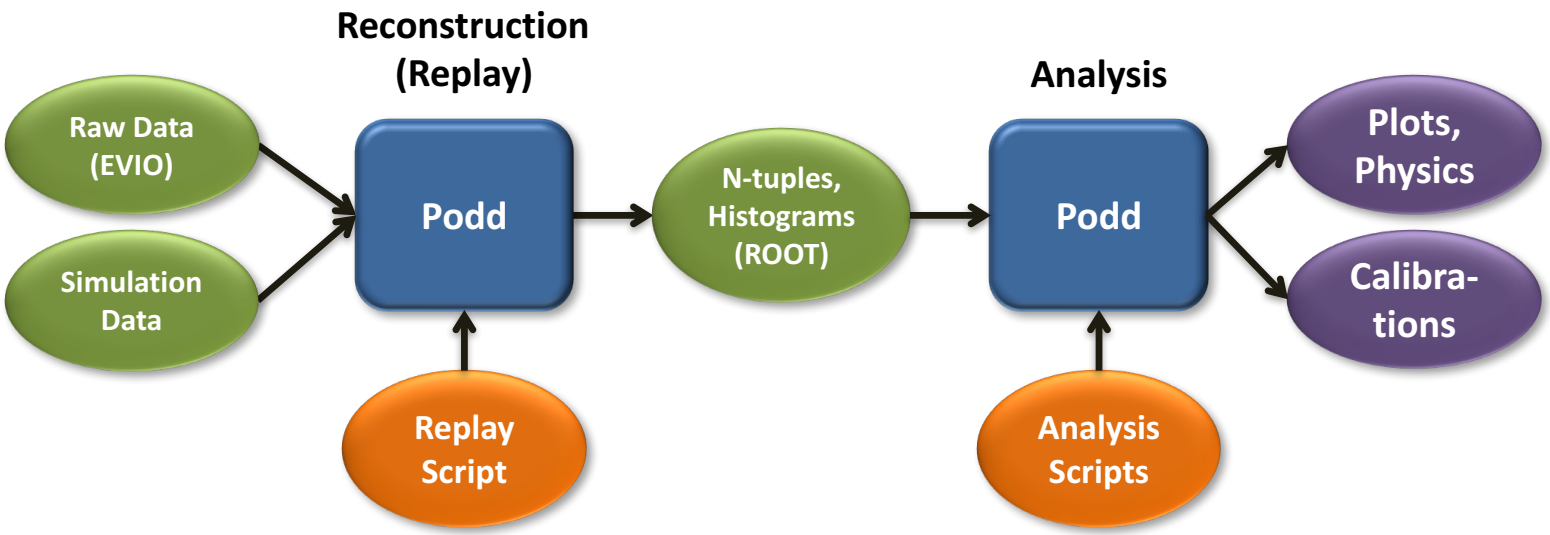
2017 Highlights of Hall A & C Reconstruction Software Development

Ole Hansen

Jefferson Lab

Computing Roundtable
December 5, 2017

The Hall A & C Analyzer ("Podd"): Data Flow



1. **Reconstruction (Replay)**
   - Runs in ROOT interpreter (`analyzer` prompt)
   - Calls mostly Podd functions & classes
   - Scripts usually set up by experiment experts or advanced users
   - After setup, usually runs in mass replay on the farm
2. **Analysis**
   - Also runs in ROOT interpreter (`analyzer` prompt)
   - Calls mostly ROOT functions and classes (but may need Podd classes)
   - Done by everyone on the experiment
   - Calibration and final physics usually done here

Podd Reconstruction: Typical Steps

- Raw data decoding
- Calibration (gains, offsets)
- Tracking in one or more spectrometers
- Vertex reconstruction using (inverted) Transport matrices
- PID likelihood analysis
- Retrieval of beam parameters, EPICS fields, scalers & other control data
- Basic physics computations (kinematics)
- Basic corrections (energy loss, etc.)

Each item is only executed if requested (configured)

