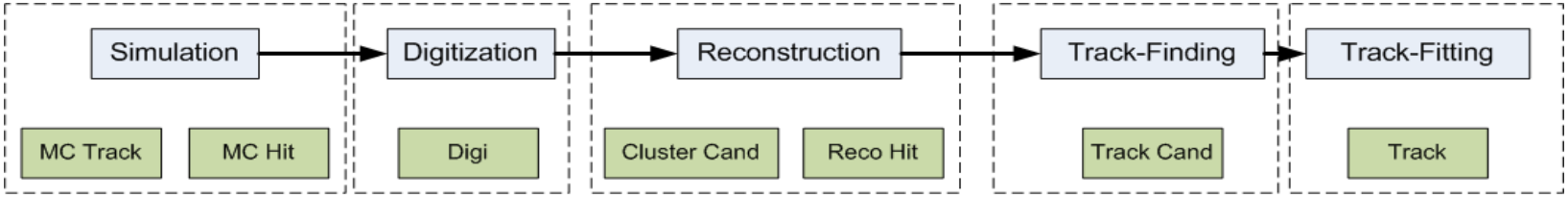
Distribution of MC Information

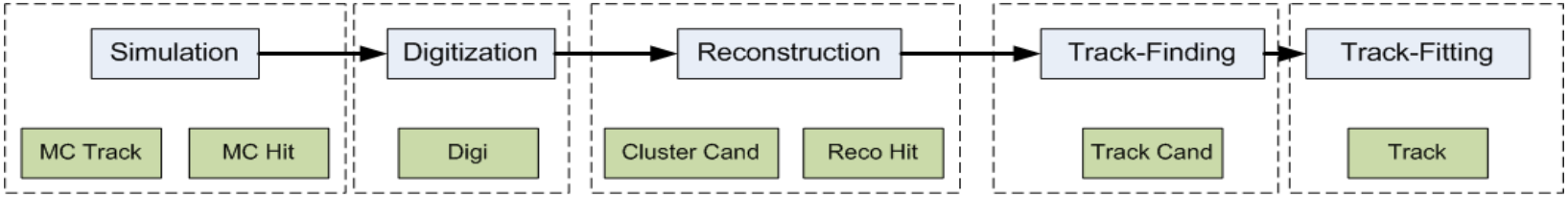
PANDA Computings Workshop - SUT

Juli 3, 2017 | Tobias Stockmanns

Motivation

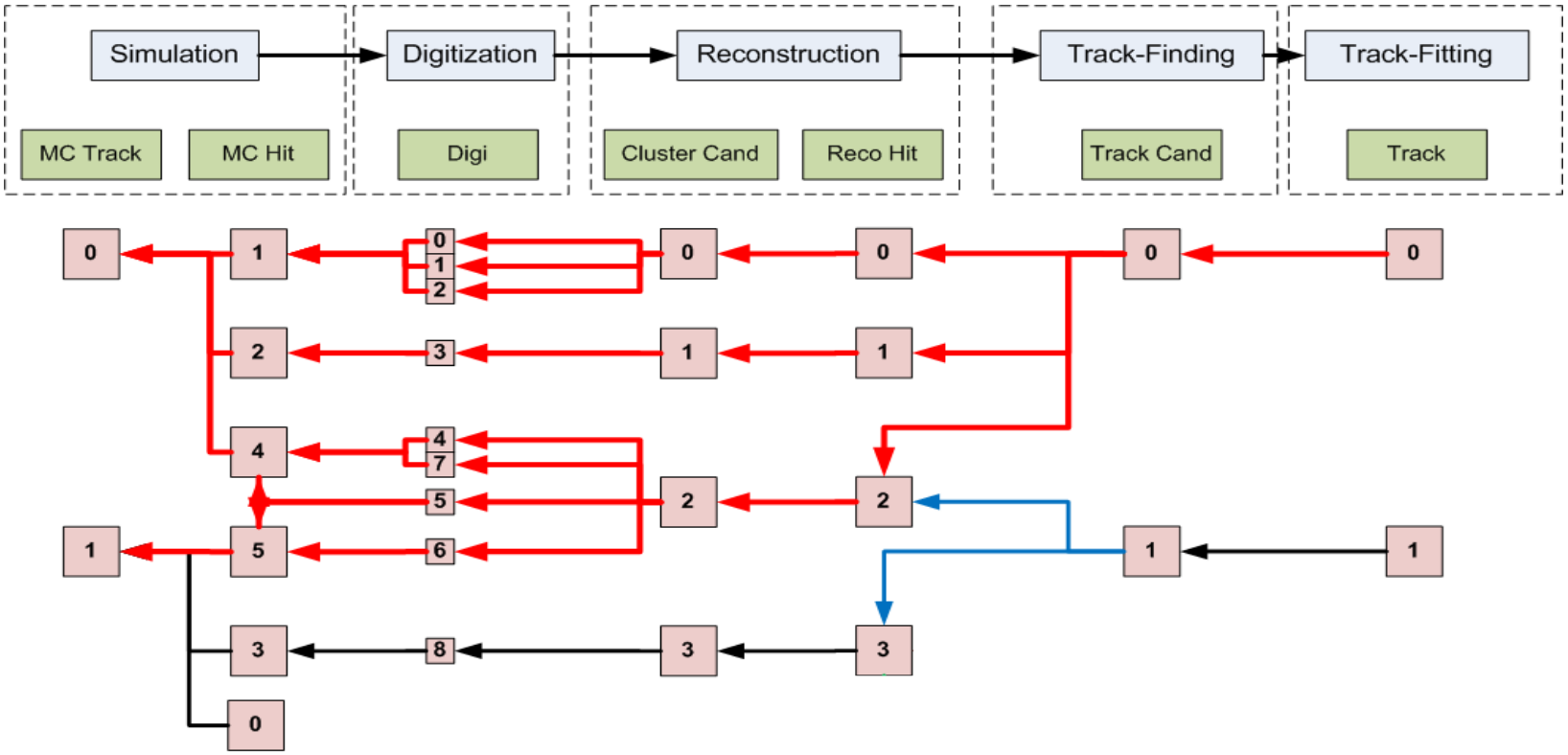

Simulation
Digitization
Reconstruction
Track-Finding
Track-Fitting
MC Track
MC Hit
Digi
Cluster Cand
Reco Hit
Track Cand
Track

Motivation



- How to transport MC information through simulation stages?
- What type of questions to answer?
  - What is the MC Track of a Reco Track?
  - What is the correct position of a Reco Hit?

