



# muon (g-2) Data Quality Monitor (DQM)

Aaron Fienberg  
MIDAS Workshop  
26 July 2017

## (g-2) DQM one year ago: ROME

June 2016 SLAC test beam: prototype DQM using ROME

- test beam run with 1.5/24 calorimeters
- DQM often fell far behind the DAQ
- frequent crashes
- turning on DQM slowed down event builder
- no local ROME experts to help us
- separate analysis software very difficult to maintain
- need DQM that will scale to 24 calos + subsystems

After test beam run, made decision to build new DQM system addressing the above issues

# Fermilab *art* framework and *midas-to-art*

*art* is an event-processing framework developed/maintained by Fermilab SCD

- used by (g-2), mu2e, NOvA, and others
- (g-2) unpacking/recon/analysis done in *art* modules
- *art* has its own file/data format
- *midas-to-art* plugin: reads MIDAS files into *art* jobs, translates MIDAS banks into *art* event data
- *midas-to-art* written originally for another (g-2) test beam, has been used successfully since
- strong preference in collaboration to develop *art*-based DQM so we can reuse offline modules

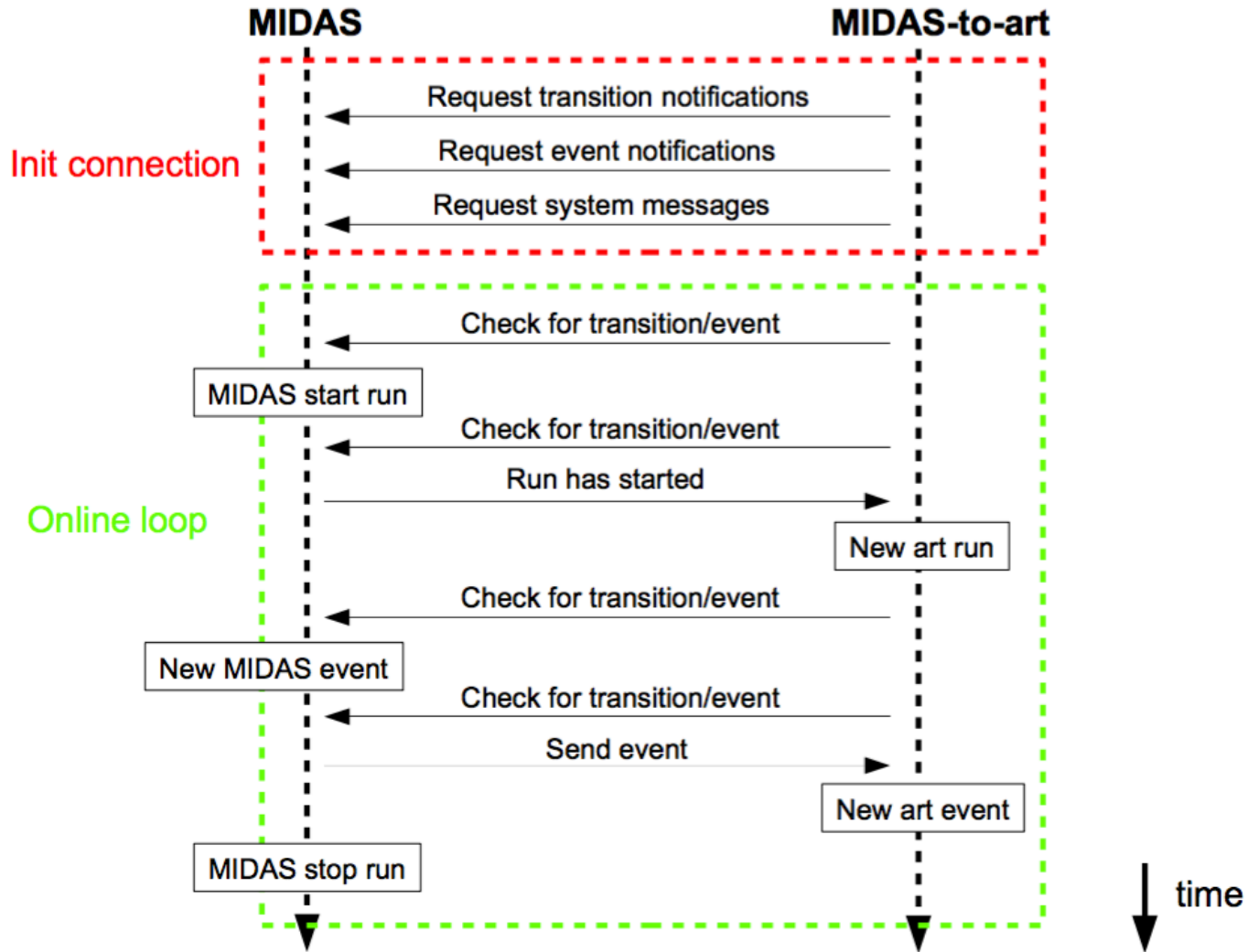


# ***midas-to-art* online extension added last summer**

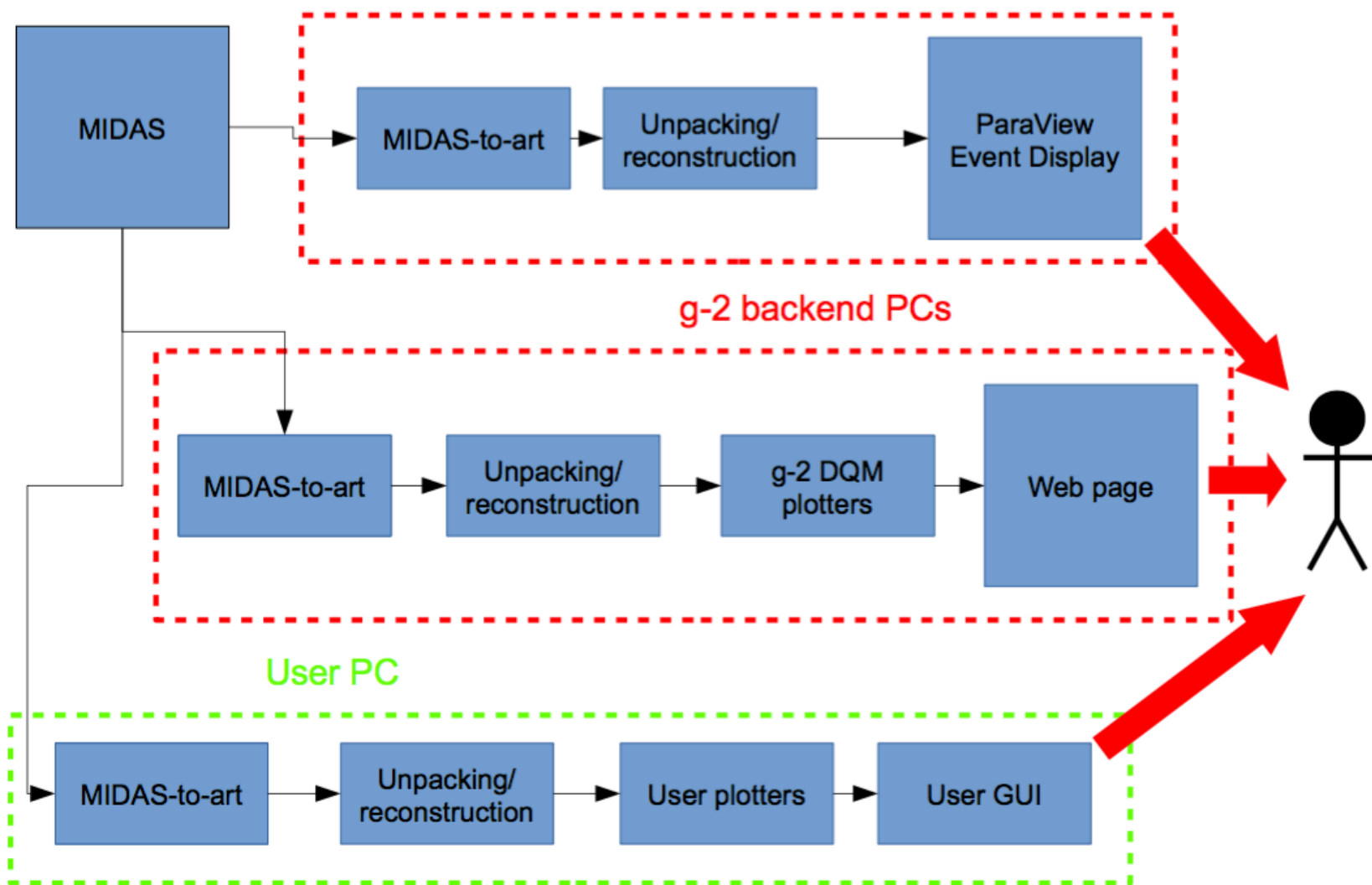
couple configuration parameters to change *midas-to-art* from offline-mode to online-mode

- connects to mserver like a remote analyzer
- uses GET\_NONBLOCKING
- *art* events created in online-mode are identical to those created in offline-mode
- allows us to run our highly tested, optimized, and multithreaded reconstruction and analysis modules online without any modification at all