Example Analysis Script (Hall C)

```
void replay_workshop_example(Int_t RunNumber=0, Int_t MaxEvent=0) {

  // Get RunNumber and MaxEvent if not provided.
  if(RunNumber == 0) {
    cout << "Enter a Run Number (-1 to exit): ";
    cin >> RunNumber;
    if( RunNumber<=0 ) return;
  }
  if(MaxEvent == 0) {
    cout << "\nNumber of Events to analyze: ";
    cin >> MaxEvent;
    if(MaxEvent == 0) {
      cerr << "...Invalid entry\n";
      exit;
    }
  }
  // Create file name patterns.
  const char* RunFileNamePattern = "raw/shms_all_%05d.dat";
  const char* ROOTFileNamePattern = "ROOTfiles/shms_replay_%d_%d.root";
  // Add variables to global list.
  gHcParms->Define("gen_run_number", "Run Number", RunNumber);
  gHcParms->AddString("g_ctp_database_filename", "DBASE/standard.database");
  // Load varibles from files to global list.
  gHcParms->Load(gHcParms->GetString("g_ctp_database_filename"), RunNumber);
  // g_ctp_parm_filename and g_decode_map_filename should now be defined.
  gHcParms->Load(gHcParms->GetString("g_ctp_kinematics_filename"), RunNumber);
  gHcParms->Load(gHcParms->GetString("g_ctp_parm_filename"));
  // Load params for SHMS trigger configuration
  gHcParms->Load("PARAM/TRIG/tshms.param");

  // Load the Hall C style detector map
  gHcDetectorMap = new THcDetectorMap();
  gHcDetectorMap->Load("MAPS/SHMS/DETEC/shms_stack.map");

  // Add trigger apparatus
  THaApparatus* TRG = new THcTrigApp("T", "TRG");
  gHaApps->Add(TRG);
  // Add trigger detector to trigger apparatus
  THcTrigDet* shms = new THcTrigDet("shms", "SHMS Trigger Information");
  TRG->AddDetector(shms);
  // Set up the equipment to be analyzed.
  THaApparatus* SHMS = new THcHallCSpectrometer("P", "SHMS");
  gHaApps->Add(SHMS);
  // Add drift chambers to SHMS apparatus
  THcDC* dc = new THcDC("dc", "Drift Chambers");
  SHMS->AddDetector(dc);
  // Add hodoscope to SHMS apparatus
  THcHodoscope* hod = new THcHodoscope("hod", "Hodoscope");
  SHMS->AddDetector(hod);
  // Add Heavy Gas Cherenkov to SHMS apparatus
  THcCherenkov* hgcer = new THcCherenkov("hgcer", "Heavy Gas Cherenkov");
  SHMS->AddDetector(hgcer);
  // Add Golden Track to physics modules.
  THaGoldenTrack* gtr = new THaGoldenTrack("P.gtr", "SHMS Golden Track", "P");
  gHaPhysics->Add(gtr);
```

```
// Add handler for prestart event 125.
THcConfigEvtHandler* ev125 = new THcConfigEvtHandler("HC", "Config Event type 125");
gHaEvtHandlers->Add(ev125);

// Set up the analyzer - we use the standard one,
// but this could be an experiment-specific one as well.
// The Analyzer controls the reading of the data, executes
// tests/cuts, loops over Acpparatus's and PhysicsModules,
// and executes the output routines.
THcAnalyzer* analyzer = new THcAnalyzer;

// A simple event class to be output to the resulting tree.
// Creating your own descendant of THaEvent is one way of
// defining and controlling the output.
THaEvent* event = new THaEvent;

// Define the run(s) that we want to analyze.
// We just set up one, but this could be many.
char RunFileName[100];
sprintf(RunFileName, RunFileNamePattern, RunNumber);
THaRun* run = new THaRun(RunFileName);

// Eventually need to learn to skip over, or properly analyze
// the pedestal events
run->SetEventRange(1, MaxEvent);    // Physics Event number, does not
                                    // include scaler or control events.
run->SetNscan(1);
run->SetDataRequired(0x7);
run->Print();

// Define the analysis parameters
TString ROOTFileName = Form(ROOTFileNamePattern, RunNumber, MaxEvent);
analyzer->SetCountMode(2);    // 0 = counter is # of physics triggers
                              // 1 = counter is # of all decode reads
                              // 2 = counter is event number
analyzer->SetEvent(event);
analyzer->SetCrateMapFileName("MAPS/db_cratemap.dat");
analyzer->SetOutFile(ROOTFileName.Data());
analyzer->SetOdefFile("DEF-files/SHMS/GEN/workshop_example.def");
analyzer->SetCutFile("DEF-files/SHMS/GEN/pstackana_cuts.def");    // optional

// File to record cuts accounting information
//analyzer->SetSummaryFile("summary_example.log");    // optional

// Start the actual analysis.
analyzer->Process(run);
// Create report file from template.
//analyzer->PrintReport(    // optional
//  "TEMPLATES/dcana.template",
//  Form("REPORT_OUTPUT/replay_shms_%05d.report", RunNumber)
//);
```

Podd: Main Accomplishments 2017

- Consistent key/value **database** (configuration & conditions)
- **Output** system performance and efficiency improvements
- Better HRS spectrometer **VDC tracking** (noise resistance)