Motivation

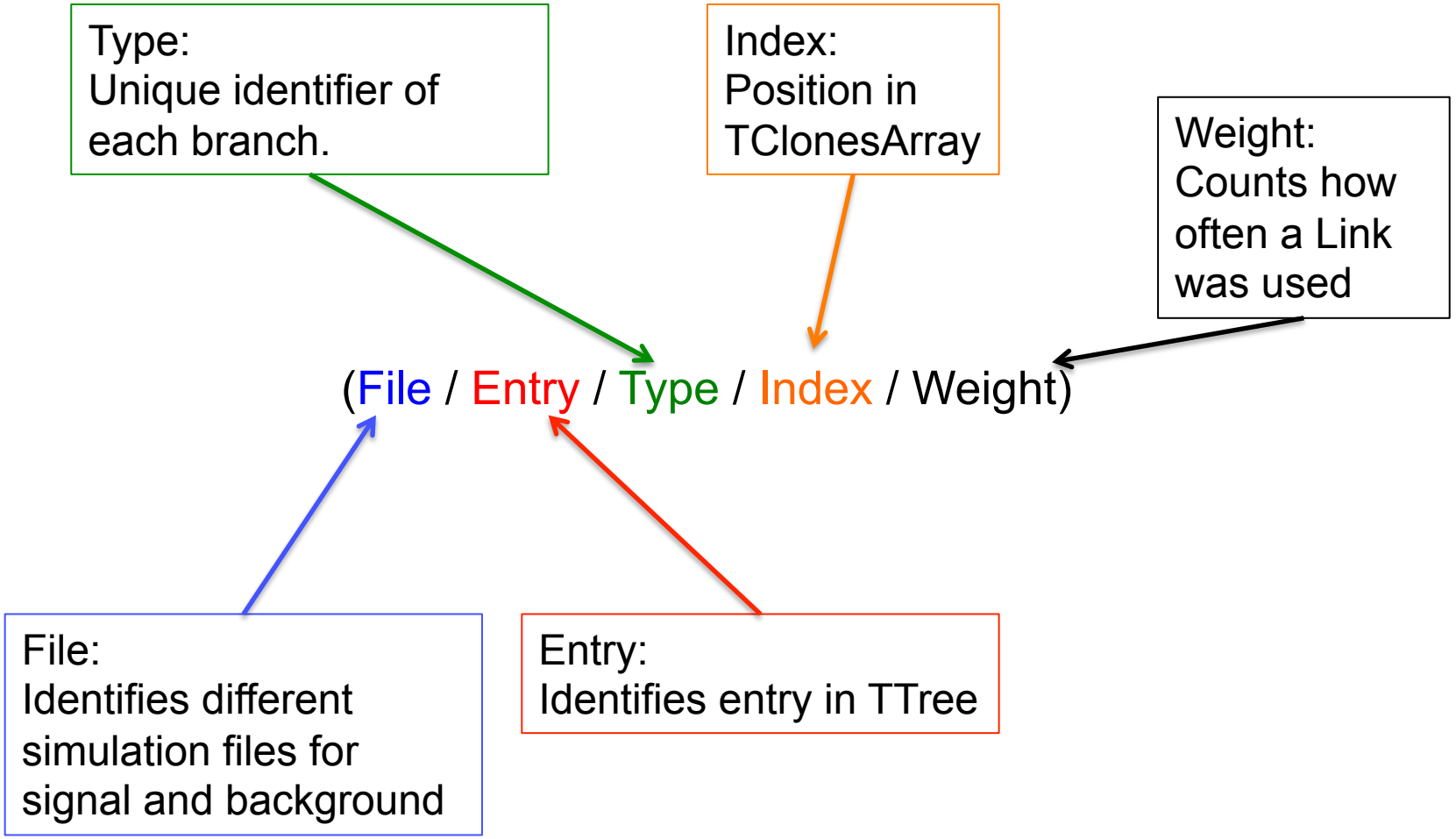

Simulation
Digitization
Reconstruction
Track-Finding
Track-Fitting
MC Track
MC Hit
Digi
Cluster Cand
Reco Hit
Track Cand
Track

Data treatment

- Each simulation stage stores its outcome in a root file
- Inside the root file the data is organized in a TTree of TBranches
- Each TBranch contains one type of information object
- Each entry in a TBranch consist of a TClonesArray which holds the data
- In event-based simulation each entry inside the tree contains all data of one event
- This is not true for time-based simulation

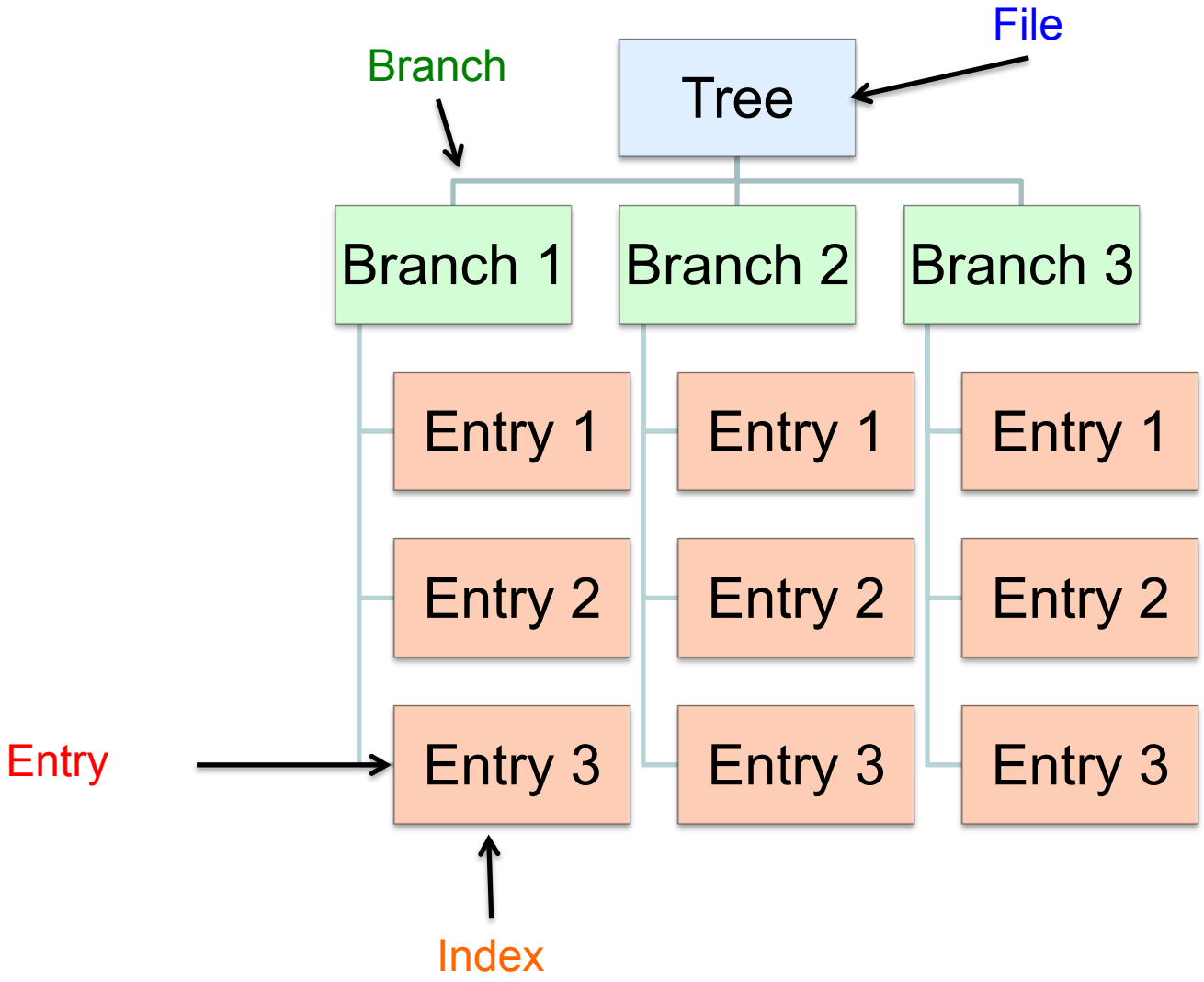
- How to connect data over branch boundaries and file boundaries?