- *art* can be used as backend for number of different frontend DQM pages/event-displays, etc.

# midas-to-art online



# potential system



# issues with *midas-to-art* online

- *midas-to-art* saves ODB XML string for each run, using *db\_copy\_xml()* as remote client proved so slow that we would miss first 10-20 events of a run (many seconds to execute command)
  - workaround: run remote ssh command to get ODB xml string
- *art* is very strict about run and event transitions, events coming in after end of run transition would cause art to throw an exception and exit
  - workaround: set very high end of run transition priority and ignore events that come in after end of run
- otherwise, *midas-to-art* online has performed very well and very reliably



# we use ZeroMQ to get data out of the art job

- *art* is not particularly interactive, need to stream data out
- we are using the ZeroMQ messaging library for this
- bindings in many major languages
- simple *art* analyzer modules take unpacked/reconstructed data and publish them
- data picked up by web GUIs, event displays, etc.



[zeromq.org](http://zeromq.org)



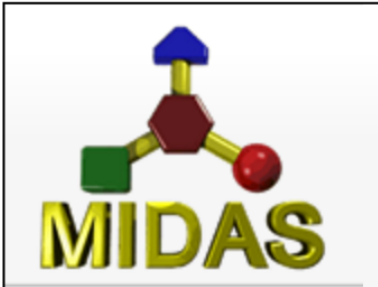
# main DQM webserver built with node.js

- server-side javascript runtime environment
- event driven architecture
- asynchronous, non-blocking IO
- built for scalability and throughput
- numerous mature, open-source plugins with lots of community support



# General Architecture

bidirectional  
communication



midas experiment/mserver



midas-to-art  
unpackers  
producers  
analyzers

any module we've written can  
run online  
(limited only by speed)