Podd: Database Example

Validity timestamp ⟶

Scalar key/value pairs ⟶

Array key/values (auto-sized) ⟶

```
-------[ 2016-02-05 00:00:00 -0500 ]

R.vdc.nwires = 368
R.vdc.wire.spacing = -0.0042426
R.vdc.tdc.min = 800
R.vdc.tdc.max = 2200
R.vdc.tdc.res = 5e-10
R.vdc.ttd.param =
    2.12e-03   0.00e+00   0.00e+00   0.00e+00
   -4.20e-04   1.30e-03   1.06e-04   0.00e+00
    4.00e-09
R.vdc.t0.res = 6e-08
R.vdc.clust.minsize = 4
R.vdc.clust.maxspan = 7
R.vdc.maxgap = 0
R.vdc.tdiff.min = 3e-08
R.vdc.tdiff.max = 1.5e-07
R.vdc.u1.detmap =
     2    7    0   95    0
     2    8    0   95   96
     2    9    0   95  192
     2   10    0   79  288
R.vdc.u1.position = 0  0  0
R.vdc.u1.wire.start = 0.77852
R.vdc.u1.wire.angle = -45
R.vdc.u1.driftvel = 50000
R.vdc.u1.tdc.offsets =
   1533.3  1533.3  1533.3  1533.3  1533.3  1533.3  1533.3  1533.3
   1533.3  1533.3  1533.3  1533.3  1533.3  1533.3  1533.3  1533.3
   1534.1  1534.1  1534.1  1534.1  1534.1  1534.1  1534.1  1534.1
   1534.1  1534.1  1534.1  1534.1  1534.1  1534.1  1534.1  1534.1
   1534.7  1534.7  1534.7  1534.7  1534.7  1534.7  1534.7  1534.7
   1534.7  1534.7  1534.7  1534.7  1534.7  1534.7  1534.7  1534.7
   1533.9  1533.9  1533.9  1533.9  1533.9  1533.9  1533.9  1533.9
   1533.9  1533.9  1533.9  1533.9  1533.9  1533.9  1533.9  1533.9
   1534.2  1534.2  1534.2  1534.2  1534.2  1534.2  1534.2  1534.2
   1534.2  1534.2  1534.2  1534.2  1534.2  1534.2  1534.2  1534.2
   1526.4  1526.4  1526.4  1526.4  1526.4  1526.4  1526.4  1526.4
   1526.4  1526.4  1526.4  1526.4  1526.4  1526.4  1526.4  1526.4
   1533.8  1533.8  1533.8  1533.8  1533.8  1533.8  1533.8  1533.8
   1533.8  1533.8  1533.8  1533.8  1533.8  1533.8  1533.8  1533.8
   1527.2  1527.2  1527.2  1527.2  1527.2  1527.2  1527.2  1527.2
   1527.2  1527.2  1527.2  1527.2  1527.2  1527.2  1527.2  1527.2
   1527.2  1527.2  1527.2  1527.2  1527.2  1527.2  1527.2  1527.2
   1527.2  1527.2  1527.2  1527.2  1527.2  1527.2  1527.2  1527.2
```

Retrieve parameter blocks in single call:

```
DBRequest request[] = {
  { "detmap",         &detmap,         kIntV },
  { "nwires",         &fNelem,         kInt,     0, 0, -1 },
  { "wire.start",     &fWBeg,          kDouble },
  { "wire.spacing",   &fWSpac,         kDouble,  0, 0, -1 },
  { "wire.angle",     &fWAngle,        kDouble,  0, 0 },
  { "wire.badlist",   &bad_wirelist,   kIntV,    0, 1 },
  { "driftvel",       &fDriftVel,      kDouble,  0, 0, -1 },
  { "tdc.min",        &fMinTime,       kInt,     0, 1, -1 },
  { "tdc.max",        &fMaxTime,       kInt,     0, 1, -1 },
  { "tdc.res",        &fTDCRes,        kDouble,  0, 0, -1 },
  { "tdc.offsets",    &tdc_offsets,    kFloatV },
  //...
  { 0 }
};
err = LoadDB( file, date, request, fPrefix );
```

Podd: User-Configurable N-tuple Output

Module header file:

```
  Int_t         GetNHits()         const { return fHits->GetLast()+1; }

protected:
  TClonesArray*  fWires;     // Wires
  TClonesArray*  fHits;      // Fired wires
  TClonesArray*  fClusters;  // Clusters

  Int_t fNHits;      // Total number of hits (including multihits)
  Int_t fNWiresHit;  // Number of wires with one or more hits
```

