Streaming Data

Graham Heyes

Data Acquisition Support Group

Thomas Jefferson National Accelerator Facility

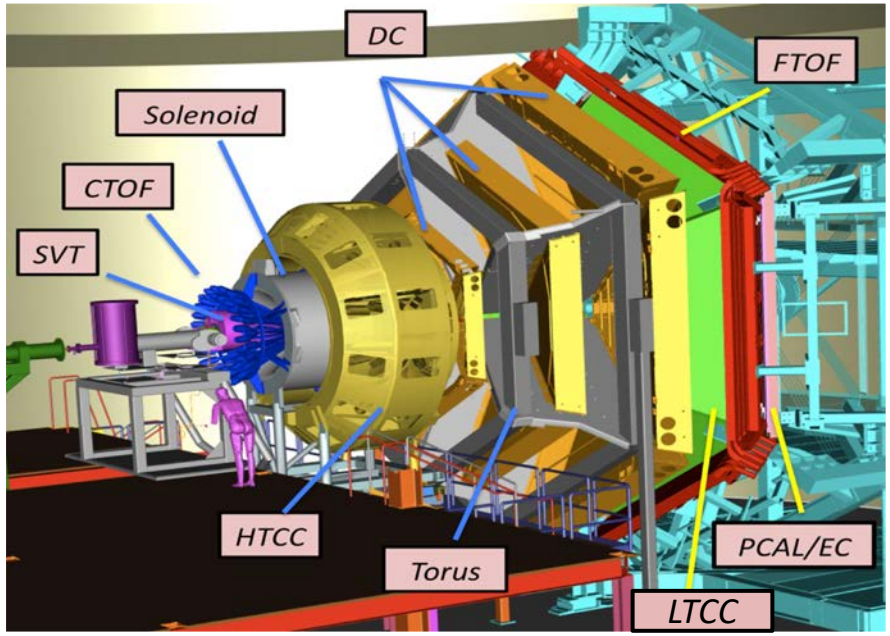
(Jefferson Lab)

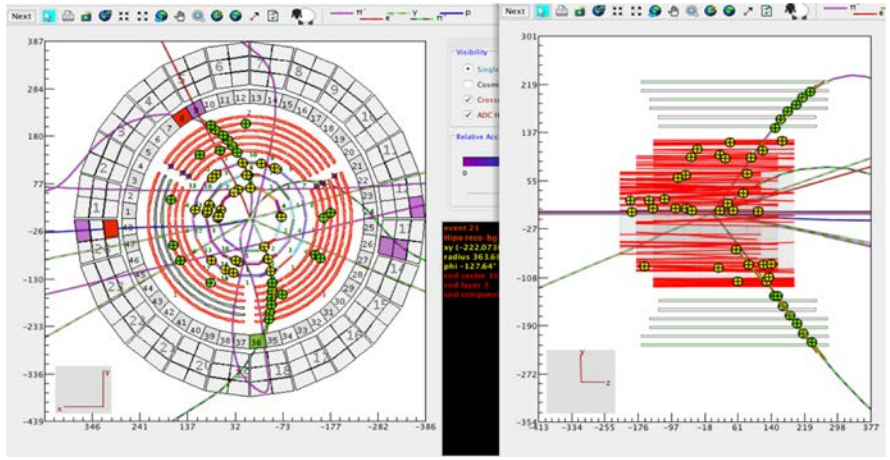
Jefferson Lab

Example data source

- This talk will use as examples data sources that I am familiar with. I hope to show that the challenges are more general.
- My lab studies nuclear physics, the structure of the nucleus, using a high energy beam of electrons and stationary targets of various materials.
- Each interaction between an electron and a nucleus, an event, is recorded using an array of various types of detector that measure properties of particles produced by the event.
- Our two large detectors are CLAS12 in hall-B and GlueX in hall-D.
- Each detector has hundreds of thousands of data sources.

