Jefferson Lab 12-GeV Experimental Computing Review
November 27-28 2018

CLAS12 Software and Computing Overview

Raffaella De Vita (INFN)
for the CLAS Collaboration

Outline

- CLAS12 experiment and collaboration
- Software & computing:
  - Organizational structure
  - Task prioritization and planning
  - Collaboration involvement
- Software architecture:
  - Detector simulation
  - Reconstruction framework and packages
  - Common tools
  - Analysis workflow and tools
- Summary

Addresses charge items 1.a, 1.e, 1.f and 2.c

CLAS12

Forward Detector (FD)

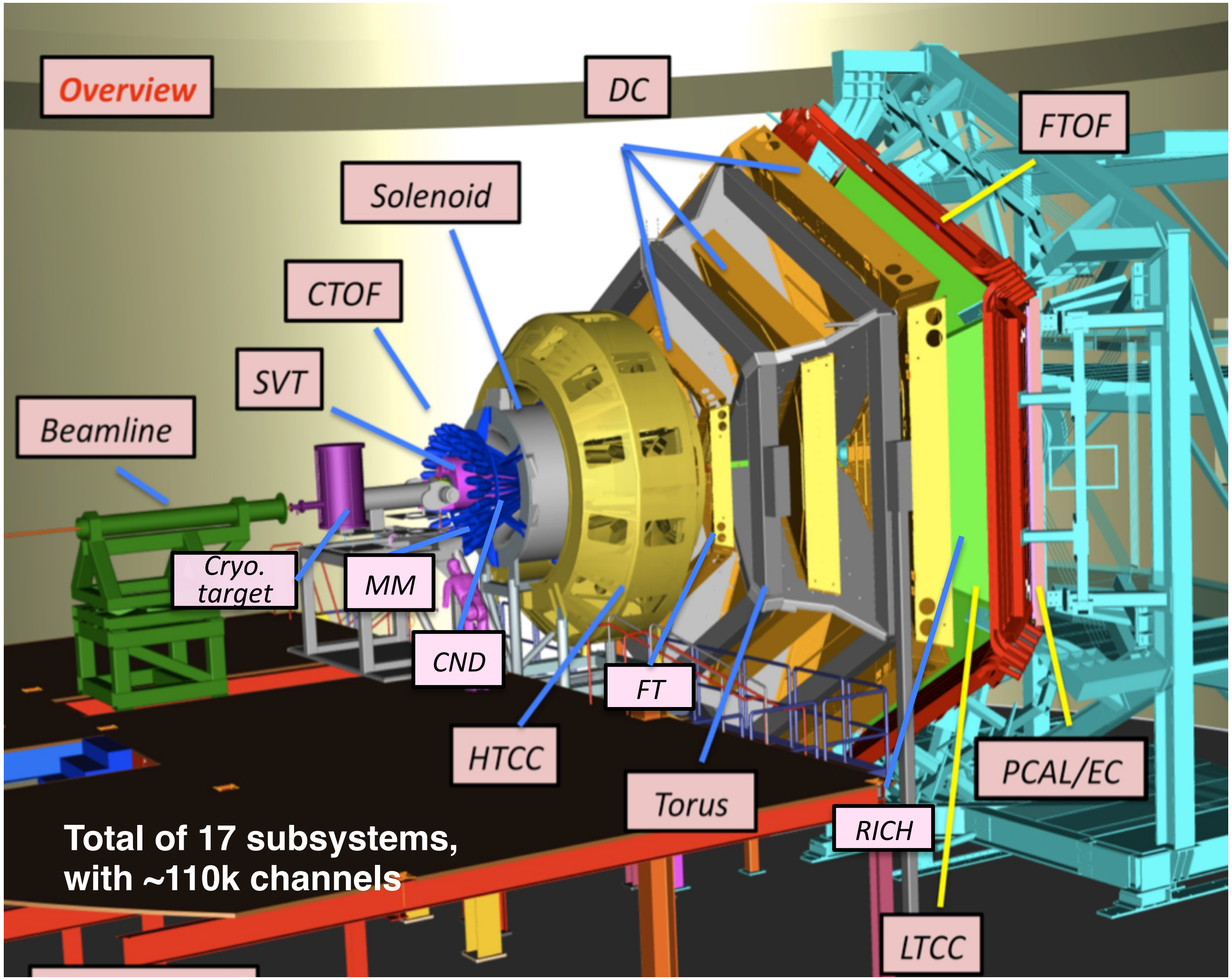
- TORUS magnet
- High Threshold Cherenkov Counter (HTCC)
- Drift chamber system (DC)
- Low Threshold Cherenkov Counter (LTCC)
- RICH detector
- Forward Time-of-Flight System (FTOF)
- Pre-shower calorimeter (PCAL)
- E.M. calorimeter (EC)

Central Detector (CD)

- SOLENOID magnet
- Silicon Vertex Tracker (SVT)
- MicroMegas (MM)
- Central Time-of-Flight (CTOF)
- Central Neutron Detector (CND)

Others

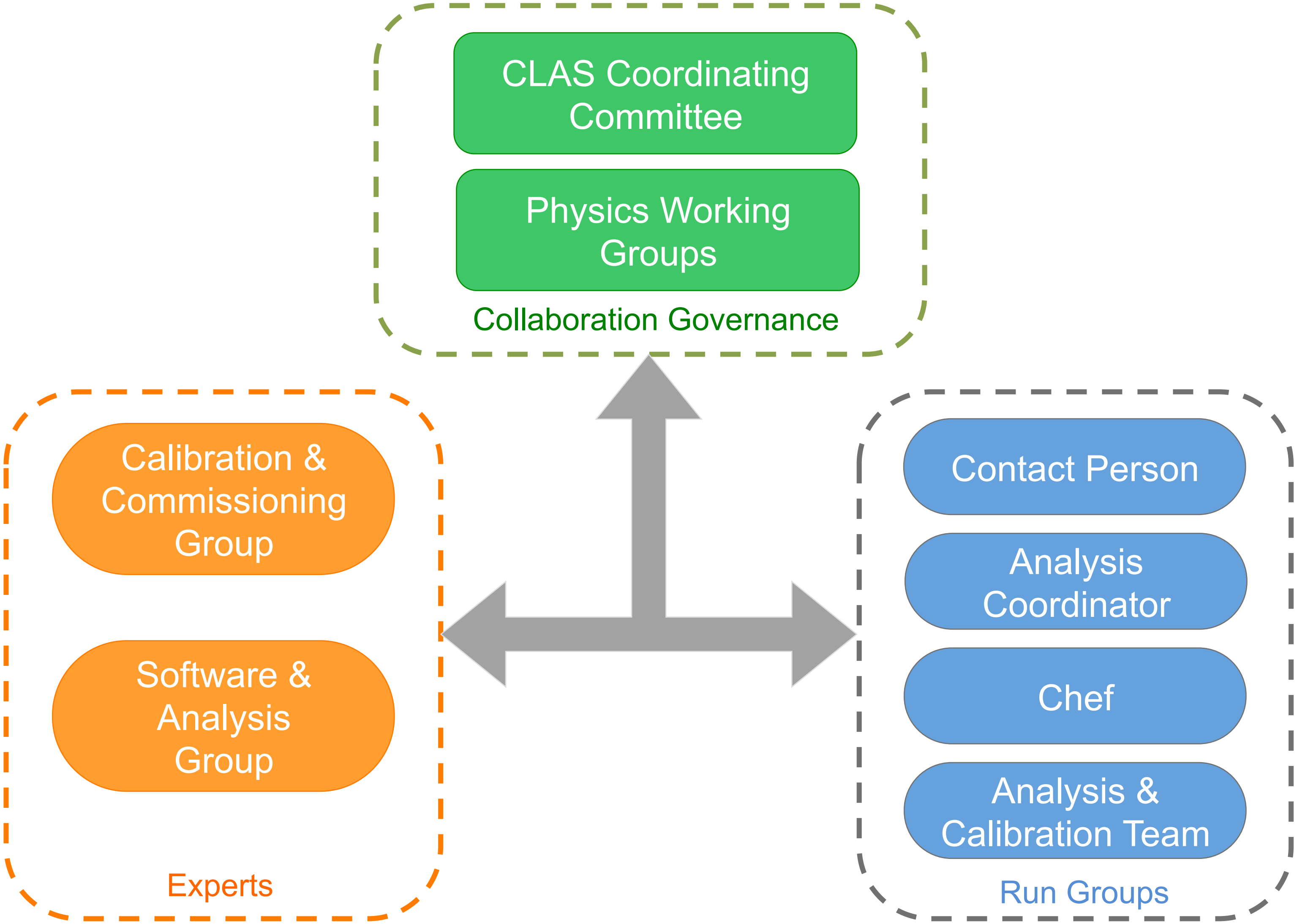
- Forward Tagger (FT)
- Beamline
- Cryo Target
- Moller polarimeter



