HPS Data Summary Tapes

Omar Moreno

Santa Cruz Institute for Particle Physics
University of California, Santa Cruz
omoreno1@ucsc.edu

June 5, 2013

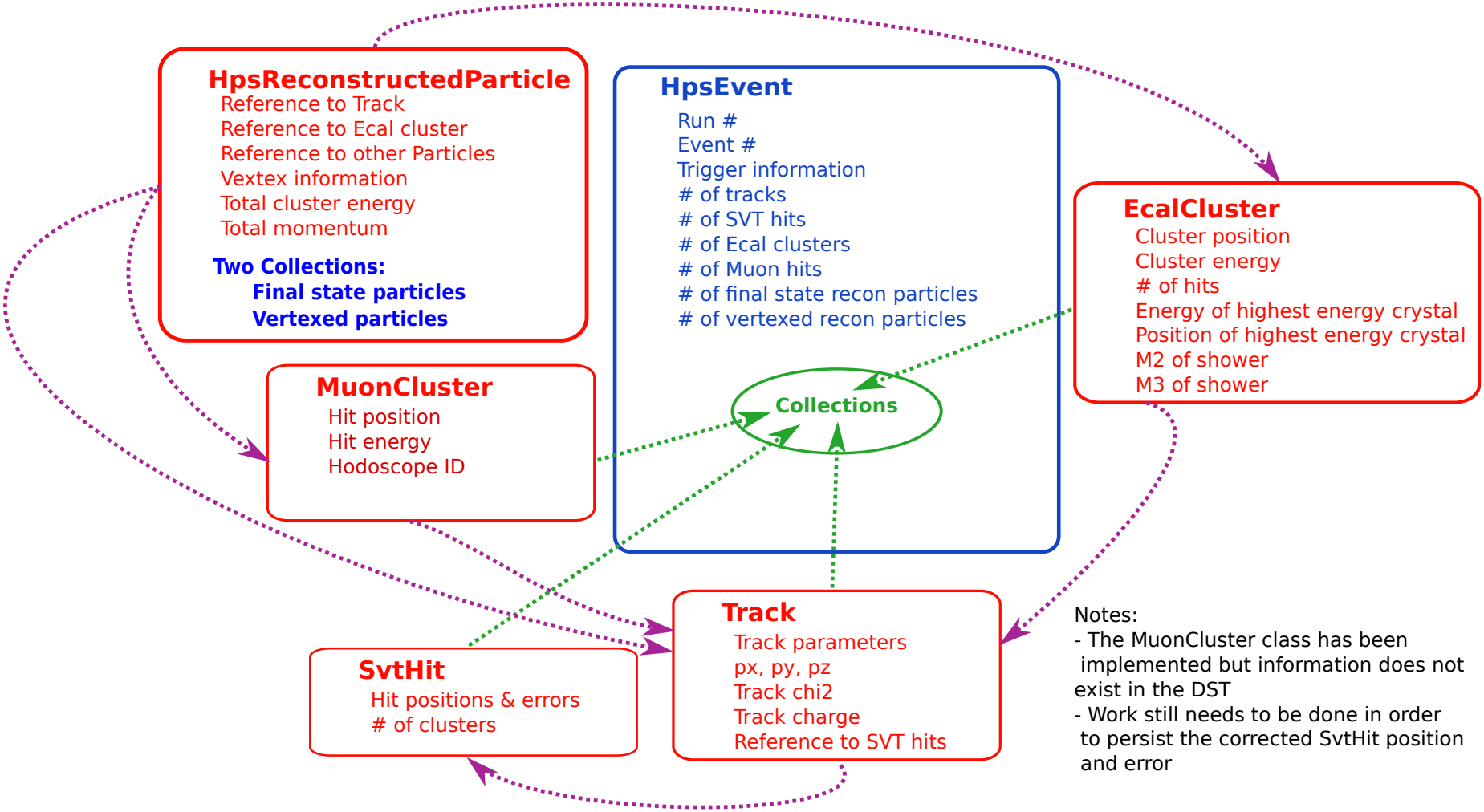
Why Do We Want DSTs?

- Currently, analysis of data is being done using the Java based `lcsim` framework in conjunction with the data analysis package AIDA
- It would be preferable to analyze data using ROOT in order to take advantage of its plotting capabilities and analysis tools
  - Members of the HPS collaboration are also more familiar with ROOT and C++ so it would allow more collaborators to get involved with analysis
- The issue then becomes converting from LCIO (output from `lcsim`) to ROOT $\Rightarrow$ The LCIO C++ API allows for this to be done easily
- The resulting DST would then be accessible using a lightweight API and would only contain physics objects
- For those users wanting access to lower level information, the LCIO C++ allows access to the LCIO files directly in conjunction with ROOT but at this point it's probably best to just use `lcsim`

HPS Event Structure

- The ROOT based DST is composed of HpsEvent objects which are used to encapsulate collections (`TClonesArray`) of the following objects
  - EcalCluster
  - SvtTrack
  - SvtHit ⇒ A class used to describe a 3D hit (stereo hit)
  - MuonCluster
  - HpsReconstructedParticle ⇒ A class used to describe a particle i.e. a track associated with a cluster
    - An HpsReconstructedParticle can either be a single particle or be a composite of several particles ⇒ vertexing information is provided for particles composed of daughther particles
    - Two collections: Final state particles, and particles which have been vertexed
    - Additional event information is also contained within HpsEvent
- Corrected SVT hit information and muon cluster data is still not available