⇓

Module `DefineVariables()` method:

```
RVarDef vars[] = {
  { "nhit",   "Number of hits",              "GetNHits()" },
  { "wire",   "Active wire numbers",         "fHits.THaVDCHit.GetWireNum()" },
  { "rawtime","Raw TDC values of wires",     "fHits.THaVDCHit.fRawTime" },
  { "time",   "TDC values of active wires",  "fHits.THaVDCHit.fTime" },
  { "dist",   "Drift distances",             "fHits.THaVDCHit.fDist" },
  { "ddist",  "Drft dist uncertainty",       "fHits.THaVDCHit.fdDist" },
  { "trdist", "Dist. from track",            "fHits.THaVDCHit.ftrDist" },
  { "ltrdist","Dist. from local track",      "fHits.THaVDCHit.fltrDist" },
  { "trknum", "Track number (0=unused)",     "fHits.THaVDCHit.fTrkNum" },
  { "clsnum", "Cluster number (-1=unused)",    "fHits.THaVDCHit.fClsNum" },
  { "nclust", "Number of clusters",          "GetNClusters()" },
  { "clsiz",  "Cluster sizes",               "fClusters.THaVDCCluster.GetSize()" },
  { "clpivot","Cluster pivot wire num",      "fClusters.THaVDCCluster.GetPivotWireNum()" },
  { "clpos",  "Cluster intercepts (m)",      "fClusters.THaVDCCluster.fInt" },
  { "slope",  "Cluster best slope",          "fClusters.THaVDCCluster.fSlope" },
```

⇓

Output definition file:

```
# All RHRS VDC raw data (lots of information, only uncomment for debugging)
# block ${arm}.vdc.*
block ${arm}.vdc.u1.*
```

⇓

ROOT file `TTree`:

```
*Br   17 :L.vdc.u1.wire : data[Ndata.L.vdc.u1.wire]/D                    *
*Entries :   265460 : Total  Size=   11970559 bytes  File Size  =    2295168 *
*Baskets :       81 : Basket Size=    2683904 bytes  Compression=   5.21     *
*Br   18 :L.vdc.u1.nclust : L.vdc.u1.nclust/D                            *
*Entries :   265460 : Total  Size=    2136991 bytes  File Size  =      45063 *
*Baskets :      131 : Basket Size=      64512 bytes  Compression=  47.36     *
*Br   19 :L.vdc.u1.nhit : L.vdc.u1.nhit/D                                *
*Entries :   265460 : Total  Size=    2136721 bytes  File Size  =     184378 *
*Baskets :      131 : Basket Size=      64512 bytes  Compression=  11.57     *
analyzer [11] T->Print("L.vdc.*")
```

Podd: Output Improvements

Original:

- Only D data type
- Multiple, redundant array size variables with parallel arrays

```
*Br   15 :L.vdc.u1.time : data[Ndata.L.vdc.u1.time]/D                    *
*Entries :   265460 : Total  Size=   11970559 bytes  File Size  =    3662638 *
*Baskets :       81 : Basket Size=    2683904 bytes  Compression=   3.27     *
*Br   16 :L.vdc.u1.trdist : data[Ndata.L.vdc.u1.trdist]/D                 *
*Entries :   265460 : Total  Size=   11970731 bytes  File Size  =   10832974 *
*Baskets :       81 : Basket Size=    2683904 bytes  Compression=   1.10     *
*Br   17 :L.vdc.u1.wire : data[Ndata.L.vdc.u1.wire]/D                     *
*Entries :   265460 : Total  Size=   11970559 bytes  File Size  =    2295168 *
*Baskets :       81 : Basket Size=    2683904 bytes  Compression=   5.21     *
*Br   18 :L.vdc.u1.nclust : L.vdc.u1.nclust/D                             *
*Entries :   265460 : Total  Size=    2136991 bytes  File Size  =      45063 *
*Baskets :      131 : Basket Size=      64512 bytes  Compression=  47.36     *
*Br   19 :L.vdc.u1.nhit : L.vdc.u1.nhit/D                                 *
*Entries :   265460 : Total  Size=    2136721 bytes  File Size  =     184378 *
*Baskets :      131 : Basket Size=      64512 bytes  Compression=  11.57     *
analyzer [2]
```

Improved:

- Full range of data types
- Automatic detection of parallel arrays, only one size variable (with limitations)
- Automatic basket size adjustment (overriding ROOT default)

```
analyzer [6] T->Print("L.vdc.*")
*****************************************************************************
*Tree    :T         : Hall A Analyzer Output DST                            *
*Entries :   280231 : Total =       325369054 bytes  File  Size =  186808850 *
*          :          : Tree compression factor =   1.74                    *
*****************************************************************************
*Br    0 :L.vdc.u1.nhit : L.vdc.u1.nhit/I                                  *
*Entries :   280231 : Total  Size=    1122414 bytes  File Size  =     160443 *
*Baskets :       12 : Basket Size=     537088 bytes  Compression=   6.99     *
*Br    1 :L.vdc.u1.wire : L.vdc.u1.wire[Ndata.L.vdc.u1.wire]/I             *
*Entries :   280231 : Total  Size=    7420715 bytes  File Size  =    2064016 *
*Baskets :       49 : Basket Size=    1556480 bytes  Compression=   3.59     *
*Br    2 :L.vdc.u1.rawtime : L.vdc.u1.rawtime[Ndata.L.vdc.u1.wire]/I       *
*Entries :   280231 : Total  Size=    7420874 bytes  File Size  =    3614951 *
*Baskets :       49 : Basket Size=    1556992 bytes  Compression=   2.05     *
*Br    3 :L.vdc.u1.time : L.vdc.u1.time[Ndata.L.vdc.u1.wire]/D             *
*Entries :   280231 : Total  Size=   13718165 bytes  File Size  =    4116377 *
*Baskets :       80 : Basket Size=    2575872 bytes  Compression=   3.33     *
*Br    4 :L.vdc.u1.dist : L.vdc.u1.dist[Ndata.L.vdc.u1.wire]/D             *
*Entries :   280231 : Total  Size=   13718165 bytes  File Size  =   12032644 *
*Baskets :       80 : Basket Size=    2575872 bytes  Compression=   1.14     *
```

Podd: VDC Algorithm Improvements

Version 1.5.38

- Disallow UV ambiguities (configurable)
- UV fiducial cut
- Proper lower-upper matching cut
- Disallow cluster sharing

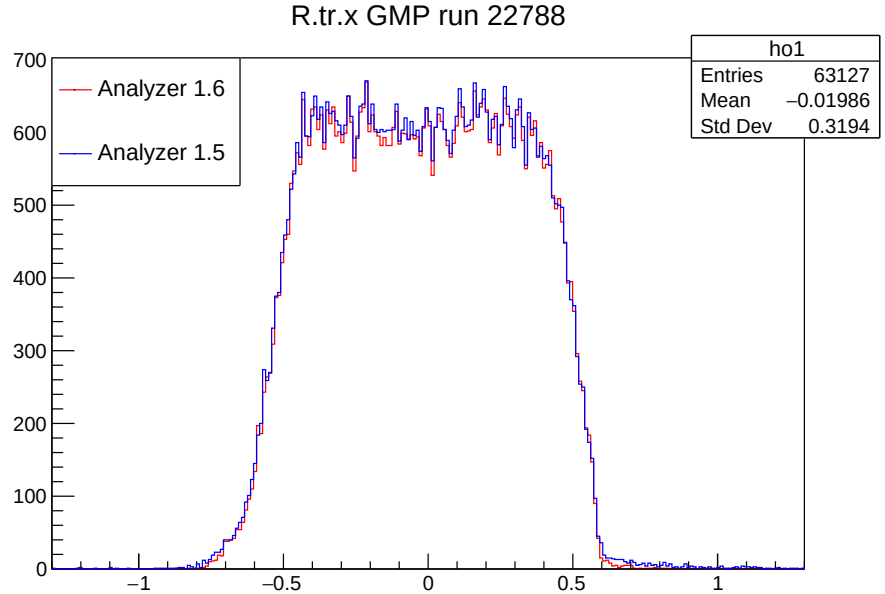
→ Guarantees clean single track at expense of slightly lower tracking efficiency