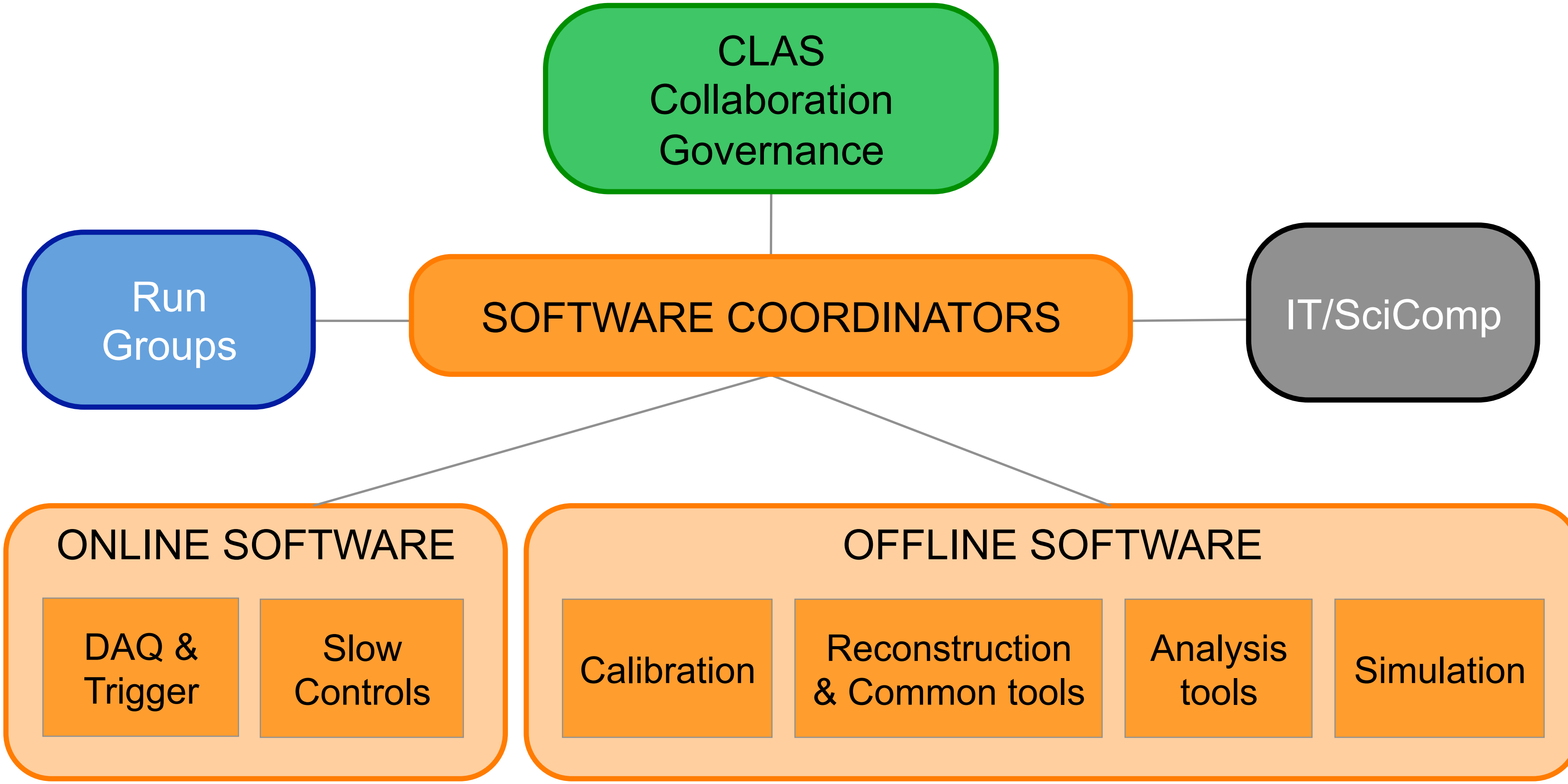
CLAS12 run groups

Run Groups	Number of experiments	Beam time (PAC-days)	Luminosity (per nucleon)	Triggers
A	13	139 (100)	10^{35} cm^{-2}s^{-1}	e^-,2μ,e^{FT},$2H^{OP}$
B	7	90	10^{35} cm^{-2}s^{-1}	e^-,2μ
C	6	180	$2x10^{35}$ cm^{-2}s^{-1}	e^-
D	1	60	$2x10^{35}$ cm^{-2}s^{-1}	e^-
E	1	60	$2x10^{35}$ cm^{-2}s^{-1}	e^-
F	1	42	$4x10^{34}$ cm^{-2}s^{-1}	e^-
G	1	55	$2x10^{35}$ cm^{-2}s^{-1}	e^-
H	3	110	10^{34} cm^{-2}s^{-1}	e^-
K	3	100	10^{35} cm^{-2}s^{-1}	e^-
L	4	55	10^{35} cm^{-2}s^{-1}	e^-
M	2	45	$2x10^{34}$ cm^{-2}s^{-1}	e^-

CLAS collaboration structure



CLAS12 software organization



Software development organization

- Task prioritization and assignment based on overall experiment priorities and input from Run Groups
- Focus over the last months on:
  - Support of preparations for the Fall-Winter data taking
  - Improvement/update of reconstruction software
  - Tuning/speed-up/portability of simulation software
  - Development of analysis framework and tools
  - Support of the Spring 2018 data processing (calibration, reconstruction, analysis, …)
- Weekly organizational meetings with software systems leaders and key developers among collaborators and staff:
  - Work planning
  - Progress monitoring
- Weekly meetings with whole software group
  - Present new results
  - Discuss issues and find solutions
  - Inform users of progress and news
- Software sessions and workshops at Collaboration Meetings

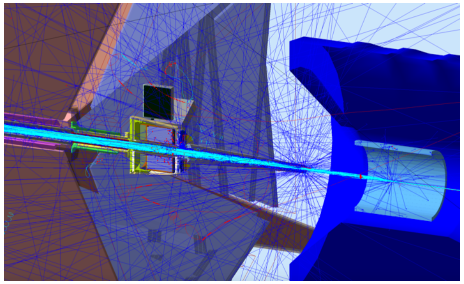
Collaboration involvement

- Support data processing and calibration:
  - Run group analyses coordinators and chefs
  - Calibration team
- Contribute to software development:
  - Detector geometry and digitization in simulations
  - Reconstruction packages
  - Calibration and monitoring suites
  - Common tools
  - Analysis tools
- Contribute to software validation:
  - Development of validation tools
  - Ad-hoc validation of reconstruction algorithms
  - Benchmarking of new releases
- Effort in strengthening user involvement via service-work tasks

Recent SW tasks:
- Extension of tracking tests in reconstruction validation suite
- Validation of TOF reconstruction
- …

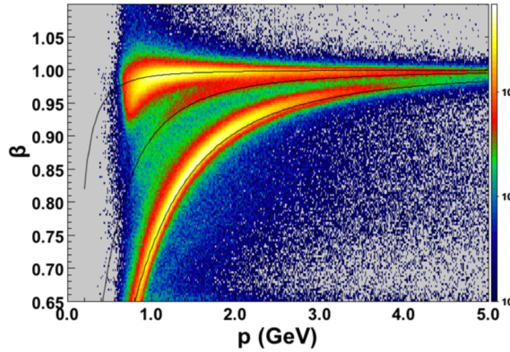
CLAS12 offline software

- Simulations
- Reconstruction framework and packages
- Common tools
- Analysis workflow and tools

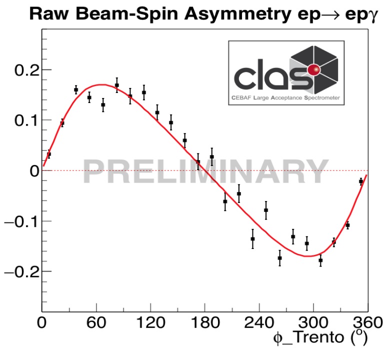


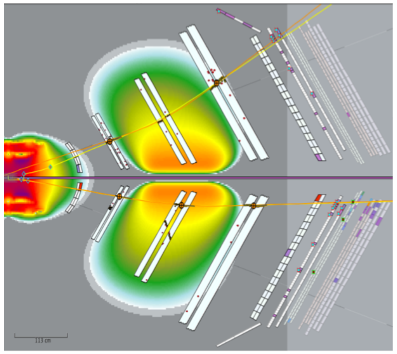
Beam background simulation in GEMC

Hadron identification based on Forward Time of flight



Example of analysis result: beam-spin asymmetry in DVCS



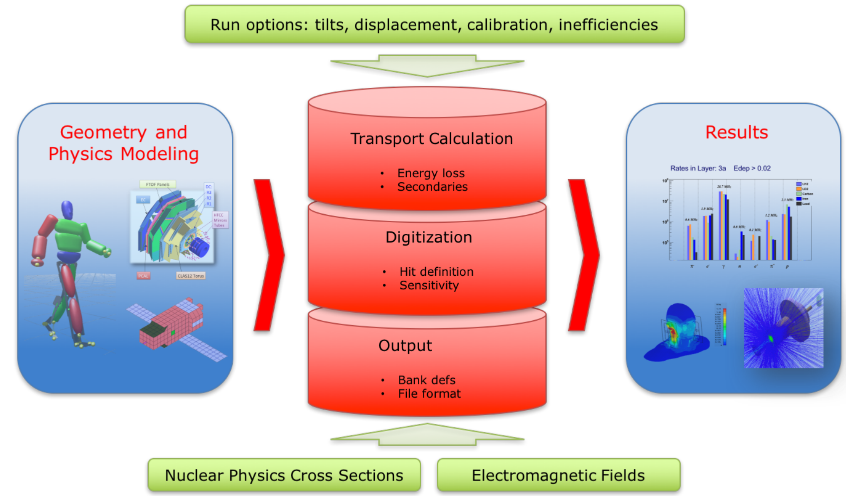


Visualization of Forward Detector tracking results

CLAS12 simulation: GEMC

GEant4 Monte-Carlo

- Geant4-based C++ framework to simulate the passage of particles through matter
- Provides:
  - application independent geometry description
  - cad/gdml imports
  - standard GEANT4 geometries stored in sql database, CAD STL geometry also supported, with easy visualization interface

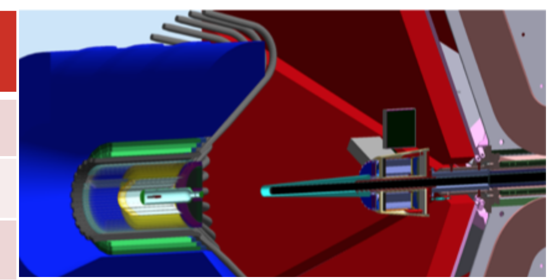


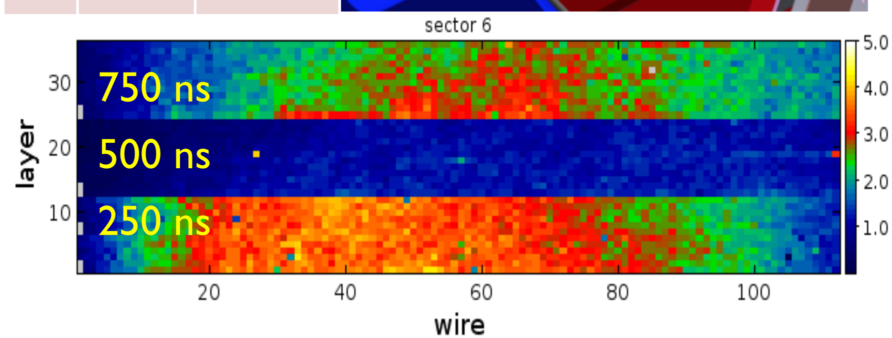
The architecture of gemc

GEMC for CLAS12:

