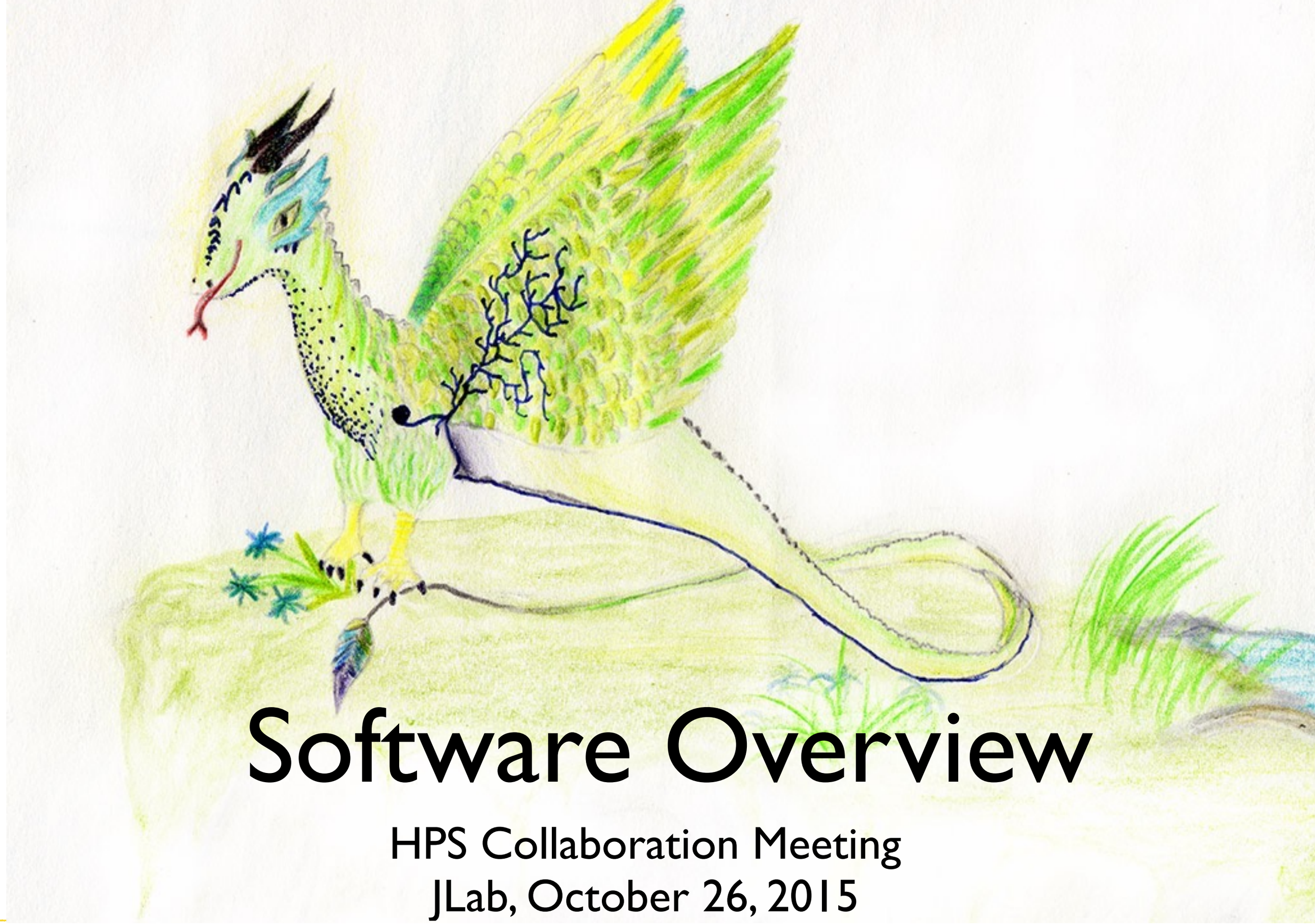

Software Overview

HPS Collaboration Meeting
JLab, October 26, 2015

University of New Hampshire

Maurik Holtrop

Status

Our software & software group are mostly working well.

- Mass production of reconstruction: Pass 3 done.
- DST updated, skims defined
- Advanced analysis started: see many later talks.
- Software bugs and questions reported on mailing list are usually resolved quickly.
- Only a few outstanding "critical" or "major" JIRA tasks.

Major Outstanding Tasks

- ECAL detector model that can be aligned properly
  - Multiple people working on this: Holly, Jeremy,…
- SVT Alignment & GBL
  - Several tasks related to this — Pelle, Allessandra
- Vertexing in varying B-Field
  - See Norman's presentation
- Time stamps — start time, event time
  - Issue around event time-stamp need to be revisited?
- Improve run database with DAQ info.
  - Should have better tracking of run info, start time etc. — See Jeremy's talk.

