GPUs for Data Acquisition and Simulation for the Muon g-2 Experiment at Fermilab

Wes Gohn*
Siemens Healthineers

Ran Hong
Argonne National Laboratory

*work performed in association with the University of Kentucky

Outline

- Wes Gohn:
  - Introduction to Experiment
  - Data acquisition for positron detection with CUDA
  - Monte Carlo simulation with CUDA

- Ran Hong
  - Introduction to the magnetic field measurement system
  - Analysis of the NMR signal with CUDA
  - Summary and Current Status

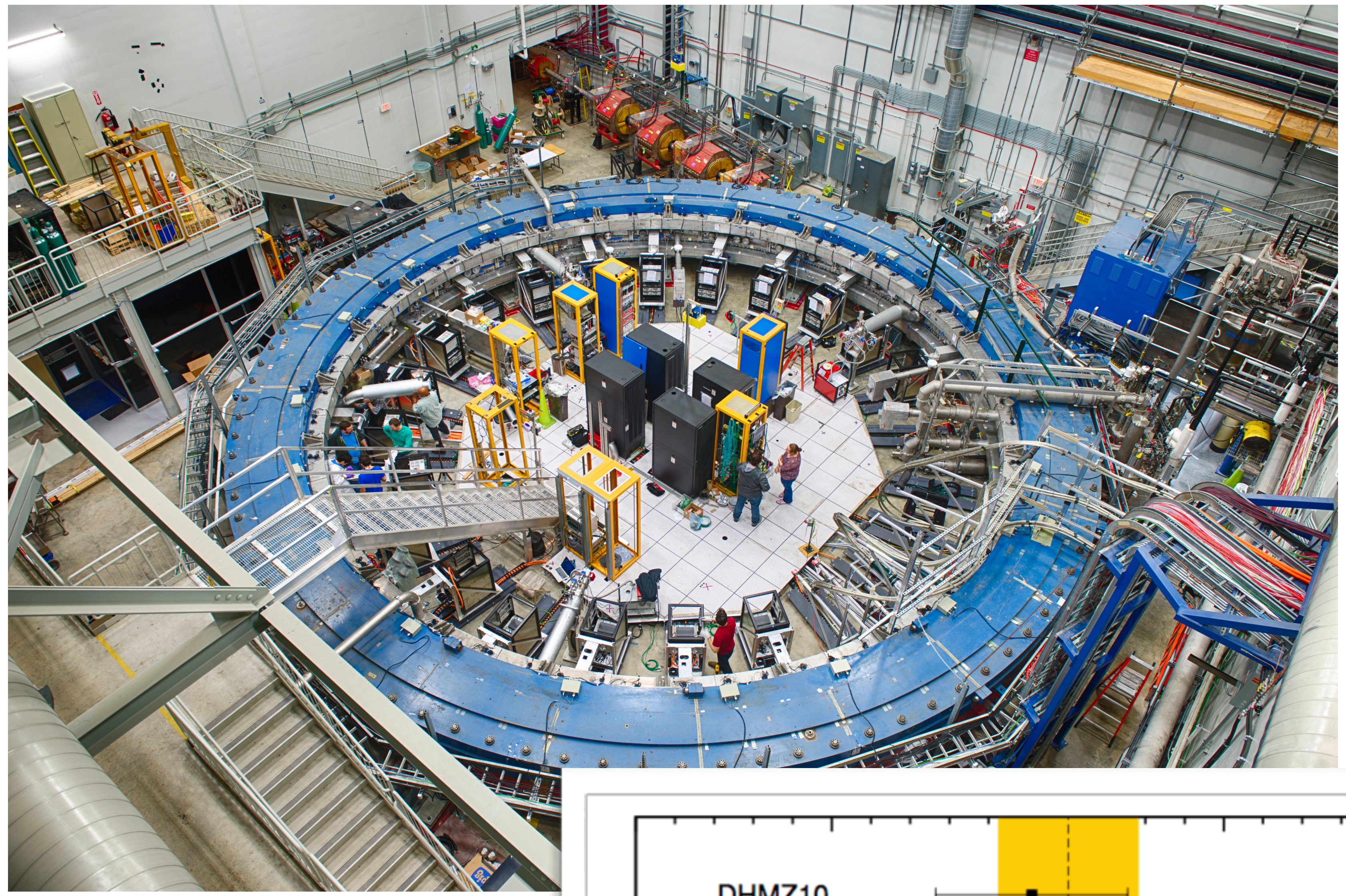
What is Fermilab?

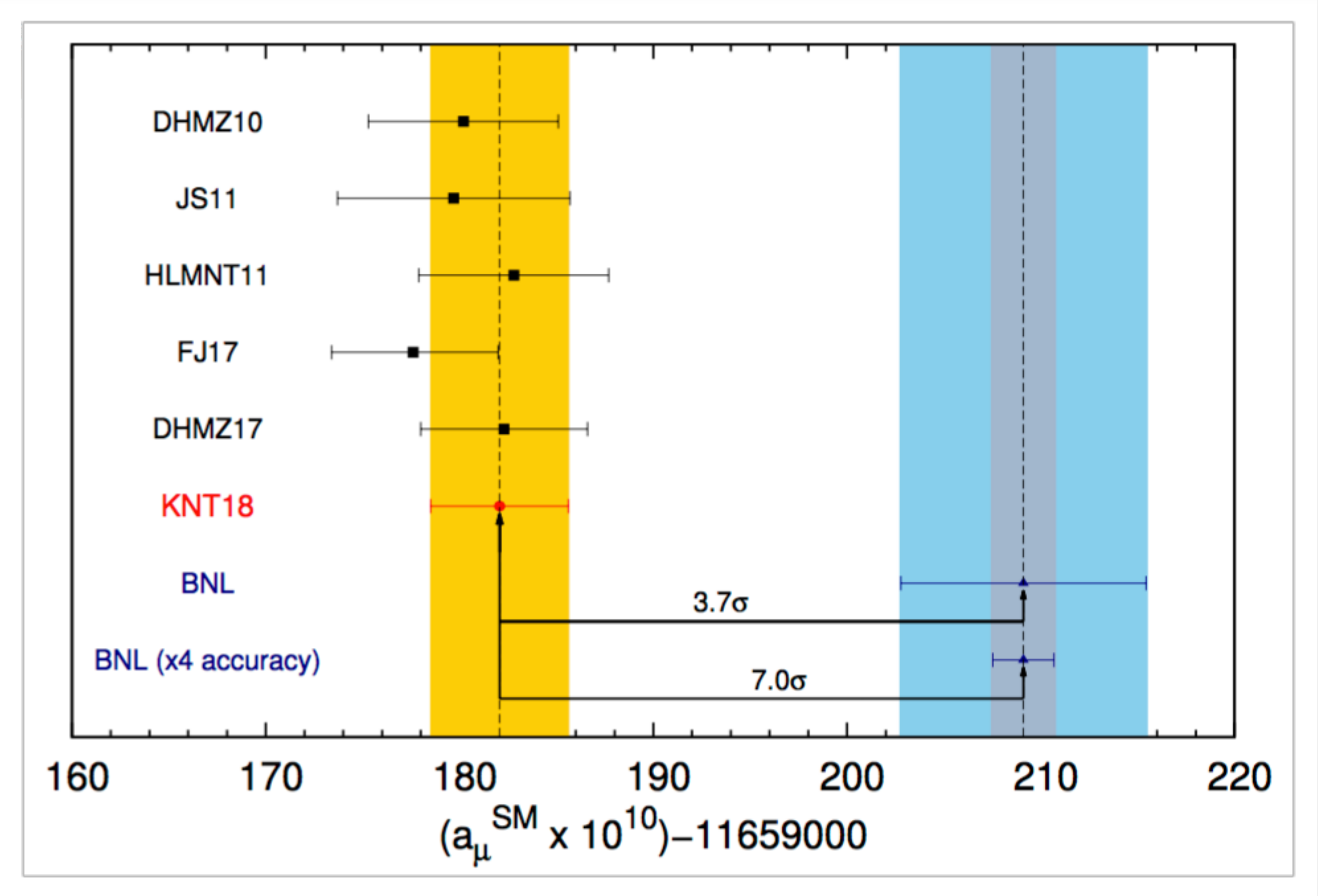
- Fermilab is the DOE national laboratory dedicated to particle physics
- It is located outside of Chicago, IL
- Hosted the Tevatron, which was formerly (2011) the highest energy particle accelerator in the world (1.6 TeV)
- Responsible for discoveries of the Top and Bottom quarks

Fermilab bison heard

Muon g-2 Experiment Overview

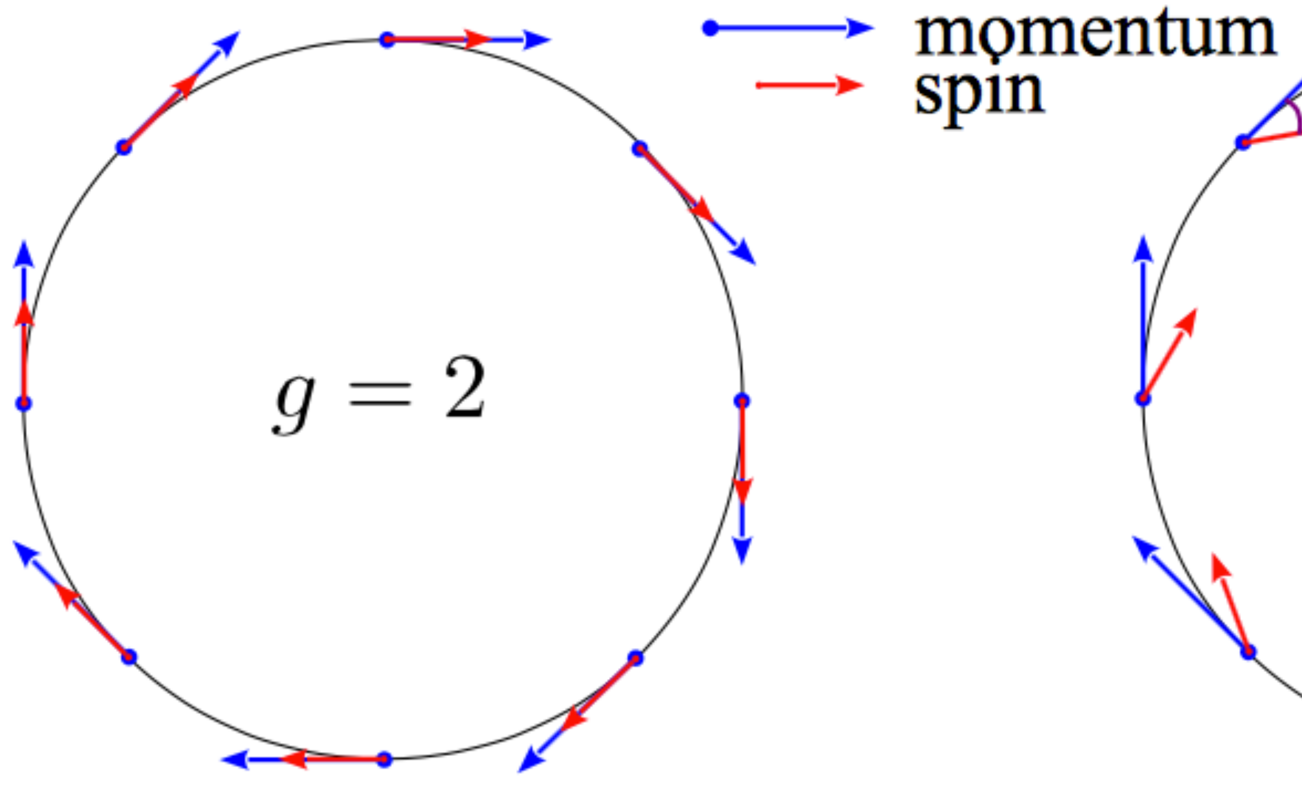
- Goal is to measure the anomalous magnetic moment of the muon to 140 ppb
- Muons injected into 50’ diameter superconducting magnet.
- Muons precess as they travel through the magnetic field.
- Muons decay to positrons, which are detected in calorimeters inside the ring.





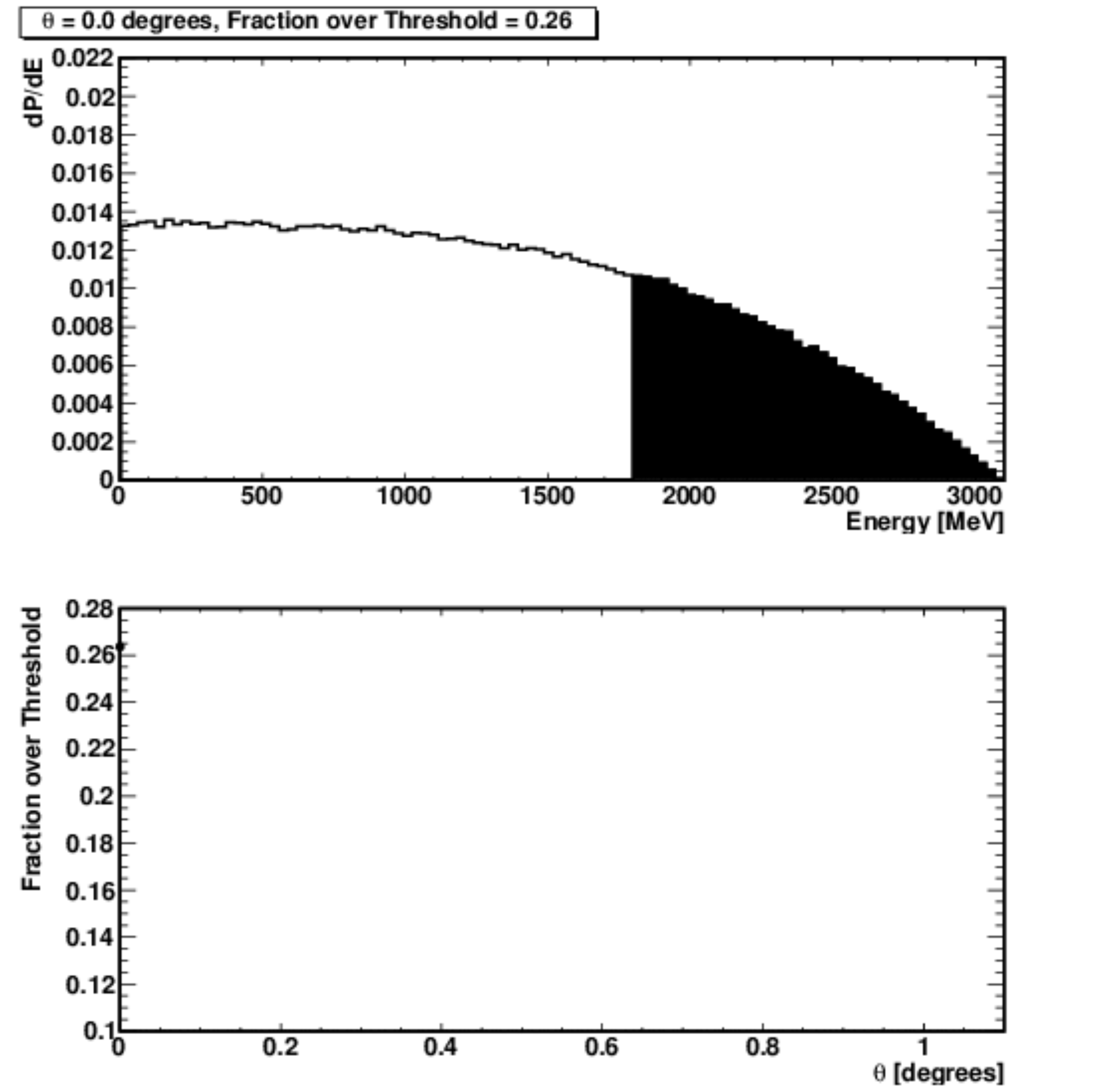
Measurement Overview

- Muons precess as they travel through the magnetic field



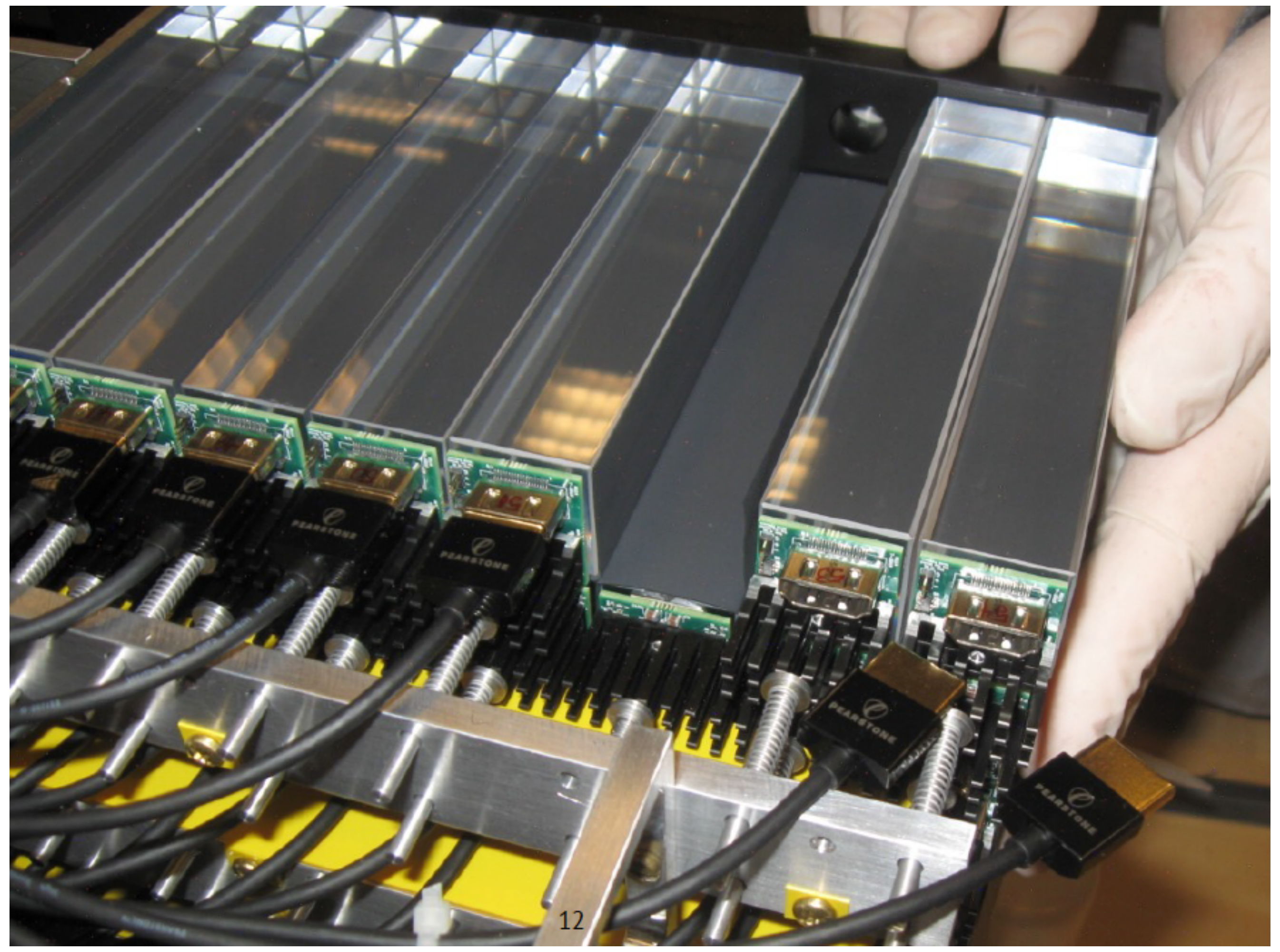
- Must measure both the muon precession and magnetic field to high precision.

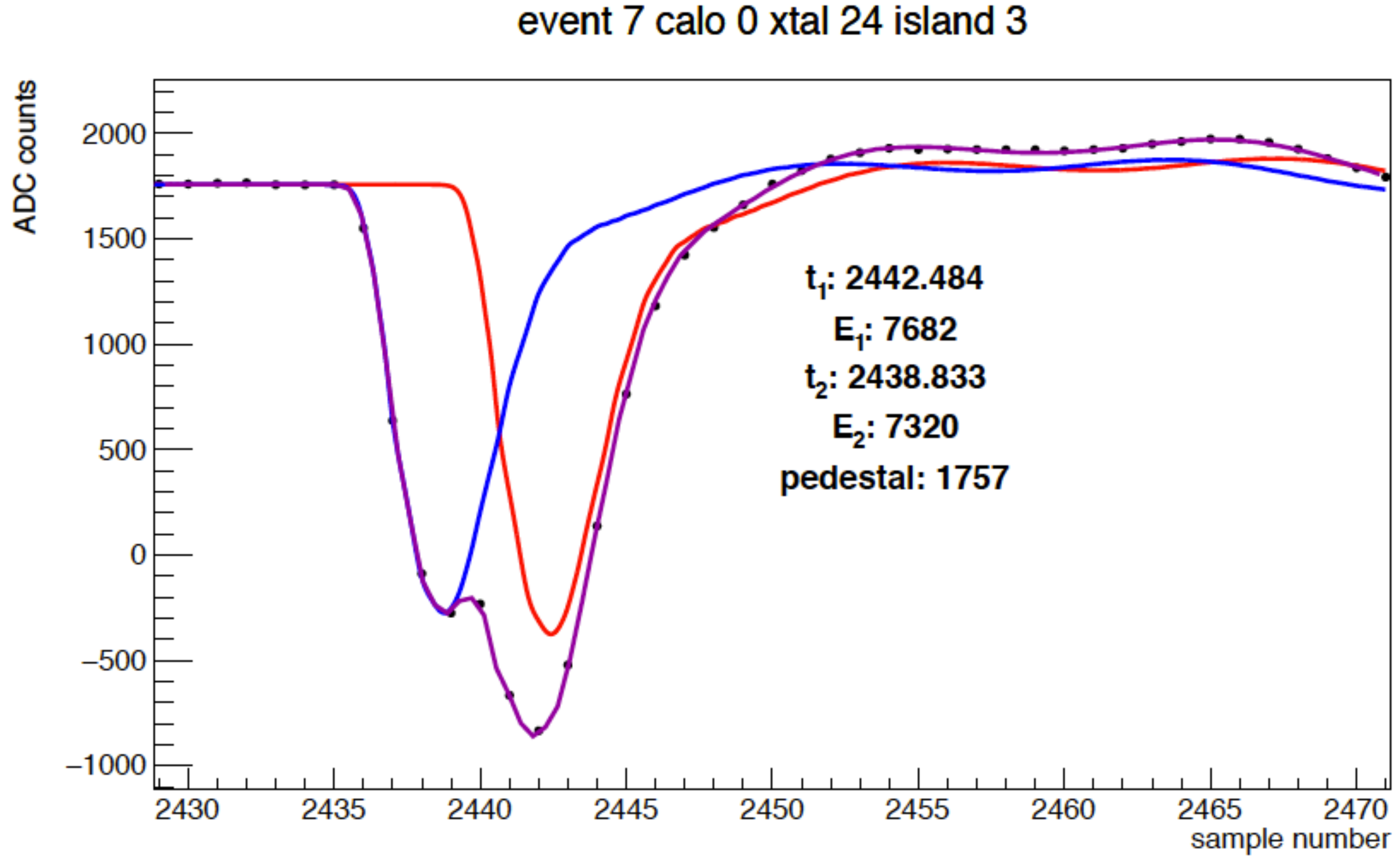
$$a_\mu = \frac{m}{q}\frac{\omega_a}{B}$$

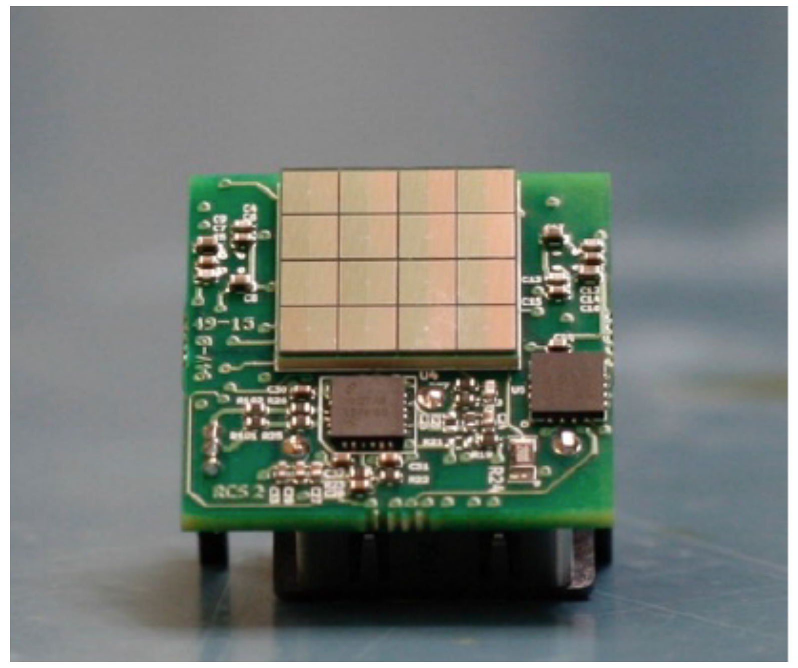


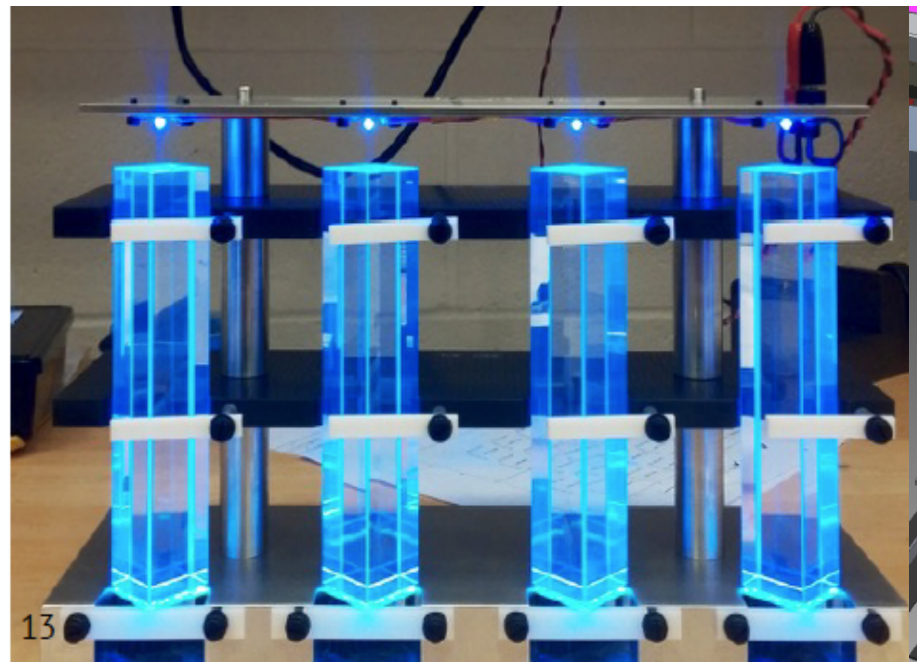
Positron Detectors

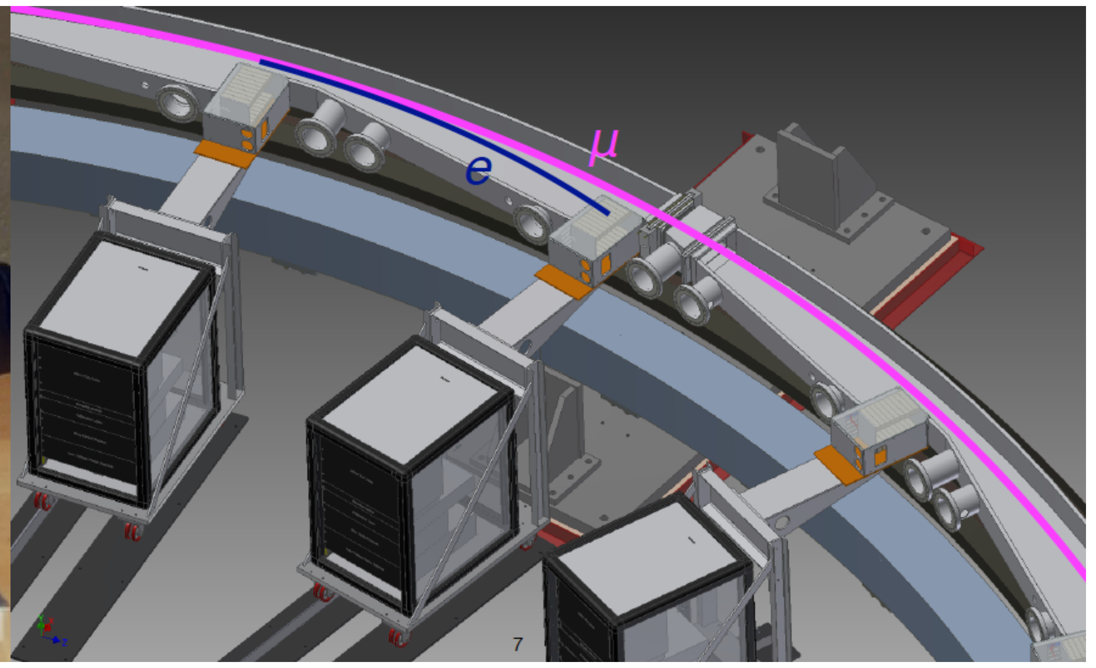
- Interior of ring is lined with 24 calorimeters to detect positrons.
- Each detector is composed of 54 PbF2 crystals with silicon photomultipliers.
- Data recorded with 800 MSPS waveform digitizers in uTCA crates.
- Digitized waveforms transmitted to DAQ via direct 10 Gbps fiber connections.





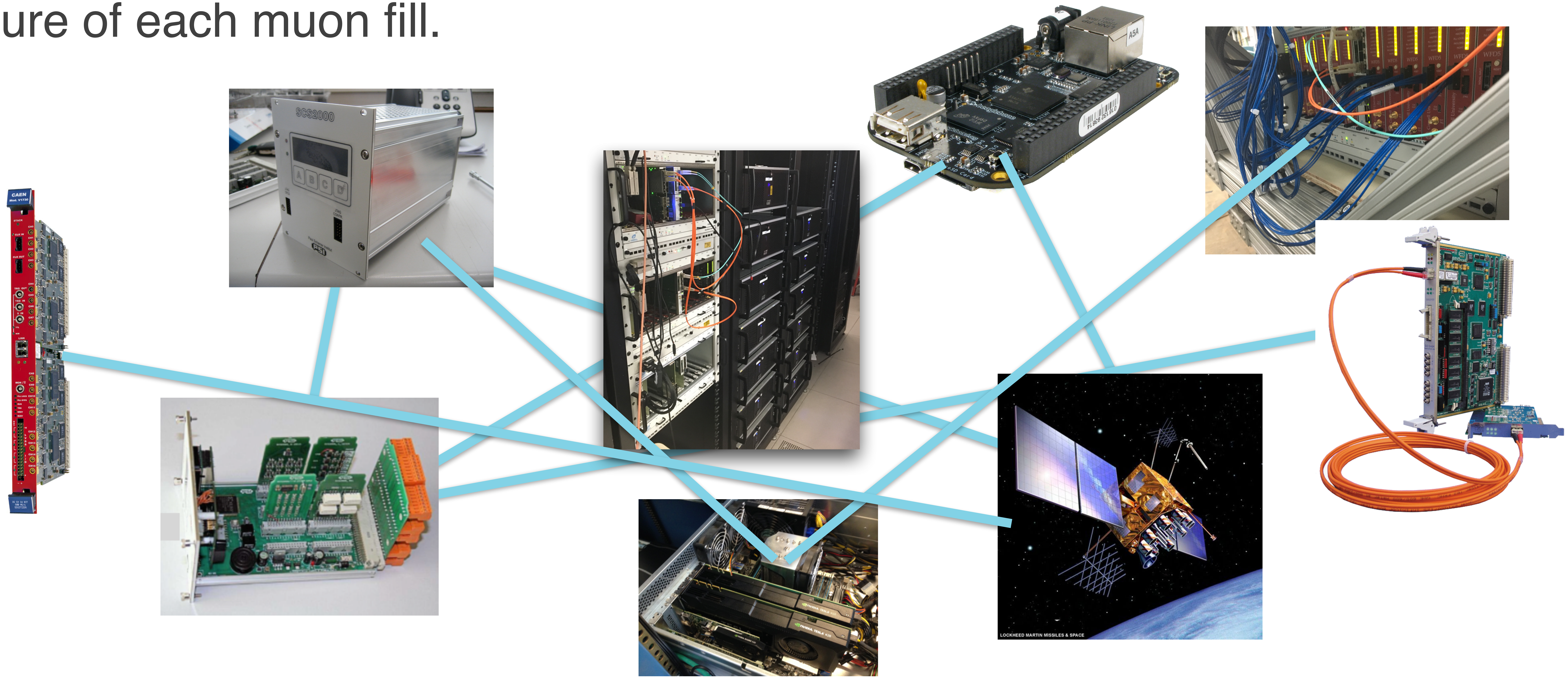






The DAQ as an “Internet of Things”

- The DAQ assembles data from a variety of sources to produce a complete picture of each muon fill.

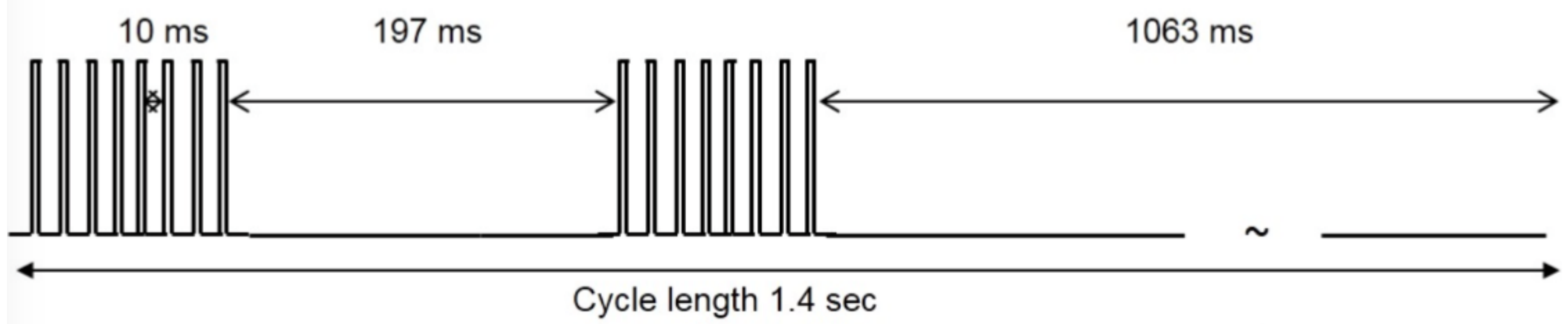


Why GPUs?

- The GPUs dramatically improve performance by parallelizing processing.
- Technology was developed for commercial applications, so it is well supported.
- Easier to code in C++ than to learn specialized language (i.e. FPGA).
- Without GPUs, we could not keep up with our data rates.

Rate requirements

- Accommodate 12 Hz average rate of muon fills that consist of sequences of eight successive 700 μs fills with 10 ms fill-separations.

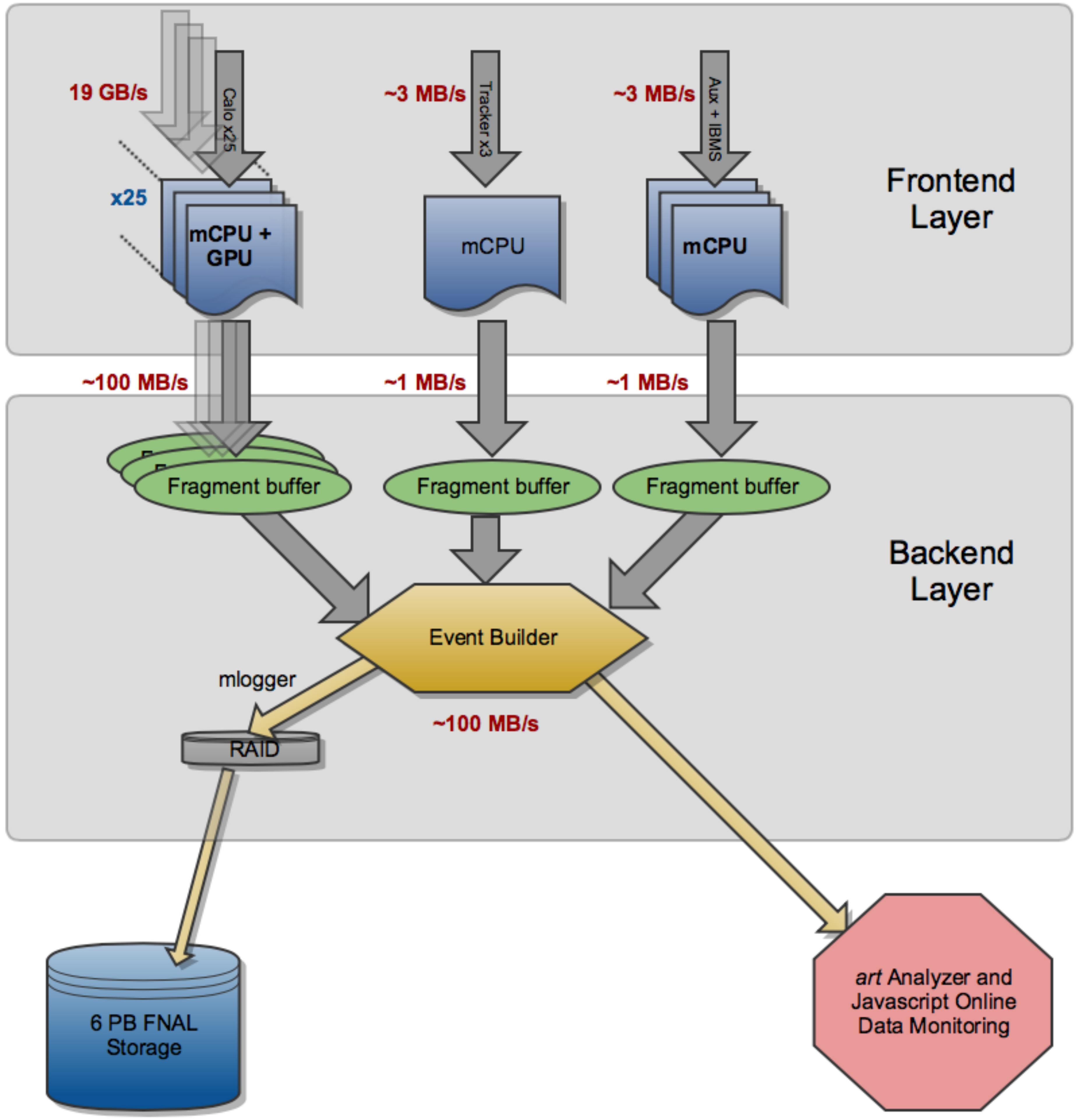


- Time-averaged rate of raw ADC samples is 20 GB/s, which must be reduced by a factor of 100.
- Data is processed in GPUs to accomplish this task.
- Total data on tape after 2 years of running will be 10 PB.

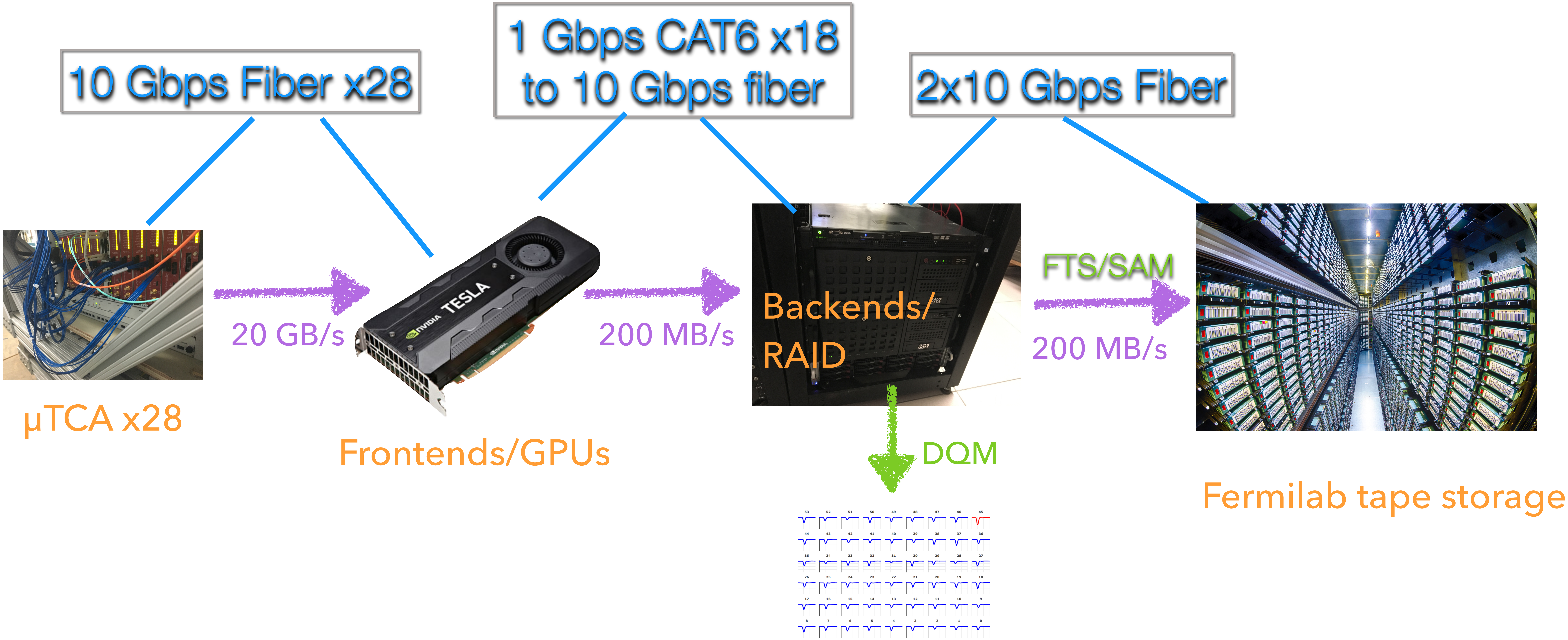
Source	MB Per Second
Raw data	20,000
T-Method	113.1
Q-Method	48.5
Prescaled Raw	20
Tracker	9
Laser Monitor	5
Auxiliary	4
Event Builder:	**200**

DAQ Design

- Layered array of commodity, networked processors
- Frontend layer for readout of detectors.
- Backend layer for assembly of event fragments.
- Slow control layer.
- Online analysis layer using *art*+JS.
- Field DAQ operates independently, but with a similar design.



Data Flow


10 Gbps Fiber x28
1 Gbps CAT6 x18
to 10 Gbps fiber
2x10 Gbps Fiber
20 GB/s
200 MB/s
FTS/SAM
200 MB/s
μTCA x28
Frontends/GPUs
Backends/
RAID
DQM
Fermilab tape storage

DAQ Hardware

- The DAQ hardware includes:
  - 17 frontend machines
  - 5 backend machines
  - 2 dedicated near line analysis machines
  - 3 computers for HV control
  - 3 servers
  - 24 beagle bones running slow controls
- Each frontend contains two Nvidia Tesla K40 GPUs
  - 2880 CUDA cores at 740 MHz
  - 288 GB/s memory bandwidth
  - 12 GB on board memory
  - ECC memory protection
- 70 TB RAID for temporary data storage.