- Incorporates digitization algorithms for all CLAS12 detectors
- Proven to be very accurate representation, regarding background rates and occupancies, and critical for beamline shielding studies
- Support background merging based on real data recorded from random triggers to study efficiency/resolution as a function of luminosity
- Provides multiple levels of accuracy vs. speed (full, or no-secondaries, or acceptance-only)
- Currently: 200-500 ms/event (single core on a farm14/16 node) for multi-particle final states with full transport and digitization
- Docker distribution for easy offsite deployment

	Data	GEMC
R1	2.8 %	2.68 %
R2	0.6 %	0.76 %
R3	1.5 %	1.18 %





CLAS12 reconstruction framework

ClaRA Framework

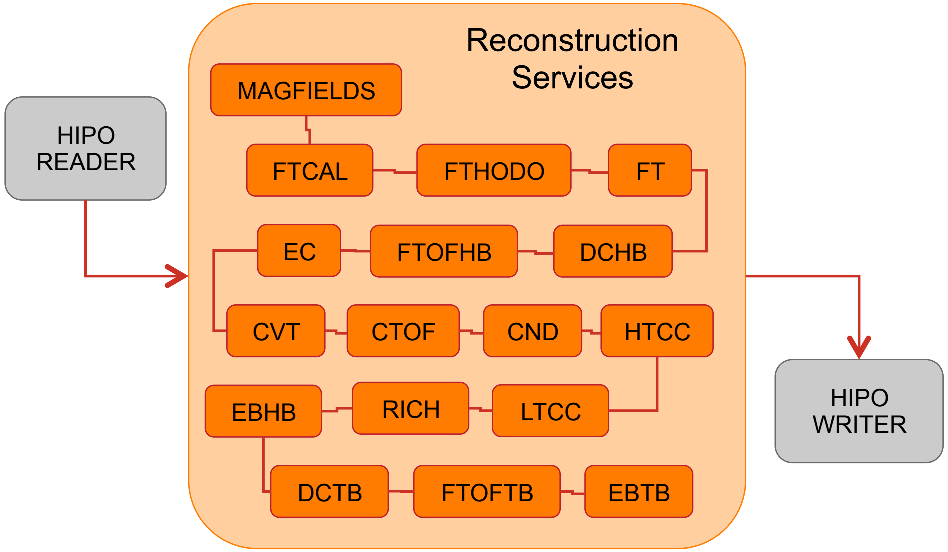
- **CLA**S12 **R**econstruction and **A**nalysis Framework
- glues together isolated, independent micro-services with reactive resource allocation
- each service runs a unique algorithm, communicating with each other through a message passing mechanism (data banks) to serve data processing goals
- provides multithreading with horizontal and vertical scaling, error propagation and fault recovery
- supports CLAS12 on JLab batch farm, multicore environments, diverse hardware
- https://claraweb.jlab.org/clara/

HIPO data format

- random access, on-the-fly high/fast LZ4 compression, no size limit
- chunked indexed file structure for fast reading
- internal dictionary describing data structures
- provides for easy bank filtering and event tagging mechanism (DST making and reading)

COATJAVA

- 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 real validation tests



CLAS12 offline analysis tools

COATJAVA:

- **Input/Output:**
  - raw data in EVIO format with composite bank structure
  - HIPO format (record based and compressed)
  - raw data decoder
  - common interfaces to read EVIO, BOS, HIPO files
  - data processing interface (format agnostic) for calibration and monitoring
- **Databases:**
  - calibration constants and geometry definitions (CCDB, SQLite)
  - caching algorithm of database constants for reconstruction and calibration
- **Geometry:**
  - geometry package used by simulation, reconstruction and event display
  - ability to import CAD files (GEANT4 tessellated shapes)
  - detector visualization package with callbacks and automated occupancy display
- **Reconstruction:**
  - reconstruction engine class to develop CLARA reconstruction services
  - automatic detector initialization from database (run dependent)
- **Analysis tools:**
  - event selector
  - fiducial cut provider and fast-mc
- **Plotting package**:
  - based on groot library (https://github.com/gavalian/groot)
  - freehep-jminuit library for fitting

CLAS12 offline analysis tools

COATJAVA:

CLAS12 specific development
Jlab development
External library

- **Input/Output:**
  - raw data in EVIO format with composite bank structure
  - HIPO format (record based and compressed)
  - raw data decoder
  - common interfaces to read EVIO, BOS, HIPO files
  - data processing interface (format agnostic) for calibration and monitoring
- **Databases:**
  - calibration constants and geometry definitions (CCDB, SQLite)
  - caching algorithm of database constants for reconstruction and calibration
- **Geometry:**
  - geometry package used by simulation, reconstruction and event display
  - ability to import CAD files (GEANT4 tessellated shapes)
  - detector visualization package with callbacks and automated occupancy display
- **Reconstruction:**
  - reconstruction engine class to develop CLARA reconstruction services
  - automatic detector initialization from database (run dependent)
- **Analysis tools:**
  - event selector
  - fiducial cut provider and fast-mc
- **Plotting package**:
  - based on groot library (https://github.com/gavalian/groot)
  - freehep-jminuit library for fitting

Calibration & Monitoring

Common Calibration Framework

- in COATJAVA, used for all CLAS12 detectors
- utilizes COATJAVA CCDB interfaces, detector visualization, data processing interface, …
- provides GUI fitting, plotting, display utilities (via groot) for extracting and checking calibration parameters, along each stage of the calibration sequence

Procedures

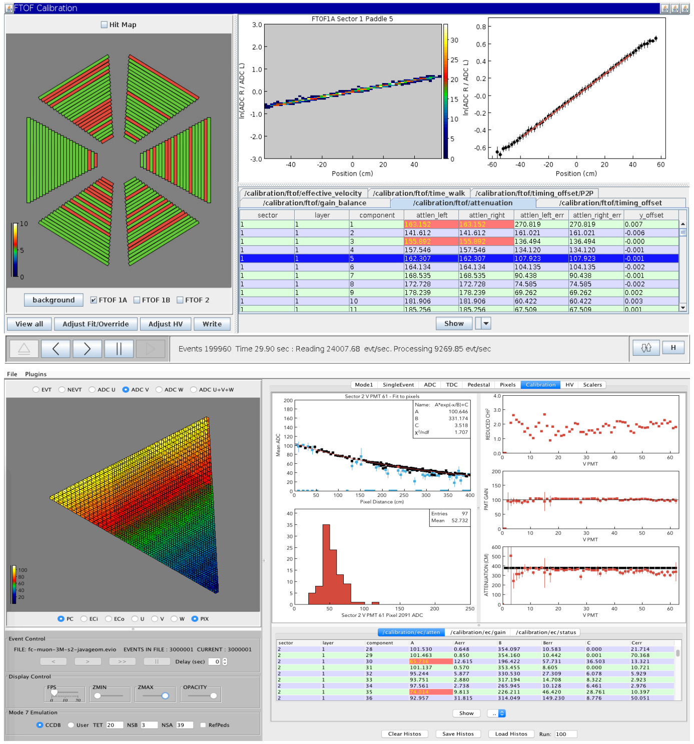
- workflow and dependencies well understood and tested, first through calibration challenges and now with real data

HIPO bank filtering

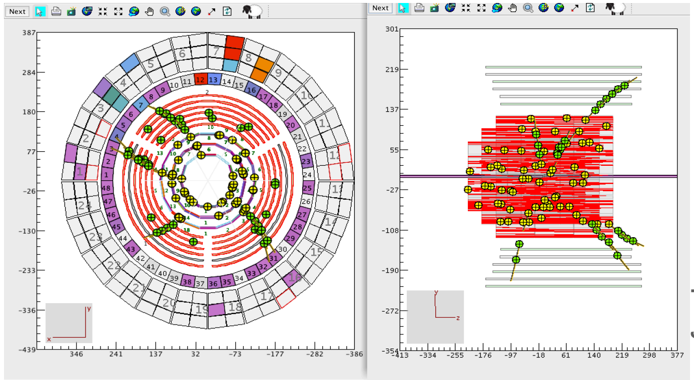
- easily generate slim skims and bank filters, at least 100X reduction, for each detector’s specific calibration needs

ClaRA+COATJAVA Service Orientation

- for fast iterations to support individual detector’s calibrations, e.g. reprocessing single service



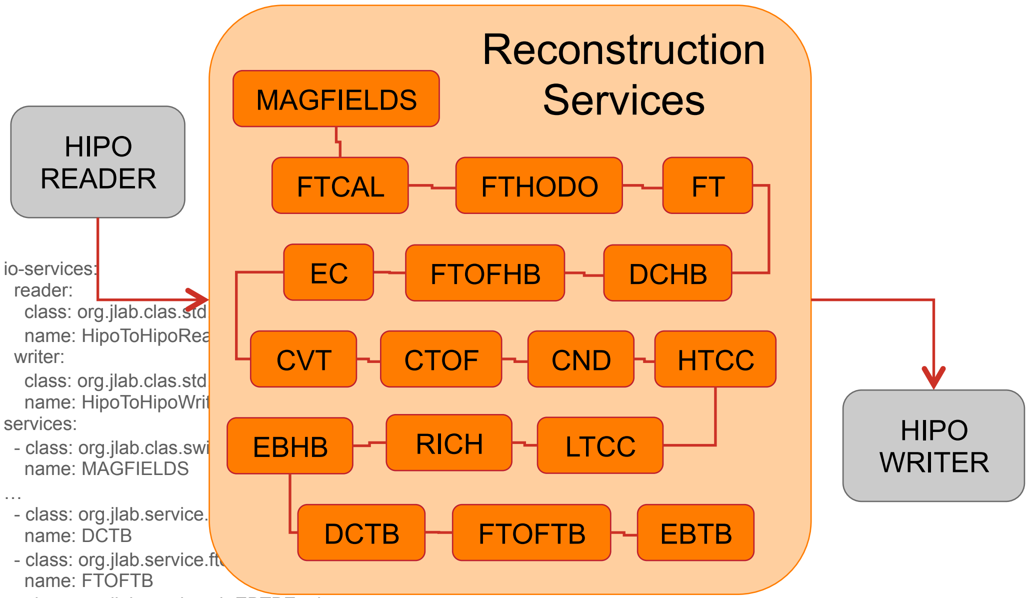
FTOF and ECAL calibration suites



CLAS12 Event Display

Reconstruction services

- Java based services for all CLAS12 detectors, event builder, I/O
- Extend COATJAVA reconstruction engine
- HIPO banks input and output
- Automatic detector configuration and calibration constants caching from database
- Can be “chained” for event reconstruction in CLARA depending on specific application (full CLAS12 reconstruction, single service reconstruction for calibration, …)
- Configurable via YAML file (database variation and timestamp, specific parameters, …)



