Calls of EmcCluster::where() and EmcDigi::where()
(intercepted during 8.8.1a Bear execution on single muon MC)

1) EmcMake2dCluster → EmcCluster::sortDigis – calls EmcCluster::where() through theta() and phi()
   (arranges digis in increasing distance from cluster center)

2) EmcTrackMatch → EmcBGFilterTrkMatchMethod::matchEmcToTracks – calls EmcCluster::where()
   (cluster – track matching, change of performance?)

3) RecoEventSelector → EmcXClMoments::EmcXClMoments
   // HepPoint cl( myCluster().where() );
   HepPoint cl( myCluster().gravWhere(&myCluster()) );
   // Hep3Vector diff = digi->where() - myCluster().where();
   Hep3Vector diff = digi->where() - cl;
   (if we don’t want to change this module we need to keep digi->where() to report the front center)
Calls of EmcCluster::where() and EmcDigi::where()
(continued)

4) RecoEventSelector → AbsRecoCalo::fourMomentum – calls EmcCluster::where()
   (direction of the 3-momentum will be changed, hopefully – improved)

5) EmcTrkClMatch → EmcGeomTrkClMatchMethod::matchEmcToTracks – calls EmcCluster::where()
   (again cluster-trk matching, don’t know why)

6) EmcTrackMatch → EmcGeomTrkMatchMethod::matchEmcToTracks – calls EmcCluster::where()
   (bump – trk matching, performance shown on last pages)
Calls of EmcCluster::where() and EmcDigi::where()
(continued)

7) EmcIdentify → EmcClusterMoments::secondMomentTP
   double clusTheta = myCluster().theta();
   double clusPhi = myCluster().phi();
   EmcDigi* current;
   double dt = current->thetaReal() - clusTheta;
   double dp = EmcCluster::findPhiDiff(current->phiReal(),clusPhi);
   (currently thetaReal() and phiReal() return the front face center angles, while
   clusTheta and clusPhi will be the angles of the depth reconstructed cluster
   center)
   Change to:
   double dt = current->thetaReal(Eclus) - clusTheta;
   double dp = EmcCluster::findPhiDiff(current->phiReal(Eclus) ,clusPhi);
   (one justification for implementing the method at digi level rather than
   cluster level)
Calls of EmcCluster::where() and EmcDigi::where()
(continued)

8) EmcPidHist → EmcCluster::where()

9) LoadRecoBtaCandidates → EmcCluster::where()

10) BtaSelectCandBase → EmcCluster::where()

11) EmcOprQAHist calls EmcCluster::where(), EmcCluster::theta(),phi() and EmcDigi::thetaReal(),phiReal()

(don’t know what these do and how they will be affected)
E-dependent method - implementation at Digi level in 8.8.1a based test release

1) New EmcDigi member functions:
   const HepPoint EmcDigi::where(double Eclus) const
double EmcDigi::thetaReal(double Eclus) const // returns where(Eclus).theta()
double EmcDigi::phiReal(double Eclus) const // returns where(Eclus).phi()

2) New member function in EmcClusterLogPos
   HepPoint EmcClusterLogPos::privateDepthWhere(const EmcCluster* me)
   (utilizes EmcDigi->where(Eclus) to calculate the centroid)

3) In EmcCluster
   New private data members:
   - HepPoint (*_algorithm) (const EmcCluster*); //replaces algPoineter()
   - CentroidMethod _whereType; // will need this in EmcClusterMoments

   Enum CentroidMethod {gravity,logarithmic,neighbor,logdepth,Default};

   New member function:
   - int EmcCluster::whereType() const // returns _whereType

V. Koptchev Umass - Amherst
E-dependent method implementation at Digi level in 8.8.1a based test release
(continued)

4) In EmcClusterMoments do either
   
   double dt = current->thetaReal() - clusTheta;
   double dp = EmcCluster::findPhiDiff(current->phiReal(),clusPhi);
   
or
   
   double dt = current->thetaReal(Eclus) - clusTheta;
   double dp = EmcCluster::findPhiDiff(current->phiReal(Eclus),clusPhi);
   
   depending on which method was used to reconstruct the cluster center

5) For the depth method – the 2.6mrad theta correction should not be applied
   not done yet because of problems compiling EmcCalib
**Bump - Track matching**

Each curve shows the method performance for different significance level requirements.

* Default trk-match method – surface
O Default trk-match method – depth

In development:
- x Doca trk-match method – depth
- ▼ Estimated centr. trk-match method – depth

- Default method does worse when given a point at a depth
- Estimated centroid Method:
  - not very good for low mom. tracks (expl. - next page)
  - provides very clean sample, but at low efficiency
    (min. ionizing pions are matched to their bumps very well)
Estimated Centroid Trk match method

• Description
  - get all Emc intersections of the track
  - make estimated cluster consisting of all intersected crystals
  - to each crystal ascribe energy proportional to the track path length in it
  - reconstruct the estimated cluster’s centroid using E-dep method assuming Eclus = E of the bump the track is being matched to
  - significance level of the match – based on 3D distance between estimated centroid and the bump centroid

• Performance
  - for muons and minimum ionizing pions – estimated cluster centroid is very close to the bump’s centroid, except for low momentum tracks

• Low momentum tracks problem
  - energy loss in the calorimeter constitutes a large fraction of the initial energy, increase in curvature should be taken into account when getting track intersections
**Pi0 invariant mass**

Single Pi0 0-5GeV, \( z = -1 \) to 0 cm

- Loop over BtaCandidates in CalorNeutral list
- Undo 2.6 mrad theta correction for Depth method
- Similar resolution

V. Koptchev Umass - Amherst