# DAQ & Software Review:
# Calibrations, alignment, online data monitoring

Jlab

*June 18, 2014.*

# Software Organization

| | Software | Analysis |
|---|---|---|
| Chairs: | Maurik Holtrop | Mathew Graham |
| Subsystem Leads: | ECAL: Stuart Fegan | SVT: Omar Moreno |

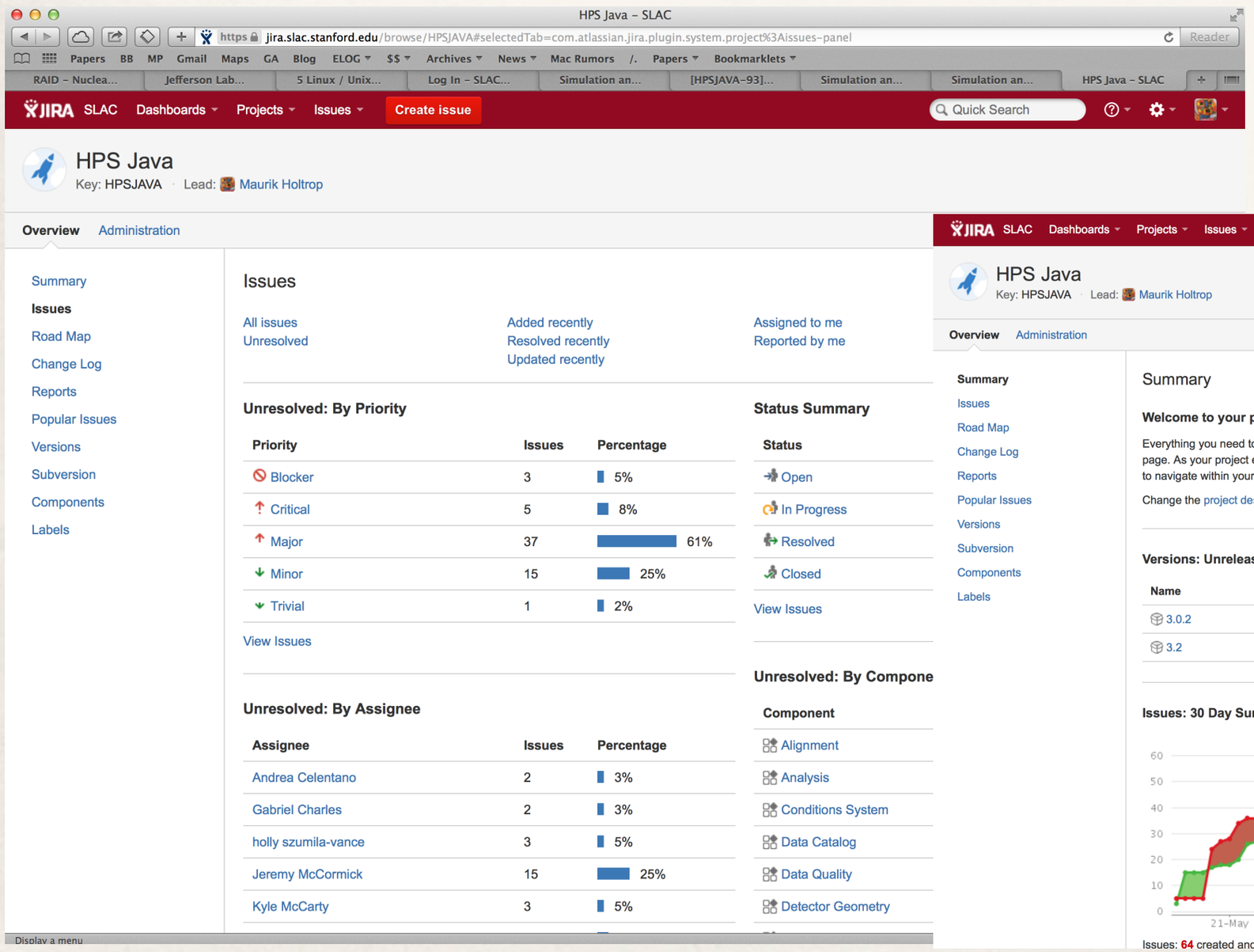
Monitor App, Conditions System, Data Catalog, Framework, … Jeremy McCormick

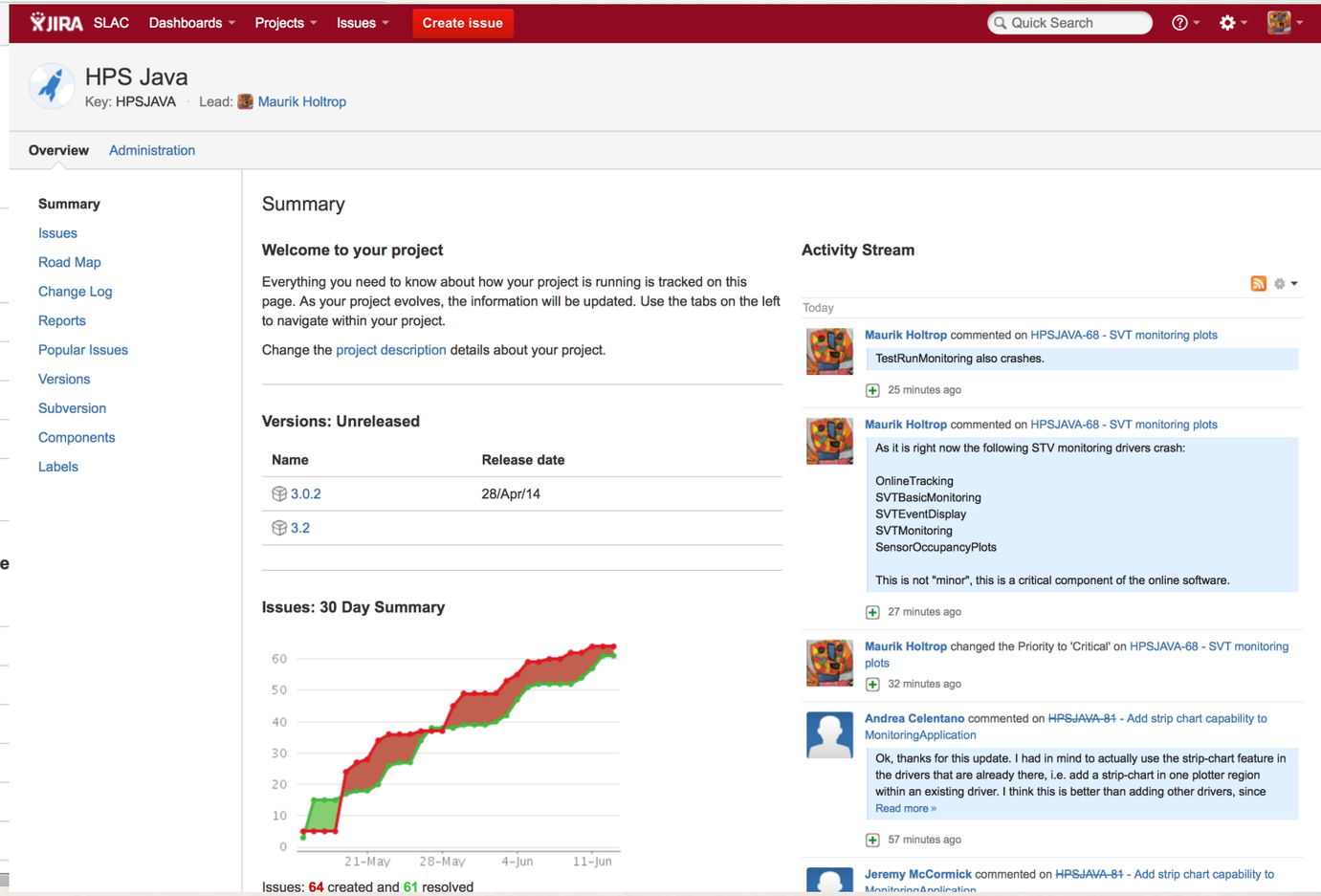
Developers: Sho Uemura, Gabriel Charles, Per Hansson Adrian, Kyle McCarty, Holly Vance, Norman Graf, Andrea Celentano, Luca Colaneri, Takashi Maruyama, Tim Nelson, Allesandra Fillipe, Gabriele Simi,…