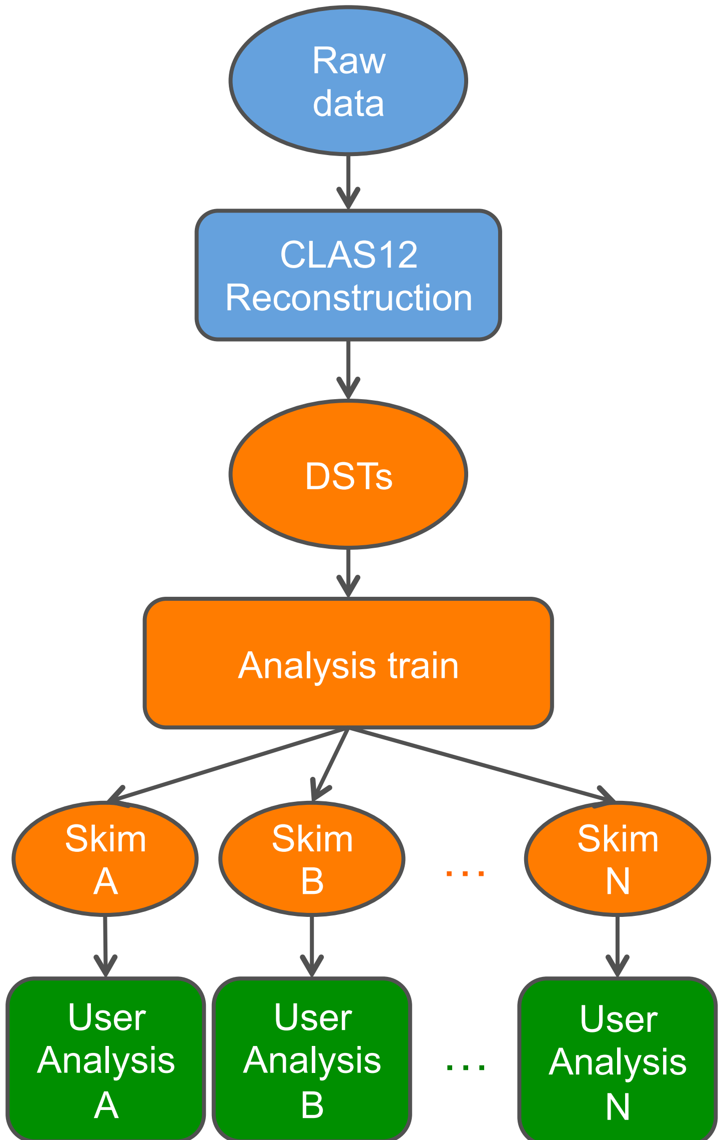

```
io-services:
  reader:
    class: org.jlab.clas.std
    name: HipoToHipoRea
  writer:
    class: org.jlab.clas.std
    name: HipoToHipoWrit
services:
  - class: org.jlab.clas.swi
    name: MAGFIELDS
…
  - class: org.jlab.service.
    name: DCTB
  - class: org.jlab.service.ft
    name: FTOFTB
  - class: org.jlab.service.eb.EBTBEngine
    name: EBTB
configuration:
  io-services:
    writer:
      compression: 2
services:
   MAGFIELDS:
      solenoidMap: Symm_solenoid_r601_phi1_z1201_13June2018.dat
      torusMap: Symm_torus_r2501_phi16_z251_24Apr2018.dat
      solenoidShift: "-1.9"
   DCHB:
      timestamp: 08/22/2018
      useStartTime: "true"
      wireDistortionsFlag: "true"
      geomDBVariation: may_2018_engineers
    DCTB:
      geomDBVariation: may_2018_engineers

mime-types:
  - binary/data-hipo
```


Example of YAML file and corresponding graphical representation

Analysis workflow and tools

- Input: **CLAS12 DSTs** produced by reconstruction chain: https://clasweb.jlab.org/wiki/index.php/CLAS12_DSTs
  - reconstructed particles
  - detector specific information to refine particle identification and selection
  - scaler information for absolute normalization
- Use **analysis trains** for event skimming and "standard" analysis steps to optimize resources usage
- Analysis trains run periodically and skims are distributed to users for **high-level analysis**
- **Analysis tools**:
  - analysis library in JAVA for accessing detector specific information from the event, basic topology selection and final state identification
  - framework for standard data corrections and pid cuts, fiducial cuts and Fast-MC for physics analysis
  - standard code for converting DST's to ROOT (with collaboration support)
  - tools in C++/ROOT for common data analysis environment (with collaboration support)



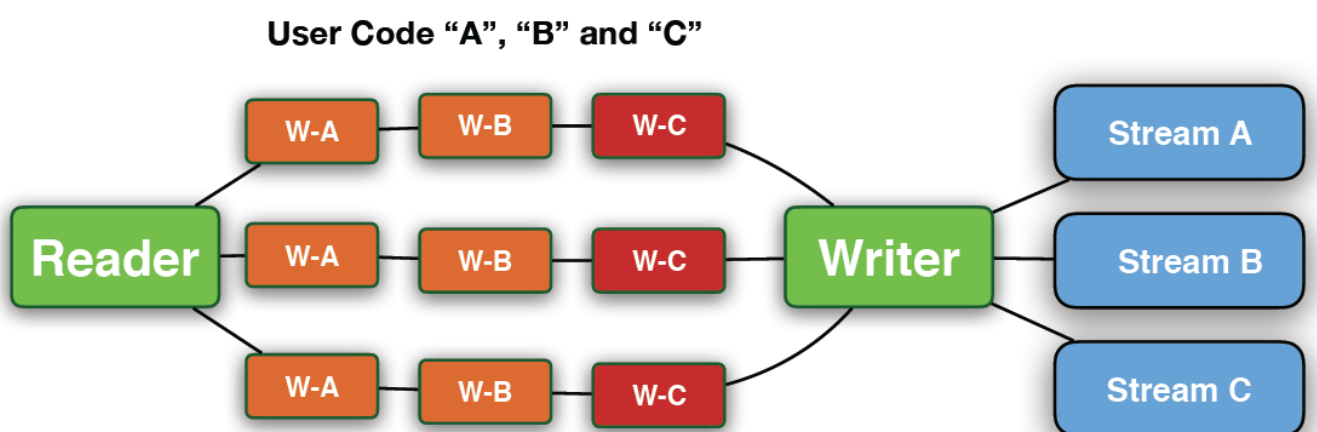
Analysis trains

ClaRA analysis services trains:

- Run on reconstruction output (DSTs) producing multiple skimmed DSTs
- Periodically running with analyzers' services plugged-in, to optimize simultaneous access to data by many analyses (Test-Trains, Mini-trains, Maxi-Trains)
- Fault isolation between services, i.e. one users' service will not affect the others
- Wagons:
  - generic wagon for standard event selection based on event topology and kinematics,
  - custom wagon for ad-hoc event filtering or corrections
- Administrative limits set on output size of each wagon, total not more than ~50% of input

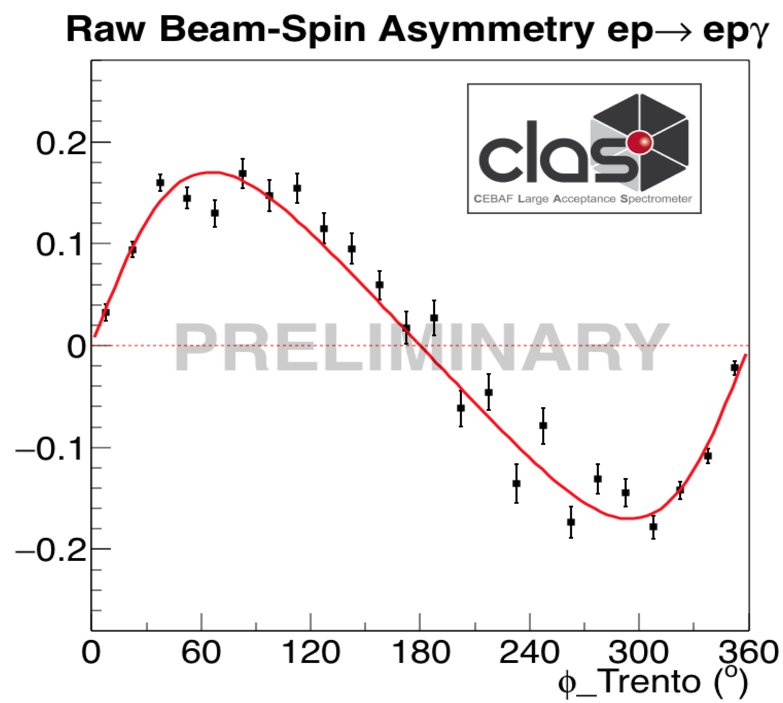
Typical file sizes:

- DST: <5% of raw data
- Skims: typically few % of full statistics, total skim size <50% of DSTs
- For example: RG-A Spring18 has 1 PB raw data, 50 TB DSTs, less than 15 TB skims



Physics analysis

- First public presentation of CLAS12 analysis at DNP2018 based on 10% of RG-A spring data
- Full “exercise” of calibration/reconstruction/analysis chain:
  - About 2×10^9 events reconstructed producing DSTs (5TB)
  - DST skimmed with analysis trains with multiple wagons:
    - Inclusive electrons for DIS and SIDIS analysis
    - Electron-proton for DVCS analysis
    - J/psi and TCS
    - Two-pion photo-production
    - Single-pi0 photo-production
    - $e\pi^+n$ for neutron detection efficiency studies
    - Photon energy correction and PID χ^2 re-calculation
  - Skims (1.5 TB) distributed to users and analyzed with Java or C++/Root tools to extract first observables for different reactions



First look at Deeply virtual Compton scattering at CLAS12:
- Residual background not yet subtracted
- Only statistical errors
- Integrated over all kinematic domain

