Jefferson Lab

EPSCI

Experience From the 12GeV Science Program

David Lawrence - JLab

Oct. 1, 2020

Data Acquisition : CODA

Historically

- Triggered systems
  - Signal splitters w/ delay cables
  - Trigger Latency: few x 100ns
  - L2 (fast clear)
  - Manually wired + Look Up Tables (MLU modules)

12GeV

- Triggered systems
  - Flash ADC - digitally split
  - Trigger Latency: few x 1μs
  - FPGA programs (e.g. VTP)
    - More customizable triggers (*experts required*)

Future

- Streaming Readout
  - Flash ADC
  - Trigger Latency: ∞ (limited by memory + disk space)
  - Whole event triggering w/ full reconstruction
  - Deadtime-less*

Calibration Triggers

- GlueX implements several non-physics triggers dedicated mostly to detector calibrations
- Impractical to run through full multi-PB dataset to process small fraction of events
- "Skim files" produced in single pass

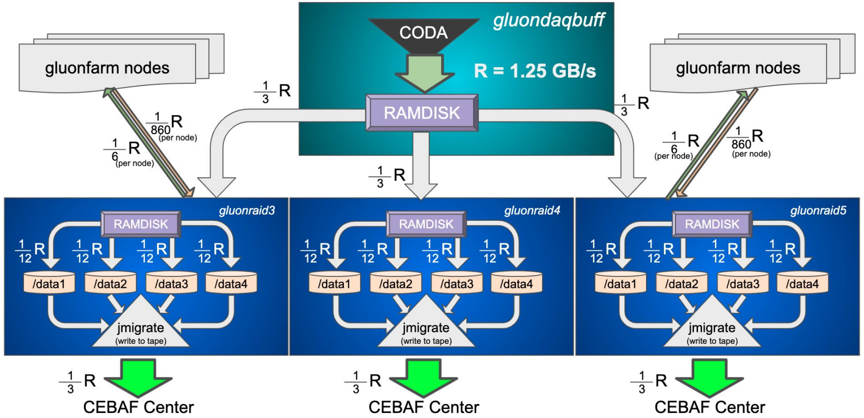
hd_rawdata_071783_337.evio	**20 GB**
hd_rawdata_071783_337.BCAL-LED.evio	6.8 MB
hd_rawdata_071783_337.CCAL-LED.evio	0.3 MB
hd_rawdata_071783_337.FCAL-LED.evio	7.1 MB
hd_rawdata_071783_337.DIRC-LED.evio*	1.7 MB
hd_rawdata_071783_337.ps.evio*	69.7 MB
hd_rawdata_071783_337.random.evio	20.1 MB
hd_rawdata_071783_337.sync.evio	0.4 MB
hd_root_tofcalib_071783_337.root	12.3 MB
TOTAL	118.4 MB

HOSS

(Hall-D Online Skim System)

Solves two big problems

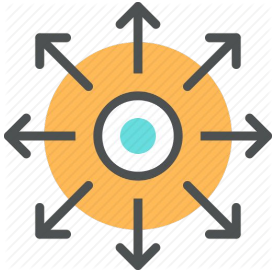
- Distributes raw data over multiple RAID partitions
- Relieves pressure on tape system



Calibration Triggers

HOSS is a great solution, but not the best solution

<u>Future:</u> Follow model of other high volume experiments:



- Distribute events from different triggers to different files at DAQ level
  - rare triggers in particular
- Triggers with large overlap can be written to same file

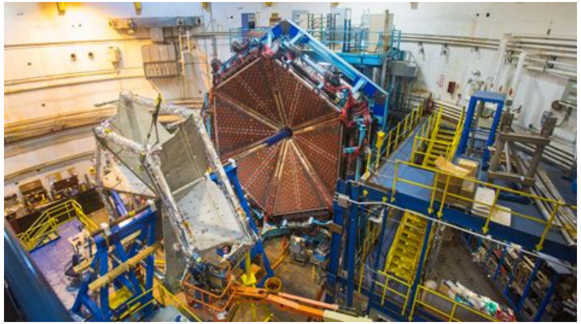
Collaborations at JLab in 12GeV era

HEP-like ⟷ **(classic) NP-like**

Hall-D

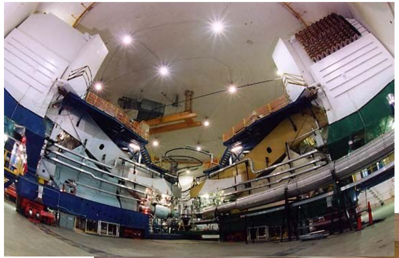
very homogenous data taking with little change to detector configuration