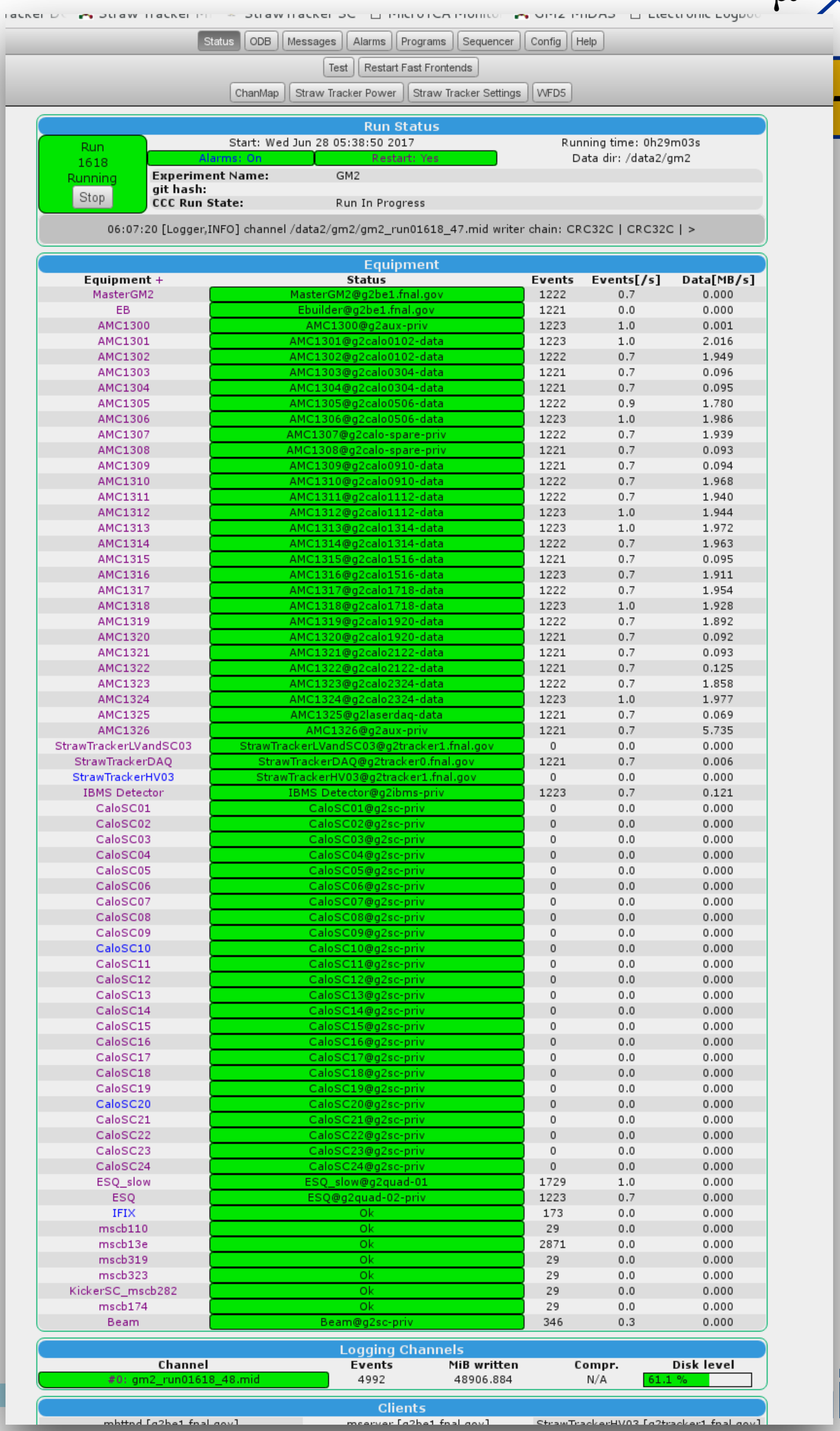
Hardware Considerations

- Using ASUS X99-E WS/USB 3.1 motherboard for frontend systems.
- Motherboard must support PCIe version 3.0 and run at least five slots in parallel.
- Require Xeon processor to support ECC memory.
- Need motherboard with PLX chip, allowing it to distribute the load over 40 lanes in order to operate all PCIe slots at 16x.

PCIe version	GPU	Host to device, Pageable	Host to device, Pinned
2.0	K20	3326.6 MB/s	5028.3 MB/s
3.0	K20	5628.6 MB/s	6003.6 MB/s
3.0	K40	6647.8 MB/s	10044.3 MB/s

MIDAS Software

- MIDAS is a DAQ software package
- Provides a web-interface for control of the experiment, data logging, and event building.
- We write frontend code in C++ and CUDA that processes the data and sends it to the event builder.
- 32 fast frontends reduce data volume for each event by a factor of 100 and store data in Midas banks.
- 35 slow control frontends process slow data.
- Includes alarm system and sequencer.
- Software configuration is dumped to a JSON file and saved to a PostgreSQL database at the end of each run.



Status | ODB | Messages | Alarms | Programs | Sequencer | Config | Help

Test | Restart Fast Frontends

ChanMap | Straw Tracker Power | Straw Tracker Settings | WFD5

Run Status

Run 1618 Running — Stop

Start: Wed Jun 28 05:38:50 2017 — Running time: 0h29m03s

Alarms: On — Restart: Yes — Data dir: /data2/gm2

Experiment Name: GM2

git hash:

CCC Run State: Run In Progress

06:07:20 [Logger,INFO] channel /data2/gm2/gm2_run01618_47.mid writer chain: CRC32C | CRC32C | >

Equipment

Equipment +	Status	Events	Events[/s]	Data[MB/s]
MasterGM2	MasterGM2@g2be1.fnal.gov	1222	0.7	0.000
EB	Ebuilder@g2be1.fnal.gov	1221	0.0	0.000
AMC1300	AMC1300@g2aux-priv	1223	1.0	0.001
AMC1301	AMC1301@g2calo0102-data	1223	1.0	2.016
AMC1302	AMC1302@g2calo0102-data	1222	0.7	1.949
AMC1303	AMC1303@g2calo0304-data	1221	0.7	0.096
AMC1304	AMC1304@g2calo0304-data	1221	0.7	0.095
AMC1305	AMC1305@g2calo0506-data	1222	0.9	1.780
AMC1306	AMC1306@g2calo0506-data	1223	1.0	1.986
AMC1307	AMC1307@g2calo-spare-priv	1222	0.7	1.939
AMC1308	AMC1308@g2calo-spare-priv	1221	0.7	0.093
AMC1309	AMC1309@g2calo0910-data	1221	0.7	0.094
AMC1310	AMC1310@g2calo0910-data	1222	0.7	1.968
AMC1311	AMC1311@g2calo1112-data	1222	0.7	1.940
AMC1312	AMC1312@g2calo1112-data	1223	1.0	1.944
AMC1313	AMC1313@g2calo1314-data	1223	1.0	1.972
AMC1314	AMC1314@g2calo1314-data	1222	0.7	1.963
AMC1315	AMC1315@g2calo1516-data	1221	0.7	0.095
AMC1316	AMC1316@g2calo1516-data	1223	0.7	1.911
AMC1317	AMC1317@g2calo1718-data	1222	0.7	1.954
AMC1318	AMC1318@g2calo1718-data	1223	1.0	1.928
AMC1319	AMC1319@g2calo1920-data	1222	0.7	1.892
AMC1320	AMC1320@g2calo1920-data	1221	0.7	0.092
AMC1321	AMC1321@g2calo2122-data	1221	0.7	0.093
AMC1322	AMC1322@g2calo2122-data	1221	0.7	0.125
AMC1323	AMC1323@g2calo2324-data	1222	0.7	1.858
AMC1324	AMC1324@g2calo2324-data	1223	1.0	1.977
AMC1325	AMC1325@g2laserdaq-data	1221	0.7	0.069
AMC1326	AMC1326@g2aux-priv	1221	0.7	5.735
StrawTrackerLVandSC03	StrawTrackerLVandSC03@g2tracker1.fnal.gov	0	0.0	0.000
StrawTrackerDAQ	StrawTrackerDAQ@g2tracker0.fnal.gov	1221	0.7	0.006
StrawTrackerHV03	StrawTrackerHV03@g2tracker1.fnal.gov	0	0.0	0.000
IBMS Detector	IBMS Detector@g2ibms-priv	1223	0.7	0.121
CaloSC01	CaloSC01@g2sc-priv	0	0.0	0.000
CaloSC02	CaloSC02@g2sc-priv	0	0.0	0.000
CaloSC03	CaloSC03@g2sc-priv	0	0.0	0.000
CaloSC04	CaloSC04@g2sc-priv	0	0.0	0.000
CaloSC05	CaloSC05@g2sc-priv	0	0.0	0.000
CaloSC06	CaloSC06@g2sc-priv	0	0.0	0.000
CaloSC07	CaloSC07@g2sc-priv	0	0.0	0.000
CaloSC08	CaloSC08@g2sc-priv	0	0.0	0.000
CaloSC09	CaloSC09@g2sc-priv	0	0.0	0.000
CaloSC10	CaloSC10@g2sc-priv	0	0.0	0.000
CaloSC11	CaloSC11@g2sc-priv	0	0.0	0.000
CaloSC12	CaloSC12@g2sc-priv	0	0.0	0.000
CaloSC13	CaloSC13@g2sc-priv	0	0.0	0.000
CaloSC14	CaloSC14@g2sc-priv	0	0.0	0.000
CaloSC15	CaloSC15@g2sc-priv	0	0.0	0.000
CaloSC16	CaloSC16@g2sc-priv	0	0.0	0.000
CaloSC17	CaloSC17@g2sc-priv	0	0.0	0.000
CaloSC18	CaloSC18@g2sc-priv	0	0.0	0.000
CaloSC19	CaloSC19@g2sc-priv	0	0.0	0.000
CaloSC20	CaloSC20@g2sc-priv	0	0.0	0.000
CaloSC21	CaloSC21@g2sc-priv	0	0.0	0.000
CaloSC22	CaloSC22@g2sc-priv	0	0.0	0.000
CaloSC23	CaloSC23@g2sc-priv	0	0.0	0.000
CaloSC24	CaloSC24@g2sc-priv	0	0.0	0.000
ESQ_slow	ESQ_slow@g2quad-01	1729	1.0	0.000
ESQ	ESQ@g2quad-02-priv	1223	0.7	0.000
IFIX	Ok	173	0.0	0.000
mscb110	Ok	29	0.0	0.000
mscb13e	Ok	2871	0.0	0.000
mscb319	Ok	29	0.0	0.000
mscb323	Ok	29	0.0	0.000
KickerSC_mscb282	Ok	29	0.0	0.000
mscb174	Ok	29	0.0	0.000
Beam	Beam@g2sc-priv	346	0.3	0.000

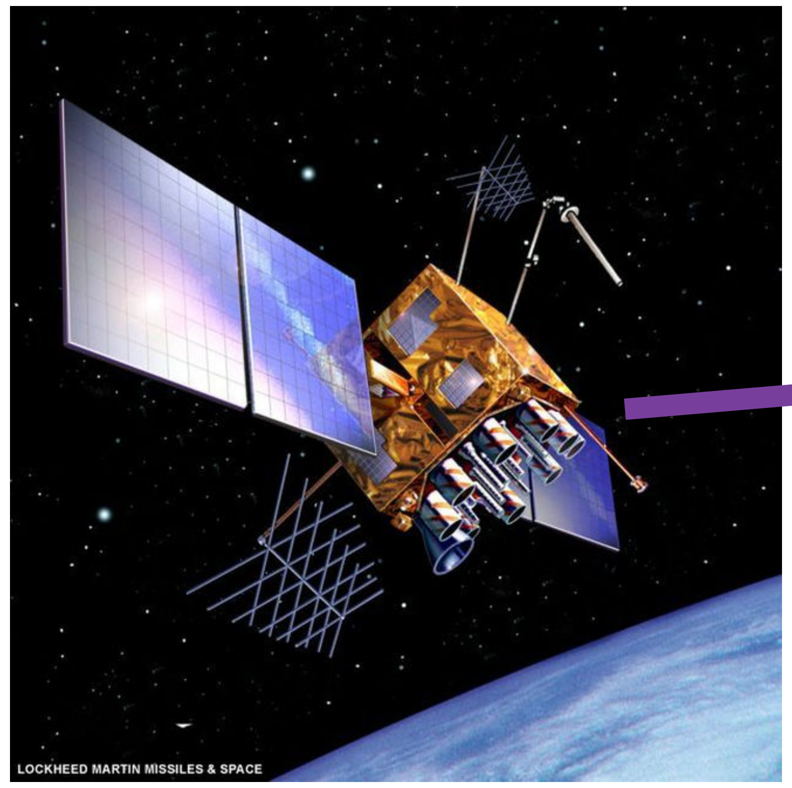
Logging Channels

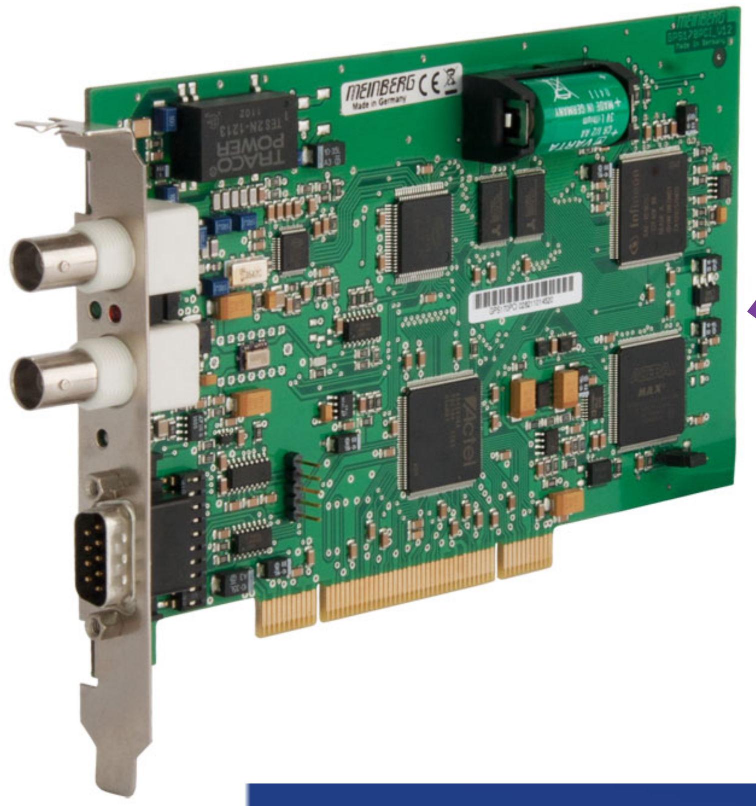
Channel	Events	MiB written	Compr.	Disk level
#0: gm2_run01618_48.mid	4992	48906.884	N/A	61.1 %

Clients

Master frontend

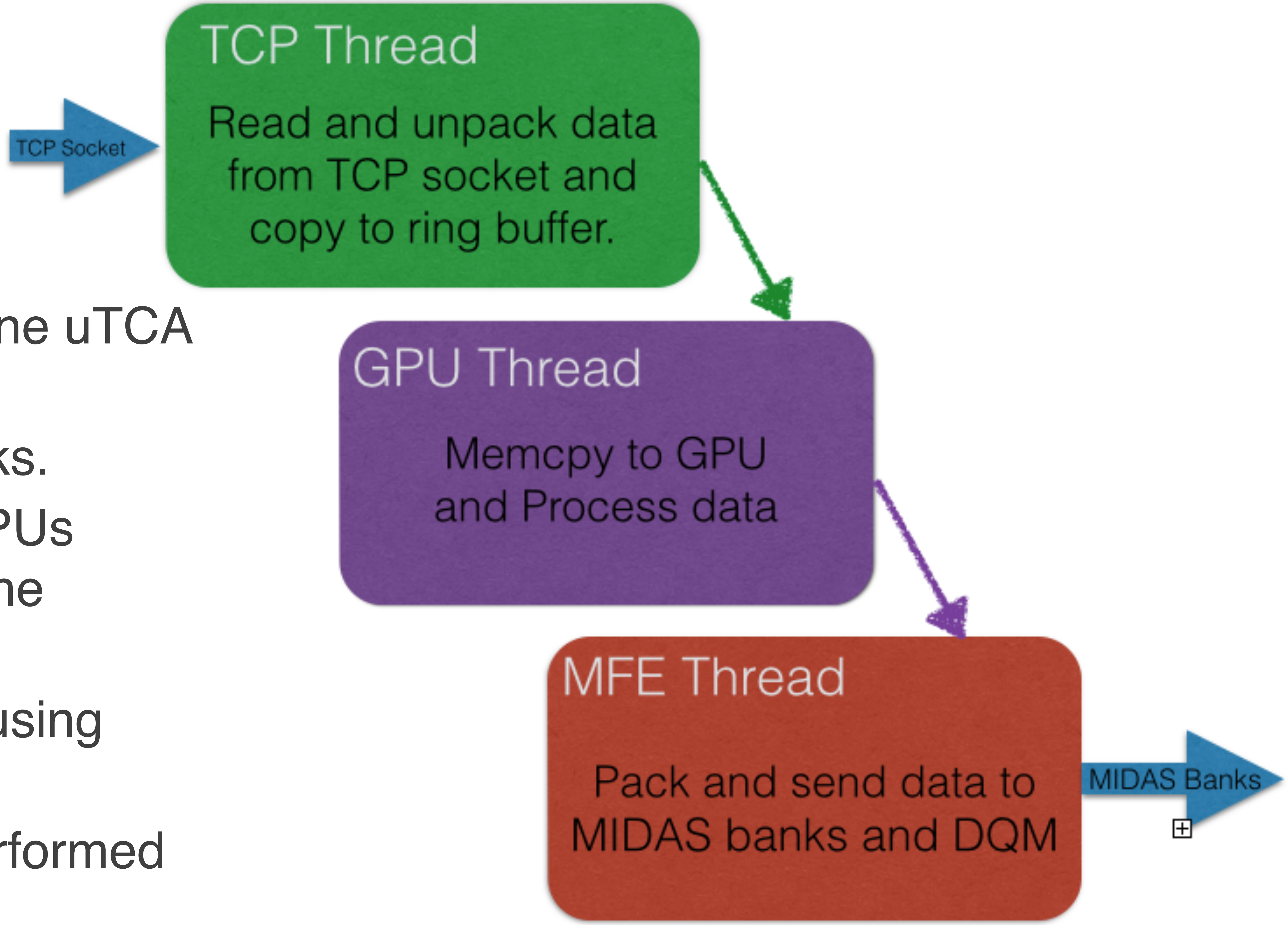
- Communicates with other frontends using RPC calls.
- Provides begin of run and end of run RPCs to all frontends.
- Provides end of fill trigger to synchronous frontends.
- Configures clock and control system.
- Reads trigger times from Meinberg GPS unit and writes them to a MIDAS bank.





```
Bank:GPS0 Length: 20(I*1)/5(I*4)/5(Type) Type:Unsigned Integer*4
   1-> 0x000018c2 0x580838bc 0x5dcabfb0 0x580838bc 0x5dd48308
```

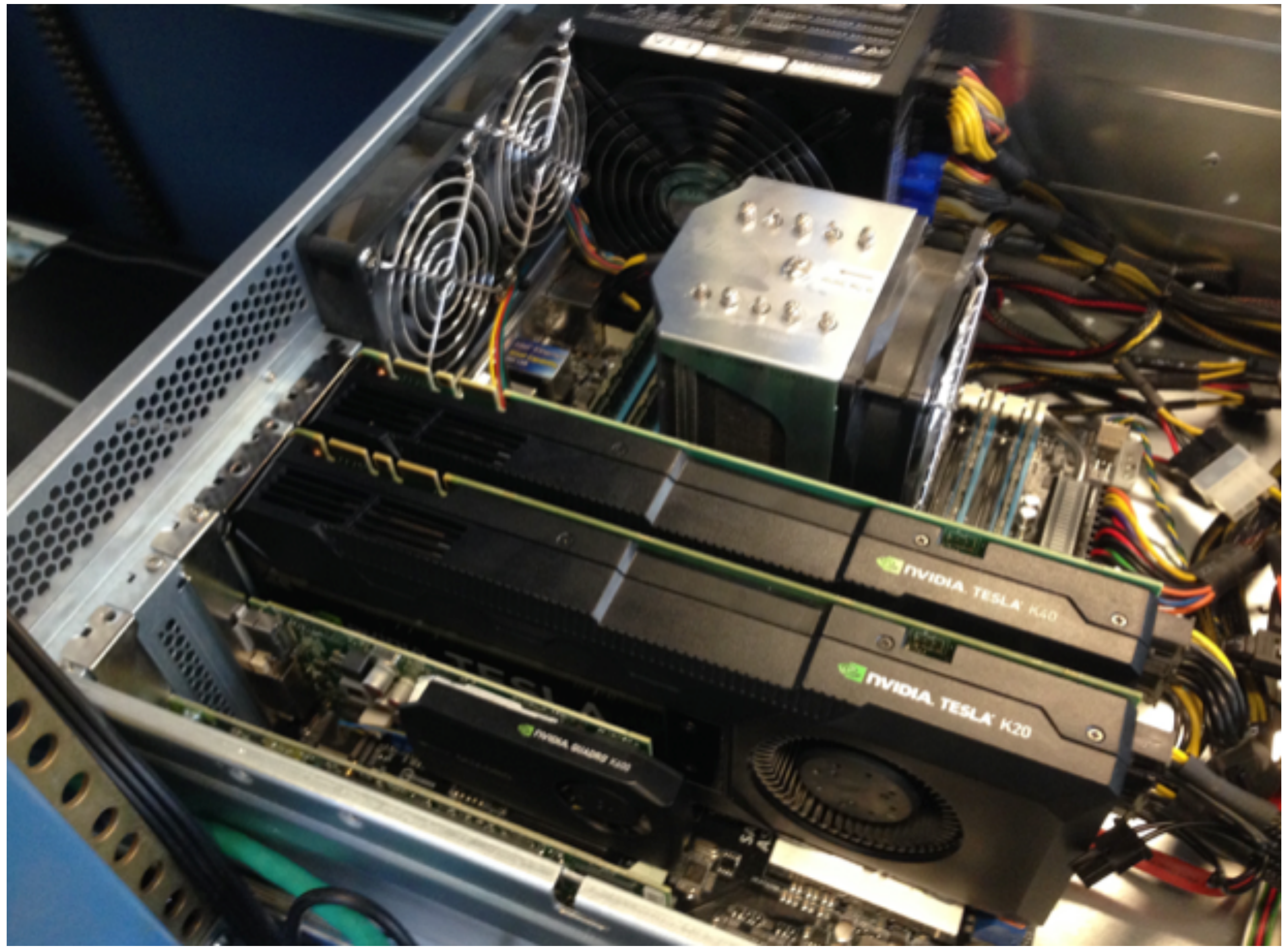
GPU Frontend



- Each frontend process reads data from one uTCA crate over 10 Gb ethernet with TCPIP.
- Frontend is multithreaded with mutex locks.
- Data is processed in Nvidia Tesla K40 GPUs using CUDA code that is integrated into the frontend.
- Midas banks are losslessly compressed using zlib.
- Full configuration of the uTCA crate is performed via the MIDAS ODB using IPBus.

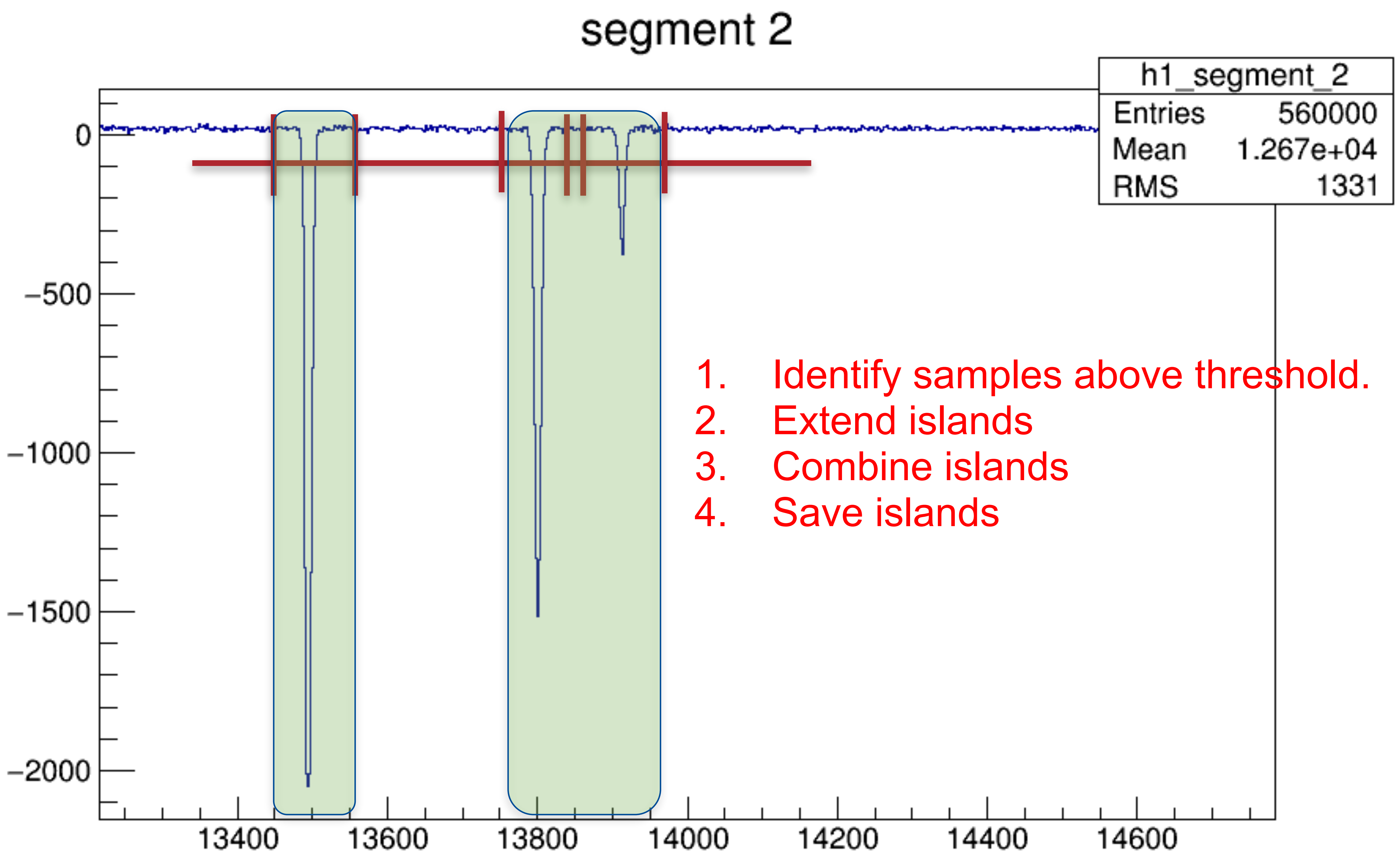
GPU Processing

- The frontend includes CUDA routines for data processing.
- Each GPU processes data from one calorimeter.
- Raw fill is copied to GPU memory, where it is reduced using T-method (island chopping), Q-method (histogramming), pedestal calculation, and template fitting.
- The output of each process is written in one MIDAS bank.



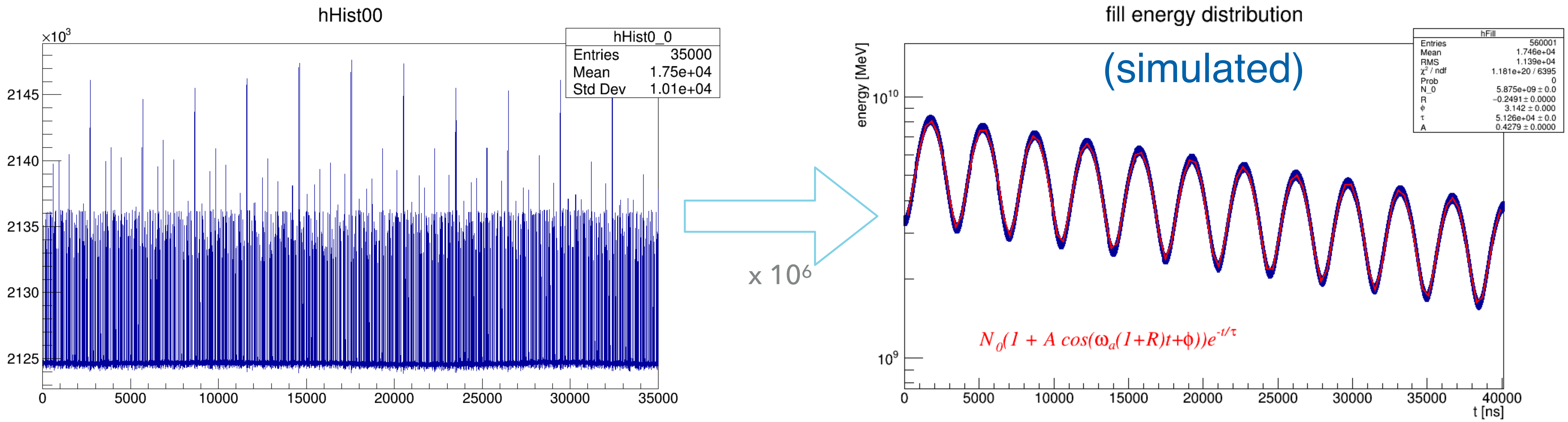
T-Method

- Identify and save regions of the waveform containing positron hits.
- A typical waveform will have ~180 islands.



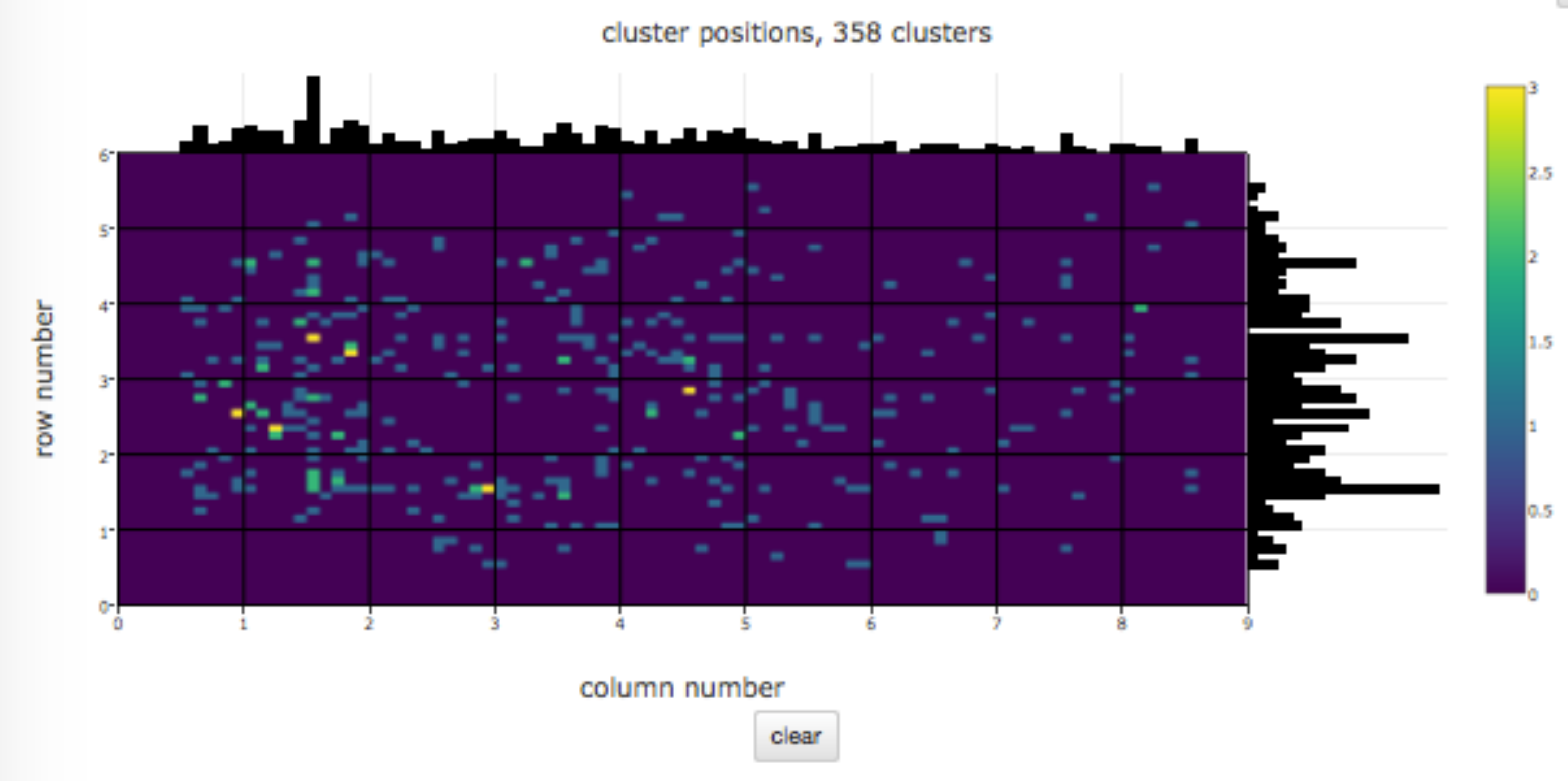
Q-method

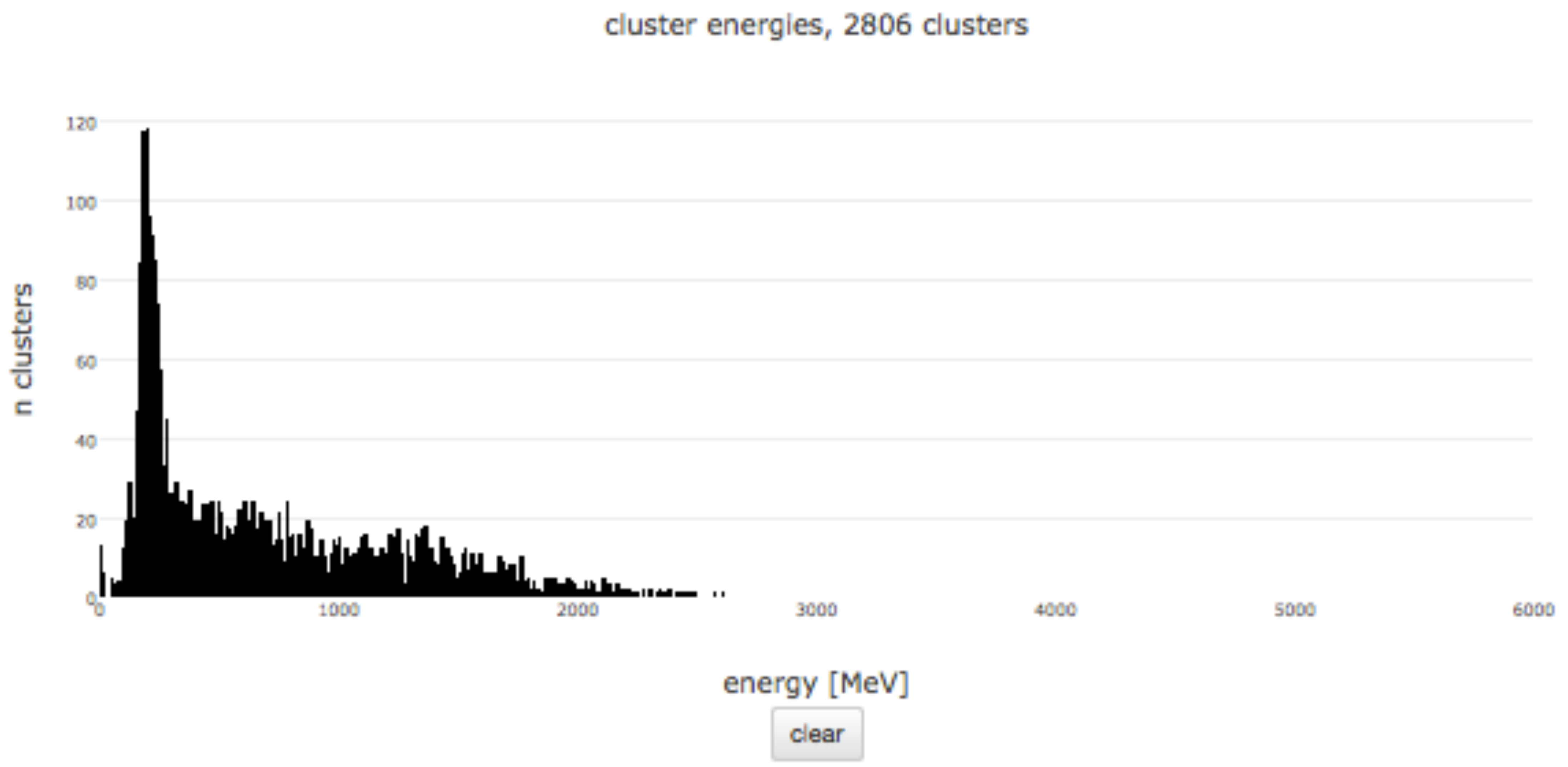
- Full waveforms are decimated in time and summed over many fills to create a histogram that is saved in the data file.
  - i.e. If we decimate in time by 10 and flush every 100 fills, we reduce the data rate by a factor of 1000, so from 20 GB/s to 20 MB/s.
- Use smaller bins at lower times and wider bins at later times to insure that we can extract the pedestal.



GPU Template Fits

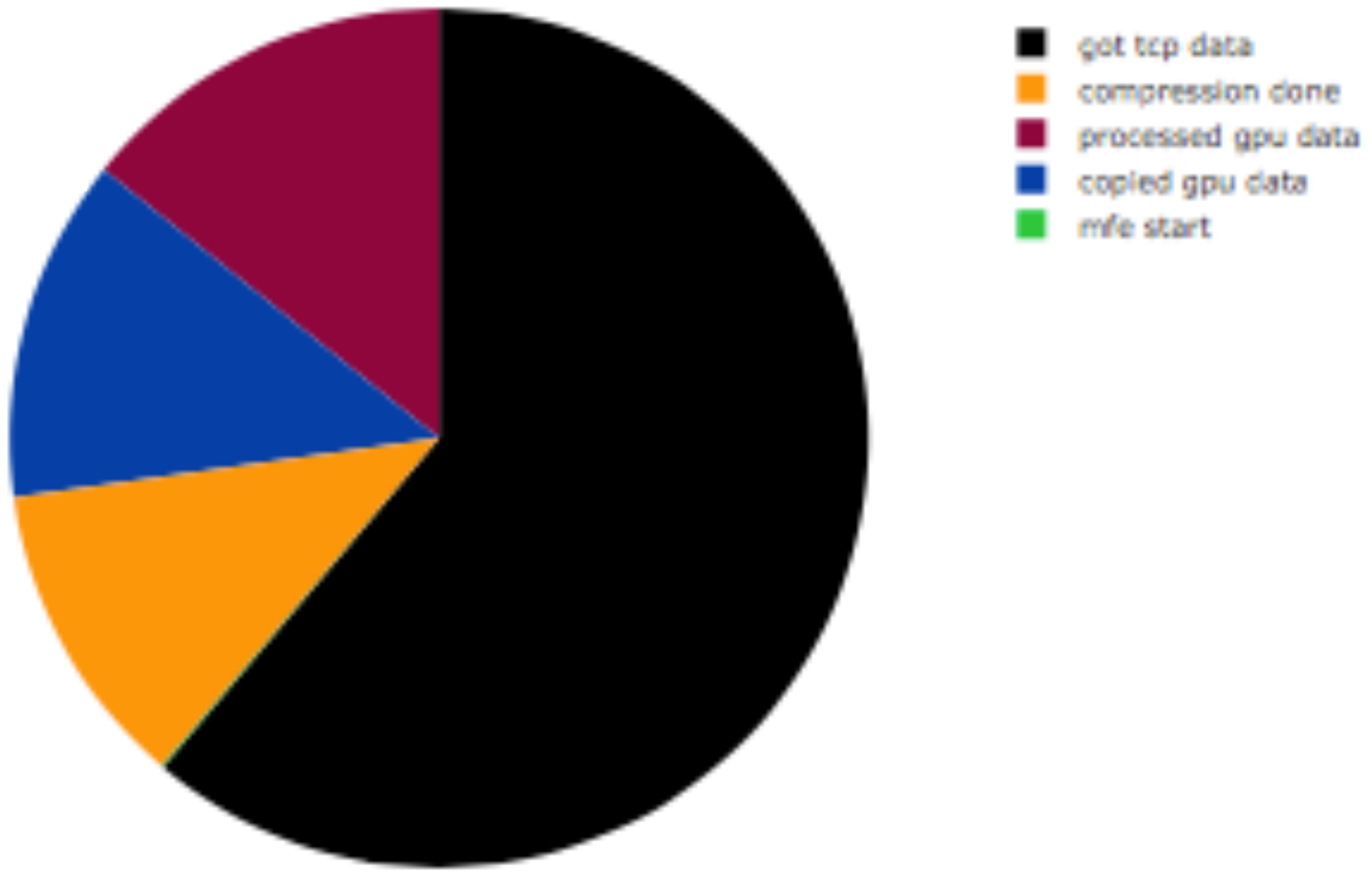
- Fit templates are loaded into the GPU memory and used to fit each peak above a certain threshold.
- A bank containing the fit results will be saved for each fill in addition to the chopped islands.
- Reduces the processing necessary in the online DQM.