Summary

Raffaella De Vita (INFN)
devita@jlab.org

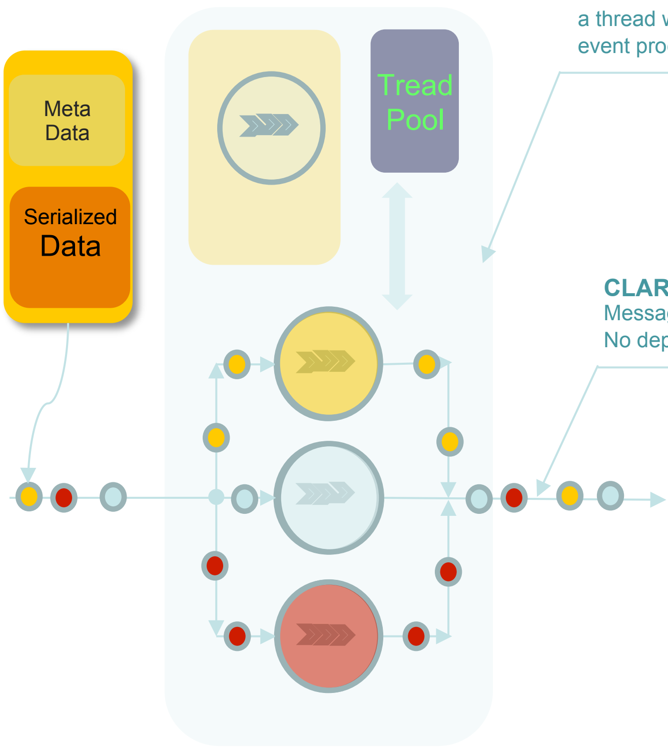
- Software and computing for CLAS12 is managed by the Software Group in coordination with the CLAS Collaboration governance, the CLAS12 Run Groups and the JLab IT division
  - Work plan and priorities discussed on a weekly basis
  - Effort in strengthening user contribution in software development and validation
- CLAS12 software:
  - GEANT4 Monte-Carlo (GEMC)
  - CLAS12 Reconstruction and Analysis framework (CLARA)
  - CLAS12 Offline Analysis Tools (COATJAVA)
  - HIPO data format
- Analysis workflow and tools:
  - Analysis trains for coordinated and efficient event skimming and first analysis
  - Toolset for high-level, user-driven physics analyses

Documentation

- GEMC: https://gemc.jlab.org/gemc/html/index.html
- CLARA: https://claraweb.jlab.org/clara/
- HIPO: https://userweb.jlab.org/~gavalian/docs/sphinx/hipo/html/index.html
- COATJAVA: https://userweb.jlab.org/~gavalian/clas12docs/sphinx/html/index.html
- CLAS12 Event Builder and DSTs:
  - https://clasweb.jlab.org/wiki/index.php/CLAS12_EventBuilder
  - https://clasweb.jlab.org/wiki/index.php/CLAS12_DSTs
- Analysis trains: https://userweb.jlab.org/~gavalian/docs/sphinx/hipo/html/chapters/analysis_train.html
- GROOT: https://github.com/gavalian/groot/wiki
- CLAS12 simulation software distribution (docker): https://clasweb.jlab.org/clas12/clas12SoftwarePage/html/index.html
- Software forum (new): https://clas12.discoursehosting.net/categories
- GitHub repositories:
  - COATJAVA: https://github.com/JeffersonLab/clas12-offline-software
  - Calibration & Monitoring:
    - https://github.com/JeffersonLab/clas12mon
    - https://github.com/JeffersonLab/clas12dc
    - https://github.com/JeffersonLab/clas12calibration-ecal
    - …
    - https://github.com/JeffersonLab/clas12calibration-tof

Backup

CLARA micro-service architecture



CLARA micro-service
Can be deployed as a separate process or a thread within a process. Multi-threaded event processing.

CLARA transient data-stream
Message passing through pub-sub middleware. No dependencies between micro-services.

About:
Generic multi-threaded framework to create software applications as a suite of independently deployable, small, modular services in which each service runs a unique algorithm, communicating with each other through a message passing mechanism to serve data processing goals. Services of the same software application can be written in different programming languages, where each service is developed, optimized and maintained independently.

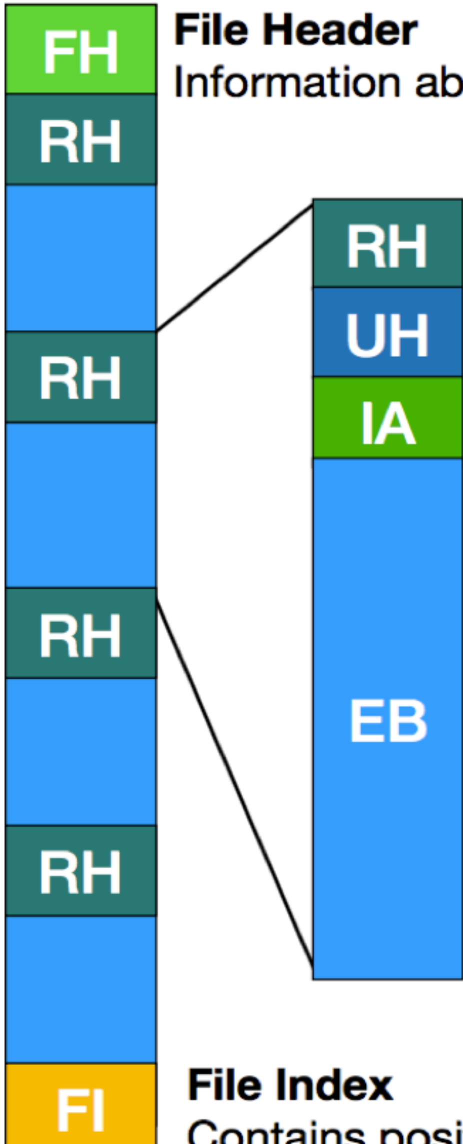
Resilient:
CLARA system stays responsive in the face of failure. Resilience is achieved by replication, containment, isolation and delegation. Failures are contained within each service, isolating services from each other and thereby ensuring that parts of the system can fail and recover without compromising the system as a whole.

Elastic:
CLARA system stays responsive under varying workload. It reacts to changes in the input data rate by increasing or decreasing the resources allocated to service these inputs (horizontal and vertical scaling). CLARA implements predictive, as well as reactive, scaling algorithms by providing relevant live performance measures. We achieve elasticity in a cost-effective way on commodity hardware and software platforms.

Robust:
CLARA uses asynchronous message-passing to establish boundaries between services that ensure loose coupling, isolation, location transparency, and provides means to delegate errors as messages. Employing explicit message-passing enables load balancing and overall data-flow, i.e. application algorithm control and orchestration.

"CLARA: A Contemporary Approach to Physics Data Processing", 2011 J. Phys.: Conf. Ser. 331 032013 doi:10.1088/1742-6596/331/3/032013
"CLARA: The CLAS12 Reconstruction and Analysis framework", 2016 J. Phys.: Conf. Ser. 762 012009 doi:10.1088/1742-6596/762/1/012009

HIPO File Format



File Header
Information about file

Record Header
Contains record properties (size, compression type, event count). Contains fields for tagging the record (up to 8 tags)

User Header
Contains information about the record dictionary, format. User specified parameters related to conditions of the experiment.

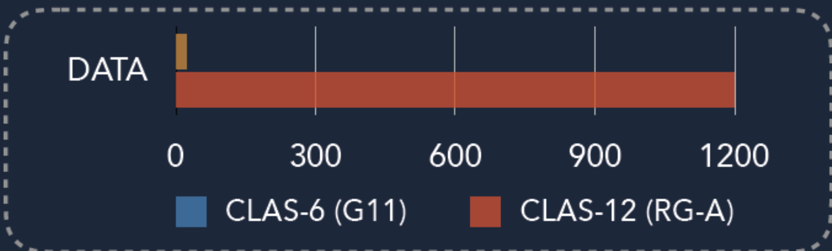
Index Array
Array of event offsets inside the event buffer. Dynamically creates event random access table.

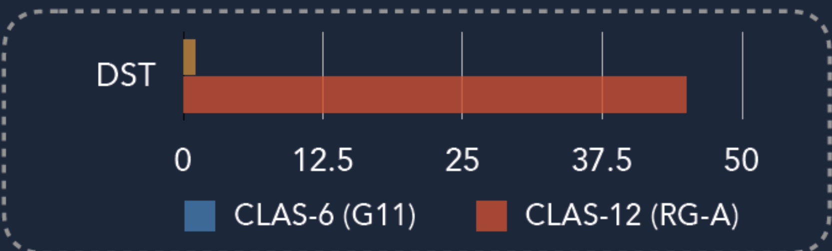
Event Buffer
Events in the record back to back. To disengage them index array is needed.

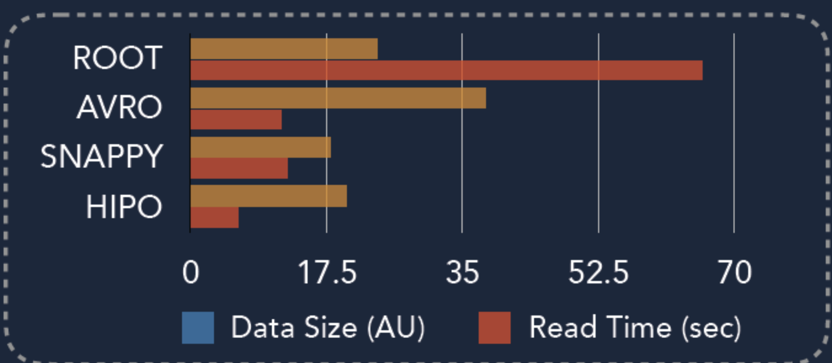
File Index
Contains positions of all records in the file

HIPO data format performance

CLAS 12 Detector







DATA SIZES:

- CLAS12 First Experiment collected ~50x more data than longest experiment in CLAS6 era (RGA~1.12 PB)
- Data Summary tapes are ~100x larger (45 TB) than G11 experiment (on the graph compressed size is shown)

HIGH PERFORMANCE OUTPUT (HIPO):

- HIPO data format was implemented for CLAS12
- Dictionary driven data format with Schema evolution.
- Fast compression algorithm (LZ4)
- Chunked data frame implementation to support multithreaded applications.
- Event chunk tagging and indexing to allow reading selective types of events from Data Summary Tapes.
- Deserialization performance better than most commonly used data formats in Nuclear Physics.
- Faster than industry standard data types.