FairLink



FairLink is a unique identifier for each data object stored in a tree

FairLink


File
Branch
Tree
Branch 1
Branch 2
Branch 3
Entry 1
Entry 1
Entry 1
Entry 2
Entry 2
Entry 2
Entry
Entry 3
Entry 3
Entry 3
Index

FairMultiLinkedData

Two important Methods:

- **SetLink(FairLink link))**: Clears the existing list of links and sets link as first entry
- **AddLink(FairLink link, bool multi):** Adds link to the vector of links. If multi is false it checks first if this link already exists and increases the weight factor for this link. In all other cases the link is added to the vector.

If you want to use MC Propagation you have to:

1. derive your classes from **FairMultiLinkedData_Interface** or **FairHit/FairMCPoint**
2. **set/add** the links to the data you have used to generate your actual data set

Example SttHit

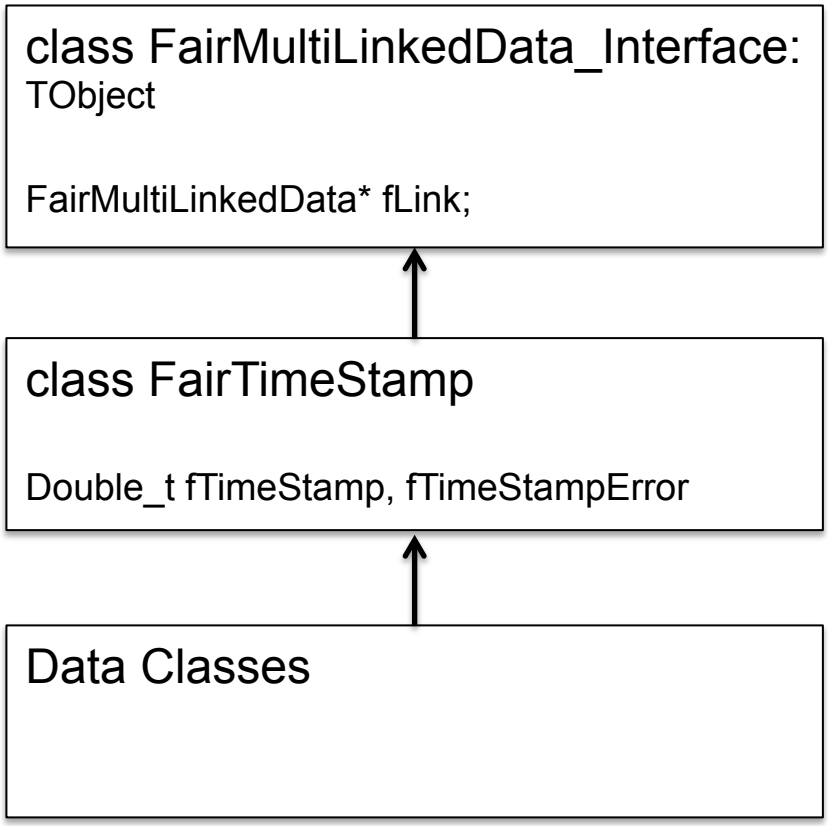
```
/** Standard constructor **/
PndSttHit::PndSttHit (Int_t detID, TVector3& pos, TVector3& dpos,
                      Int_t index, Int_t flag, Double_t isochrone,
                      Double_t isochroneError, TVector3 wireDir)
  : FairHit(detID, pos, dpos, index)
{
  fIsochrone = isochrone;
  fIsochroneError = isochroneError;
  fRadial = TMath::Sqrt(pos.X() * pos.X() + pos.Y() * pos.Y());
  fWireDirection = wireDir;
  fAssigned = kFALSE;
  // stt1
  fXint = fX;
  fYint = fY;
  fZint = fZ;
  SetLink(FairLink("STTPoint", index)); //short version
  SetLink(FairLink(-1, FairRootManager::Instance()->GerEntryNr(),
   "SttPoint", index));
}
```

Example PndTrack – Full Information

[(-1/9/MCTrack/2/154)
(-1/9/STTPoint/55/3) (-1/9/STTPoint/56/3)
(-1/9/MVDPoint/5/15) (-1/9/MVDPoint/6/20) (-1/9/MVDPoint/7/20)
(-1/9/STTHit/55/2) (-1/9/STTHit/56/2)
(-1/9/MVDPixelDigi/6/4) (-1/9/MVDPixelDigi/7/4) (-1/9/MVDPixelDigi/8/4)
(-1/9/MVDStripDigi/2/4) (-1/9/MVDStripDigi/3/4) (-1/9/MVDStirpDigi/4/4)
(-1/9/MVDStripDigi/5/4) (-1/9/MVDStripDigi/6/4) (-1/9/MVDStripDigi/7/4)
(-1/9/MVDStripDigi/8/4) (-1/9/MVDStripDigi/9/4)
(-1/9/MVDPixelCluster/4/3)
(-1/9/MVDPixelHit/6/2)
(-1/9/MVDStripCluster/0/3) (-1/9/MVDStripCluster/1/3)
(-1/9/MVDStripCluster/2/3) (-1/9/MVDStripCluster/3/3)
(-1/9/MVDStripHit/1/2) (-1/9/MVDStripHit/2/2)]

