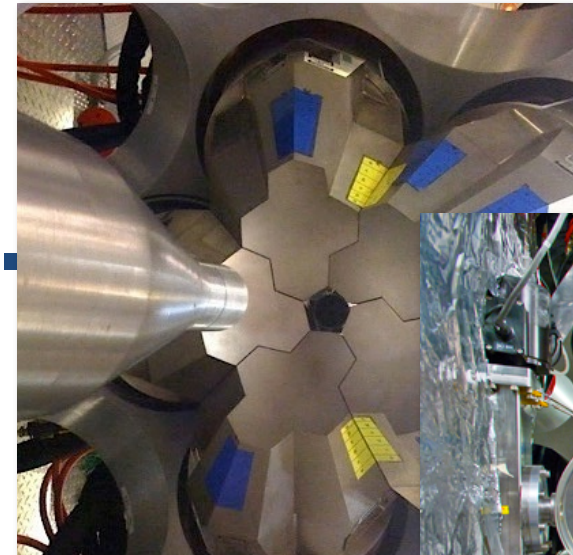
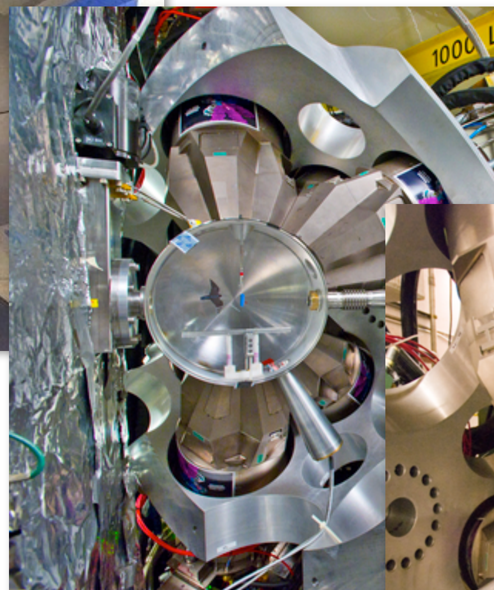
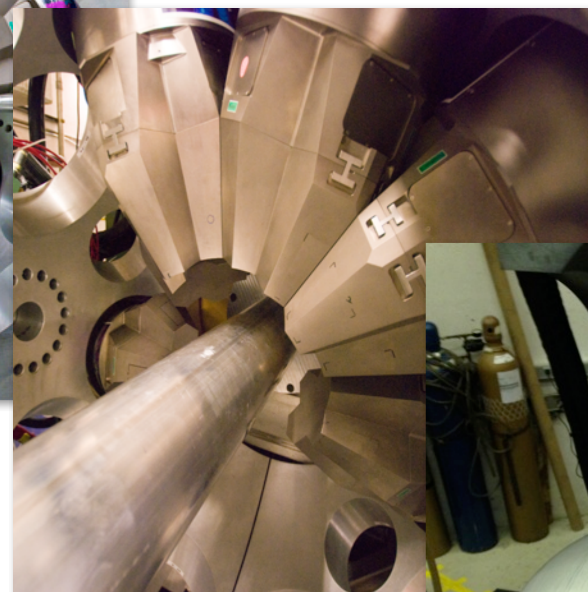
Data Streaming

Mario Cromaz, LBNL


CHEP 2016

BERKELEY LAB
Lawrence Berkeley National Laboratory

U.S. DEPARTMENT OF ENERGY | Office of Science

Work supported under contract number DE-AC02-05CH11231.

Outline

- Motivations
- GRETINA/GRETA - a streaming example from gamma-ray spectroscopy
  - The Present
    - Signal decomposition (FPGAs + online computing)
  - A Future
    - Moving real-time algorithms from FPGAs to general purpose computing
- Summary

Data Streaming

- ***The transport and prompt analysis of "raw" data streams from detectors to extract physics observables through the application of computational resources.***

Enabled by Technological Developments

- **Local Compute**
  - Large, local computing clusters (10^3,10^4 cores now ..10^5 cores soon!)
  - High performance GPUs
- **Network improvements**
  - Availability of high-speed, commodity networking
  - 10Gb, 40Gb commonplace .. 100 Gb available
  - Lower power, now easier to incorporate in front-end electronics
- **Deep Memory Buffers**
  - TB's of memory potentially available for large experiments
  - Allows latencies necessary for this model to work

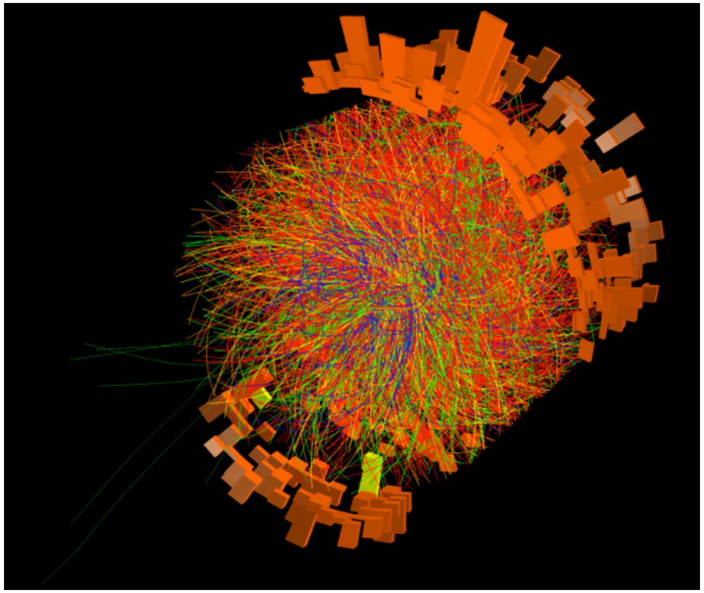
Why?