Software presentations

Software presentations this meeting:

- Update on New Tracking Strategies — Omar Moreno
- Update on ECal code and Calibrations — Holly Szumila-Vance
- SVT Alignment — Alessandra Filipi/Pelle Hansson
- Tracking and Vertexing in 3D B-Field — Norman Graf
- Run DB, Data Catalog, Conditions — Jeremy McCormick
- Organizing DQM — Matt Graham

+ Lots of analysis talks.

Focus shifting to analysis!

Speed & Memory

Current code is slow: 8 Hz (0.1228 sec/event)

Speed currently this is not a problem.

but at 100 days of 25 kHz data, it *will* be.

(7.5M CPUhours = 312,500 CPU-days, about a year on a 1000 core farm.)

Memory hog?

- Unconstrained: 1.6 GB active (6.7 GB virtual)
  - 800 MB is actual allocations, rest is Java?
  - Use depends on invocation and environment: Java obscures actual required amount of memory.
  - 26 threads
- Actual Minimum (-Xmx256m): 400MB real (3GB virtual)
  - So "EvioToLcio" is mostly OK with the standard driver.

Profiling Code

Code profiling is a bit of an "art", results can be influenced by sampling methods used.

Output is a web-browsable file, showing where the most time is spend in the code (see https://confluence.slac.stanford.edu/display/hpsg/Code+Documentation)

Tree: Call Tree

- [redacted] *calls: 0,* local time:.000 ms , total time: 1,248,251.705 ms , **100.00 %** -- *org.lcsim.util.Driver.doProcess*
  - [redacted] *calls: 0,* local time:6.092 ms , total time: 822,772.673 ms , **65.90 %** -- *org.hps.recon.tracking.TrackerReconDriver.process*
  - [redacted] *calls: 0,* local time:28.717 ms , total time: 288,693.455 ms , **23.10 %** -- *org.hps.recon.tracking.RawTrackerHitFitterDriver.process*
  - *calls: 0,* local time:13.053 ms , total time: 49,700.665 ms , **4.00 %** -- *org.hps.recon.ecal.EcalRawConverterDriver.process*
  - *calls: 0,* local time:.000 ms , total time: 30,794.082 ms , **2.50 %** -- *org.lcsim.util.loop.LCIODriver.process*
  - *calls: 0,* local time:47.428 ms , total time: 17,355.011 ms , **1.40 %** -- *org.hps.recon.tracking.TrackDataDriver.process*
  - *calls: 0,* local time:6.278 ms , total time: 12,775.682 ms , **1.00 %** -- *org.hps.recon.tracking.gbl.HpsGblRefitter.process*
  - *calls: 0,* local time:11.985 ms , total time: 9,850.229 ms , **.80 %** -- *org.hps.recon.tracking.DataTrackerHitDriver.process*
  - *calls: 0,* local time:11.529 ms , total time: 4,789.175 ms , **.40 %** -- *org.hps.recon.tracking.HelicalTrackHitDriver.process*
  - *calls: 0,* local time:22.818 ms , total time: 4,731.071 ms , **.40 %** -- *org.hps.recon.particle.HpsReconParticleDriver.process*
  - *calls: 0,* local time:25.256 ms , total time: 4,284.412 ms , **.30 %** -- *org.hps.recon.tracking.gbl.GBLOutputDriver.process*
  - *calls: 0,* local time:18.095 ms , total time: 1,097.107 ms , **.10 %** -- *org.hps.recon.ecal.cluster.ReconClusterDriver.process*
  - *calls: 0,* local time:74.776 ms , total time: 429.162 ms , **.00 %** -- *org.lcsim.recon.tracking.digitization.sisim.config.RawTrackerHitSensorSetup.process*
  - *calls: 0,* local time:75.647 ms , total time: 344.651 ms , **.00 %** -- *org.lcsim.recon.tracking.digitization.sisim.config.ReadoutCleanupDriver.process*
  - *calls: 0,* local time:.000 ms , total time: 307.923 ms , **.00 %** -- *org.hps.recon.ecal.EcalRunningPedestalDriver.process*
  - *calls: 0,* local time:5.522 ms , total time: 215.162 ms , **.00 %** -- *org.hps.recon.tracking.MergeTrackCollections.process*
  - *calls: 0,* local time:53.366 ms , total time: 92.681 ms , **.00 %** -- *org.hps.recon.ecal.cluster.CopyClusterCollectionDriver.process*
  - *calls: 0,* local time:18.564 ms , total time: 18.564 ms , **.00 %** -- *java.lang.System.nanoTime*

