

Summary/Status of LArSoft

Eric, Yale

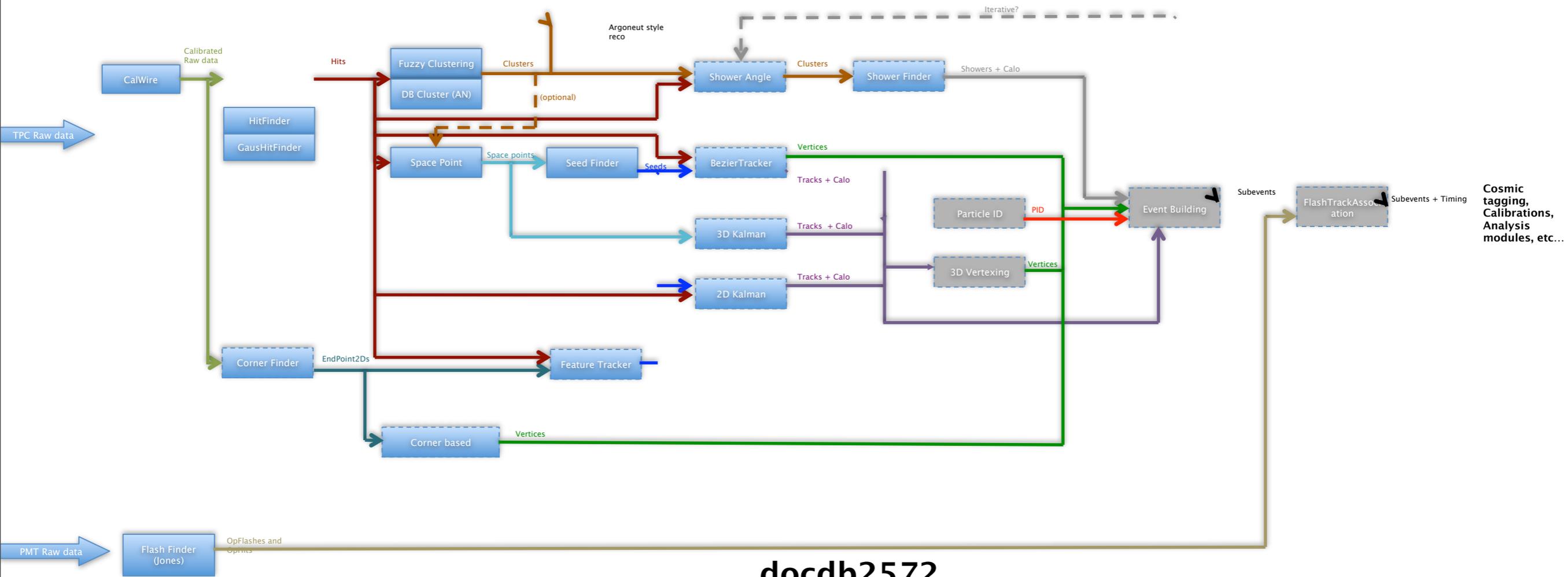
7–Nov–2013

- What is LarSoft?
 - Set of libraries

- LArSoft Today
 - Releases
 - Monte Carlo challenges
 - Timeframes

- LArSoft in near-ish future

- Offsite users
 - today
 - ~2months hence



LArSoft classes are in the Blue boxes (modules)
(LArSoft classes inherit from ART classes)

ART is the framework:
the glue represented by the lines

Libraries

□ LArSoft

- `/grid/fermiapp/lbne/lar/code/larsoft/releases/development/lib/Linux/`
 - `libRecoBase.so`
 - `libRecoAlg.so`
 - `libSimWireMicroBoone_module.so`
 - `libTrackAna_module.so`

 - `libDisambigCheater_module.so` WTF?! LBNE-specific code!

 - ... and ~200 others

Other libraries

- LArSoft modules call GEANT4, GENIE, ROOT, CLHEP, ... libraries
 - /nusoft/app/externals/genie/v2_8_0
 - /nusoft/app/externals/clhep/v2_1_3_1/

- These are so-called ups areas. Versions of src, headers, libs, bins for many externals preserved in a consistent way and available upon “setting up.”

- LArSoft indeed at compile/link time picks a particular set of versions

ART

- LArSoft's whole modular structure -- produce and analyze methods, services, and the fhicl job and parameter scripting, etc -- all rests on ART.
- It is itself just another ups package
 - [/nusoft/app/externals/art/v1_06_00](#)

another example: Pandora

- You may have heard about this!
- Cambridge is releasing its reconstruction classes, algorithms in a ups package
 - `/nusoft/app/externals/pandora/v00_11a`
 - `v00_13` Pandora in development as of today!

Now and future

- Currently we use something called SRT that allows one to checkout a corner of the svn repository, say, TrackFinder.
 - Edit, recompile, run with your new `libTrackFinderXYZ_module.so` and all the public libraries.
- The typical work consists of creating/editing to your purposes a producer or analyzer module, and running that in your fcl job, forcing your code in your library to be picked up.

repository

- After all's working for you, you check it into the repository
- Every night at 3am the HEAD of the repository is checked out into an official area and compiled, and now your code is in
 - `/grid/fermiapp/lbne/lar/code/larsoft/releases/development/lib/Linux/libTrackFinder.so`
 - where everyone makes use of it.
 - You're famous!