Hall-B

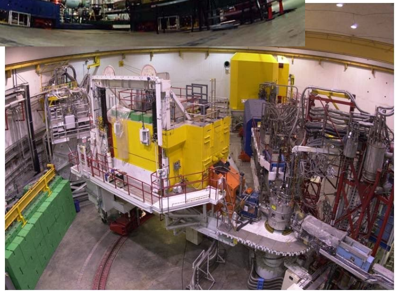


federation of experiments run in large/small run groups

Hall-A



Hall-C



Many independent experiments with frequent configuration changes

Offline Software

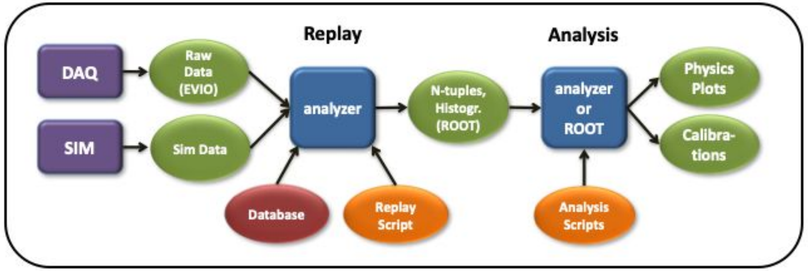
- Each experimental Hall developed its own reconstruction software framework

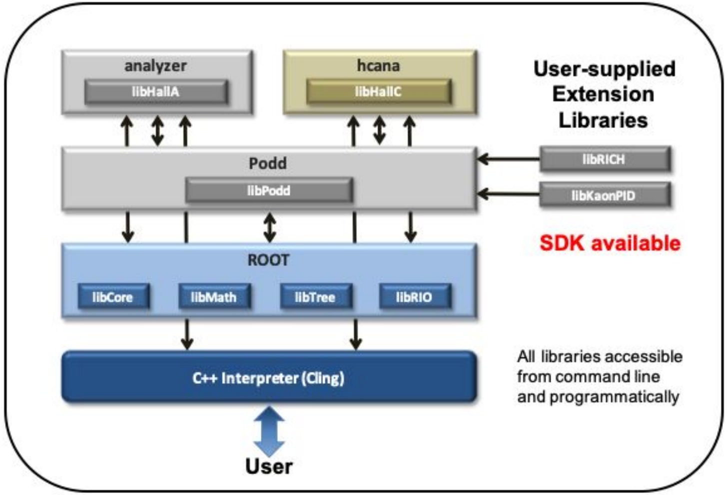
- General consensus that there should be collaboration

- Somehow, there was actually very little code sharing
    - *is there a word for this phenomenon?*

Hall A Analyzer/Hall C hcana

- C++ class library built on top of ROOT.
- In production use since 2003 (Hall A); 2012 (Hall C).
- Primarily used for reconstruction and calibration.
- Features
  - Modular. Easily accommodates changing experimental setups.
  - Run-time configurable through ROOT scripts and text files (no recompilation).
  - Light-weight. Minimal dependencies. Small memory footprint.
- One-shot processing: Raw or sim data → n-tuples & histograms in ROOT files. (Object writing possible, too.)
- Rapid development of experiment-specific plugins through user-friendly Software Development Kit.
- Actively maintained: https://github.com/JeffersonLab/analyzer





from Veronique Ziegler

CLAS12 Reconstruction and Analysis Framework (CLARA)

- Glues together isolated, independent micro-services with reactive resource allocation
- Each service runs a unique algorithm, communicating with each other through a message passing mechanism (data banks) to serve data processing goals
- Provides multithreading with horizontal and vertical scaling, error propagation and fault recovery
- Provides relevant live performance measures & supports CLAS12 on JLab batch farm, multicore environments, future diverse hardware

CLAS12 Reconstruction Tools

- Common tools, e.g. I/O interfaces, geometry, framework, & analysis utilities
- Reconstruction engines, monitoring and analysis services as plugins to CLARA

https://github.com/jeffersonlab/clas12-offline-software
– master/development branches for organization
– issue tracking, automatic Travis build with validation tests

