Hall A Analysis Software Status

Ole Hansen

Hall A Collaboration Meeting
January 19, 2017

Status

- Latest stable version: 1.5.32 (1 Nov 2016) ▸ web
  - ROOT 6 compatible
  - Bugfixes
- Preview release: 1.6-beta3 ▸ web
  - **New database format!** Use `dbconvert` utility to convert old to new.
  - Many new features (see next), not all fully implemented/tested yet.
  - Hope to finalize by summer 2017 for fall run (yes, it's delayed . . .)
  - Preliminary Release Notes ▸ web
- Development ▸ GitHub
  - For experts. Things may change unexpectedly.
  - Download:

    ```
    git clone https://github.com/JeffersonLab/analyzer.git
    ```

New in 1.6: Completed Items

- Modular decoder (Bob Michaels) ✓
  - Easy to add support for new front-end electronics via plug-ins.
  - Processing of different trigger types configurable via "event type handlers".
- ROOT 6 support ✓
- EVIO library unbundled, now easily upgradeable ✓
- Miscellaneous ✓
  - Improved formula & test package (removed limitations).
  - Simulation event data decoder API and example implementation.
  - `scons` build system.
  - Rewritten, modular hardware channel decoder (`THaDecData`).
  - Many small performance improvements.

New in 1.6: Work In Progress

- Universal database interface ◐
  - All analysis modules now use this interface.
  - Users must convert their existing databases. Conversion utility program available (`utils/dbconvert`, largely complete) ▸ doc
  - Will eventually support multiple backends (text files, Hall C parameters, SQL database, CCDB, etc.) Currently, have backend for Hall A-style text files.
- Improved VDC track reconstruction ◐
  - Known bugs fixed
  - Reconstruction of multi-cluster events should be greatly improved.
  - Needs testing/optimization. Testers welcome.
  - Old code will remain available as an alternative tracking algorithm.
- Support for 12 GeV pipelined electronics (FADC250, F1TDC, etc.) ◐
- Test suite ◐
  - "`make test`" runs tests to check correctness of results, find bugs, etc.
  - Standard practice, employed by most modern software packages.

Database Format Conversion

Old Fixed-Format Database db_L.s1.dat

```
Number of Left Scintillator 1 paddles --------------------
       6
Crate,Slot,1st,Last ADC chans,Beg S1 chan, model ----------------
       3      18      6        11      1      1881 - ADCs pads 1-6  (right)
       3      18      0        05      7      1881 - ADCs pads 7-12 (left)
       3      10      88       93      1      1877 - TDCs pads 1-6  (right)
       3      10      80       85      7      1877 - TDCs pads 7-12 (left)
      -1       0       0       0       0
X,Y,Z coords (in m) of S1 front plane in spectrom cs --------------
   -0.129     0.0   1.2873                              - Meters
Half of X, half of Y, full Z sizes (in m) of S1 ----------------
    0.88    0.18     0.005                              - Meters
TDC time offsets of S1 in TDC channels ---------------------
     2.45 6.38 7.58 3.78 -13.25 3.75                    - Left Paddles
    -14.13 -16.83 -0.40 -3.78 -22.70 -0.12              - Right Paddles
```

