Some ideas for DUNE DAQ Architecture, Triggering, Reduction (focus on single-phase)

Brett Viren

Physics Department

BROOKHAVEN NATIONAL LABORATORY

DUNE FD DAQ Workshop

Columbia, 30-31 OCT 2017

Outline

Some Numbers

Conceptual Design

Collection Plane Triggering

To Do

Some Relevant Numbers

^{39}Ar	1 Bq/kg $\times$ 10 kt / 200 TPC[1] = 50 kHz/TPC = 50/ms/TPC
SNB	$\mathcal{O}(0)$ event/ms/TPC, $\sim$1000 events/burst/10kt
Beam	$\sim$1Hz rate, trigger latency = MINOS: 0.6-5s, NOνA: 20s
APA	10 GByte/sec (full-stream @ 2 Byte samples)
10 kt	1.5 TByte/sec = 12 Tbps
PCIe	15 GByte/sec (v3x16 today, v4 will be $\times$ faster)
RAM	25 GByte/sec (DDR4-3200)
NIC	10 Gbps "common", 40 or 100 Gbps available
FFT	5ms / GPU card, 500ms / CPU-FPU core (2D, round trip, per plane, single-prec. FP)
NF+SP	Wire-Cell noise filter + sig.proc. 2 min/event (μB).
reco	protoDUNE LArSoft reco full chain 30-45 min/event.

[1] Aka "drift cells" aka "LAr volumes"

Some Numbers

Conceptual Design

Collection Plane Triggering

To Do

Concept In a Nutshell

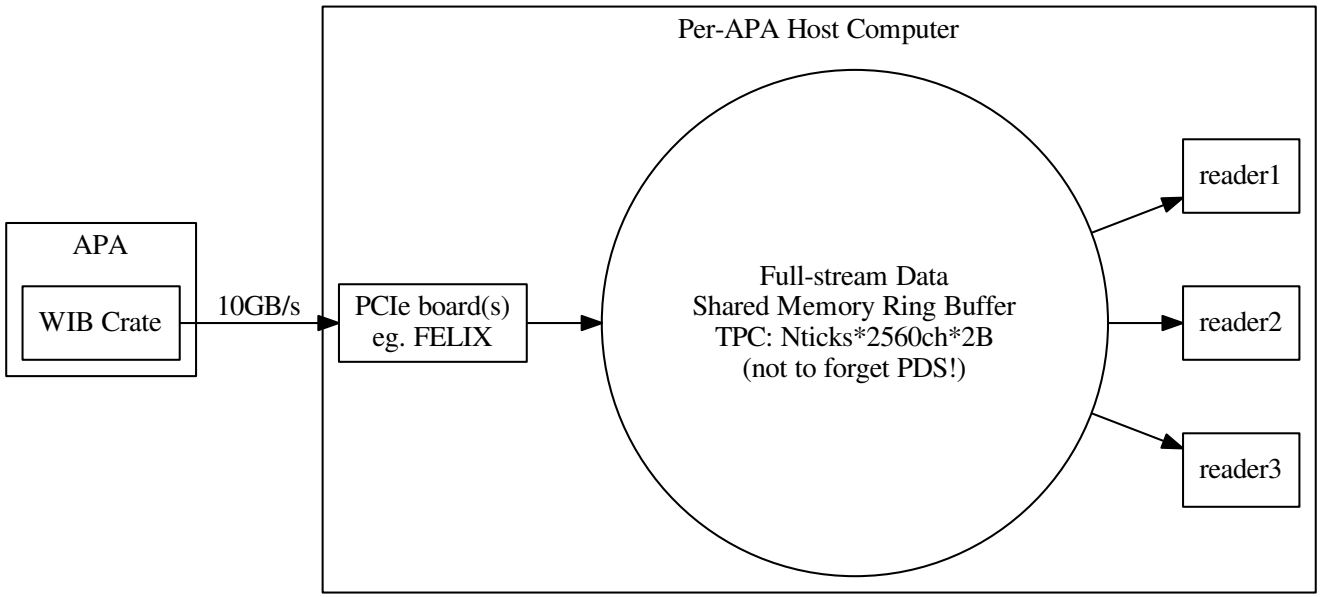
- **Flow** all data through **RAM in commodity computers**.
- **Process** data quickly while still in RAM.
- **Trigger** locally (per-TPC), process triggers globally, return trigger commands.
- **Interpret** trigger commands locally (per-APA) to dispatch requested subset of data.
- **Scale** commodity computers **up and out** as needed to accommodate above.