Clas12/Hall B Detector

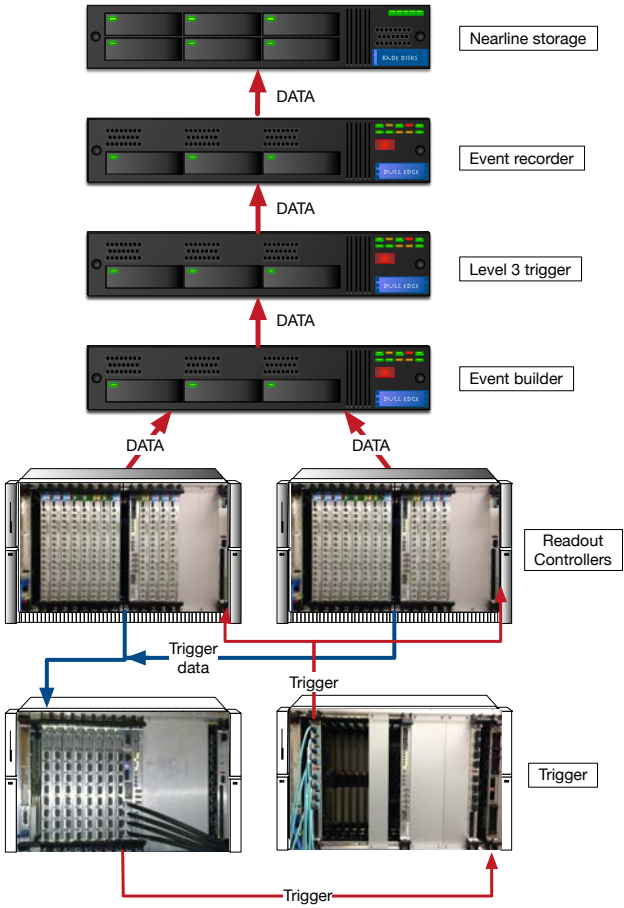




Jefferson Lab

Data/work flow

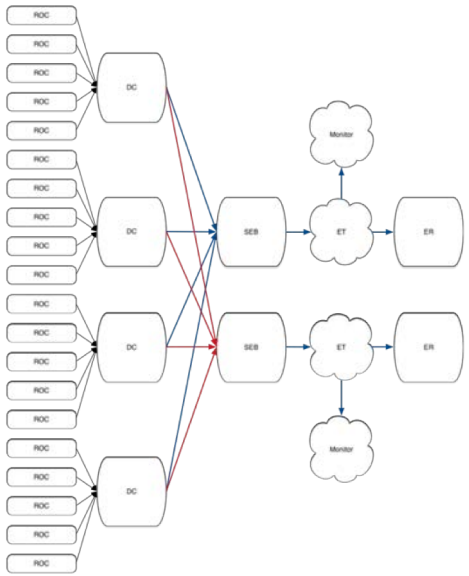
- Custom and commercial electronics in VME format digitizes signals from the detectors.
- Most of the signals are noise or uninteresting events.
  - Algorithms implemented in FPGA firmware use prompt data to form a ‘trigger’. – analogy: camera shutter
- Online Event Builder.
  - Merges event fragments from different detectors.
  - CLAS12 has ~100 Readout controllers and a 12 kHz event rate ~1.2 MHz rate of event fragments at the EB.
- Event Recorder software writes events to files on disk. and mass storage system archives to tape.
  - Tens of PB per year of raw data.
- Each file contains a linear sequence of self contained events each tagged with an event number.
  - Analogy, events = individual frames on rolls of cine film.
- Groups of files are processed offline by “jobs” on a batch system fronted by a workflow management tool.
  - Onsite cluster, OSG and NERSC in use.
  - Algorithms validated using simulated data based on theory.

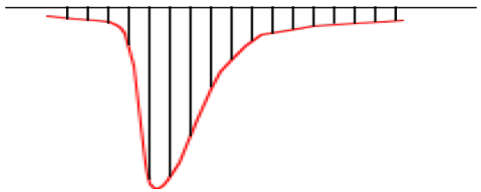


Jefferson Lab

What is wrong with this model?