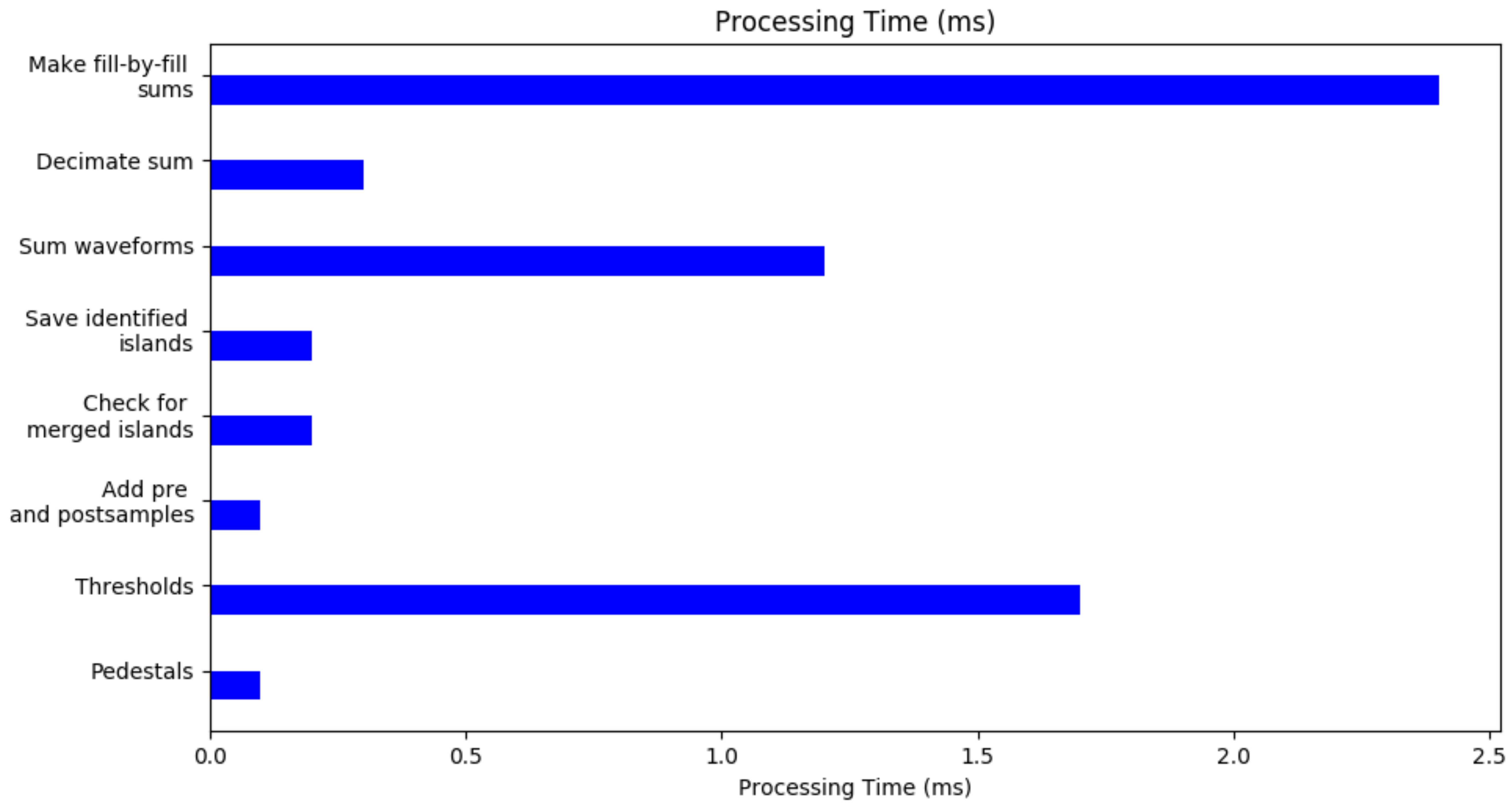




Processing time

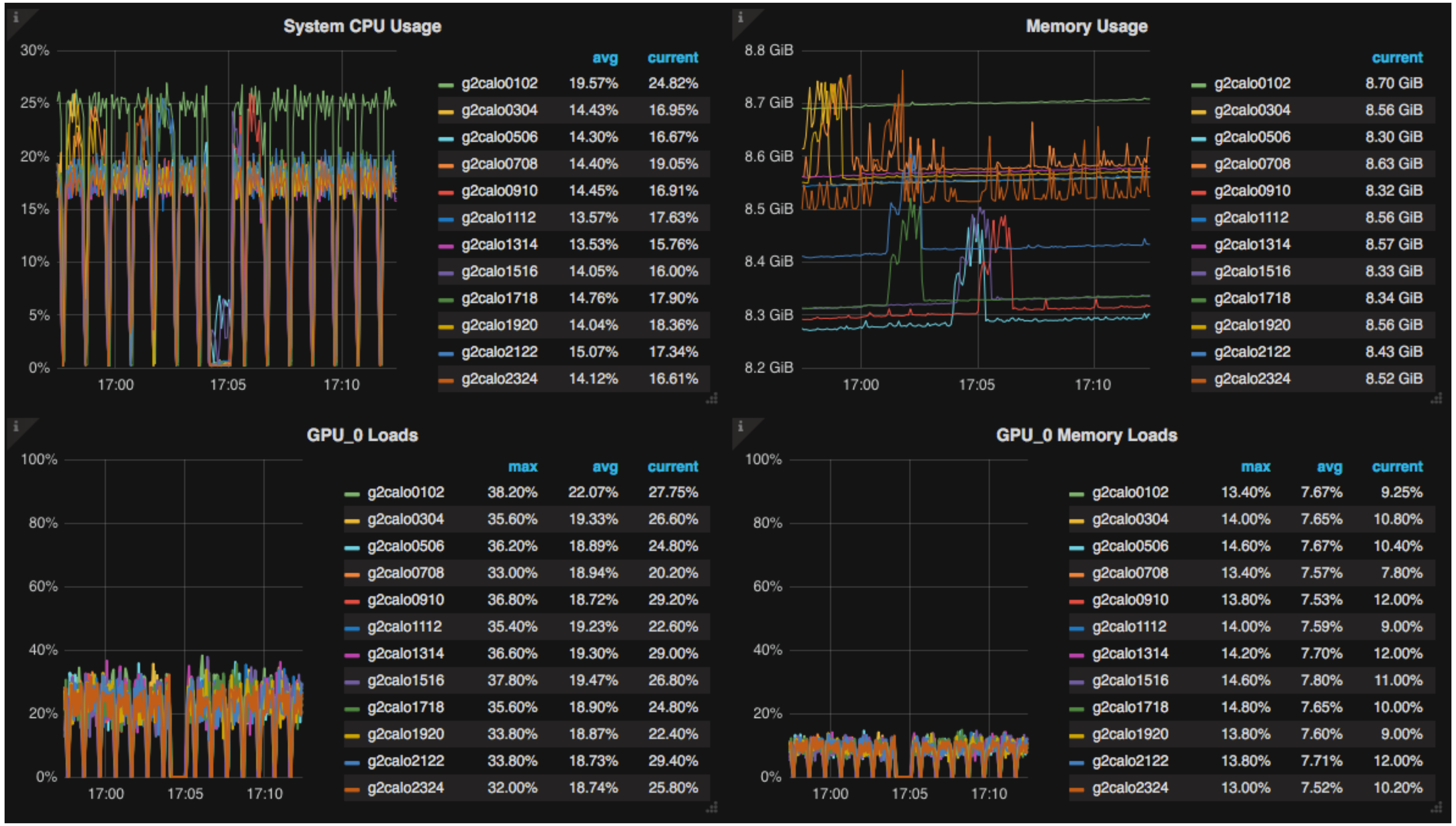
- Must process each event in 83 ms to keep up with average beam rate of 12 Hz.
- Most time is spent reading data from TCP socket and copying it to the GPU.
- Processing time in the GPU is very small.





DAQ Health Monitor

- Monitored health of DAQ systems using:
  - netdata for system monitoring
  - prometheus for short term data storage
  - grafana to display data.
- Monitors GPU temperatures, loads, and memory
- Alarms via Slack if any quantity is out of bounds



Data Quality Monitor

MIDAS

midas-to-art
unpackers
producers
analyzers

node JS

Plotly

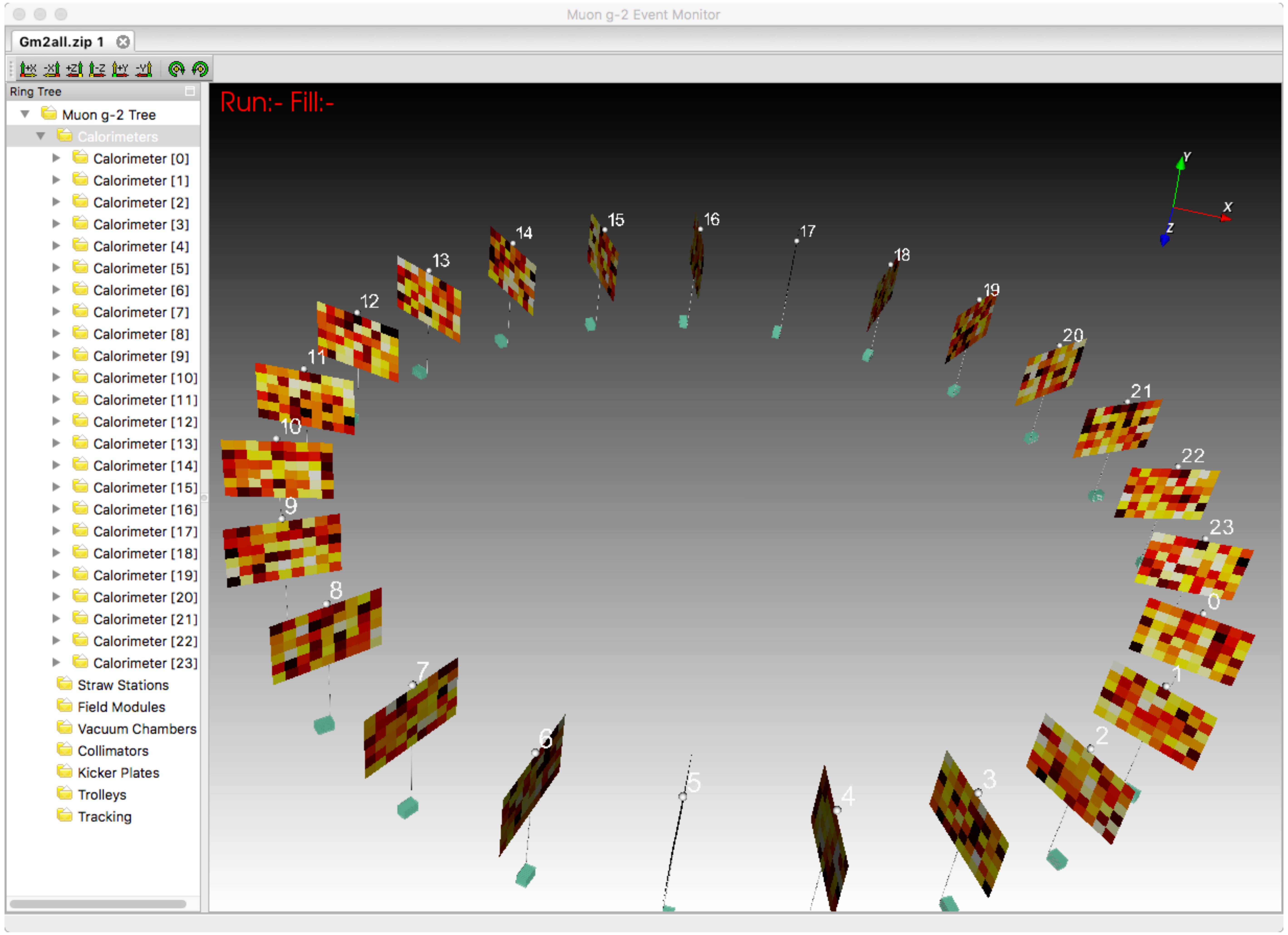
Plotly

Plotly

Fermilab

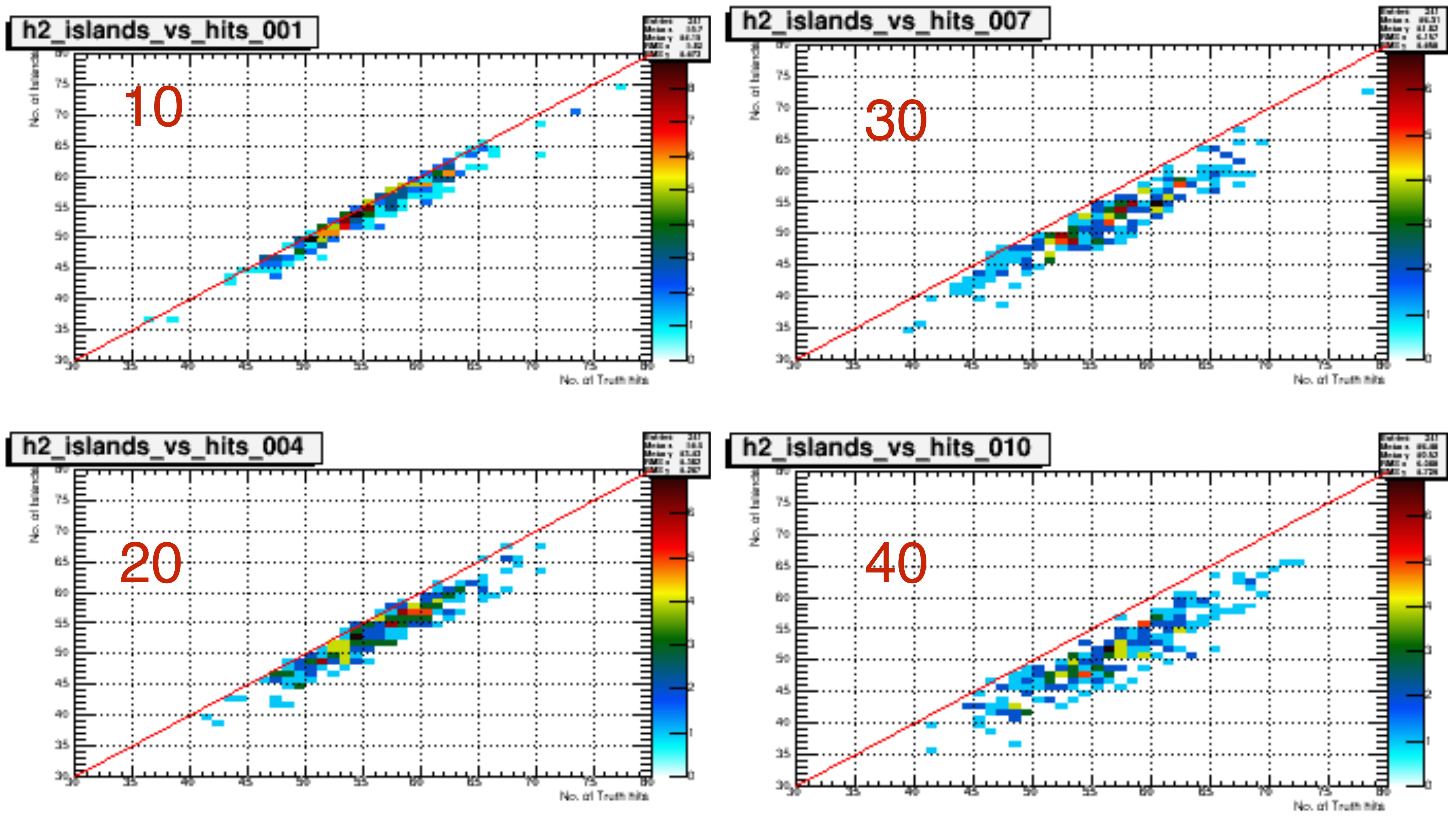
Event Display

- Based on Paraview (vtk).
- Reads data from GPU template fits to create heat map on each calorimeter.
- Display can rotate, zoom
- Access to data quality plots
- Currently includes only calorimeters, but plan to add more subsystems soon.



Testing With Simulation

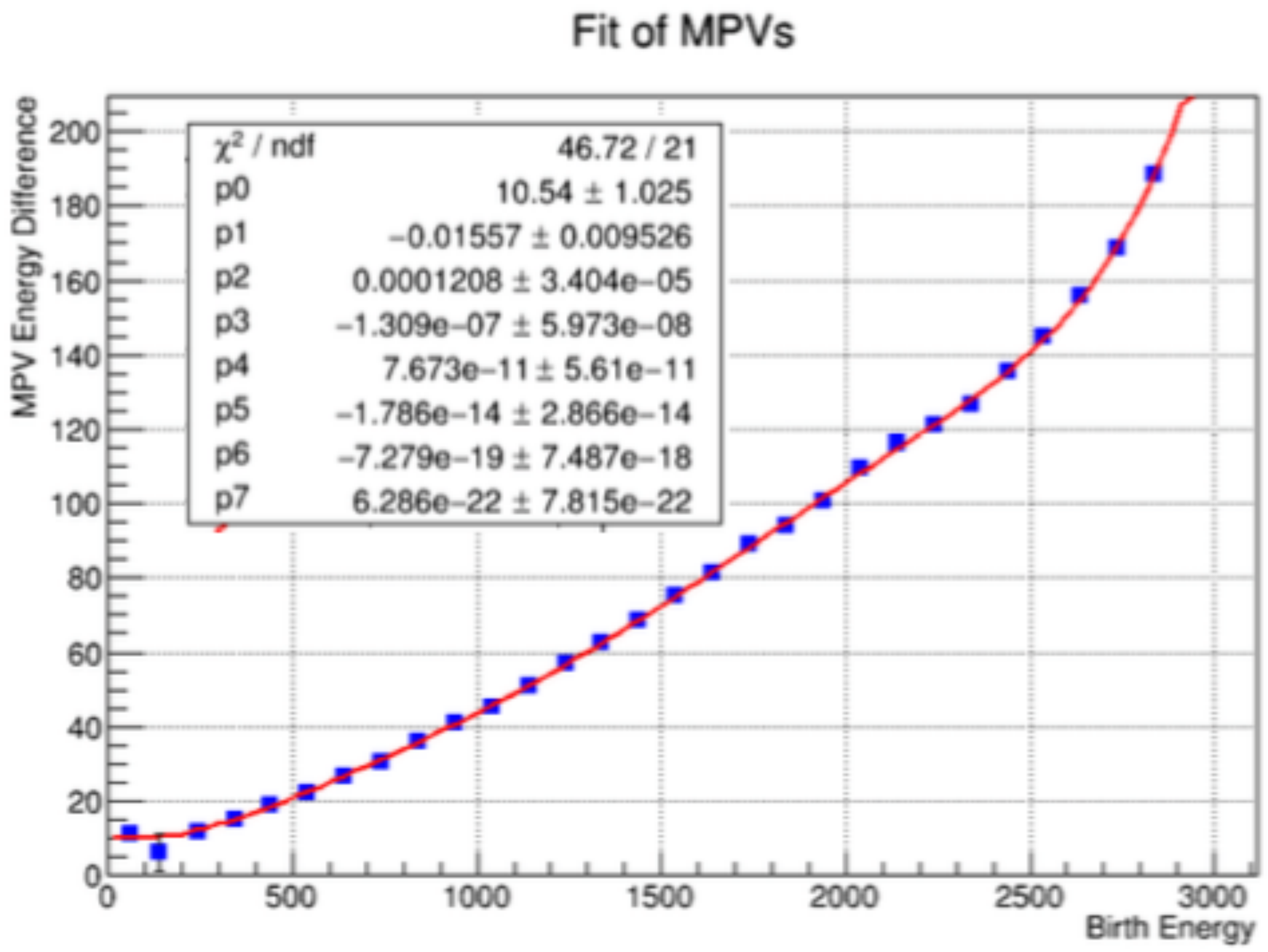
- Two types of simulation used to test DAQ:
  - Online simulation that runs in MIDAS
  - Offline CUDA-based simulation that runs on OSG
- Online Simulation:
  - Fills generated and passed through tcp socket at localhost to AMC13 frontend
  - Truth data saved in separate data bank



- Detected hits vs truth for four threshold values
- Red line shows slope of 1

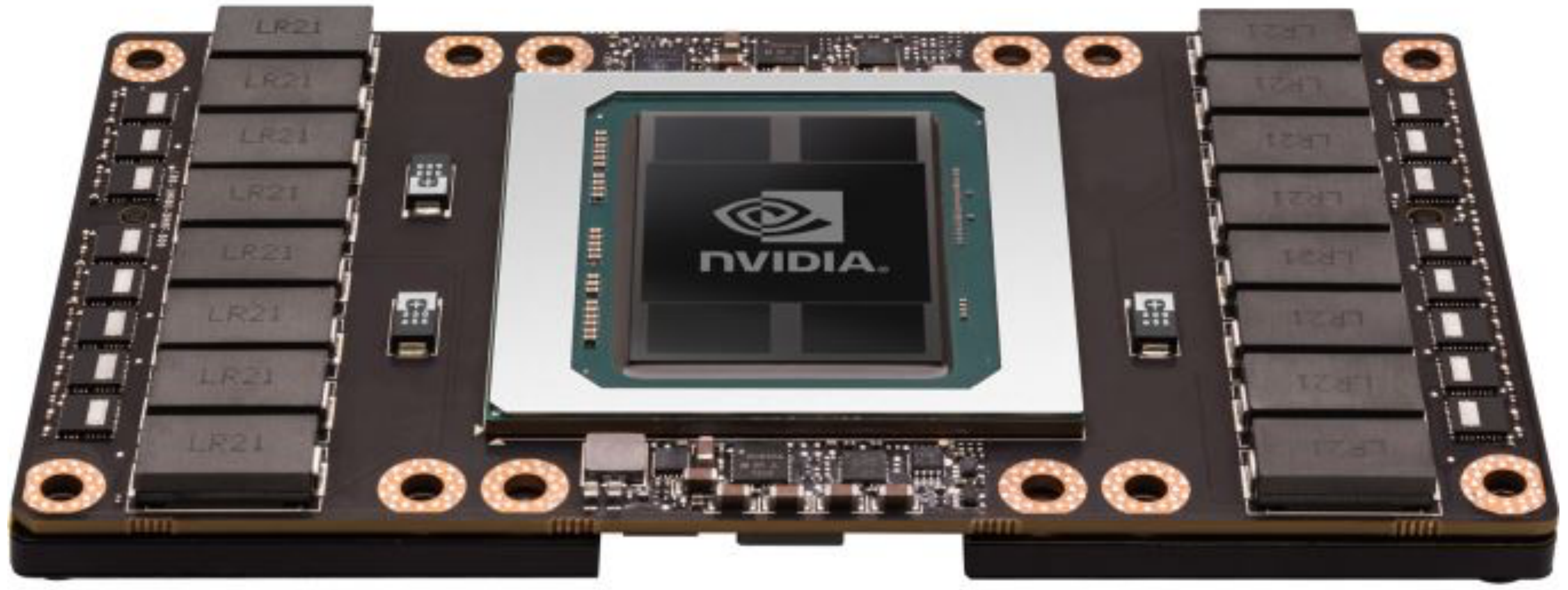
CUDA Based Monte-Carlo

- Cuda-Based Monte Carlo simulation used to test analysis and determine optimum parameters at which to run the data acquisition system.
- Utilizes cuRand to generate Monte Carlo dataset.
- Each fill is generated in a single thread, so many fills are generated in parallel.
- Utilize input from Geant simulation to implement realistic energy, time, and spatial distributions.



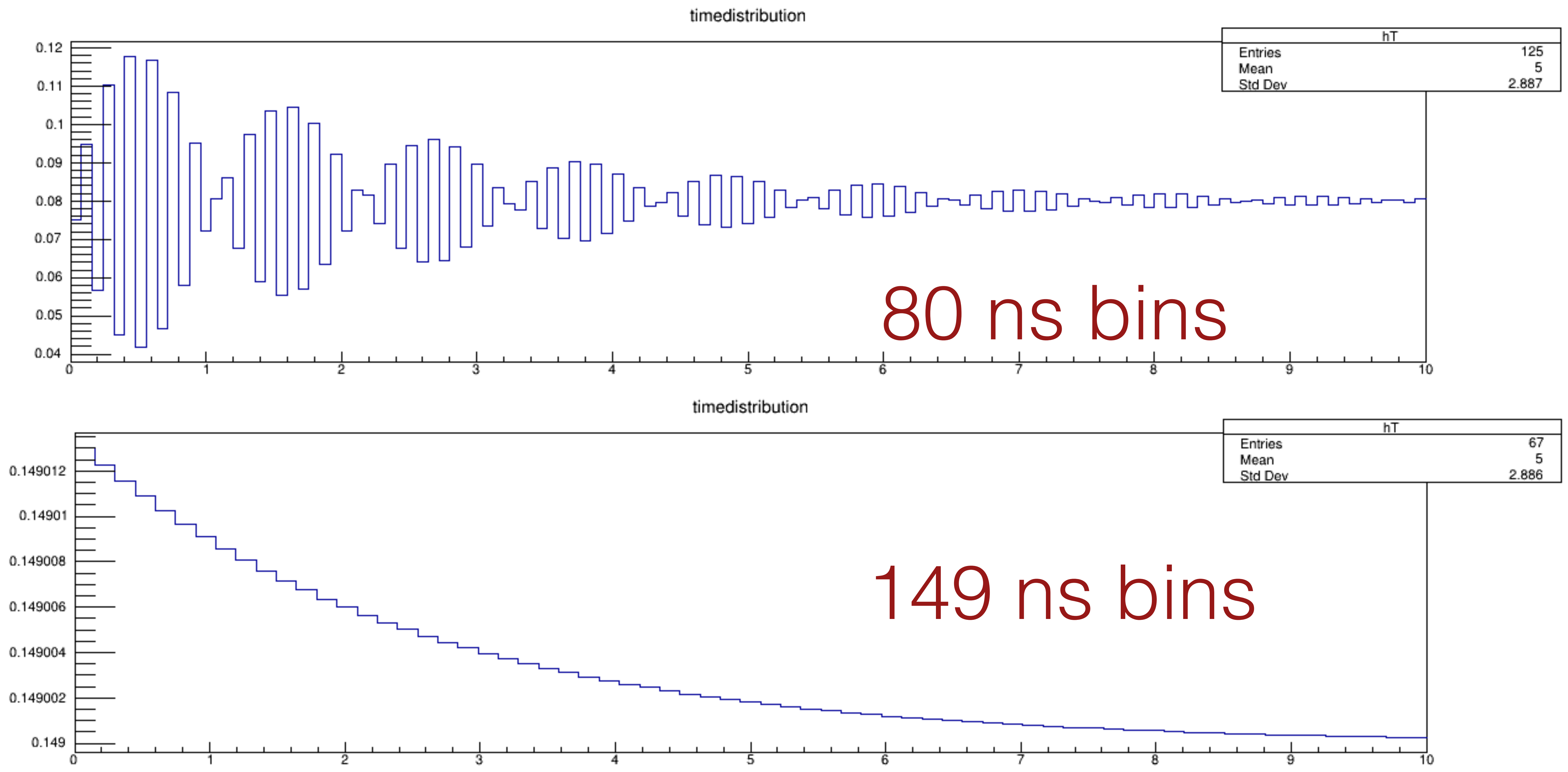
GPU Nodes on Open Science Grid

- Run on Open Science Grid GPU nodes to simulate 10^{11} positrons in 2.5 hours.
- Currently running on two OSG sites with GPU nodes:
  - Omaha: Nvidia Tesla P100, 3584 Cuda cores
  - Syracuse Orange Grid: GeForce GTX 750 Ti, 640 Cuda cores
- Must insure that code is optimized for either architecture.

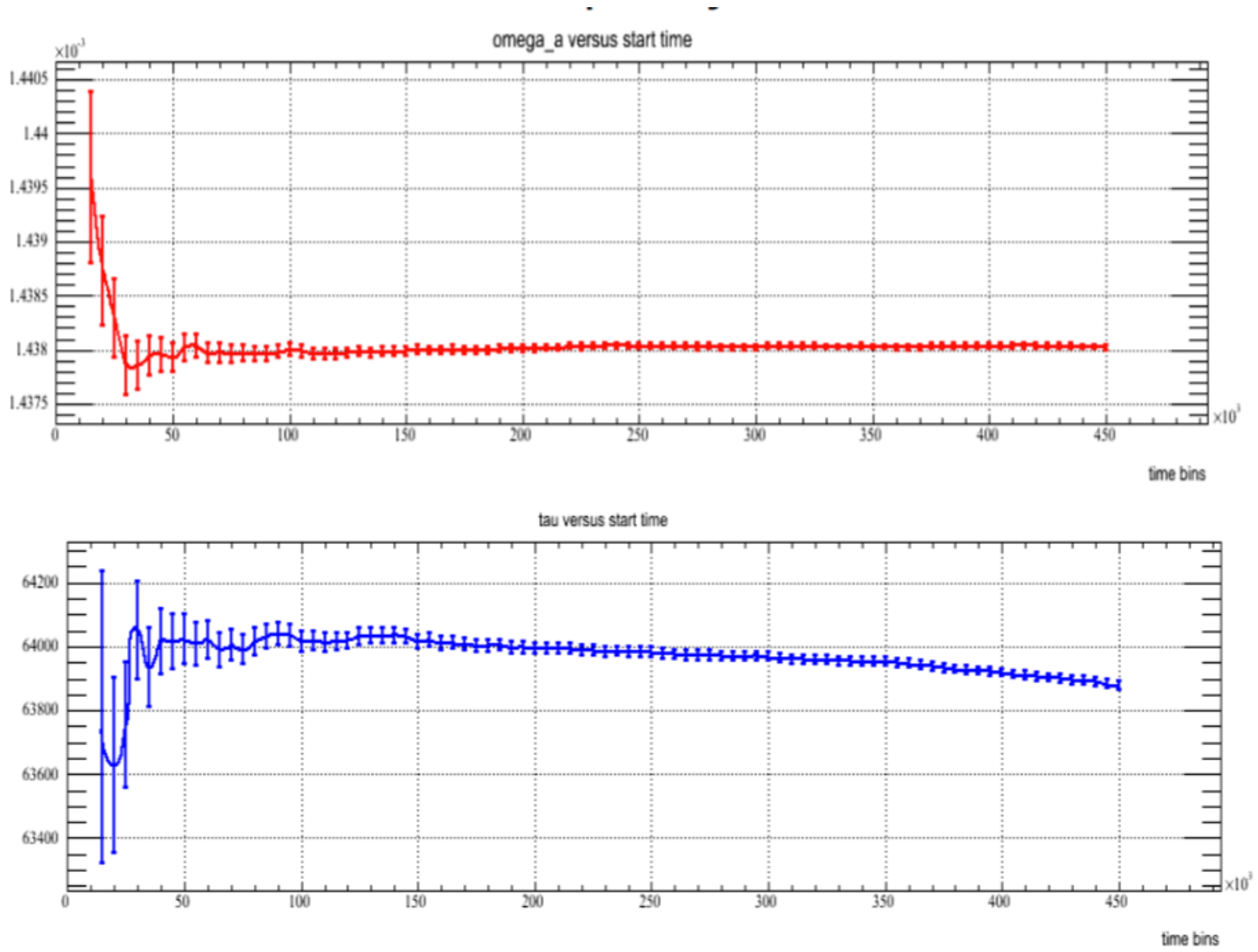


What we learn from the simulation

- Results of the simulation were used to set DAQ parameters.



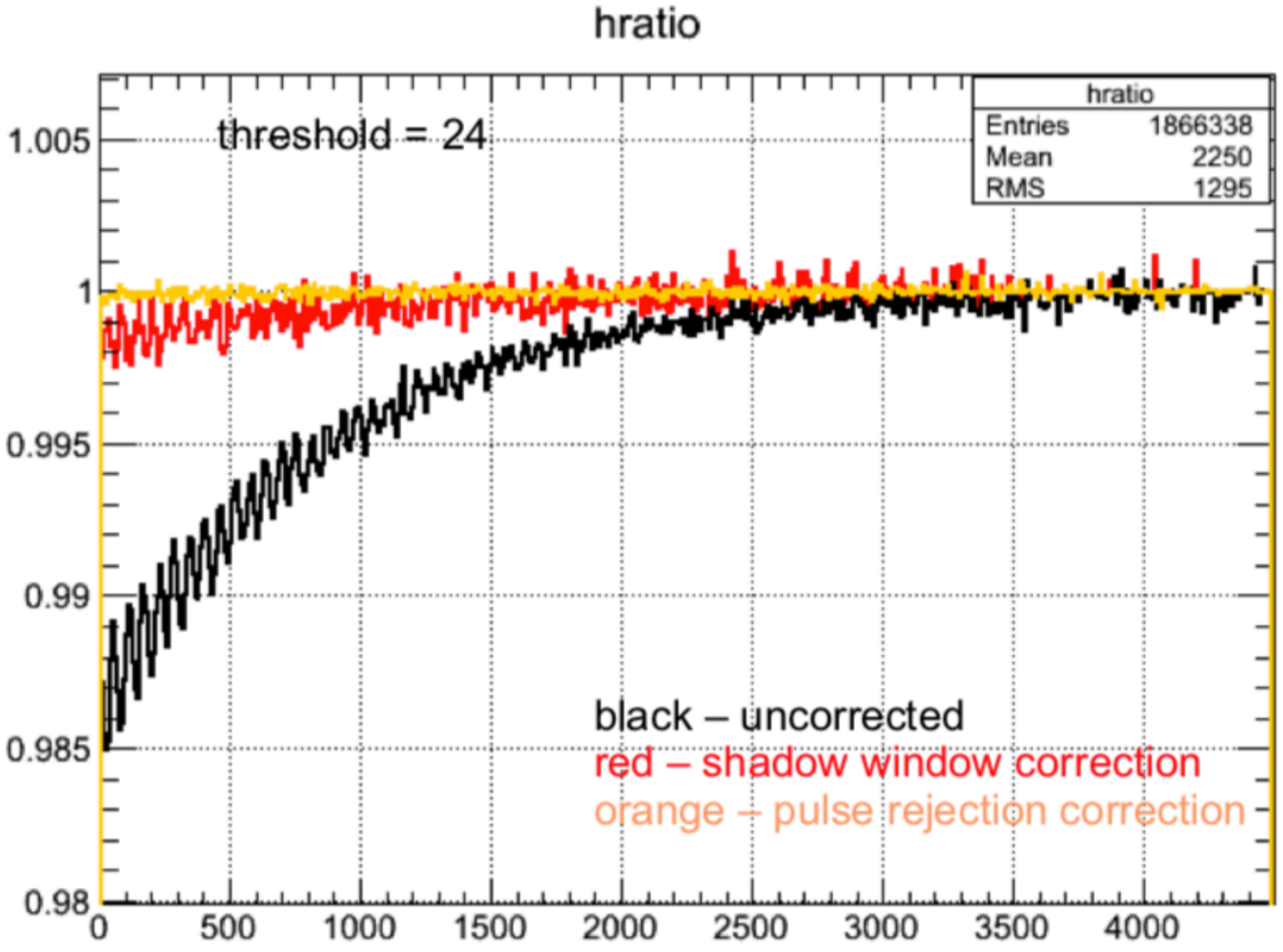
Beam dynamics effects on time spectrum



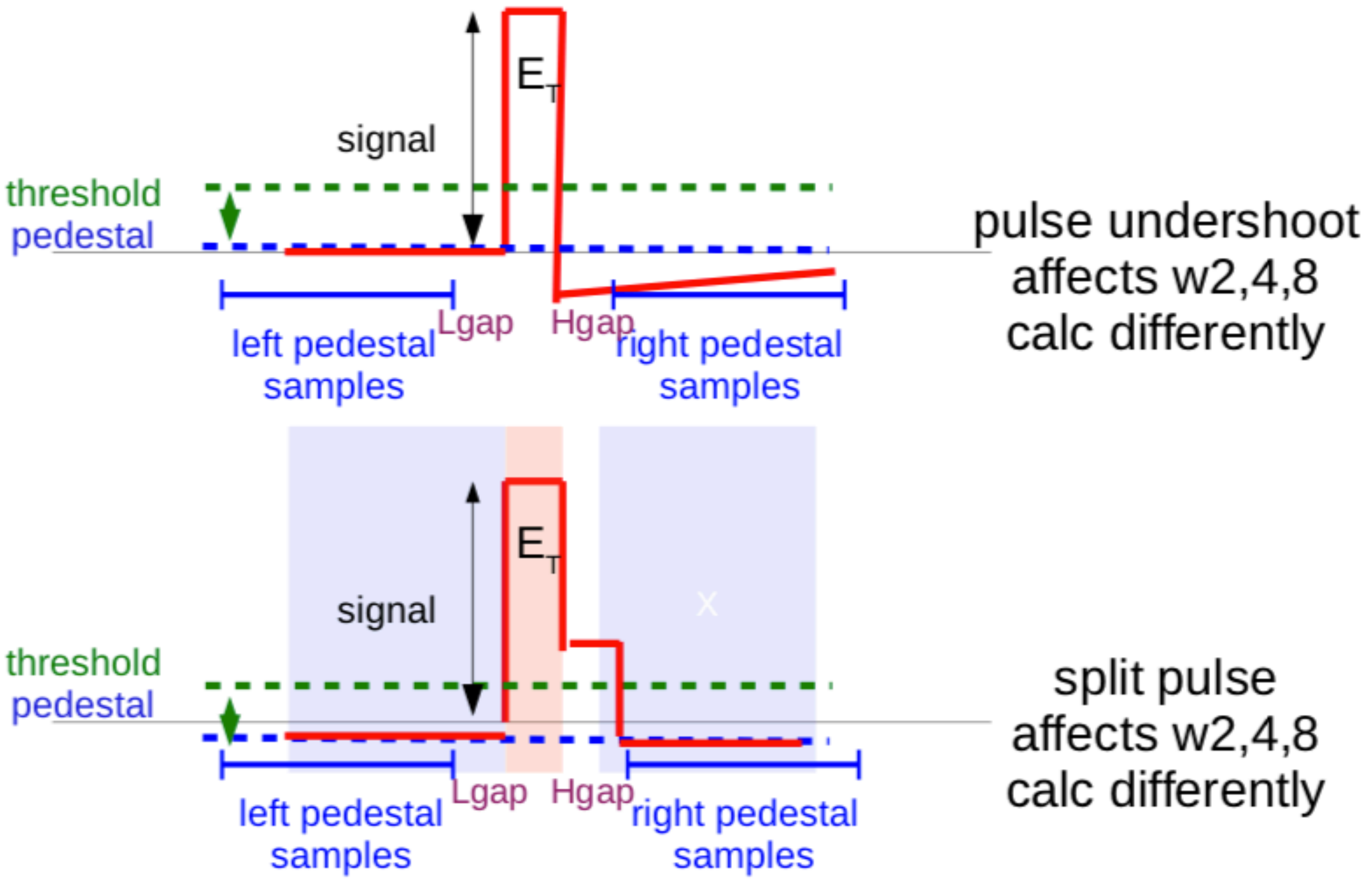
Fit end-time scans

Better understanding of systematic error

- Can simulate effects from pileup, pedestal variations, beam effects.

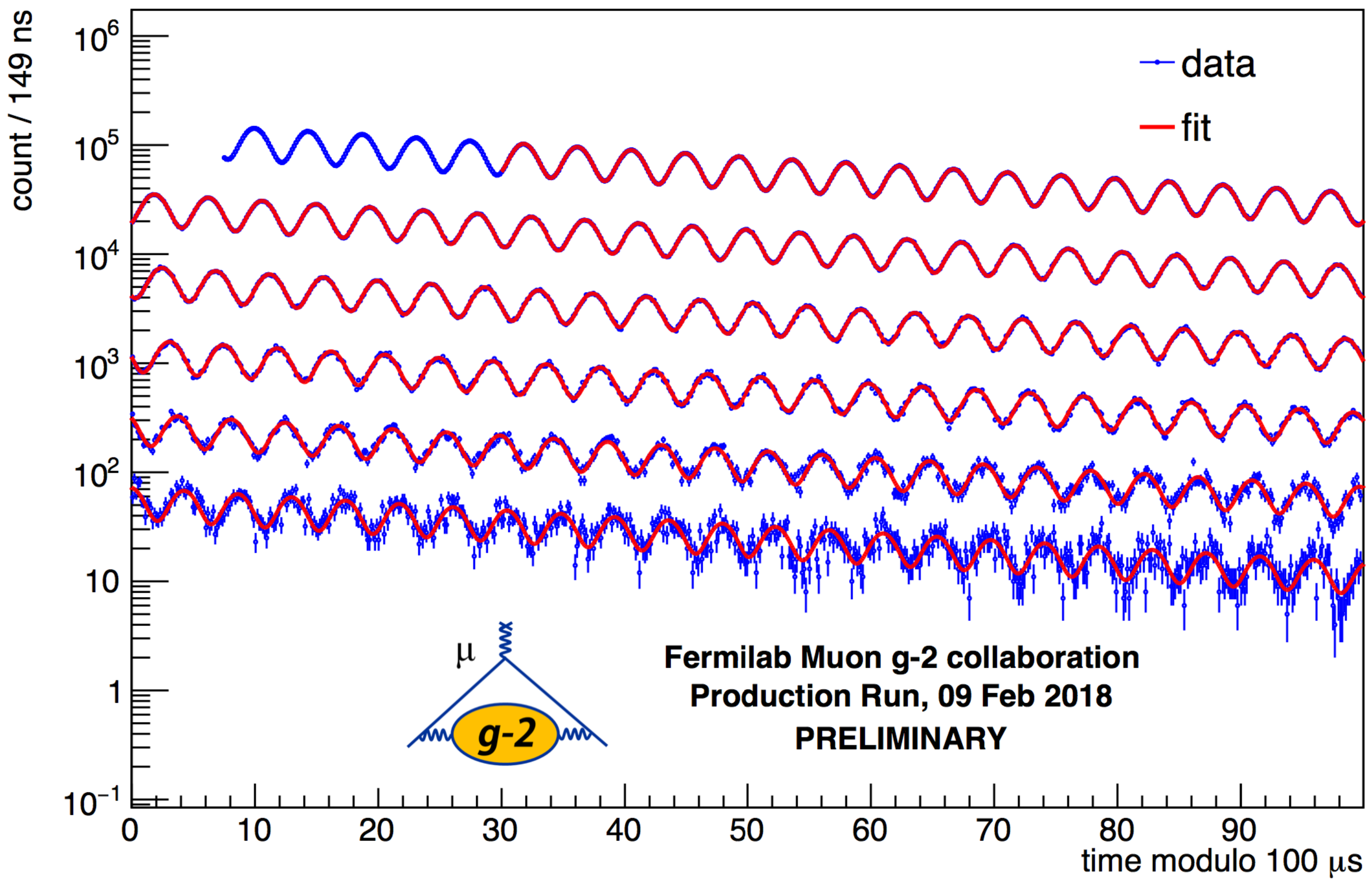


Comparison of different pileup correction methods.



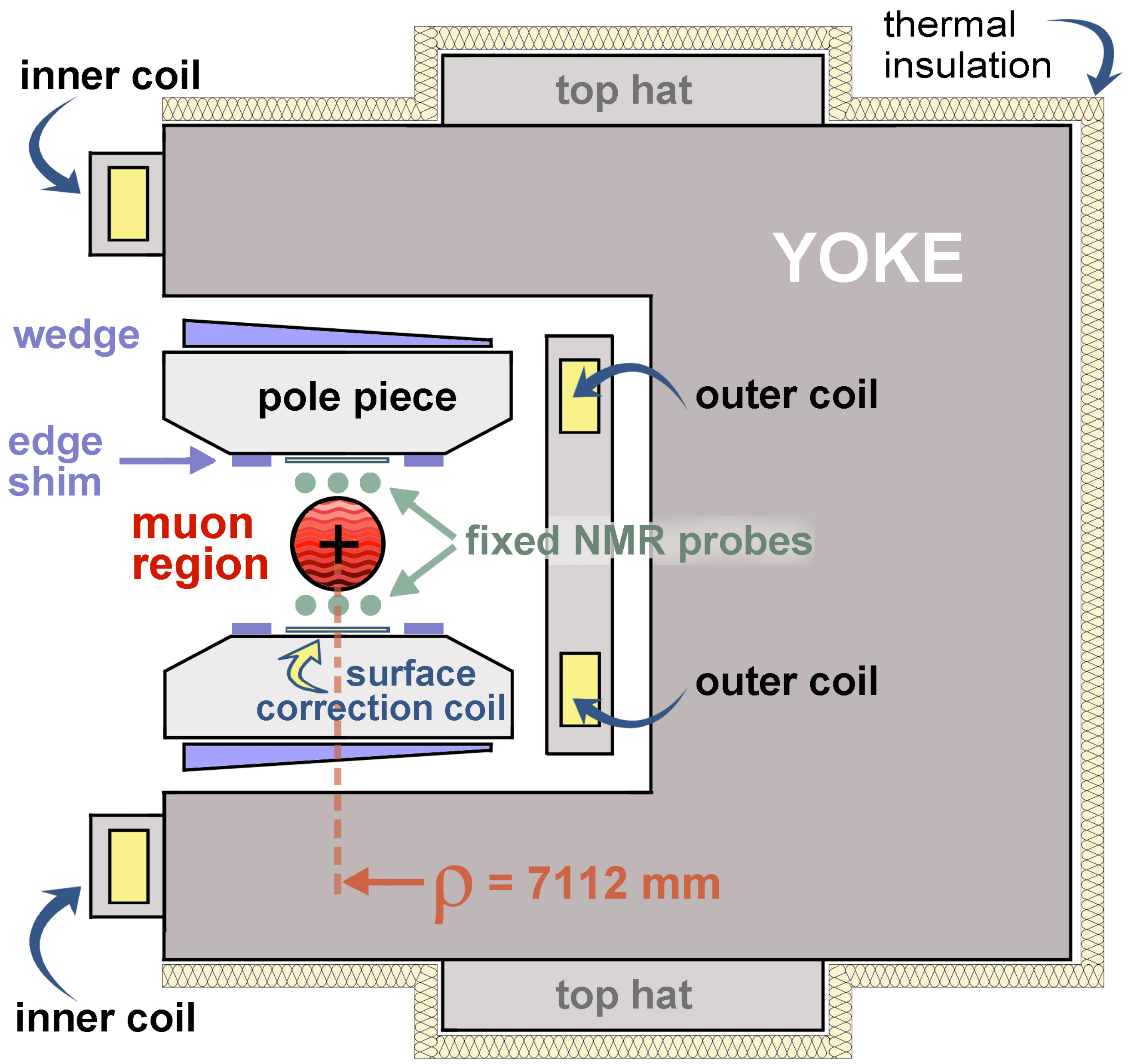
Fast DAQ Summary

- The DAQ for Muon g-2 at Fermilab has been fully commissioned and is performing well.
- The DAQ hardware includes 17 frontend machines, 5 backend machines, 2 dedicated nearline analysis machines, 3 computers for slow control, 3 servers, and 24 beagle bones running 67 MIDAS frontends.
- Takes an input data rate of 20 GB/s, and reduces that to < 200 MB/s in the event builder via GPU processing and lossless compression.
- DAQ is fully operational and currently is writing data with >5000 positrons per second.



