Three Loosely Connected Things:

A Design Sketch,

Commodity Design Update and

Some General Concerns

Brett Viren

Physics Department

BROOKHAVEN
NATIONAL LABORATORY

DUNE FD DAQ Architecture – 28 Nov 2017

Outline

Overall Design Sketch

Commodity Design Conceptual Updates

Generic Concerns

Caveat

This design sketch is mostly to help clarify my own thinking.

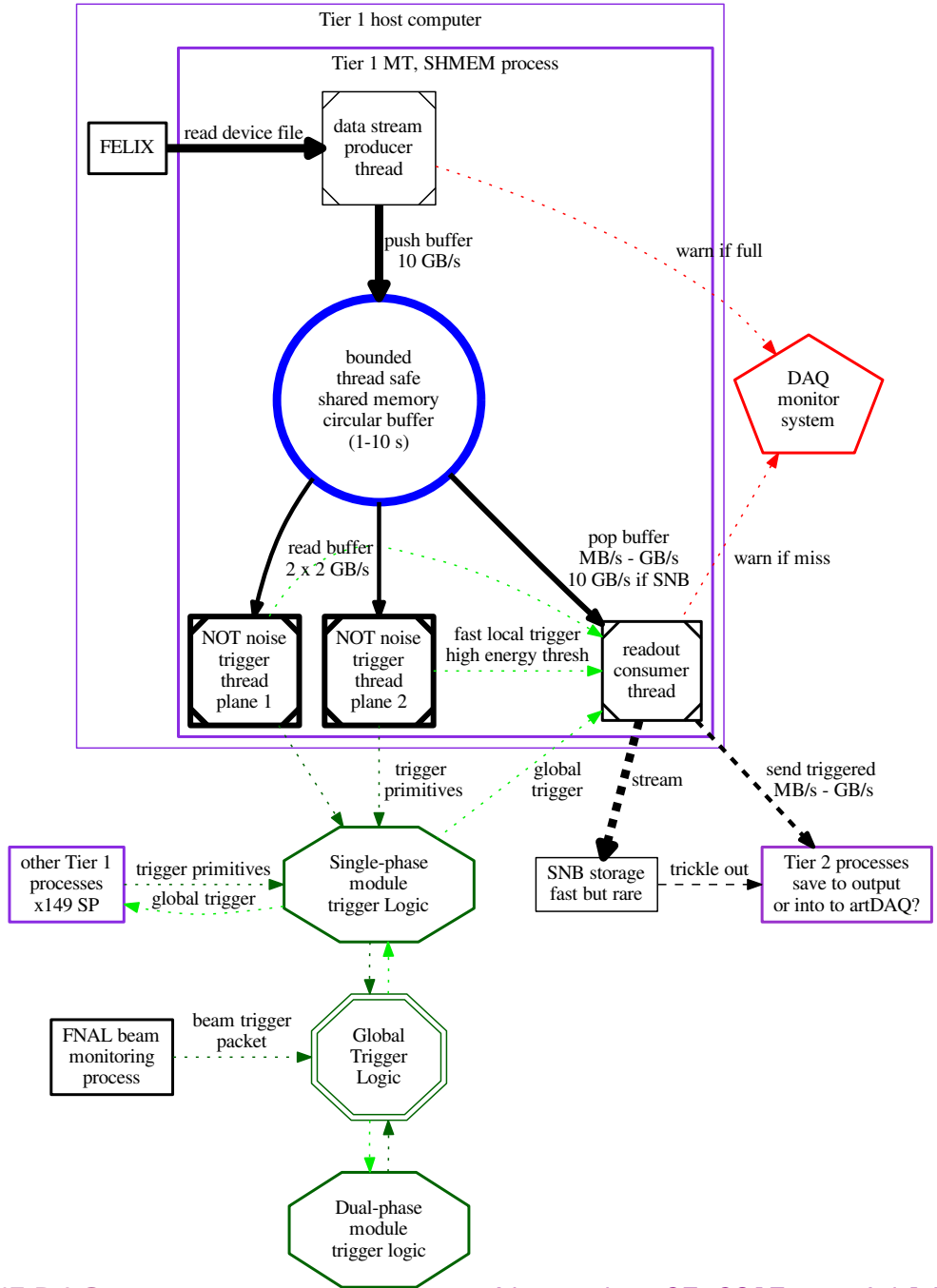
I don't mean to imply things are set in stone.

