Event Builder

Matt Toups

What does our data look like?

What does our data look like?

- It's a binary file with
  - Data packets (p),
  - Unix time stamps (t),
  - Diagnostics (c).
- Cf. Camillo Mariani’s June08 DC Collaboration Meeting presentation

Data Packets

- Trigger

    11 100000 1111111111111111

    11 100000 0,mod#[6:0],wdcnt[7:0]

    11 100000 time[31:16]

    11 100000 time[15:0]

    11 100000 disc[15:0]

    11 100000 disc[31:16]

    11 100000 disc[47:32]

    11 100000 disc[63:48]

    11 100000 parity[15:0]

Data Packets

- ADC packets

    11 100000 1111111111111111

    11 100000 1,mod#[6:0],wdcnt[7:0]

    11 100000 time[31:16]

    11 100000 time[15:0]

    11 100000 0000,adc[11:0]

    11 100000 0000000000,ch#[5:0]

    "   "   "

    "   "   "

    11 100000 parity[15:0]

Unix time stamps

- Inserted directly into the data stream (roughly every second)

  11 001000 timestamp_hi[31:16]

  11 001001 timestamp_lo[15:0]

Diagnostics

- Total Words counter
- Inserted directly into the data stream

    11 000110 data_words_received_hi[31:16]

    11 000110 data_words_received_lo[15:0]

    11 000110 data_words_lost_hi[31:16]

    11 000110 data_words_lost_lo[15:0]

What is an event?

What is an event?

- An event will be defined as
  - At least one x hit and one y hit
  - Above an n pe threshold
  - In geometrically overlapping strips
  - Within m clock cycles (after taking into account time offsets)

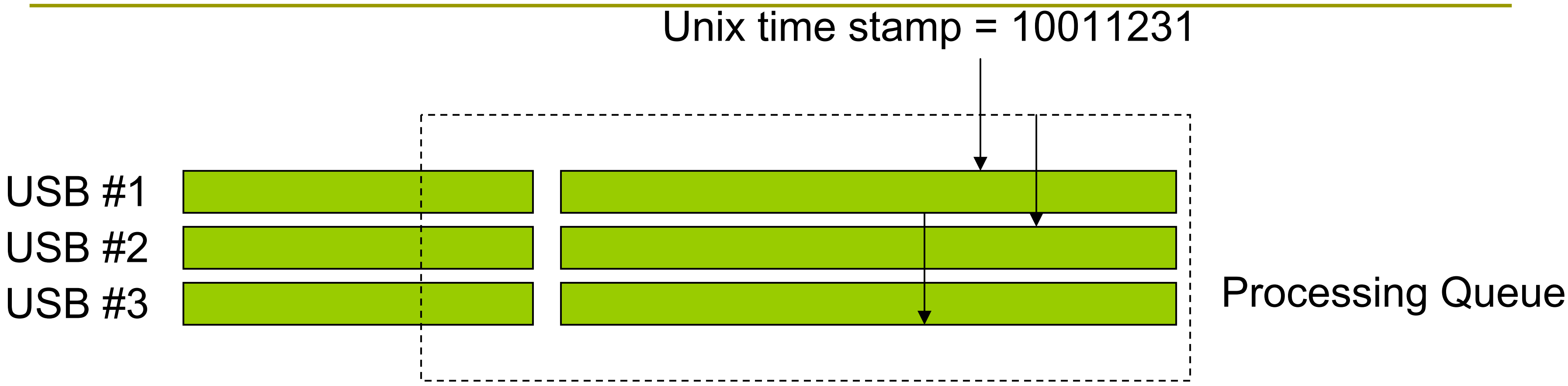
Information needed to construct this event:

Information needed to construct this event:

- The board number for each data packet
- The unix time stamp for each data packet
- ADC data / hit info for each data packet
- Clock counter for each data packet
- Calibration data corresponding to timing offsets for each PMT board
- Topological data corresponding to whether 2 modules overlap
- Time of last sync pulse
- (And if we use > 64 modules in the near detector: USB board number for each packet).