Software Wiki Pages: https://confluence.slac.stanford.edu/display/hpsg/Computing+and+Software

# Task Tracking - JIRA

Online system to keep track of tasks/issues/bugs.
Documents progress on tasks, allows to have archived conversations.
Tied to the software release cycle.
Developers are starting to make better use if this tool…




https://jira.slac.stanford.edu/browse/HPSJAVA

# Task Tracking - Schedule

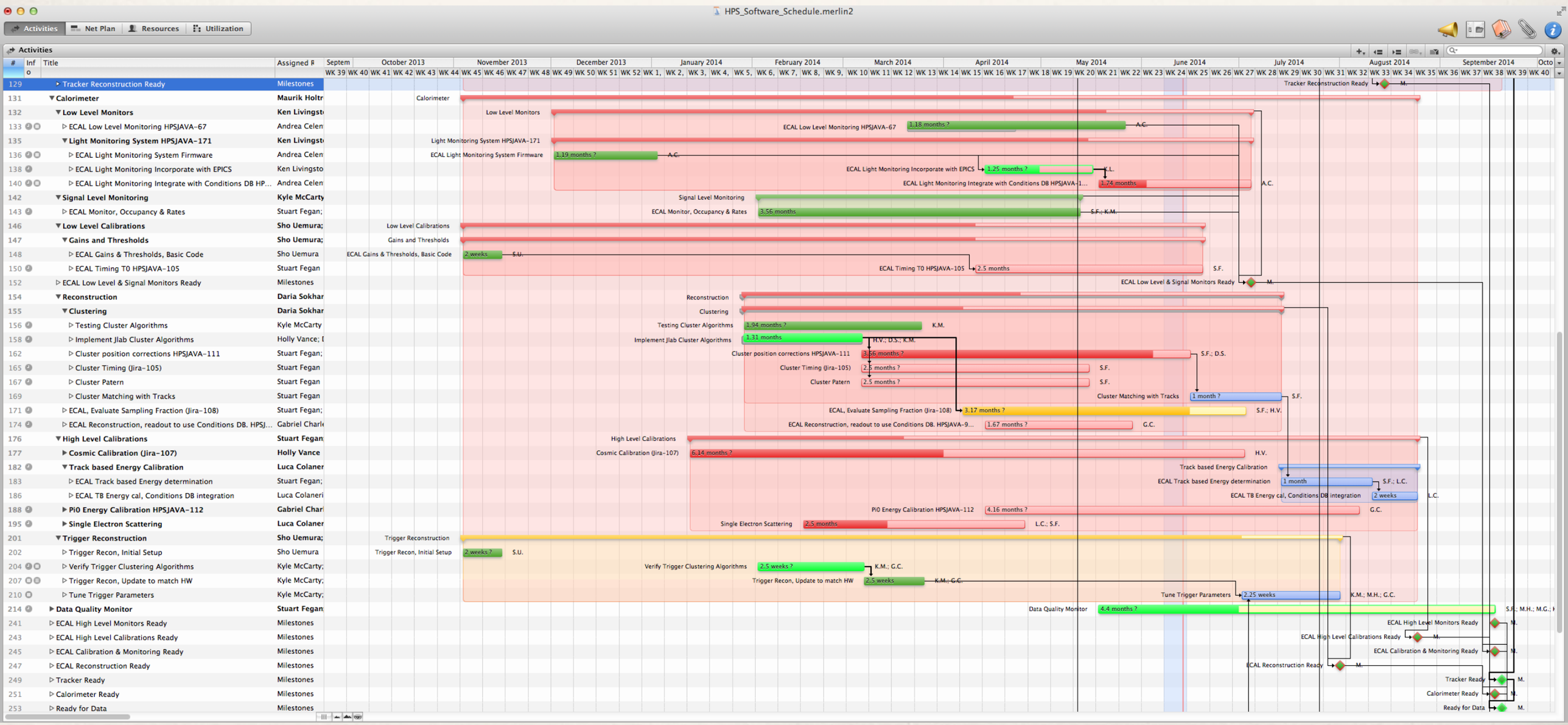
Keeps track of timeline of development.

Exported to web pages

http://nuclear.unh.edu/HPS/HPS_Software_Schedule

Updated by hand, started to synchronize with JIRA tasks.

Does not indicate wether a task is critical or not.



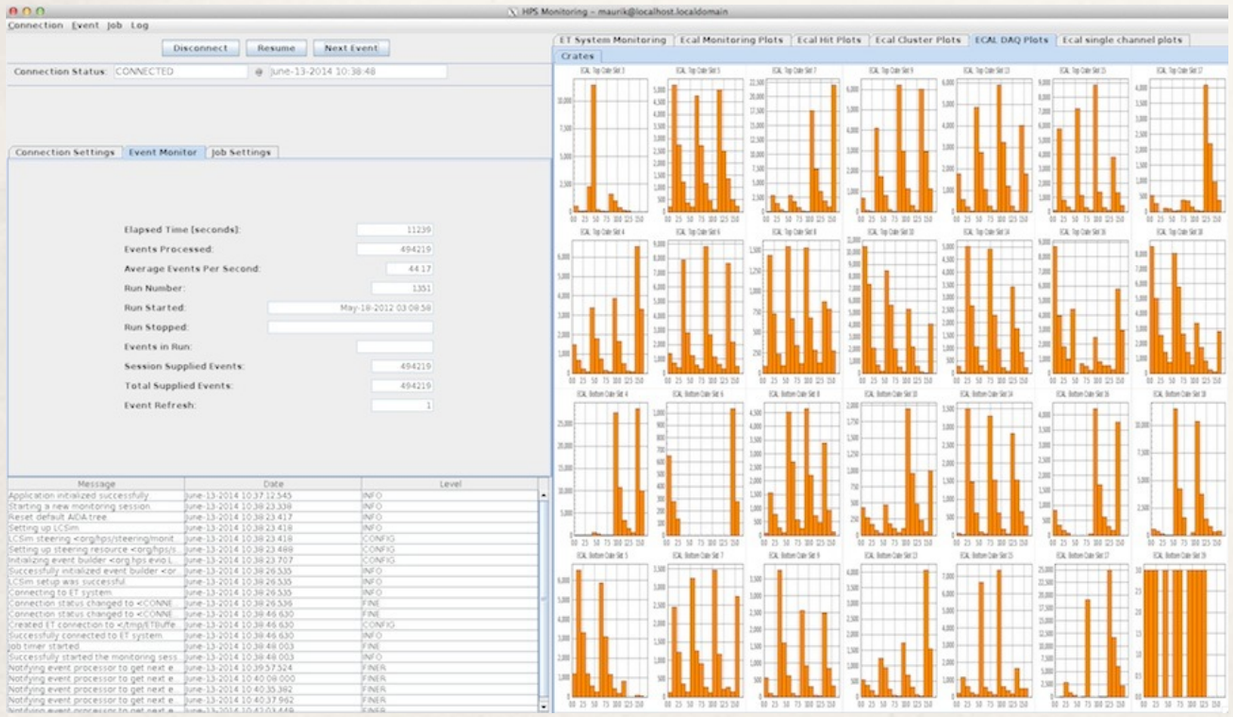
# Software Components for HPS

- Online Monitoring
- Reconstruction software components for ECAL & SVT
- Calibrations
  - ECAL and SVT calibrations and alignment.
- Conditions System (Database)
- Data Catalog
- Experiment Monte Carlo Simulation
- Offline reconstruction, Data Analysis, DST, data throughput, farm resource requirements …

— Matt Graham will present after this talk.