2

HIPO data format (V4)

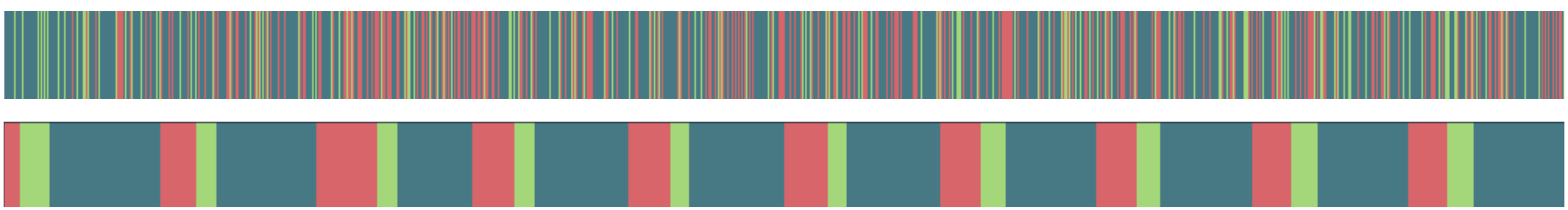
File Format

- Fast compression algorithm (LZ4)
- Chunked indexed file structure for fast reading
- Tagging of event chunks allows random access to relevant data
- Random access to any event in the stream
- Automated filtering, merging and search algorithms through the file

Event Format

- Data Frame support for multi-threaded processing
- No copy event reading, very low memory footprint
- Support of tabular banks with relation resolution
- Schema driven data description
- Many to one relation of banks resolved at run time

EVENT TOPOLOGY FROM CLAS12 (RUN #3856)



59.7% Trigger particle is not an electron.
No electron Forward Tagger.

25.6% Electron trigger.
Forward Detector

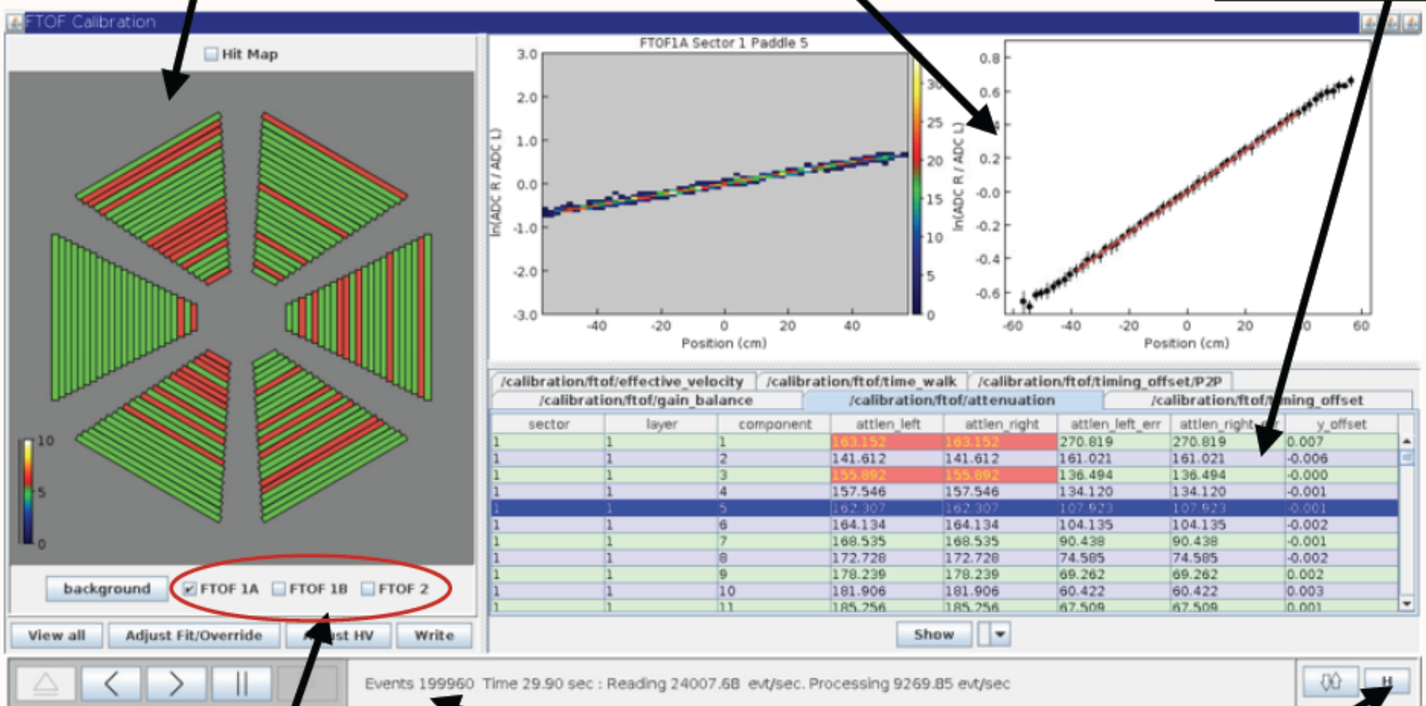
14.7% Forward Tagger
No Electron in ECAL

Calibration suites

Detector Visualization
callback on clicks
coloring by occupancy

Data Canvas
callback for detector
callback for table

Database Table
component callback
constraint coloring
comparison utils



Detector Visualization
Layers

Event Processing Pane
event by event,or whole
opens: File, ET ring, CLOUD

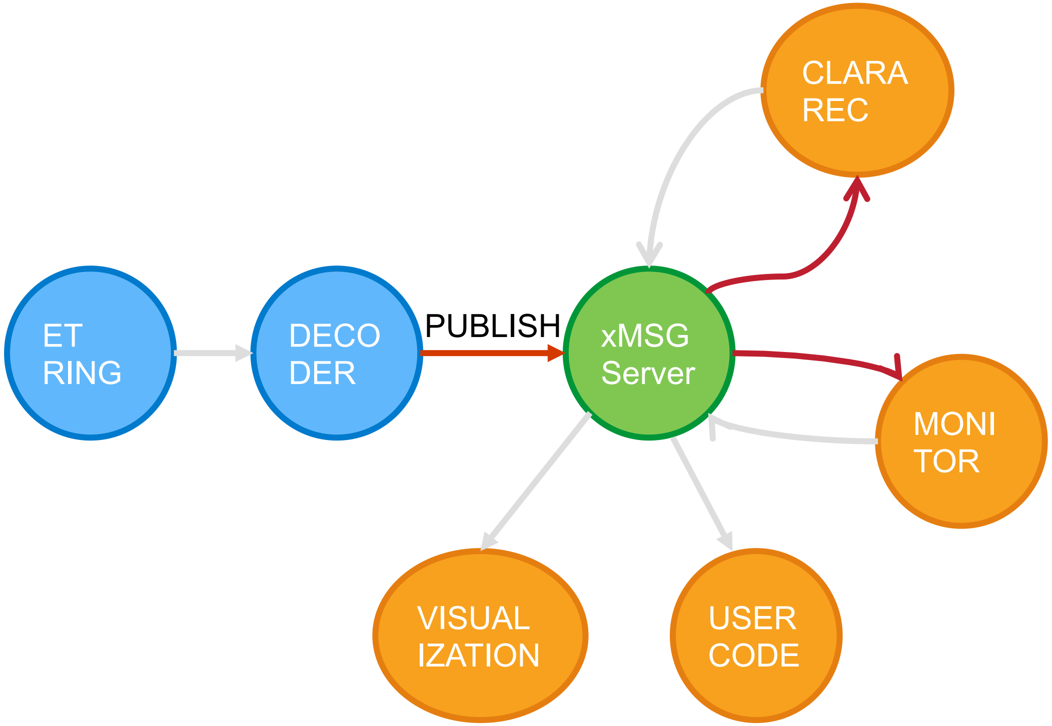
Online reconstruction & monitoring

Online System

- Single connection point to ET-Ring
- Decoding of data only done once and distributed to client applications
- Client application publish analyzed data to be used by monitoring, and calibration applications

Reconstruction

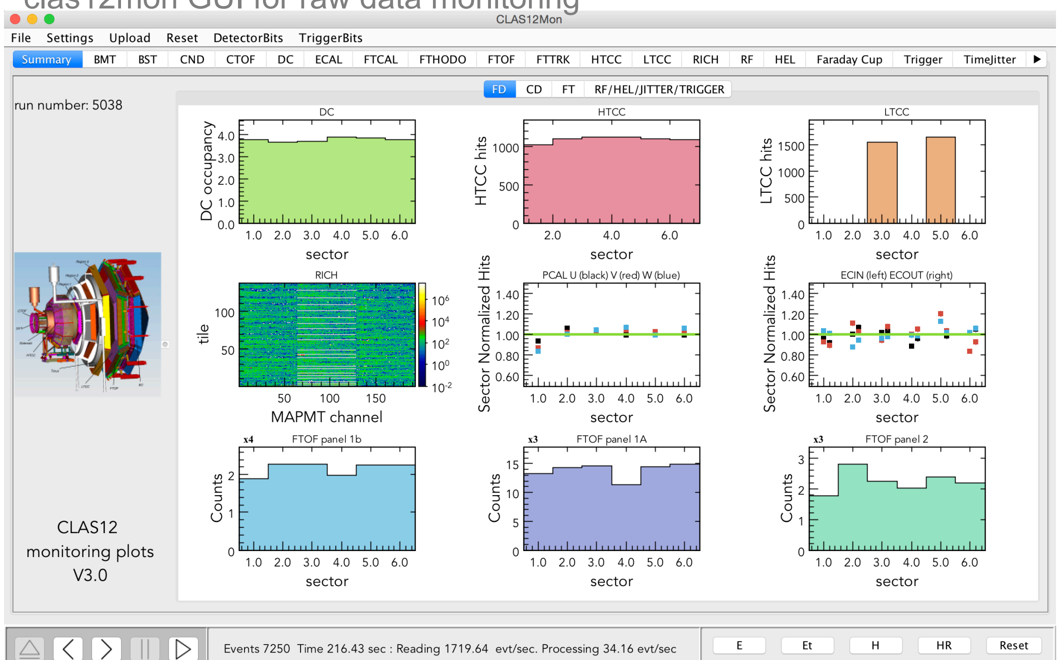
- Reconstruction can run in streaming mode
- Message passing is done using Data Frames (HIPO-4) for higher efficiency
- User code can attach to raw or reconstructed stream depending on application