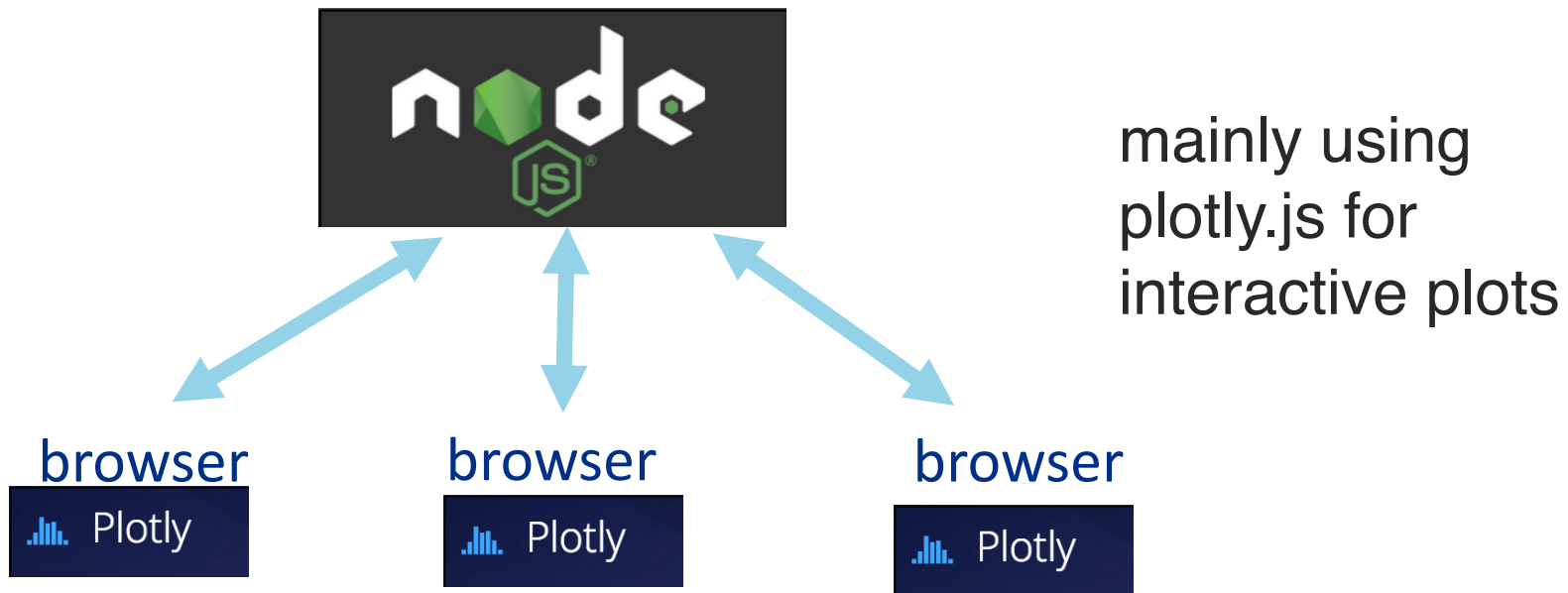


node.js webservers  
data aggregation  
client communication

connections to multiple clients

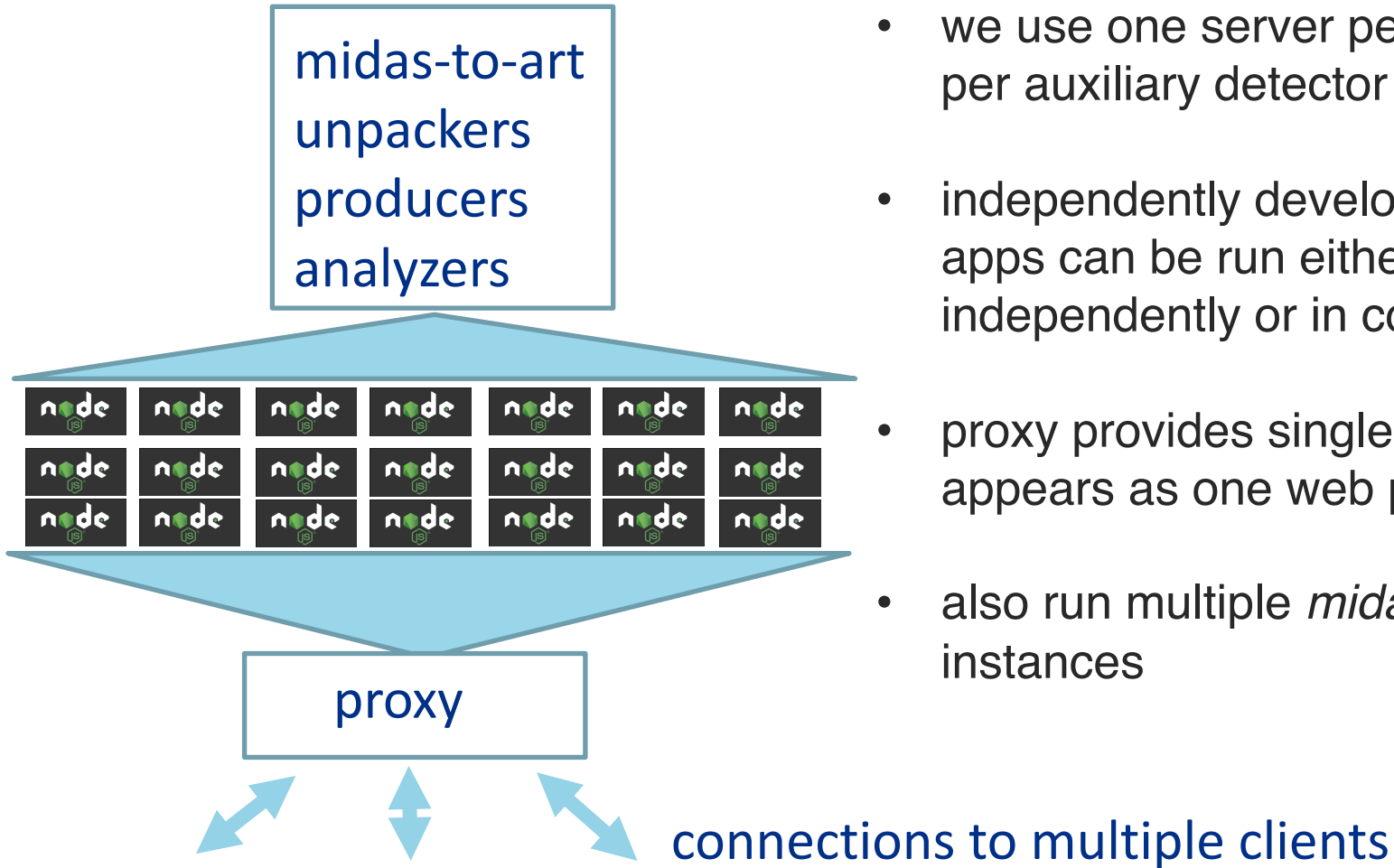


# plot rendering is outsourced to clients



- server sends data through WebSockets on client request
- clients are insulated from both *art* job and MIDAS
- we use some additional js visualization tools (D3.js)

