Computing Requirements

N. Baltzell

CLAS Collaboration Meeting

July 10, 2018

Introduction

These slides are largely copied from a talk given at the CLAS12 Computing Review in May 2018. The "as-is" or "Current Resource Requirements" numbers presented here are exactly as they were at the time of the review and presented there.

Meanwhile there are new optimizations, anticipated at the time of that review, but now under active development and materializing, appended in this talk.

Note, the "PAC-Week" convention quoted here was non-standard, as it already accounted for 50% efficiency. This was known and accounted for during the review. So a "PAC-Week" here is really half of one.

Outline

- reconstruction
- calibration
- analysis
- simulation
- resource summary
- improvements

The first charge in the CLAS12 Computing Review in May, 2018:

1. *Evaluate the CLAS12 computing architecture.*
   - *1a. Assess the computing plan, including descriptions of the workflows and processes for calibration, reconstruction, simulation and analysis.*
   - *1b. Are the CLAS12 computing requirements including simulation and the impact of backgrounds well understood and motivated by physics considerations? Are the estimates stable?*

Reconstruction

ClaRA + COATJAVA + HIPO

- **CLA**S12 **R**econstruction and **A**nalysis Framework
  - Glues together isolated, independent services with reactive resource allocation
  - 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
  - https://claraweb.jlab.org/clara/
- **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
- **HIPO** data format
  - 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

Current Resource Requirements, per PAC-week

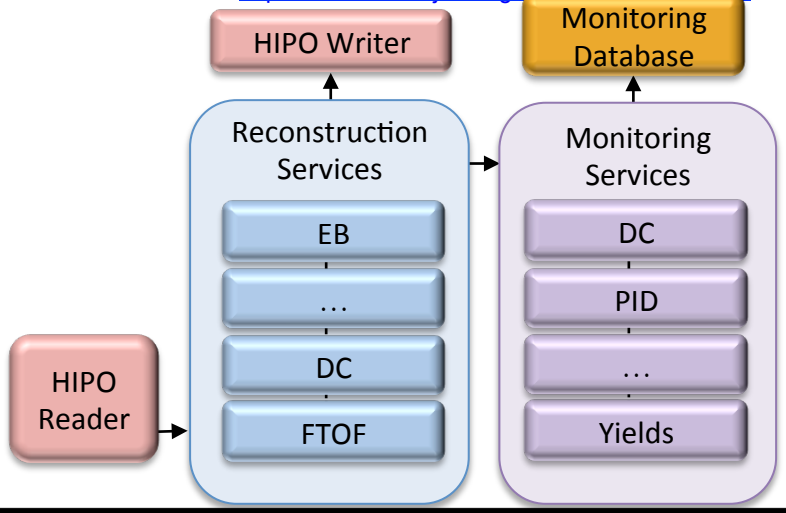
- Assuming one processing of full data set (after calibration)
- cpu: **100K core-days**
- disk: DSTs HIPO: **1.1 TB** (10%, "permanent")
- tape:
  - Raw EVIO: **170 TB** (1.5X compression)
  - Decoded HIPO: **65 TB**
  - DSTs HIPO: **11 TB (**30X reduction from full HIPO)

JLab Batch Farm

- Occupy entire node(s) for most efficient resource usage and leveraging of ClaRA multithreading and scaling
- We've stressed the system and learned well with current data and ClaRA

Databases

- **CCDB**
  - **C**alibration **C**onstants **D**ata**B**ase
  - Hardware translation tables, geometry parameters, calibrations
  - Access optimized, safeguarded for ClaRA services via COATJAVA utilities
  - https://clasweb.jlab.org/cgi-bin/ccdb/objects
- **Monitoring**
  - Populated by monitoring services in ClaRA
  - RESTful API service on Apache webserver with Python FLASK
  - HIGHCHARTS for javascript display in webbrowser
  - These tools are in progress, scheme tested, but full offline monitoring services in development
    - https://clas12mon.jlab.org/mon/dc/trkDoca/
    - https://clas12mon.jlab.org/mon/dc/timeResidual/