Minimal Concept Requirements: ADC to RAM

- Get data across cold $\rightarrow$ warm boundary
  - Introduce **no excess noise** in the process!
- Aggregate data through whatever needed stages (ADC, FEMB, WIB, FELIX) finally to RAM via **just a few, fast links**.
- Reliable, constant data transfer of 82 Gbps/APA.
- Perform **minimal processing** on the way:
  - **NO compression/decompression** needed/wanted.
  - **Reformat** data from hardware packing to "software-friendly" format for a **RAM ring buffer** (channel$\times$tick block and 12bit$\rightarrow$2Byte)
- A single interface board per host computer preferred for simpler firmware and software interface to RAM buffer.

WIB+FELIX provides a working example.

High Level, Minimal Concept



- Full-stream, **constant data flow** from ADCs host system RAM.
- Buffer N_{ch} TPC waveforms in N_{tick}-deep **shared-memory ring buffer**.
- Operate directly on data with "**reader**" processes running on `localhost`.

Call this a "Tier 1 Host".

Generalize and Scale

Likely one host is not enough to process all data from one APA.

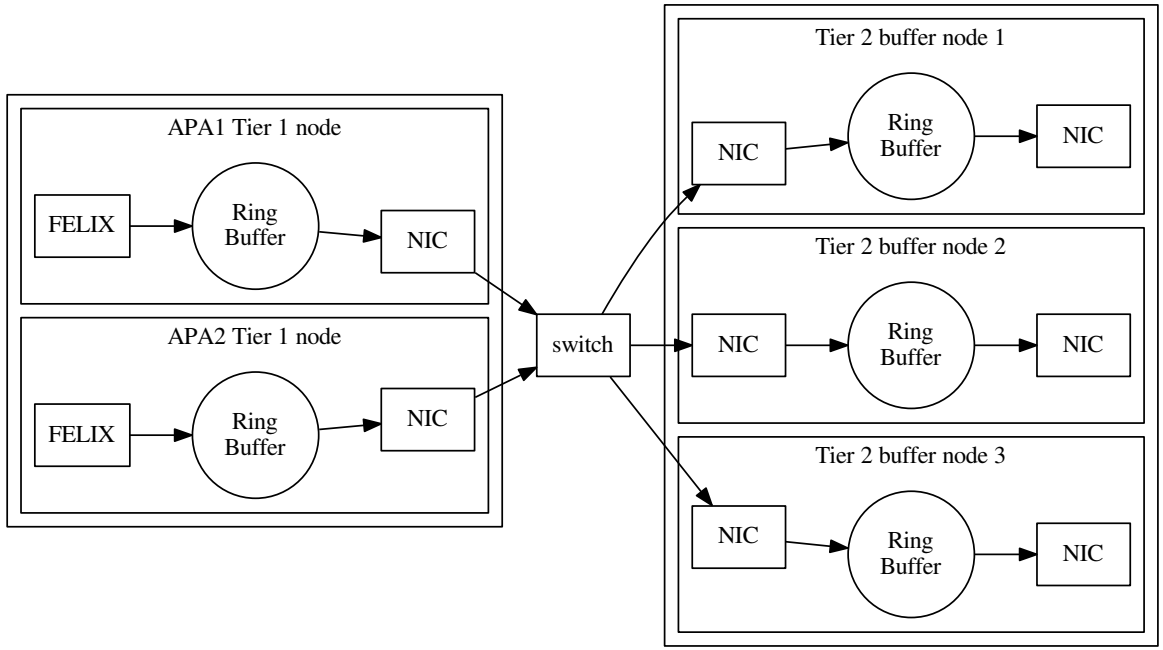
Build a **tiered hierarchy of hosts**.
Each generic host node has:

1. Data stream input source (FELIX, NIC).
2. RAM ring buffer.
3. Local processing.
4. Result stream output sink (NIC, disk).

Data flows generally "outward" toward higher tiers, but some upstream messaging needed (in particular global triggering).

Tier may specialize: RAM-heavy, CPU-heavy, have GPUs, perform meta processing (eg, trigger, DQM), etc.