`dbconvert DB-old/ DB-new/` (converts entire directory) $\longrightarrow$

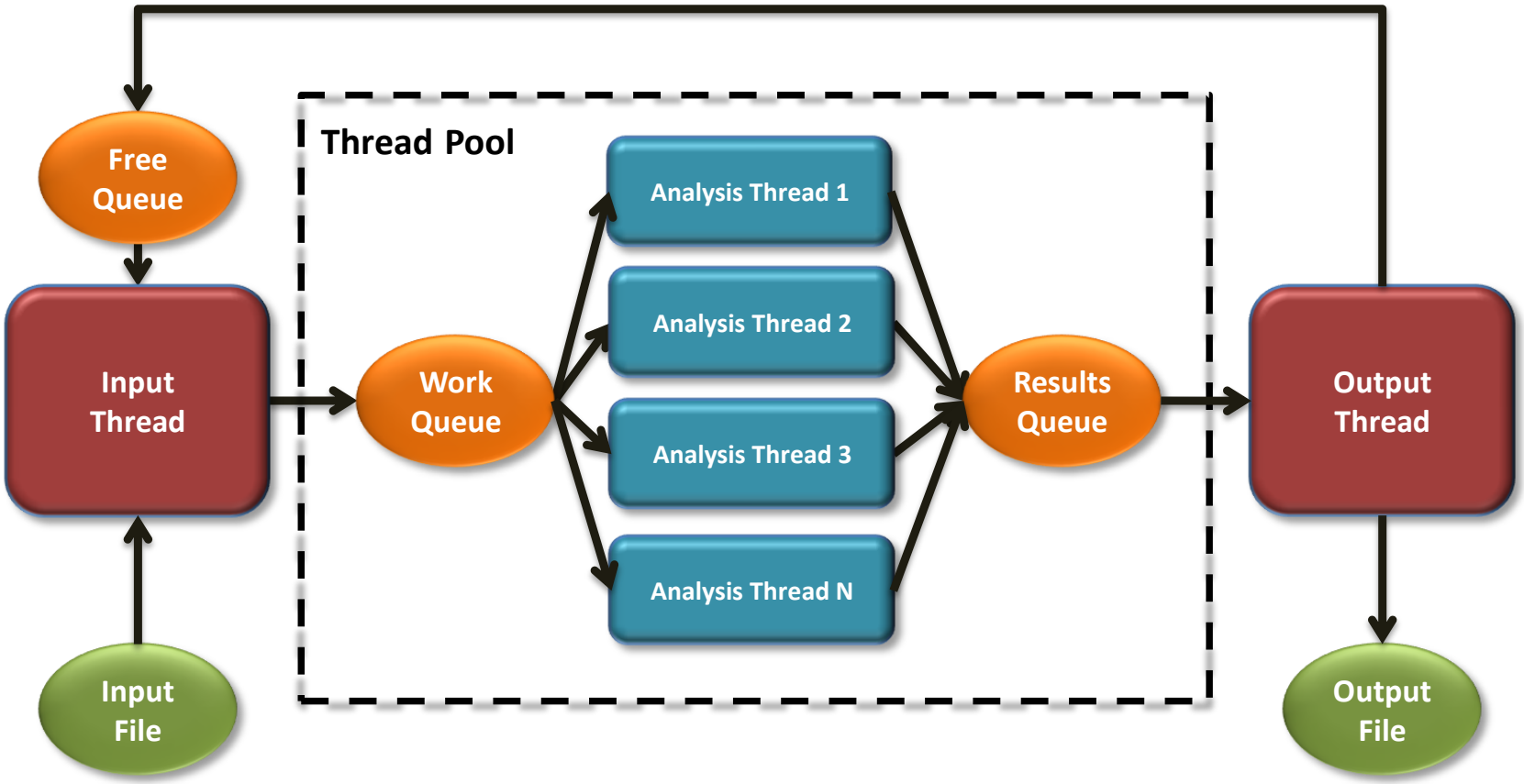
New Free-Format Key-Value Database

```
----[ 2015-03-24 00:00:00 -0400 ]
L.s1.detmap =  3    18     6    11     1  1881
               3    18     0     5     7  1881
               3    10    88    93     1  1877
               3    10    80    85     7  1877
L.s1.npaddles = 6
L.s1.position = -0.1290  0.0000  1.2873
L.s1.size = 0.88  0.18  0.005
L.s1.L.off = 2.45  6.38  7.58  3.78  -13.25  3.75
L.s1.R.off = -14.13  -16.83  -0.40  -3.78  -22.70  -0.12
```

Roadmap

Task	Status	Prio
Object-oriented decoder — plug-ins for front-end electronics	Done	
Improved VDC tracking efficiency in presence of noise	Done	
Automated/nightly builds	Done	
ROOT 6 support	Done	
Consistent key/value organization of databases	Done	
Generic database API	In progress (50%)	Low
Integrate code validation procedures (cppcheck, valgrind) in build process. Automated binary comparisons of results (“make check”)	Beta (80%)	Med
Decoder support for multiblock data from pipelined electronics	Beta (80%)	Med
Analyzer Release 1.6	Beta (90%)	High
Metadata class, self-documenting output	In progress (20%)	Low
Event-level parallelization	In progress (20%)	High
High-rate VDC track reconstruction for APEX	Beta (90%)	Med
Multiplexing decoder for A_1^n (if experiment is scheduled)	Prototype (10%)	Med
Detailed SBS requirements & development timeline	Done	
High-background GEM track reconstruction, simulations & analysis	In progress (25%)	High