I will try to evolve this kind of drawing as we make decisions.

Design Sketch

Major parts

- Local Processing (LP)
- **Module Trigger (MT)**
- Global Trigger (GT)
- Monitoring

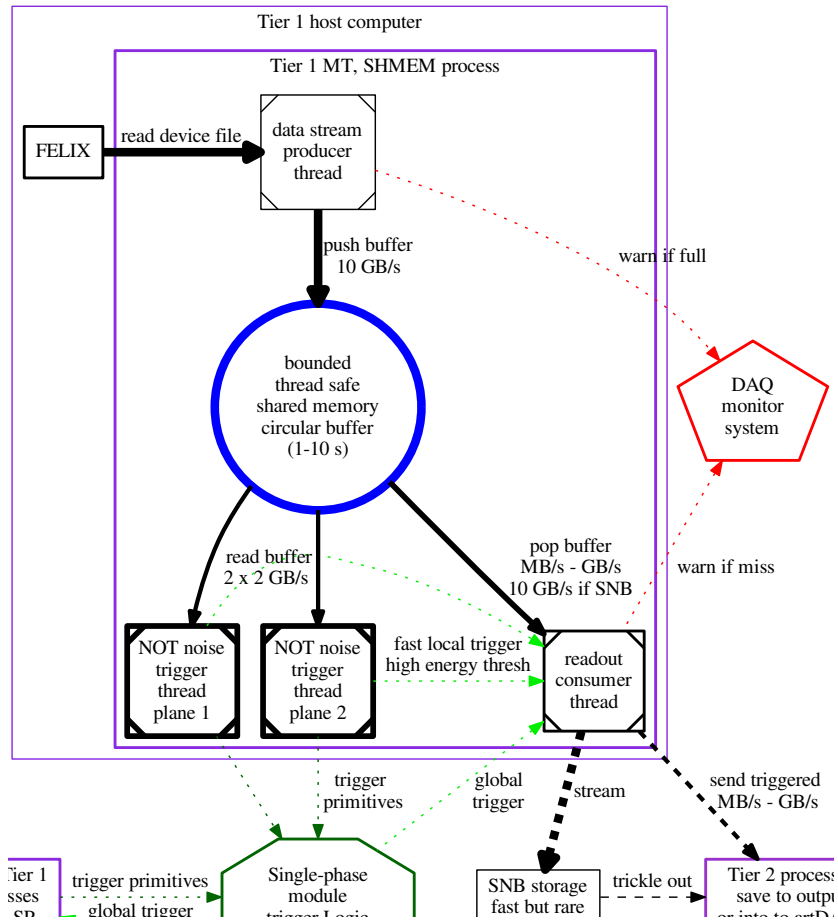


Local Processing (LP)

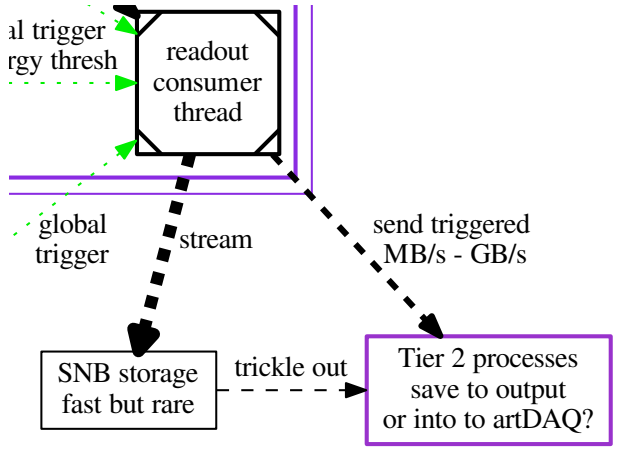
General duties/requirements:

- Local to one detector "chunk" (eg, one APA collection plane)
- Buffer full-stream data.
- Generate trigger primitives.
- Interpret trigger commands
  - → ie: "beam at time *t*"
  - ?: Allow fast "**knee-jerk**" triggers directly to readout?
  - → readout must combine trigger commands to avoid double-readout

(Replace box with your favorite tech.)