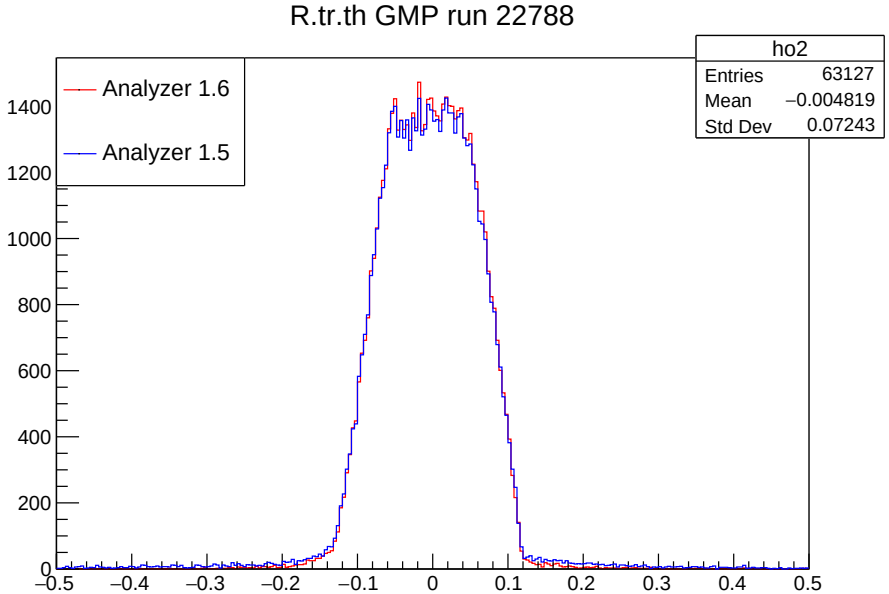
Version 1.6

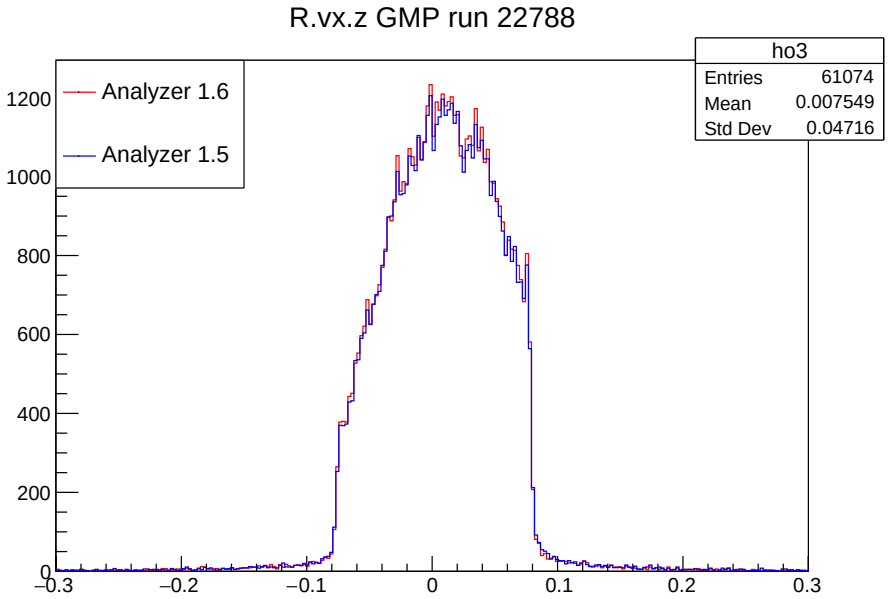
- Cluster shape analysis
- Overlapping cluster splitting
- 3-parameter cluster fit
- Cluster t_0 cut
- UV fiducial cut
- Proper lower-upper matching cut
- Disallow cluster sharing
- Old VDC code for reference

→ Allows multi-tracks, improves tracking efficiency, high-rate capable

Podd: VDC Tracking Comparison v1.5 vs. v1.6







Improved algorithm:

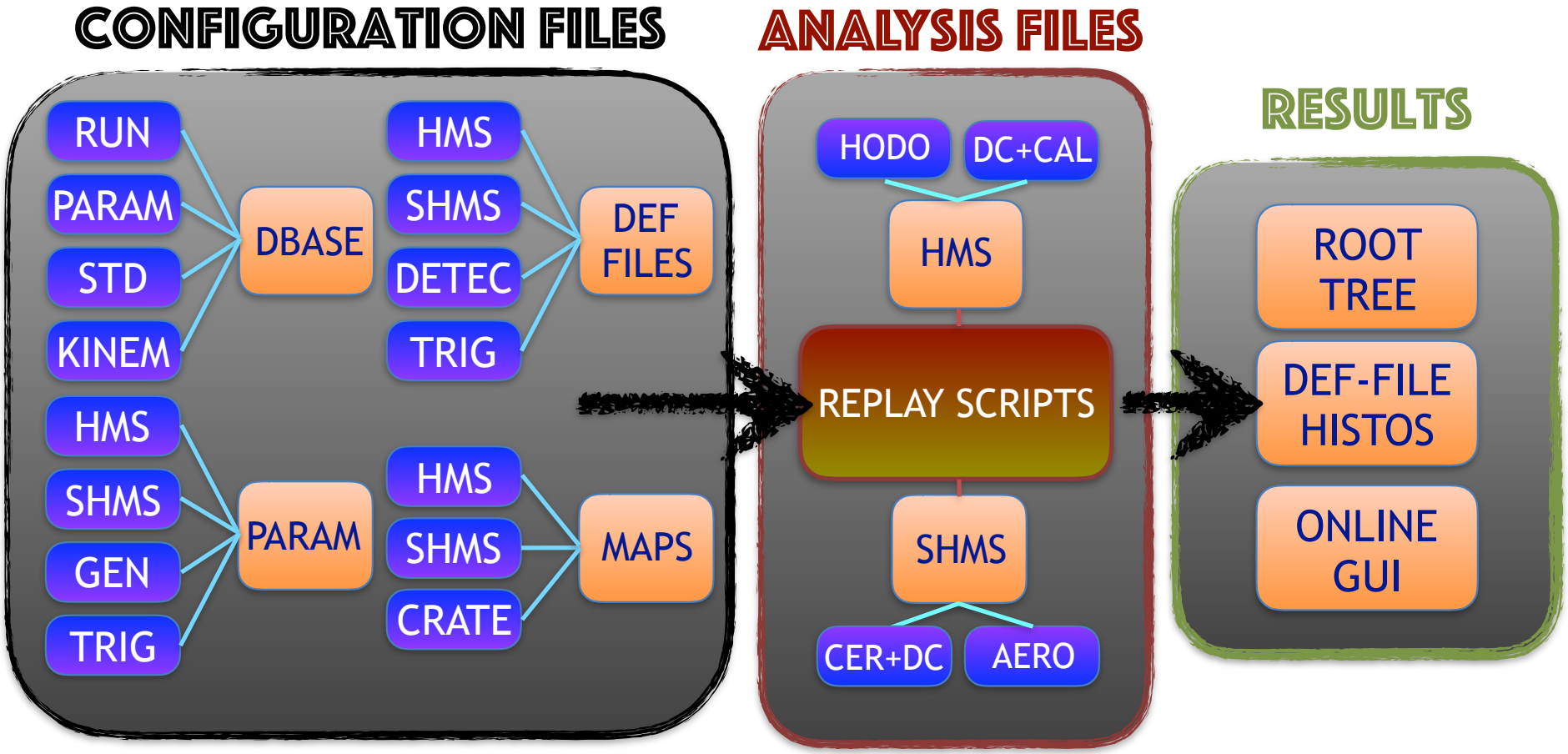
- Fewer reconstruction errors: smaller tails, larger signal
- No need for explicit cuts on "clean" events
- Should be particularly helpful with high-rate/high-noise data (to be tested)