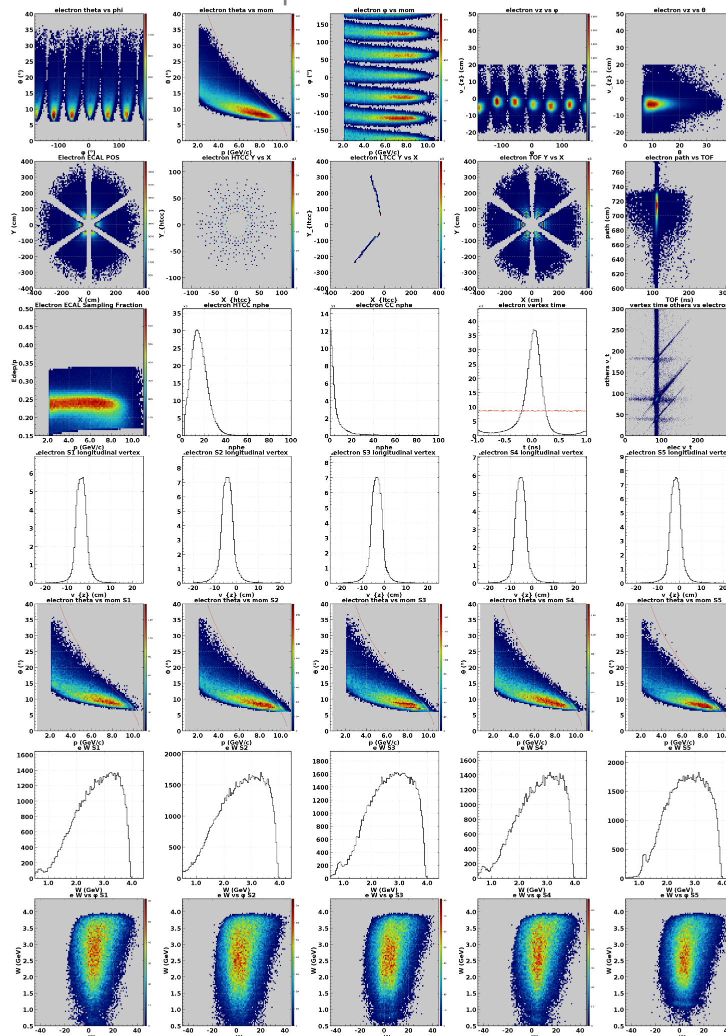
Data quality monitoring

- Monitoring of data quality:
  - Raw data monitor (clas12mon) running online during data taking
  - Offline reconstruction monitoring from routing cooking of few files per run

clas12mon GUI for raw data monitoring

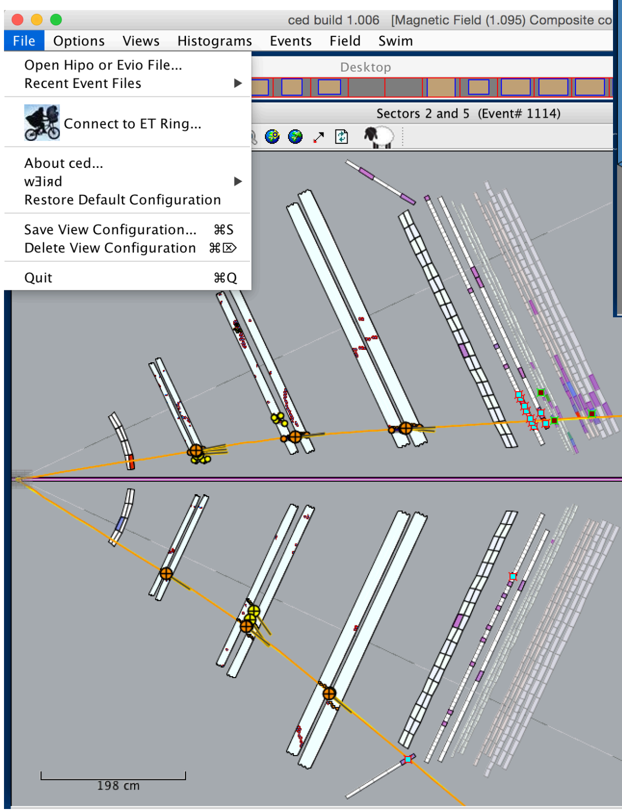


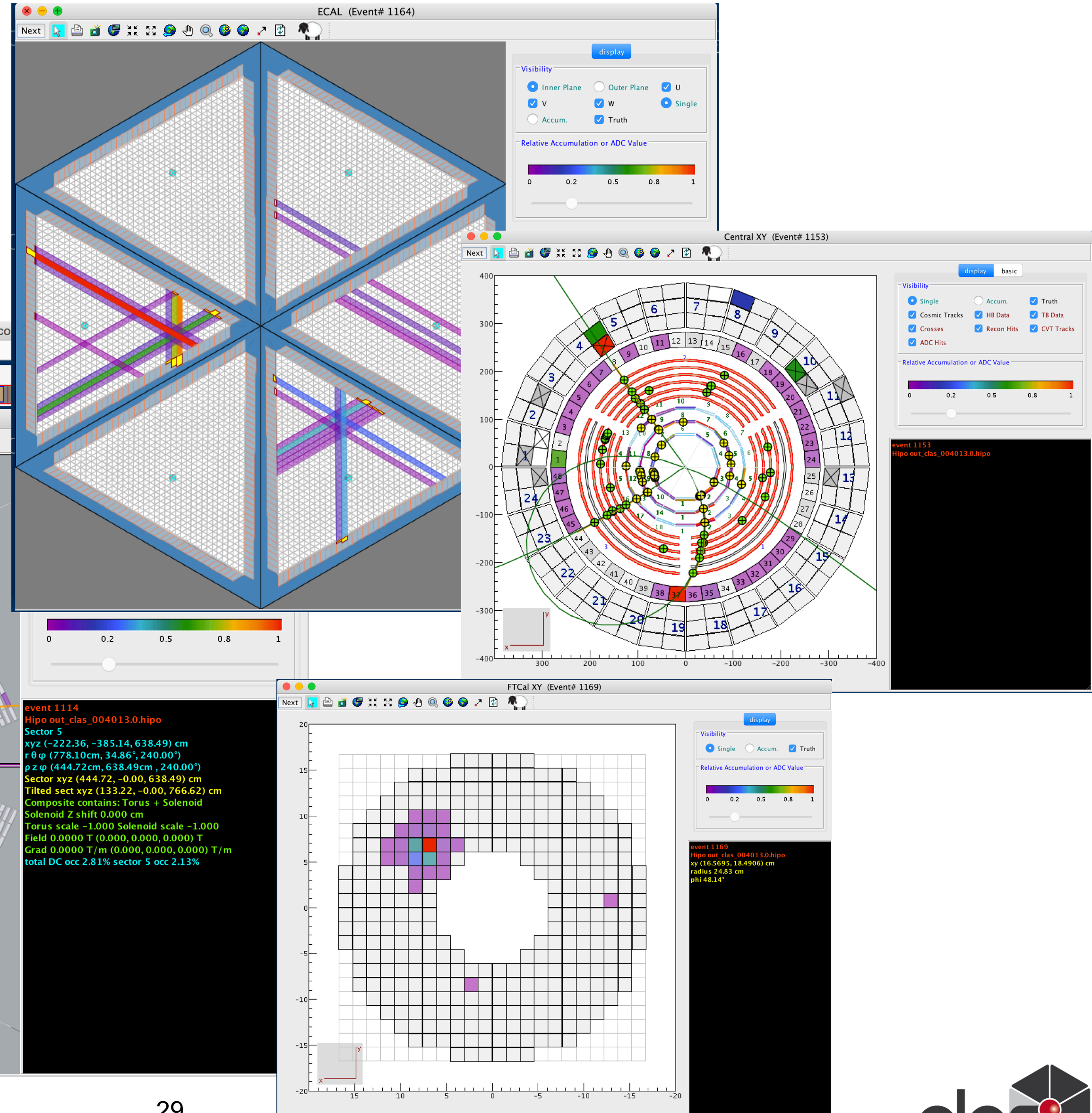
Electron plots from offline reconstruction



CLAS12 Event Display (CED)

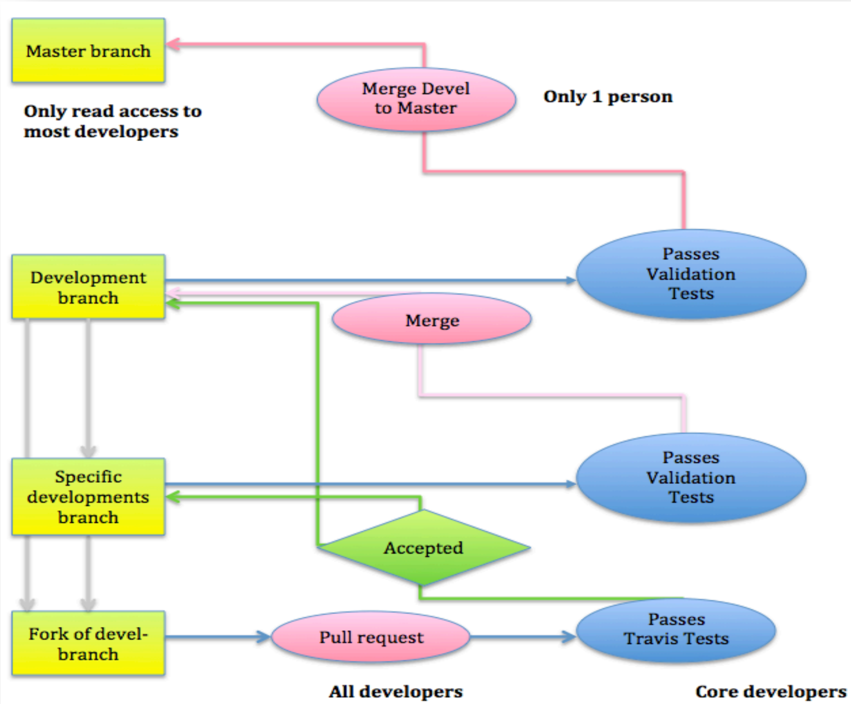
- Java based event display:
  - Based on COATJAVA
  - 2D and 3D views
  - Running on raw EVIO and HIPO banks from ET ring and file