# load distribution and modularity



- we use one server per calo + one per auxiliary detector system
- independently developed DQM apps can be run either independently or in concert
- proxy provides single point of entry, appears as one web page
- also run multiple *midas-to-art* instances

## uses and tests so far

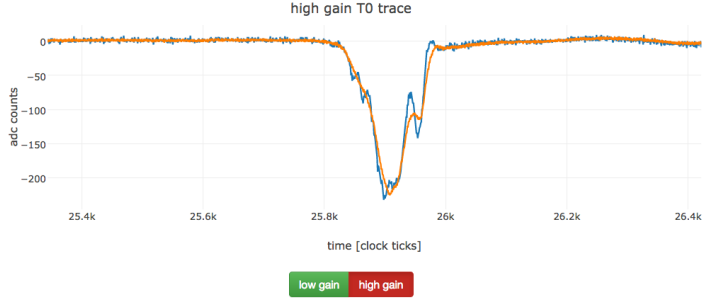
- initially tested on full-scale, full-rate 24 calo AMC13 simulator experiment, no apparent issues after a few days of running
- used in calorimeter, tracker, magnetic field test stands
- used for DAQ tests over the past year
- used during recent (g-2) commissioning run
- was able to serve numerous remote and local clients during the run (about 20 peak concurrent users)

# example DQM pages

# injected beam intensity and storage efficiency

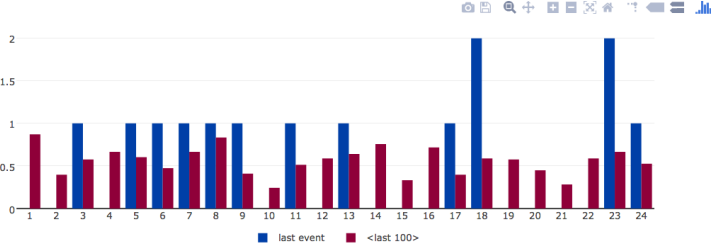
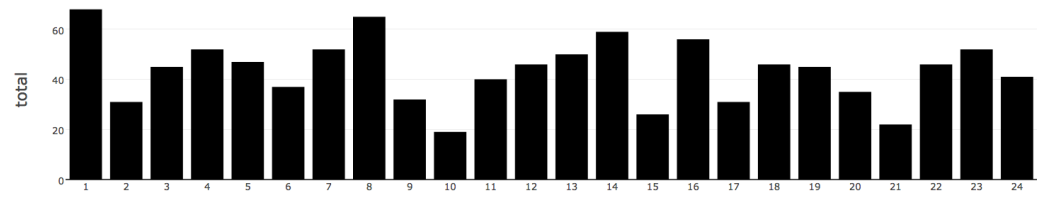
## CTAG

clear	NFILLS	TOTAL	LAST	AVG (last 100)
TO INTEGRAL	78	5.03e+6	6.48e+4	6.45e+4
ALL CALO SPLASH	78	50733	666	650.4
ACROSS RING SPLASH	78	3657	49	46.9
ALL CLUSTERS	78	14749	239	189.1
PROTON LAUNCH	78	7189	112	92.2
DECAY POSITRONS	78	1043	14	13.4