- Trigger and event building require strict online synchronization.
  - Have to delay prompt data until slowest data appears.
  - All parts of DAQ have to work. One failure stops the pipeline.
- Relies on good understanding of the trigger.
  - Triggering has the potential to throw away useful data.
- Doesn't work well when events overlap in time.
- Obvious bottlenecks force us to
  - Limit overall event/data rate – burns accelerator time.
  - Deploy complex system topologies.
  - Make science compromises to limit data volume.
- Batch processing files of events means that each job gets all of the data in a file even if only a part is of interest.
  - Each event is treated in isolation – impossible to deal with events that overlap in time.
- The whole workflow is slow moving. Typically take months of data before starting analysis which can then take years.





Streaming Data Model.

- Why take still photos when you can make a movie?

- Data is continuously read, timestamped at source, and flows in parallel through the system with minimal online processing.
  - Minimal or no trigger.
    - "Save it all and figure it out later".
    - No science compromise, all data saved.
  - Parallel data flow by default.
    - Reduced or eliminated bottlenecks.
    - Slow detectors stream at their own rate, no need to delay fast detectors.
  - Complicated issues dealt with offline.
    - Detector topology.
    - Event overlap.
  - Makes possible new data analysis methods.

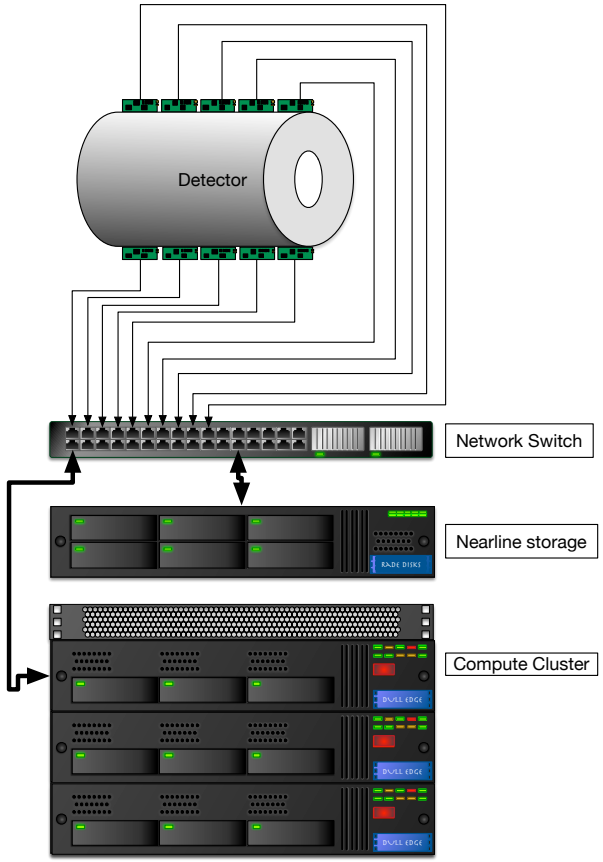
Streaming data

- Streaming data processing requires two layers:
  - A storage layer
    - Record ordering – example time vs detector
    - Strong consistency – implied by reproducible streams of data.
    - Fast reads and writes of streams of data.
  - A processing layer.
    - Consumes data from the storage layer.
      - Running computations on data.
      - Notify the storage layer to delete data that is no longer needed.
      - Return stream of results to the storage layer.

- Have to plan for quality of service, scalability, data durability, and fault tolerance in both the layers.
  - Jefferson lab NP experiments run 24/7 (>60% uptime) for 34 weeks per year.
  - Running the accelerator is expensive – can't lose data.