- **Physics Opportunities:**
  - **Smart Triggers / Trigger-less operation:** make better *(physics-based)* decisions based on a more complete and computationally intensive event analysis
  - **Rate:** mechanisms for pile-up recovery

- **Operational Motivations:**
  - Real-time monitoring of physics observables to optimize experiments, calibration
  - Online diagnostics
  - Accelerate analysis

TPCs

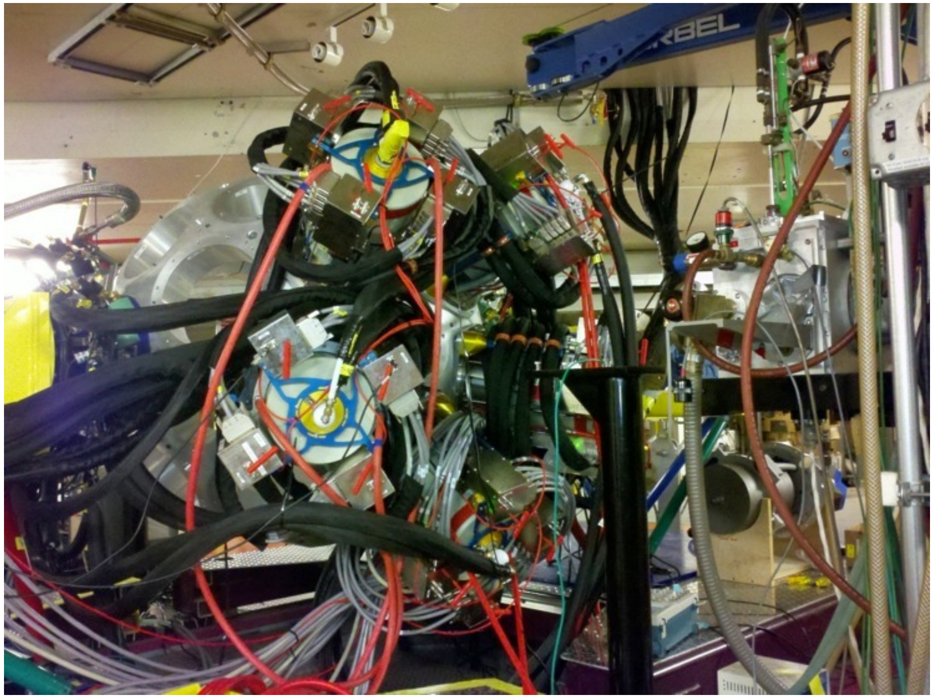
- Streaming in the High Level Trigger (HLT) - great deal of development in this area .. examples from high energy NP
  - ALICE
  - STAR
- On-line calibrations (updated on minutes timescale)
- Physics-based triggers to select output
- Compression, enable intrinsically non-specific triggers



An ALICE event display of Pb–Pb collision at 5.02-TeV nucleon center of mass energy. Image provided courtesy of CERN from the CERN document server.

A detailed example - GRETINA/GRETA

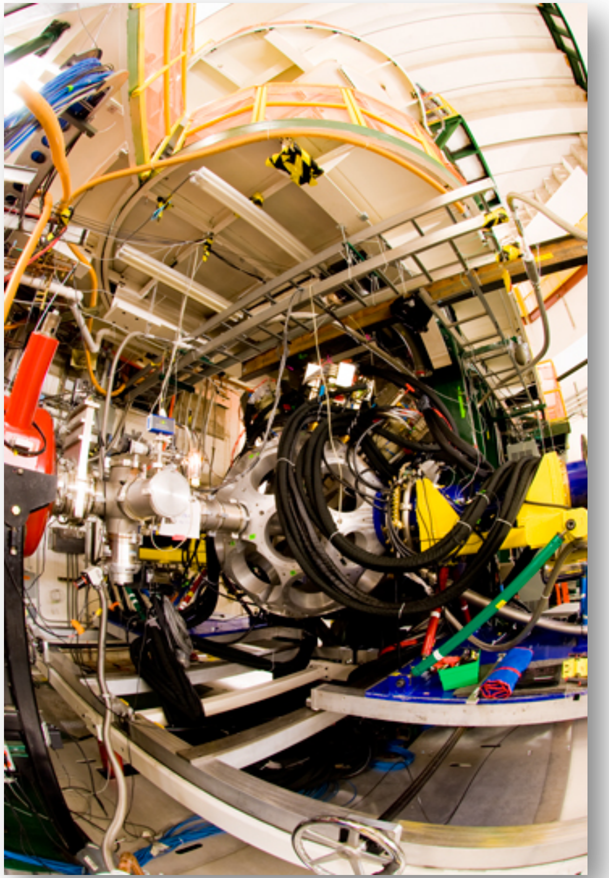
- First-generation HPGe gamma-ray tracking array (low energy - 10’s keV to several MeV)
- Spherical shell of segmented Ge surrounding target - GRETA - full 4π implementation approved; ready for FRIB
- Carried out very successful physics campaigns at NSCL/ MSU, ATLAS/ANL
- Employs full streaming model; 1-pass analysis