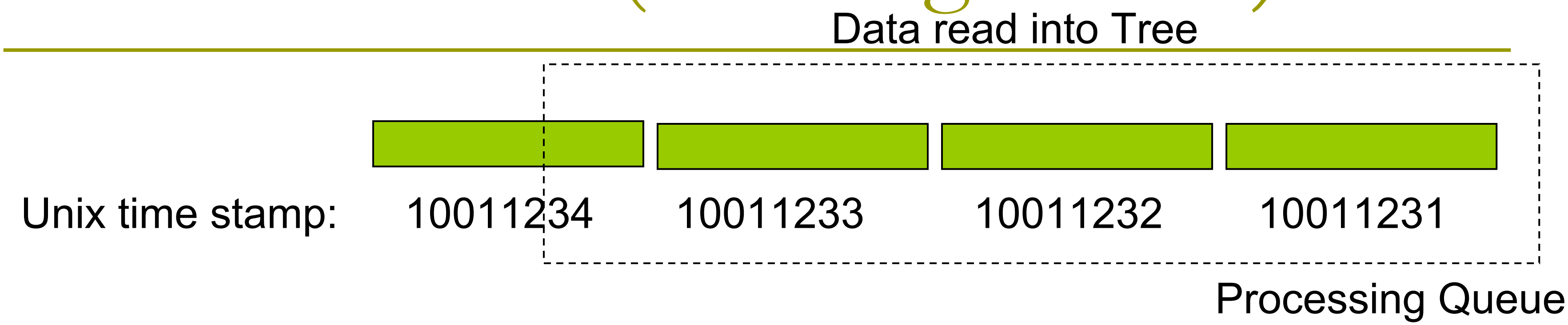
Event Builder (If sync pulse accurately known)

Event Builder (reading raw data)



- Stored in memory:
  - Time of last sync pulse (TOLSP)
  - Last unix time stamp processed
- Add all data packets to Tree with a corrected global time derived from TOLSP and timing offsets until next unix time stamp is found.
- If more than 64 PMT's are used, store 10*(PMT board number) + USB board number instead of just the PMT board number to identify each PMT uniquely

Event Builder (forming events)



- Stored in memory:
  - Next unix time stamp to be processed (NTSP)
  - Next unix time stamp added to be added to the queue (NTSA)
- Sort data from last few unix time stamps on global time
- Using a sliding *m* * 16 ns global time window, check to see if there are any data packets from overlapping PMT modules and form an event from these packets, if the event does not already exist.
- Once all data with unix time stamp == NTSP has been processed, flush this data and add packets with unix time stamp == NTSA to the queue.

Event Builder – How big is *m*?

- Contributions to *m*:
  - Uncertainty in TOLSP
  - Uncertainty in timing offsets
  - Clock Counter uncertainty (+- 1)

Event Builder – How big can *m* be?

- Using Leslie's note on rates & dead time:
  - Noise rate per pixel: 2.7 Hz → Coincidence rate per plane: 4 * *m* * (16 ns) * (64 * 2.7 Hz)^2 = *m* * 1.9 x 10^{-3} Hz
  - Worst case: 4 * *m* * (16 ns) * (3000 Hz)^2 = *m* * 0.576 Hz
  - Muon rate per module: 32 * 0.1 Hz (far) → Muon rate per 3.2 m x 3.2 m plane: 2 x 3.2 Hz = 6.4 Hz
- In the best case, *m* can be very large 20-30 without affecting the experiment's dead time
- In the worst case, we would increase the experiment's dead time by a factor of ~ *m*/10

Event Builder (if sync pulse not accurately known)

Event Builder – Considerations

- How would one correlate data packets that due to timing offsets, span two sides of a sync pulse?
  - Data packet #1: unix time stamp = 1000121
    clock counter = 625010101
  - Data packet #2: unix time stamp = 1000121
    clock counter = 000000002
  - Known timing offset: 23 clock counts
- Do these packets fall within an *m* clock count window (i.e., *m* * 16 ns)?

Event Builder – Considerations

- Answer: Undecidable.
  - The relative time between these 2 packets is given by 2 + (max clock count) – 625010101.
    - How well we know max clock count is related to how precisely we know:
      - The duration/frequency of the sync pulse, or
      - The global time of the sync pulse
  - If there is an uncertainty in this number of, say, q 16 ns time intervals, then one must look in windows of $m + q$ clock counts in order to be certain to find all events that fall within m * 16 ns.

Event Builder – Considerations

- Conclusion:
  - We must know the duration or time of arrival of the sync pulse to sub-µs accuracy