A job (scripted by a fcl script)

```
physics:
{

  producers:
  {
    opflash:      @local::microboone_opflashnew
    caldata:      @local::microboone_calwire
    ffhit:        @local::microboone_gaushitfinder
    dbcluster:    @local::microboone_fuzzycluster
    cccluster:    @local::microboone_clustercrawler
    trackkalmanhit: @local::microboone_track3Dkalmanhit
    spacepointfinder: @local::microboone_spacepoint_finder
    #spacepointcheater: @local::microboone_spacepointcheater
    trackkalsps:  @local::microboone_kalman_cosmic
    stitch:       @local::microboone_trackstitcher
    beziertracker: @local::microboone_beziertrackermodule
  }

  analyzers:
  {
    AnaTreeBezier: @local::microboone_analysistree
    AnaTreeKalmanSPS: @local::microboone_analysistree
    AnaTreeKalmanHit: @local::microboone_analysistree
  }

  #define the producer and filter modules for this path, order matters,
  #filters reject all following items. see lines starting physics.producers below
  reco: [ opflash, caldata, ffhit, dbcluster, cccluster, trackkalmanhit, spacepointfinder,
  trackkalsps, stitch, beziertracker ]
  ana: [ AnaTreeKalmanHit, AnaTreeKalmanSPS, AnaTreeBezier ]

  ...
}
```

Releases

- development or the HEAD of repository is the daily, bleeding edge
- Every now and then we “freeze/tag” a release
- Meaning the code as of a certain timestamp is labeled and dev is declared S2013.10.21, say.
- You set your environment variables to see libraries there. They’re “guaranteed” to never change. No nightly rebuild here. Bug fixes only merged in, and then, only rarely.
- The MCCs are run from frozen releases.
 - I did the last few tags/builds
 - Will do one more at Thxgvg. (My last, is my guess.)₁₁

Toward a stable build

- After End of June, 2013 SCD has run LArSoft
 - Rick Snider, Lynn Garren, mainly
 - Brian, I, still are doing some management
- Future Release Model
 - **MicroBooNE wanted assurance of independence from LBNE complications. Wanted removal of circular dependences.**
 - Daily, bleeding edge Dev builds still available
 - Integration Releases
 - Work off these, but beware they eventually are overwritten
 - on the many-week timescale
 - Will keep a short history of these
 - Code built off trunk, not necessarily latest revision, into each Integration Release
 - Branch and Merge
 - Code merged to some official branch

repository change

- 1 svn \rightarrow O(9) git repositories
 - to help make the experiment builds orthogonal
 - to make user's experience easier. Meaning...
 - if you want to work on TrackFinder only you would git clone ...LArReco repository only
 - all other code libs you need will come from the other 8 LArSoft code worth of libraries out of the 8 upses
- Certain modules, will go to experiment specific git repositories, including MicroBooNE's
- Librarians will be assigned to packages
 - to approve merges, vet code, dictate revision number from which to build Integration Releases, e.g.

build tool must change

- cmake/mrb hybrid coming to replace SRT
 - git fully pulls down whole repositories
 - Only this allows completely solving LArSoft's inconsistent builds problem, which is inherent in SRT.
 - parallel builds: `make -j4`
- mrb knows how to grab the experiment-specific repository code. Alpha testing going on now.
- Given git's branching, mrb will also need to ensure consistent branches in the build.
- Makefiles are still at the root of all building!
 - They are built by CMake running over CMakeList.txt files, which are hand-constructed, but less arcane than Makefiles.

Going forward

- The MicroBooNE Analysis Tools group is very enthusiastic generally, and gave its imprimatur to this entire proposal.
- We have a lot of git and CMake experience in the MicroBooNE DAQ.
- No one has mrb experience.

Cutover

- Alpha testing going on in Rick's team now.
- There will necessarily be a day on which the svn repository is locked out to check-ins and we cut over to git and mrb.
- We will begin testing in advance of this, during the "beta" period, and focus much Thursday effort on the imminent change.
- Will likely hold tutorials, once we have a grasp on our new tools.

External users

- One may happily function on uboonegpvm01,02,03.fnal.gov unless perhaps you're off the North American continent and/or want to use the Event Display.
- You may wish to make your own External Install of LArSoft: Nevis, Cambridge, KSU, Ben C, Wes K, have done this. The MicroBooNE DAQ will do this. Maintaining these is not entirely trivial. But, on SLF5(6) machines this is quite do-able.
 - What about dB access?
 - You must stay synched to the repository, at least with some regularity.

N (3–6?) months out

- **CVMFS (Cern Virtual Machine File System/Server)**
 - Game-changer
 - **coming After new LArSoft paradigm takes hold.**
 - NOvA institutions are doing this now
 - Access from offsite to a read-only mounted file system on the OSG on which LArSoft libs/bins, data are available
 - Eventually, bins/libs for OSX, ubuntu,
 - our effective Grid goes from $O(1000)$ FNAL nodes to 50–100k nodes.
 - Questions, offers to help to Mike Kirby
 - If it were me, I would no longer maintain my own LArSoft installation if this came to pass.

Links

- <https://cdcvcs.fnal.gov/redmine/projects/larsoftsvn/wiki>
- <https://cdcvcs.fnal.gov/redmine/projects/larsoft-alpha/wiki>
- http://www-microboone.fnal.gov/at_work/AnalysisTools/