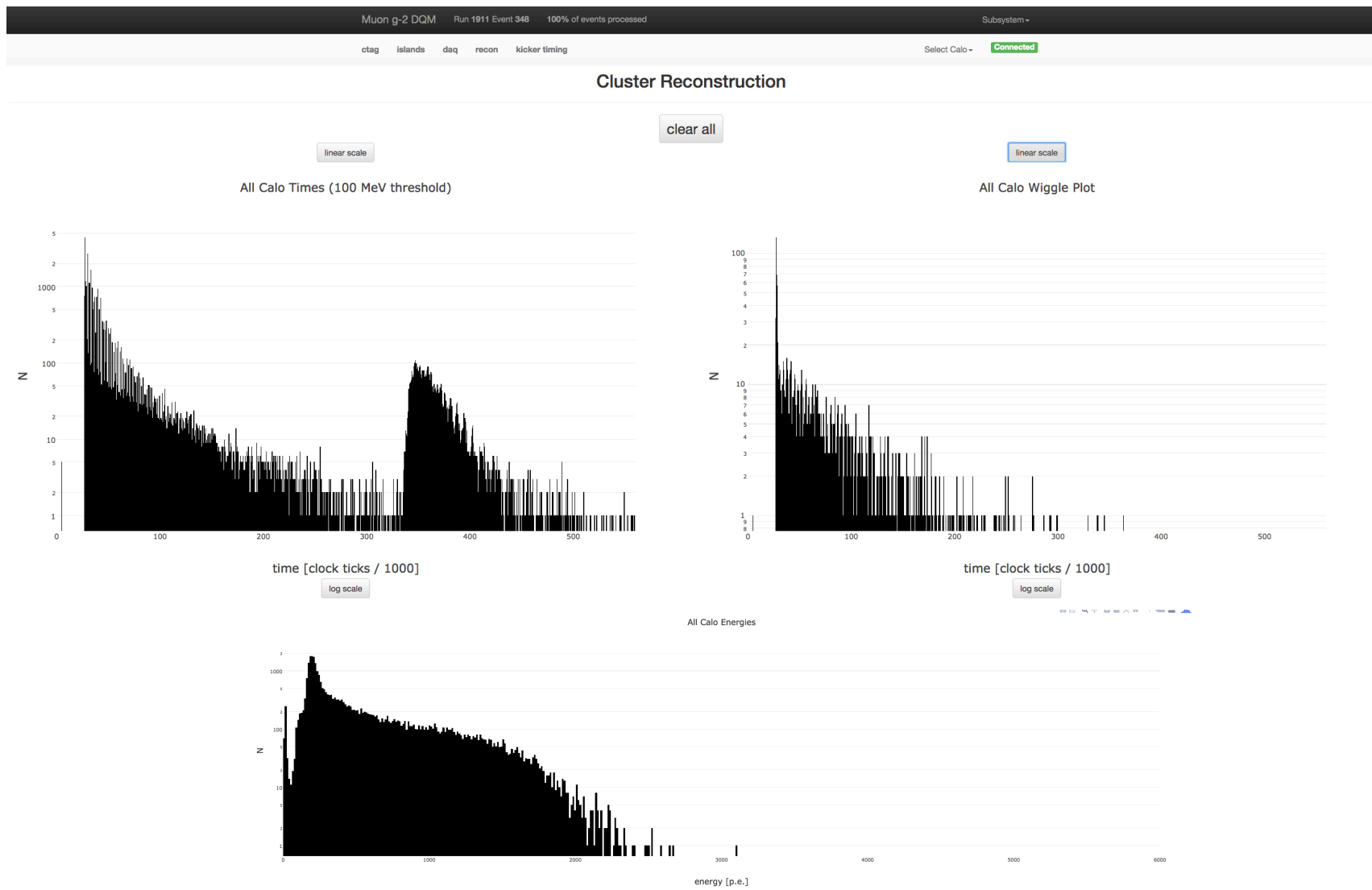
click on table to change plots below

## DECAY POSITRONS





# reconstructed calo hit times and energies



# positron pulse in a calorimeter

Muon g-2 DQM Run 1911 Event 348 100% of events processed

Subsystem ▾

ctag islands daq recon kicker timing

Calo 8 ▾ **Connected**

## calo 8 traces

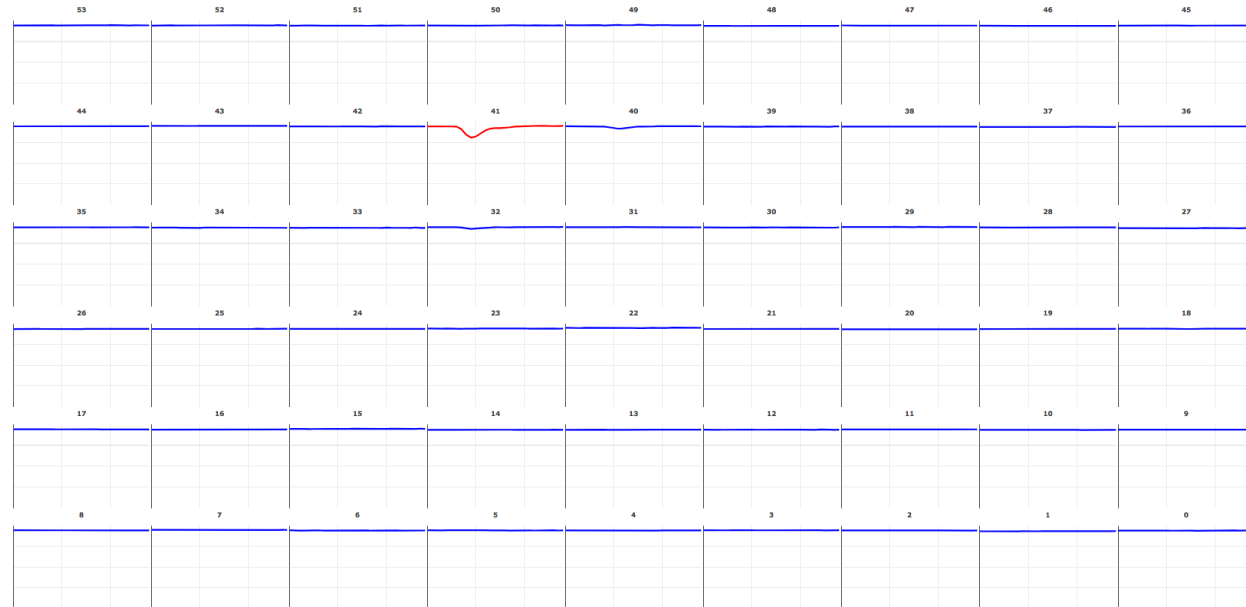
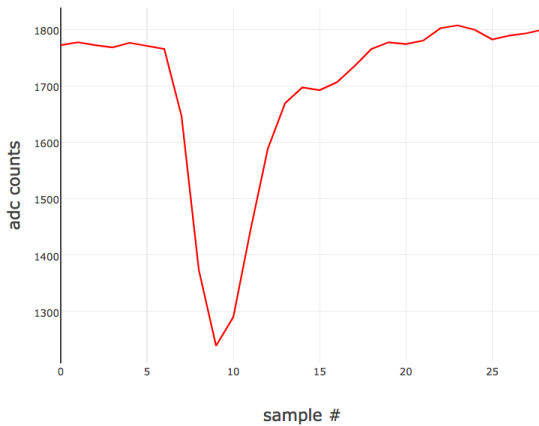
RUN 1911 EVENT 347 ISLAND 12