* Type number replaced by branch name – Reduced Number of STTHits

FairLinks in Data Classes



class FairMultiLinkedData:
TObject

set<FairLink> fLinks;
FairLink fEntryNr;

FairLinkManager

- Often not the complete history data is wanted
- Mostly MCTrack, sometimes MCPoint
- `FairLinkManager` controls what is stored as a `FairLink`
- `FairLinkManager` is an instanton created in `FairRun`
- Access via `FairLinkManager::Instance()`
- Two ways how to control what is stored:
  - `AddIncludeType(Int_t type);`
    - *This branch type is stored*
  - `AddIgnoreType(Int_t type);`
    - *This branch type is not stored*
  - Cannot be mixed!
- Example in macro

  `FairLinkManager::Instance()->AddIncludeType(0);`

  only stores MCTracks

What to do with the FairLinks?

- You can ask each object with FairLinks where it was coming from:
  - `vector<FairLink> GetSortedMCTracks();`
    - *returns all MCTracks sorted by their weight*
  - `FairMultiLinkedData GetLinksWithType(FairRootManager::Instance()->GetBranchId("MyType"));`
    - *returns all FairLinks with the given type*

- You can even get the object the FairLink is pointing to:
  - `TObject* FairRootManager::Instance()->GetCloneOfLinkData(FairLink);`
    - *You have to cast it to its original data type*
    - *You have to destroy it at the end (it is a clone!)*

- Have a look at `class PndMCTruthMatch`

What to do with FairLinks?

- Ideal track finder `PndIdealTrackFinder` based on FairLinks

- Tracking quality assurance based on FairLinks

Remarks

- FairLinks can be switched on and off
- Without FairLinks the data classes only contain an empty pointer
- Level of detail for FairLinks can be set via `FairLinkManager`, everything from only MC Tracks to complete history possible
- `FairRootManager` can return a clone of a data object for a `FairLink`
- `FairRootManager` adds history data (optional)

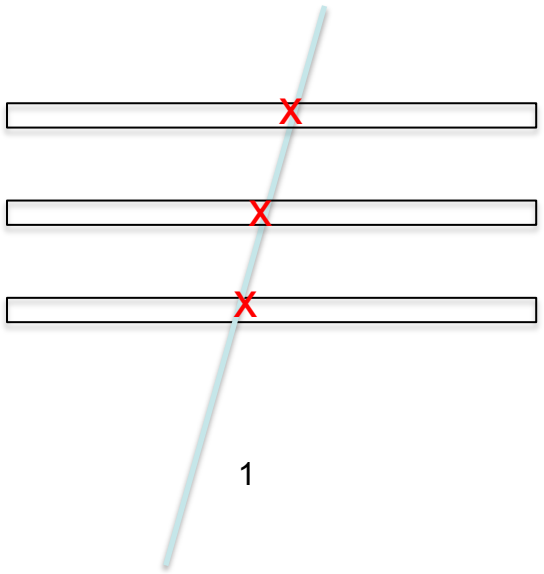
- Size increase strongly depends on settings:

No FairLinks	Full FairLinks	Only MCTrack
3,132,578 byte	3,926,688 byte	3,188,179 byte
	+ 25 %	+ 2 %

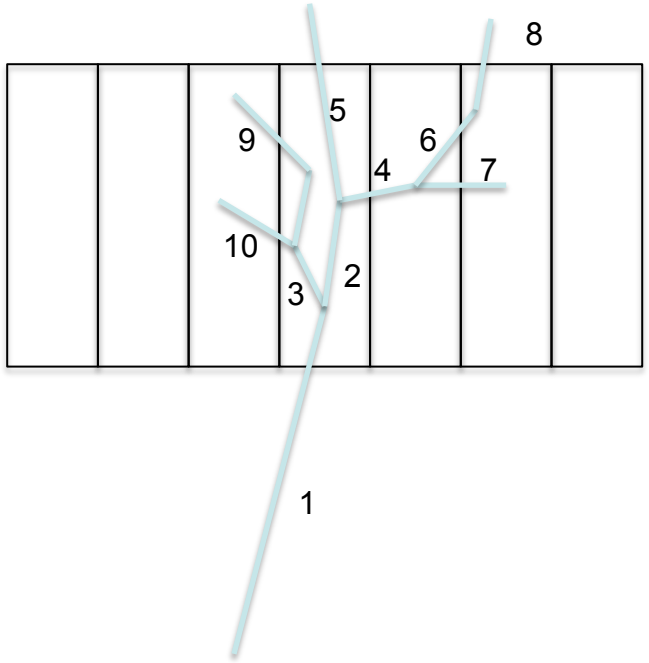
Reconstructed Tracks for 1000 events DPM