Example: Scalable Ring Buffer



- Can make **deeper buffer** by flowing data through 2^{nd} (or 3^{rd}, etc) tier.
- Tier 1 "reader" processes would flow data between tiers at fixed rate.
- Tier 2 "writer" process replaces the FELIX spot in Tier 1
- A switch can allow dynamic routing, but the **12 Tbps/10 kt** total throughput may require some segmentation.

Note on RAM Buffer Size and Cost

How big must it be?

- Beam trigger packet latency: MINOS: 0.6-5s, NOνA: 20s.
- ? Latency of software trigger processors needs study.
- ? Latency for SNB trigger formation? (10s?)
  - ? once raised, can we **stream** SNB data to sink?
  - ? or do we need dedicated **long-term buffer** for whole SNB period?

Costs?

- Let's not speculate on Moore's law?[2]
- **Today:** RAM costs $\sim$\$10/GB (but it's currently **rising!**)
  - $\rightarrow$ \$100/APA·sec, **\$6k/APA·min**

$\rightarrow$ 1 minute RAM buffer is not so crazy.

- ✓ per-APA, costs less than RCE and comparable to FELIX.
- ✓ can avoid bespoke concentrated high-RAM systems by scaling-out, as above.

[2] Okay, I know you are all dividing by factors of 2 in your head!

Readers

Q: What do these "readers" do?

A: Whatever you want! (if you have enough CPU)

Some likely readers:

- Form and emit **local trigger primitives**
- Accept and interpret **global trigger commands**
- Perform data reduction or selection processing.
- Transfer data to 2^{nd} tier hosts for deeper buffer or more distributed CPU.
- Data quality monitoring, "express lane" processes.
- Save data to file.

Some Numbers

Conceptual Design

Collection Plane Triggering

To Do

Induction vs Collection Waveforms

- Raw **induction** waveforms
  - Biploar, no direct charge measure, often in the noise.
  - Any threshold will be inefficient or noise-dominated (or both).
  - Need relatively expensive signal processing to use properly.
  - Each channel is sensitive to activity on both sides of APA.
- Raw **collection** waveforms
  - Good measure of ionization energy as 2D profile (drift $\times$ transverse)
  - 1 view, so can only reduce data by time slices, no spatial reduction.
  - Immediately useful signals, no expensive processing required.
  - Activity from either side of APA can be distinguished.

$\Rightarrow$ Try to use **collection planes** to form basic **trigger primitives** independently for each **TPC** on either side of an APA.

Collection Waveform Segment Categories

Consider some **mutually exclusive categories** for collection waveform segments:

noise contiguous waveform chunks **consistent with noise**, eg contain no samples above some $n\sigma$ RMS level.

blips a set of waveform samples that are "**connected**", "**compact**" and "**isolated**" (by some metrics)

- Ie, "small, compact islands" in channel vs tick space surrounded by "enough" noise samples.
- Intention is to efficiently select a rich sample ^{39}Ar decays, SNB interactions and similar low-energy activity.

signals "everything else"

- Waveform chunks **not** consistent with **noise** nor **blips**.
- With more thought, additional categories may present themselves.

Local Collection Plane Trigger Primitives

Raise a local collection plane trigger primitive whenever any **non-noise waveform** is present.

Local trigger packet holds:

- **type** follows waveform category ("blip" or "signal").
- **ident** TPC number (ie, APA+face)
- **extent** rectangular extent in time and channel.
- **charge** baseline subtracted ADC sum over extent.
- **stats** any other stats which can be quickly calculated.

Example Global Trigger Logic

- **"SNB" trigger** accept stream of local "**blip**" triggers
  - Select blips above some charge threshold,
  - ignore blips inside "signal" trigger, veto known "extra blippy" TPCs.

  → trigger when remaining blip rate is above some threshold.
- **"High Energy" trigger** watch local "**signal**" trigger.
  - Merge overlapping local extents.