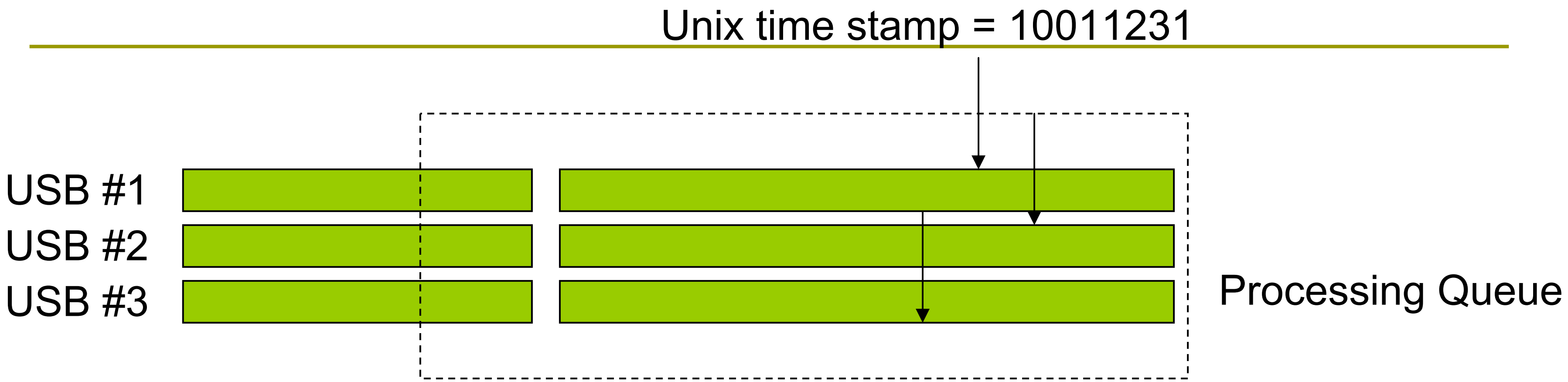
OR

  - We must exclude events which come from data packets that span a sync pulse.

OR

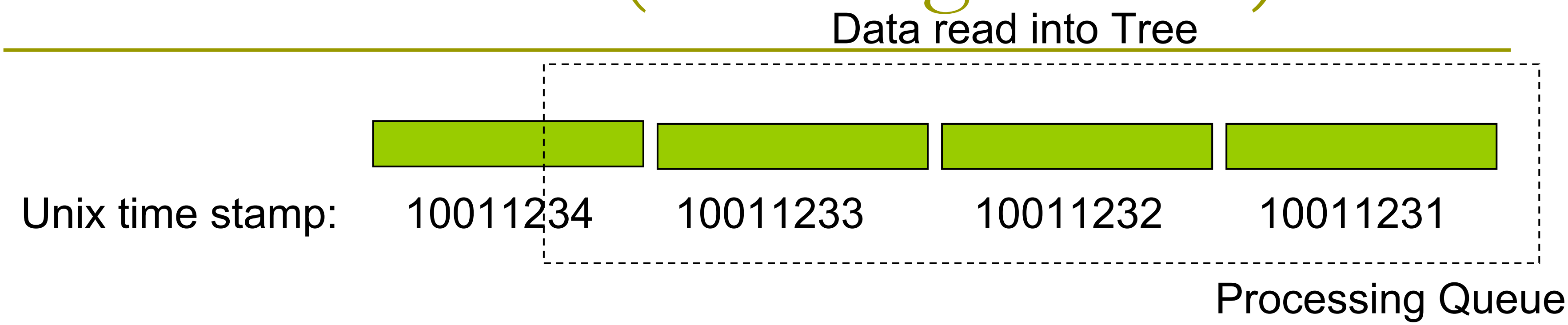
  - We must handle these events as special cases with larger coincidence windows corresponding to the inaccuracies in our knowledge of the sync pulse (not discussed here)

Event Builder (reading raw data)



- Stored in memory:
  - Time of last sync pulse (TOLSP)
  - Last unix time stamp processed
  - Max timing offset (MTO)
- Add all data packets to Tree with a corrected clock counter variable derived from timing offsets until next unix time stamp is found.
- If sync pulse occurs during time stamp
  - Find max clock counter (MCC)
  - For events with clock counters < 312,500,000 (5s), increment TOLSP
  - Flag data packets whose clock counter is within MTO of 0 or MCC as exceptions
- If more than 64 PMT's are used, store 10*(PMT board number) + USB board number instead of just the PMT board number to identify each PMT uniquely

Event Builder (forming events)



- Stored in memory:
  - Next unix time stamp to be processed (NTSP)
  - Next unix time stamp added to be added to the queue (NTSA)
  - Next TOLSP to be processed (NTOLSPP)
- Sort data from last few unix time stamps with TOLSP == NTOLSPP on clock counter
- Using a sliding *m* clock count window, check to see if there are any data packets from overlapping PMT modules (which are not flagged as exceptions) and form an event from these packets, if the event does not already exist.
- Once all data with unix time stamp == NTSP has been processed, flush this data and add packets with unix time stamp == NTSA to the queue.
- Alternatively, once all data with TOLSP == NTOLSPP has been processed, flush this data and increment NTOLSPP.

Event Builder – More Considerations

- For the same reason we cannot build events from ADC packets which span a sync pulse:
  - We cannot accurately time-correlate muons we detect to events in the rest of the DC detector if the combined event spans a sync pulse
- We can, however, accurately time-correlate events in both systems which occur completely within a given sync pulse (taking into account any timing offsets)

Conclusions

- GPS timing information for sync pulse distribution recommended
  - Cheap, accurate units commercially available (CNS Clock II, $1500, accuracy: <20 ns 1-σ)
  - Maximizes efficiency of OV and minimizes additional deadtime due to false triggers
  - Makes event builder conceptually simpler