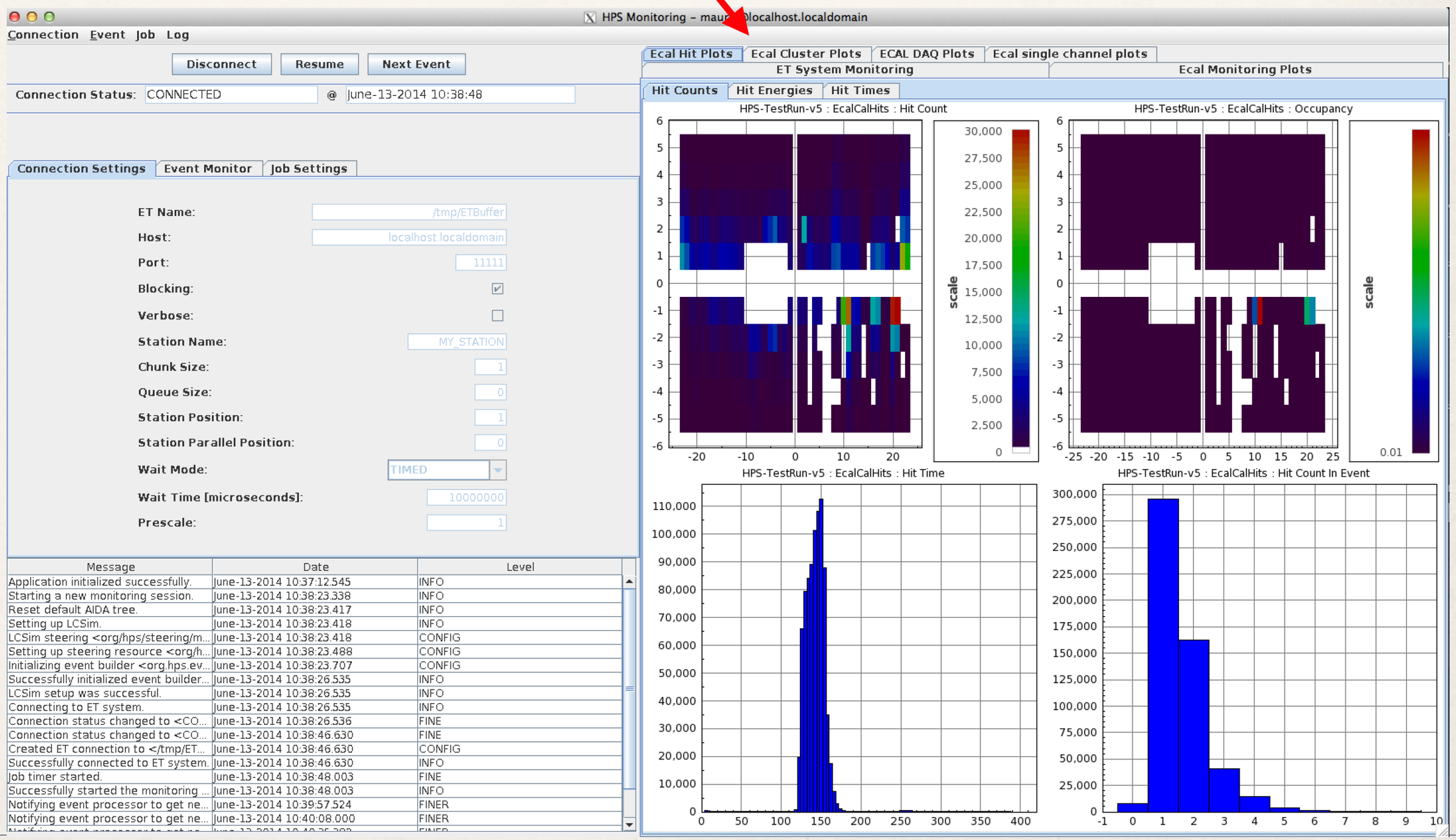
# Monitoring Application

- Monitoring application written in JAVA
- Provides flexible platform to display live updating histograms
- Reads data from ET in EVIO format, converts internally to LCIO (data analysis framework file format)
- Can run multiple copies, i.e. separate ones for different sub-systems.
- Histograms can be saved at end of run.
- Reasonable data rate: ~ 100 Hz (for complicated ECAL monitor)
- Full reconstruction framework available to app to make high level plots.



# Monitoring Application

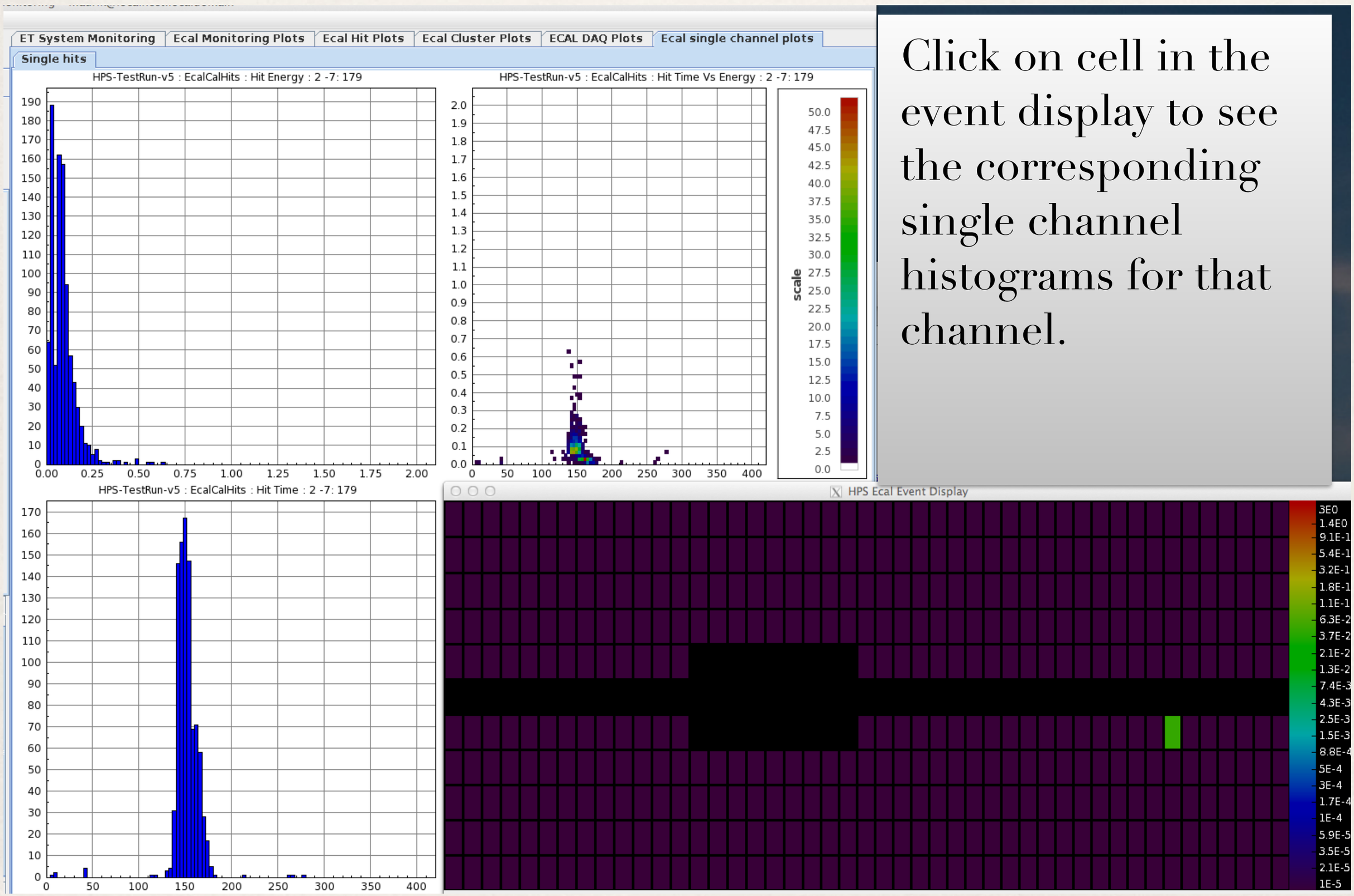
Tabs allow shift taker to select various histogram panels.