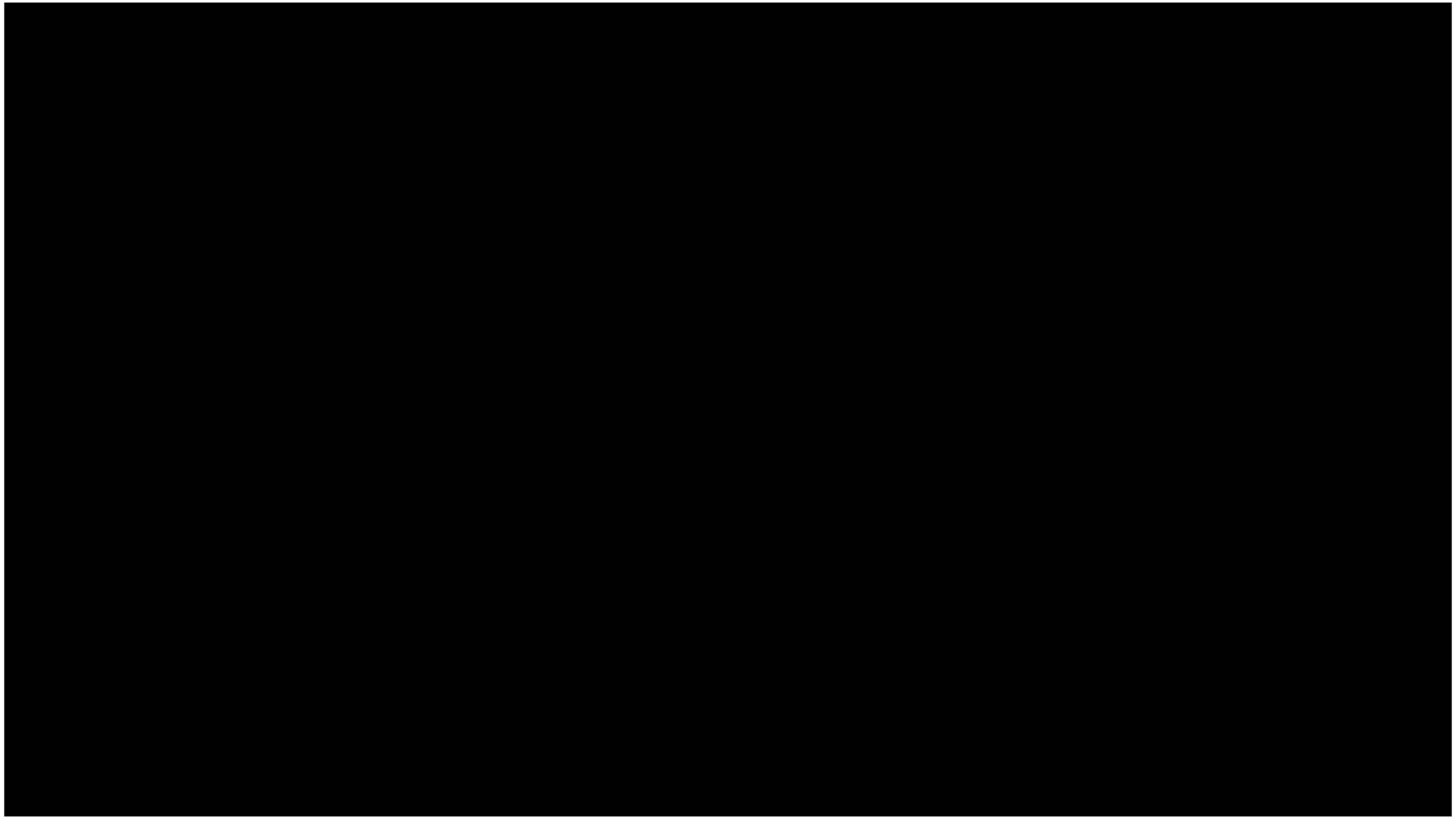
High-precision measurement of the magnetic field

- Muons are stored in a ring magnet
- Target precision: <70 part-per-billion
- Uniformity requirement: +/-1 ppm cross-section, 100~ppm azimuthal
- Stability requirement: drift within 100s of ppb
- GPU: speed up the online analysis and the field stabilization feedback loop



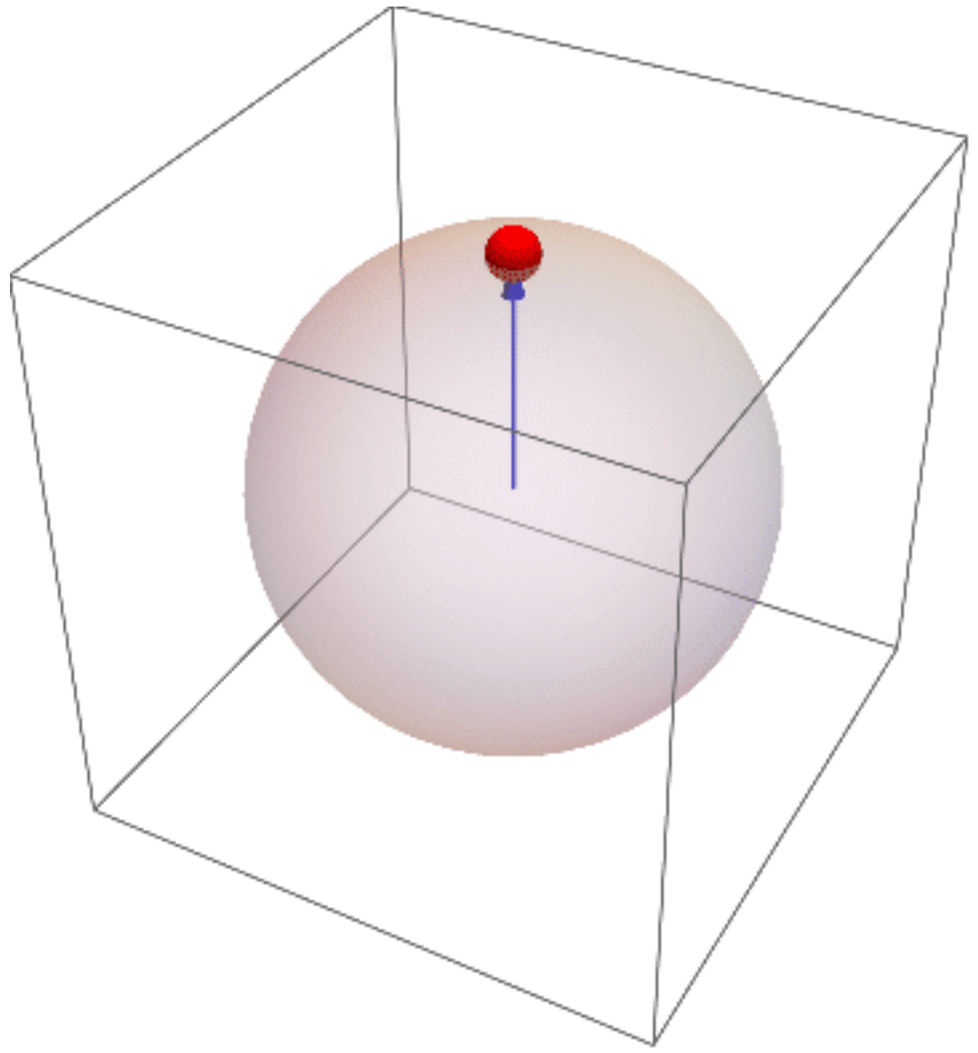
Transportation and re-installation of the magnet

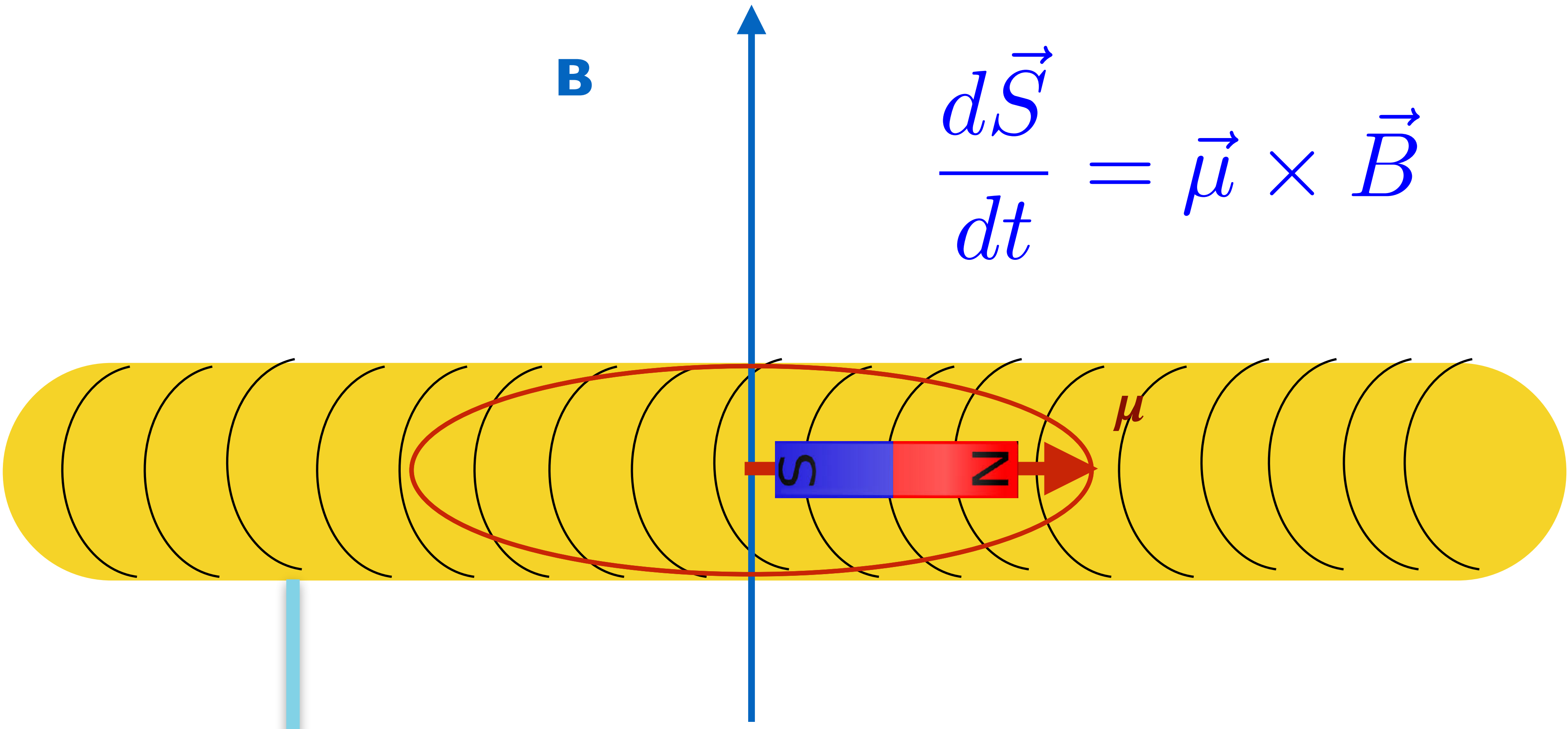
Transportation and re-installation of the magnet



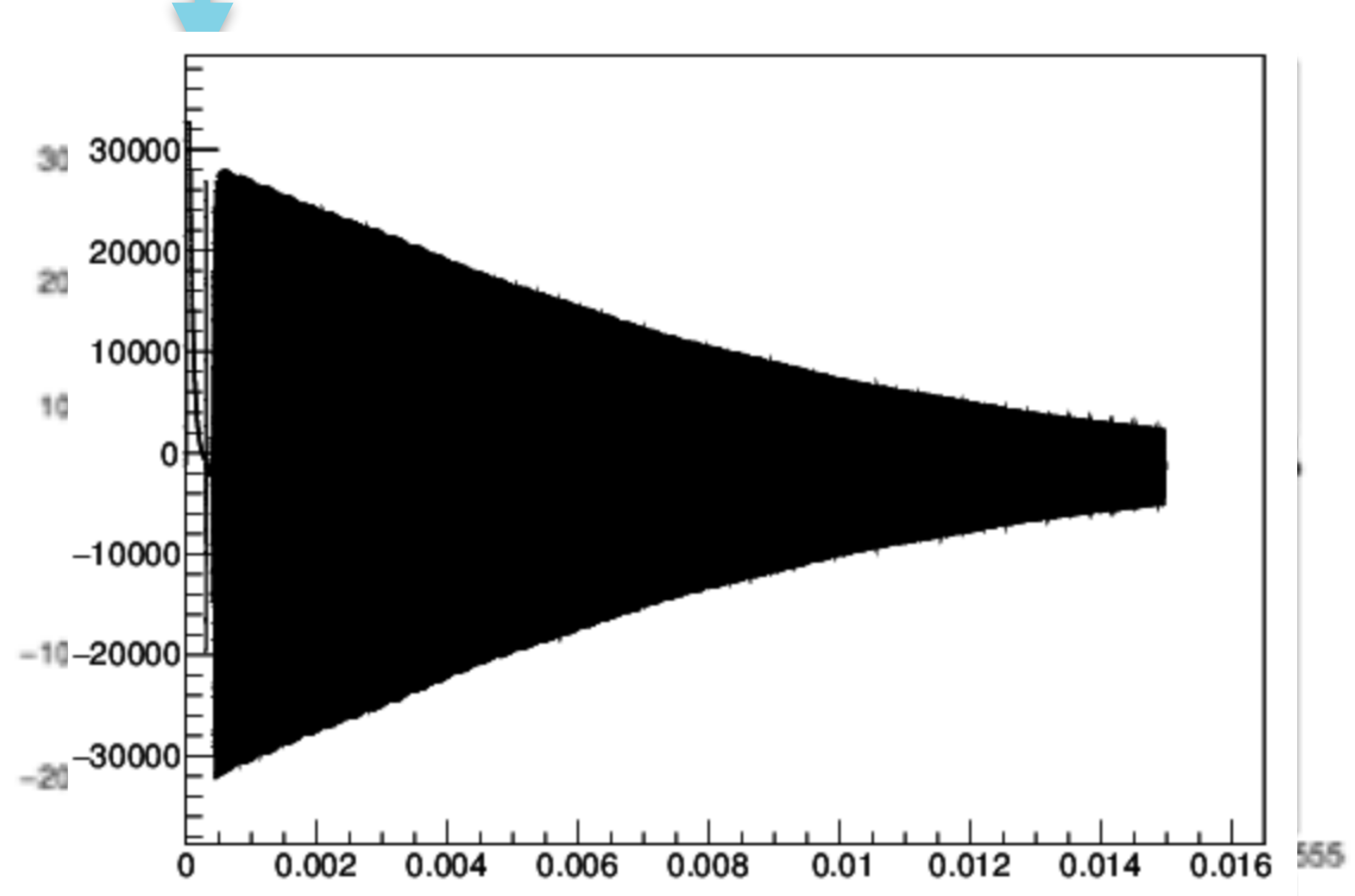
Measurement principle

- Nuclear Magnetic Resonance (NMR) measurement principle
  - Detecting signals from precessing protons
  - Measure their precession angular frequency ω_p
  - Signal decays due to relaxation
    - Free Inductor Decay signal (FID)



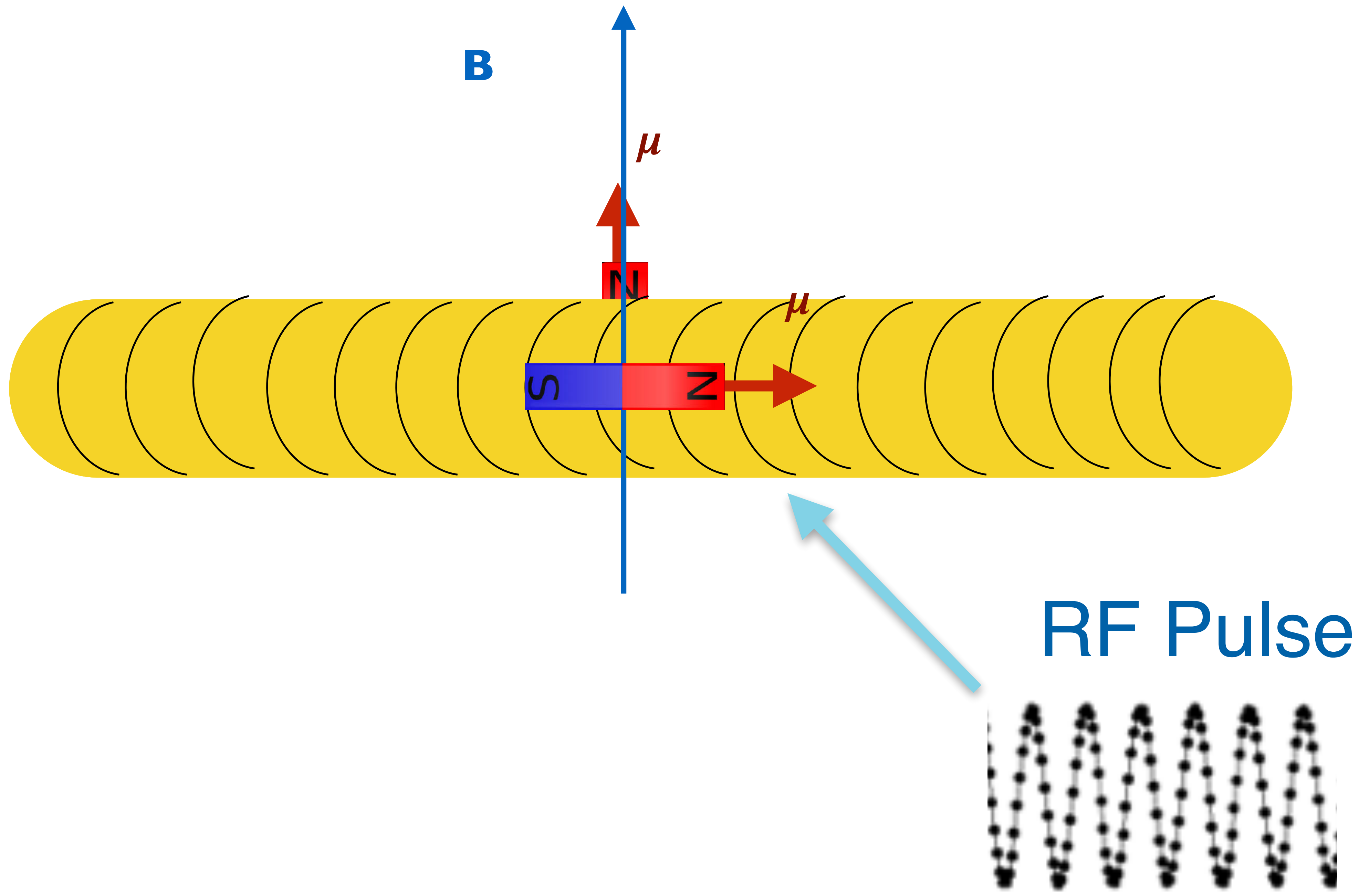


$$\frac{d\vec{S}}{dt} = \vec{\mu} \times \vec{B}$$



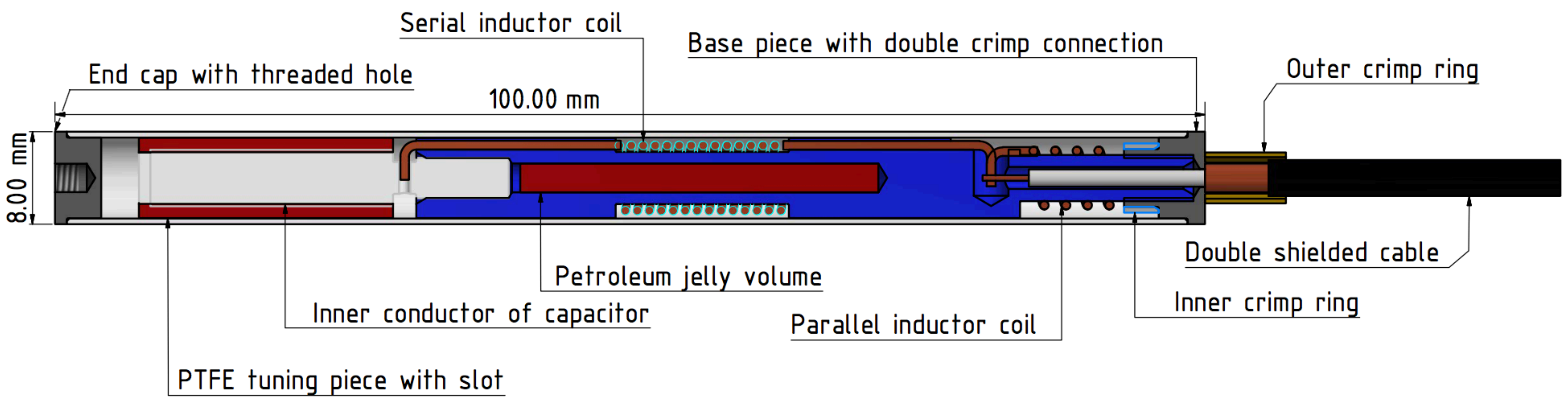
Measurement principle

- Nuclear Magnetic Resonance (NMR) activation principle
  - Protons' magnetic moments aligned with the magnetic field
  - Inject a short Radio Frequency (RF) pulse with the same angular frequency ω_p (**on resonance**) through the coil
  - The RF pulse tips the proton spins by 90 degree, with well-tuned amplitude and duration

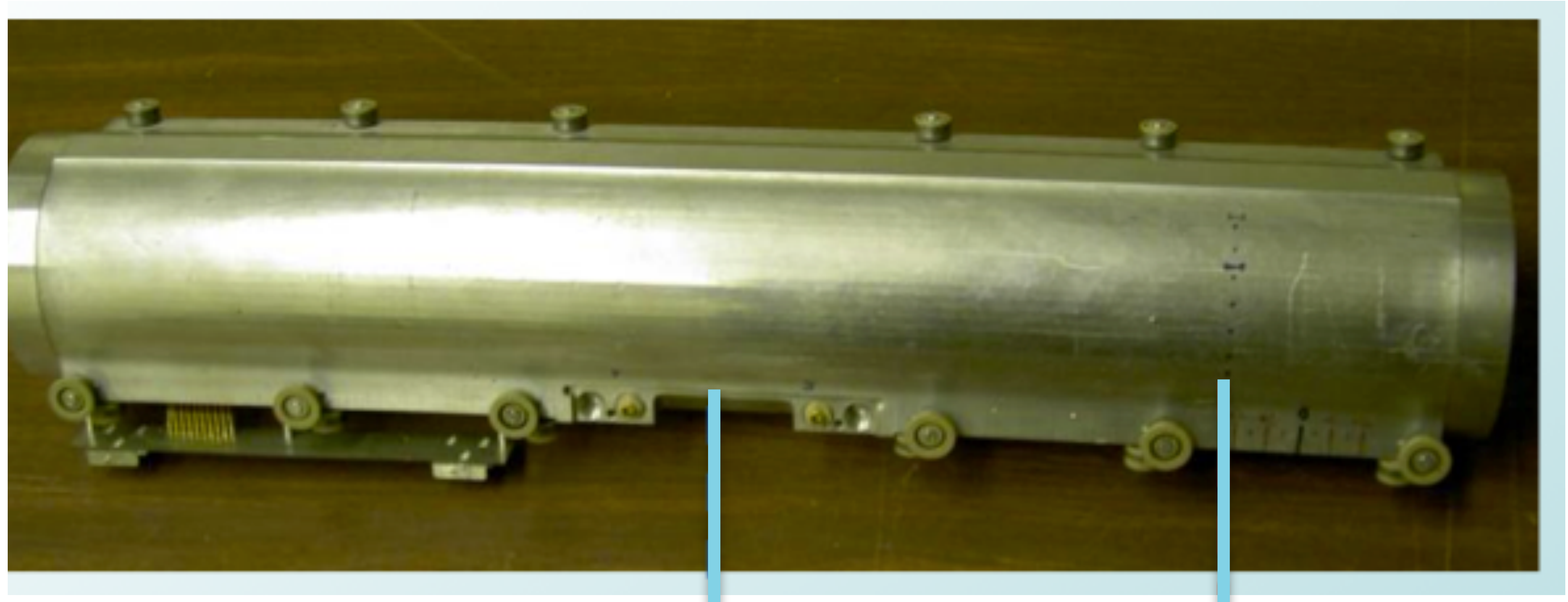


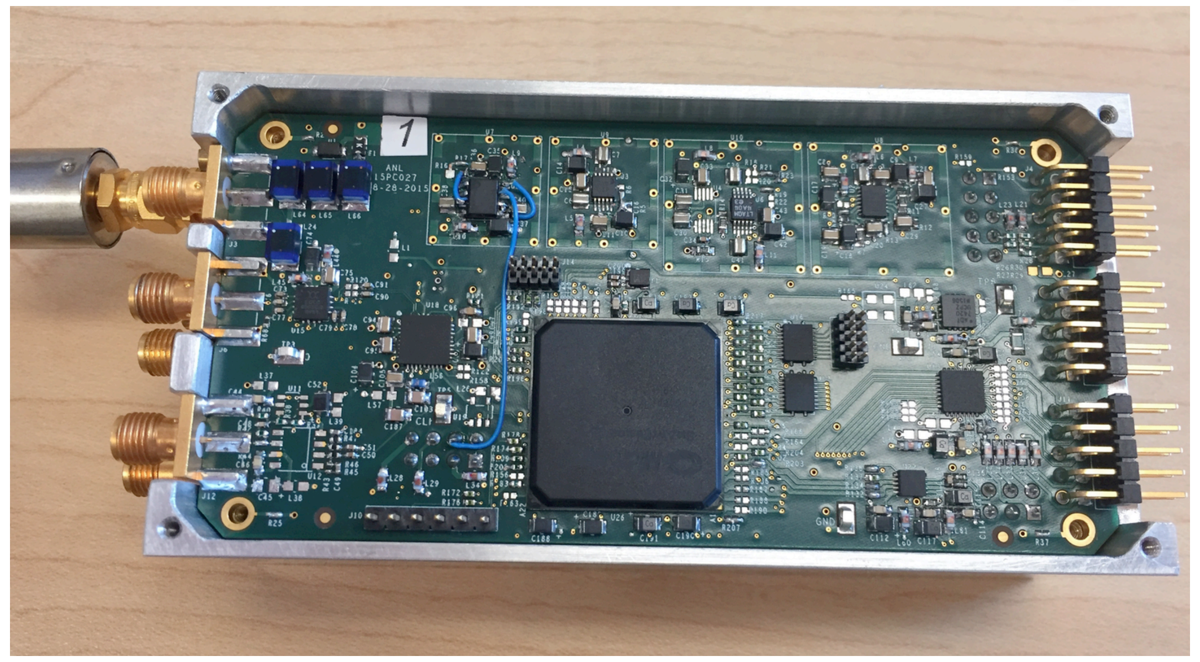
NMR probes and devices

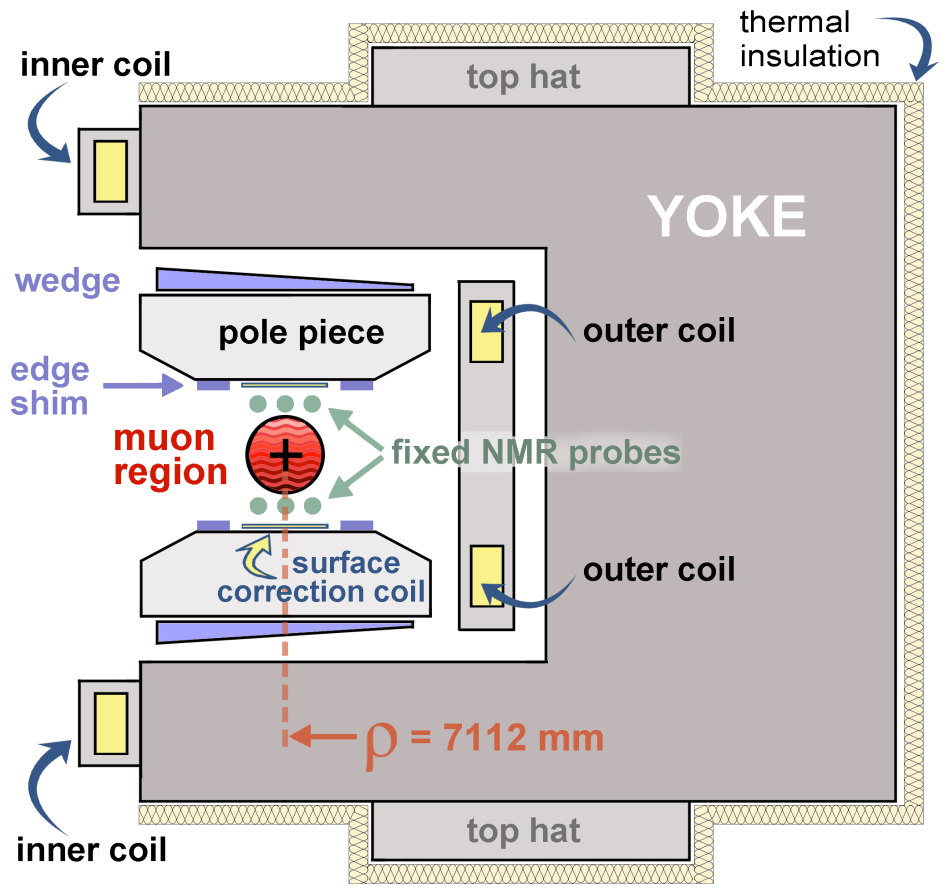
Probe Schematic

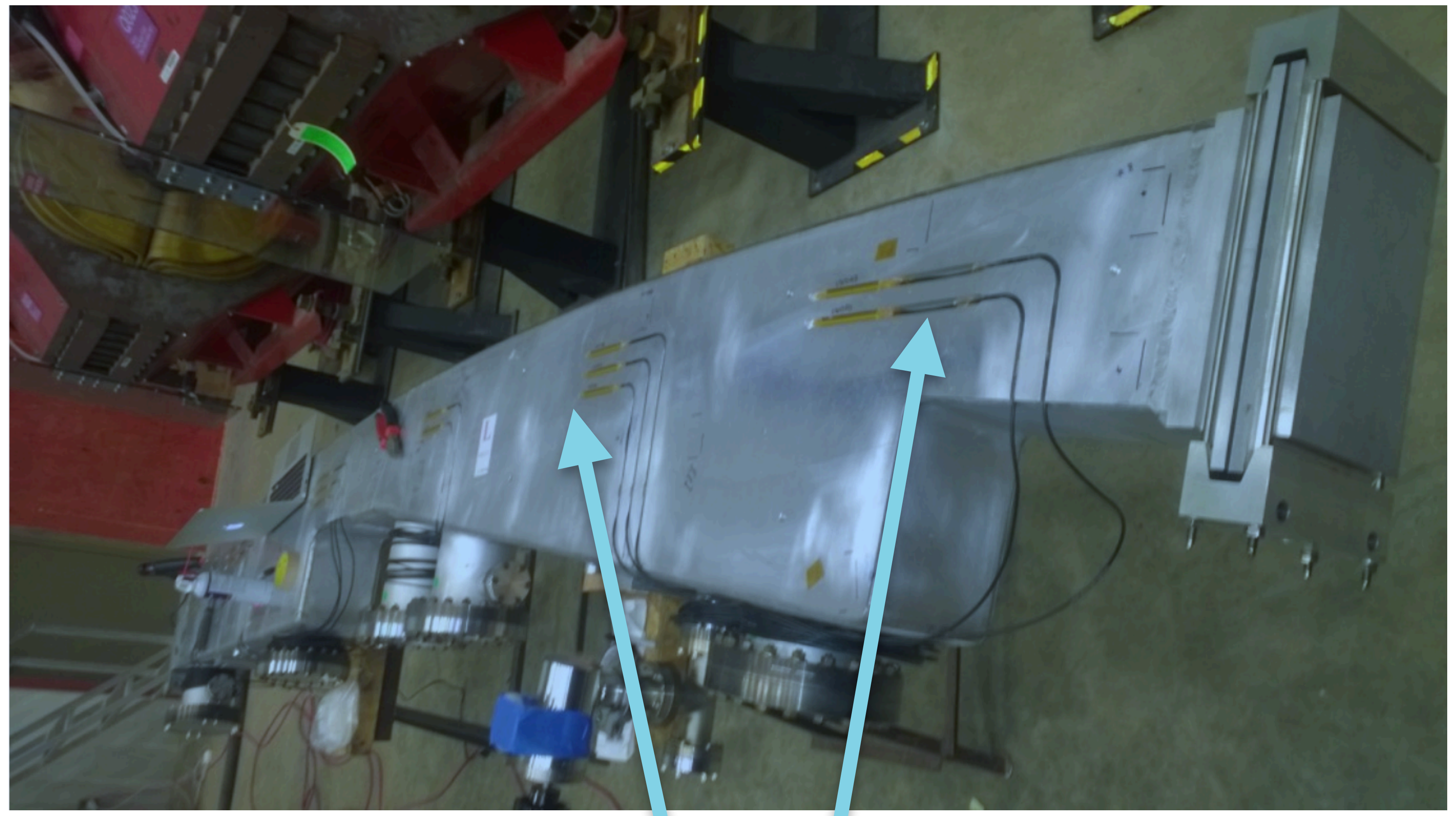


Field Scanner





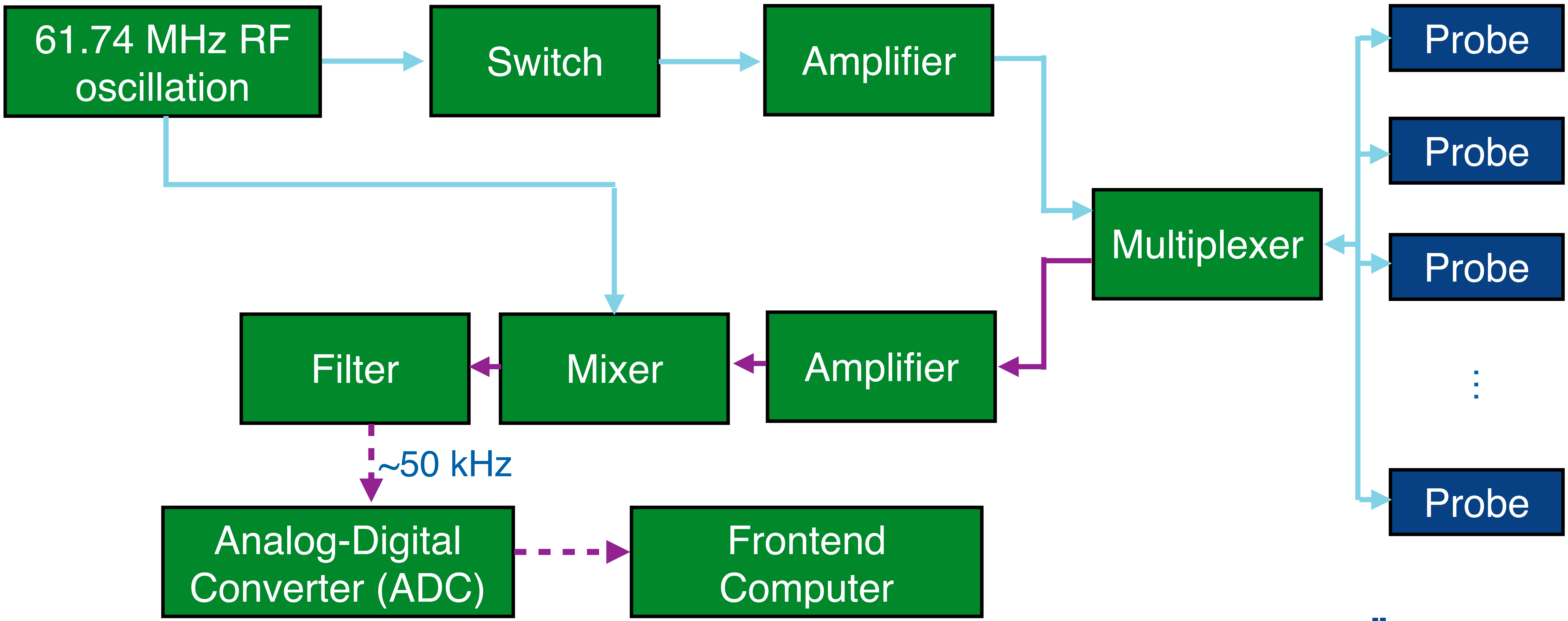




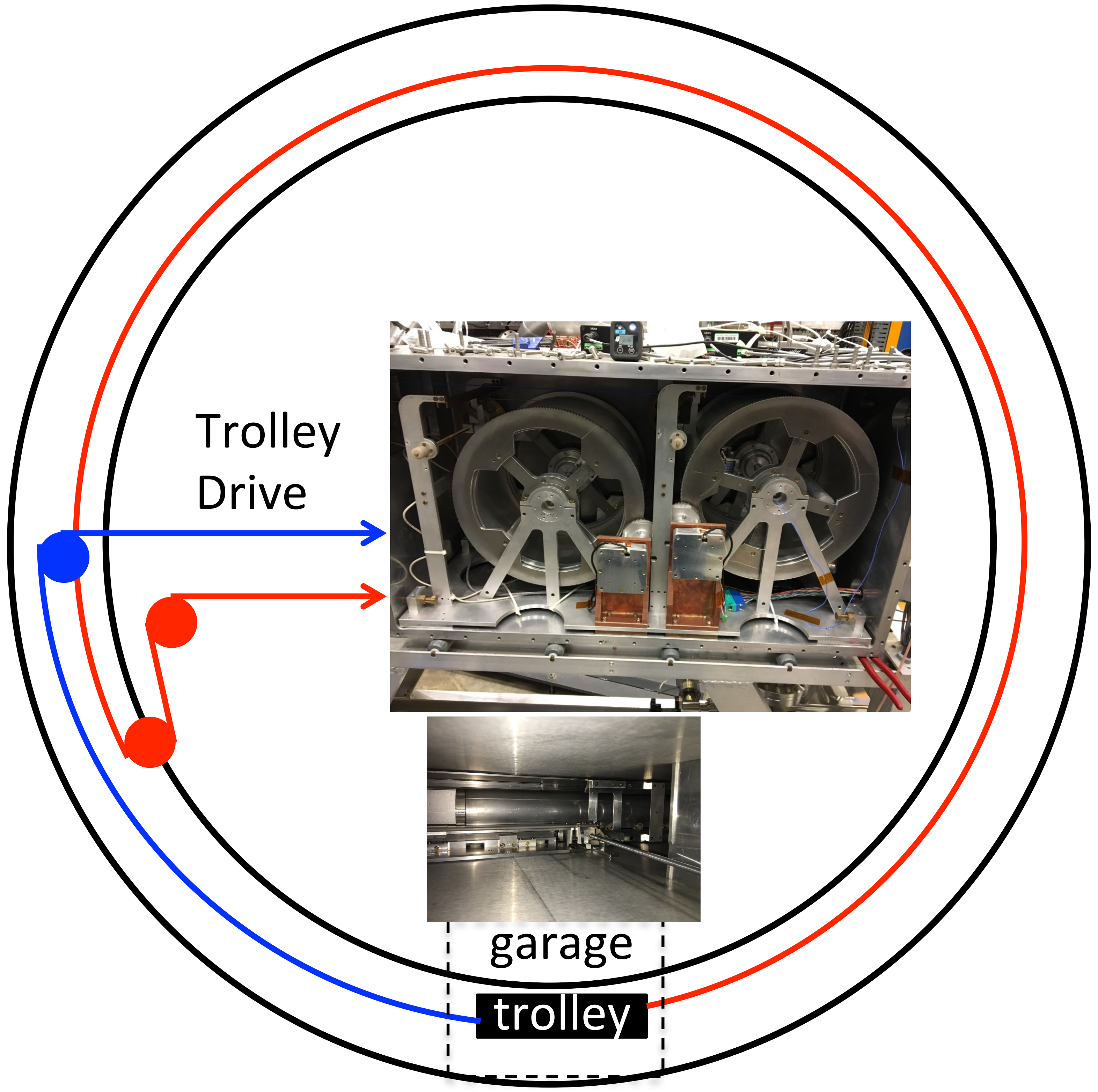
Fixed Monitoring Probes (378)