Longer Term: Analyzer Parallelization



- Thread Pool with three thread-safe queues
- Queues hold working sets: event object, analysis chain & modules
- Option to sync event stream at certain events (*e.g.* scaler events, run boundaries)
- Option to preserve strict event ordering (at a performance penalty)

HRS Analysis Unlikely to Benefit from Parallelization

Typical HRS reconstruction & analysis timing

```
Number of output variables: 56  (relatively small)

314292  events read
313476  physics events analyzed

Timing summary:
Init      : Real Time =   0.84 seconds Cpu Time =   0.79 seconds  startup time per run
RawDecode : Real Time =   2.36 seconds Cpu Time =   1.96 seconds  6.2 us/event
Decode    : Real Time =   2.04 seconds Cpu Time =   2.27 seconds  7.2 us/event
CTracking : Real Time =   1.07 seconds Cpu Time =   1.06 seconds  3.4 us/event
CDetRecon : Real Time =   0.87 seconds Cpu Time =   0.73 seconds  2.3 us/event
Tracking  : Real Time =   2.87 seconds Cpu Time =   2.69 seconds  8.6 us/event
DetRecon  : Real Time =   0.54 seconds Cpu Time =   0.43 seconds  1.3 us/event
Physics   : Real Time =   0.59 seconds Cpu Time =   0.61 seconds  1.9 us/event
Output    : Real Time =  10.91 seconds Cpu Time =  11.38 seconds 36.2 us/event
Cuts      : Real Time =   2.54 seconds Cpu Time =   2.57 seconds  8.2 us/event
Total     : Real Time =  29.64 seconds Cpu Time =  29.36 seconds
```

$$\begin{aligned} \text{Non-parallelizable time} &= 36.2\ \mu\text{s/event} \\ \text{Parallelizable time} &= 39.1\ \mu\text{s/event} \\ \text{Parallelizable fraction } p &= 52\% \\ \text{Maximum expected speedup } S &= 1/(1-p) = 2.1 \end{aligned}$$