GRETINA, engineering tests - 88” cyclotron, LBNL

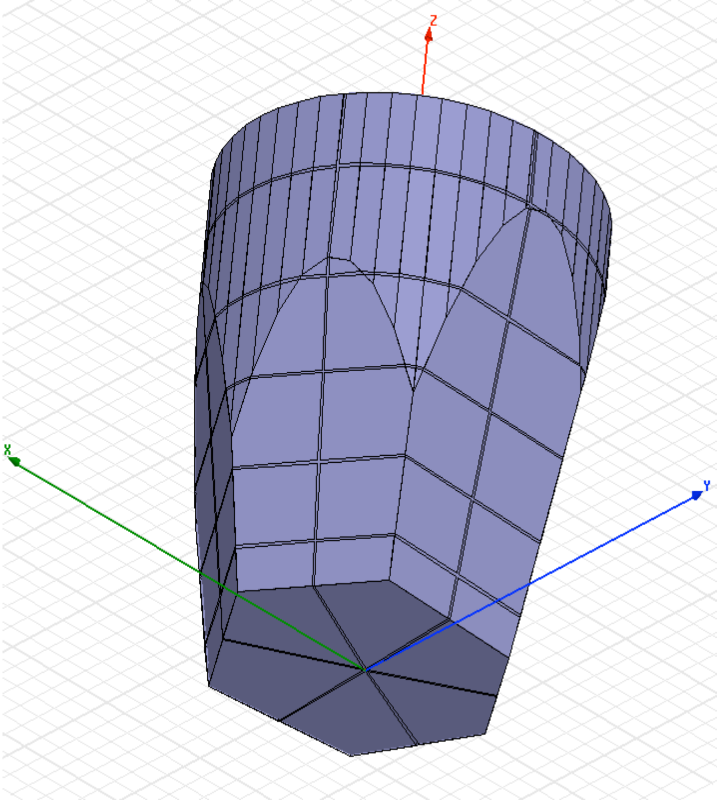
Running Spectroscopy Experiments

- **Streaming is an experimental requirement**
- Fixed target experiments: 2-10 day duration, can have multiple beam changes
- Varying experimental configurations - short setup times
- Small, often independent teams
- Ability to examine physics output to debug and optimize experiments in real time is an operational necessity



HPGe Signal Processing

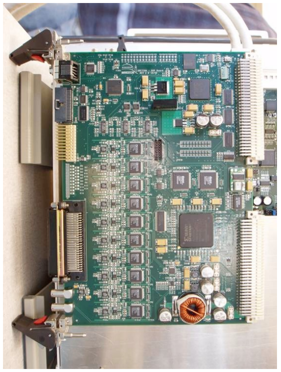
- Detector element - 36-way segmented HPGe crystals
- Two main observables:
  - gamma-ray tracks (positions of scattering points)
  - gamma-ray energies - 0.1% resolution
- Both derived from sampling the charge signal on each segment with a flash ADC (100MHz, 14-bit)
- Each observable has it’s own time scale: 400 nS for tracks, 10 µS for energies



schematic of a GRETINA segmented HPGe detector

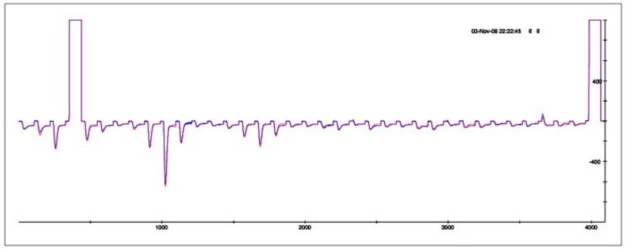
Data processing in HPGe Tracking Arrays

digitize segments, cc (36 + 1)