# ECAL Event Display

Shows the hits in the ECAL, color coding indicates energy of hit.
Red box (not shown) indicates a found cluster.
Text box below shows values of cell that was hovered over.

# Single channel histograms.




Click on cell in the event display to see the corresponding single channel histograms for that channel.

# Monitoring Application

- Monitor app is fully functional and ready for run as is.
- Further Development for monitor app:
  - Add strip-chart capability (nearly done)
  - Update the SVT histograms
  - Add SVT event display
  - Clean up interface panel and move to pop-up
  - Add systems status page (Information on the system other than histograms)
  - Tweaks: Further refine the histograms displayed

# Reconstruction Software Component: ECAL

- Readout and raw data conversion working
- 2 Clustering algorithms working
  - Sampling fraction study nearly completed, +1 week?
  - Cluster position correction nearly completed, +1 week?
- Trigger simulation updated to reflect FPGA algorithms
  - Refinement of trigger parameters in progress
- ECAL: Monitoring histograms defined and implemented
- Integration with Conditions System in progress (+1 month?)

# Reconstruction Software Component: SVT

- SVT Readout and cluster finding fully working.
- Track finding - fully working.
- Tracker Generalized Broken Line algorithm:
  - Fully working in C++
  - Porting to JAVA in progress (not critical).
- Monitoring histograms since test run - need updating (+2 weeks?)
- Track Based Alignment is working,
  - Millipede II, Implementation improvement desirable.
- Integration with conditions system needed (+1 month?)

# Calibrations: ECAL

- ECAL needs calibrations for Gains, Thresholds, Timing.
  - Test run analysis shows all the basics are in place, but we can do (a lot) better.
- We will be using a multi-pronged approach:
  - Cosmic Calibration
    - Using cosmic rays and two scintillator paddles for trigger to get initial gain/threshold
    - Simulation in place, shows rates are reasonable.
    - Fitting and finalizing conditions code in progress ( +2 weeks?)
  - Single Electron Calibration
    - Basic principles discussed, studied and written up.
    - Code needs completion and Conditions database integration (+3-4 weeks?)
  - Pi0 Calibration / Calibration checking.
    - Initial MC studies performed
    - Code development not yet started. (+2 months?)
  - Track Based Calibration / Calibration checking.
    - Not started. Code has a lot of overlap with e- calibration. (+1 month)
- ECAL Timing
    - Work started, 20% done. (+3 weeks?)

# Calibrations: Tracker

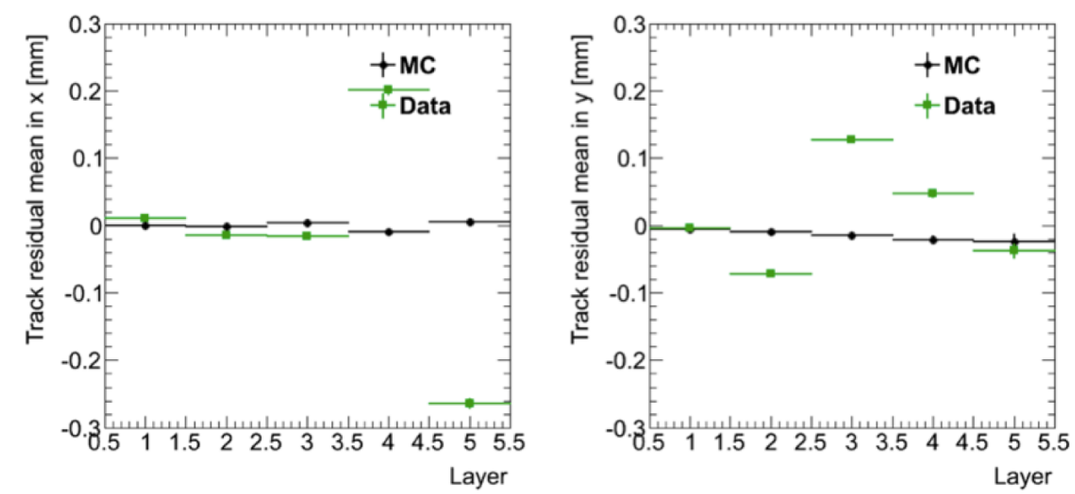
- Tracker needs Pedestals, Noise, Gains, Offsets + Timing in the SVT latency + Finding Alignment Constants.
  - Test run analysis shows all the basics are in place, but we can do (a lot) better.
- Pedestals & Noise - calibrated in one step using existing hardware/DAQ based procedure. Work well during test run.
- Gains & offset (charge scale) - determined once at SLAC with source.
- Timing in the SVT latency - The latency needs to be set so a trigger takes the correct SVT samples. During the test run this was done "by hand". An improved procedure is very desirable. Task is started and is more a procedure than software development. (+2 weeks?)
- Alignment - Tracker needs to be internally aligned to very high precision.
  - Test run shows it can be done, but procedure is cumbersome.
  - Active progress on improving the Millipede II integration with our software.
  - Much progress here, but still a lot to do. (+1.5 months?)
    - Geometry model needs updating.
    - Integration with Conditions System needed.

# Track-based Alignment

V. Blobel, C. Kleinwort, F. Meier, Computer Phys. Communications (2011)
Kleinwort, NIM A, 673 (2012), 107-110

Survey is starting point for alignment

- Test run reached ~100um residuals w/ some manual corrections
- Expect better for 2014 (see Shawn's, Takashi and Tim's talk)
- Roughly, ~10um is sufficient for HPS




*Mean of biased track residuals vs tracker layer*

Multiple ways to achieve similar performance

Our approach:

- Do a least square fit of local (track) and global (alignment) parameters
- Millepede-II can do this for us and is "supported" by GBL
- Great support from C. Kleinwort (GBL/ Millepede developer)

Residual for measurement *i*:

5 track parameters

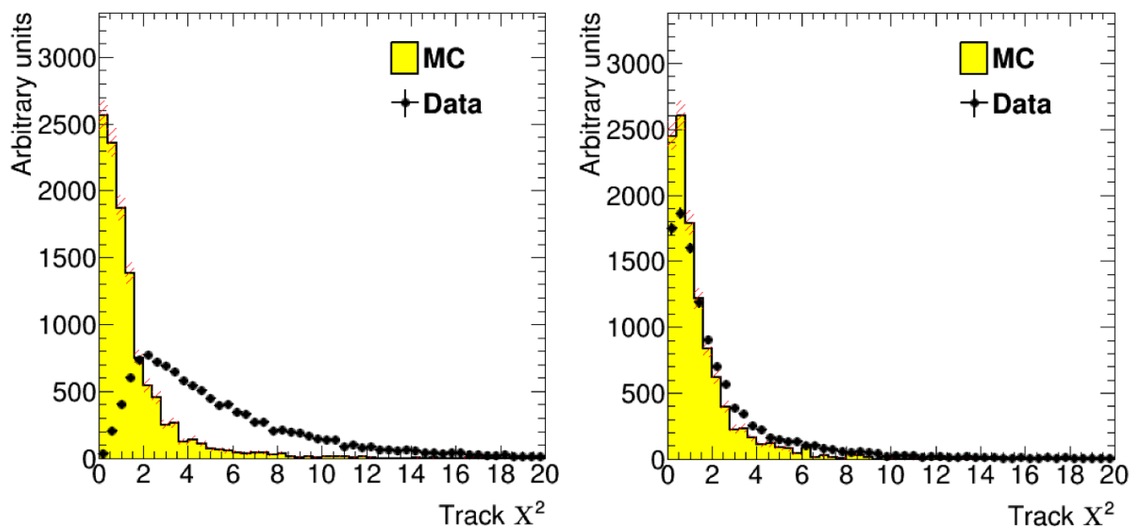
Subset Ω of *n* alignment constants

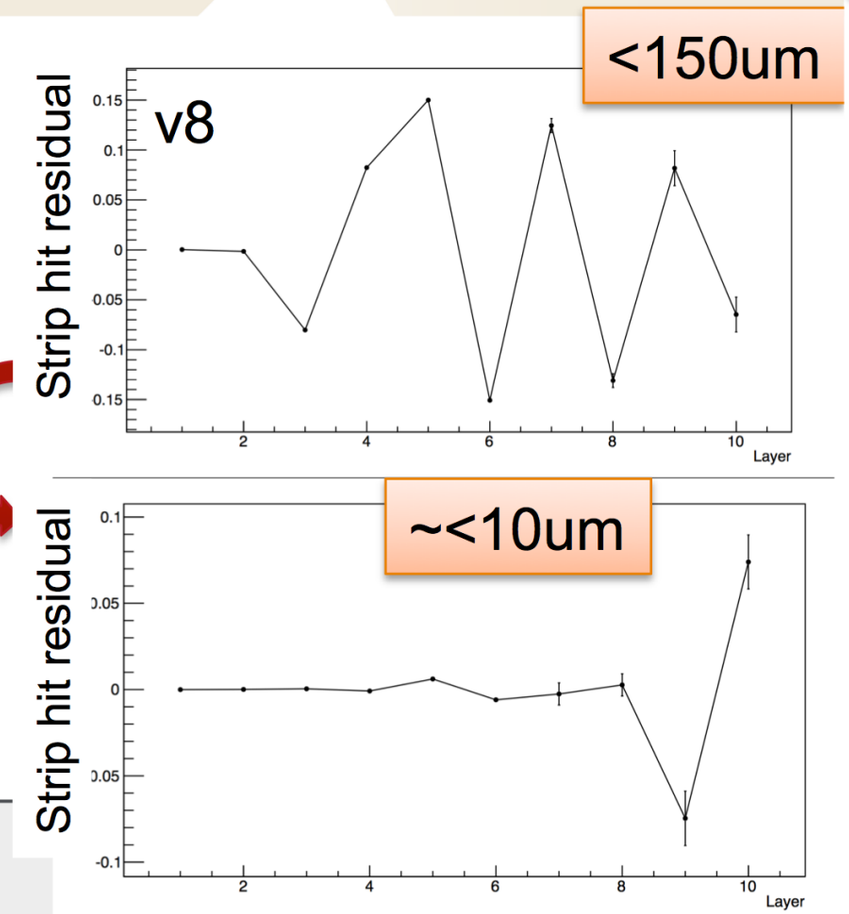
$$y_i - f(x_i, \mathbf{q}, \mathbf{p}) = \sum_{j=1}^{5} \left( \frac{\partial f}{\partial q_j} \right) \Delta q_j + \sum_{l=1}^{\Omega} \left( \frac{\partial f}{\partial p_l} \right) \Delta p_l$$

# Track-based Alignment

Use Millepede-II to align the Test run detector

- It works! But translations only here; rotation corrections look ok but need updated geometry description

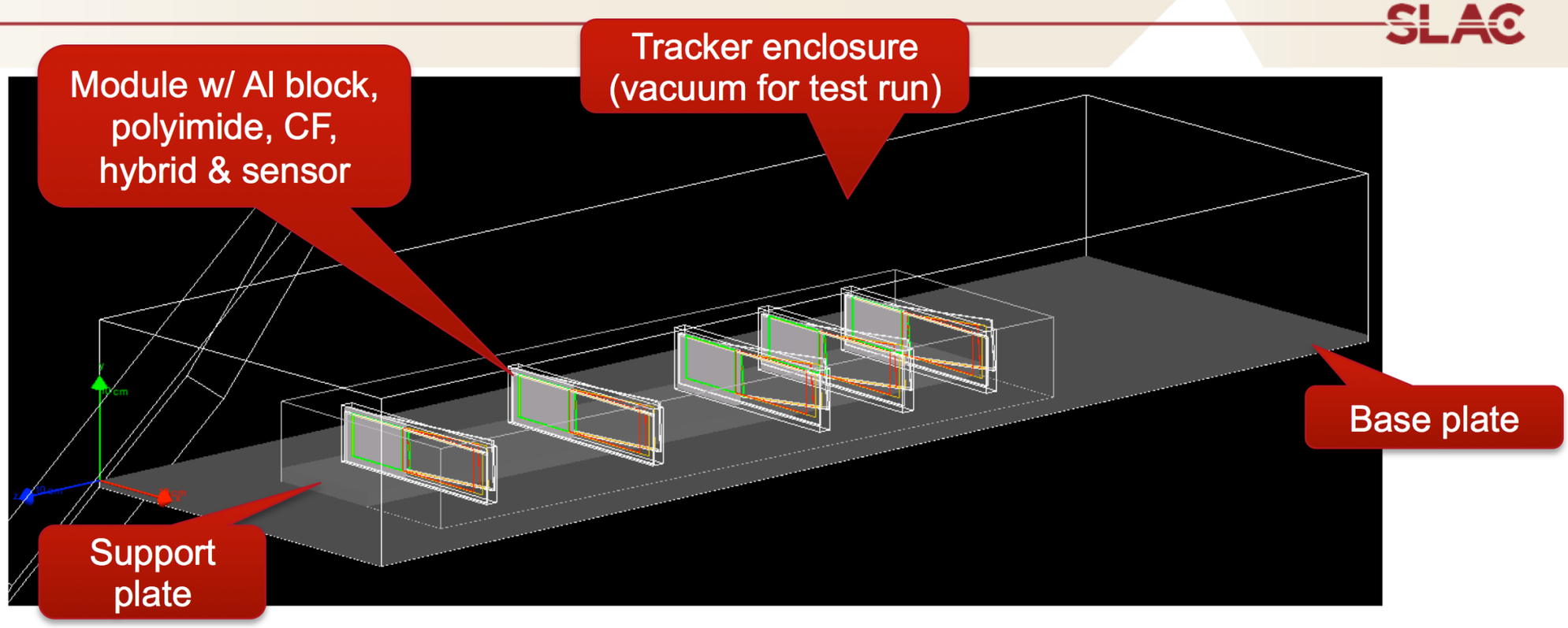



| | | |
|---|---|---|
| Millepede-II | • | L1-3 alignment global constants |
| | • | Include vertex in minimization |
| Geometry tools | • | Geometry implementation based on detector survey |
| Special run | • | Include straight line track sample and check improvement |
| | • | Determine trigger and sample size needed |
| Operational procedures | • | Streamline software |
| | • | Monitoring – rapid feedback during run (beam spot, chi2, track residuals) |
| | • | Offline and online shifter responsibilities |

12

# Track-based Alignment




New geometry based on production drawings

- Simplify bootstrapping from survey
  - Built from surveyed positions on mech. drawings
- Accommodate alignment constants better
  - Global and local translations/rotations automatically ok
- More complete dead material
- Need a couple of more weeks until finished

Test run used as test bed, “simple” to go to new SVT

# Conditions System

- This system keeps track of all the calibration constants and all quantities that can change with time
- Original system was simple and file based, it was recently replaced with a MySQL database based system
- The framework is completed and fully functional
- Main tasks remaining are integration with all the other software parts
- Same system will also be used to keep track of Data Quality Monitoring information.