Hall C: 2017 Accomplishments

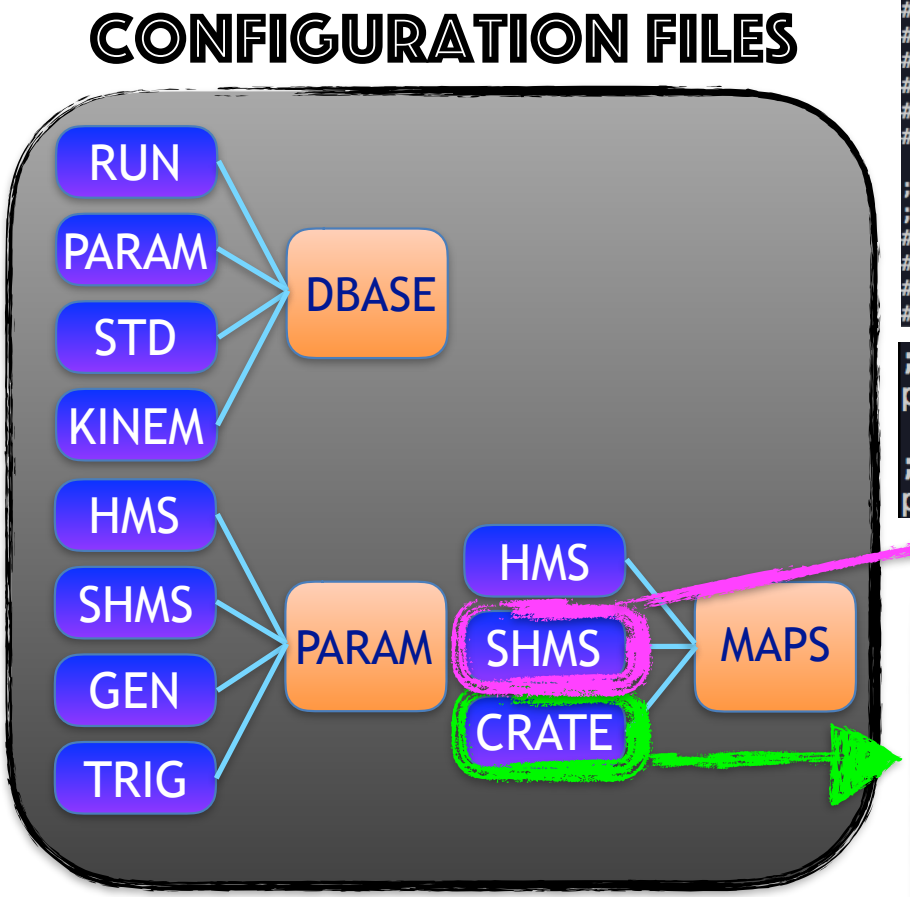
Hall C's replay software ("hcana") is a direct extension of Podd

- Got software and replay **ready** for, and demonstrated performance in Spring KPP run
- Many small **bugfixes** and usability improvements over past few months. Many of these have benefited Hall A as well.
- Very well organized **replay setup** (databases, configuration parameters, standard replay scripts, output & cut definitions, etc.)