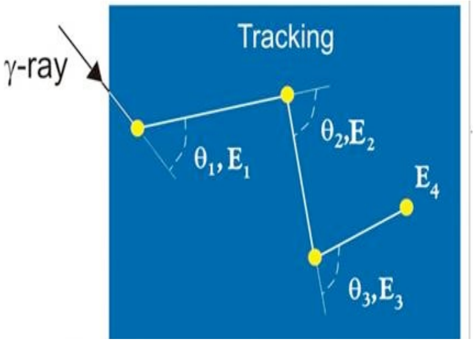


derive energy from trace

locate interaction points by fitting to crystal simulation

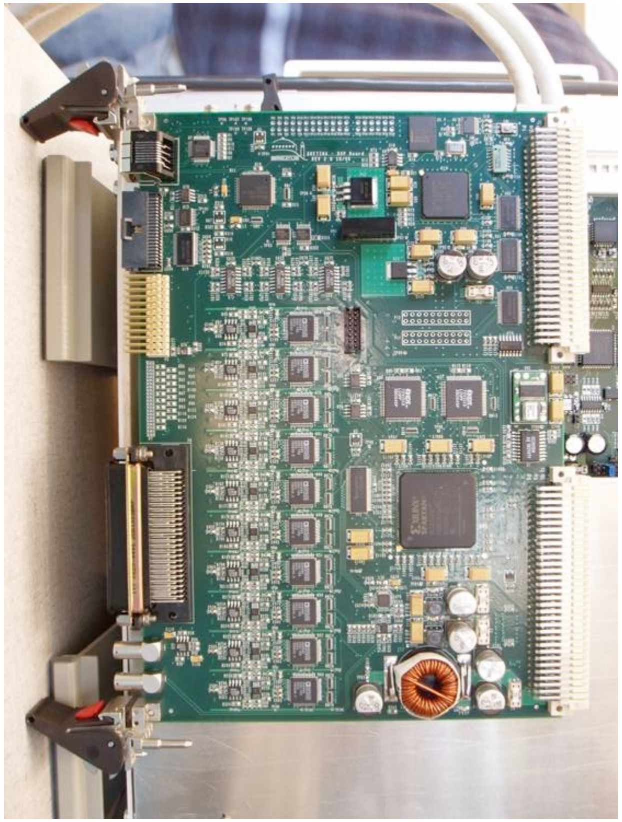


group/order interaction points by fitting to Compton scattering formula



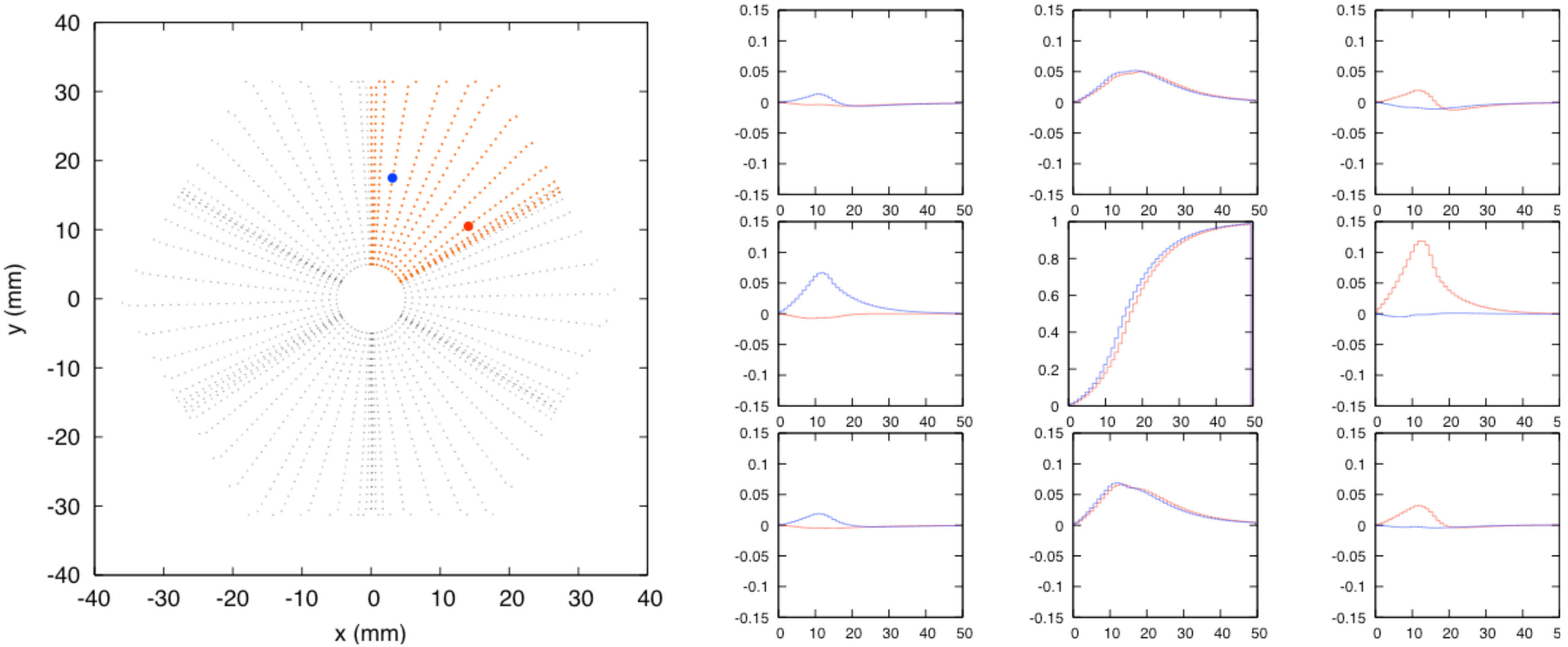
Today’s Way

- **FPGA processing + online computing**
- Two timescales (tracking, energies) “baked in” to signal processing architecture
- Generate local trigger primitives, run energy filters in an FPGA to reduce front-end bandwidth
- Window off short sections of trace during local triggers for downstream online analysis in cluster (signal decomposition / Compton tracking)



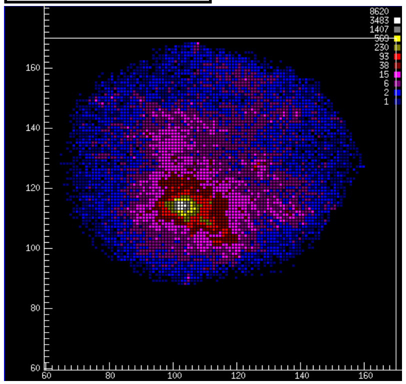
GRETINA 10 ch digitizer board

Gamma-ray Tracking; finding interaction points

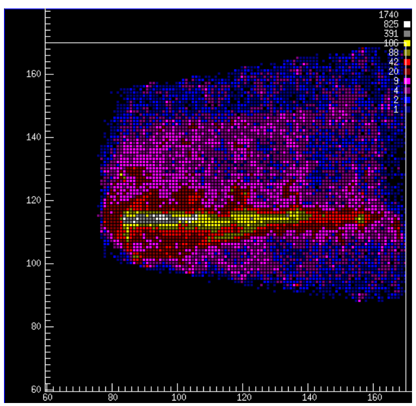


- drifting charge induces charge on segments
- requires flash ADC trace during charge collection (~400 nS)