Streaming NP Data Readout

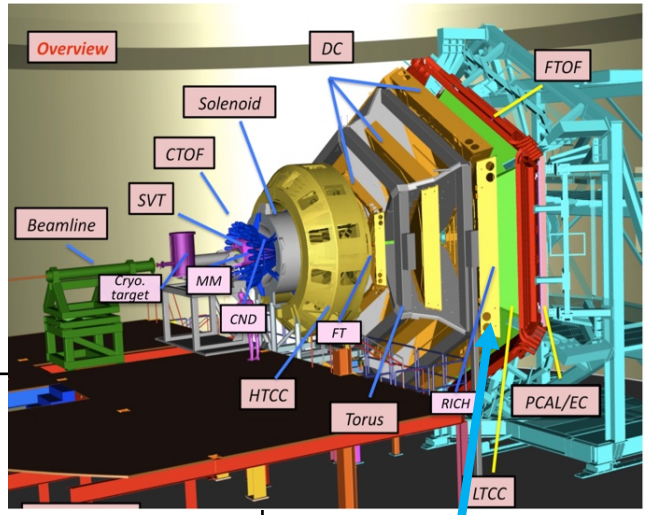
- Detector specific interfaces stream data on a fiber with a **well defined protocol**.
  - Low cost plug and play kit of parts.
- Data streams directly to nearline managed storage.
  - RAM, SSD, or RAID?
- Local compute cluster processes data in pseudo real time.
  - Scale? 500 mS/event * 12 kHz
  - Minimum ~5000x 2.5 GHz Broadwell cores.



Jefferson Lab

Real world project

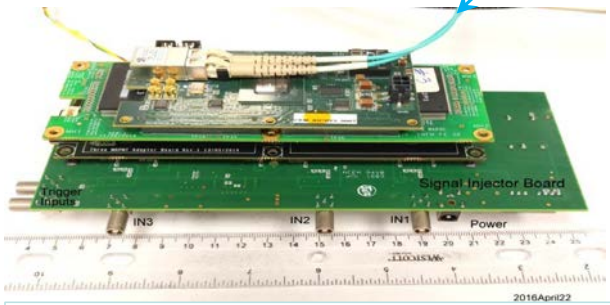
- CLAS12 is a general purpose detector with a lot of subdetectors.
- RICH detector is read out via fiber to Sub-System Processor (SSP) boards in VXS crates.
- Same setup is used by GlueX DIRC.



- **RICH (CLAS12) and DIRC (GlueX) examples**
  - **ALL FPGA boards have been tested(Completed in May 2016)**
  - **Production ASIC board(s) [2-MAROC and 3-MAROC] completed**
  - **Detector final assembly is ongoing**

391 -- H12700 Hamamatsu
64-anode PMT
Total anodes: 25,024

32 LC Fiber Links



On Board 192 channel FPGA Readout Board
MAROC3 ASIC mates to maPMT
Artix 7 FPGA drives LC fiber optic transceiver

VXS Sub-System Processor
32 - 2.5Gbps links to RICH
FPGA Readout Boards

RICH Detector

150,144 channels

Jefferson Lab

Other groups working in this space

- BNL - PHENIX and STAR are developing streaming detector readout systems.
- CERN
  - LHCb
  - Other LHC groups are looking in this direction too.
- Greta experiment at FRIB.

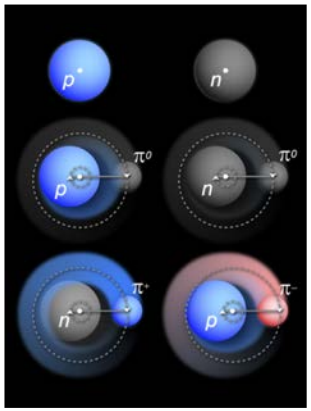
- The streaming data model is common in other areas:
  - Astronomy
  - Climate
  - Plasma
  - Many others

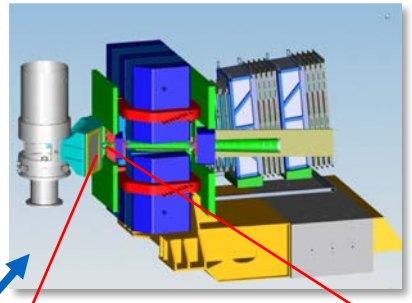
Why change now?