Contents of an HPS Event



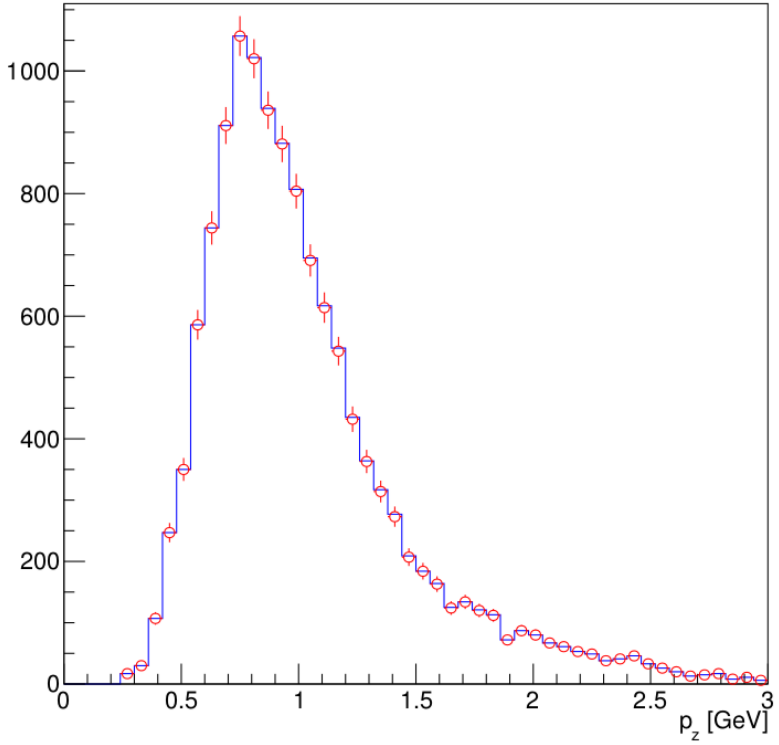
Notes:
- The MuonCluster class has been implemented but information does not exist in the DST
- Work still needs to be done in order to persist the corrected SvtHit position and error

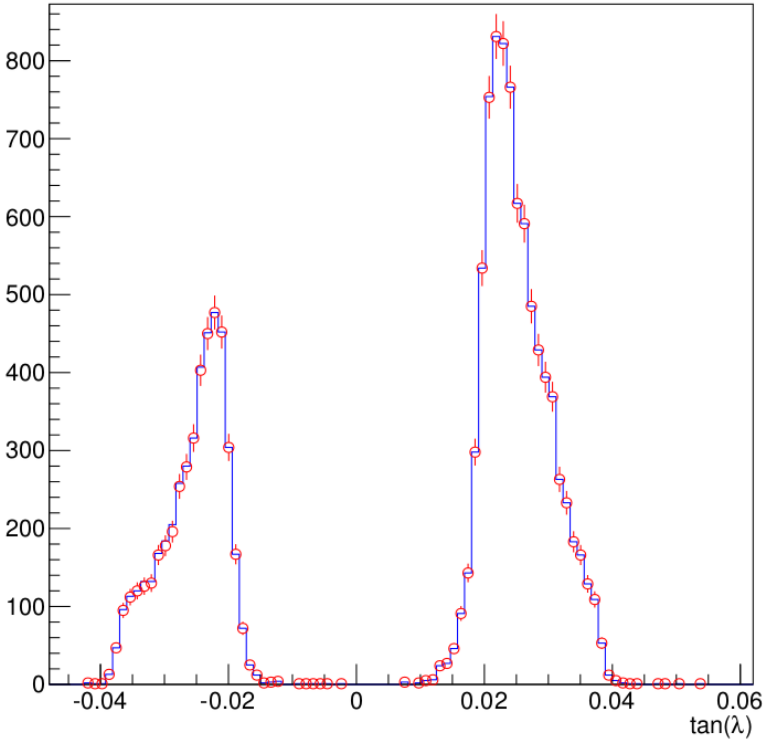
Performance

- Brief performance studies were performed using data gathered during the Test Run
  - Writin a DST with the data described above from a reconstructed LCIO containing is taking $\sim$ 20 seconds per 50,000 LCIO events
  - The resulting ROOT file is $\sim$ 8 Mb per 50,000 LCIO events
- The resulting size of the LCIO file was reduced from a size of $\sim$ 1 Gb per 100,000 events to $\sim$ 100 Mb per 100,000 events by reducing some collections
  - Most of the size was coming from raw tracker hits and associated collections

Data Integrity

- In order to make sure that data is being written to the DST correctly, histograms of parameters generated from reading the LCIO file and DST file are compared
  - Use Kolmogorov Test to compare the histograms
- Eventually, will use test suite to run these test on all DST variables





Getting Started

- The DST writer along with the HPS Event API is available through github

```
https://github.com/omar-moreno/hps dst
```

- Build instructions and a tutorial can be found here

```
https://confluence.slac.stanford.edu/display/hpsg/Reading+and+Writing+HPS+Data+Summary+Tapes
```

- Reconstructed LCIO files can be found here:
  `http://www.slac.stanford.edu/ omoreno/dst/`
- DSTs can be found here:
  `http://www.slac.stanford.edu/ omoreno/dst/recon`
- The source code above also contains a directory (examples/) which contains three sample analyses
  - TwoTrackAnalysis example.C/.py make use of the DST
  - TwoTrackAnalysis example.cxx reads from an LCIO directly

Need people to start writing using the DSTs