Tracking gamma-rays



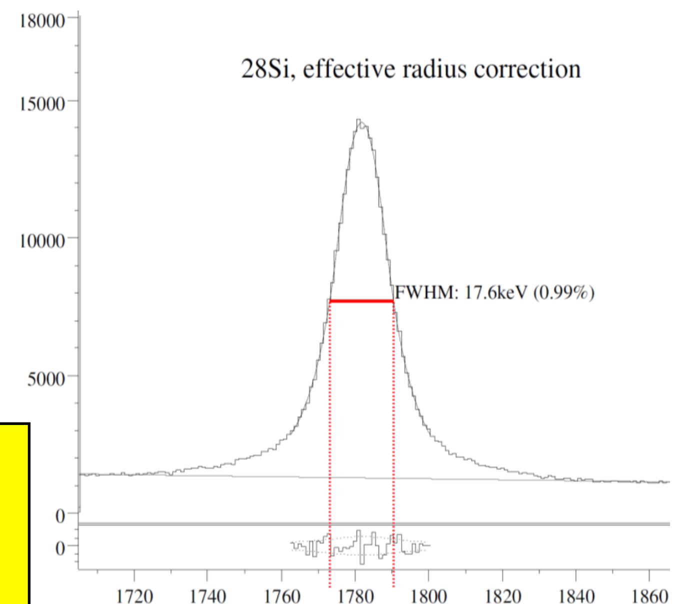
Measured spatial resolution of $\sigma \approx 2$mm



Collimated Cs source measurement

Processed in real time - waveforms not saved!

^{28}Si from ^{36}Ar on 47mg/cm^2 Be, v/c=0.38



Scale: Adaptive Grid + Gradient Search

- Aggregate I/O rates:
  - 16 kB / xtal evt * 1 kHz / detector * 36 detectors → 0.6 GB / s
  - 8 kB / xtal evt * 4 kHz / detector * 120 detectors → 4.8 GB / s

- CPU:
  - **1-pass - no iterative refinement step**
  - 1 kHz / detector * 36 detectors * 10 ms / core → 360 cores
  - 4 kHz / detector * 120 detectors * 10 ms / core → 4800 cores

- Memory:
  - 1.5 GB / signal basis (2 signal basis currently mapped to each node)
    (.. a worry ..)

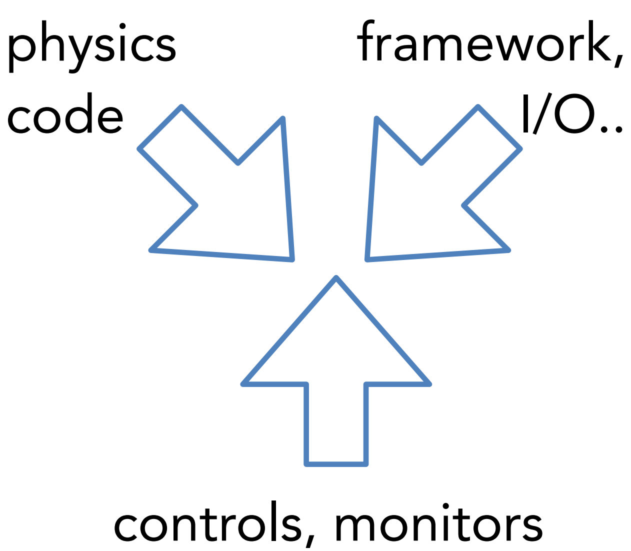
- **Total system latency: ~10s end-to-end**