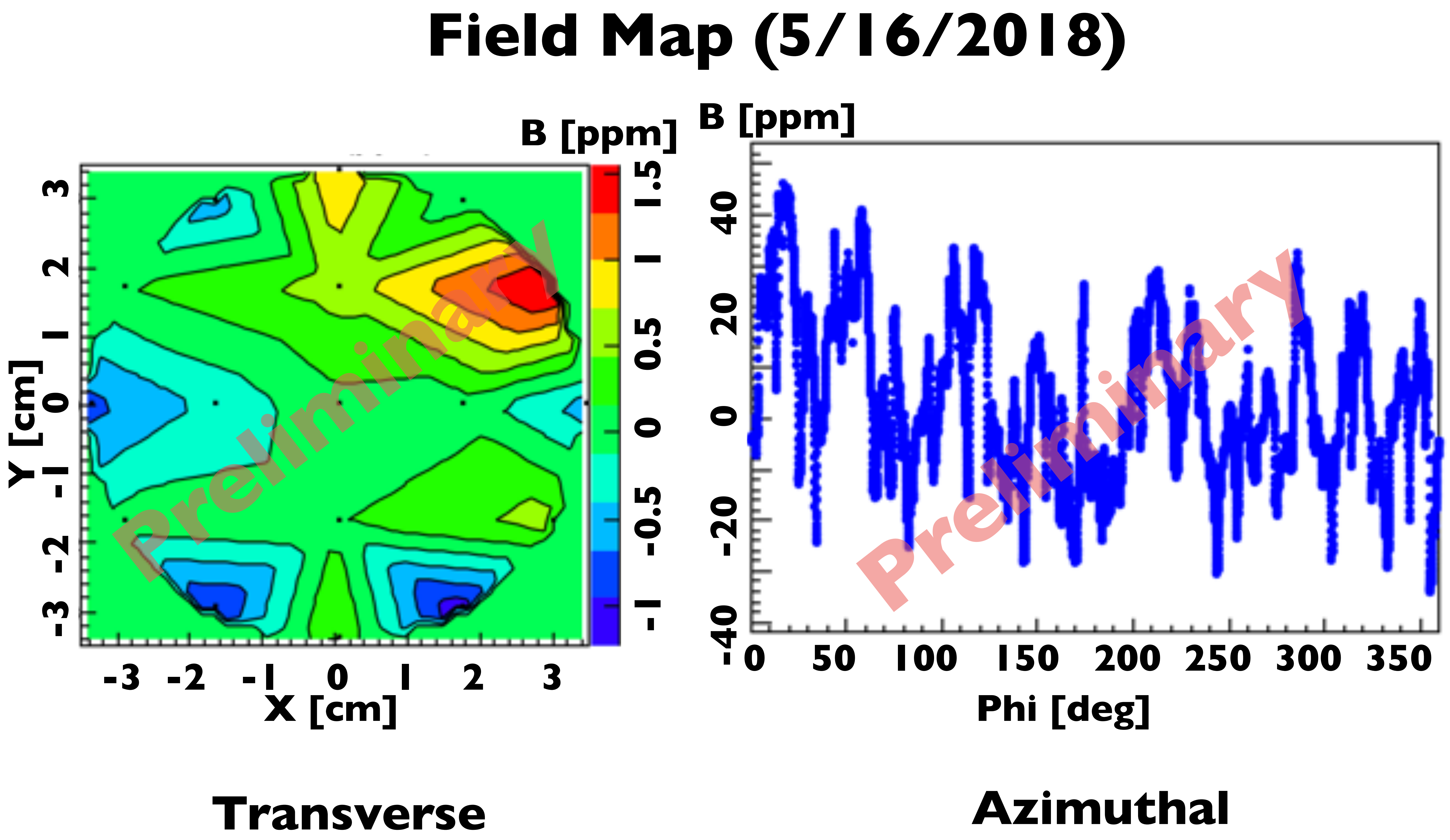
NMR probe read-out system

~50 kHz off resonance

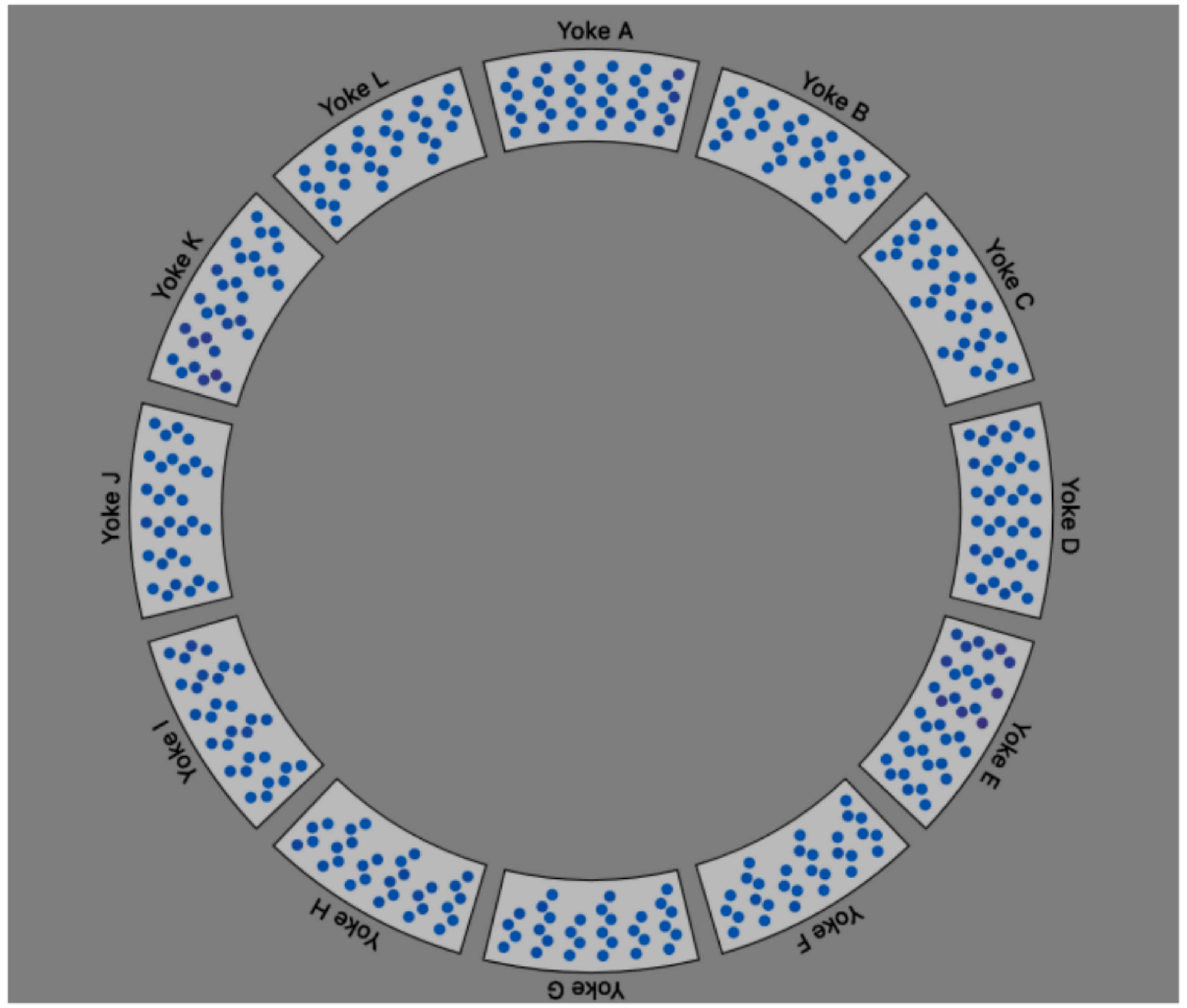


Scanning the magnetic field

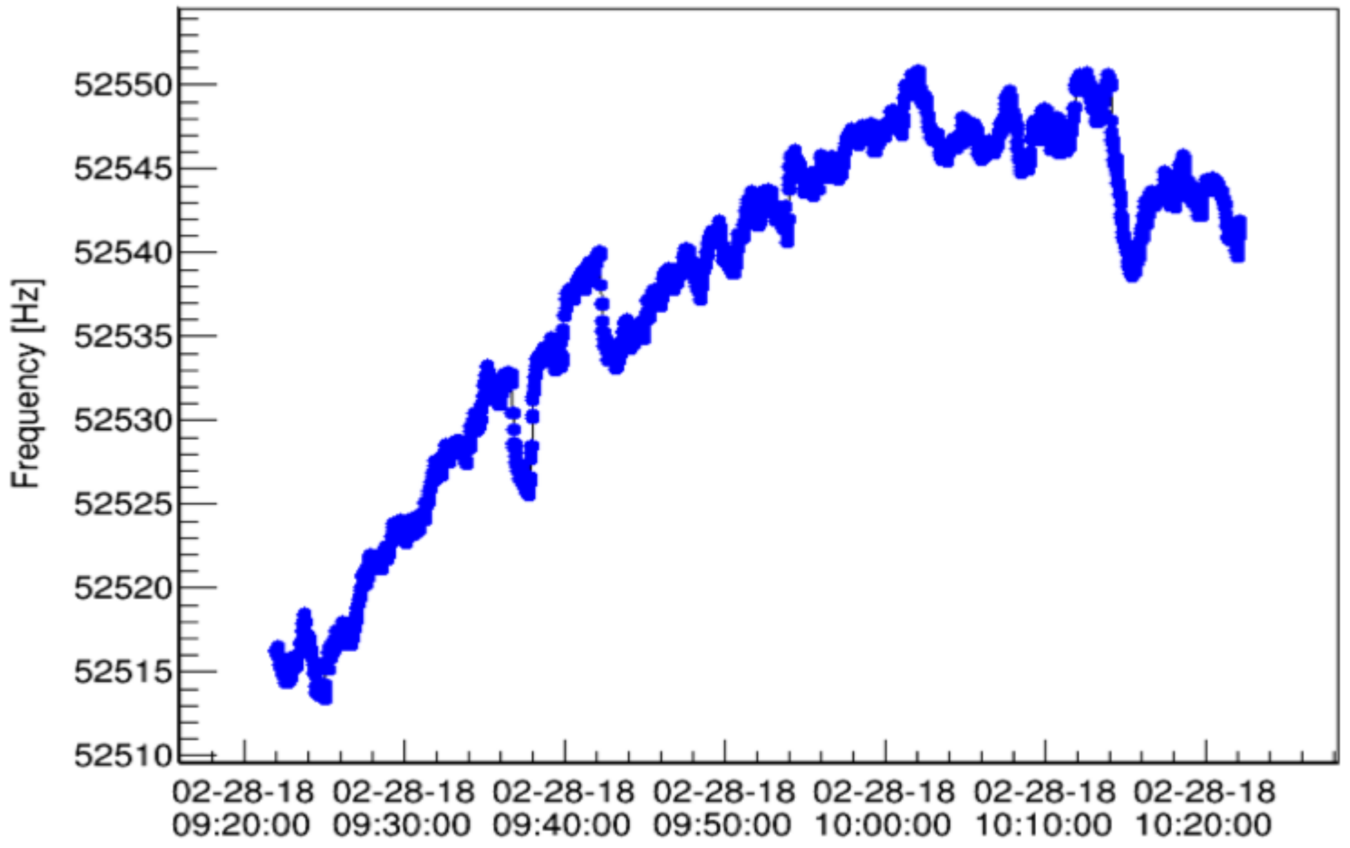




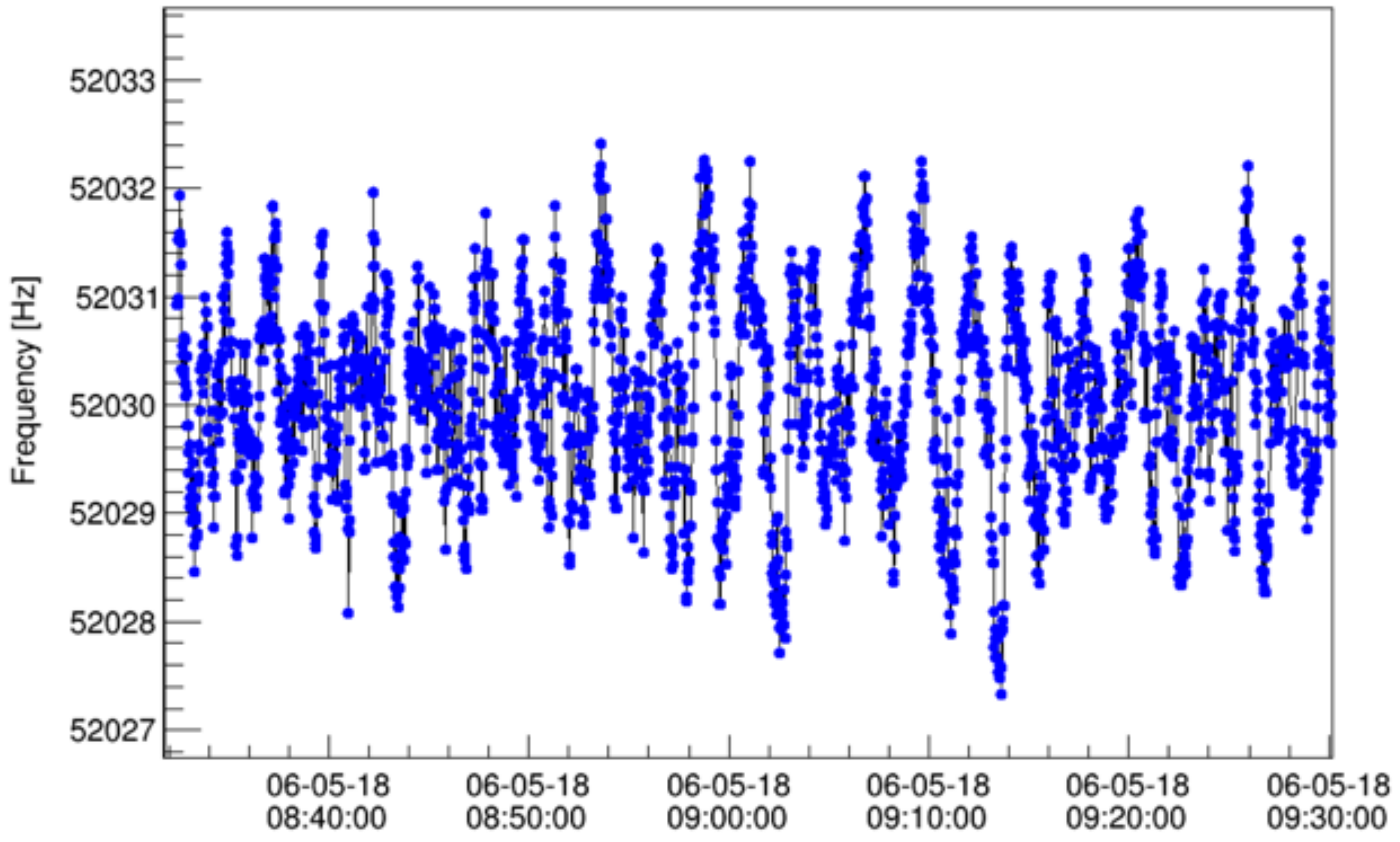
Magnetic field stabilization



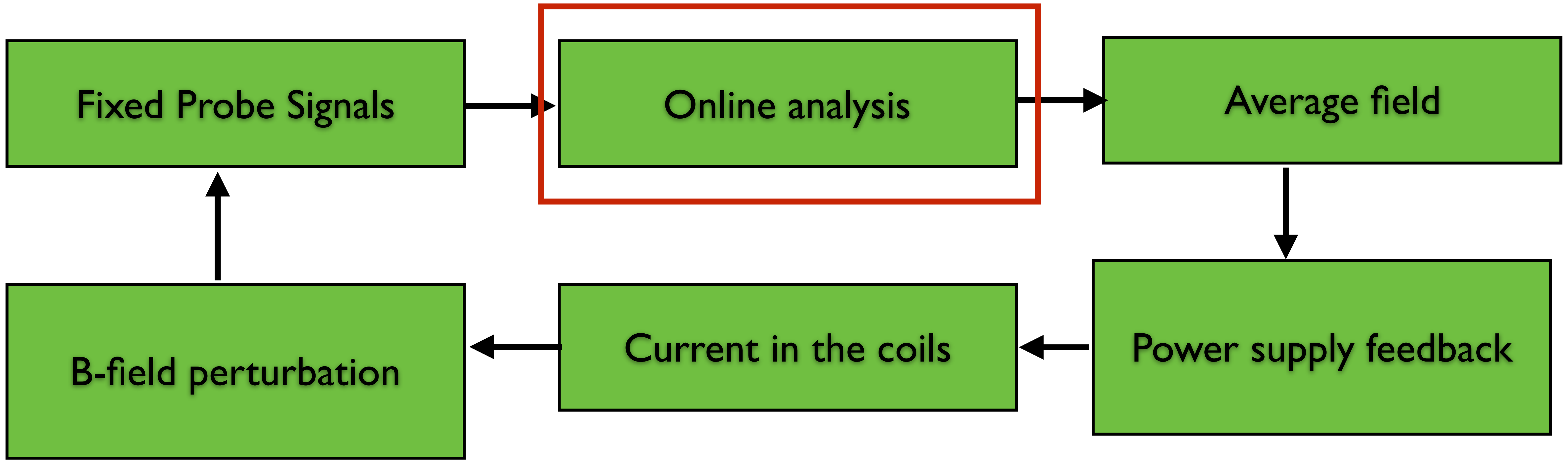
Monitoring Field drift across the ring



Field drift without feedback



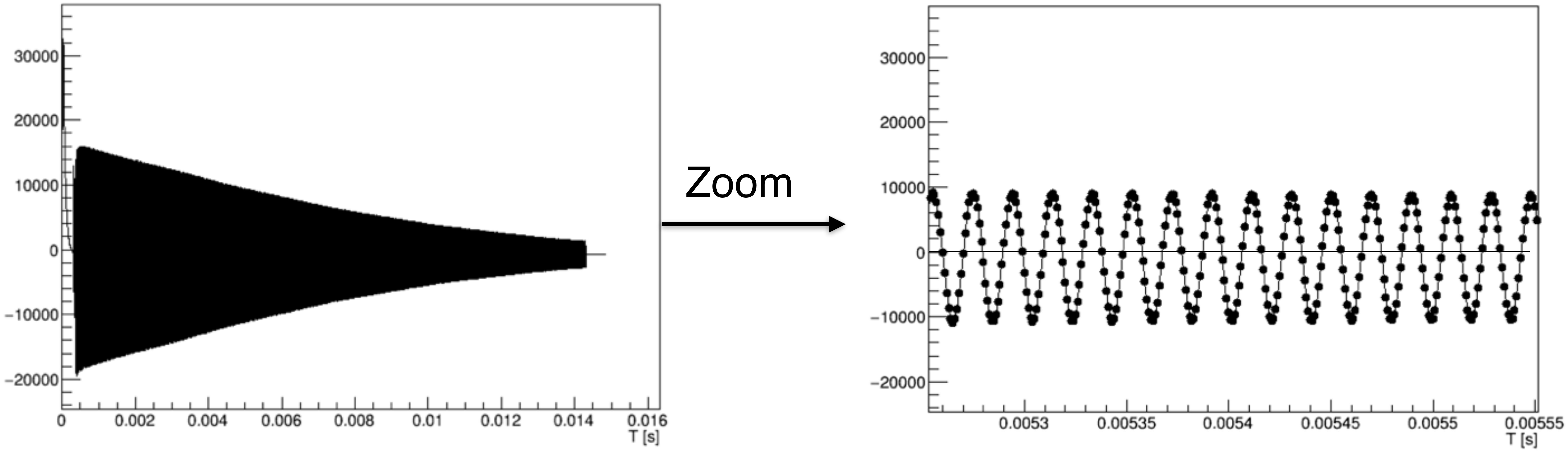
Field drift with active feedback



Need fast online analysis to speed up the feedback look.

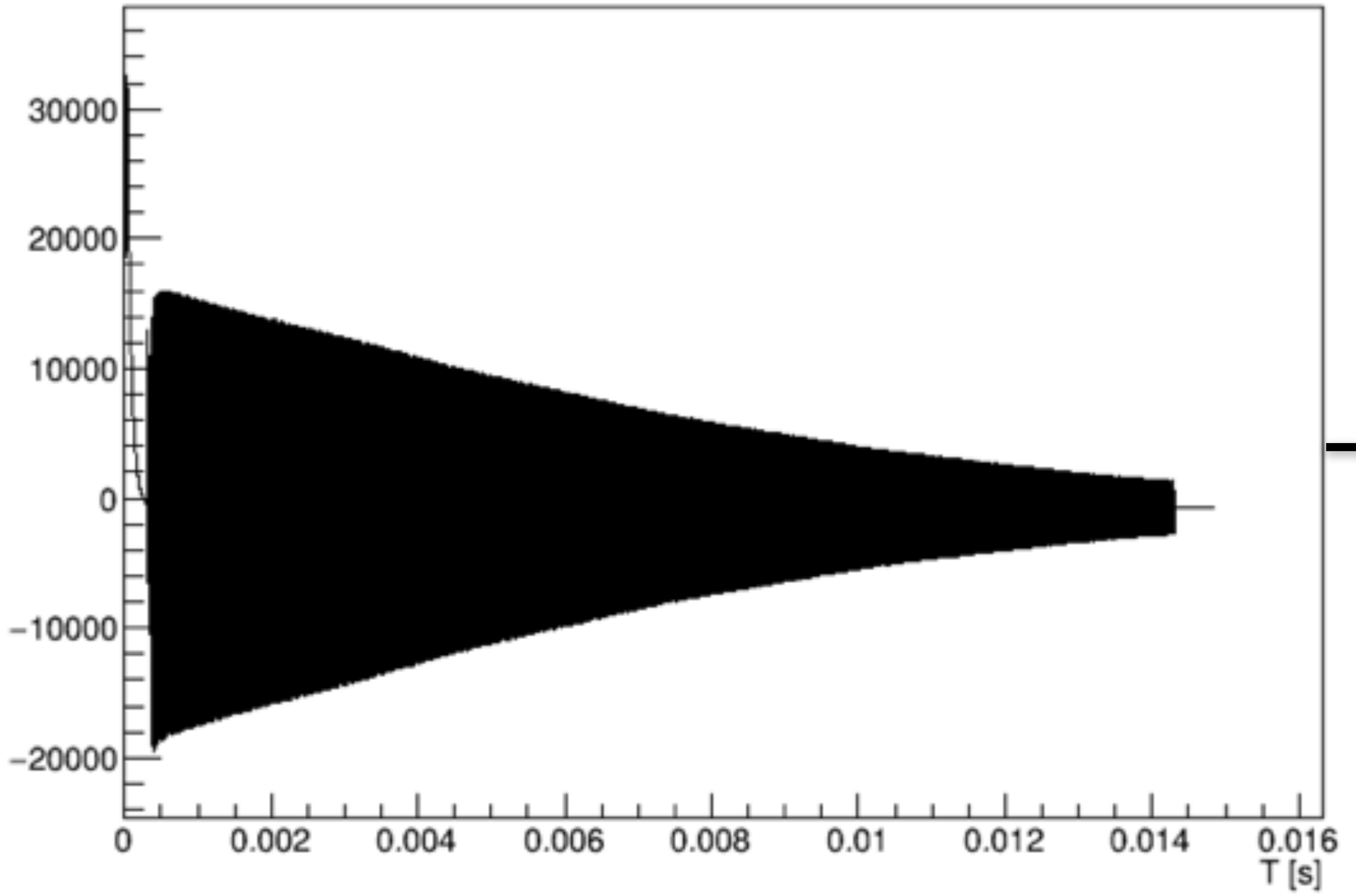
Analysis of the NMR signal

- Basic frequency extraction methods
  - Zero-counting

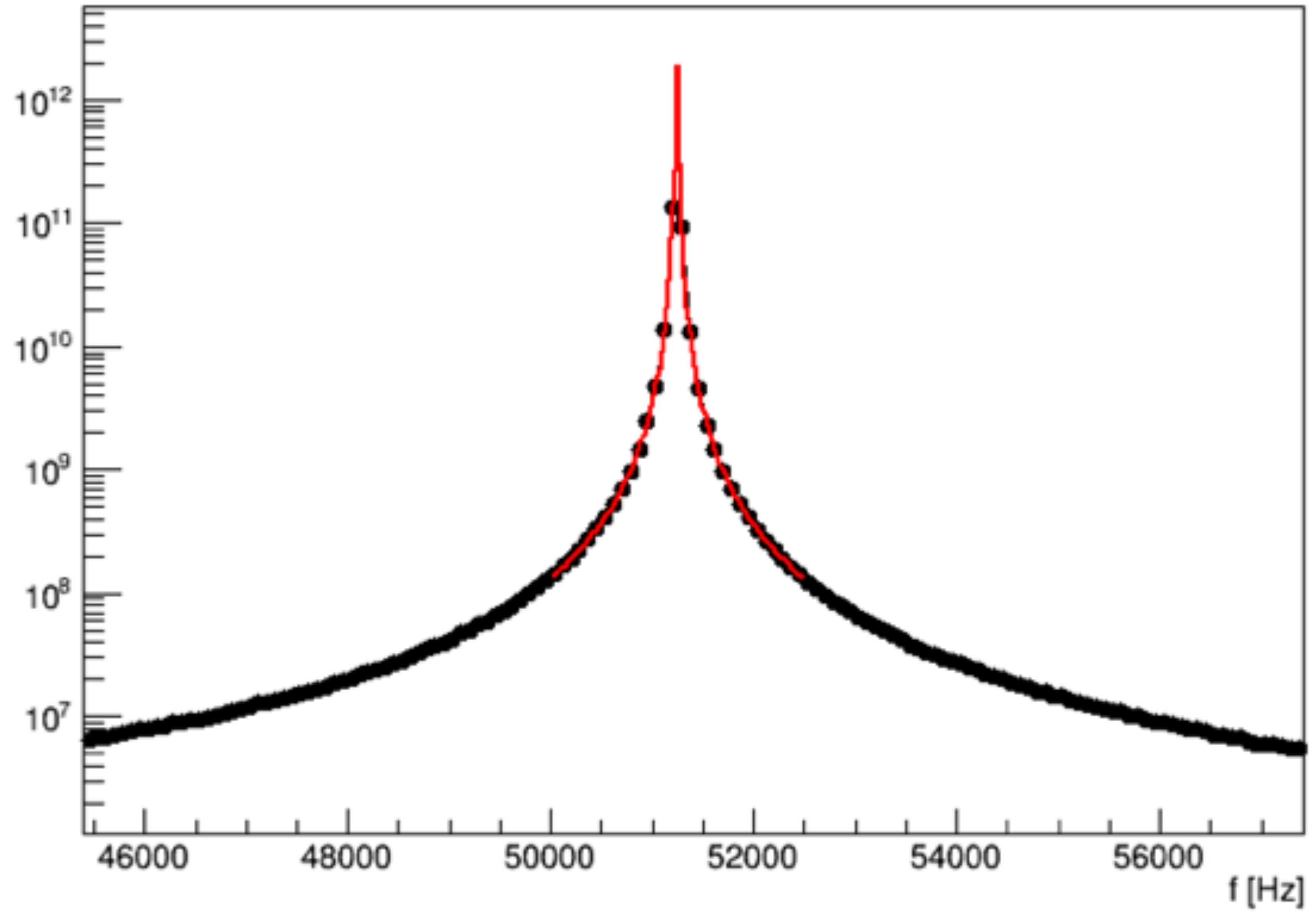


Analysis of the NMR signal

- Basic frequency extraction methods
  - Zero-counting
  - Frequency spectrum fitting/averaging

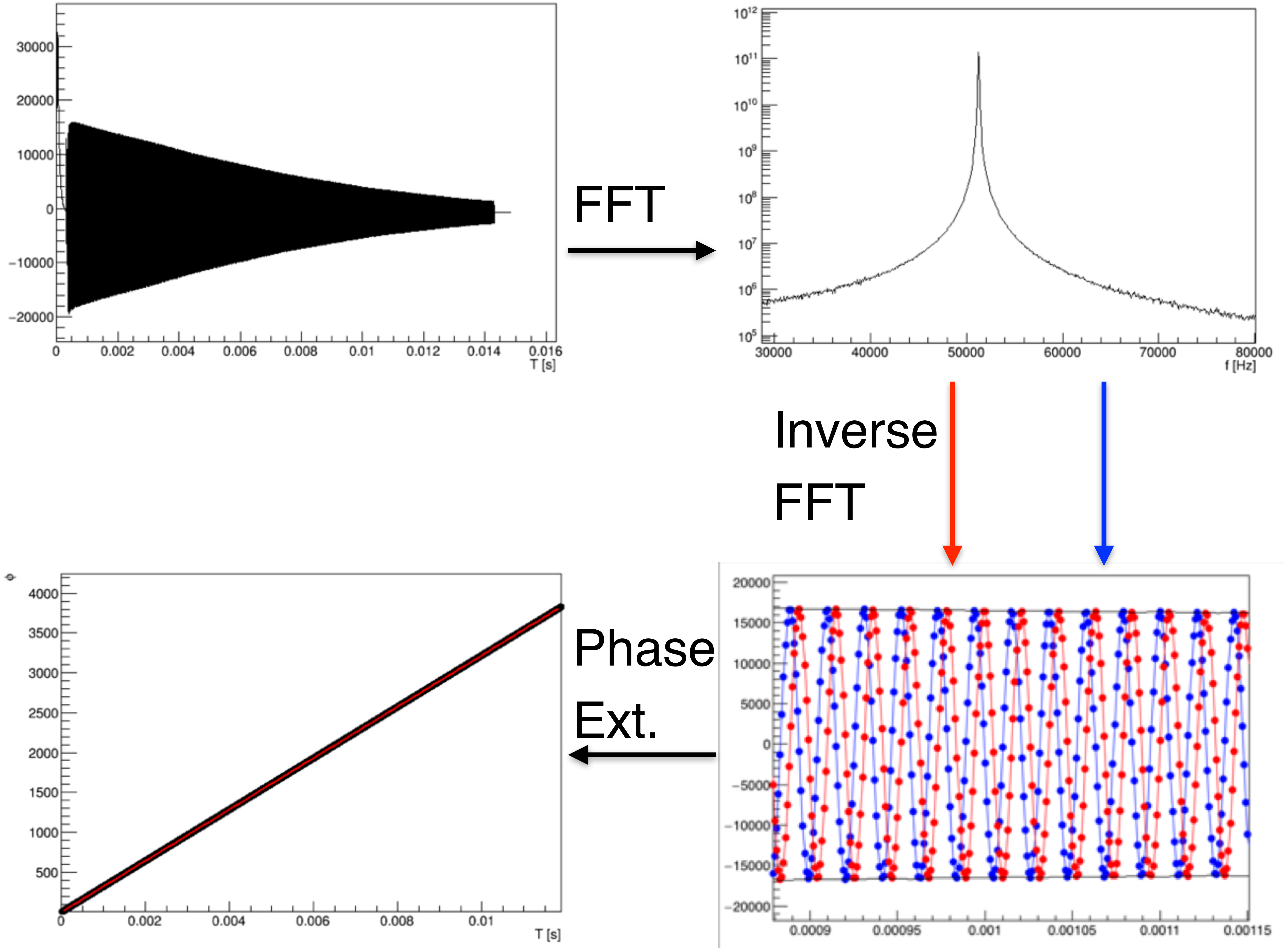


Fourier Transform



Analysis of the NMR signal

- Advanced frequency extraction methods
  - Hilbert transform and phase fit
    - Signal model: A(t)*sin(ωt+ϕ)+baseline
    - Apply Fast Fourier Transform (FFT)
    - Apply low-pass filter to get rid of the non-zero baseline
    - Inverse FFT -> A(t)*sin(ωt+ϕ)
    - Multiply the Fourier image by ***i*** and then inverse FFT -> A(t)*cos(ωt+ϕ)
      - Add in square and take square root: envelope A(t)
      - Divide and calculate $\tan^{-1}$: Phase **ωt+ϕ**

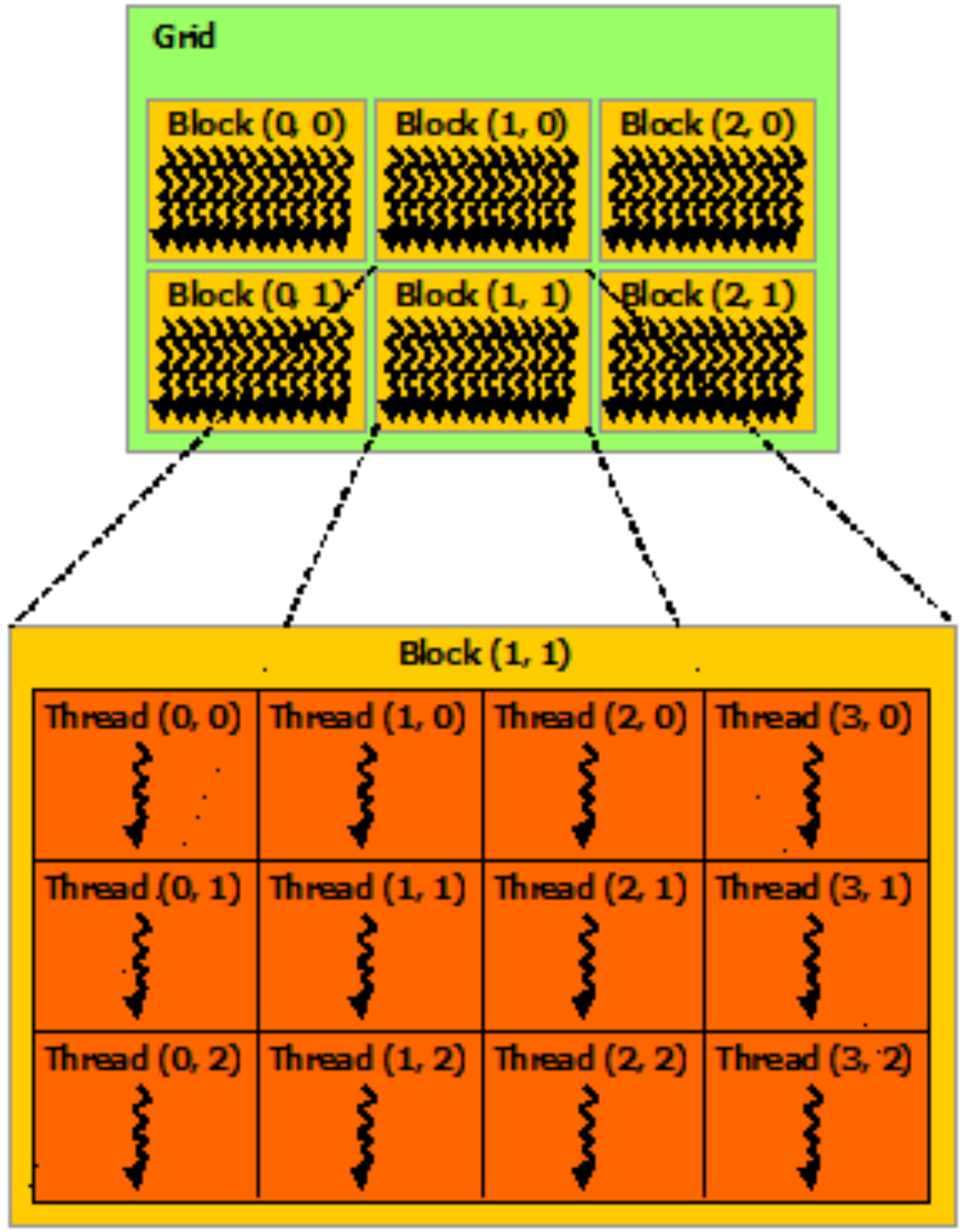


Analysis of the NMR signal

- Hilbert transform method is time consuming
  - 3 FFTs
  - Power spectrum: absolute value of the complex Fourier image
  - Envelope calculation: adding squares and calculate square root
  - Phase extraction: calculating $\tan^{-1}$
  - Phase unwrapping: adding 2π after each cycle
  - Fit range determination: finding the good range (good signal to noise) for fitting
  - Fit the phase function to a polynomial
  - There are **378 NMR pulses** to analyze per measurement cycle
- Goal
  - Analysis time **< 1.4s** (cycle period of the accelerator)
  - Integrating basic algorithms (zero crossing, frequency spectrum centroid, etc)

GPU acceleration for the NMR analysis

- GPU can accelerate the analysis
  - GPU is good at performing large quantity of simple operations in parallel
    - Same operations on each vector component
    - Reduction operations
    - Matrix multiplications, solving linear systems
  - CUDA allows user to define GPU kernels
  - Ready-to-use cuda libraries for FFT, thrust, linear solver, etc.



GPU acceleration for the NMR analysis

- Acceleration and optimization using CUDA (double precision libraries)

	Event Level Parallelization	*Batch Parallelization*	*Solution*
FFT	*Yes*	*Yes*	*cufft library*
Spectrum, Envelope, Phase Extraction	*Yes*	*Yes*	*thrust library*
Phase Unwrapping	*No*	*Yes*	*customized kernel*
Fit Range Finding	*No*	*Yes*	*customized kernel*
Finding Zero Crossings	*Yes*	*Yes*	*customized kernel*
Linear Fit	*Yes*	*No*	*cublas, cusolverDn*

GPU acceleration for the NMR analysis

- Additional considerations for CUDA optimization
  - **Reducing coping data** between the main memory and the graphic memory
  - **Avoid creating objects** (like cufftPlan1d) repeatedly
    - The first event takes significantly longer time than subsequent events due to constructing the utility objects
  - **For small data set, GPU is not necessarily faster than CPU**: setting a batch size threshold to execute GPU-version functions
- Code management
  - Developing CPU and GPU versions of the same function together, and cross check the results. Keep the interfaces the same.
  - Compile analysis functions and classes to shared libraries, and then link them to the DAQ frontend and the off-line analysis.
  - Switching between CPU and GPU versions by environmental variables at the compile time.

Performance comparison

- 1 GPU
- 378 pulses
- 4096 samples per pulse (decimated waveforms for offline analysis)
- 1st event is not included
- No improvement!

	CPU (Xeon E5 1630)	GPU
FFT	235 ms	186 ms
Phase unwrapping	2.82 ms	2.80 ms
Fit range finding	4.80 ms	4.54 ms
Linear fit	186 ms	261 ms
Total	431 ms	469 ms

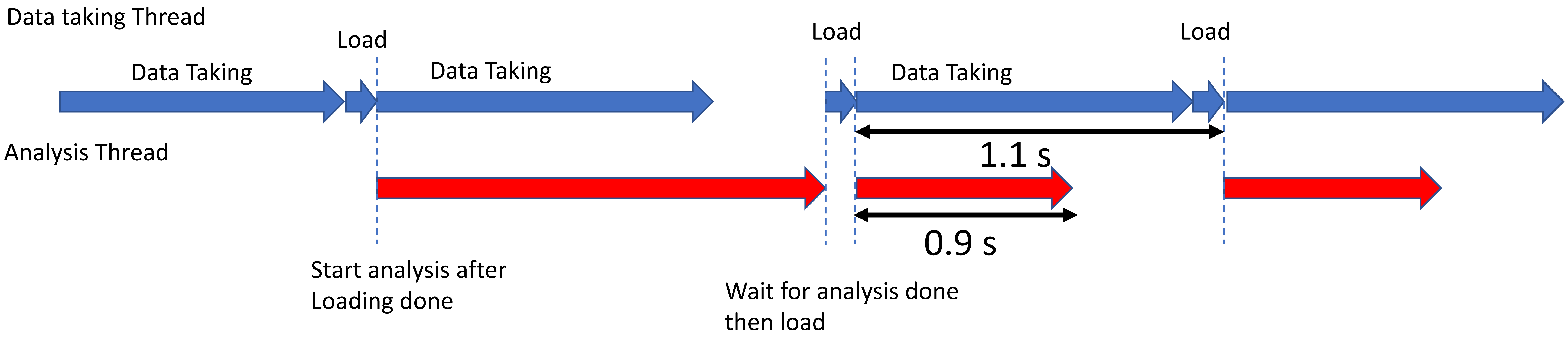
Performance comparison

- 1 GPU
- 378 pulses
- 40960 samples per pulse (raw waveform for online analysis)
- 1st event is not included
- GPU wins! CPU algorithms cannot meet the specification.
- The GPU algorithms can be faster by employing batch matrix operations after CUDA 9.1

	CPU (Xeon E5 1630)	GPU
FFT	2.73 s	338 ms
Phase unwrapping	28.0 ms	24.7 ms
Fit range finding	47.3 ms	42.9 ms
Linear fit	1.14 s	489 ms
Total	3.96 s	940 ms

CPU thread-level optimization

- The data taking takes ~1.1 s
- Data taking and analysis have to be in parallel threads to achieve total data processing time < 1.4 s
- Synchronized parallelism

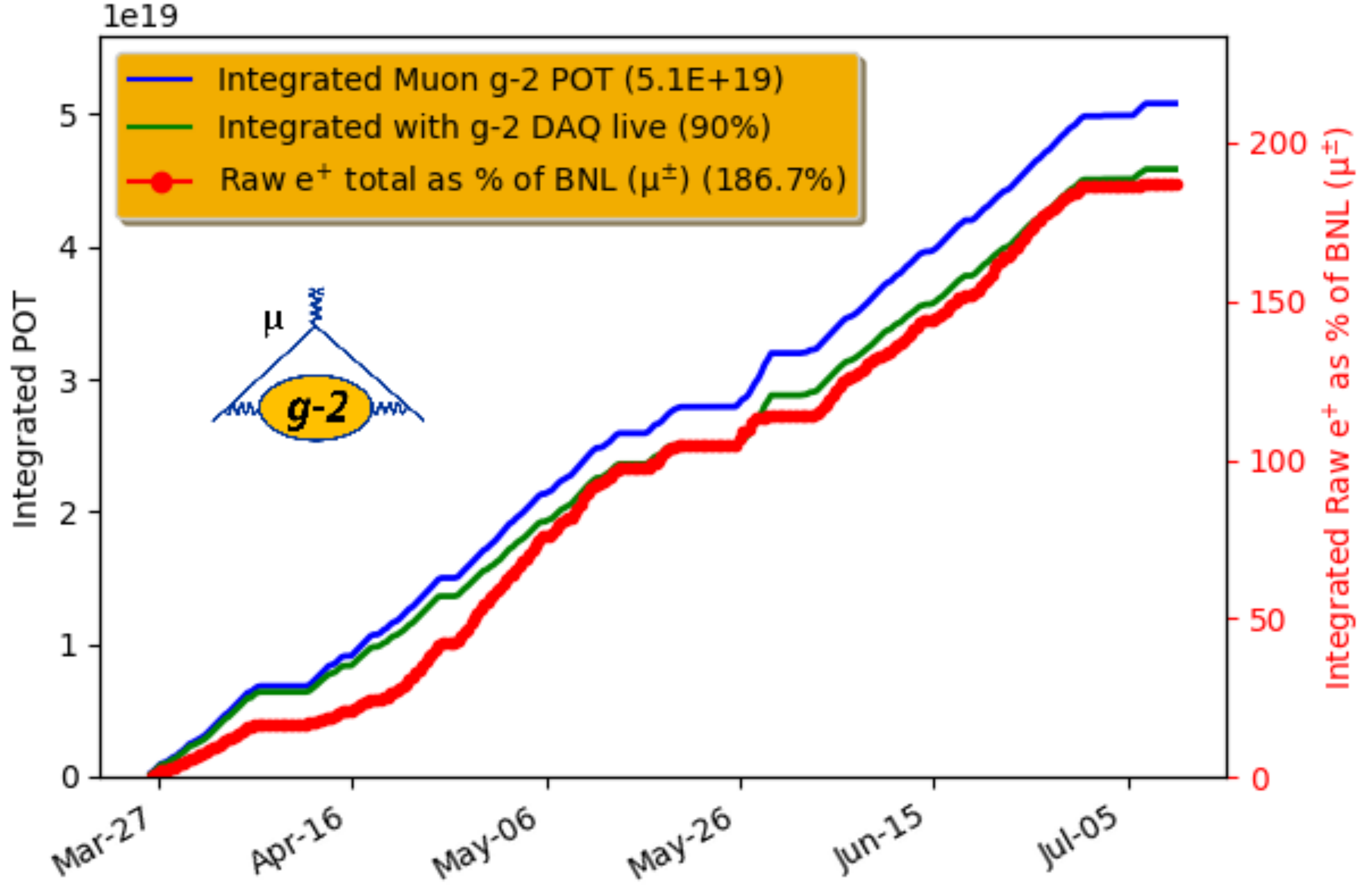


Summary of the magnetic field measurement

- The magnetic field for the muon storage ring in the Muon g-2 experiment is measured by NMR probes.
- Magnetic field is stabilized by a feedback system
- Fast online analysis method is needed to speed up the feedback loop in order to achiever better field stability
- GPU processing speeds up the online frequency extraction by a factor of 4 comparing to pure CPU processing
- With the synchronized parallel scheme, the total processing time (data taking + analysis) for each measurement cycle is 1.1 s. Dominated by data taking time.

