# Ideas on Nearline Processing for protoDUNE-SP

Brett Viren

Physics Department

BROOKHAVEN NATIONAL LABORATORY

DUNE S&C

17 Oct 2016

# Commissioning Experts Expectations

From a few discussions with some 35-ton and MicroBooNE commissioning experts that intend to work on protoDUNE/SP:

- The big problems typically require:
  - one or just a few triggers in time.
  - one channel up to a few FEMBs across.
- Need fast (few minutes) turn around time to run/re-run custom code.
  - $\rightarrow$ "A couple of hours wait here and there and the day is gone".
- Want to quickly locate specific file and copy to a local computer.
- Need to quickly read that file into familiar software systems.
  - Specifically mentioned: bare ROOT, Mathematica and Python/PyROOT.
  - These interfaces were actually used for 35t/uB.
  - But, at the cost of dumping raw file to a simpler file format.

$\Rightarrow$ Raw data in an accessible format will facilitate quickly finding and fixing detector problems during commissioning.

$\Rightarrow$ Enabling experts to use the software they are familiar with is important. We must not lock them in to using larsoft - they will "escape" in order to do their job.

# Non-commissioning Expectations

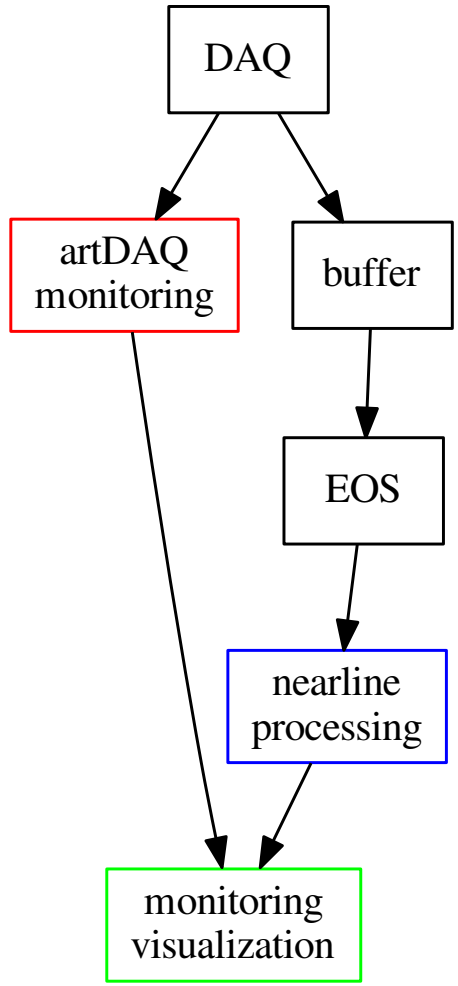
My bias:

- **Operational shifters** want **stable, succinct and well considered** results to assure them that current data taking is nominal.
- **Experts** want **specific, detailed** results to help them respond to problems found by shifters.
- **Collaborators** (in general) like some **high level** understanding of current status.

These imply different requirements than needed during commissioning.

# Prompt Processing Context

- pD/SP DAQ will nominally produce 25Hz trigger, 1.5GByte/sec data rate for 4.5 second spill, lower rate between spills.
- Disk storage dedicated to raw data transfer will be attached to artDAQ hosts and will temporarily buffer all data.
- Low-latency ($\lesssim$ 1 min) transfer from buffer to CERN EOS disk.
- Desire some immediate processing on some fraction of new data starts.
- Perform "nearline" or "prompt" provided ($\lesssim$ 10 min) processing on some fraction of the data as soon as it lands on EOS.
- Results in the form of human-consumable visualization.



# Three Prompt Processing Contexts

expert hacking grab file, hack, profit!

artDAQ monitoring run *art* modules on computer hosts on the DAQ LAN. Opportunistically, access some fraction of "live" data fragments via *art*'s UDP multicast.

nearline processing run arbitrary programs on raw data files served from EOS on computers hosts at CERN or in principle elsewhere around the world.

Each provides complimentary trade offs.

Note: also needed, but not considered here is "meta monitoring": health of artDAQ nodes (buffer disk space, temperatures, artDAQ job feedback). I think this should be handled by run/slow control systems. Yes?

# artDAQ Monitoring Considerations

+ Freshest access of the data! $\sim$ms after a trigger.
+ Some amount of processing may fit into current CPU budget, can cohost some monitoring with artDAQ processes.
+ UDP multicast means no/low I/O cost for the extra `read()`'s.
- Tight CPU constraint limits what job can compute.
- Input data are fragments, whole-event needs buffer `read()`.