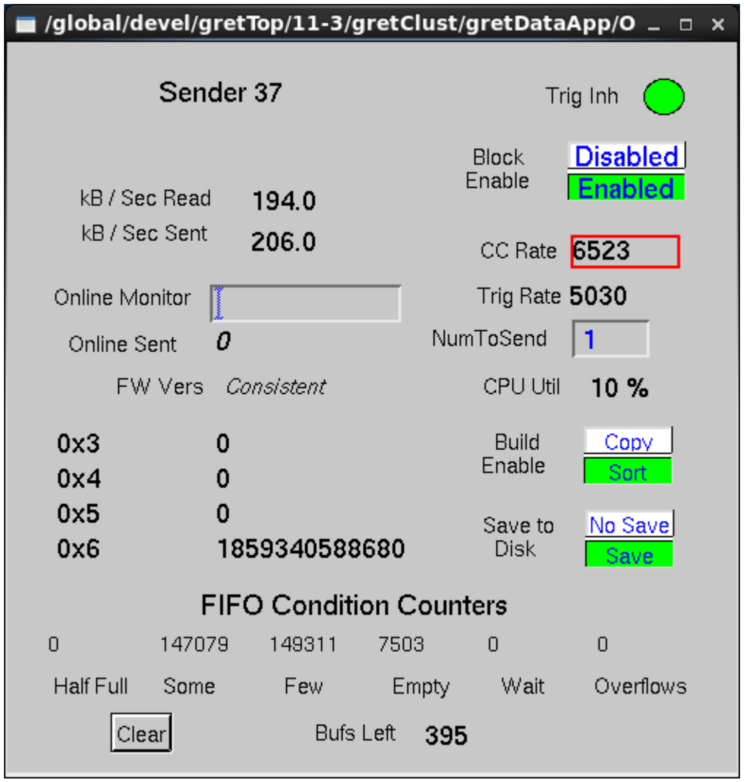
Parallelism

- Parallelism .. two-ways:
  - event level / time slices
  - detector level (individual detectors or detector sections can be treated independently)
- (For us..) nodes run in fixed association with given groups of detectors
  - U*pside* - simplicity of implementation, locality or large simulation sets
  - D*ownside* - core utilization, scalability (both up and down ..)
- Future (GRETA)
  - Take advantage of existing/developing frameworks (FairRoot/MQ, Apache Spark?)
  - Ideally .. we do physics code, infrastructure for “free” :)

‘Slow’ Controls too …

- Moving processing ‘online’ necessitates slow controls
- GRETINA - EPICs state machines running on soft IOCs



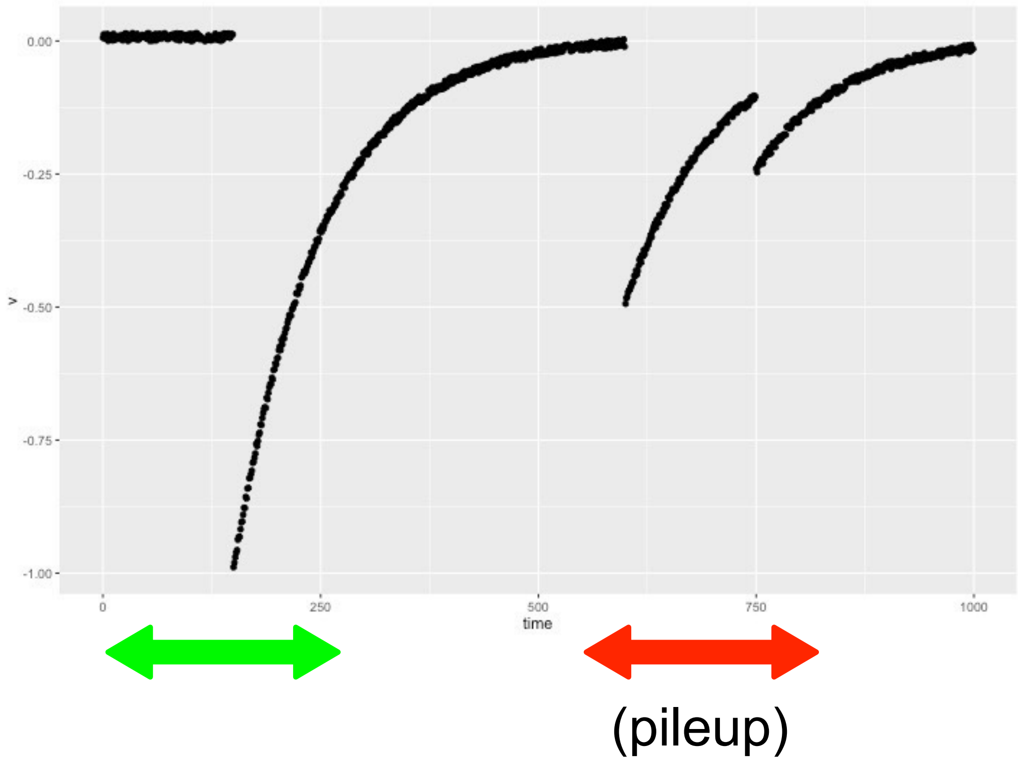


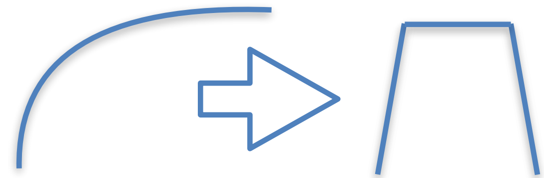
Rate - the next challenge

- Operating at maximal rate entails a discontinuous jump in complexity and computational resource requirements.

- Requires complete knowledge of signal
- Reading (1000’s of) waveforms from high-speed flash ADCs and run signal processing algorithms in high-performance computing system.
- Enables complex energy filters / algorithms to be applied:

Energy Determination

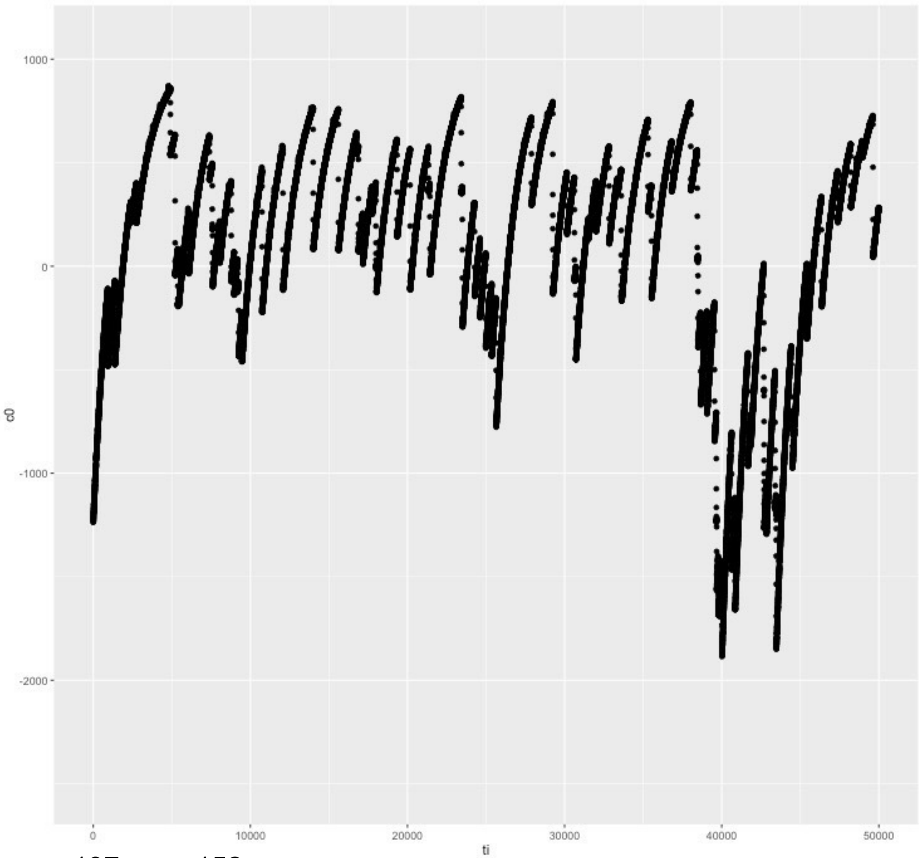




online trapezoidal filter implemented in FPGA memory pipeline (6μs flattop)

- HPGe detectors have very high intrinsic energy resolution (< 0.1%)
- require long integration times

Waveform taken at 50 kHz



^{137}Cs, ^{152}Eu sources; HPGe clover detector

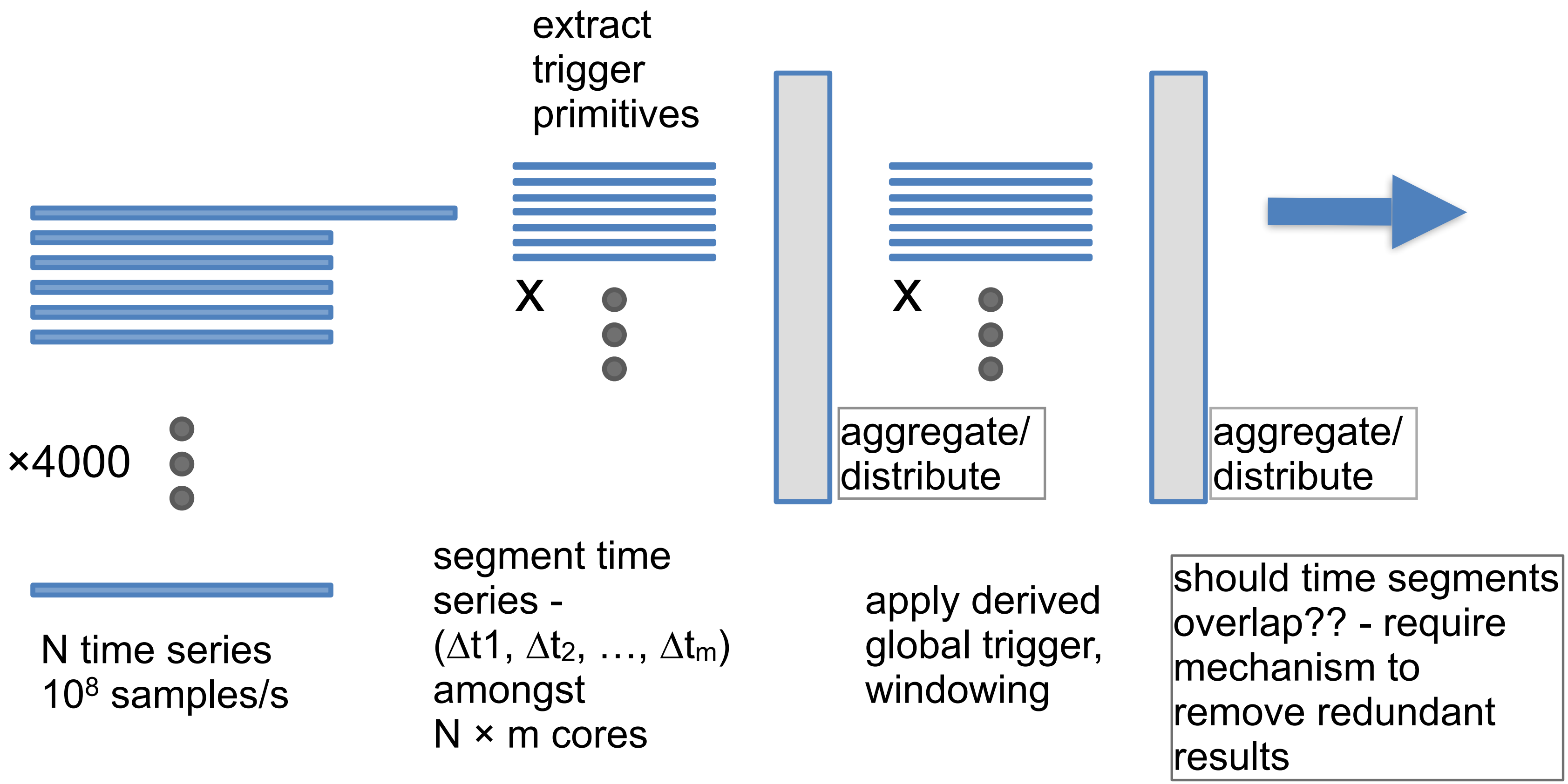
- Most signals overlap
- Erratic baseline - accounting for electronic response and history is important
- Difficult to maintain 0.1% energy resolution for sizable fraction of events - throughput losses

? - 100 kHz, .. more ..