EMC – a special case

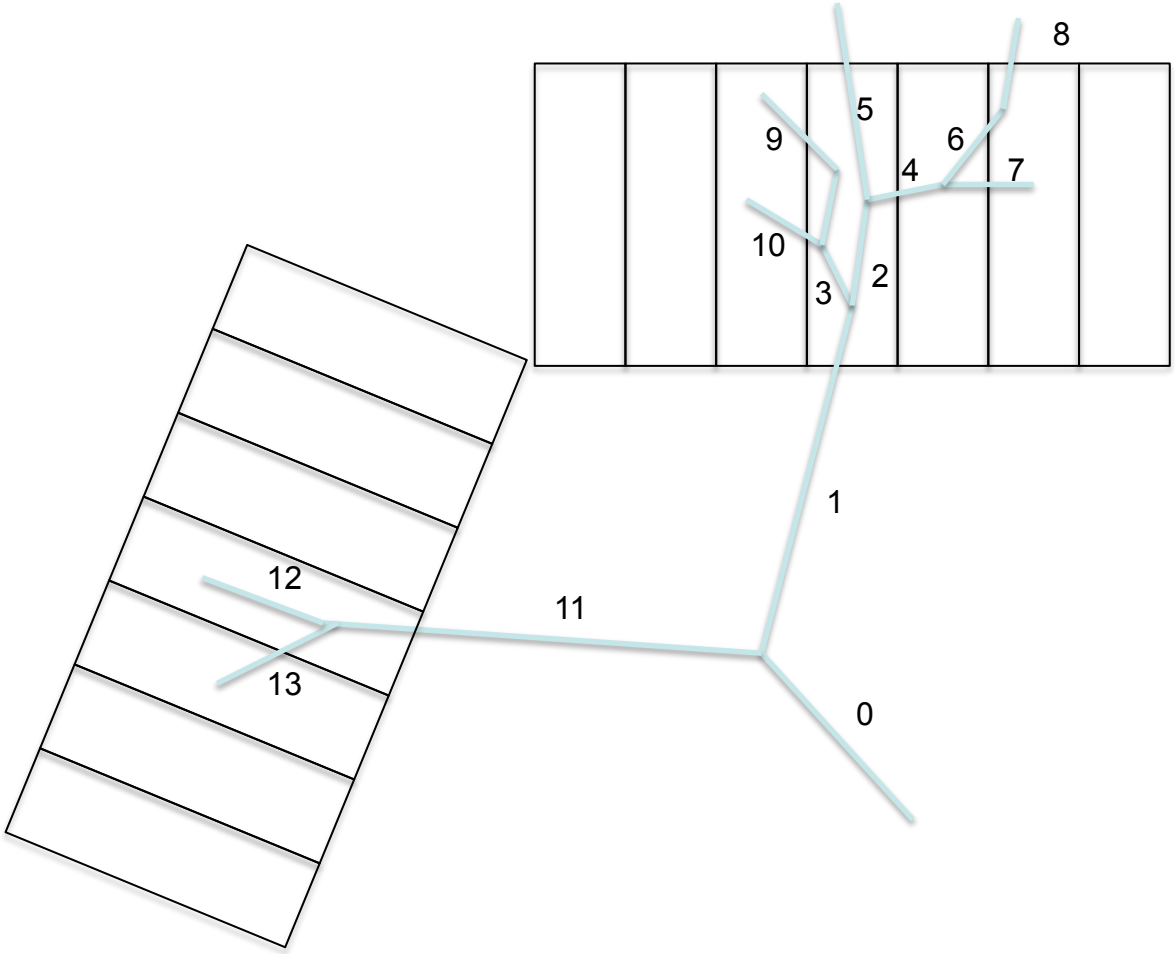
Tracking Detectors



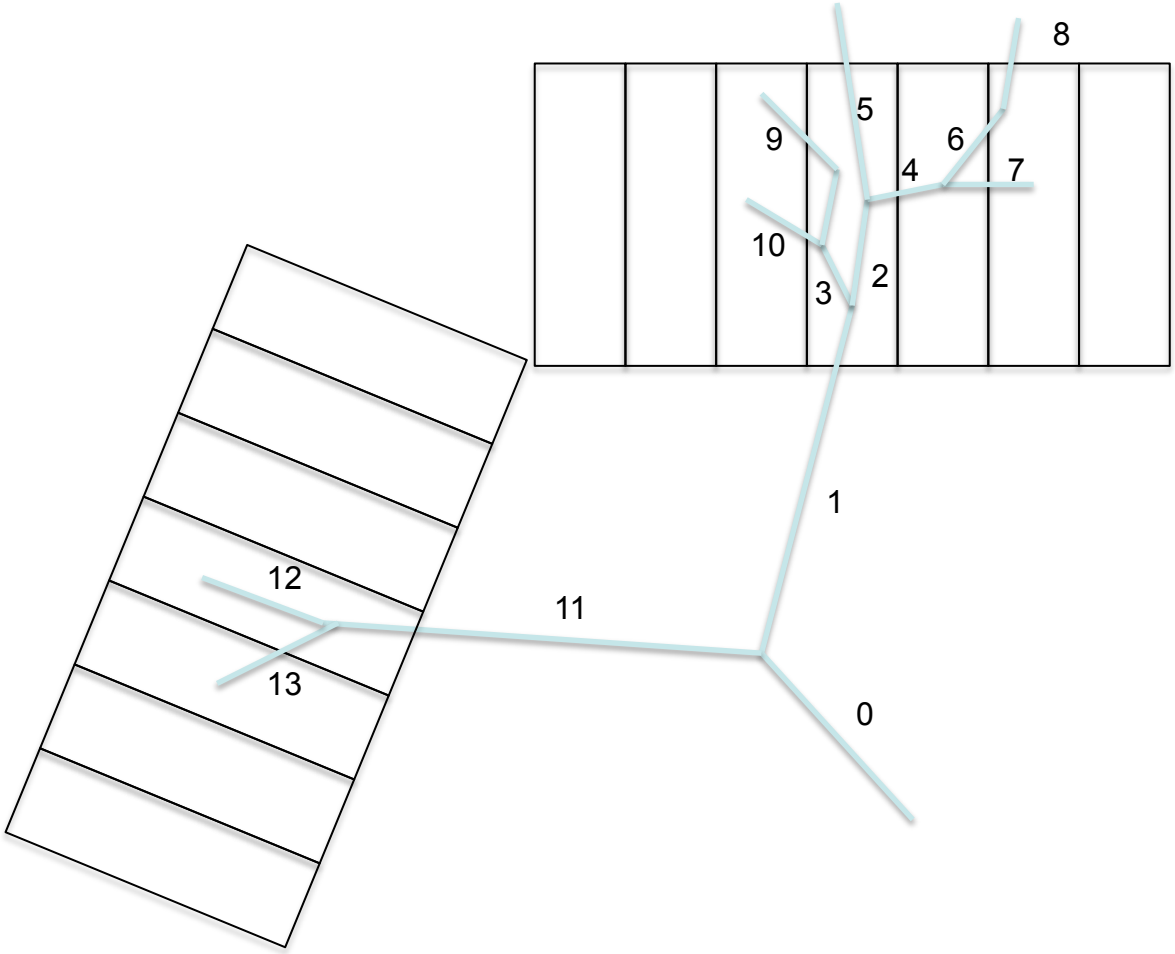
EMC



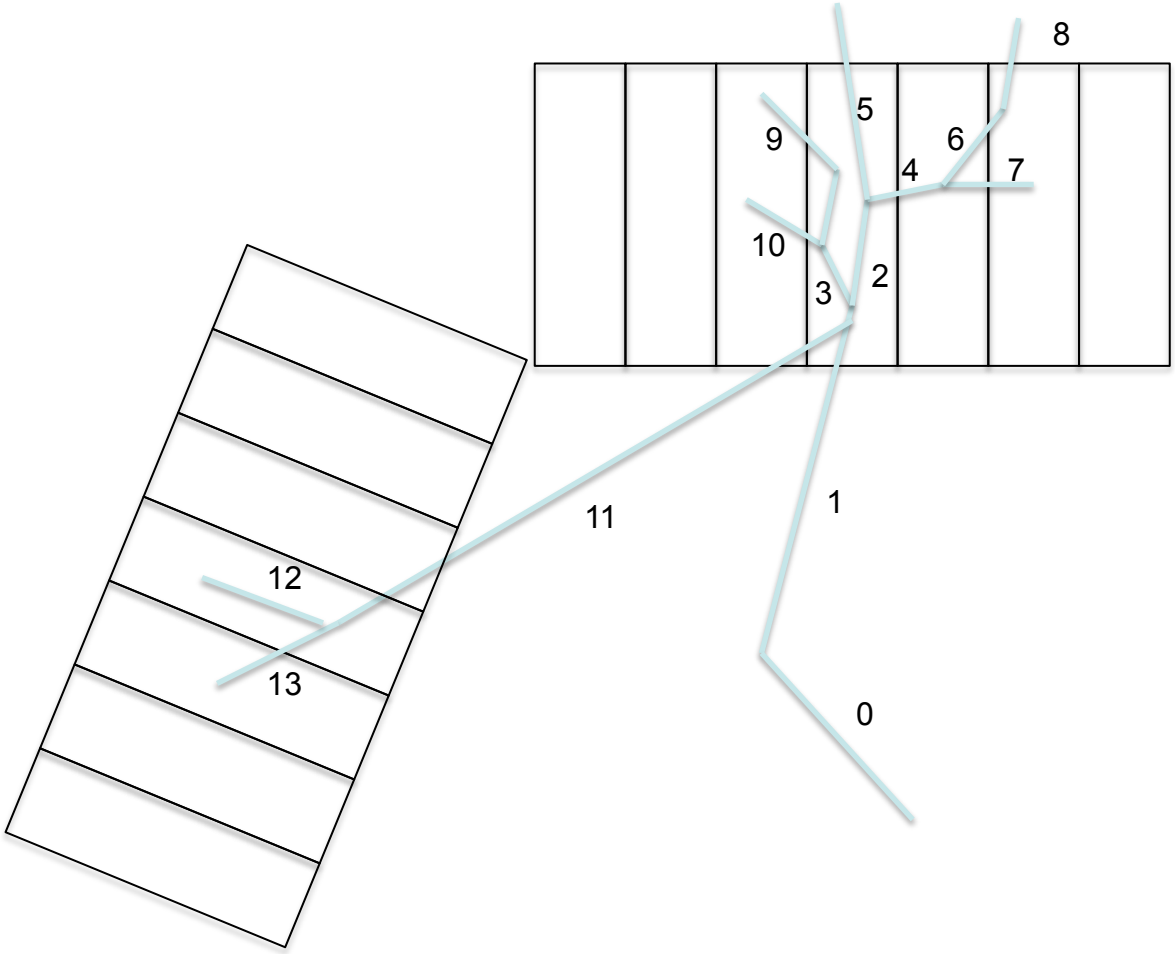