summary traces Q S recon laser headers

auto update: ON **pause** update late ▾

late island: first sample number 124428

xtal 41



# straw tracker TDC hits

Muon g-2 DQM Run 1912 Event 5

Subsystem

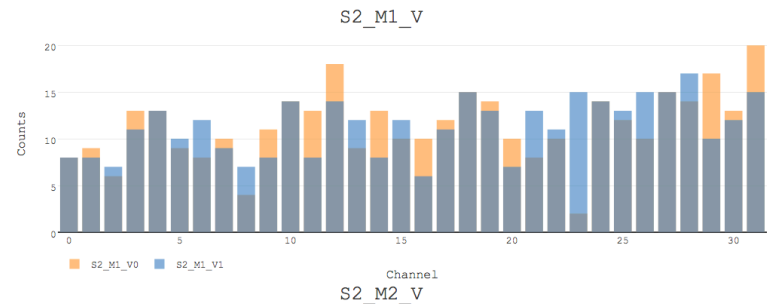
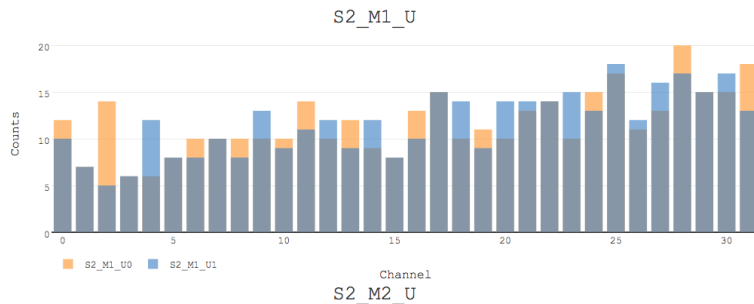
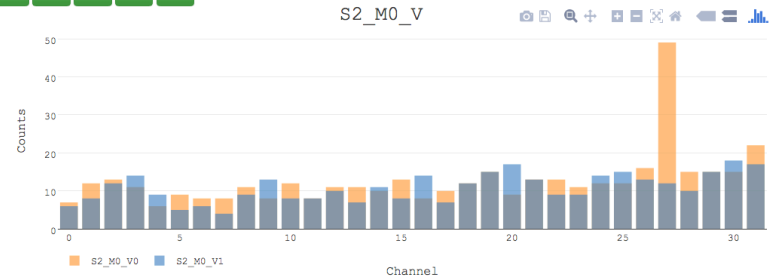
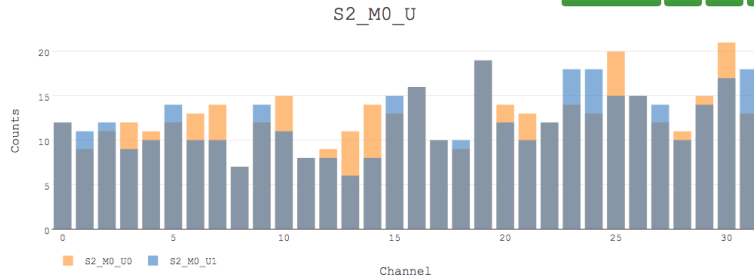
Straw Tracker Overview Station 2 Physics Plots

Connected Clear Histos

<b>Module 0</b> Straw Tracker Module 9 Fitted 127/128 Straws Present VB12 Not Present	<b>Module 1</b> Straw Tracker Module 2 Fitted 127/128 Straws Present VB24 Not Present	<b>Module 2</b> Straw Tracker Module 0 Fitted X/128 Straws Present	<b>Module 3</b> Straw Tracker Module 1 Fitted 127/128 Straws VB14 Not Present
<b>Module 4</b> Straw Tracker Module 6 Fitted 128/128	<b>Module 5</b> Straw Tracker Module 5 Fitted 126/128 Straws UA8, UA17 Not Present	<b>Module 6</b> Straw Tracker Module 00 Fitted 127/128 Straws (Straw 31)	<b>Module 7</b> Straw Tracker Module 4 Fitted 127/128