Profiling Code

Tree: Call Tree

- *calls: 0,* local time:.000 ms , total time: 1,248,251.705 ms , **100.00 %** -- *org.lcsim.util.Driver.doProcess*
  - *calls: 0,* local time:6.092 ms , total time: 822,772.673 ms , **65.90 %** -- *org.hps.recon.tracking.TrackerReconDriver.process*
    - *calls: 0,* local time:.000 ms , total time: 822,761.330 ms , **65.90 %** -- *org.lcsim.util.Driver.process*
      - *calls: 0,* local time:.000 ms , total time: 822,761.330 ms , **65.90 %** -- *org.lcsim.util.Driver.processChildren*
        - *calls: 0,* local time:.000 ms , total time: 822,761.330 ms , **65.90 %** -- *org.lcsim.util.Driver.doProcess*
          - *calls: 0,* local time:36.326 ms , total time: 822,761.330 ms , **65.90 %** -- *org.lcsim.recon.tracking.seedtracker.SeedTracker.process*
            - *calls: 0,* local time:51.774 ms , total time: 798,430.923 ms , **64.00 %** -- *org.lcsim.recon.tracking.seedtracker.SeedTrackFinder.FindTracks*
              - *calls: 0,* local time:10.821 ms , total time: 597,882.168 ms , **47.90 %** -- *org.lcsim.recon.tracking.seedtracker.ConfirmerExtender.Extend*
                - *calls: 0,* local time:93.067 ms , total time: 597,799.347 ms , **47.90 %** -- *org.lcsim.recon.tracking.seedtracker.ConfirmerExtender.doTask*
                  - *calls: 0,* local time:25.443 ms , total time: 385,338.618 ms , **30.90 %** -- *org.lcsim.recon.tracking.seedtracker.HelixFitter.FitCandidate*
                    - *calls: 0,* local time:.000 ms , total time: 363,597.020 ms , **29.10 %** -- *org.hps.recon.tracking.MultipleScattering.FindScatters*
                    - *calls: 0,* local time:69.280 ms , total time: 18,803.096 ms , **1.50 %** -- *org.lcsim.fit.helicaltrack.HelicalTrackFitter.fit*
                    - *calls: 0,* local time:5.710 ms , total time: 2,894.336 ms , **.20 %** -- *org.lcsim.recon.tracking.seedtracker.SeedCandidate.setScatterAngles*
                    - *calls: 0,* local time:12.288 ms , total time: 12.288 ms , **.00 %** -- *org.lcsim.recon.tracking.seedtracker.HelixFitter.CorrectStereoHits*
                    - *calls: 0,* local time:6.435 ms , total time: 6.435 ms , **.00 %** -- *org.lcsim.fit.helicaltrack.HelicalTrackFitter.getFit*
                  - *calls: 0,* local time:11.878 ms , total time: 150,255.819 ms , **12.00 %** -- *org.lcsim.recon.tracking.seedtracker.FastCheck.CheckHitSeed*
                  - *calls: 0,* local time:33.355 ms , total time: 29,222.273 ms , **2.30 %** -- *org.lcsim.recon.tracking.seedtracker.FastCheck.CheckSector*
                  - *calls: 0,* local time:.000 ms , total time: 29,212.201 ms , **2.30 %** -- *org.lcsim.recon.tracking.seedtracker.SeedCandidate.addHit*
                  - *calls: 0,* local time:256.860 ms , total time: 2,223.570 ms , **.20 %** -- *org.lcsim.recon.tracking.seedtracker.SeedCandidate.<init>*
                  - *calls: 0,* local time:80.211 ms , total time: 486.845 ms , **.00 %** -- *org.lcsim.recon.tracking.seedtracker.MergeSeedLists.Merge*
                  - *calls: 0,* local time:238.153 ms , total time: 244.001 ms , **.00 %** -- *java.util.Collections.sort*
                  - *calls: 0,* local time:228.938 ms , total time: 228.938 ms , **.00 %** -- *java.util.LinkedList.addFirst*

Code speedup possible:

- Raw data fitters are slow, using Minuit
- FindScatters - Matt already looked at improving this, but not in trunk.
- Optimize tracking strategies
- … many of the speed optimizations will be hard …

Continuity

- Many of our previous crew of developers (i.e. the students) are now moving on.
- We need to start identifying who will take over specific areas of responsibility.
  - ECAL codes (readout, analysis, calibration, …)
  - SVT/tracking code (readout, tracker, GBL, analysis, calibration, …)
  - DST
- We need to make sure sufficient **documentation** is in place!

Next Run Preparations.

- There have been no new feature requests for online software (?)
- We need to do a "shake down" of all online codes to make sure everything still works. (not guaranteed with all the changes)
  - Test all modes of monitor app, test with latest DAQ setup.
  - Test all calibrations codes.
  - Make sure of smooth operations when run starts.
- Organize offline data processing and Data Quality Monitoring (DQM) — See Matt's talk.

Analysis Codes

- We now have many different people making their own analysis codes.
  - Different approaches: Java, C++/LCIO, C++/Root.
  - Different event selection cuts.
  - Different histograms used for comparisons.
- We have a confusing number of different track collections, particle collections…
- At some point, we will need some standardization of event selection etc. This started with skims.
- Make sure your analysis code is well documented!
- Make sure your analysis code is in SVN or GitHub.

Conclusions

- HPS Software is in fairly good shape.
  - All main components are in place
  - Need an online shake-down before the next run.
- Emphasis now shifts to analysis code.
- We need to document what our code does, on confluence and in the code itself.
- Online code shake-down.