1st Implementation – Go back to primary



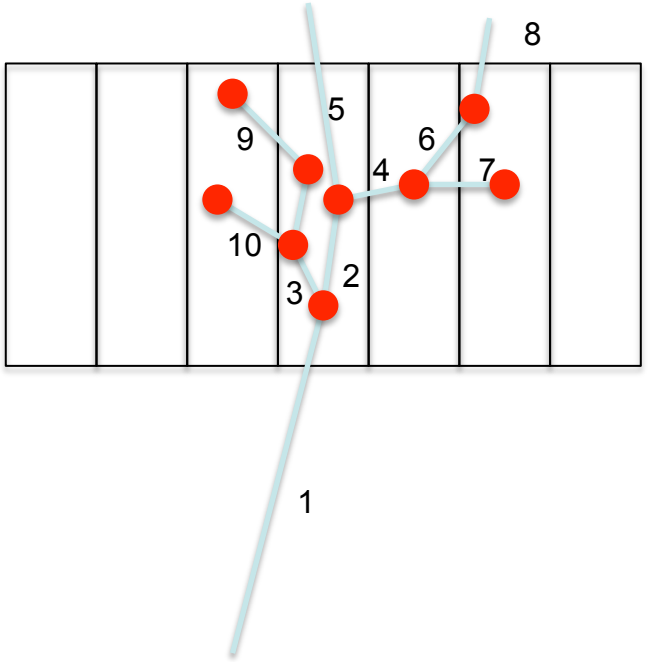
2nd Implementation – Go Back to first outside EMC