## Channels

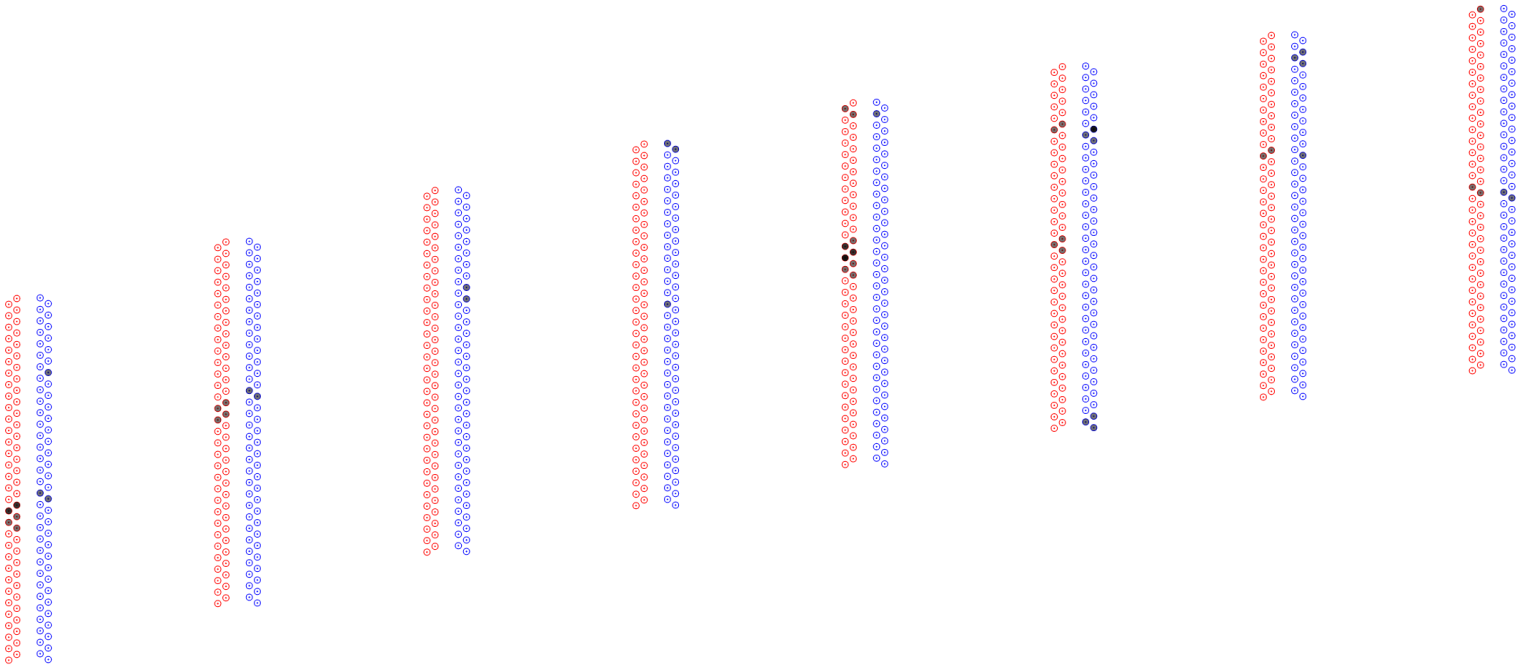
Tracker 2 INFO M0 M1 M2 M3 M4 M5 M6 M7



# straw tracker occupancy



Middle



# magnetic field NMR probe health and FID traces

Muon g-2 DQM

Run806 Event552

Subsystems



## Ring Yoke Probe

Update Options

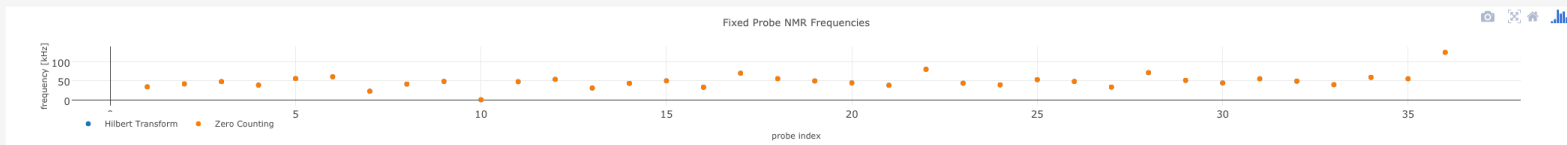
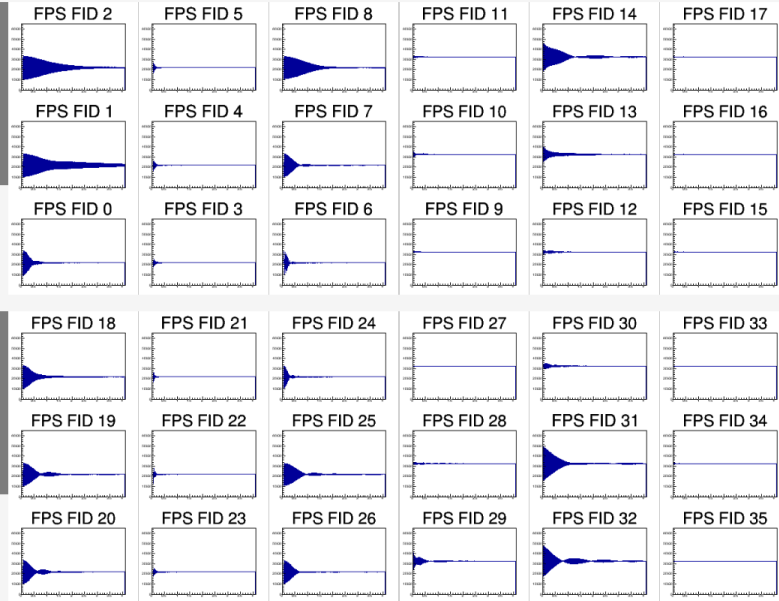
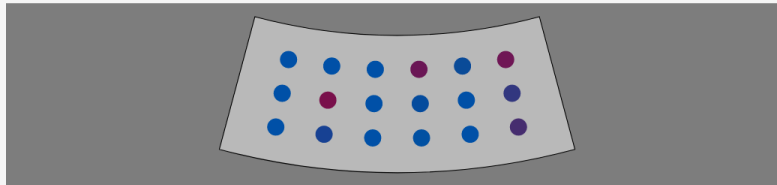
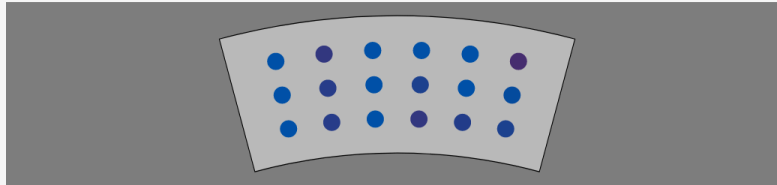
Refresh

Automatic On

Every 5 seconds

Display Options

Yoke A



# single NMR probe FID and frequency histories

Muon g-2 DQM

Run806 Event552

Subsystems



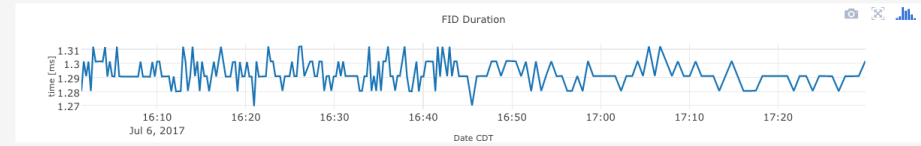
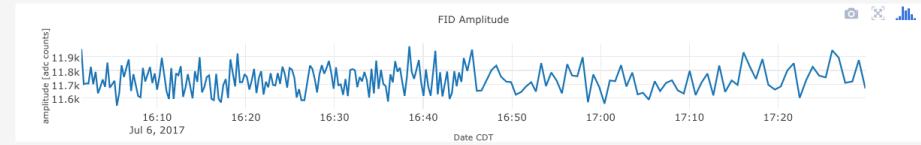
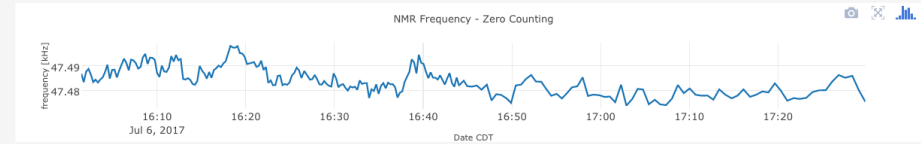
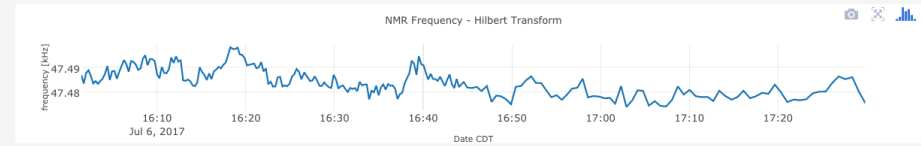
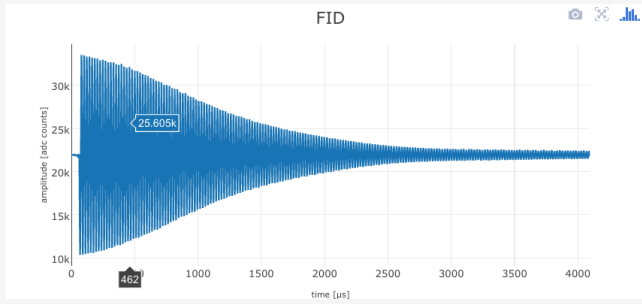
## Ring Yoke Probe