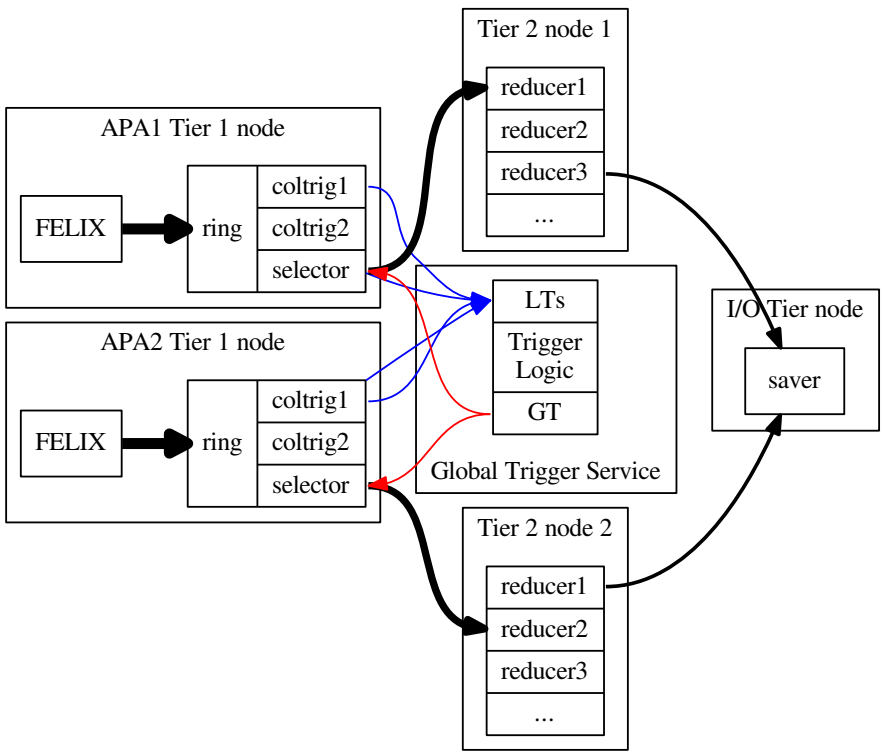
  → trigger on all.
- **Trigger command** sent back to **multiple, specific** APA host computers with readout instructions. Eg:

  cmd: Save select time region in all "signal" triggered APAs and their nearest neighbors.

  cmd: Save all data in all APAs for next N-seconds as a SNB is happening.

Depending on trigger type, triggered data may then undergo **further processing** (per-trigger data reduction, compression, etc).

One Possible Triggering and Data Flow



Just two APAs shown for simplicity.

- One collection trigger processor per APA face (`coltrig`)
- Global trigger command interpreter (`selector`)
- Farm selected data some worker (`reducer`)
- Save reduced data to concentrated sink.

? Lots of possible options.

- Trigger reduction is enough, no "reducers" needed?
- Single "reducer" stage is NOT enough?
- Add "stream branches" (DQM, "express lane")?
- Sink I/O is still too high for just one node, add more?

To Do

triggering much more thought is needed about local trigger categories, algorithms as well as global trigger logic and commands.

simulation use **Wire-Cell signal and noise sim** for accurate event "size" determination and estimating trigger alg CPU (GPU, etc) requirements.

scenarios develop 2 or 3 conceptual designs (eg, the one shown here, many another exploiting GPU accel, etc).

scaling Use sim to estimate the requirements for number and sizes of Tiers. Estimate throughput between tiers and final data rate to disk/tape. If "too much", iterate scenario definition.

costing Use commodity costs from today. If "too expensive", consider conservative Moore scaling and/or iterate the scenario(s).

Simulation Samples

noise + ^{39}Ar understand signal/noise energy threshold and the rate of local "blip" triggers.

- MicroBooNE (M.Mooney+co) now looking at ^{39}Ar in data.
- Can join forces.

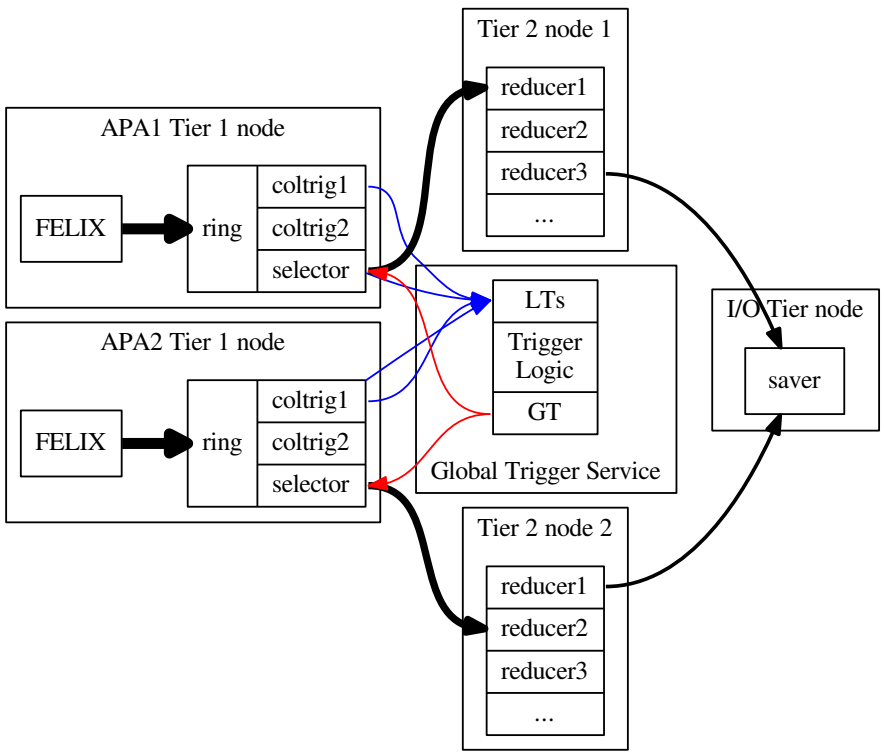
noise + SNB understand SNB trigger logic (ie, how to separate ^{39}Ar and SNB "blip" triggers).