# Data Catalog

- The Data Catalog will keep track of where the data resides and what status of processing it is in.
- Data will go through stages and we need to keep track of each step:
  - Raw data - # events, location of files, basic run conditions.
  - Reconstructed data - … + versions of software, data quality
  - DST - … + filters applied if any
  - + Data Quality Monitoring plots and information.
- Choice was made to use the SLAC based SRS Data Catalog
  - Decision made after surveying JLab tools available.
  - Uses Oracle Database, Java middle layer and Web interface and scripts for front end.
  - This is a mature, professionally managed system used by several experiments: Fermi, EXO and LSST.
  - Runs at SLAC: information needs to be pushed/pulled over the net using ssh.
  - Task started to write the integration scripts and data crawlers.

http://srs.slac.stanford.edu/DataCatalog/

# Data Catalog Details

A thorough survey was taken of JLAB tools, and nothing suitable was found that could be used without major alterations and additions for our needs. So we are using the SRS Data Catalog application recommend by SCS at SLAC. It has a Java middle layer, Oracle backend database, and a front-end script for executing commands.

In order to execute data catalog commands from JLAB, we will be using an SSH tunnel to SLAC, as the Oracle database is only accessible behind the SLAC firewall. The SLAC account 'hpscat' has been setup as a connection point from 'clashps' at JLAB, in order that password-less login with shared keys can be used to connect. (Users with SLAC accounts may also use those credentials.) The plan is to catalog all our data from the updating run, including raw data, LCIO recon files, and DSTs. Both the primary copies at JLAB and the replicas at SLAC will have entries in the catalog. A data crawler script will be automatically activated via a cron job in order to find and add new records to the database. Time permitting, I am also planning to catalog (some) existing data from the 2012 Test Run and current Monte Carlo samples, as a test of the various scripts and functionalities.

Support is provided through SCS at SLAC. Specifically, Tony Johnson has already given information and implemented some small changes that were required for us. The data catalog is already a complete application that is used for Fermi, EXO and LSST. In terms of work specific for HPS, I have written a series of more "user friendly" scripts for updating, querying, deleting, etc. which wrap the standard front-end script.

Links...

This system has a web front-end which is accessible here:
http://srs.slac.stanford.edu/DataCatalog/

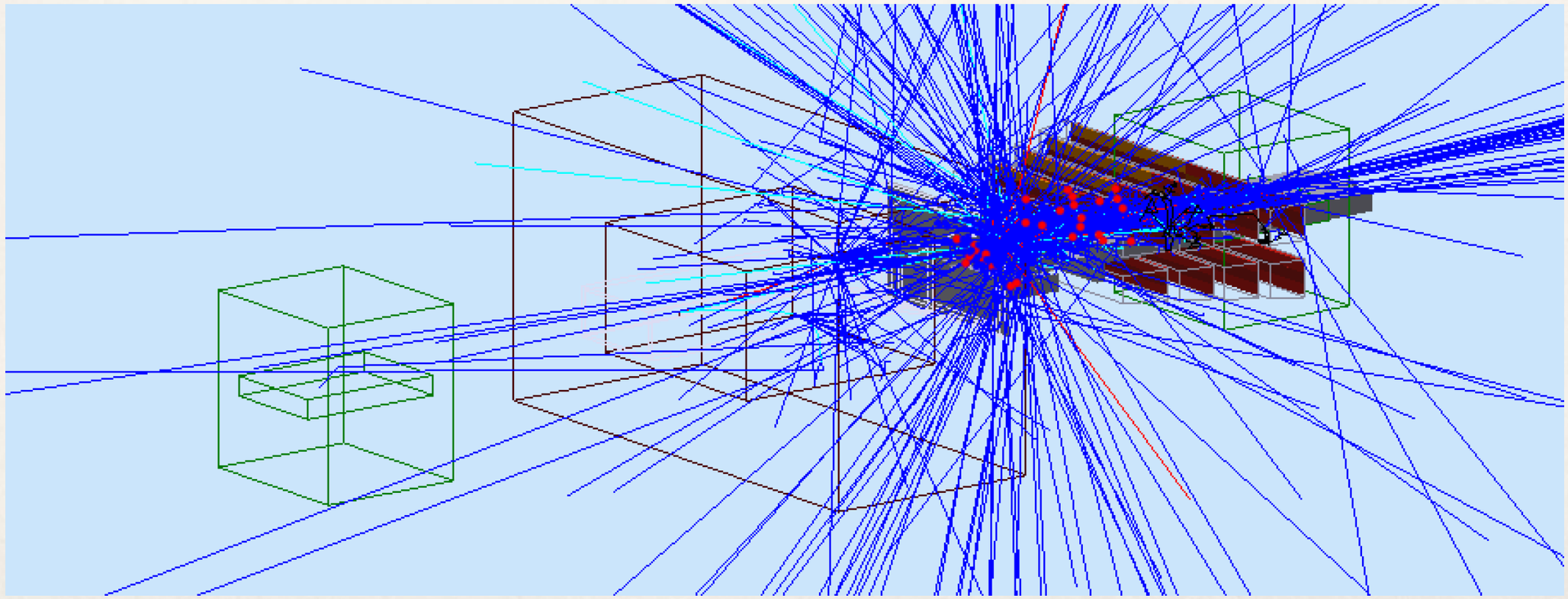
List of Jira items related to the data catalog deployment:
https://jira.slac.stanford.edu/browse/HPSJAVA-148

Proposal for record meta data:
https://confluence.slac.stanford.edu/display/~jeremym/HPS+Data+Catalog+Meta+Data

Technical notes about using the scripts:
https://confluence.slac.stanford.edu/display/~jeremym/HPS+Data+Catalog+Notes

# Experiment Monte Carlo Simulation

- Experiment Monte Carlo Simulation based on GEANT4.
- Multiple physics generators: beam, trident background, signal.
- Two separate GEANT4 based packages: SLIC & GEMC.
- All of the geometry and readout details included in SLIC.
- Production running for 2.2 GeV Mock Data Challenge in progress.