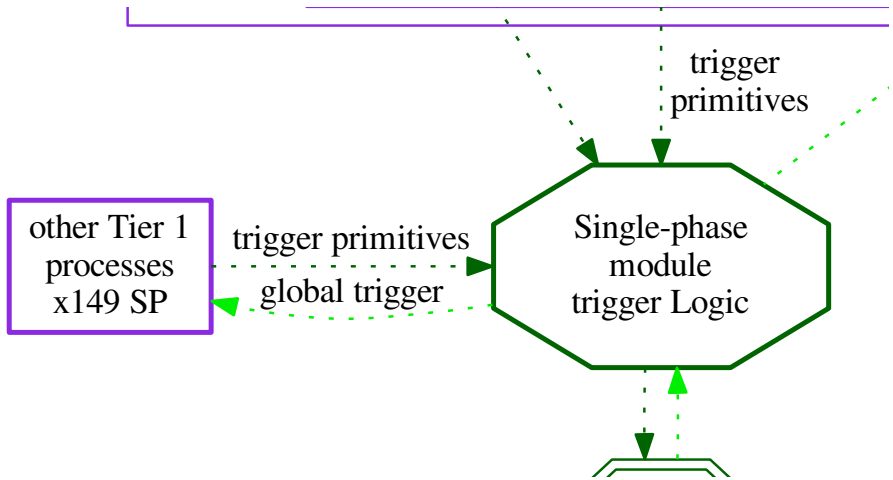


Local Process Data Egress



- LP, MT or GT send readout command: $\{(t, t + dt) \rightarrow \text{target}\}$
  - "target" indicates how to dispatch data
- Data now nominally in discrete time slabs, not continuously streamed.
- Inject into artDAQ for further reduction/compression, event building.
- Supernova Burst trigger command may need special processing:
  - ? Can we rely on the detailed "blip" triggers to select fine-grained time slices even during SNB? If so, maybe data rate is low enough to send through same artDAQ path.
  - ? O.w. must dump N-seconds at up to 10 GB/s to "local" storage.
    - Can find are only 2.5 GByte/sec NVMe PCIe cards,
    - so "local" may really need to be "distributed".

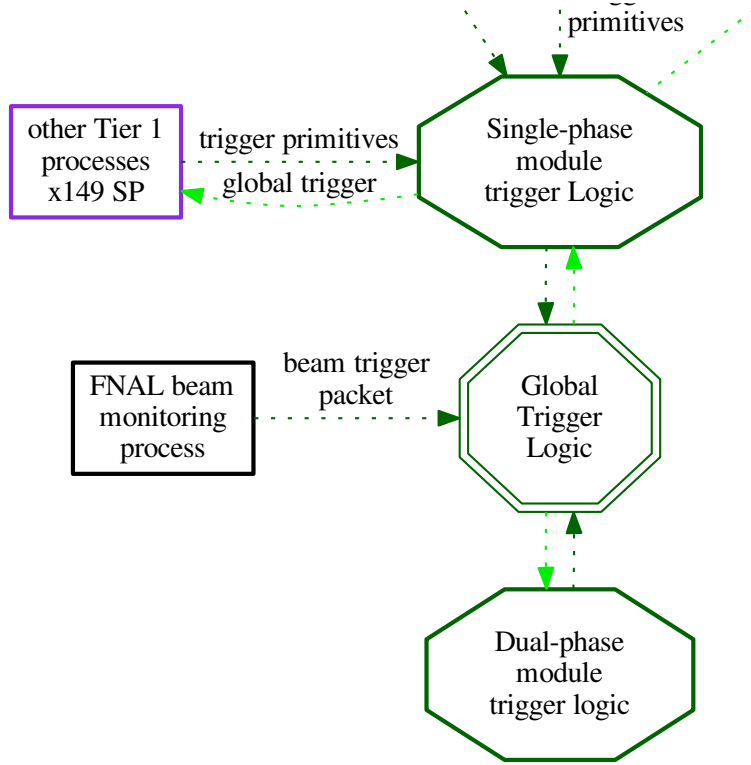
Module Trigger (MT)