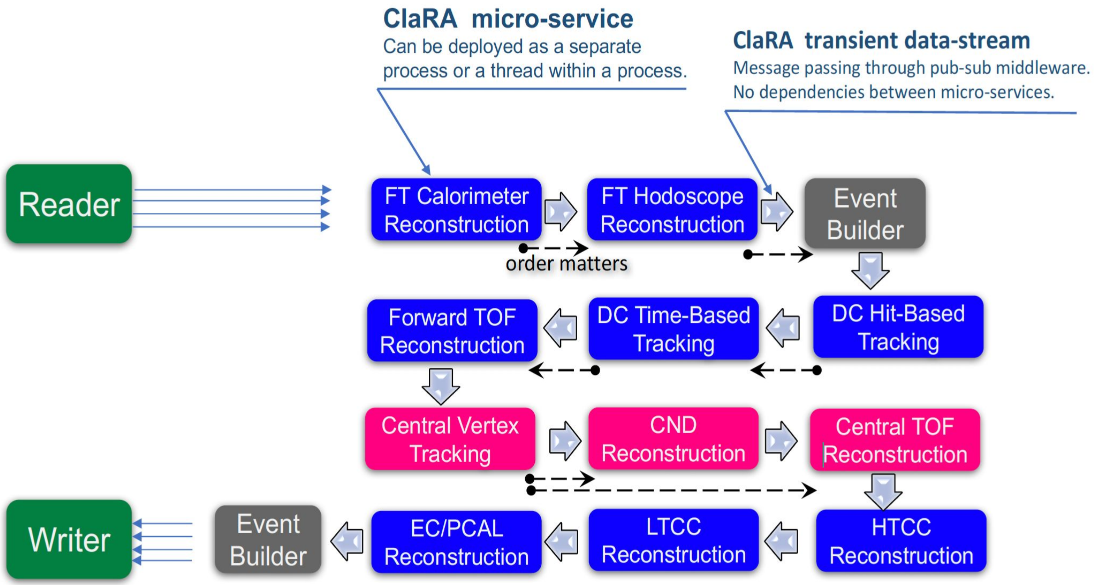
CLAS12 Data Formats

- Random access, on-the-fly high/fast LZ4 compression, no size limit
- Internal dictionary describing data structures
- Provides for easy bank filtering and event tagging mechanism (DST making and reading)

from Veronique Ziegler

CLAS12 Event Reconstruction Service Composition

- Each detector reconstruction component is a ClaRA service.
- Event building services (EB) combines info from individual services output banks to reconstruct particle candidate.