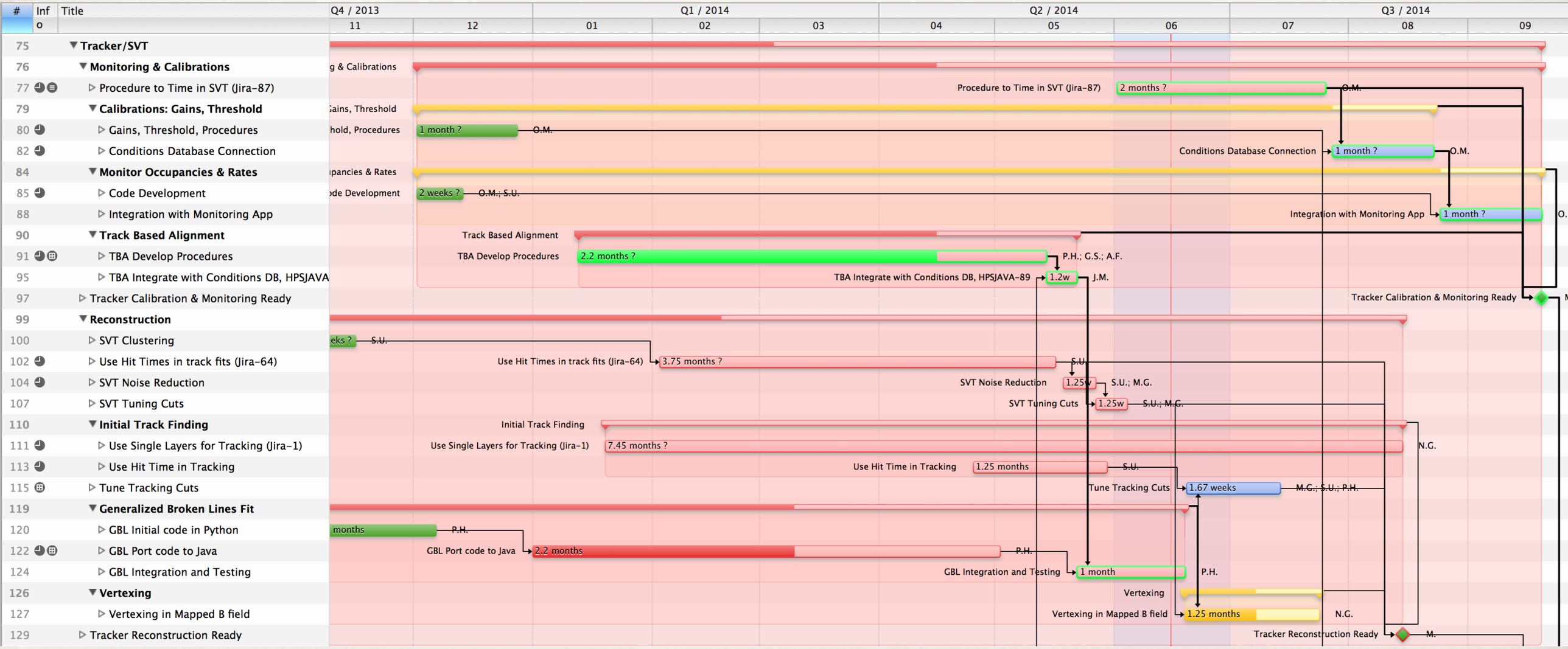


# Current Status of Software

- The basics of a working system are in place.
  - Test run data paper submitted to NIM.
- Improvements since test run:
  - Better/faster monitoring framework + conditions framework.
  - Full B-field for simulation and tracking.
  - ECAL event display and monitoring implemented.
  - ECAL clustering code implemented.
  - Trigger algorithm update.
- A lot of refinements are still very desirable, some are critical:
  - Bug fixes (there are always new bugs introduced).
  - Good calibration of ECAL is critical, several implementations in progress.
  - Tracker alignment procedures needed much improvement.
    - Requires geometry re-work, (GBL code in Java convenient).
  - Monitoring improvements, now also adding offline data quality monitoring.
  - Data Catalog

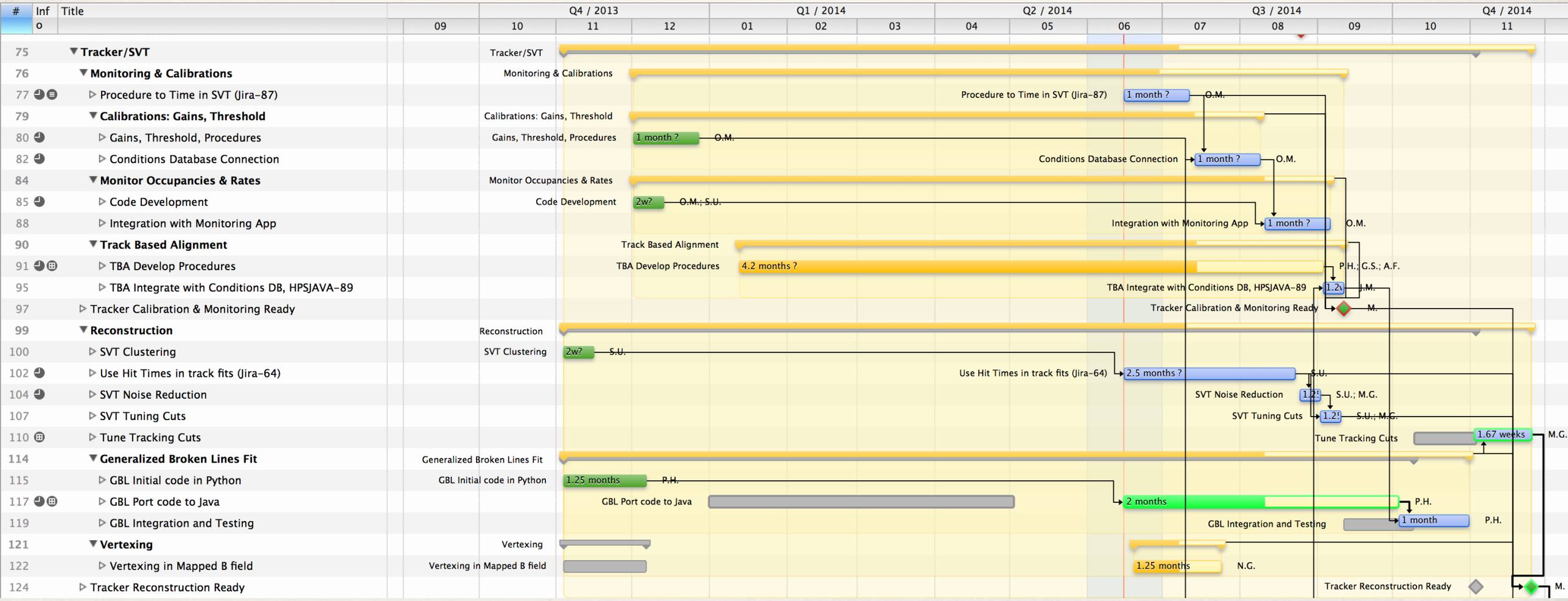
# Tracker Schedule, previous

- Several tasks falling behind originally planned schedule.
  - Hardware development takes precedence, same people.
- Track finding using single layers is de-scoped, postponed for later.
  - Note essential, but would give small tracking efficiency improvement.
- GBL code is working in C++, port to Java is delayed.