2nd Implementation – Go Back to first outside EMC

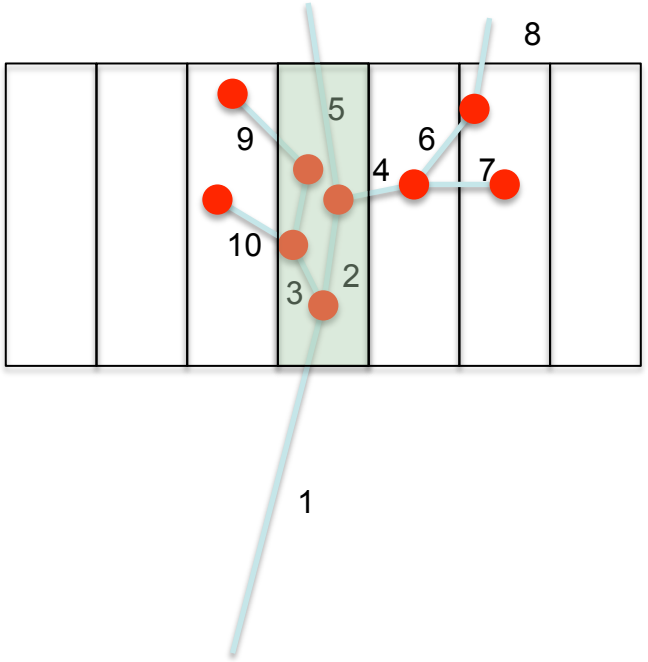


New Implementation – EMC Nomenclature



Emc Point

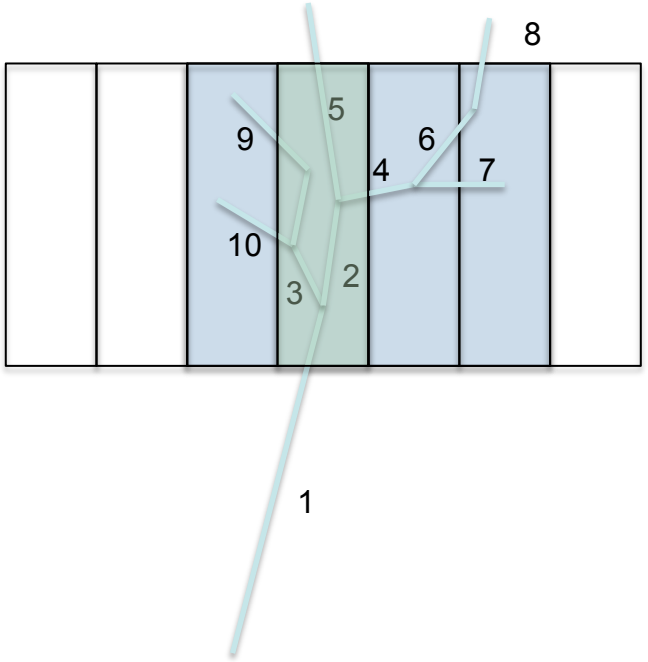
New Implementation – EMC Nomenclature



Emc Point

Emc Hit –
All points in one crystal

New Implementation – EMC Nomenclature

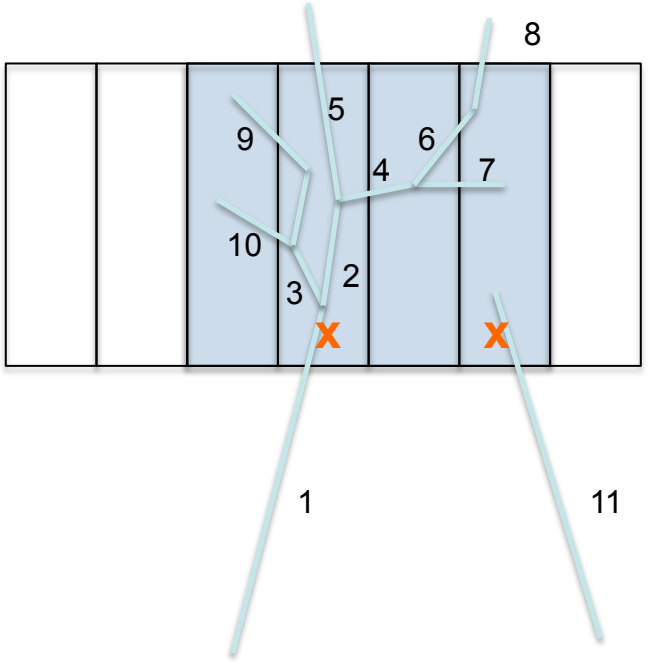


Emc Point

Emc Hit –
All points in one crystal

Emc Cluster –
All hits in neighboring crystals

New Implementation – EMC Nomenclature



Emc Point

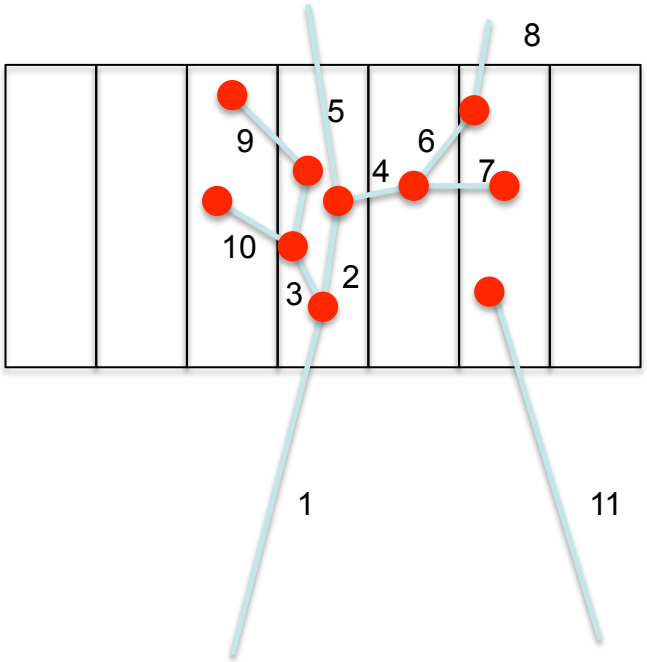
Emc Hit –
All points in one crystal

Emc Cluster –
All hits in neighboring crystals

X Emc Bump –
Subdivision of Cluster for multiple particles

New Implementation

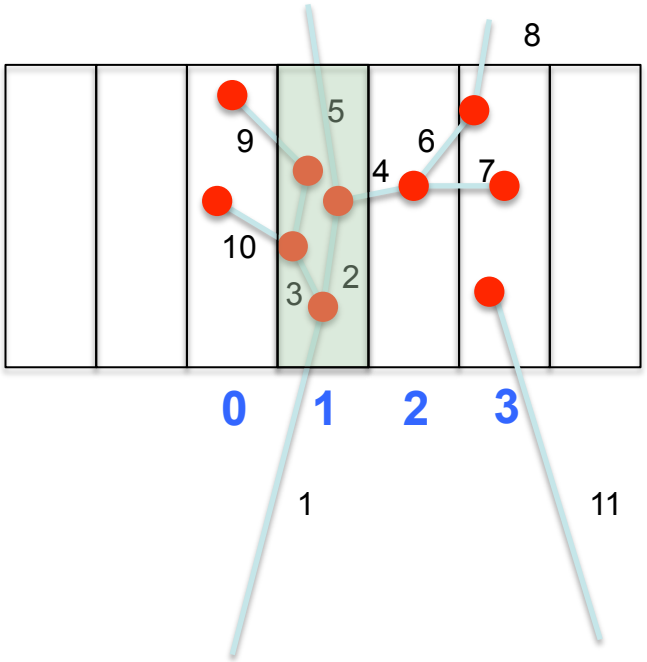
- For each EmcPoint if track is entering / exiting the crystal is stored