Calibration

Common Calibration Framework

- In COATJAVA, used for all CLAS12 detectors
- Provides GUI fitting, plotting, display utilities (via groot)
- For extracting and checking calibration parameters, along each stage of the calibration sequence
- Generates the final calibration tables formatted for to CCDB

Procedures

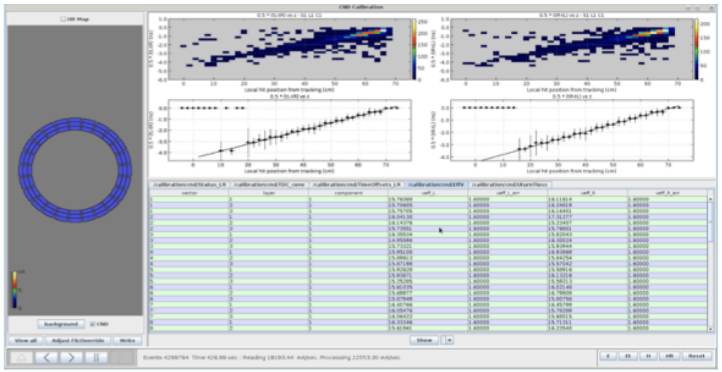
- Workflow and dependencies well understood and tested, first through calibration challenges and now with real data
- Detectors' calibrations are mostly decoupled after rough, initial calibrations

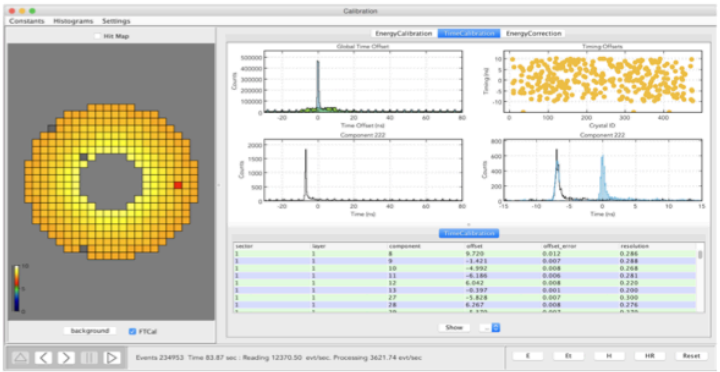
HIPO bank filtering

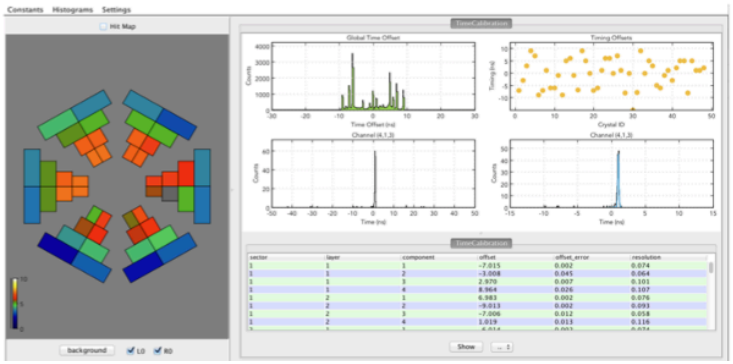
- Easily generate slim skims and bank filters, at least 100X reduction, for each detector's specific calibration needs

ClaRA+COATJAVA Serivce Orientation

- For fast iterations to support individual detector's iterations, e.g. single-service reprocessing







Current Resource Requirements per PAC-week

- Assuming processing of 10% of data sample, total, integrated, including iterations on calibration runs and timelining
- cpu: **11K core-days**
- disk:
  - Full Recon HIPO: **33 TB** (transient)
  - Decoded HIPO: **6.5 TB** (transient)

Analysis

Skim/Analysis Service Trains

- Periodically running with analyzers' services plugged-in, to facilitate "simultaneous" access to data by many physics analyses
  - 11 TB of input DSTs per week, not realistic I/O for all analyzers to be accessing the full data independently
  - new train is run as calibrations/reconstruction changes, or as analyzers' selections change
- ClaRA provides fault isolation between services, i.e. one user's service will not affect the others
- Output HIPO DST skims, or histograms for larger analyses (e.g. inclusive reactions)
- Some administrative control will be necessary
  - e.g. total output not more than ~50% of input
  - e.g. single wagon not delaying the train

