

# NP04 DAQ Status and Plans

Wes Ketchum

DUNEDAQ CT Meeting

24 June 2024

# Overall

- Things have gone pretty well
  - fddaq-v4.4.3 patch release tagged, built, and deployed Tuesday evening
  - DAQ was ready for beam on Wednesday evening
  - Various prescale configurations on beam trigger allow us to run with high instantaneous beam trigger rates
    - E.g. untuned beam with >250 Hz instantaneous rates, so *significant* pileup → prescales of x16 bring to reasonable DAQ rates
    - Indications are we handle a 30-35 Hz instantaneous rate reasonably well
  - CTB performing well, doing what it should
  - DQM adequate
- Most importantly, huge thanks for people who have been here / sent people here
  - Lots of positive interaction, efficient problem-solving, lots of quality of life improvements
- If this was the DAQ we had for the summer, we could make it work and meet the baseline physics goals

# Targeted improvements before the full beam run starting July 10

## TOP PRIORITY

- CRT readout debugging / fixes
- Optimization of networking in readout and dataflow
- Trigger merging / overlap / pileup handling

## HIGHLY DESIRED

- Stable and 'shifter-ready' TriggerPrimitive generation and running
- k8s running
- Standardizing DQM deployments

## (LIKELY) NEEDING DAQ SUPPORT

- Laser running
- PDS readout optimizations

# CRT Readout

- Haven't adequately tested CRT readout recently to diagnose DAQ issues
  - Reminder: possible segmentation fault pointing to pushing data from readout application to dataflow
  - Updates on handling of CRT processes should make testing easier now
    - Testing most robust if we can use some external-to-CRT-data trigger
  - Needs some time and attention
    - With some knowledge transfer from Wes on running the configurations

# Data flow

- We are admirably handling  $\sim 30$  Hz instantaneous rates in beam spill periods
  - From what we can tell in limited sampling of data, data quality is generally good
- However, we know there are issues
  - Push to slightly higher rates and we see trigger matching results come up periodically empty, higher rates of missed Rx packets, eventually trigger timeouts and inhibits
- Could use a push on multiple fronts
  - Can we upgrade readout servers to  $>10$ g output?
  - Do we have network configured well for readout and dataflow servers?
  - Do we have buffering, timeouts, pinning, etc. tuned well to optimize data rates given beam spill structure?
  - Are we properly isolating readout traffic and dataflow traffic?
- We are pushing at the edge, so likely hitting bottlenecks in both hardware and software
- Many aspects of investigation here coincide well with performance measurements, specifications on networking, and review of software

# Trigger merging / overlapping / pileup

- Merging logic is doing what it's supposed to, but not providing what's most desired for protoDUNE-II: veto a second TD if its readout overlaps with the first
  - Desired result: 3 ms readout windows for all trigger records, no two trigger records have any overlapping readout window
- Logic likely not difficult to implement, but some care/decisions to be made on ...
  - What trigger bits get set in TD? (e.g. if two TCs come in with same timestamp, what bits get set?)
- We should understand that the needs/configuration choices here are likely different than the far detector
  - But may not be so different than a high-pileup near detector...

# TriggerPrimitives

- TPG algorithms work\*, and in system where we aren't stressing the readout, seem somewhat well-behaved
  - \*generally keep up with data, don't seem to give complete garbage results?
- Still, there are a number of issues related to TP dataflow
  - TPs complicate readout server performance when we have high instantaneous rate of readout
  - Dynamics of TPSet creation seem ill-understood, causing downstream problems on trigger (ok, now effectively disabled), and tpwriter (Kurt took an initial look...) with either low or high rates of TPs
- What's needed?
  - It would be desirable to run with TPs at a reasonable threshold for 'physics' on at least one plane per APA
  - It would be highly desirable to run TPs at even a high threshold for use in overall detector monitoring
  - (not to mention how it'd be nice to have higher-level algs available for such checks too, but here there is detailed work that needs to be done to improve higher-level trigger alg performance...)

# K8s running

- We have configurations being made for k8s consistently
- But ... immediately hit a stumbling block this week in trying to run k8s with the CTB
  - I spent no time investigating, I'm not sure of the status here actually
- IMO: we need to make a strategic decision about what our goals are here
  - Onboarding applications into k8s seems to never be a trivial process
    - Maybe CTB, CIB (laser), and (NP04)CRT are the last ones?
  - I am not confident that we will be ready to run stably with k8s for shifters
  - I think we need a better deployment strategy if we intend to use this in actual detector operations
    - And if this is a priority, we need to apply that deployment strategy to NP02 at minimum



# DQM Deployments

- We've managed to hack together a number of processes to make some version of online DQM available
  - Scripts for copying portions of data to rscratch area, and managing the size of data in that area
  - 'analyzer' processes that can process data in that area and produce plots in an interval
  - a new 'display' simple web-page that shows those plots and automatically update them
  - Some version of 'justintime' looking at data in that area
- These are all currently running out of tmux sessions or crontab entries
- Now that they area all ~defined, and not requiring specific data resources, likely we can handle this better
  - Need to revisit discussions on making that happen, while still acknowledging that there will be continuing development there → need to be able to redeploy easily

# DAQ Support Tasks

- Laser runs
  - I've not heard any additional plans, but I suspect the laser team will want to make use of this off-beam time to try some runs
  - Obviously we should push for integration with the CIB / triggered readout
- PDS optimizations
  - PDS team would like to try to lower thresholds, which means increasing bandwidth on self-triggered boards by running more links to FELIX
    - Firmware updates on DAPHNE
    - Some mapping updates in the DAQ (~easy)
    - Probably have to care more about optimizing readout on DAQ (doable of course, but work...)

# Discussion

- Other big things I've missed?
- Obviously we can't do everything in a short period of time, and we need to prioritize
  - What's most impactful for ProtoDUNE-II?
  - What's truly important and useful for DAQ?
  - What can we put effort into and reasonably achieve in the time available?