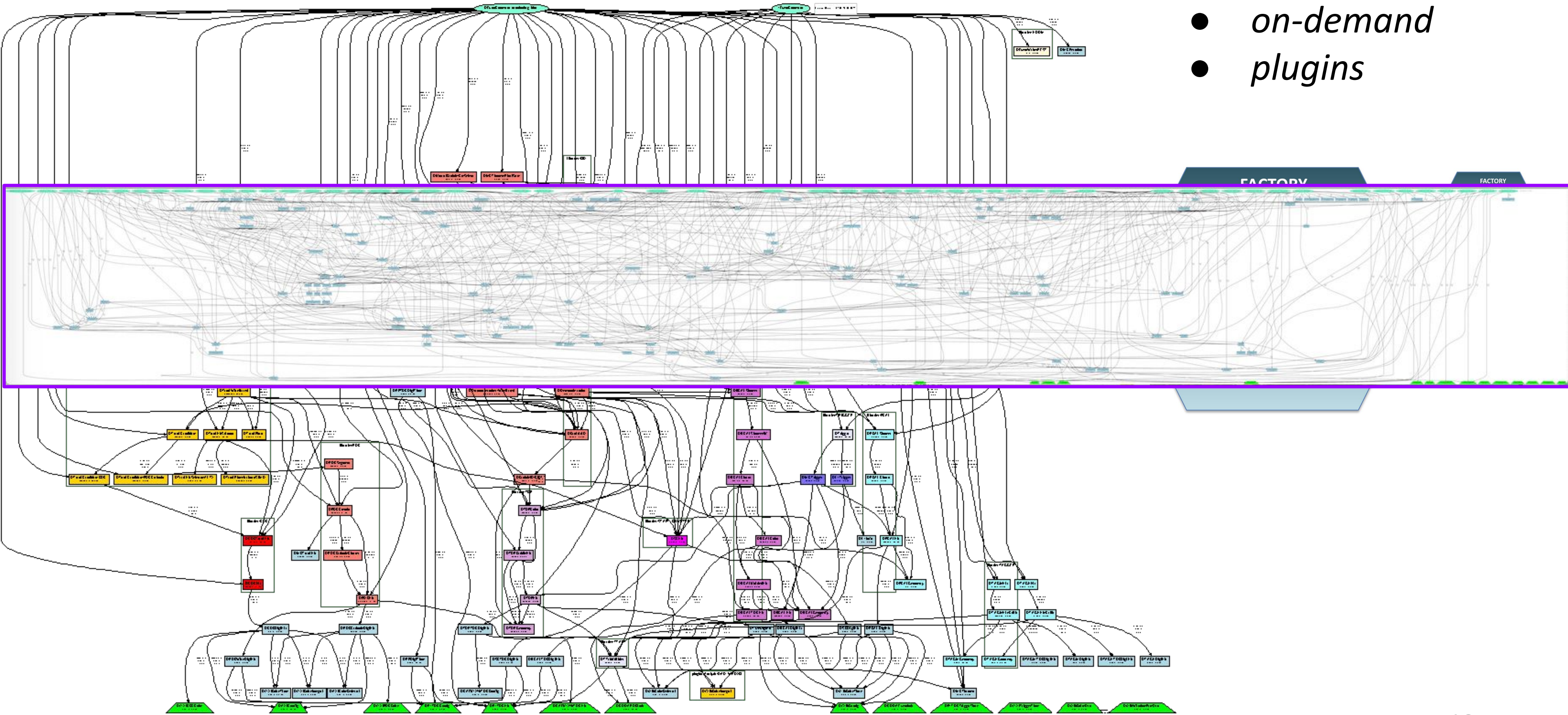
Data Processing Workflow

- Decoding to CLAS12 data format, implementing translation tables and fADC pulse analysis.
- Event reconstruction using detector-specific packages running in CLARA and producing DSTs.
- Use analysis trains to skim different event topologies and produce separate reconstructed event files.
- Skimmed files distributed to users for physics analysis.

GlueX Reconstruction Software

JANA C++ Framework

- *multi-threaded*
- *on-demand*
- *plugins*



Offline Software

	Hall-A/C (Analyzer)	Hall-B (CLARA)	Hall-D (JANA)
PROS	• Compact and streamlined user configuration • ROOT is familiar platform to all users	• Loose coupling allows horizontal scaling and easy unit testing • Java avoids a host of memory issues that plague C/C++	• Tight coupling allows highly performant code • On-demand interface makes it easy to customize jobs for mon./L3 trig./recon. through choice of plugins
CONS	• Lack of multi-threading limits scalability • Limited by ROOT limitations(?)	• Choice of Java controversial within the collaboration (sociological) • JVM Heap memory allocation *(does not play nice with other processes)*	• Low-level memory access can lead to very insidious bugs • User must use locks correctly for some common tasks (e.g. ROOT)

Data Quality Monitoring

Jefferson Lab

EPSCI

Hall-A

- "online-replay" system run manually by shift crew after each run (30-60min)
- uncalibrated reconstruction
- dedicated "onlineGUI" for viewing ROOT files + reference plots

Hall-B

- occupancy histograms filled live from ET (DAQ) system
- (semi?)-automated entries in e-log
- Reference plots with side-by-side comparison