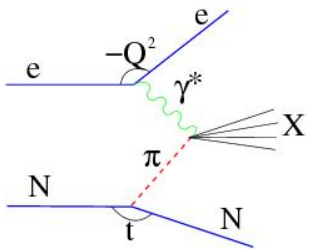
- Affordable technology to make streaming readout work is starting to appear.
- Experiments are planned that would be much simpler with streaming data.
- A few things are critical but need further R&D.
  - Timing
    - For a streaming data source to work timing is critical.
    - How do we accurately distribute the timestamps and clocks?
  - Data flow.
    - A single channel of a 250 MHz flash ADC 14-bit samples every 4 nS ~ over a gigabyte per second of data per channel - we have thousands of channels.
    - What real time algorithms can reduce this rate without discarding useful science?
    - What is the optimum data transport protocol?
    - How do we control the flow of data?
  - Data storage.
    - We are used to a single dimension file indexed by event number.
    - What is the optimum way to store multidimensional time ordered data?
  - Data retrieval and processing
    - Offline what is the optimum way to access and process this data?

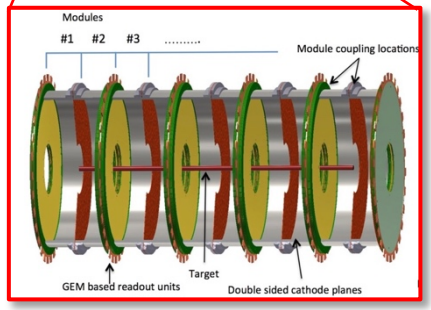
Timing

- A proposed experiment at Jefferson Lab will bounce electrons off nuclei to look for a rare **inelastic** event where a proton is ejected.
  - The event is “tagged” by the coincidence between a proton in one detector and an electron in another.
- The electron is detected within nanoseconds of the interaction but the associated proton is detected in microseconds.
- There are many **elastically** scattered electrons that are not associated with a proton.
- For streaming readout of this detector to work we need to tag the data with a timestamp to sub-nanosecond accuracy.
  - The timestamps from thousands of individual channels must be synced.
  - The timestamp must be implemented in a way which doesn’t blow up the data volume.









Jefferson Lab

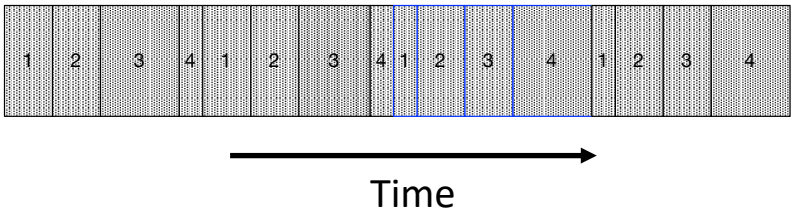
Data flow

- Streaming data requires.
  - Well defined quality of service and latency.

