms, settings) can exist simultaneously
- ▶ Within a given stream, events guaranteed not to share photons to prevent double counting
- ▶ For each event, monotonically increasing subevent ID per stream
- ▶ Tuple (Run, Event, Stream, Subevent) globally unique

Writing a Splitter

Splitters inherit from mix-in class I3Splitter, which provides a utility method GetNextSubEvent():

```
void MyFancySplitter::DAQ(I3FramePtr frame) {  
    I3FramePtr subfr;  
    PushFrame(frame);  
    for (int i = 0; i < n; i++) {  
        subfr = GetNextSubEvent(frame);  
        FillSubEvent(subfr);  
        PushFrame(subfr);  
    }  
}
```

Event Views (Saving disk space)

Subevents, especially with multiple streams, and hit cleanings can waste lots of disk space, so there are now pulse masks

- ▶ Store a reference to a subset of the master PMT pulses
- ▶ Typically $< 1/10$ the size of a full new pulse series
- ▶ Preferred way to describe a subevent
- ▶ Transparently cast to a pulse series by IceTray for use by reconstructions → requires no software changes to use

Case Study 1: InIce/IceTop Coincidence

IceCube is made of two detectors, with independent reconstructions and event splittings. What if we want to reunify events (vetoing, cosmic ray studies, etc.)?

Answer: I3PacketModule

- ▶ Can see master readout (Q) and all subevents (P)
- ▶ Make this into a "splitter" that combines readouts from existing splitters → new stream

Case Study 2: Slow Particle Trigger

IceCube DAQ readouts are extendable. The slow monopole search has a trigger that reads out second-long events. How not to make this deadtime?

- ▶ "Retrigger" events offline based on stored online triggers into new subevent stream
- ▶ Keeps original full event for monopole search, but provides regular triggered events for other analyses

Status

- ▶ Used in IceCube starting with IC79
- ▶ Allows better triggering on coincident/long events
- ▶ Saves disk space
- ▶ In core IceTray modules
↔ available to SeaTray