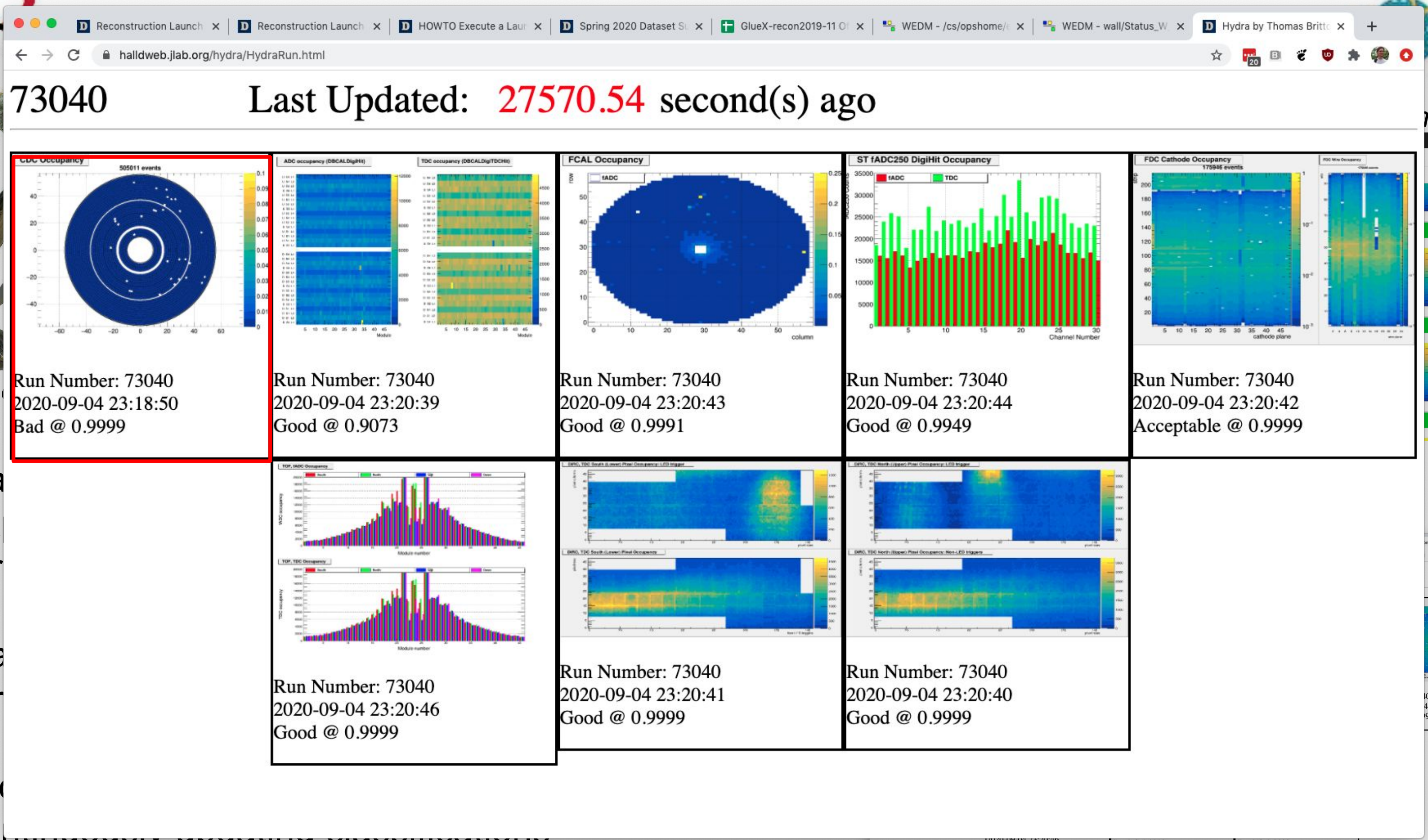
Hall-C

- live system fed by events from ET (DAQ) system
- web-based viewer based on new ROOT features
- comparison with reference run

Hall-D

- "RootSpy" reads from ET system (optional modes reads from raw data files)
- occupancy and full event reconstruction using most recent calibrations
- semi-automatic logbook entries (shift crew must push button)
- Root-based viewer application w/ reference histograms
- time-series DB entries (e.g. π^0 mass vs. time)
- "incoming data" automated farm jobs for first 5 files of run -> Plot-Browser
- and then there's Hydra ….

Koboldpress.



nages

- Ea
  ex
  alr
- Ca
  a r
- Gl
  co... continuously updating classifications

So many more things ….

- Simulation
- Offsite processing (OSG, NERSC, …)
- Analysis trains
- Containers
- Event viewers
- Calibration & Conditions DB
- Web-based tools
  - Submitting simulation jobs
  - User specified reactions for analysis
- Jupyterhub
- AI, ML, AI, ML, AI, ML, ….

Final Thoughts

- The 12GeV program has seen enormous growth in the software and computing being implemented at JLab

- We need to continue that momentum to keep developing new tools and integrating new technologies
    - For the betterment of the 12GeV Science program
    - To be prepared for the next generation of experiments

- What we should be doing right now:
    - We need to keep moving towards automating calibration online as much as possible to reduce backlog in data processing
        - CEBAF is trying to move 33+ weeks of running per year
    - We need to automate data quality monitoring as much as possible
    - Make AI/ML as common a tool as **TH1D::Fit()** or **jcache get**