Experiment status update

- Run 1 was successfully done in 2018
- Data analysis is still on-going
- Run 2 is commissioning



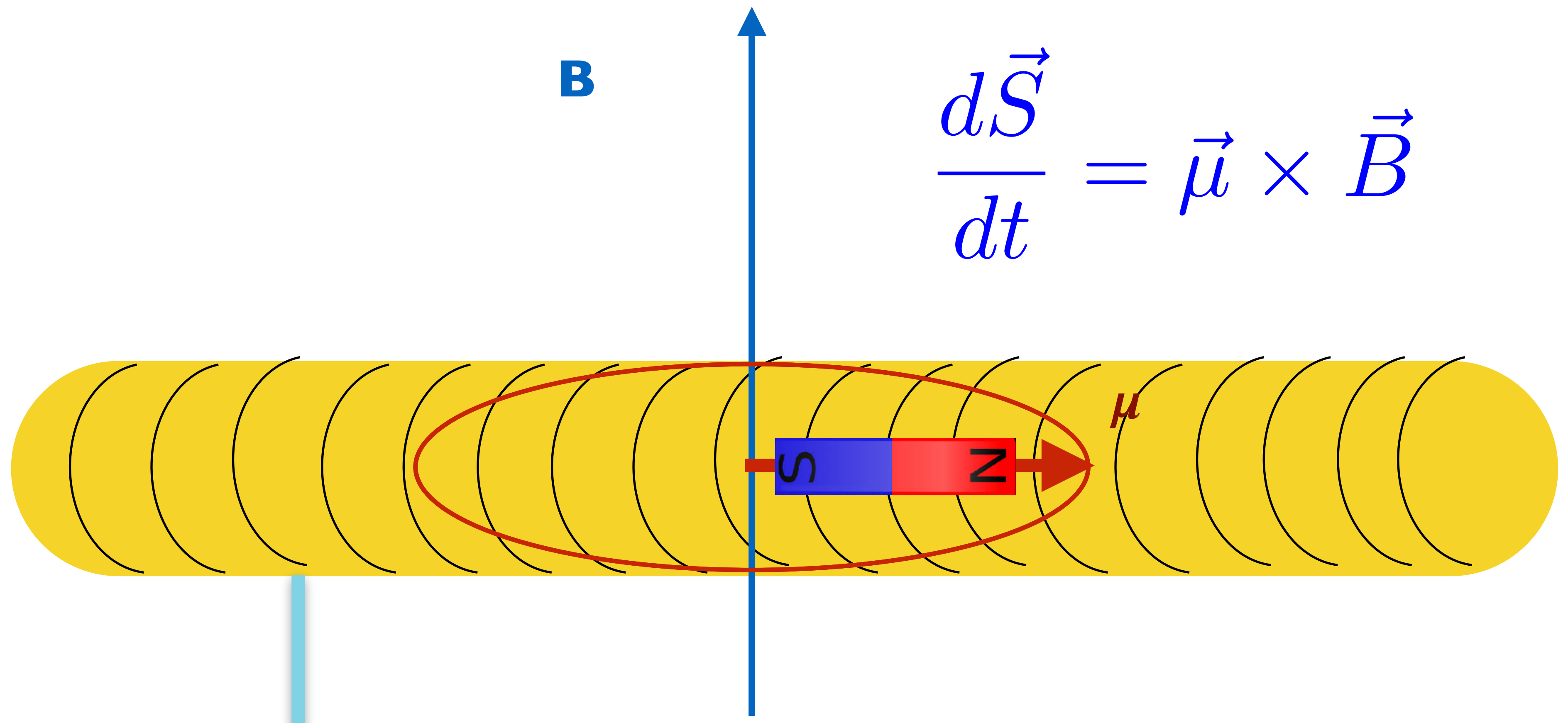
Conclusion

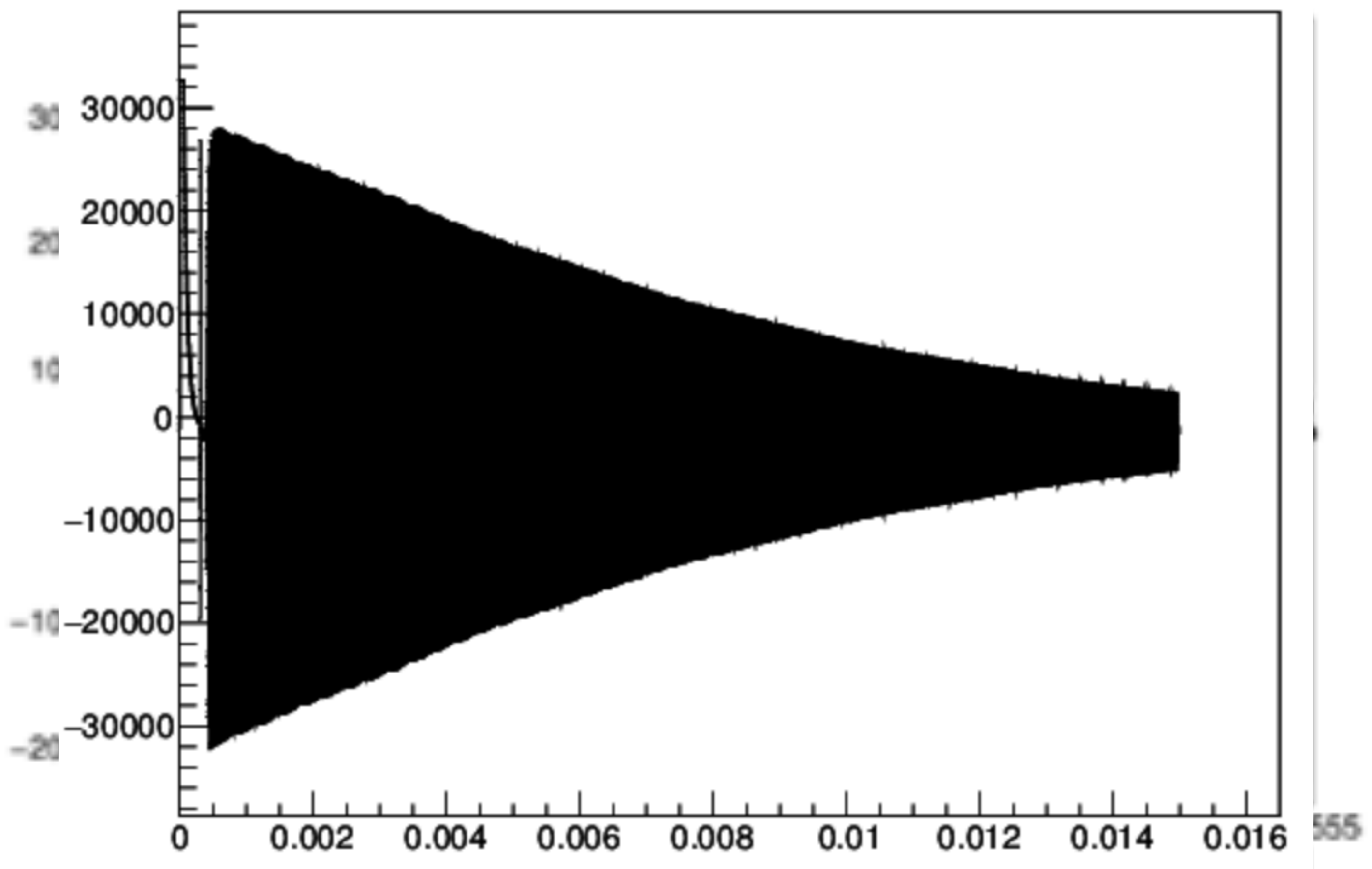
- GPUs are used in the Muon g-2 experiment for data processing and Monte Carlo simulations
  - Data reduction
  - Speed up the online analysis to keep the magnetic field stable

Backup

Measurement principle

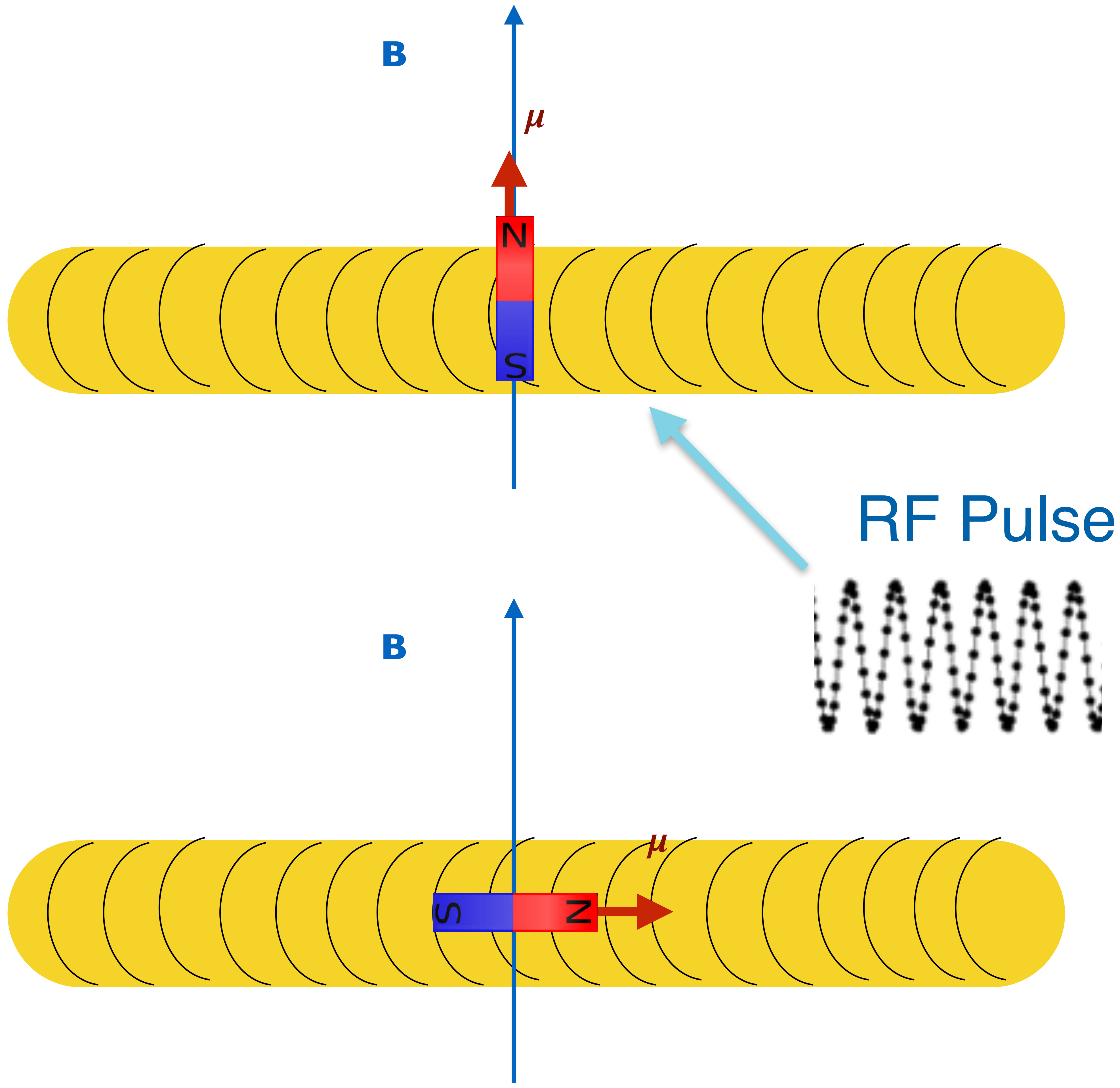
- Nuclear Magnetic Resonance (NMR) measurement principle
  - Protons have a magnetic moment.
  - The motion of proton spin:
    - Precession around the external magnetic field
    - Precession angular frequency ω_p
  - Precession of the proton spin (magnetic moment) creates a rotating magnetic field, then induces electromotive force (EMF) in the pick-up coil
  - EMF is amplified and read out by electronics and then recorded
  - Eventually the protons lose energy and their magnetic moments are re-aligned with the magnetic field — Free Inductor Decay signal (FID)





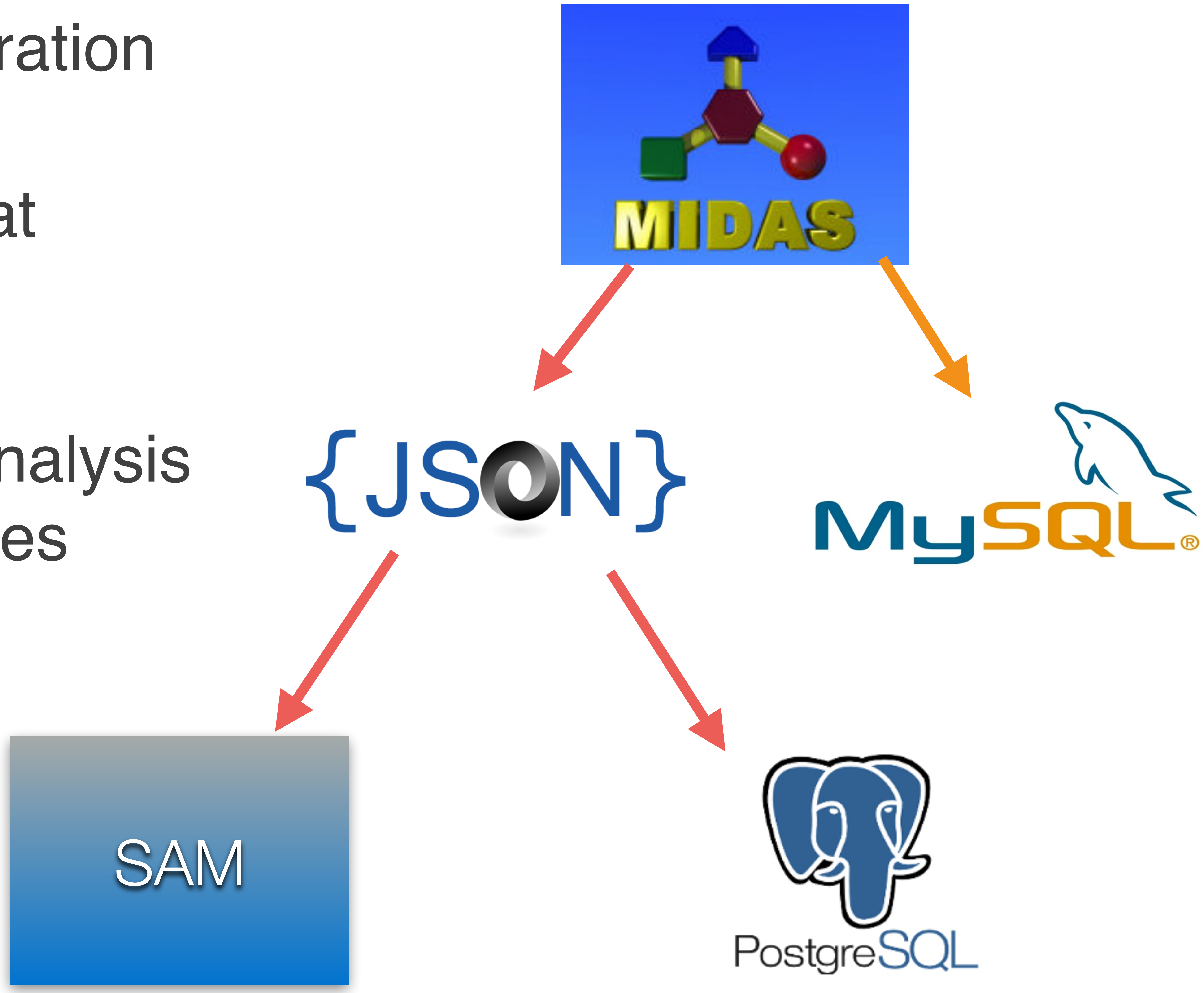
Measurement principle

- Nuclear Magnetic Resonance (NMR) activation principle
  - Protons' magnetic moments are aligned with the magnetic field
  - Inject a short Radio Frequency (RF) pulse with the same angular frequency ω_p through the coil
  - The RF pulse tips the proton spins by 90 degree, with well-tuned amplitude and duration



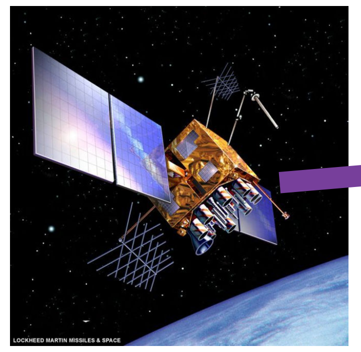
Online State Database Archive

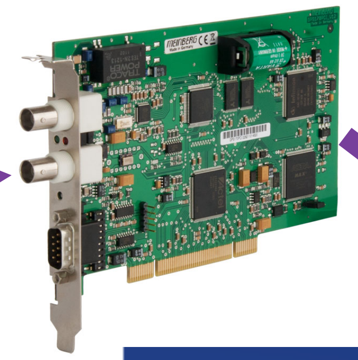
- At each end of run, the DAQ configuration is dumped to a JSON file.
- A python routine is then executed that imports the entire JSON file into a PostgreSQL database.
- Metadata that is used in the offline analysis is also extracted from these JSON files using python plugins.



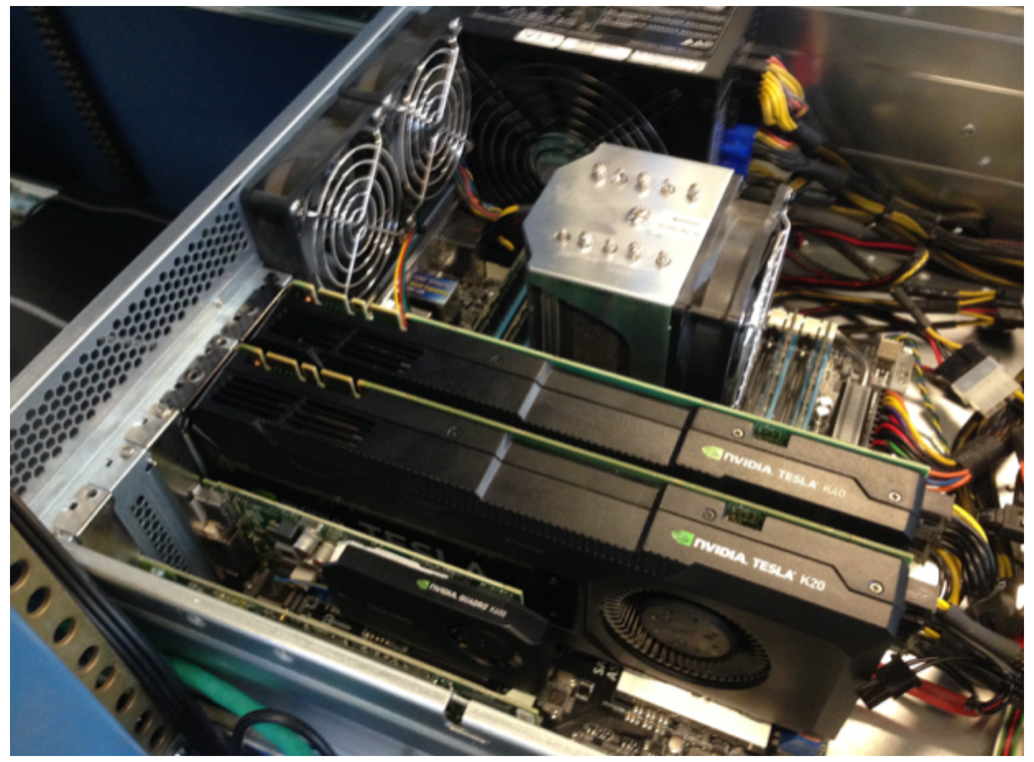
MIDAS Frontends

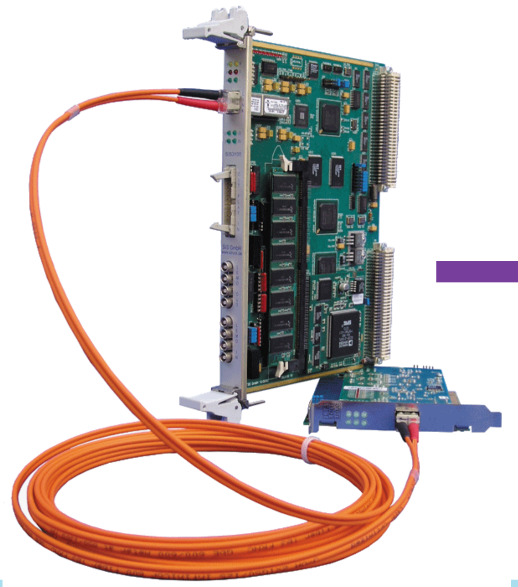
- Master:
  - Communicates with other frontends with RPCs, sets up CCC, and writes GPS timestamps from Meinberg GPS unit.
- AMC13 frontend:
  - Main frontend for processing data from calorimeters, laser, fiber harps, quads, and kickers.
  - Processes the data with Nvidia Tesla K40 GPUs
- Tracker frontend:
  - Data comes from multihit TDCs that are read via FC7 cards.
- IBMS frontend:
  - Data from the inflector beam monitoring system (IBMS) is read out via a CAEN digitizer.

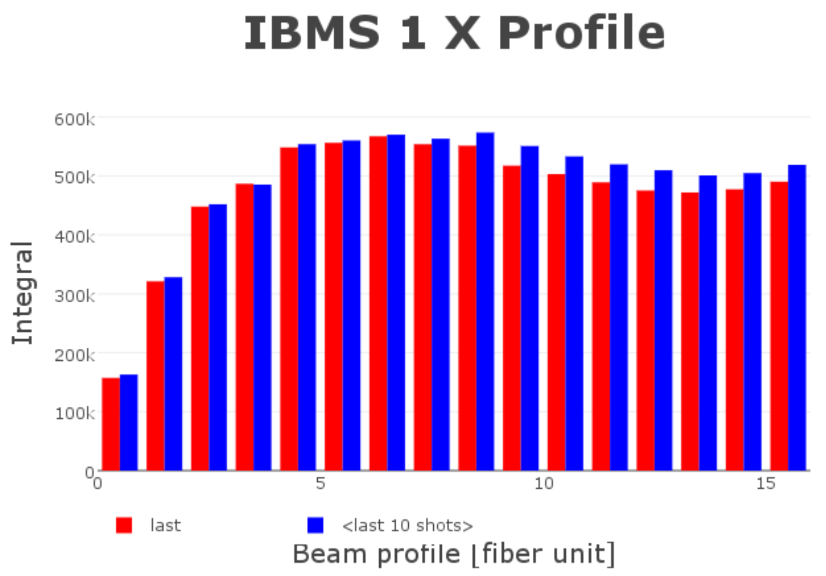




```
Bank:GPS0 Length: 20(I*1)/5(I*4)/5(Type) Type:Unsigned Integer*4
   1-> 0x000018c2 0x580838bc 0x5dcabfb0 0x580838bc 0x5dd48308
```







Contributing Institutions and Responsibilities

Institution	Responsibilities	Personnel
University of Kentucky	AMC13 Readout, GPS, DAQ Hardware, IFIX integration, Online Systems Management	PI + PD (on site) + GS (on site)
University College London	Tracker readout, Online Management	PI (on site) + PD (on site) + GS (on site)
University of Washington	DQM, IBMS readout, Field, Nearline	GS + 2x GS + PD
Northern Illinois University	Slow controls	PI
Argonne National Lab	Field DAQ	PD (on site)
JINR, Dubna	Web interface, Event display	2x Scientist
Novosibirsk	Django Web Display	PD
University of Michigan	Field	GS
Shanghai	Database management	PD
Italy (Frascati + Napoli)	Laser system integration	PI + PD (on site) + PD + GS
Cornell	Auxiliary detector integration, Clock	PD + GS (on site)
Fermilab	Beamline integration	Scientist

~24 total

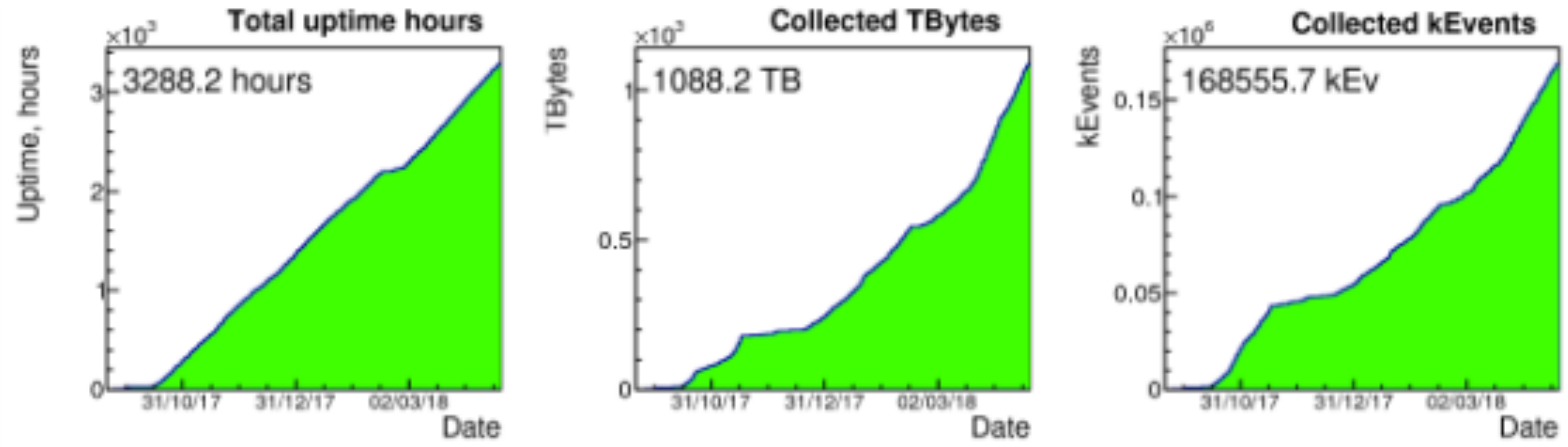
RED = DAQ on-call

File Handling

- Writing 100 TB of data to tape per week.
- 70 TB local RAID keeps copy of data until it receives a tape label.
- File transfers are handled with Fermi File Transfer Service and catalogued with SAM.

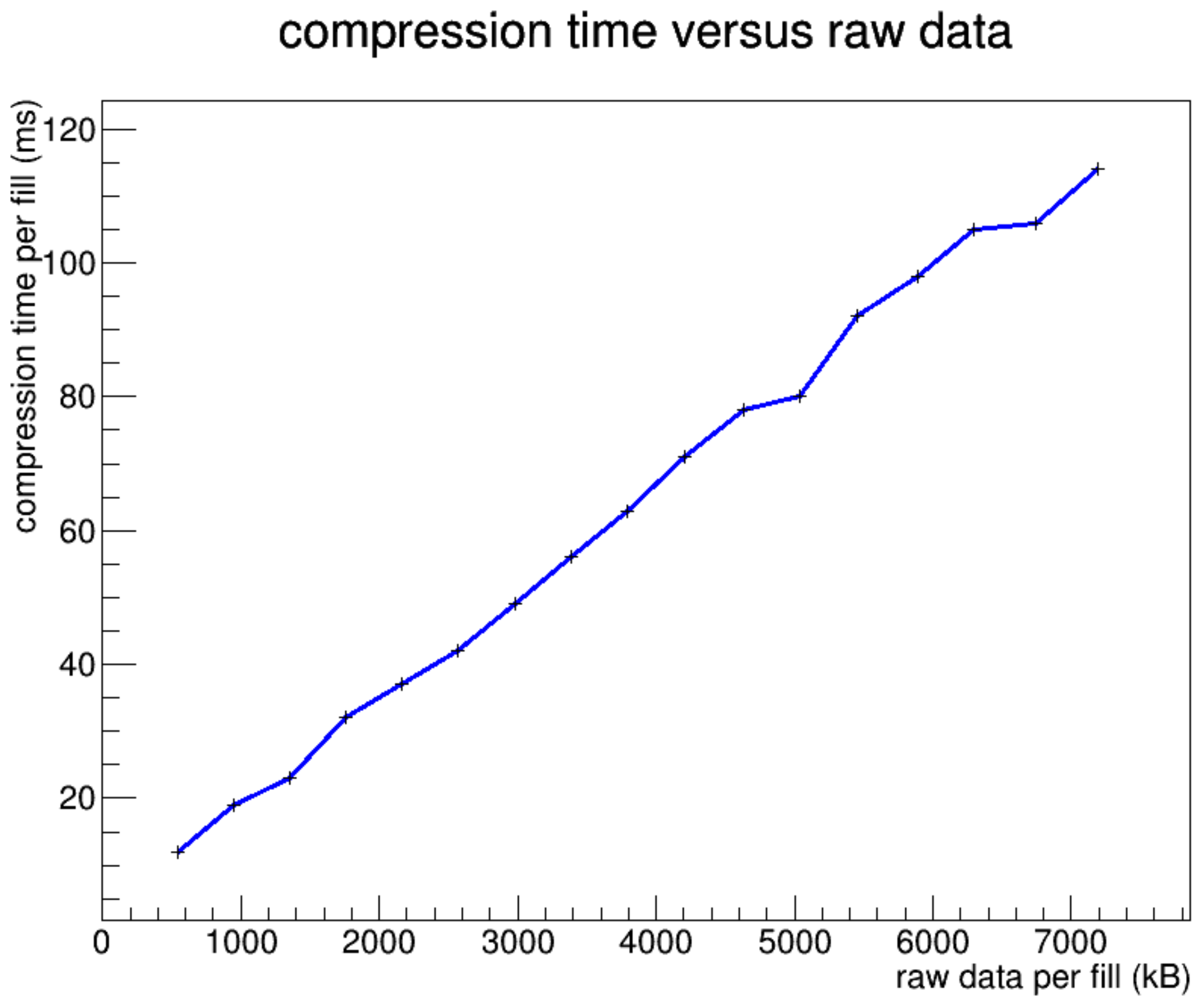
DAQ Performance

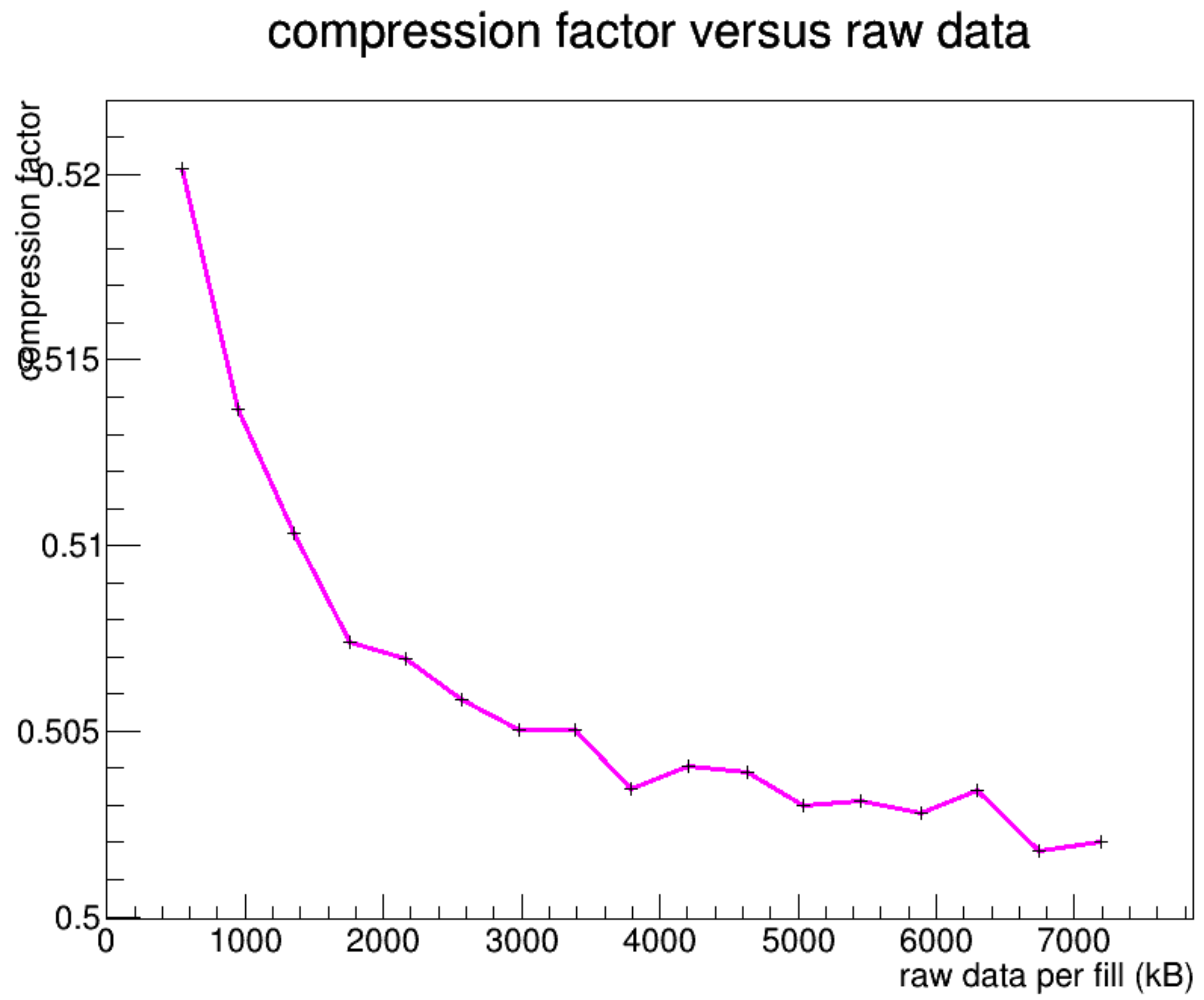
- DAQ is currently running well and writing 200 MB/s to tape.
- Uptime averages better than 90%.
- During past 11 months only two hardware outages that were handled by hot spares.



Data Compression

- We are compressing the data using the zlib libraries in each of the frontends.
- The frontend compresses all banks into one compressed bank, which is unpacked later in an art input module.
- We achieve a factor of 2 compression without any significant impa





Input sources

- Digitization is performed in custom uTCA based waveform digitizers.
- Each digitizer runs at 800 MSPS, so each time bin is 1.25 ns, and a 700 us fill is 560,000 clock ticks.
- Each uTCA crate contains 12 WFD5s or 60 channels of digitization.
  - Crate 0 reads data from the clock and control center (CCC)
  - Crates 1-24 each read data from one calorimeter (+ spare channels)
  - Crate 25 reads data from the laser system
  - Crate 26 reads data from the Auxiliary detectors (Harps, Quads, and Kickers)
  - Crate 27 reads data from the three tracker detectors.
- Data from each crate is sent to a DAQ computer via a dedicated 10 Gb fiber. The total data rate is 20 GB/s.
- The data is then processed in Nvidia K40 GPUs.

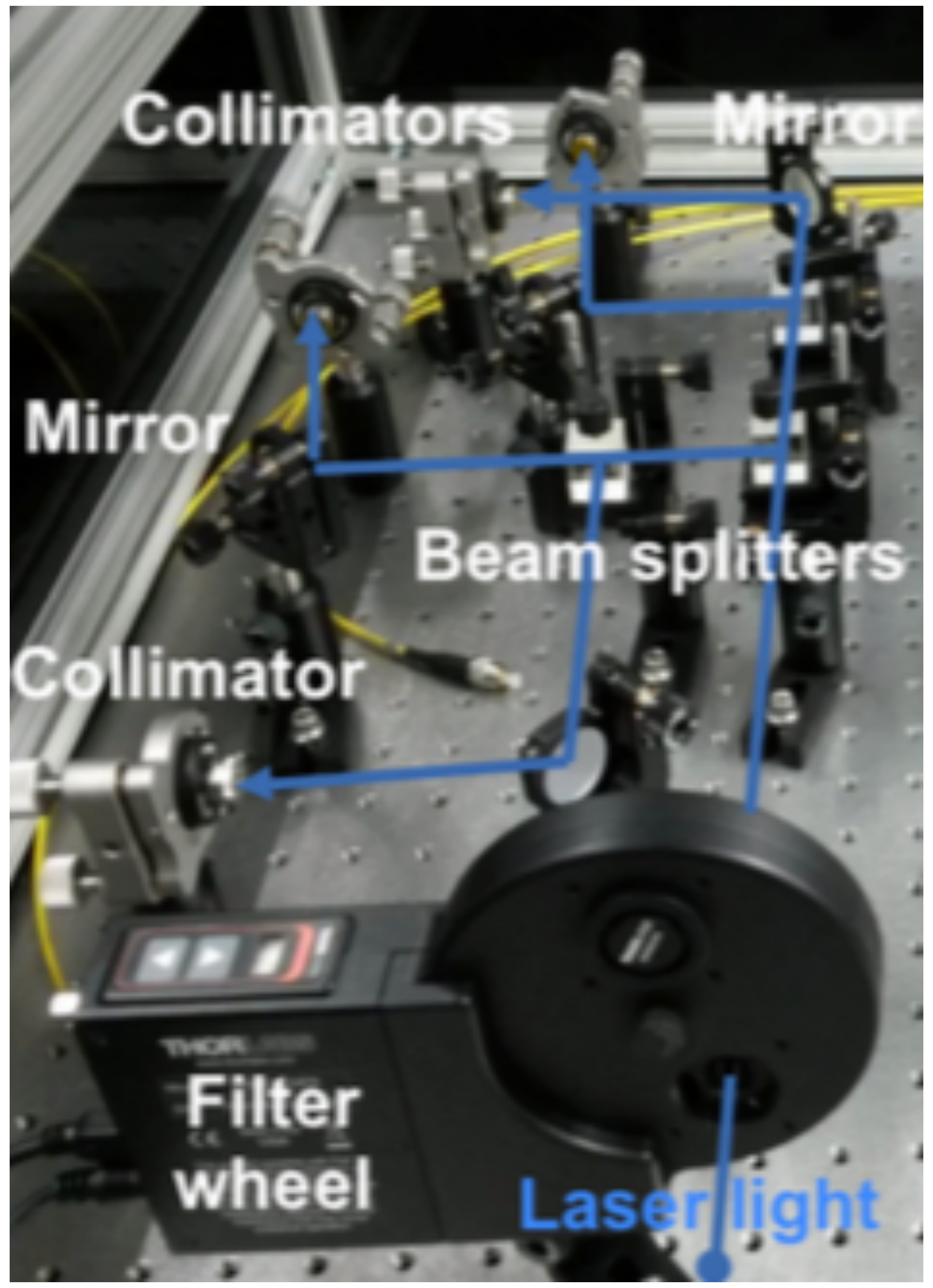
g-2 Detector Systems

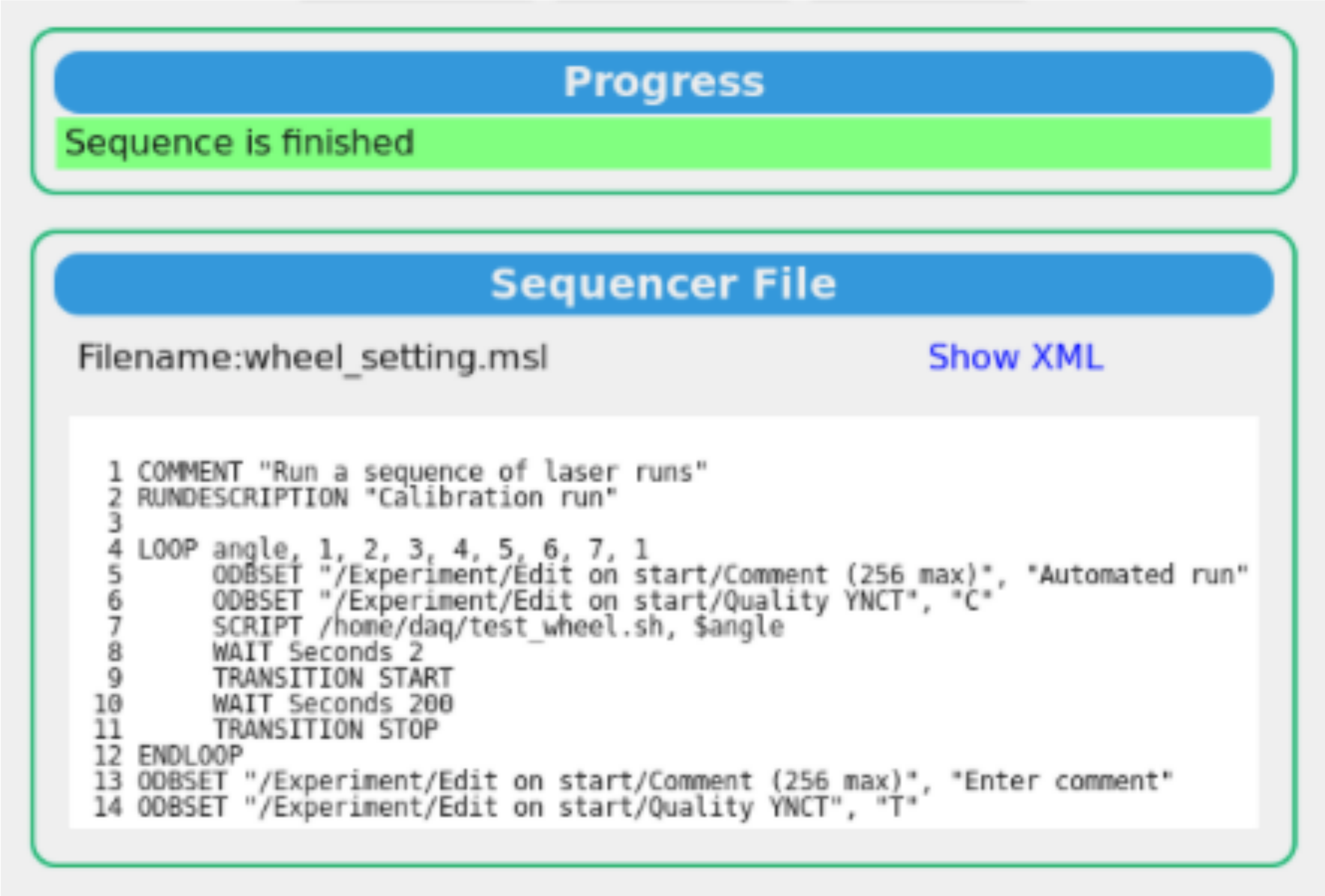
- 24 Calorimeters
  - 1 uTCA crate for each calorimeter
  - 54 channels * 24 calos = 1296 channels of digitized data.
  - Data provided by 12 Cornell waveform digitizers.
- 4 Fiber Harps
  - 7 channels * 4 harps = 28 channels
  - Data provided by Cornell waveform digitizers
- Quads and Kickers
  - Write 4 quad channels and 15 kicker channels
  - Data provided by Cornell waveform digitizers
- 3 Trackers
  - Data from Multihit TDCs sent from FC7s in a uTCA crate
- IBMS and quads
  - Running on CAEN digitizers
- Slow control SCS3000 devices

MIDAS Sequencer

- The sequencer was used extensively for calibration runs such as filter wheel scans and bias voltage scans.

