Software updates: Data reduction

February 4th 2025

Matthew Gignac

U.S. DEPARTMENT OF ENERGY

Office of Science

SLAC NATIONAL ACCELERATOR LABORATORY

Introduction

- Some software updates that are worth sharing, relating to data reduction and processing
  - Several changes needed to remove hit collections. These changes have reduced the LCIO size by an additional 80%, above reductions already implemented for the 1% processing
    - Our 1% dataset would reduced from 27 TB to 5.4 TB
      - We still will need further reductions by using event filtering, as proposed by Matt a few weeks back

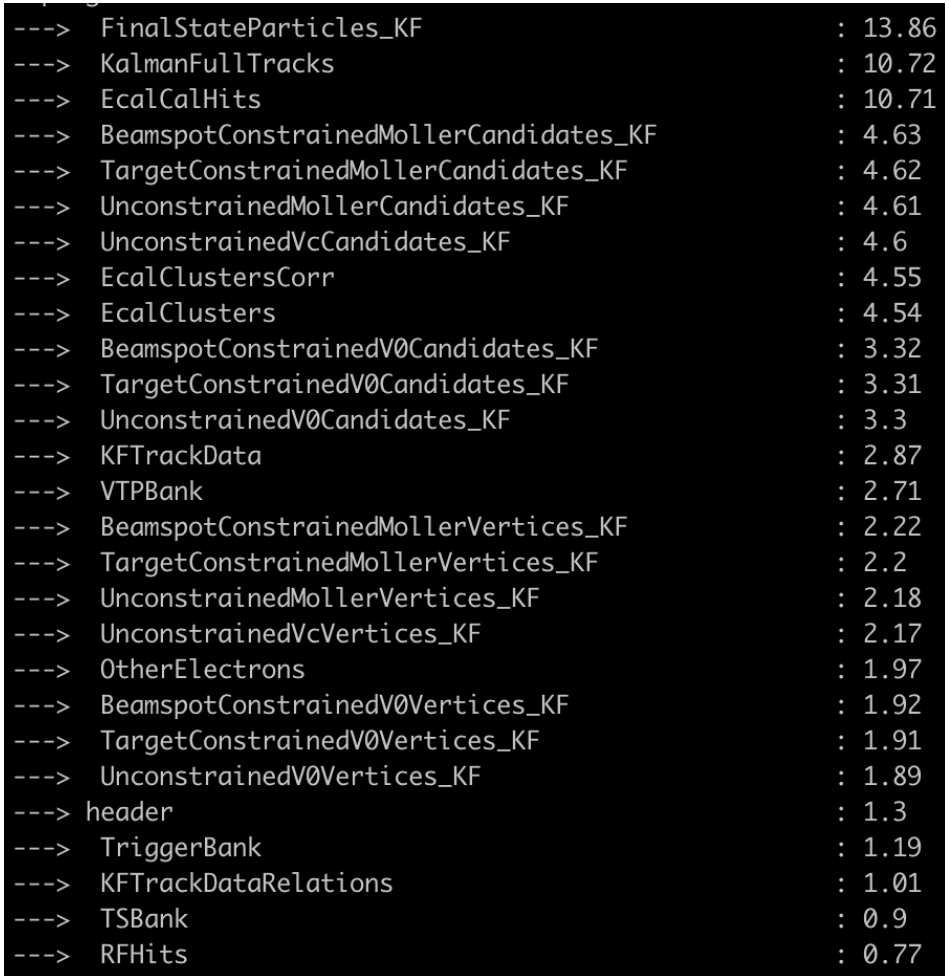
Software changes: data reduction

- Removed hit collections
  - Filled so-called "*subdetectorHitNumbers*" lcsim track property (array of ints indicating which layer is hit):
    - https://github.com/JeffersonLab/hps-java/pull/1077
  - Updated HPSTR to use "*subdetectorHitNumbers*" for hit layer and number of hits in ROOT n-tuples.
    - Note this doesn't contain functionality for shared hits (yet)
    - This also fixes the conflicting argparser for truthHits
    - https://github.com/JeffersonLab/hpstr/pull/203
  - Updated the recon steering file for "pass 1", with most hit collections removed:
    - https://github.com/JeffersonLab/hps-java/pull/1078

Some comments

- If you are updating your code, you should update both hps-java and hpstr — changes in both codes bases!
- A new `*recoTuple_noHitColl_cfg*` should be used if you want to use LCIO files without hit collections
  - The default `*recoTuple_cfg*` is <u>unchanged</u> and configured to use hit collections
- The *subdetectorHitNumbers* was previously unfilled, so if you are using old SLCIO files and processing them through the updated *recoTuple_noHitColl_cfg*, you will be missing hit information in your n-tuples
- Simple validation performed, but please report any issues you encounter!

Remaining collections



```
--->  FinalStateParticles_KF                          : 13.86
--->  KalmanFullTracks                                : 10.72
--->  EcalCalHits                                     : 10.71
--->  BeamspotConstrainedMollerCandidates_KF          : 4.63
--->  TargetConstrainedMollerCandidates_KF            : 4.62
--->  UnconstrainedMollerCandidates_KF                : 4.61
--->  UnconstrainedVcCandidates_KF                    : 4.6
--->  EcalClustersCorr                                : 4.55
--->  EcalClusters                                    : 4.54
--->  BeamspotConstrainedV0Candidates_KF              : 3.32
--->  TargetConstrainedV0Candidates_KF                : 3.31
--->  UnconstrainedV0Candidates_KF                    : 3.3
--->  KFTrackData                                     : 2.87
--->  VTPBank                                         : 2.71
--->  BeamspotConstrainedMollerVertices_KF            : 2.22
--->  TargetConstrainedMollerVertices_KF              : 2.2
--->  UnconstrainedMollerVertices_KF                  : 2.18
--->  UnconstrainedVcVertices_KF                      : 2.17
--->  OtherElectrons                                  : 1.97
--->  BeamspotConstrainedV0Vertices_KF                : 1.92
--->  TargetConstrainedV0Vertices_KF                  : 1.91
--->  UnconstrainedV0Vertices_KF                      : 1.89
---> header                                           : 1.3
--->  TriggerBank                                     : 1.19
--->  KFTrackDataRelations                            : 1.01
--->  TSBank                                          : 0.9
--->  RFHits                                          : 0.77
```

Possible to reduce further?

- Maurik requested “EcalCalHits”
- Are other Ecal cluster collections needed?
  - EcalClustersCorr
  - EcalClusters
- Other collections?
  - VTPBank?
  - OtherElectrons?
  - RFHits?
- Difference between collections with “Candidates” vs “Vertices”?
  - These are both vertex fits?

Questions