2015 Engineering Run Pass0

Norman Graf
SLAC
May 14, 2015

Background

- In January, 2015, a change was made to SIOCluster.java in org.lcsim that changed how Calorimeter Clusters were being written out.
- Native Java and C++ classes in LCIO unchanged.
- int nId (=N)
  - id0
  - id1
  - …
  - idN-1
- became
- int Id (defaulted to 0)

Issue

- As long as no particle Id was being run, this bug had no effect, since the default particle Id was 0.
- Native LCIO code (Java and C++) simply read that there were no Ids, org.lcsim (hps-java) read that Id was 0.
- So although this bug was introduced in January and we have run this code this winter and spring we were able to read the output lcio files using the C++ LCIO API.

Problem

- Problem arose when we started to assign particle Ids to the clusters.
- Now, when particle Id was 22, the LCIO API expected to read 22 particle Ids for this cluster.
  - FAIL.
- Note, however, that the data itself is OK
- org.lcsim/hps-java readers were fine.

Patch

- I created a branch in the LCIO package to allow people to access the existing pass0 output via the native java and C++ APIs.
  - svn co svn://svn.freehep.org/lcio/branches/hps-2015-branch
- SIOCluster.java in org.lcsim has been changed to only write out 0.
  - All files written going forward will be readable by both hps-java and native LCIO.

Paths to analysis

- Analysis within hps-java was never compromised.
- Using the LCIO branch, Omar has built the DST maker and verified that it can read the pass0 files.
  - We can make DSTs from the existing pass0 files.
- Individuals can also check out the branch for use in their own analyses.
- Data is readable.
  - People still need to analyze the data and make sure it is OK!

Path Forward

- Will resume reconstruction with the patched hps-java.
  - Files from now on will be readable.
- Using LCIO branch, can create DSTs from the existing pass0 files.
  - Should probably fix the LCIO file at the same time.
- Don’t think we need to re-reconstruct the already processed data.

Proposal

- Start processing the new data, especially the 0.5mm data, as soon as possible with the latest haps-java.
  - Should produce the DSTs at the same time
  - It saves us writing the data to tape, then restaging to produce the DSTs
  - Provides immediate feedback on final data quality.
- Using LCIO branch, produce DSTs from the existing pass0 data, also fix the SIOCluster in the LCIO files
- Be prepared for any confusion!