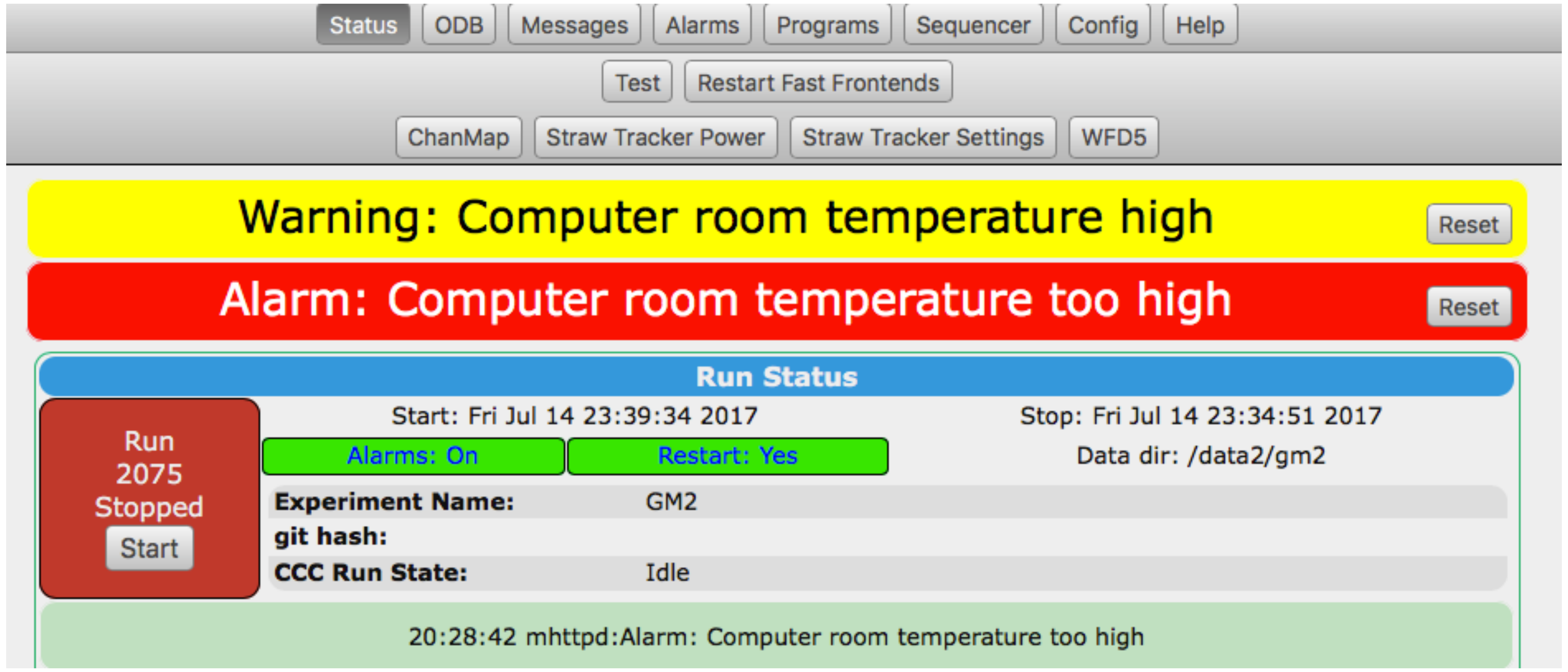
- A typical sequence would be:
  - Execute script to move wheel.
  - Update ODB values
  - Take data for 10 minutes
  - Repeat





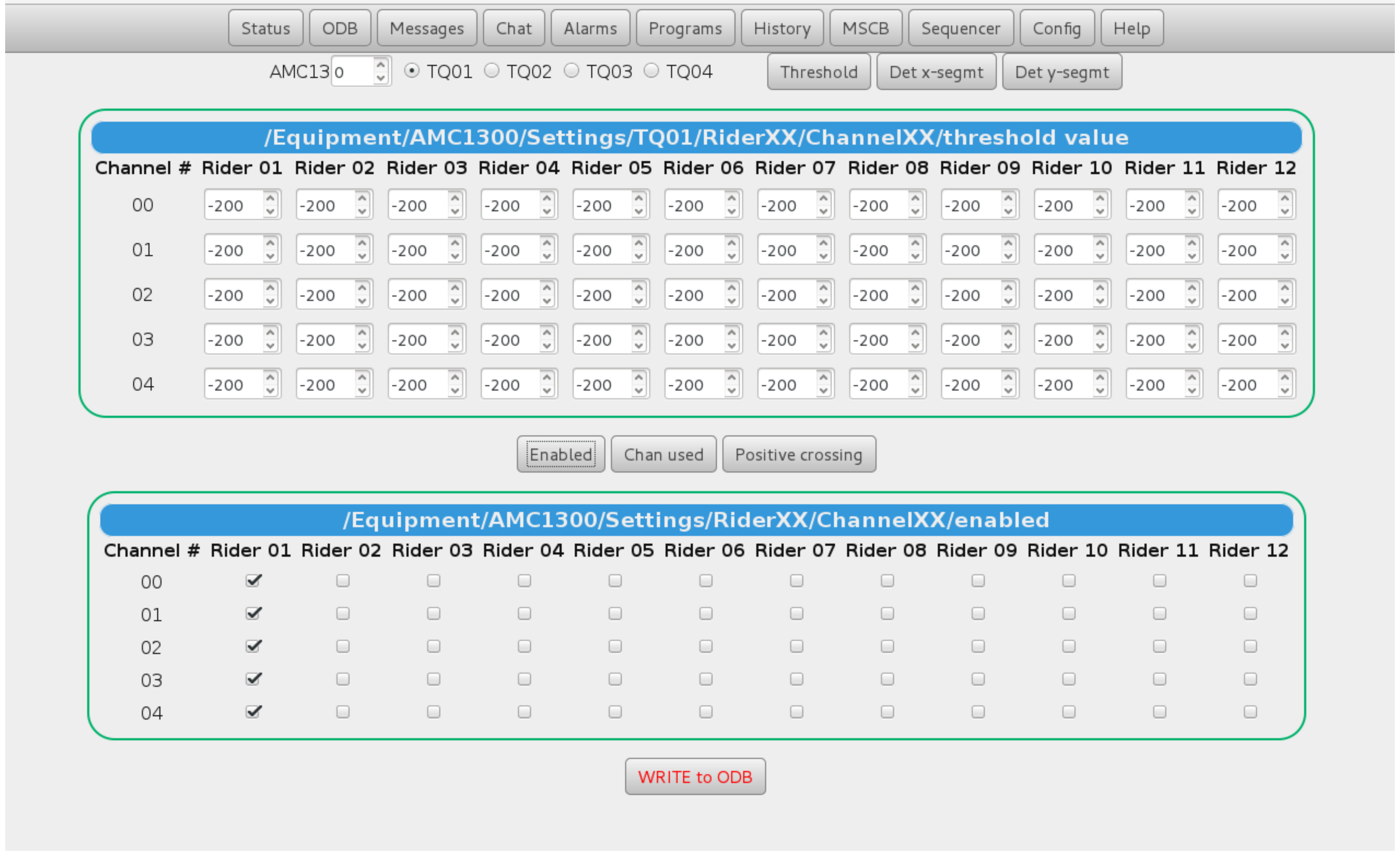
MIDAS Alarms

- The MIDAS alarm system was used as the primary alarm system.
- Alarms were set on temperatures and voltages from MSCB devices.
- Other slow frontends set alarms automatically when encountering an error.
- Periodic alarm reminded shifters to perform shift checks.
- Had problems at first with alarm audio, which was traced to a recent lack of mp3 support in scientific linux — this was later rectified with a recent update.



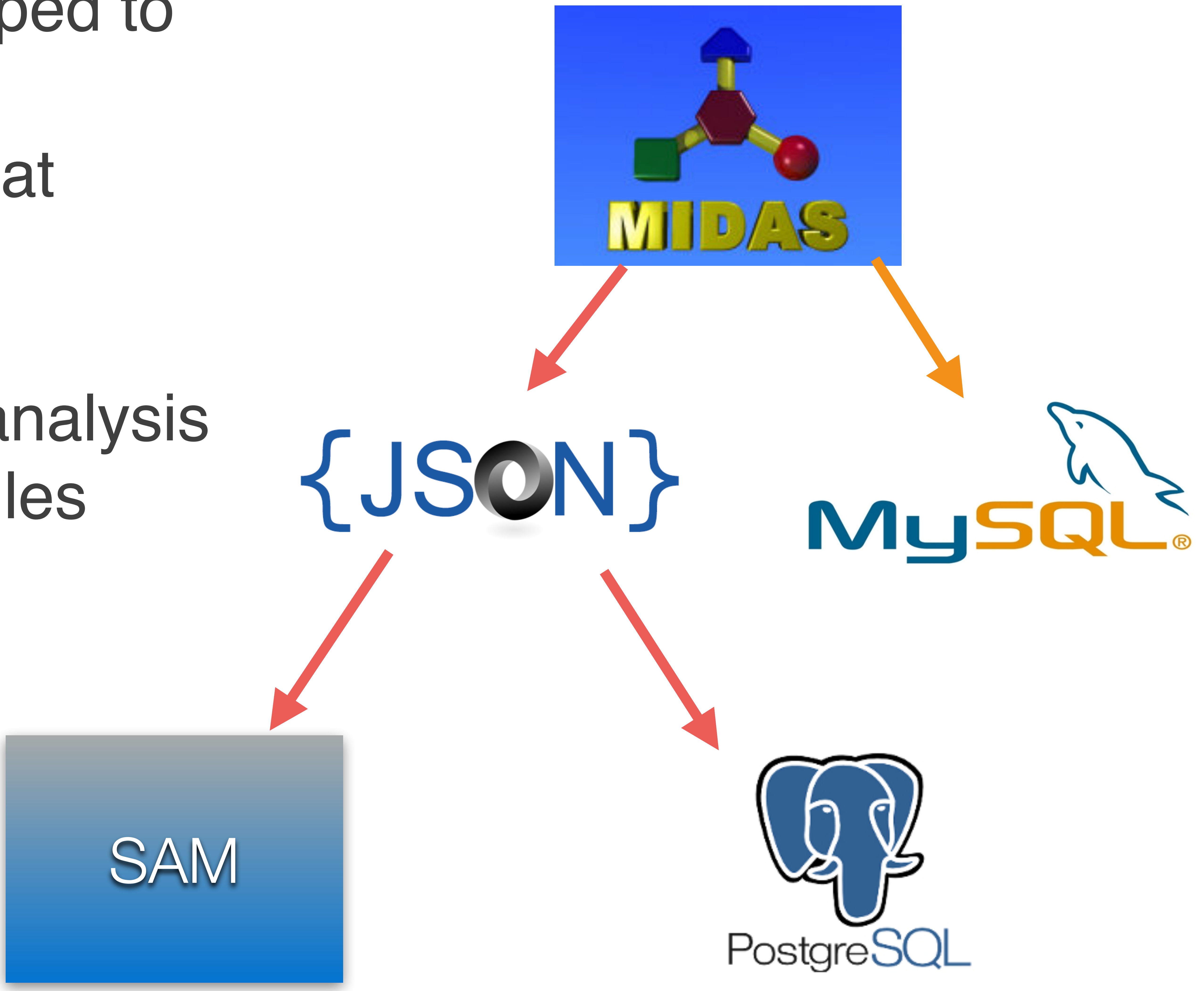
Custom Controls Page

- With this number of frontends, configuring settings via the standard ODB tree is very cumbersome.
- A set of custom Javascript pages were written to manipulate ODB values en masse.



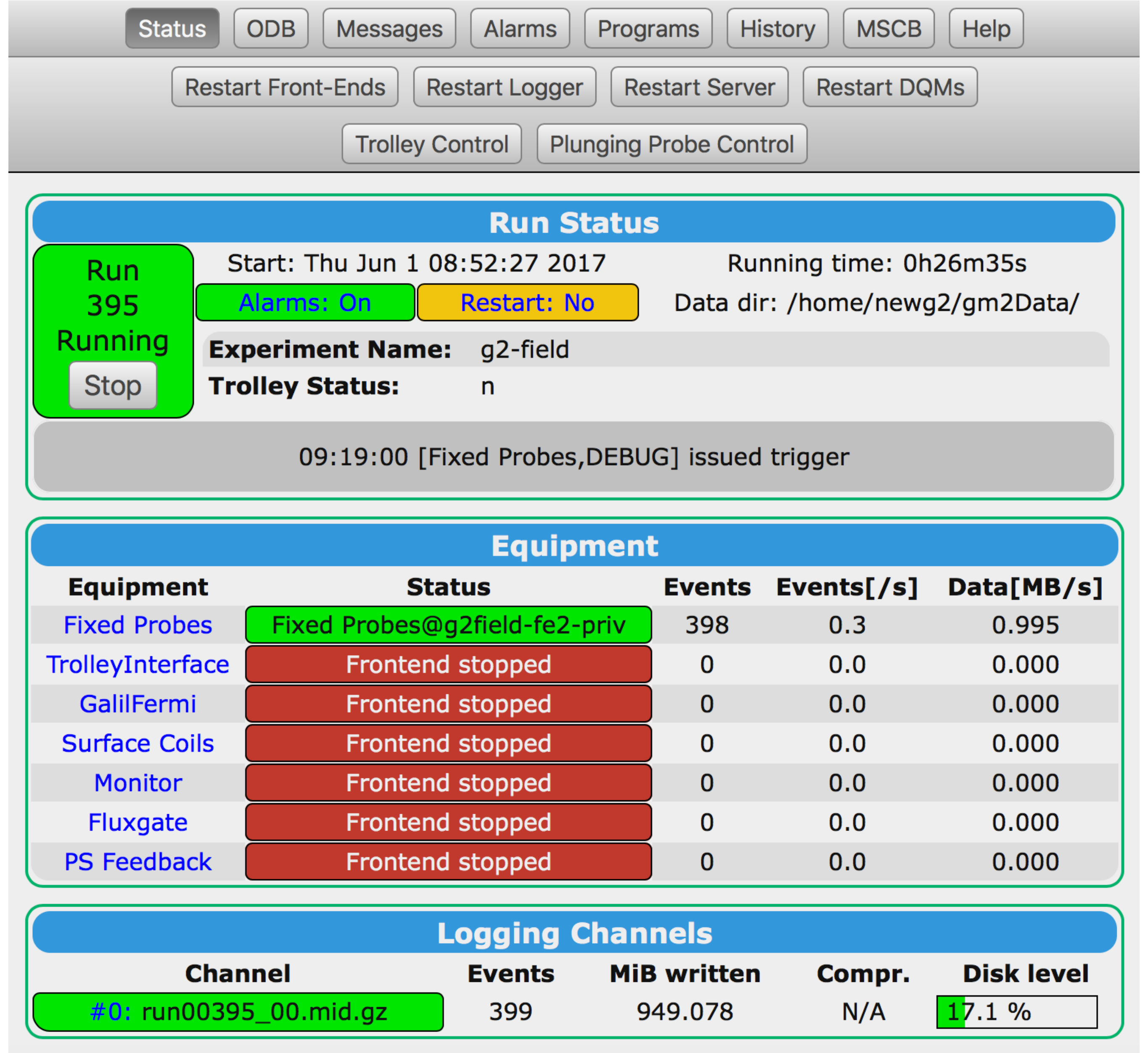
MIDAS ODB Archive

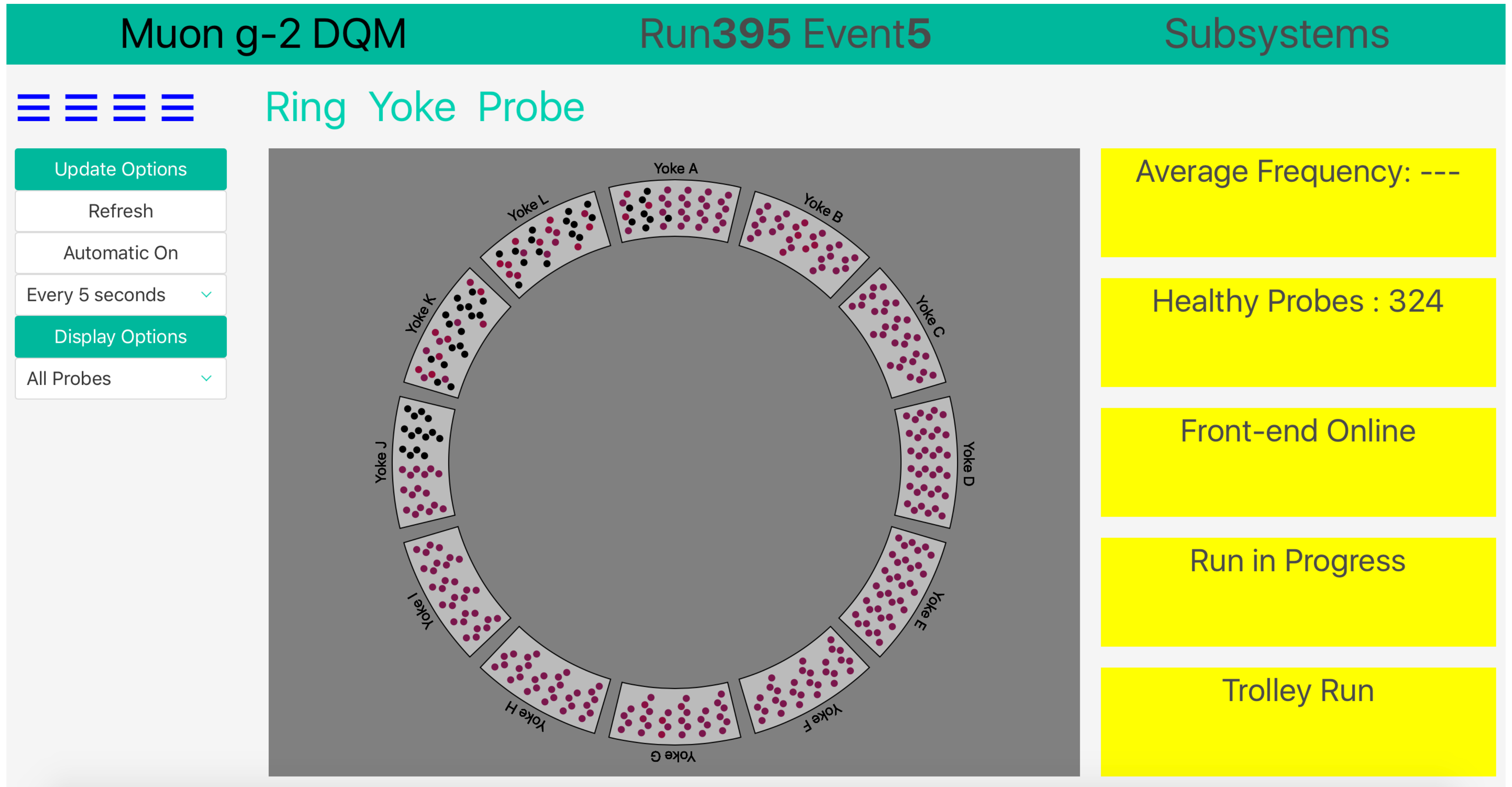
- At each end of run, the ODB is dumped to a JSON file.
- A python routine is then executed that imports the entire JSON file into a PostgreSQL database.
- Metadata that is used in the offline analysis is also extracted from these JSON files using python plugins.



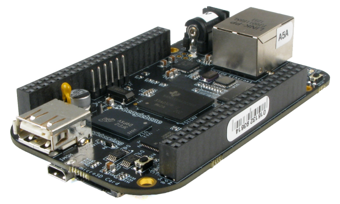
Field DAQ

- Field DAQ runs in independent MIDAS experiment.
- Contains seven asynchronous frontends reading data from fixed magnetic field probes and from a trolley that periodically transverses the ring to perform precision measurements of magnetic field.
- Data is correlated with the fast DAQ offline using GPS timestamps.





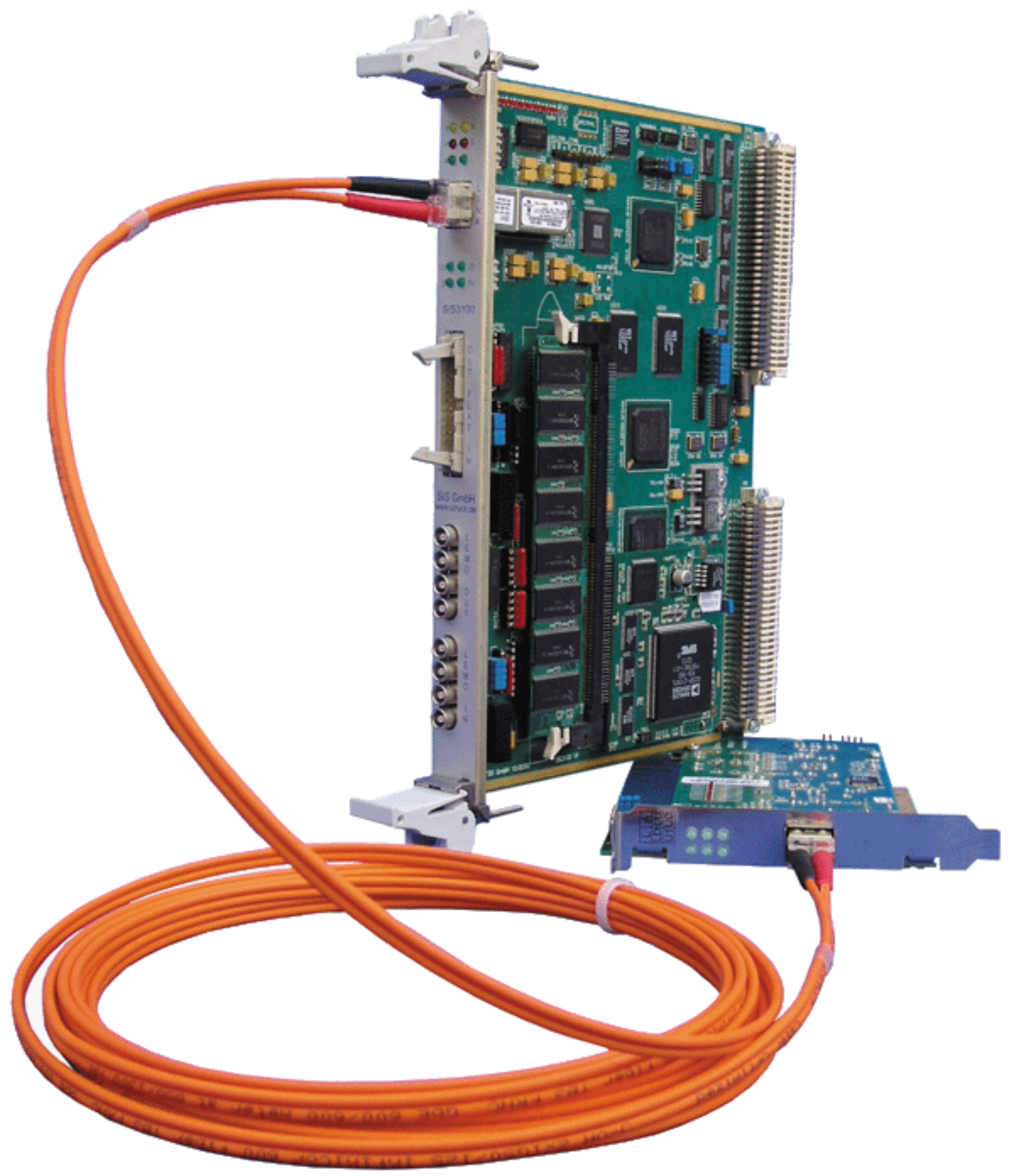
Slow Controls

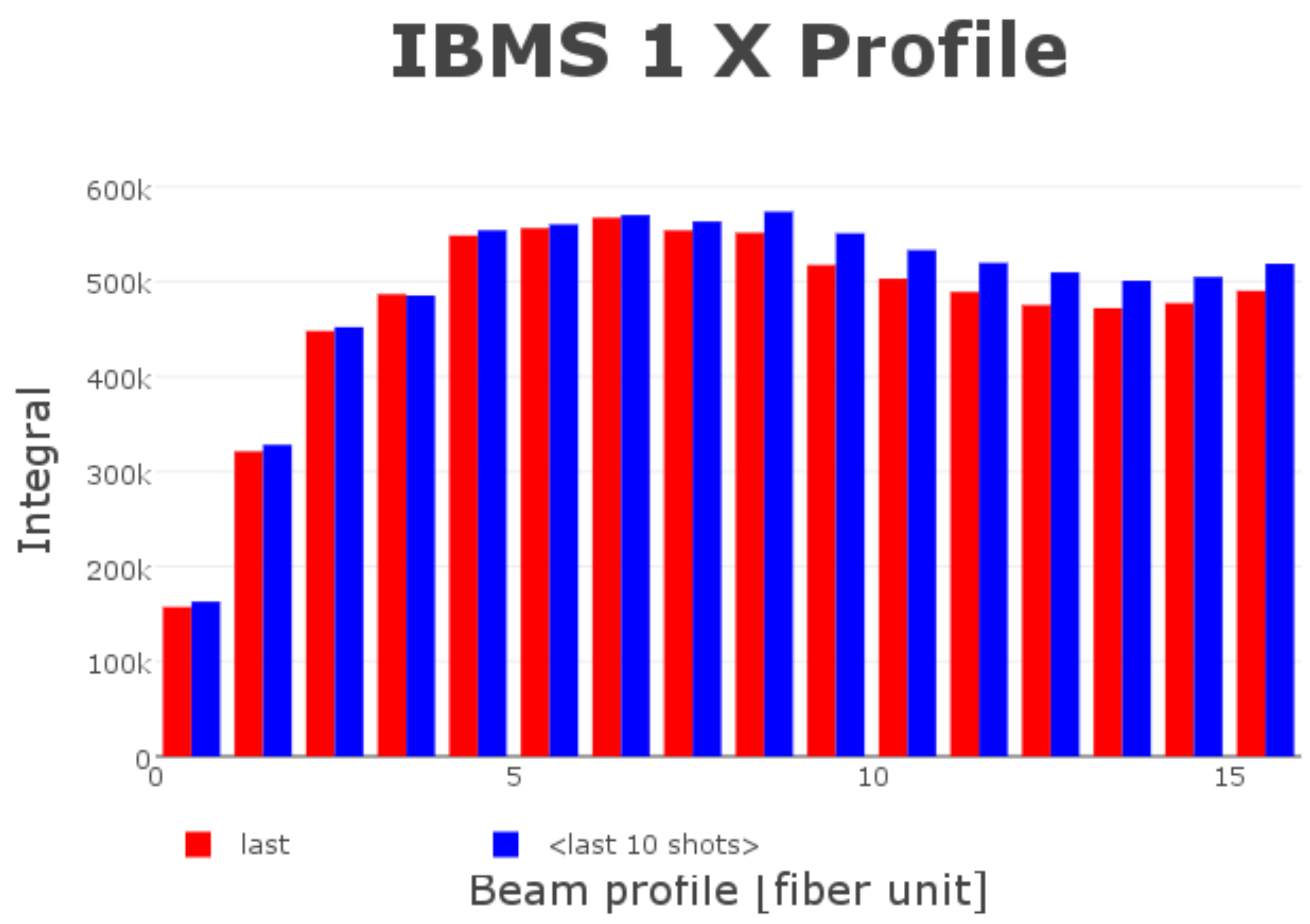


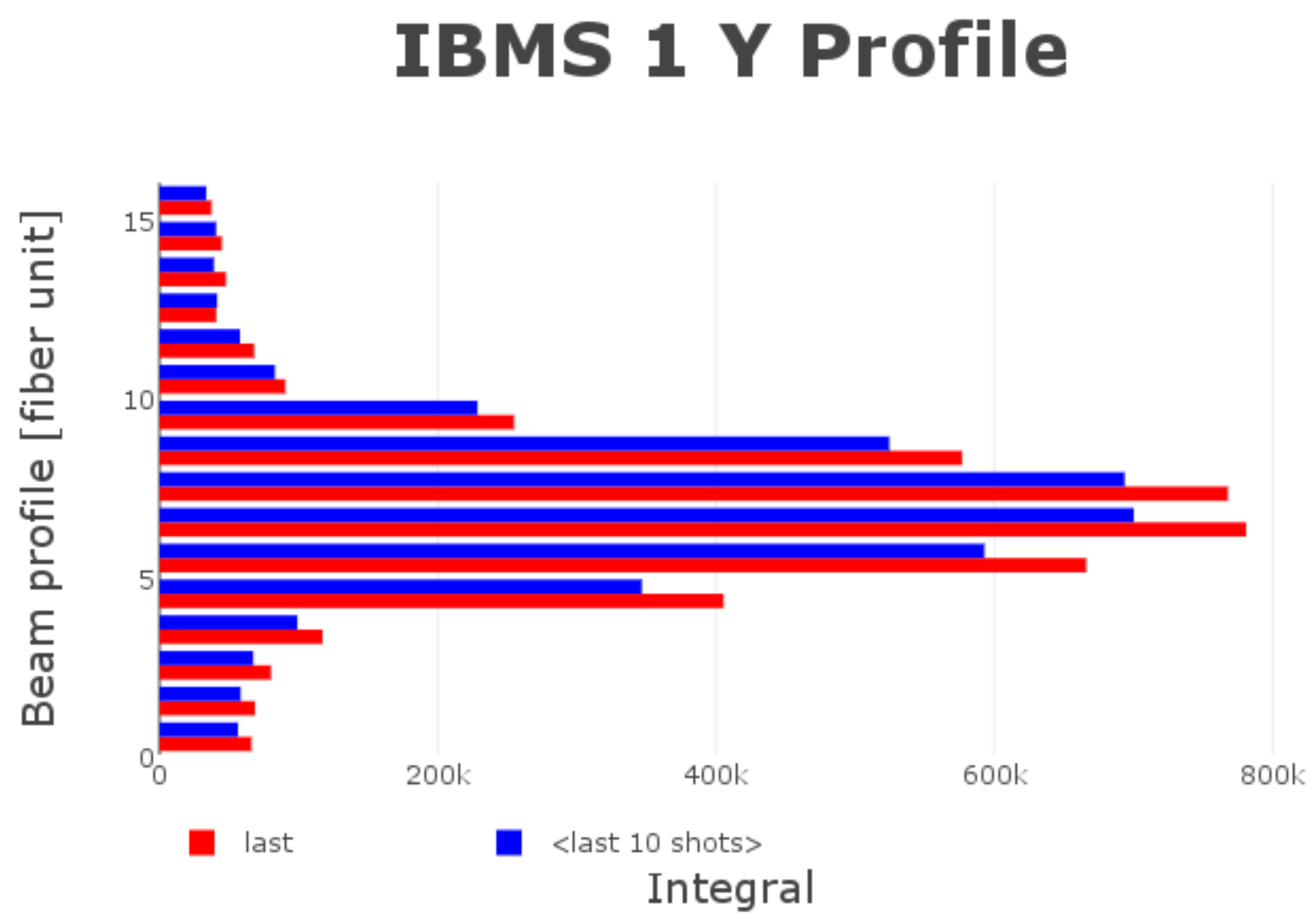
- DAQ includes six SCS3000 mscb devices.
- 24 beaglebones reading slow control data from calorimeters.
- HV and LV frontends for tracker system.
- Slow frontend reading magnet properties from IFIX via an OPC client.
- Beamline frontend periodically reading output of beam components from database.
- Slow control data is stored in a Postgres database and displayed using a custom Django web display.

IBMS Frontend

- Data from the inflector beam monitoring system (IBMS) is read out via a CAEN digitizer.
- A custom MIDAS frontend was written to integrate this detector into the DAQ.

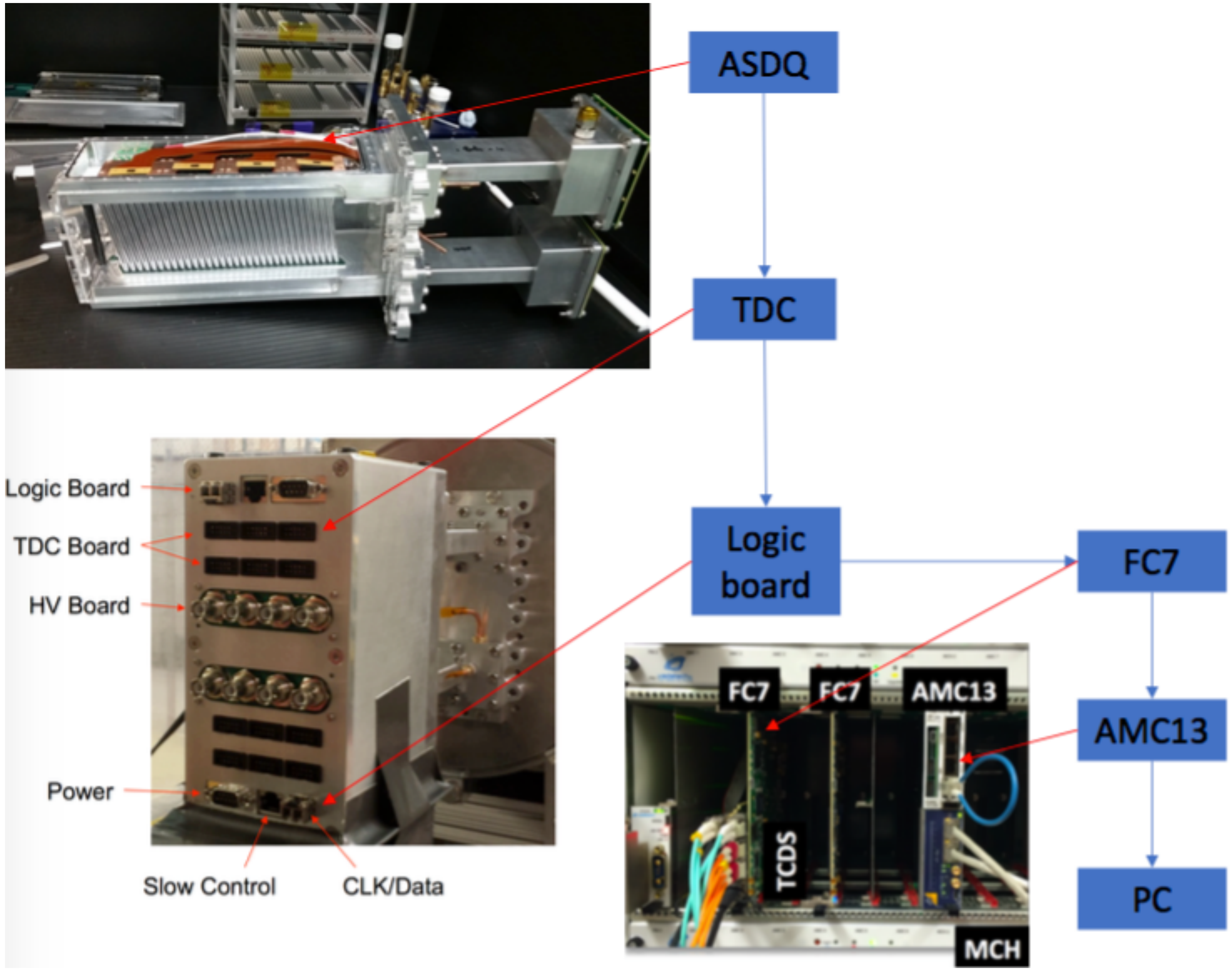






Tracker Frontend

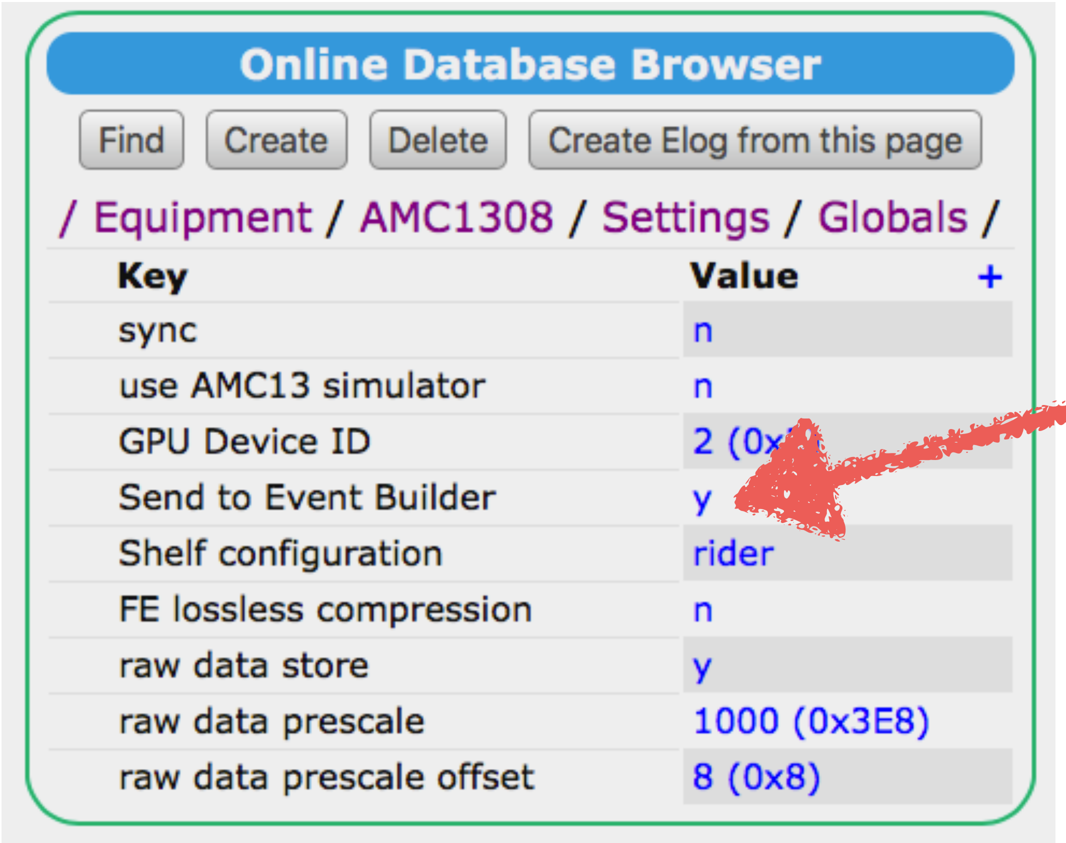
- Three tracker stations will be read via one uTCA crate.
- Reads data from AMC13.
- Instead of digitizers, data comes from multihit TDCs that are read via FC7 cards.



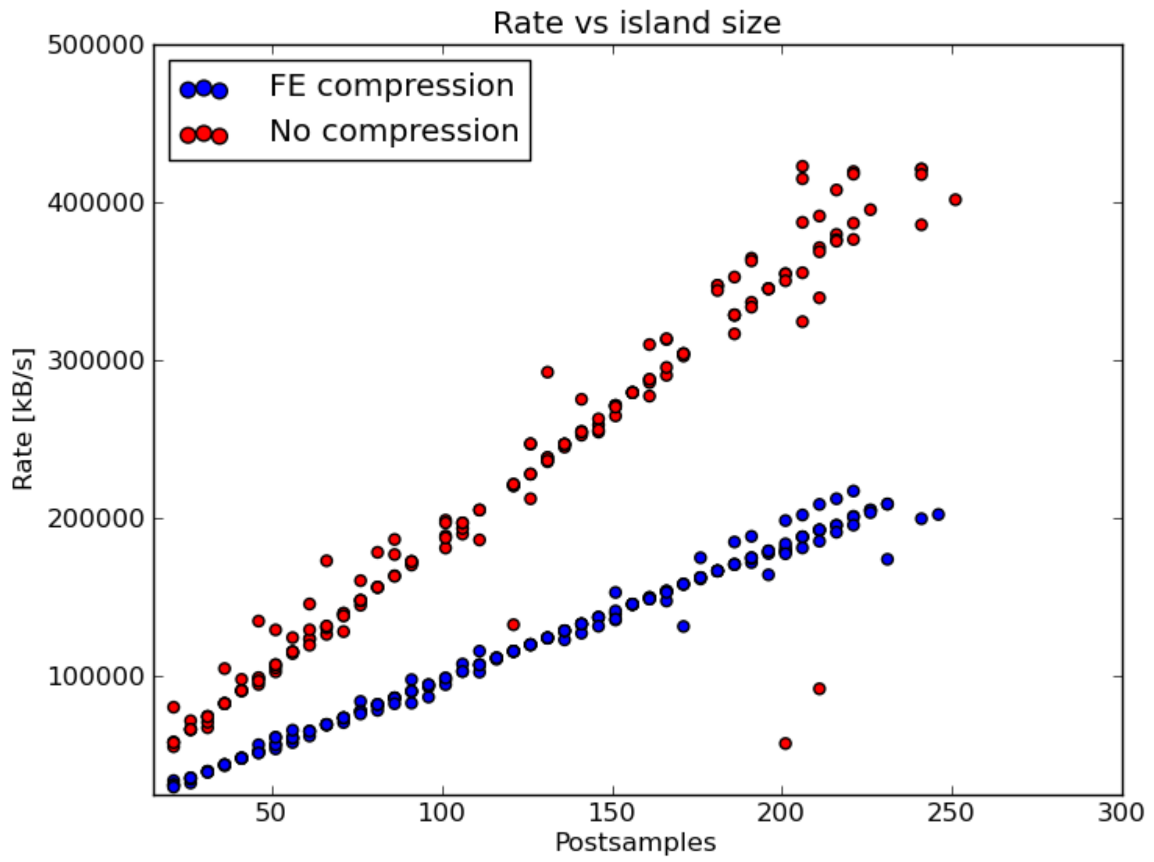
(Thanks R. Chislett for the diagram)

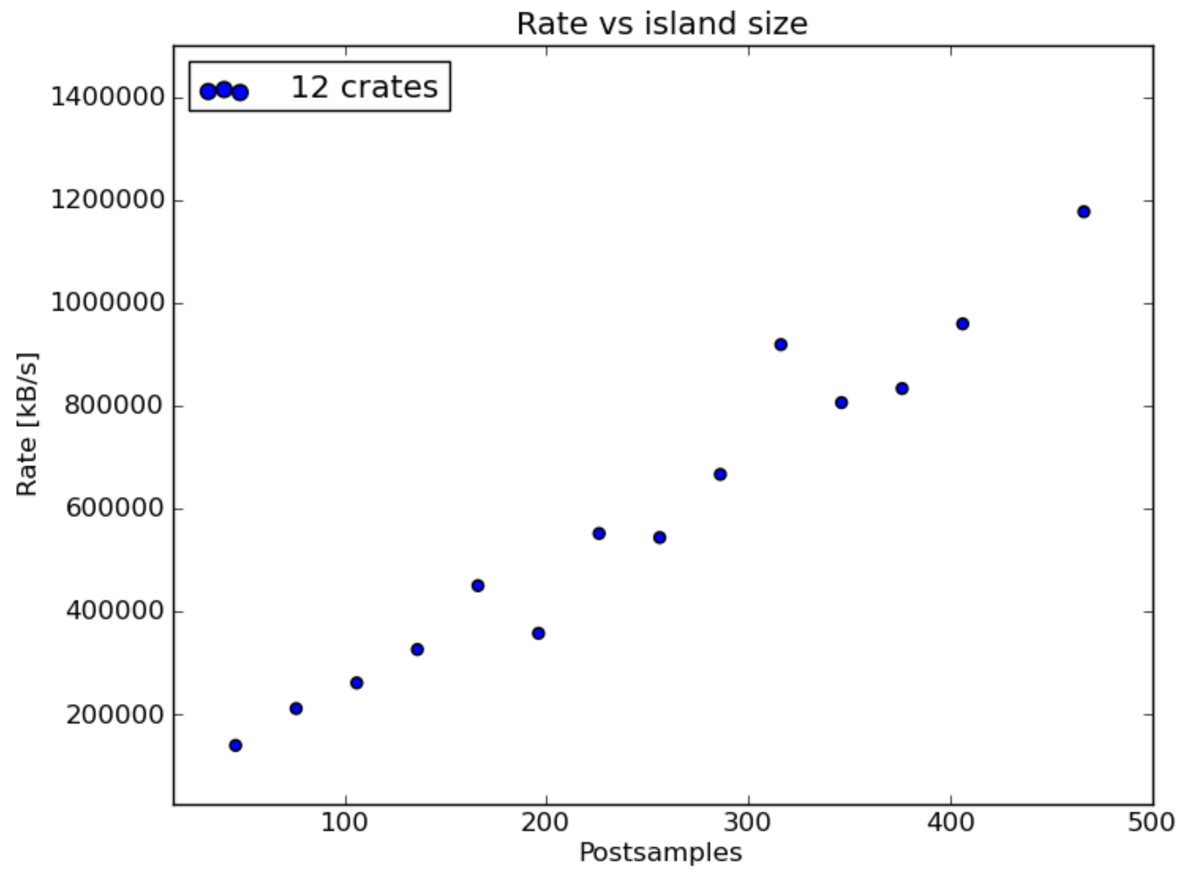
Event builder

- Modified event builder to change how it is enabled — now done entirely in ODB Equipment of each frontend.



Enable here!





- Total EB rate maxed out at > 1.2 GB/s (limited by network bandwidth)
- The event builder combines up to 270 banks for each event.