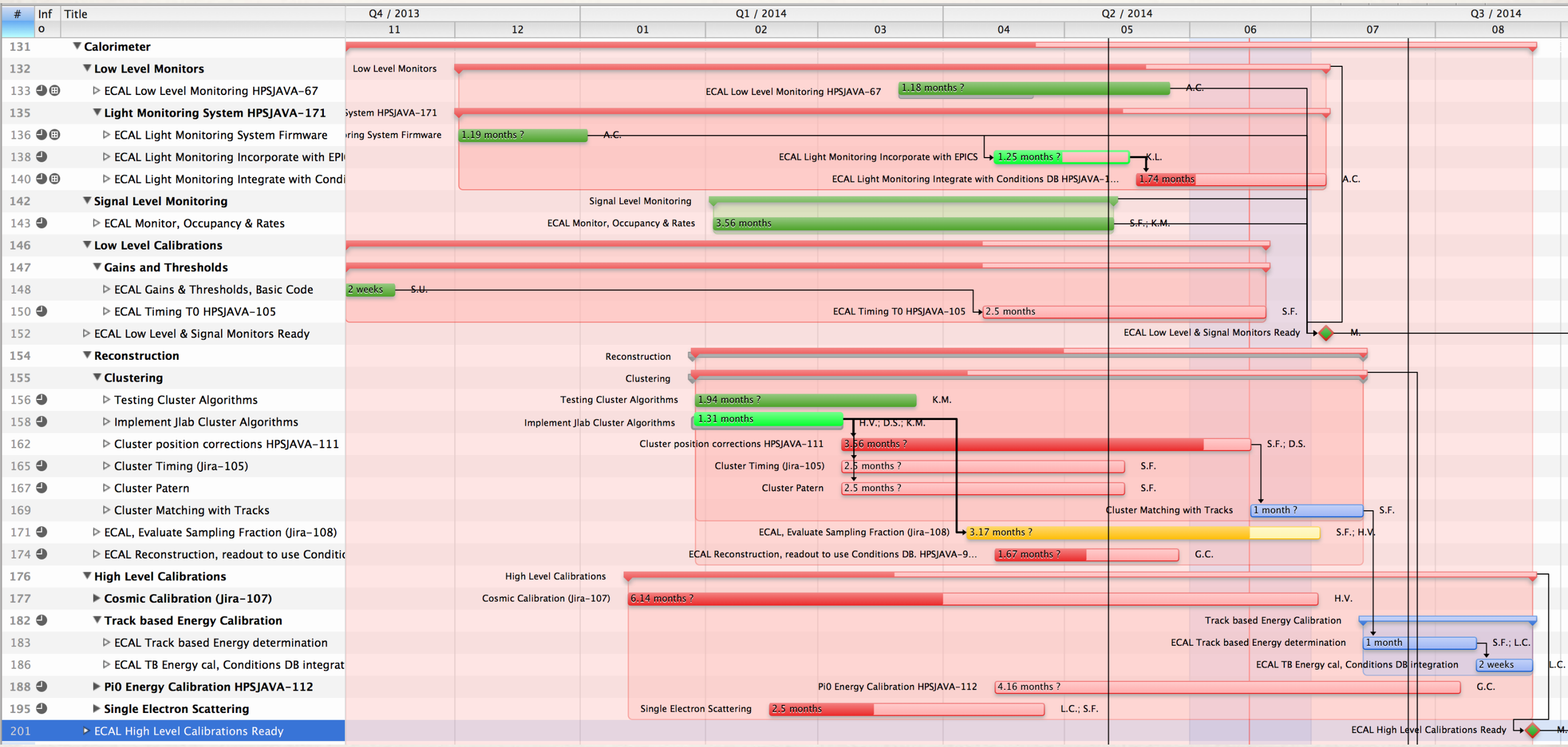
# Tracker Schedule, updated

- Using hit times in tracking, GBL to Java, are delayed.
- Tracker calibration and monitoring ready stays at mid September.
- Tracker reconstruction ready moves from Sept 19 to November 25.
  - Biggest contribution to shift is GBL port.



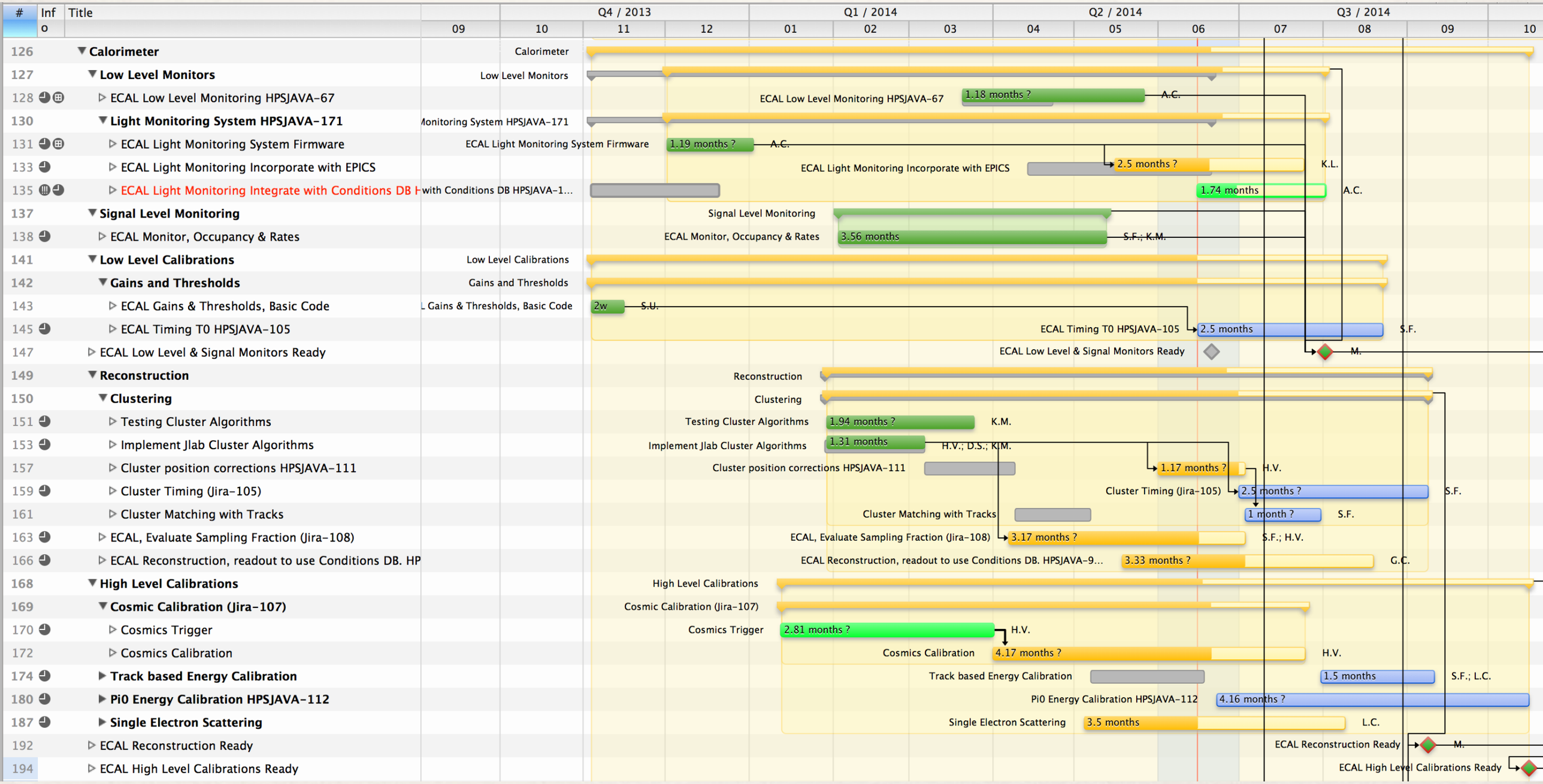
# Calorimeter Schedule, original

- Schedule was too optimistic about completion dates for calibrations.
- January Software workshop got a number of new people up to speed.



# Calorimeter Schedule, updated

- Several tasks take longer or are started later.
- Calorimeter software ready moves from Aug 25 to Oct 16th.



# Accelerating the Schedule

- We can realistically do better on this schedule!
- Bottlenecks for readiness are:
  - GBL to Java port.
    - This is not essential for alignment, the C++ code works.
    - With some (real) effort, could be finished by early September.
  - Tracker alignment codes
    - We have a working system, but would like something better, easier to use.
    - Could be finished by early September.
  - Pi-0 calibration of Ecal.
    - We really do want this component.
    - It is not trivial, but development can be accelerated: finish early September.
    - Already, more people are contributing to this task.
    - We also have track based ECAL calibration.
- ➡ Software can be ready on October 1st.
  - And then we keep improving on it…

# Summary & Conclusions

- Test run results show that the basics are all in place, for some time!
- An awful lot has been accomplished!
- The software team had new members come up to speed since January, they are now successfully contributing.
- There is some (significant) slip in the schedule.
  - Ready for data would move from Sept 20 to Nov 25th.
  - Speeding up some tasks, **ready for data will be Oct. 1st.**
  - Cannot tolerate further schedule slips.
- The software system *is* coming together and will be completed before hardware is ready.
- Software getting more stringently exercised with Mock Data Challenge.