Streamers (II)

- Faced with new requirements:
  - at the highest rates most of the waveform from the ADC is now necessary for accurate energy determination
  - the algorithms for extracting energy are progressing towards complex fits rather than filters
- **A Sol'n:** Extract all waveforms in their entirety and perform real-time processing using high-performance computing resources (rather than FPGAs)

The Computational Problem



Scale

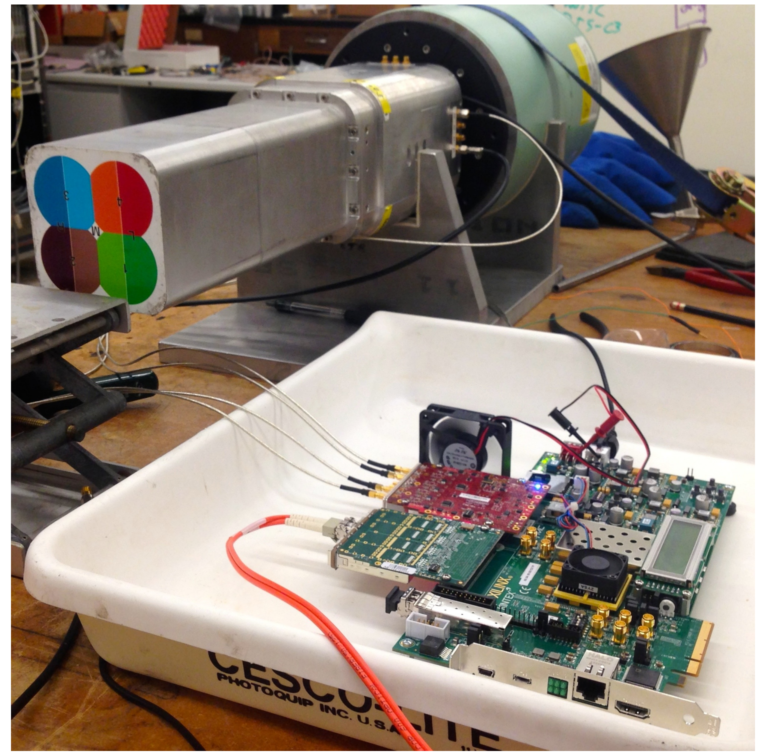
- I/O rates:
  - 100 Mhz * 2 bytes/sample * 0.5 compression factor → 100 MB/s/ch
  - 4000 ch → 400 GB/s

- CPU:
  - 100k γ/s/ch * 1 ms / core (???) → 100 cores/ch
  - 4000 ch → 4×10^5 cores

- Memory:
  - ~1s, + intermediate buffers .. a fewTB's

Prototyping

- Many people are prototyping these systems - we are too!
- Xilinx Kintex 7 eval board w quad 10 Gb interface + 4DSP FMC104 4-channel ADC board
- ADCs: 14 bit 200 Mhz, de-clocked to 100 Mhz,
- UDP packets from Xilinx core sent via 10 Gb interface to workstation, energy filters run in real-time.



Prototype streamer w HPGe Clover Detector

Frameworks for Streaming Time-Series Data

- Possible to employ a “hard-coded” system optimized to our specific problem.
- ***Best!!*** - a general framework for handling time series streams - separate concerns of:
  - writing physics algorithms for extracting observables
  - partitioning / distributing time-series data amongst computational resources and providing aggregation mechanisms
- General problem with wide applicability
- Such a framework outside the scope of a project of GRETA’s scale - *however* - *electronics will be designed to support future streaming model***

Applicability

- Not all detector systems are going to fit to a full streaming model ***but* some soon will!**
- In those cases where is the "line"?
  - Full streaming or FPGA+stream? How far do you take computing into the front-end?
  - Algorithm dependent
  - Ultimately will be a cost/resource/risk tradeoff for each detector system (FPGA engineering resources vs computer hardware, software engineering)

Cross-Collaboration

“ We will use your stuff and tell you when we don’t understand :) “

- Size of problems, resource limitations require collaboration and software re-use.
- Theme in the recent Exascale requirements workshops (cuts across DOE: HEP, NP, ASCR).
- Many of the problems are common; the solutions adaptable.
- Expertise is distributed amongst programs and projects.

Summary

- Streaming systems are being implemented today, in conjunction with pre-processing in FPGAs, across a range of detector systems
- Provides direct physics and operational benefits
- Technological developments are enabling streaming to move into traditional front-end electronics
- We believe full streaming of flash ADC data from large detector systems is becoming a tractable problem - can increase the physics scope of these devices.
- Generalized frameworks to support streaming architectures will have a broad impact for physics instrumentation (and beyond)