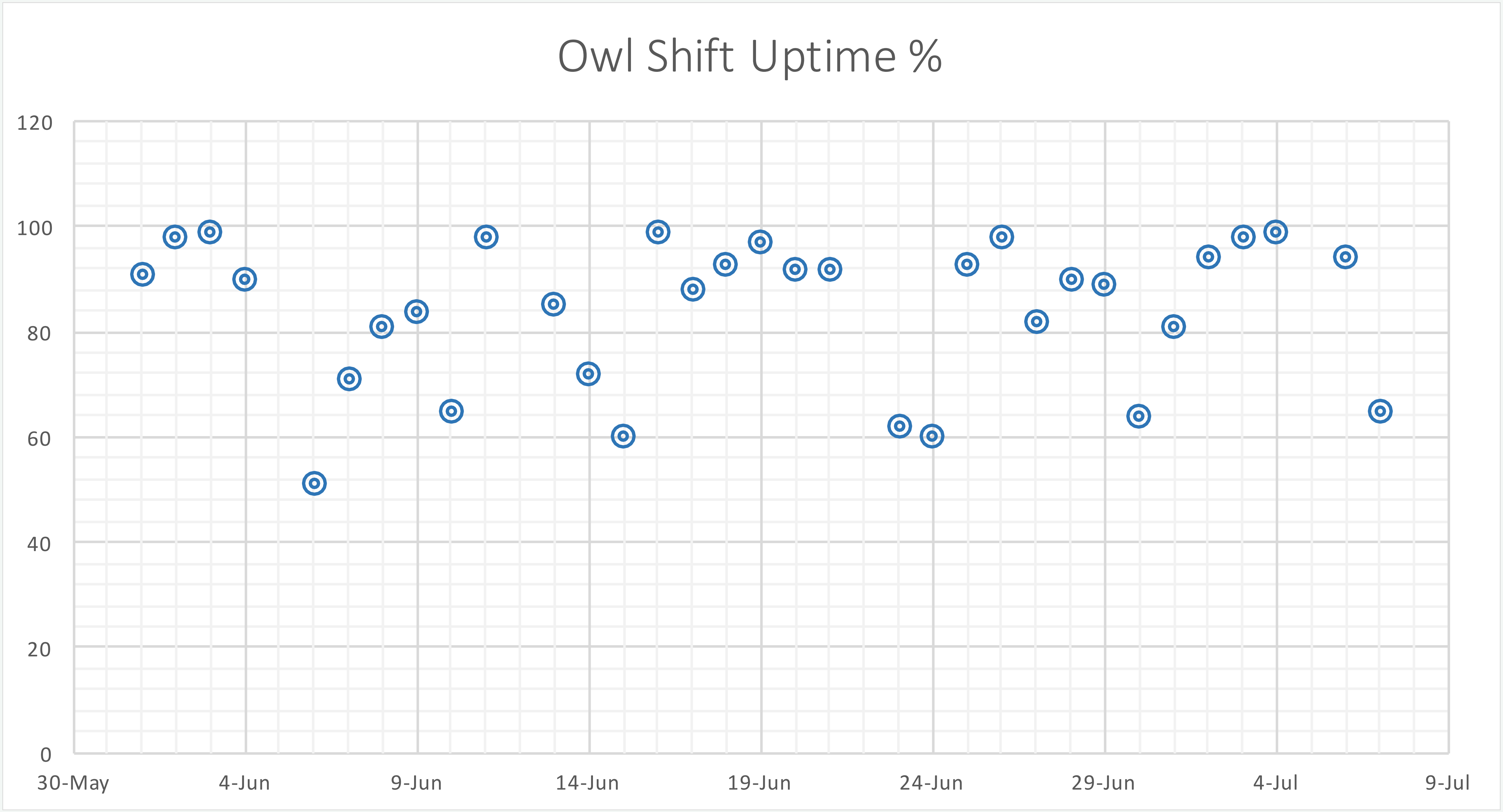
Fermilab

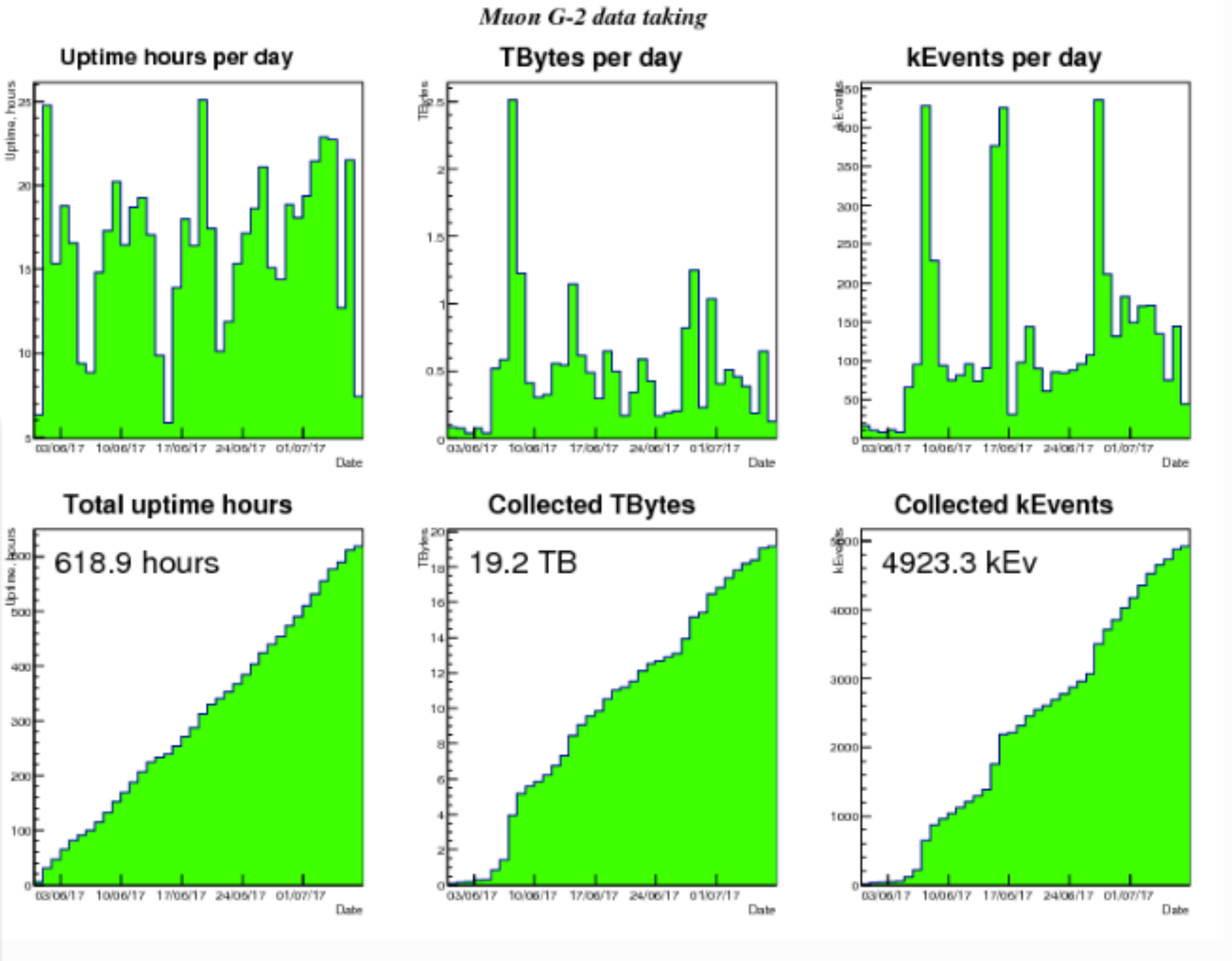
Data Format


MIDAS Run: 1 Hour, 120 subruns
Subrun: 2 GB
Event: 1 Fill (700 us)
GPS Timestamp
T-method
Q-method
Prescaled Raw
Pedestals
Header/Trailer
Process monitoring
Auxiliary banks
Tracker bank

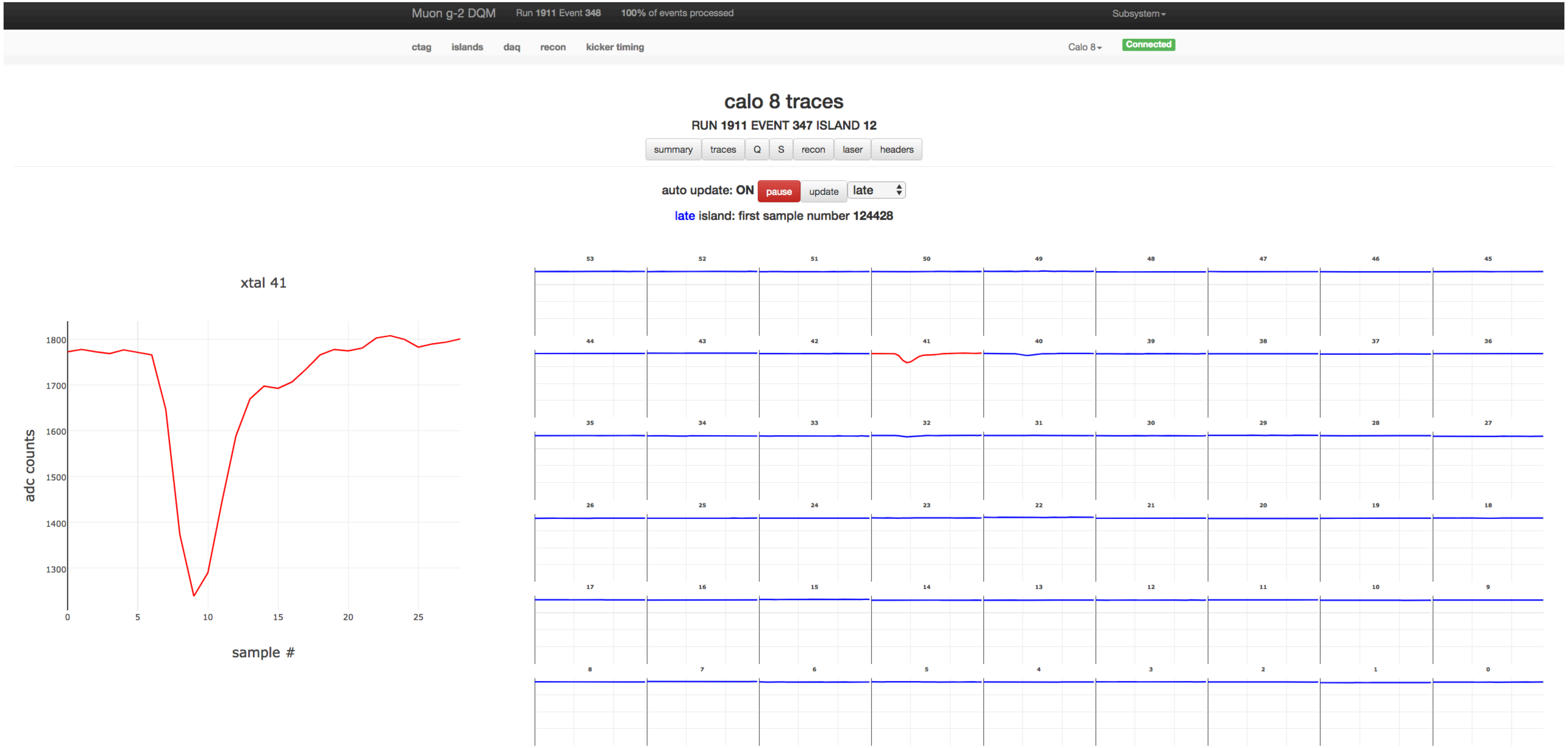
DAQ Performance During First Run

- The MIDAS DAQ performed well during our first run.
- Day shifts were mostly dedicated to beam line commissioning, so production data was taken at night.

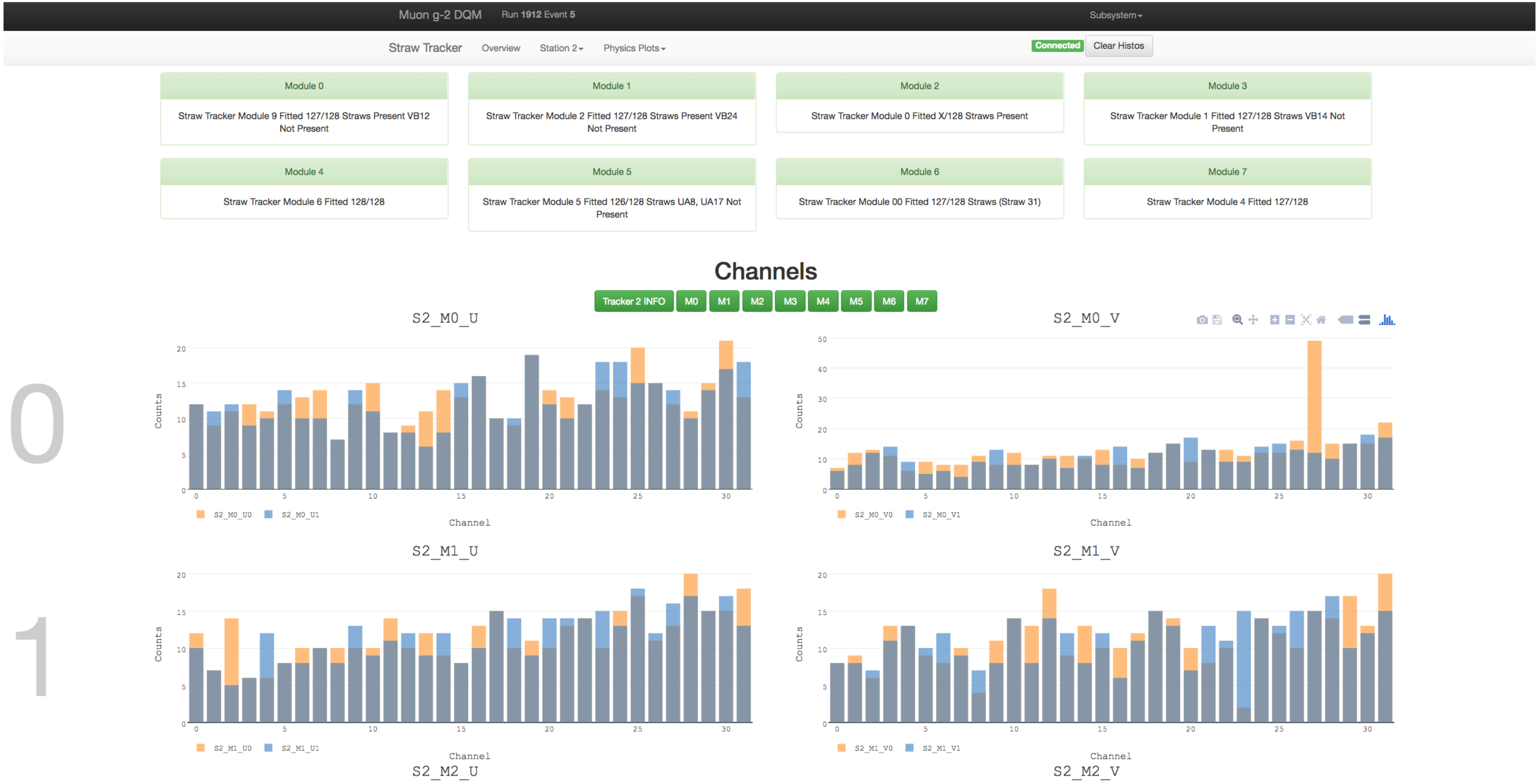




Position of pulse in calorimeter



Tracker Monitor

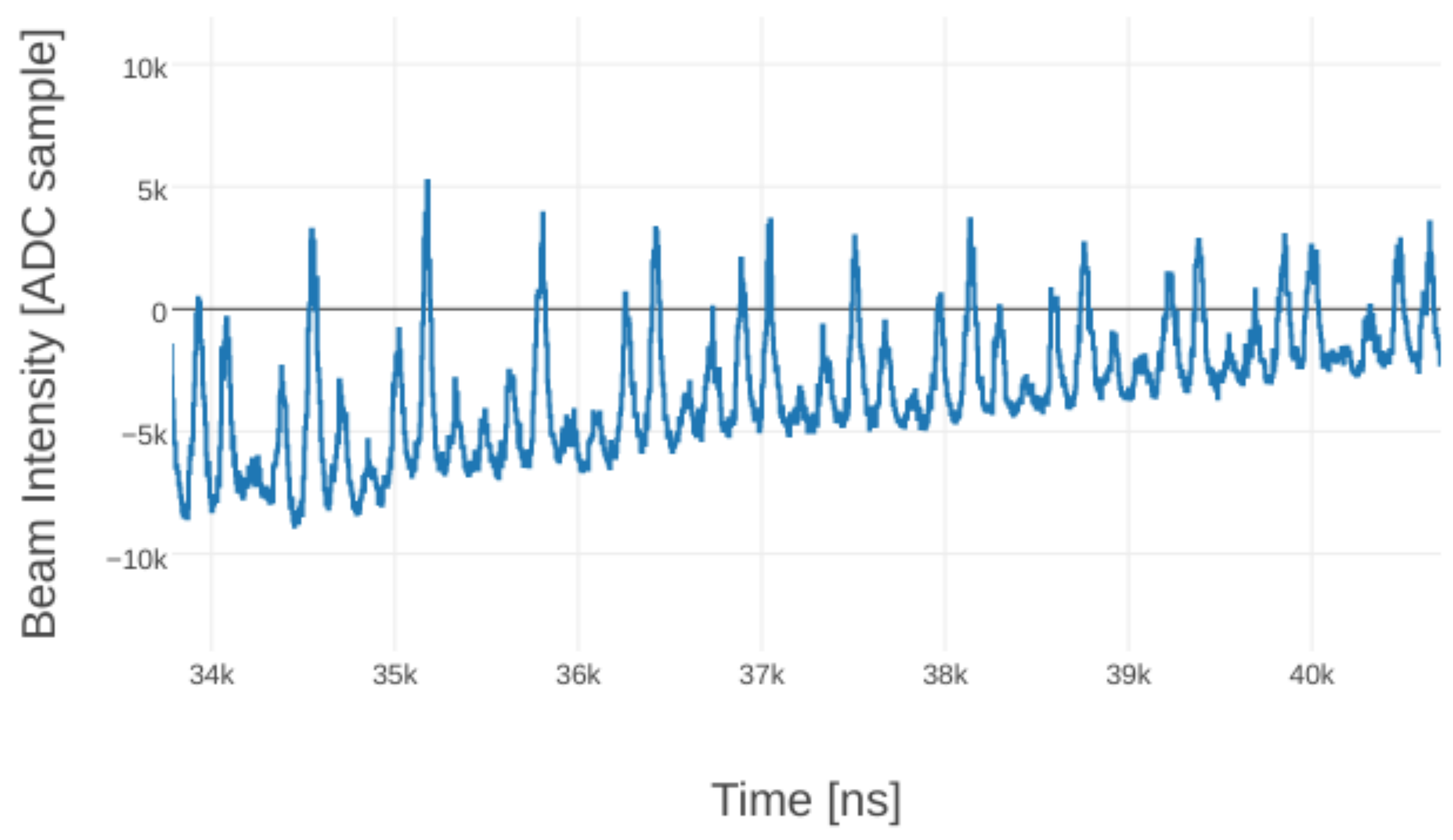


0

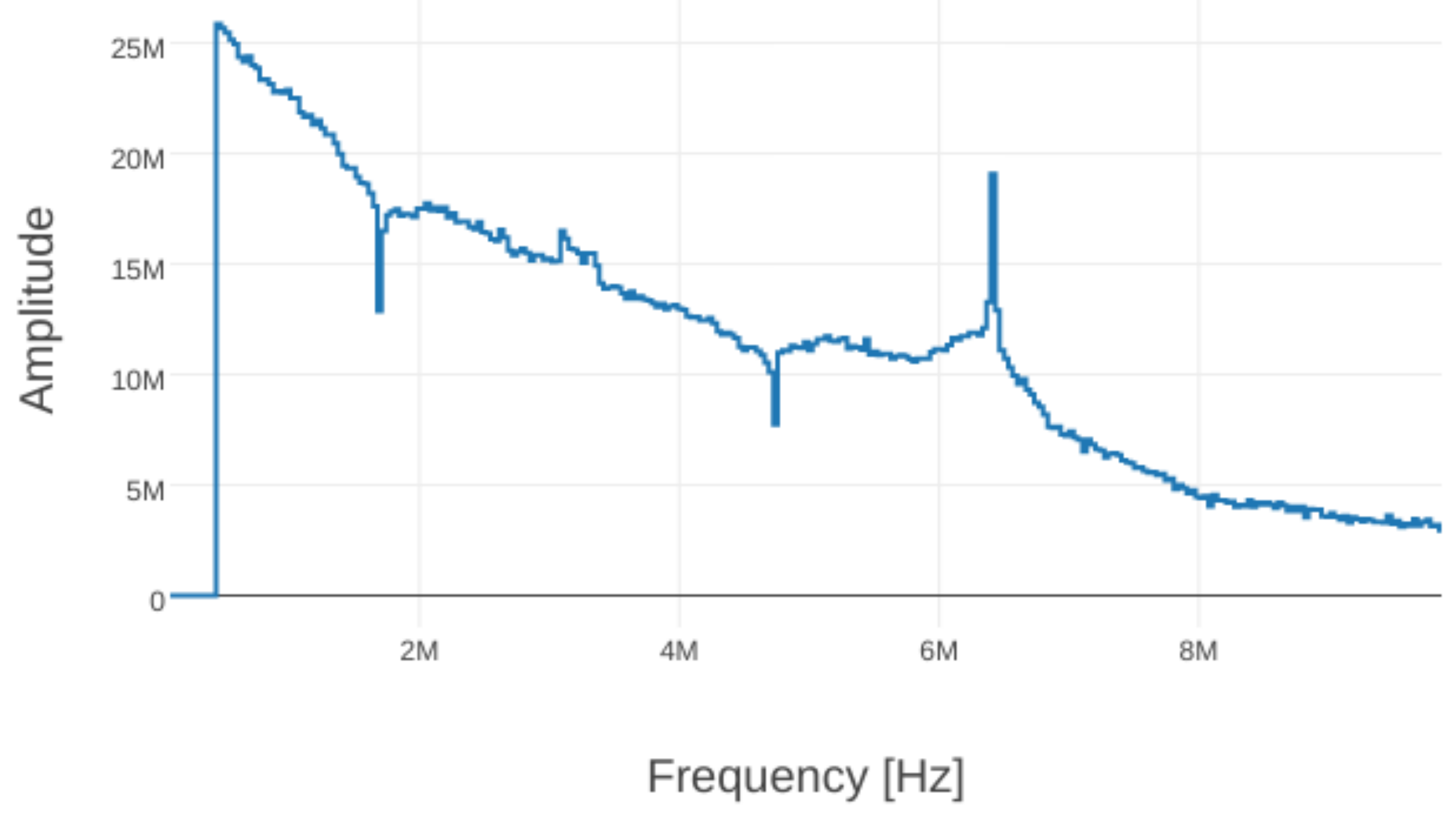
1

IBMS and Fiber Harps

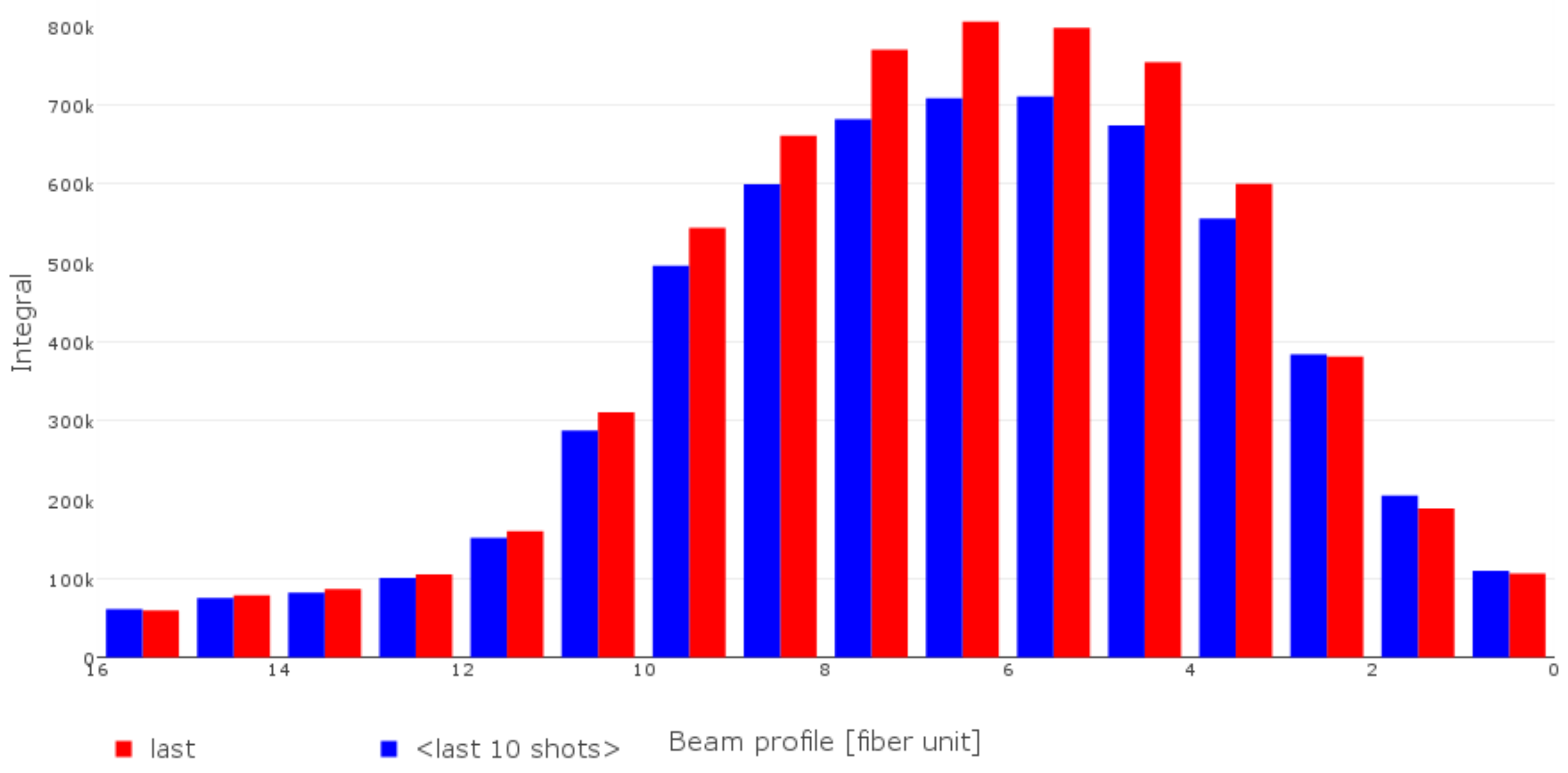
270 deg Y-profile Harp



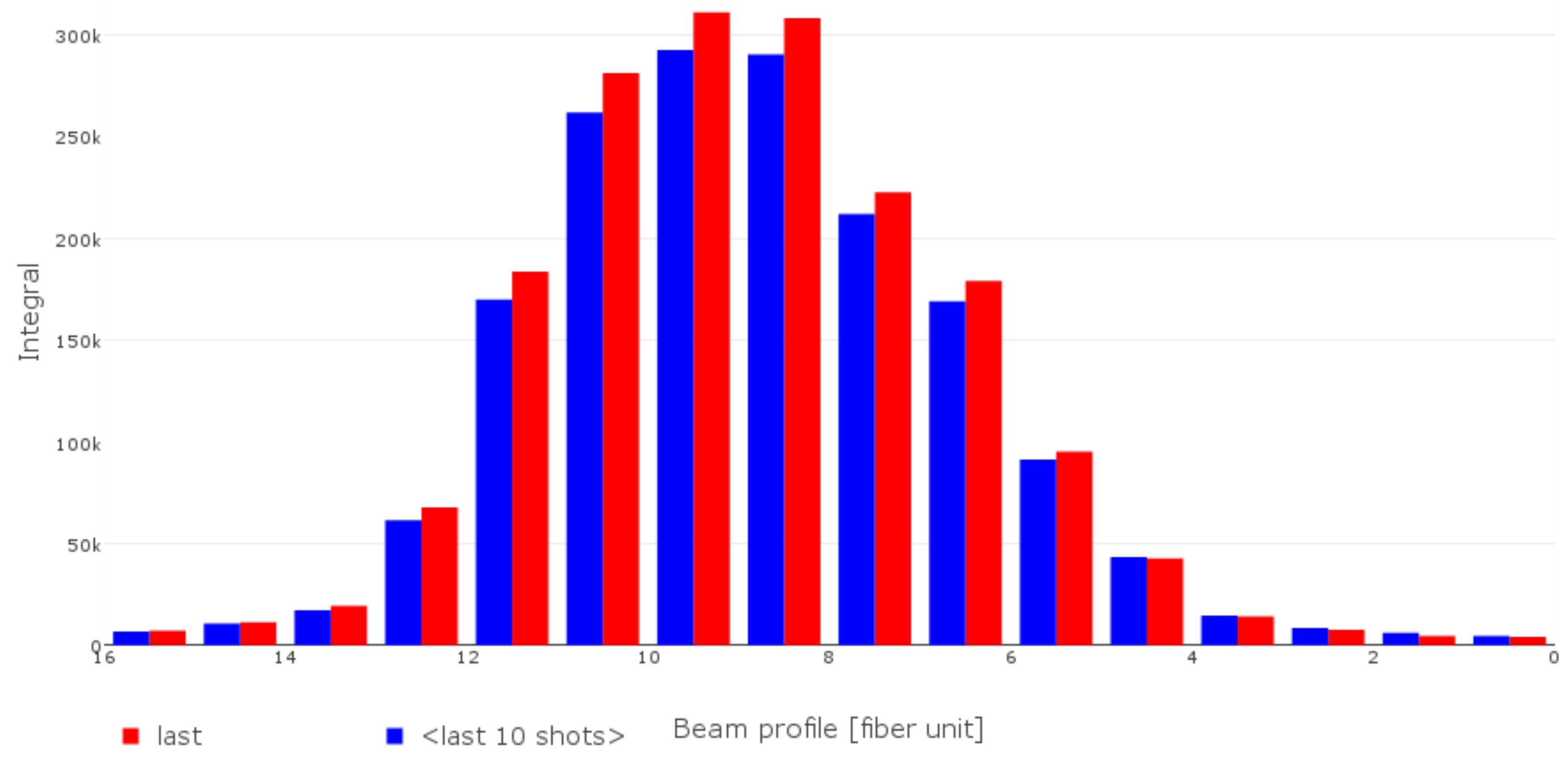
FFT BEAM INTENSITY (Y-profile Harp at 180 deg)



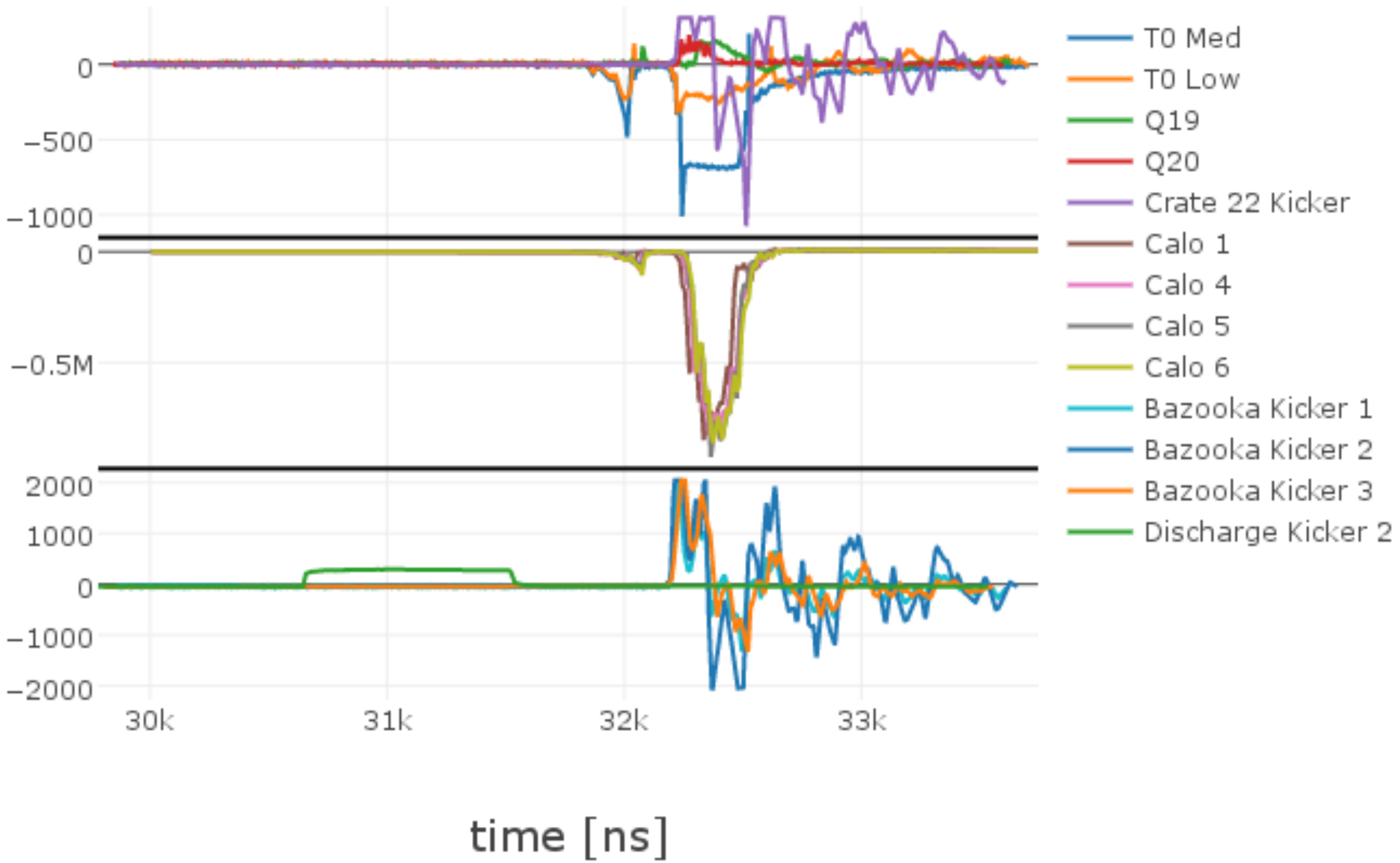
IBMS 1 Y Profile (0 up)



IBMS 1 X Profile (0 radially out)



Monitoring Beam Injection



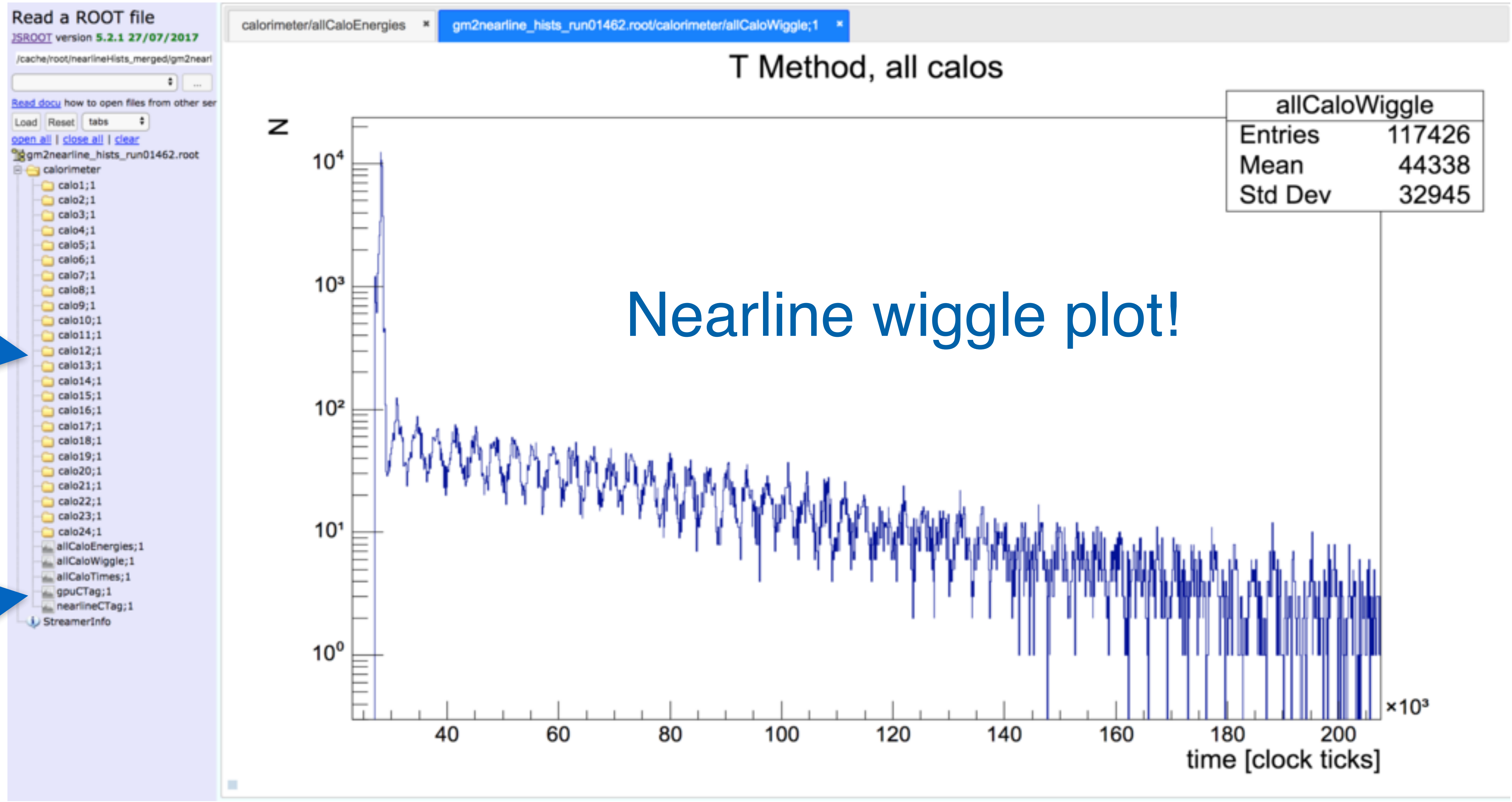
Fermilab

Nearline analysis

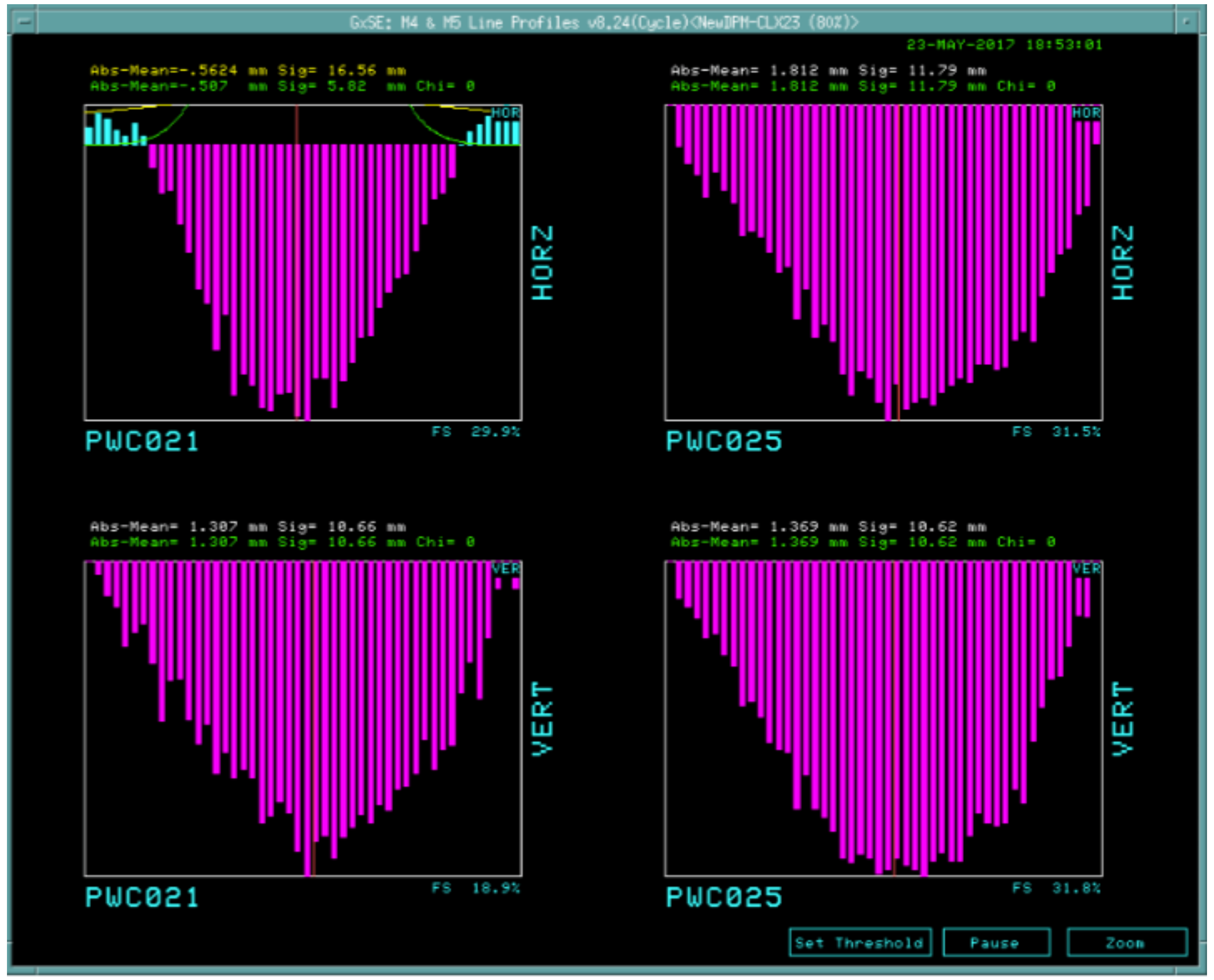
- Two dedicated machines in the control room are used for near line analysis.
- The near line analysis processes data from the most recent run, but performs the full art-based analysis on every run it processes.
- It is allowed to get behind the current run, but so far it can keep up with the live running.
- A set of histograms is made available for each processed run via the Django based web display utilizing JSRoot.

Select run, detector, and plot from menu

Plots combining data from multiple detectors.



Beam monitoring



- ACNET
  - A dedicated ACNET terminal is in the control room as is used by shifters to observe the status of beam line components.
- IFBeam
  - An IFBeam display is kept visible for the shifter on one of the control room monitors to view trend plots of beam line measurements such as protons on target.
- MIDAS Beam frontend
  - A dedicated MIDAS frontend reads data from the IFBeam service and writes it in the data stream.

To-do before next run: Update MIDAS frontend to account for relative timing between beam line components.

Hardware redundancy plan

- Working with SLAM to develop a detailed plan for the replacement of any machine in the DAQ system.
  - Hot spare for GPU-based frontend machine for calorimeter readout (hot spare was successfully used during commissioning!)
  - Redundant backend systems that could share load temporarily if necessary.
  - Plan to add second database server.
  - IBMS requires custom hardware — only one machine now that can handle it.
- All computers are under warranty.
- Agreement negotiated with Dell on-site support to provide routine maintenance for hardware issues to reduce the likelihood of sending a machine off-site for repair.

Contributing Institutions and Responsibilities

Institution	Responsibilities	Personnel
University of Kentucky	AMC13 Readout, GPS, DAQ Hardware, IFIX integration, Online Systems Management	PI + PD (on site) + GS (on site)
University College London	Tracker readout, Online Management	PI (on site) + PD (on site) + GS (on site)
University of Washington	DQM, IBMS readout, Field, Nearline	GS + 2x GS + PD
Northern Illinois University	Slow controls	PI
Argonne National Lab	Field DAQ	PD (on site)
JINR, Dubna	Web interface, Event display	2x Scientist
Novosibirsk	Django Web Display	PD
University of Michigan	Field	GS
Shanghai	Database management	PD
Italy (Frascati + Napoli)	Laser system integration	PI + PD (on site) + PD + GS
Cornell	Auxiliary detector integration, Clock	PD + GS (on site)
Fermilab	Beamline integration	Scientist

~24 total

RED = DAQ on-call

DAQ On-site support

Level 1	Shifter tries to debug the problem.	2 shifters in control room at all times
Level 2	DAQ On-calls are called.	2 DAQ experts on-call at all times
Level 3	Local online experts (Wes and Becky) are called.	
Level 4	SLAM is paged by Wes or Becky.	SLAM provides 24/7 support
Level 5	Dell on-site hardware support is contacted.	Maintenance agreement to have computer hardware serviced locally.