Hall C Replay Overview (slides courtesy of E. Pooser)



Hall C Replay Configuration Files




```
; HMS detector specific paramter files
#include "PARAM/HMS/AERO/haero.param"
#include "PARAM/HMS/CAL/hcal.pos"
#include "PARAM/HMS/CAL/hcal.param"
#include "PARAM/HMS/CER/hcer.param"
#include "PARAM/HMS/DC/hdc.param"
#include "PARAM/HMS/DC/hdc.pos"
#include "PARAM/HMS/DC/hdc_tracking.param"
#include "PARAM/HMS/DC/hdriftmap.param"
#include "PARAM/HMS/HODO/hhodo.pos"
#include "PARAM/HMS/HODO/hhodo.param"

; General SHMS parameter files
; Note: shmsflags.param includes spectrome
#include "PARAM/SHMS/GEN/pcana.param"
#include "PARAM/SHMS/GEN/pdebug.param"
#include "PARAM/SHMS/GEN/shmsflags.param"
#include "PARAM/SHMS/GEN/ptracking.param"
```

```
1-99999
gpbeam=6.4
gtarg_num = 1
htheta_lab = 15.
ptheta_lab = 15.
hpcentral = 3.
ppcentral = 3.
hpartmass = 0.00051099
ppartmass = 0.00051099
```

```
; Number of heavy gas Cherenkov PMT's
phgcer_tot_pmts = 4

; Garth H. gain calibration from run 486, March 9 2017
phgcer_adc_to_npe = 1/436., 1/393., 1/364., 1/372.
```

```
==== Crate 2 type vme Bank Decoding
# slot  model   bank
  3     250     250
  4     250     250
  5     250     250
  6     250     250
  7     250     250
  8     250     250
  9     250     250
  10    250     250
  13    250     250
  14    250     250
  18    1190    1190
  19    1190    1190
  20    1190    1190
```

```
! SAERO_ID=25      ::  ::,ADC+,ADC-

DETECTOR=25

ROC=2

SLOT=10
  10,  1,  1,  0  ! sAER01+
  11,  1,  2,  0  ! sAER02+
  12,  1,  3,  0  ! sAER03+
  13,  1,  4,  0  ! sAER04+
  14,  1,  5,  0  ! sAER05+
  15,  1,  6,  0  ! sAER06+

SLOT=11
   0,  1,  7,  0  ! sAER07+
   1,  1,  1,  1  ! sAER01-
   2,  1,  2,  1  ! sAER02-
   3,  1,  3,  1  ! sAER03-
   4,  1,  4,  1  ! sAER04-
   5,  1,  5,  1  ! sAER05-
   6,  1,  6,  1  ! sAER06-
   7,  1,  7,  1  ! sAER07-
```

Hall C Replay Repository

Open issues by version - Hall | Changelog 3 4 - Redmine | JeffersonLab/hallc_replay: Re

https://github.com/JeffersonLab/hallc_replay

This repository | Search | Pull requests | Issues | Marketplace | Explore

JeffersonLab / hallc_replay

Watch 17 | Star 1 | Fork 30

Code | Issues 4 | Pull requests 1 | Projects 0 | Wiki | Insights | Settings

Replay directory for Hall C hcana

Edit

Add topics

616 commits | 2 branches | 1 release | 17 contributors

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

johnmattr committed with pooser Update onlineGUI configs to point to new macro locations from commit #... | Latest commit 16ad411 7 hours ago

CALIBRATION	Devel (#288)	2 days ago
DATFILES	New optics reconstruction matrix for KPP run	6 months ago
DBASE	Update SHMS KPP hodo	2 days ago
DEF-files	Hms online monitoring (#298)	9 hours ago
MAPS	Trigger work (#296)	a day ago
PARAM	Trigger work (#296)	a day ago
SCRIPTS	Trigger work (#296)	a day ago
TEMPLATES	Trigger work (#296)	a day ago
onlineGUI	Update onlineGUI configs to point to new macro locations from commit #...	7 hours ago
.gitignore	Pooser coin work (#286)	4 days ago
.rootrc	Updating repo	9 months ago
setup.csh	Rearrange db_run.dat and db_cratemap.dat	a year ago
setup.sh	Added setup.sh .	a year ago

Help people interested in this repository understand your project by adding a README. | Add a README

Summary

- Hall A/C software has seen significant improvements in 2017
- Several long-standing annoyances with the Hall A wire chamber track reconstruction have been corrected
- Hall C's replay setup has been very well organized and is ready for the upcoming Dec–Mar commissioning/physics runs
- Because of the code sharing between Hall A and C, general code improvements immediately benefit both Halls