Possible quantities:

- Data/fragment rate in/out and per artDAQ node.
- Some ADC-level stat sampling (requires decompression)?

$\rightarrow$ Ultimately we face a functionality/hardware balance.

  $\rightarrow$ With numerator and denominator still unknown.

# Nearline Processing Considerations

+ Can scale to many more cores by exploiting `neut`, wider CERN, or even FNAL and wider computing resources.
+ Has access to whole-event data.
+ Arbitrary programs can be run.
- Prescaling must be explicitly managed.
- Data (file) management and I/O becomes issues.
- Likely need more complex tiered system of processes.

Possible quantites:

- ADC-level: mean, RMS as running averages, strip charts, hit maps.
- Per trigger, whole detector "event" displays, by channel, ASIC, etc.
- Fast raw ADC reco (eg Michele's track finder)
- Deconvoluted signal quantities: total charge above noise per trigger, per channel, per ASIC, etc.
- Full reconstruction (Pandora, Wire-Cell, etc).

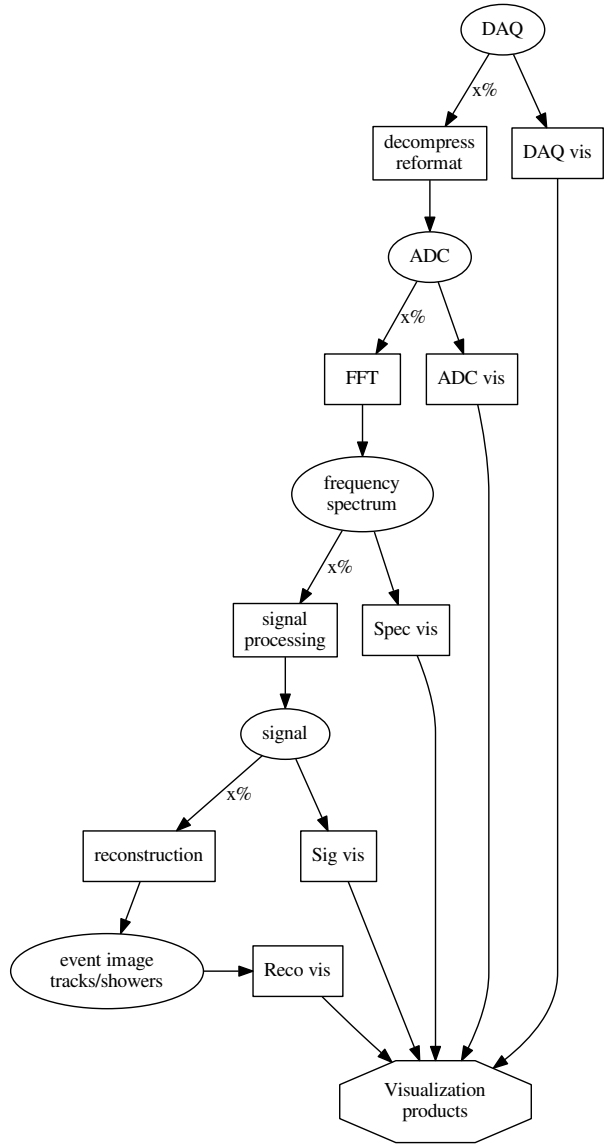
# Nearline Processing Requirements

- Some system is needed to manage nearline processing data, jobs and results.
- DocDB 1811 gives some initial requirements for that system
- It describes a chain of tiers of processing:

| | |
|---|---|
| DAQ | no decompression, only DAQ metadata. |
| ADC | decompression, reformat to ADC vs tick/channel. |
| FFT | ADC waveforms in frequency space. |
| Sig | After software noise subtraction and signal processing. |
| Reco | After running reconstruction algorithms |
| Vis | Visualization of output from the above tiers. |

# A Vision for a Nearline Processing System

- Construct DAG of processes (boxes) connected by data (bubbles).
- Data flows through DAG with prescale factors at the input of each processing node (the "x%"s).
- DAG processing is implemented as batch jobs.
- Intermediate files are marshaled by system and then removed.
- Visualization products collected for immediate and archival display.
- System enforces max latency by killing tardy jobs.



# Monitoring Visualization

- Does not need to be "fancy".
- I think it must be web-based and for on/off site viewing.
- Some pages must refresh automatically/periodically in order to display fresh visualization products.
- Desired special case UI: zooming/panning large graphics (eg, channel vs. tick).
- Need variety of curated pages targeted at each user expectation type.
- Must be responsive to novel and obsoleted visualization products as they are developed or removed.
- Need not be hooked into other systems (eg, RC/SC) but this is a bonus if it does not slow down its development.
- What else?