Mini Trains

- Run frequently on small data sample kept on disk, e.g. every few days/week

Maxi Trains

- Run less frequently on larger or full data sample, e.g. every week or two or …

Folks Wagons

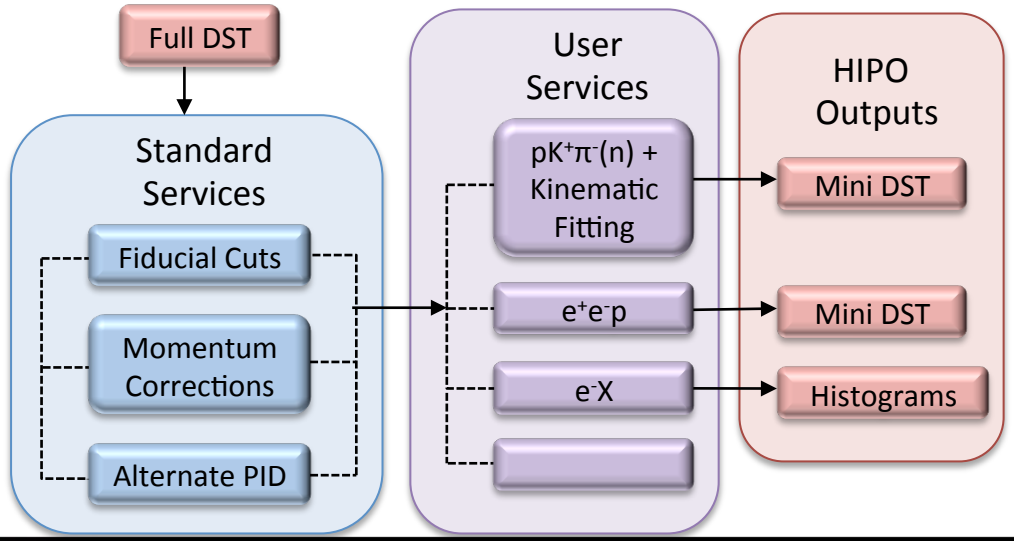
- Very short, just for quick feedback and testing software

Analysis Tools

- HIPO's Java/C++/Fortran libraries exist and in use
- Conversion to ROOT and HBOOK exists and in use
- JAVA Analysis Workstation

Current Resource Requirements, per PAC-week

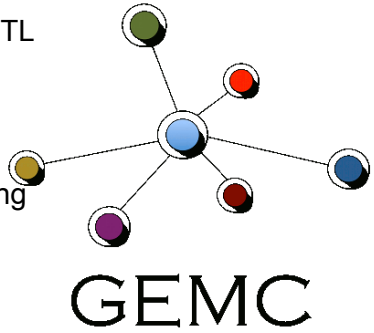
- Assume keeping most recent 2 versions on disk, and that output of each is less than 50% of input
- cpu: **1K core-days** (1% of current recon cpu)
- disk: **< 10 TB** (transient)
- tape: **< 5 TB**



Simulation

GEMC

- Standard GEANT4 geometries stored in sql database, CAD STL geometry also supported, with easy visualization interface
- Incorporates digitization algorithms for all CLAS12 detectors
- Provides multiple levels of accuracy vs speed (e.g. full, or no-secondaries, or acceptance-only)
- Proven to be an accurate representation, based on Engineering Run data, regarding background rates and occupancies, and critical for beamline shielding studies



DC Occupancies @ 10^{35} cm^{-2} s^{-1}		
	ER-A DATA (rescaled)	GEMC
R1	1.7 %	1.12 %
R2	0.3 %	0.44 %
R3	0.9 %	0.73 %