- Each 10 kton module provide intermediate trigger logic.
- Combines its LP's trigger primitives.
  - ?: Allow for fast "**gut-feeling**" triggers directly back to LP?
- Sends module-level primitive to Global Trigger.
- Interprets module-agnostic GT commands into ones that LP understand.

Global Trigger (GT)

- Perform "**brain**" level triggering
  (continuing with the "**knee-jerk**" and "**gut-feeling**" metaphors)
- Accept FNAL beam timestamp packets.
- Accept external triggers.
- Accept module level primitives (SP + DP + #3 + #4)
- Return module-agnostic trigger commands.
  - $\{(t, t + dt)$, info$\}$
  - The "info" is enough to interpret to LP's "target": "beam", "SNB", etc.
- Send SNB candidate triggers to SNEWS (not shown)



Overall Design Sketch

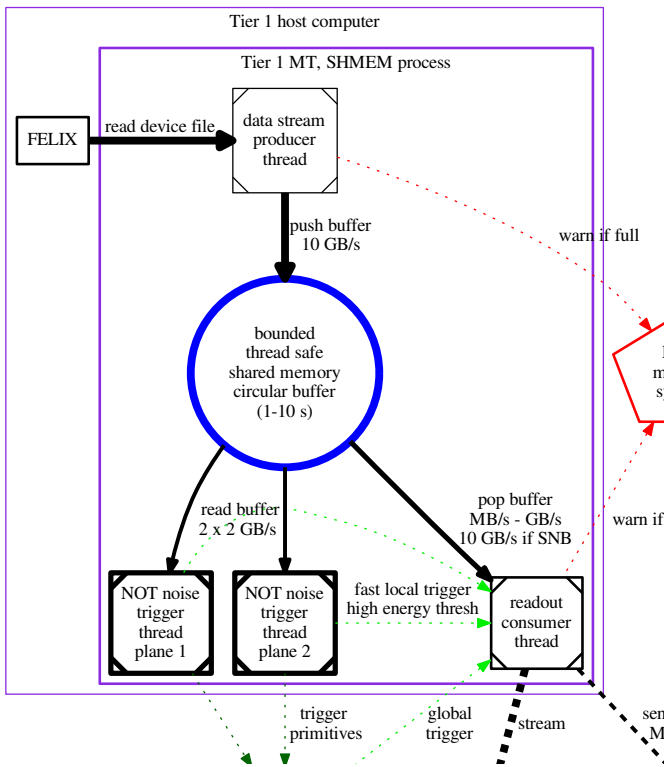
Commodity Design Conceptual Updates

Generic Concerns

Commodity Local Processing

- Input source: FELIX (or fast NIC if select "WIB-as-NIC" design).
- Multi-thread / multi-process.
- Shared circular buffer
  - Prefer "bounded" to "free", more later
- One thread per collection plane
  - Form LP's trigger primitives. Nominally anything NOT noise and NOT ^{39}Ar.
  - Try to design process as multi-threaded, streamed pipeline to use more cores.
- Began software tooling for development and evaluation environment:
  - Identifying existing multi-threaded implementations (mostly BOOST)
  - Wire-Cell Toolkit simulation
  - noise + ^{39}Ar + few μ/ν interactions
  - RAM-to-RAM buffer fill at 10GB/s

    → Incubating some ideas for a hardware-based devel/eval environment.