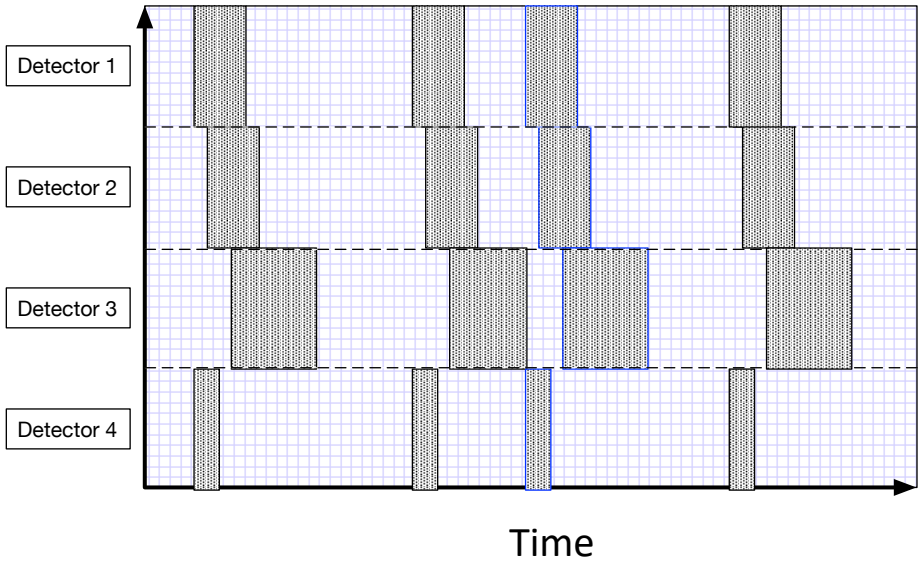
    Analogy, you can't stand at the bottom of a waterfall and control the flow by pushing upstream.
    - Control at source or
    - Discard at destination.
    - An area requiring modeling.
  - What exists already?
  - What are known pitfalls?
    - Drivers and software stacks are often not optimized streaming.

Data storage

- In existing systems data is 1-D sequenced by event stored in files.

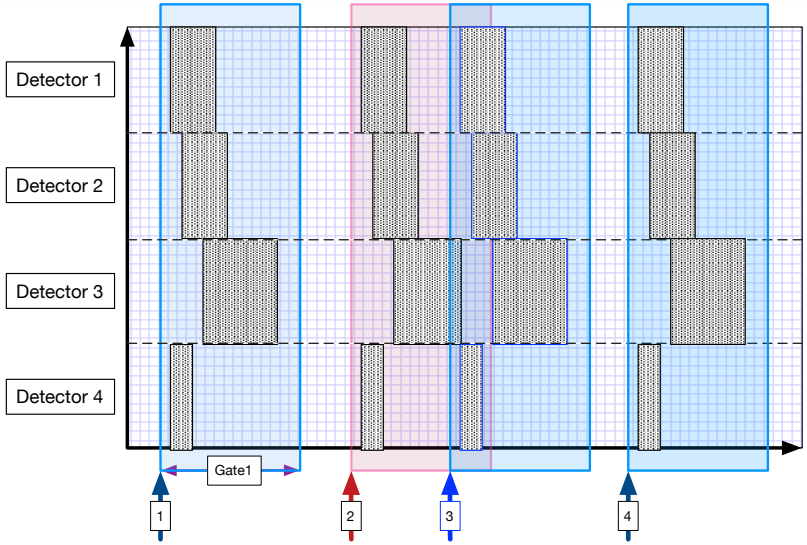


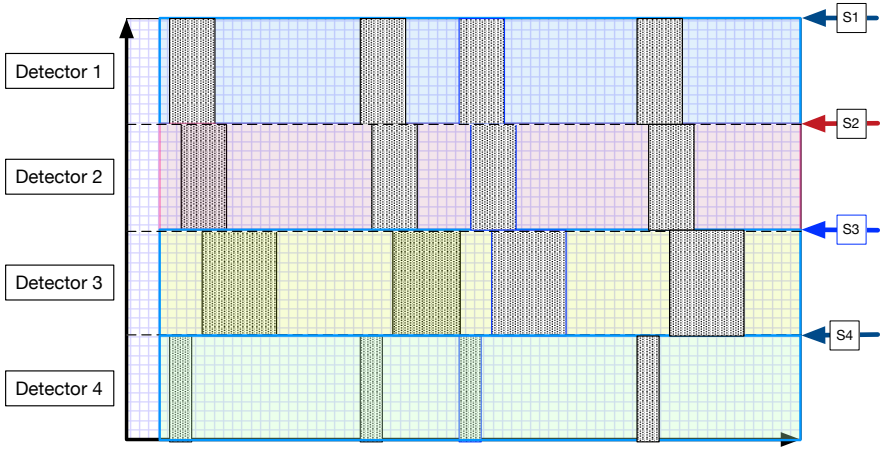
- In a streaming system data is at least two dimensional :
  - Time vs Detector
  - Probably multidimensional depending upon experiment.