Code management

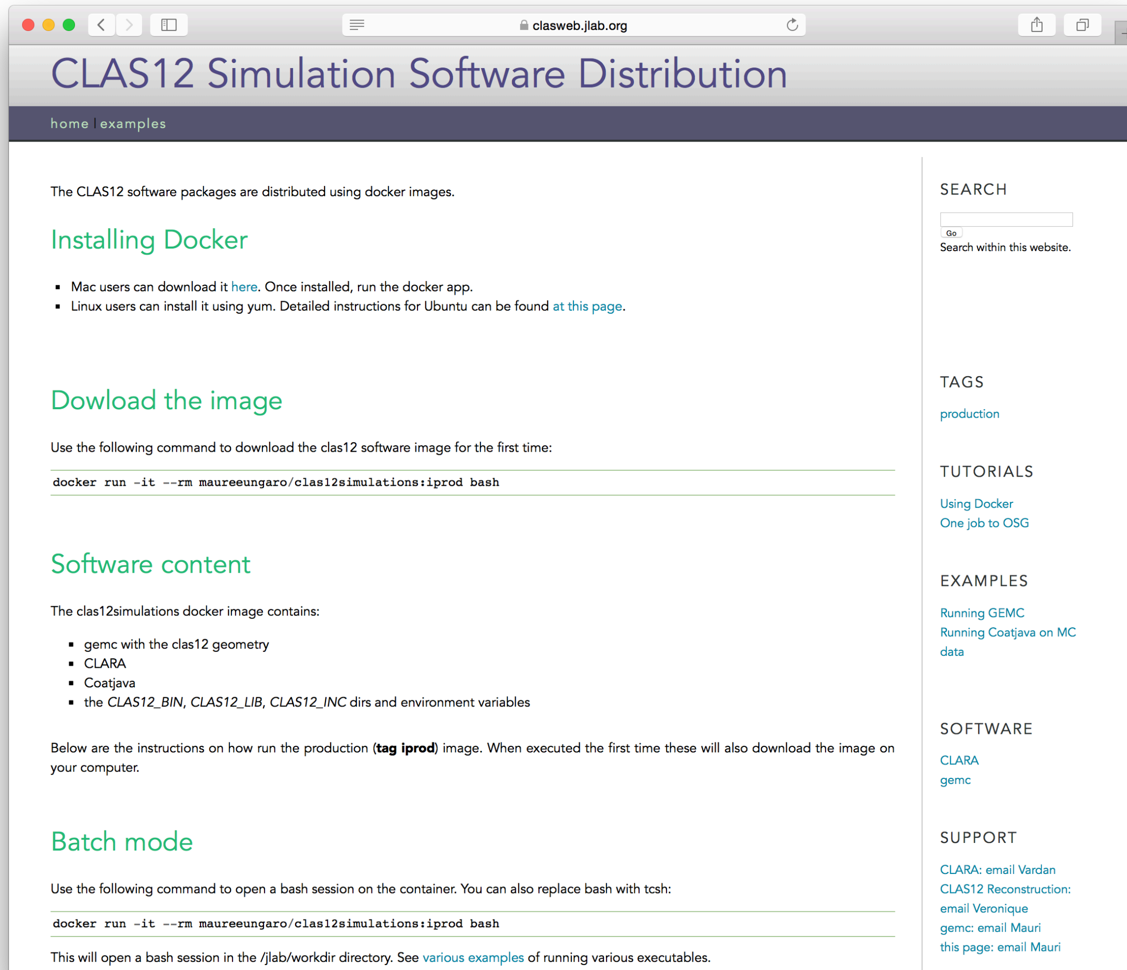
- clas12-offline software kept under github repository
- Maven build system
- Teams responsible for various packages of the Clas12 offline software
- Code validation (validation suites, bug finding tool spotBugs) included in Travis build system
- Code development and release tagging scheme
- Release notes
- Issue reporting



Software workshops

- Workshops organized at Collaboration meeting to help users becoming familiar with the different software tools
- Last event hosted on Nov. 16: tutorial on the use of the new "CLAS12 Simulation software Distribution" based on Docker containers
  - Tutorial on how to launch the simulation and reconstruction chain in Docker
  - Tutorial on how to submit simulation jobs at the OSG

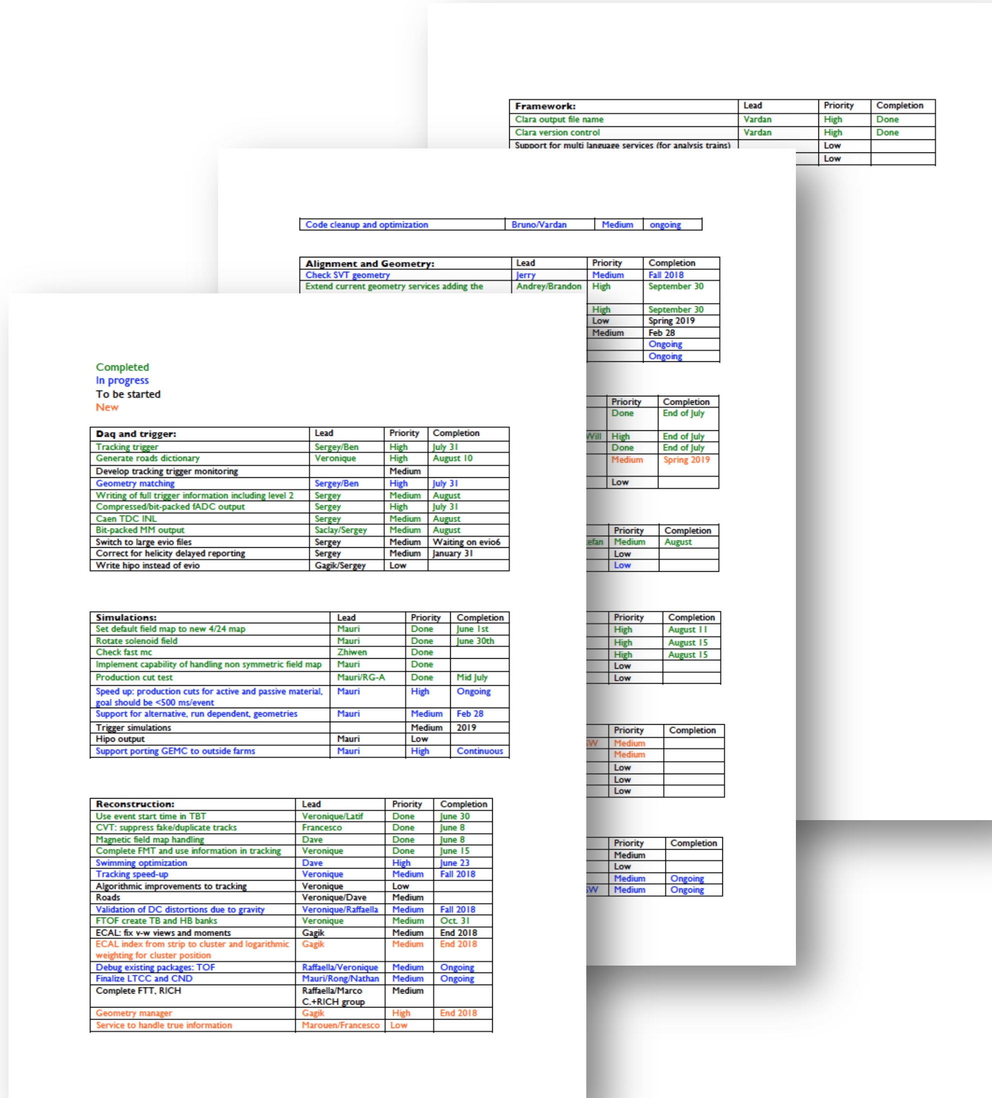
https://clasweb.jlab.org/clas12/clas12SoftwarePage/html/index.html



Software task list

Task prioritization and assignment based on overall experiment priorities and input from Run Groups:

- list of software tasks with details on the work assignments, timeline and level of priority: https://userweb.jlab.org/~devita/software/20181116_tasklist.docx
- periodically updated to:
  - track progress in critical areas,
  - monitor the work load of the key software developers
  - optimize the allocation of resources



Completed
In progress
To be started
New

Daq and trigger:	Lead	Priority	Completion
Tracking trigger	Sergey/Ben	High	July 31
Generate roads dictionary	Veronique	High	August 10
Develop tracking trigger monitoring		Medium	
Geometry matching	Sergey/Ben	High	July 31
Writing of full trigger information including level 2	Sergey	Medium	August
Compressed/bit-packed fADC output	Sergey	High	July 31
Caen TDC INL	Sergey	Medium	August
Bit-packed MM output	Saclay/Sergey	Medium	August
Switch to large evio files	Sergey	Medium	Waiting on evio6
Correct for helicity delayed reporting	Sergey	Medium	January 31
Write hipo instead of evio	Gagik/Sergey	Low	

Simulations:	Lead	Priority	Completion
Set default field map to new 4/24 map	Mauri	Done	June 1st
Rotate solenoid field	Mauri	Done	June 30th
Check fast mc	Zhiwen	Done	
Implement capability of handling non symmetric field map	Mauri	Done	
Production cut test	Mauri/RG-A	Done	Mid July
Speed up: production cuts for active and passive material, goal should be <500 ms/event	Mauri	High	Ongoing
Support for alternative, run dependent, geometries	Mauri	Medium	Feb 28
Trigger simulations		Medium	2019
Hipo output	Mauri	Low	
Support porting GEMC to outside farms	Mauri	High	Continuous

Reconstruction:	Lead	Priority	Completion
Use event start time in TBT	Veronique/Latif	Done	June 30
CVT: suppress fake/duplicate tracks	Francesco	Done	June 8
Magnetic field map handling	Dave	Done	June 8
Complete FMT and use information in tracking	Veronique	Done	June 15
Swimming optimization	Dave	High	June 23
Tracking speed-up	Veronique	Medium	Fall 2018
Algorithmic improvements to tracking	Veronique	Low	
Roads	Veronique/Dave	Medium	
Validation of DC distortions due to gravity	Veronique/Raffaella	Medium	Fall 2018
FTOF create TB and HB banks	Veronique	Medium	Oct. 31
ECAL: fix v-w views and moments	Gagik	Medium	End 2018
ECAL index from strip to cluster and logarithmic weighting for cluster position	Gagik	Medium	End 2018
Debug existing packages: TOF	Raffaella/Veronique	Medium	Ongoing
Finalize LTCC and CND	Mauri/Rong/Nathan	Medium	Ongoing
Complete FTT, RICH	Raffaella/Marco C.+RICH group	Medium	
Geometry manager	Gagik	High	End 2018
Service to handle true information	Marouen/Francesco	Low	