Resources

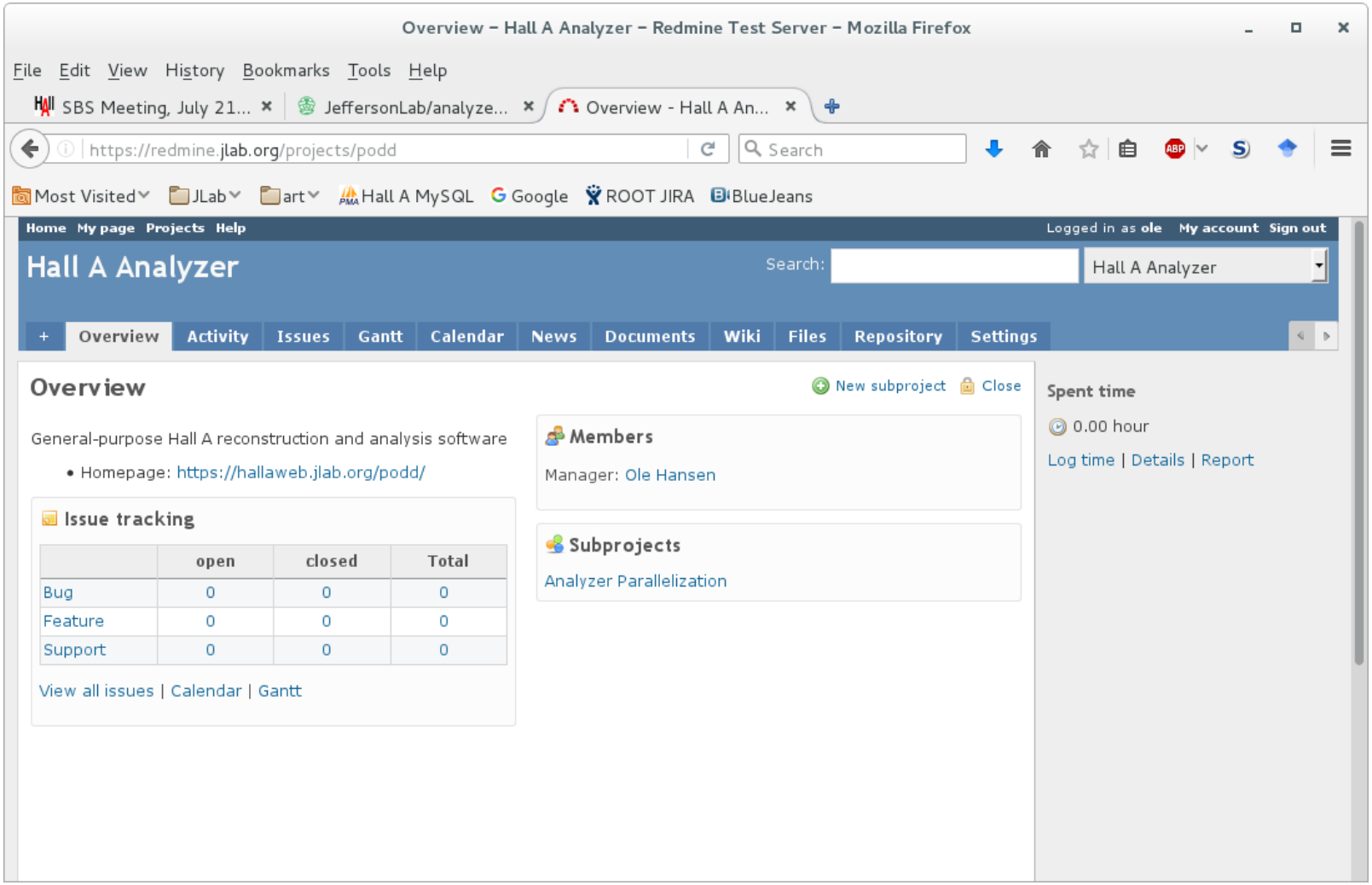
- Web site ▸ home page
  - Documentation
  - Release Notes
  - Software Development Kit (SDK)
  - Source code downloads
  - Archived tutorials & example replays
- Bug tracker ▸ GitHub
- Bi-weekly Hall A/C software meeting: Wednesdays, 11am, L210A
- Mailing list: `halla_software@jlab.org`. Subscribe on ▸ mailman
- Analysis Workshop archive ▸ archive (includes tutorials)

Analysis Workshop

- Next workshop planned for Summer 2017 (in preparation for fall run)
- Topics under consideration
  - `Podd/hcana` introduction & news
  - Hands-on tutorials w/ example replays
  - Collected wisdom for setting up the software for a new experiment
  - ROOT Tips & Tricks
  - Optics optimization howto
  - Introduction to JLab scicomp resources & batch farm
- Announcement later this spring
- Suggestions & contributions welcome

Improving Project Management?

- GitHub bug tracker rather basic, considering Redmine or similar tool
- Would provide a single location for practically all resources



Backup

Self-Documenting Output: Candidates for inclusion

Candidate metadata items to write to replay output (preliminary)

- Experiment name & run number
- Full paths of input & output file(s)
- Replay time, host, user, work directory
- Analyzer version (version & git commit)
- ROOT version loaded at run time
- EVIO version loaded at run time
- Build time, host, compiler, user, directory
- Build flags (CXXFLAGS, others)
- Database URL (DB_DIR, SQL host, etc.)
- All actually used database keys & values (typ. $\mathcal{O}(1000)$)
- Replay script? Copy of source?
- Copies of output & cut definition files
- All parsed output definitions
- Optional comment/description created at replay time
- Analyzer class, decoder class
- Analyzer runtime flags (scalers, helicity)

Desired level of detail configurable in analysis script.