**Update Options**

- Refresh
- Automatic On
- Every 5 seconds

**Display Options**

- Probe 3
- Yoke A
- Top Layer
- Azimuthal - 1
- Radial - Inner



# DAQ event builder monitor

Muon g-2 DQM Run 1911 Event 348 100% of events processed

Subsystem -

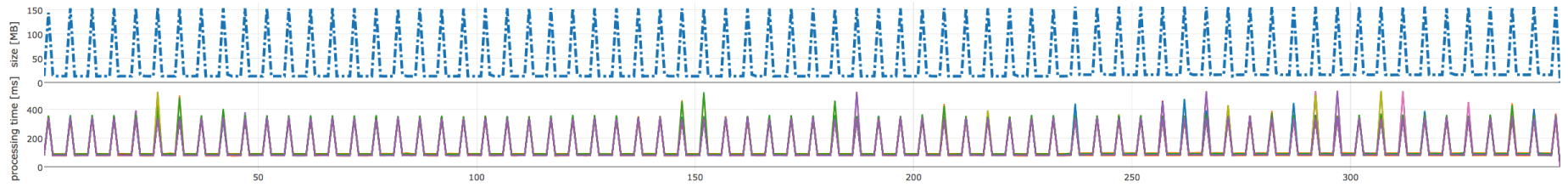
ctag islands daq recon kicker timing

Select Calo - **Connected**

## DAQ monitor

RUN 1911 EVENT 348  
EVENT SIZE: 0.02 MB

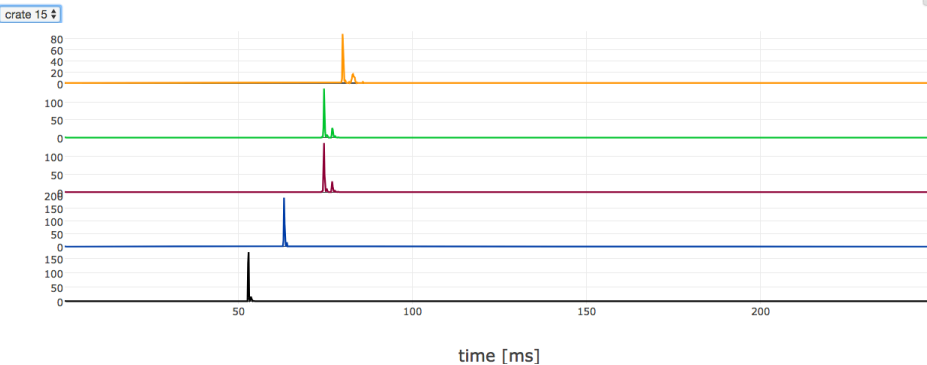
BANK LIST: TRIG GPS0 TTCA TTCB TTCZ FZ01 AC01 FZ02 AC02 FZ03 AC03 FZ04 AC04 FZ05 AC05 FZ06 AC06 FZ07 AC07 FZ08 AC08 FZ09 AC09 FZ10 AC10 FZ11 AC11 FZ12 AC12 FZ13 AC13 FZ14 AC14 FZ15 AC15 FZ16 AC16 FZ17 AC17 FZ18 AC18 FZ19 AC19 FZ20 AC20 FZ21 AC21 FZ22 AC22 FZ23 AC23 FZ24 AC24 FZ25 AC25 FZ26 AC26 STRW ESQ1



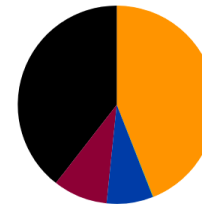
event number

calo 15 timing

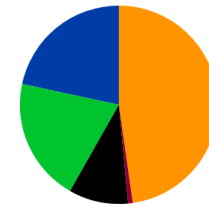
clear



Average



Last Event



- compression done
- got tcp data
- processed gpu data
- copied gpu data
- mfe start



# reflections after the run

- DQM system received positive feedback from users, remote collaborators especially appreciated ease of access
- beam and storage diagnostics were invaluable in making optimal use of our short beam time
- system was stable throughout the run, and was easily adaptable as new plots/diagnostics were requested
- *art*-users seemed to have an easy time adding DQM pages for their systems, whereas subsystems that had yet to develop *art* offline code had a more difficult time
- weakest link is network IO between mserver and *midas-to-art*, our events can just get very large (max about 2 GB)
- average data rate is fine, instantaneous rate can cause lost events (perhaps should implement buffering in *midas-to-art*)

# possible future directions

- 3-dimensional real-time event display with ParaView, reading data from the online *midas-to-art* job
- database integration allowing for easy generation of trend-plots based on analyzed parameters and for recording data necessary to quickly regenerate key DQM plots from old runs
- production reverse-proxy server with authentication so users can access without opening ssh tunnels
- Thank you!