Beam Background

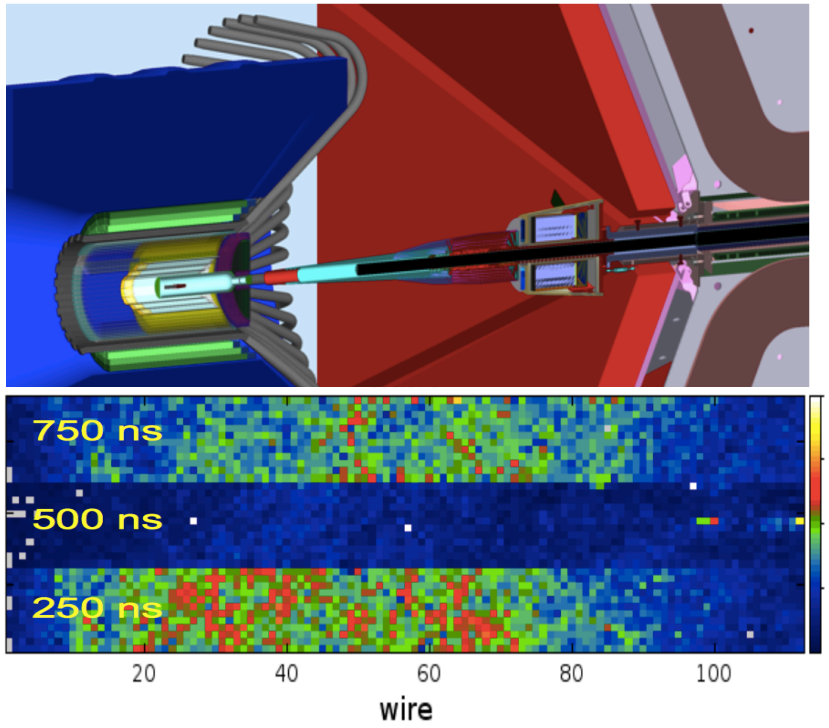
- Merging scheme developed and in use, based on real random triggers for a given luminosity.
- Result will be shared by all analysis groups in terms of efficiency and resolution effects, and its small contribution to resource requirements is neglected.

Physics Reactions

- Individual analysis groups perform their own simulation and reconstruction of physics reactions
- Simulate **1X (full) + 10X (fast)** events of final physics data sample

Exploring external resources

- Open Science Grid and collaborating institutions' clusters
- Singularity+Docker containers for portability (now in gemc 4a.2.4)



Current Resource Requirements per PAC-week

- Assuming final, combined physics samples are 10% of triggers
- cpu:
  - gemc: **450 + 450 core-months**
  - recon: **45 + 450 core-months**
- tape / disk:
  - recon DSTs: **2.2 + 22 TB**
  - evio2hipo: **13 TB** (full only)

Resource Usage Calculations (May 2018)

Base Inputs

- All at nominal luminsity, 75 nA
- Trigger Rate = 18 kHz
- Recon Processing Time per Event = **1.5 s**
- Raw EVIO Event Size = 50 KB
- Full Recon HIPO Event Size = 65 KB
- Full DST HIPO Event Size = 2.2 KB

- *Raw event size and rate are "as is", without expected future improvements, e.g. FADC and MicroMega bit backing (size), trigger roads (rate).*
- *Reconstruction and Simulation time are "as is", with fully validated software, without expected, in development, future improvements.*
- *DST size is "as is", excluding expected improvements, e.g. up to 2X due to current CVT track combinatorics/ghosts.*
- *For simulation, based on previous experience and current data we're assuming only **10%** of data will end up in final samples for physics analyses, and that we'll simulate **1X + 10X** statistics **twice** with **full + no-secondaries** gemc.*
- *"PAC-Week" assumes 7 days on the floor with 50% duty factor. (This is not the usual usage!)*

	Event	Beam-Second	PAC-Week
Raw EVIO	50 KB	850 MB	260 TB
Decoded HIPO	13 KB	210 MB	65 TB
Full Recon HIPO*	65 KB	1.1 GB	330 TB
DST HIPO	2.2 KB	37 MB	11 TB
Recon	1.5 core-seconds	450 core-minutes	3400 core-months
Recon @ 3000 cores		9.0 s	4.5 weeks
gemc	1 core-seconds	60 core-minutes	450 core-months
gemc w/o Secondaries	0.1 core-seconds	60 core-minutes	450 core-months
gemc Recon	0.1 core-seconds	6 core-minutes	45 + 450 core-months
gemc Decoded HIPO			13 TB (full only)
gemc DST HIPO			2.2 + 22 TB