high energy ν and cosmic-μ sim to understand local trigger alg's CPU requirements.

- For trigger-based reduction, their data size can be estimated with just flux, cross-section and dE/dX.
- More sophistication, eg signal-processing based reduction, needs understanding of processing time variances.

Extras

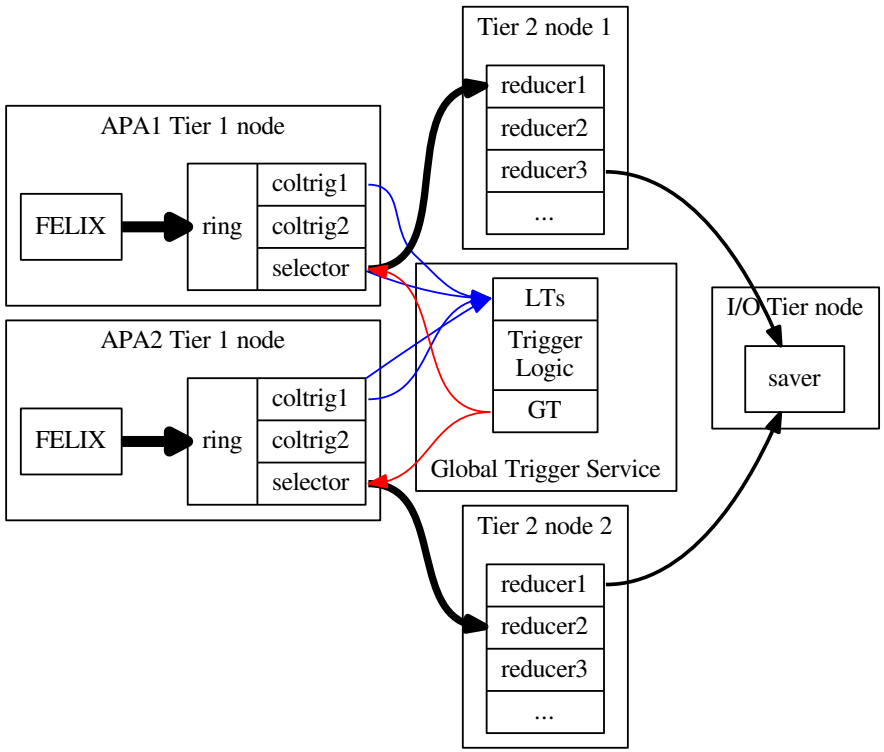
Infrastructure Implementation



Try to reuse as much as possible

- Ring-buffer implies async, distributed, real-time, multi-processing.
- Does/can this model map to artDAQ? (hope "yes"!)
- Many software roles:
  - RAM filled by FELIX Linux driver(?)
  - long-lived steam processors (`coltrig`, `selector`)
  - triggering is client/server model
  - chunked transfer / processing clearly artDAQ'ish (`selector`→`reducer`→`saver`)

What Might Go Where?



Possible locations:

under **Tier 1** and **Global Trigger Service**

above **Tier 2+**

- Gives some underground autonomy if network fibers cut.
- Need bandwidth to surface for just triggered readout and not full 12 Tbps.
- Allows scaling up (possible majority) of nodes (Tier 2+) in the cheaper, above ground environment.