EmcPoint: entering: 1, 11, 4, 6, 7, 9 ,10
exiting: 4, 5, 6, 7, 8, 9, 10

New Implementation

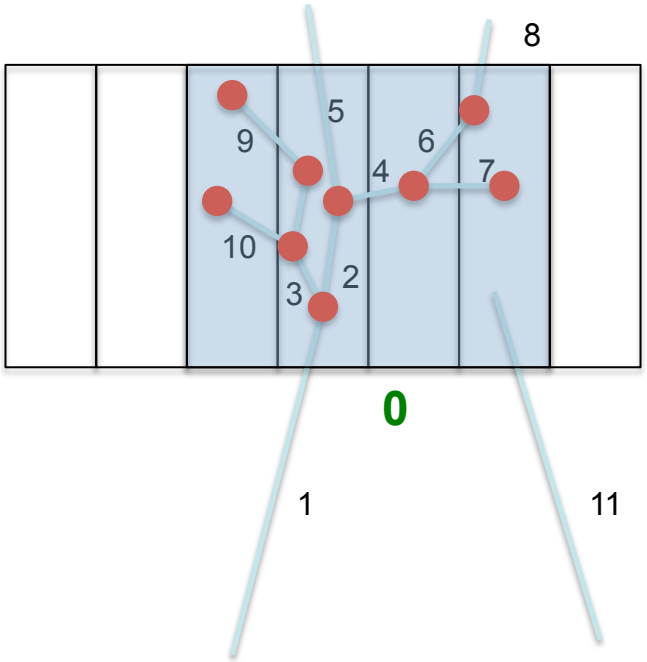
- For each EmcPoint if track is entering / exiting the crystal is stored
- Each EmcHit stores tracks entering and exiting



EmcHit **0**: in: 9 ,10 out -
1: in: 1 out 4, 5, 9, 10
2: in: 4 out 6, 7
3: in 6, 7, 11 out 8

New Implementation

- For each EmcPoint if track is entering / exiting the crystal is stored
- Each EmcHit stores tracks entering and exiting
- Each EmcCluster stores tracks entering and exiting

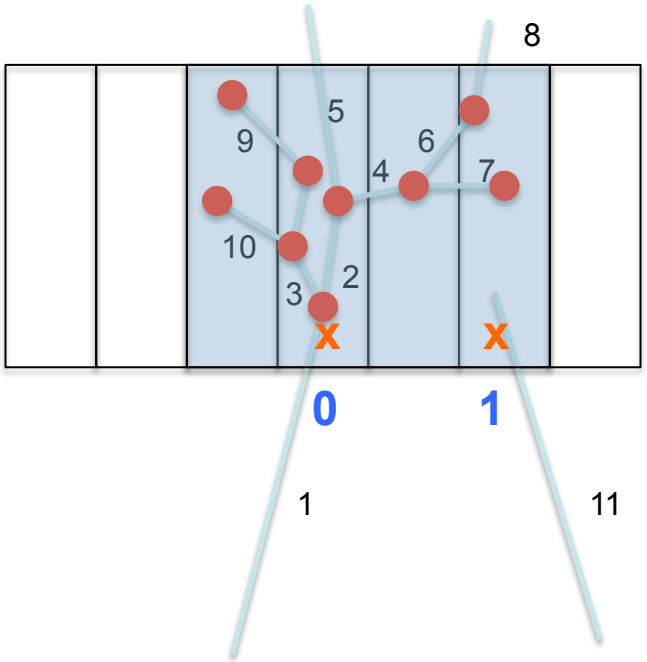


EmcHit **0**: in: ~~9~~ ,~~10~~ out -
1: in: 1 out ~~4~~, 5, ~~9~~, ~~10~~
2: in: ~~4~~ out ~~6~~, ~~7~~
3: in ~~6~~, ~~7~~, 11 out 8

EmcCluster **0**: in 1,11 out 5, 8

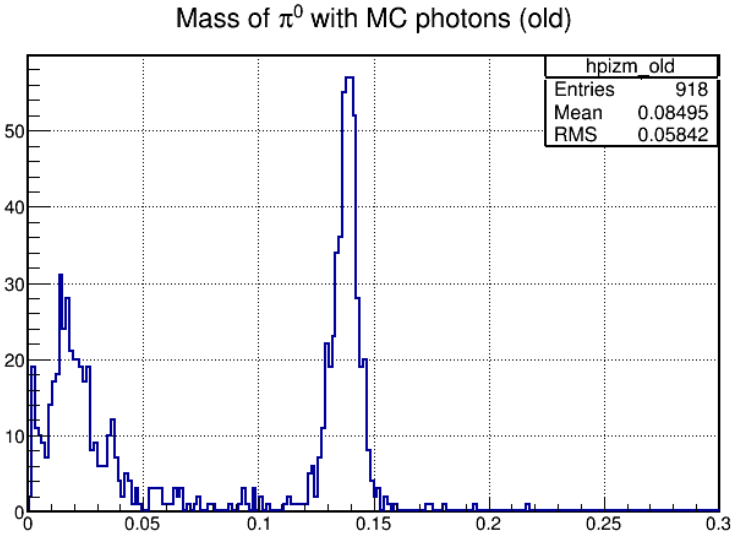
New Implementation

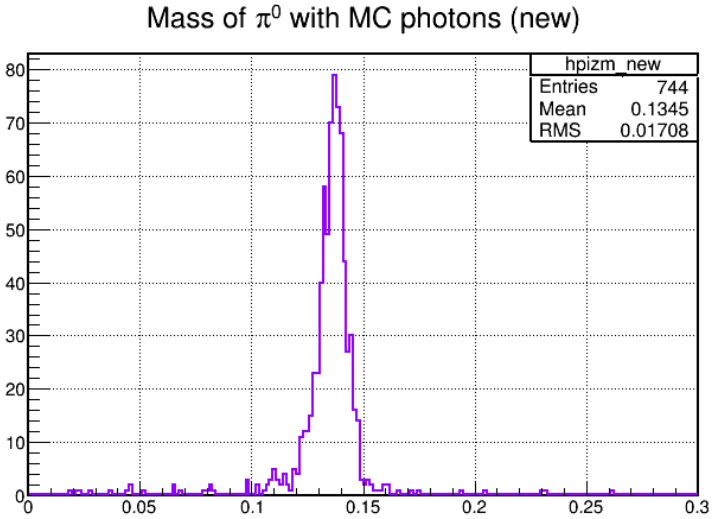
- For each EmcPoint if track is entering / exiting the crystal is stored
- Each EmcHit stores tracks entering and exiting
- Each EmcCluster stores tracks entering and exiting
- Each EmcBump stores nearest track entering (not implemented yet)



EmcBump **0**: in: 1
1: in 11

Result





Example from Lu: $\bar{p}p \rightarrow D_s^- D_s^+ \rightarrow K^- K^+ \pi^- \pi^+ \pi^- \pi^0 \nu_e e$

π^0 reconstructed with MC matched photons

Summary

- Monte Carlo truth propagation is a non-trivial task
- FairLinks: unique pointer to all data objects inside FairRoot
- MC information stored as collection of FairLinks pointing to (up to all) older data objects used to generate the current data object

- What to store can be handled by FairLinkManager

- EMC data needs special treatment