- What is the optimum format?
  - Database? Could be slow and cumbersome for many PB/yr of data.
- What is the optimum hardware?
  - Performance vs cost.

Data retrieval and processing





- Data can be retrieved as time based slices across streams or by stream.

- In the vertical slice based method:
  - A virtual trigger defines the start of a slice.
  - Each slice is wide enough to contain all of the data from an interaction.
  - Slices may overlap and contain data from other interactions.
  - Slices would be processed in parallel.

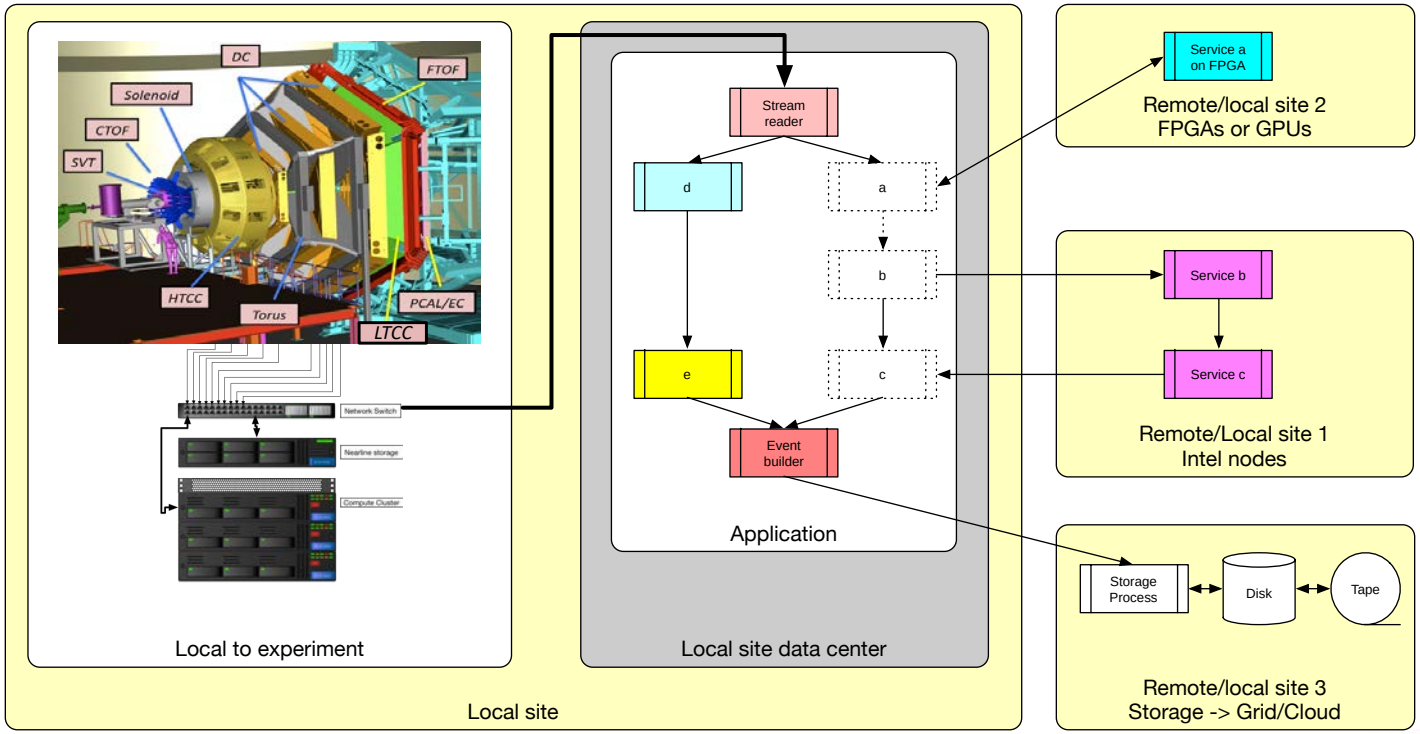
  - This is a different way for us to access our data but the processing is familiar and well understood.
    - Moved event building from online, where data is in motion, to offline where it is stationary/

- An alternative is to delay looking at data as events and deal with individual streams.