* included for completeness, not expected to be preserved in full

Ongoing Computing Resource Optimizations

Raw Data Size

- 3X reduction of FADC waveforms via bit-packing (30% overall data size decrease)
  - CODA group implementing firmware, to be tested with full DAQ and offline software soon
- micromegas also pursuing bit-packing
- for more efficient networking and tape silo usage

Simulation

- speed
  - 2X for non-fiducial particles by more optimized handling of secondaries *(gemc 4a.2.4)*
  - fast acceptance modes already available in both gemc and coatjava and to be leveraged
- non-JLab resources
  - Open Science Grid (now expected to be available) and CLAS12 institutions' farms
  - facilitated by Docker+Singularity containers *(gemc 4a.2.4)*

Event Rate

- anticipate ~40% reduction via addition of tracking trigger *(partially underway)*

Reconstruction

- speed
  - 2X in magnetic field implementation (*realized)*
  - Up to 10X in swimming algorithms for some cases (*realized)*
  - future optimization of reconstruction services, e.g. tracking *(not underway yet)*
- data size
  - support for EVIO file sizes above 2 GB
    - more efficient tape silo usage
  - DST size reducing due to improved reconstruction algorithms
    - (*e.g. CVT track seeding realized*)

Resource Usage Calculations (July 2018)

Base Inputs

- All at nominal luminsity, 75 nA
- Trigger Rate = 18 kHz
- Recon Processing Time per Event = **1.5 s**
- Raw EVIO Event Size = 50 KB
- Full Recon HIPO Event Size = 65 KB
- Full DST HIPO Event Size = 2.2 KB

- *Raw event size and rate are "as is", without expected future improvements, e.g. FADC and MicroMega bit backing (size), trigger roads (rate).*
- *Reconstruction and Simulation time are "as is", with fully validated software, without expected, in development, future improvements.*
- *DST size is "as is", excluding expected improvements, e.g. up to 2X due to current CVT track combinatorics/ghosts.*
- *For simulation, based on previous experience and current data we're assuming only **10%** of data will end up in final samples for physics analyses, and that we'll simulate **1X + 10X** statistics **twice** with **full + no-secondaries** gemc.*
- *"PAC-Week" assumes 7 days on the floor with 50% duty factor. (This is not the usual usage!)*

	Event	Beam-Second	PAC-Week
Raw EVIO	~~50 KB~~	850 MB	260 TB
Decoded HIPO	13 KB	210 MB	65 TB
Full Recon HIPO*	~~65 KB~~	1.1 GB	330 TB
DST HIPO	~~2.2 KB~~ ~50%	37 MB	11 TB
Recon	<50% ~~1.5 core-seconds~~	450 core-minutes	3400 core-months
Recon @ 3000 cores		9.0 s	4.5 weeks
gemc	~~1 core-seconds~~	60 core-minutes	450 core-months
gemc w/o Secondaries	0.1 core-seconds	60 core-minutes	450 core-months
gemc Recon	0.1 core-seconds	6 core-minutes	45 + 450 core-months
gemc Decoded HIPO			13 TB (full only)
gemc DST HIPO			2.2 + 22 TB

* included for completeness, not expected to be preserved in full

Summary

- During the software review in May, we said:
  - *Current resource requirements are understood and well motivated, although software improvements and optimizations are in progress and anticipated to significantly increase performance.*

- But resource requirements were (too) large for JLab's systems, in some cases much larger than anticipated and/or quoted in previous years, and critical, targeted work is needed and planned to address that.

- Now those improvements are really materializing and will enable us to fit into JLab's computing resources for the upcoming runs/years.

- Meanwhile, much optimization work remains, and wise use of resources will always be necessary,
  - e.g. not reprocessing large data sets excessively, should always aim for single-pass of 100% reconstruction
  - e.g. not loading JLab systems with large simulation cycles