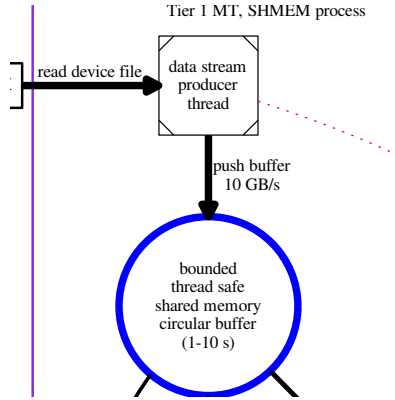
Overall Design Sketch

Commodity Design Conceptual Updates

Generic Concerns

Generic Local Processing Concerns - Keeping Up

What happens if/when
Local Processing can't keep up?



Consider two designs for LP ring buffer:

free running write to ring without limit.

- Reader and processors need thread-safe mechanism to know they have fallen behind and must have a heuristic for jumping ahead.
- Data is then lost but at least can notify Monitoring.
- **Upstream doesn't know** (and is probably happy about it)

bounded write only when buffer not full.

- Readout thread explicitly "frees" (advances ring marker)
- Processing threads may also hold back a "free".
- LP won't lose data but it **pushes the problem upstream.**

Both cases: what happens to inter-APA sync?

Generic Local Processing Concerns - Chunking

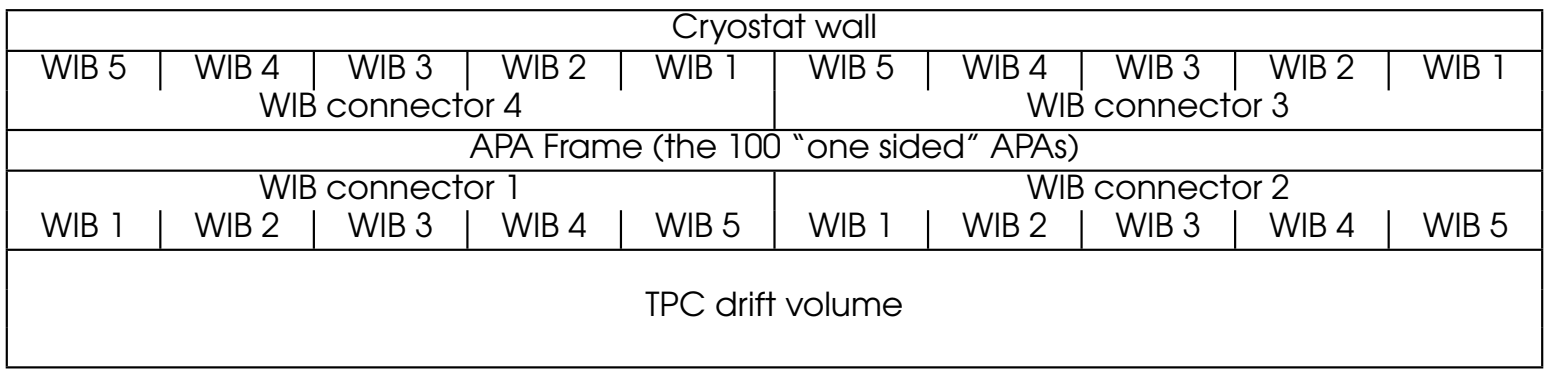
Dave Newbold:

> *(...) identify if there are irreducible sizings of data in time or space that must be seen by **one** algorithm to achieve data reduction.*

Consider Single-Phase : (key: ✓=ok, ×=bad, ?=dunno. For each: physics/computing)

- 150 full APAs. Fine for physics, high computation cost due to required deconvolution. ✓/×
- 200 collection planes (480 wires) which "see" the full drift volumes. ✓/✓
- 400 half-coll.-planes (240 wires) at WIB connector boundaries. Maybe ok for physics, artificial boundary in middle of drift cell, but spread computation to 4× more threads. ?/✓✓
- 2000 deci-coll.-planes (48 wires) at FE board boundaries. Same as above, but even more artificial boundaries. Still likely to be one unit of coherent noise. ??/✓✓✓
- In addition to channel chunks, also chunk into mutually exclusive time slabs. ???/✓✓✓✓

→ All need careful checking on quality simulation!



fin.