Jefferson Lab

Data retrieval and processing

- Reimagine applications as nets of services processing streams of data objects.
- Example CLARA, used in CLAS12 and in collaboration with NASA (NAIADS project)
  - Standardized application building blocks – one data type in, another out.
  - Streams route data to services running on appropriate hardware.
  - Need a method of associating cost with services.
- Currently we ship an application plus data to OSG or NERSC in a container.
- Instead deploy services at remote sites and connect them with streams.



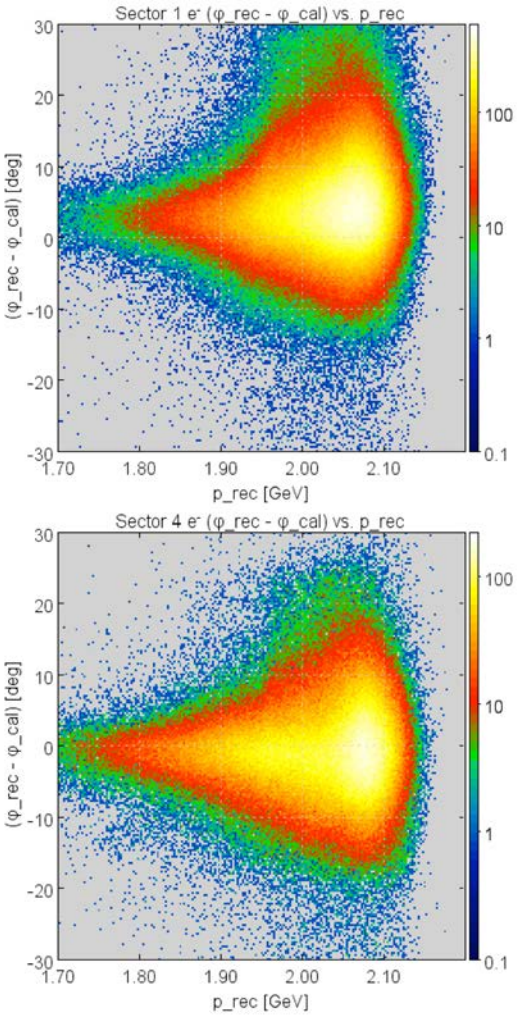
Jefferson Lab

Theory, machine learning and AI

- NP currently has a very slow experiment/theory cycle.
- Part of the problem is how the data is analyzed.
  - We laboriously track the particles through the detectors boil it down to a few numbers.
  - Simulation does the same thing then we compare the results.

- Alternative :
  - A streaming DAQ generates a rich multidimensional data set.
  - Simulation does the same thing.
  - Compare patterns in simulated and real data directly using AI technology.

- Analogy : X-ray crystallography.
  - You don't ray trace every x-ray through a crystal.
  - You compare diffraction patterns to determine structure.



Jefferson Lab

Summary

- Many science fields use the streaming data model for data acquisition and NP is moving in this direction.
- We would like to use the same model for data processing.
- Critical areas for R&D are:
  - Timing – accurate determination of when data was generated.
  - Data Transport – true plug-and-play high performance transport of streams of data with guaranteed quality of service.
  - Data Storage – Store multi-dimensional datasets efficiently.
  - Data processing – Transition from monolithic Apps to services processing streams.
  - Integration with AI.
- I believe all of these areas would be of common interest.