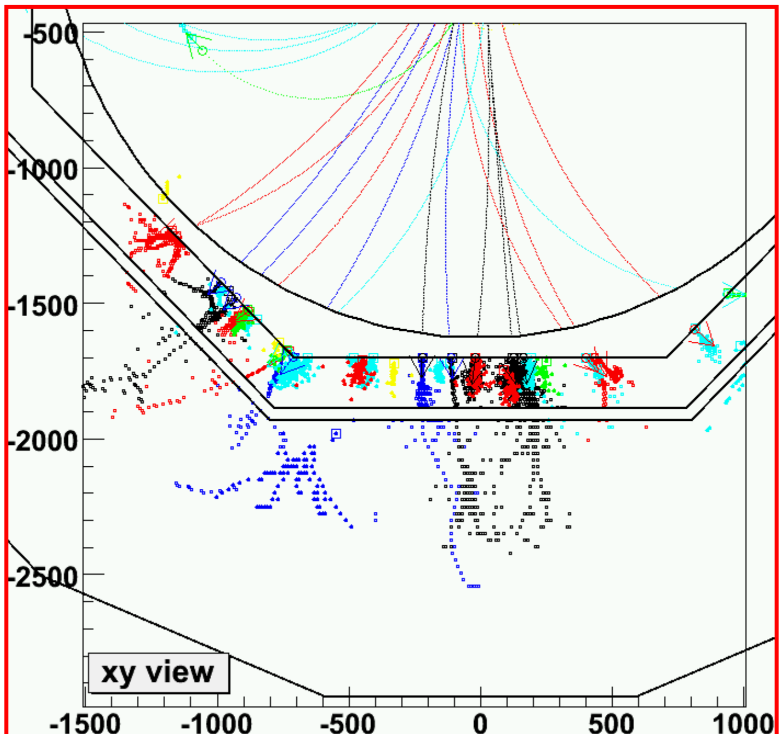
# Towards Full Simulation ILC Physics
# PFA, Vertexing,...

**Mark Thomson**
**University of Cambridge**




## This Talk:

- ❶ Why?
- ❷ Overview of Marlin Processors
- ❸ Progress with Tracking
- ❹ Progress with Vertexing Software (ZVTOP)
- ❺ Progress with PFA
- ❻ Where are we now?

**Apologies to American/Asian colleagues – this will be an Eurocentric view**

# ❶ Why, oh why?

★ To date, almost all ILC physics studies used fast simulations with parameterised detector response (including many Tesla TDR studies)

**So, why move to (less convenient) full simulation?**

i) PHYSICS PERFORMANCE:

- ★ Parameterisation in fast sim. is almost certainly wrong, e.g. at 1 TeV, doubt we will achieve $30\,\%/\sqrt{E(\mathrm{GeV})}$ jet energy res.
- ★ Not clear to me that for reliable statements about physics performance can use fast simulation

ii) DETECTOR OPTIMISATION:

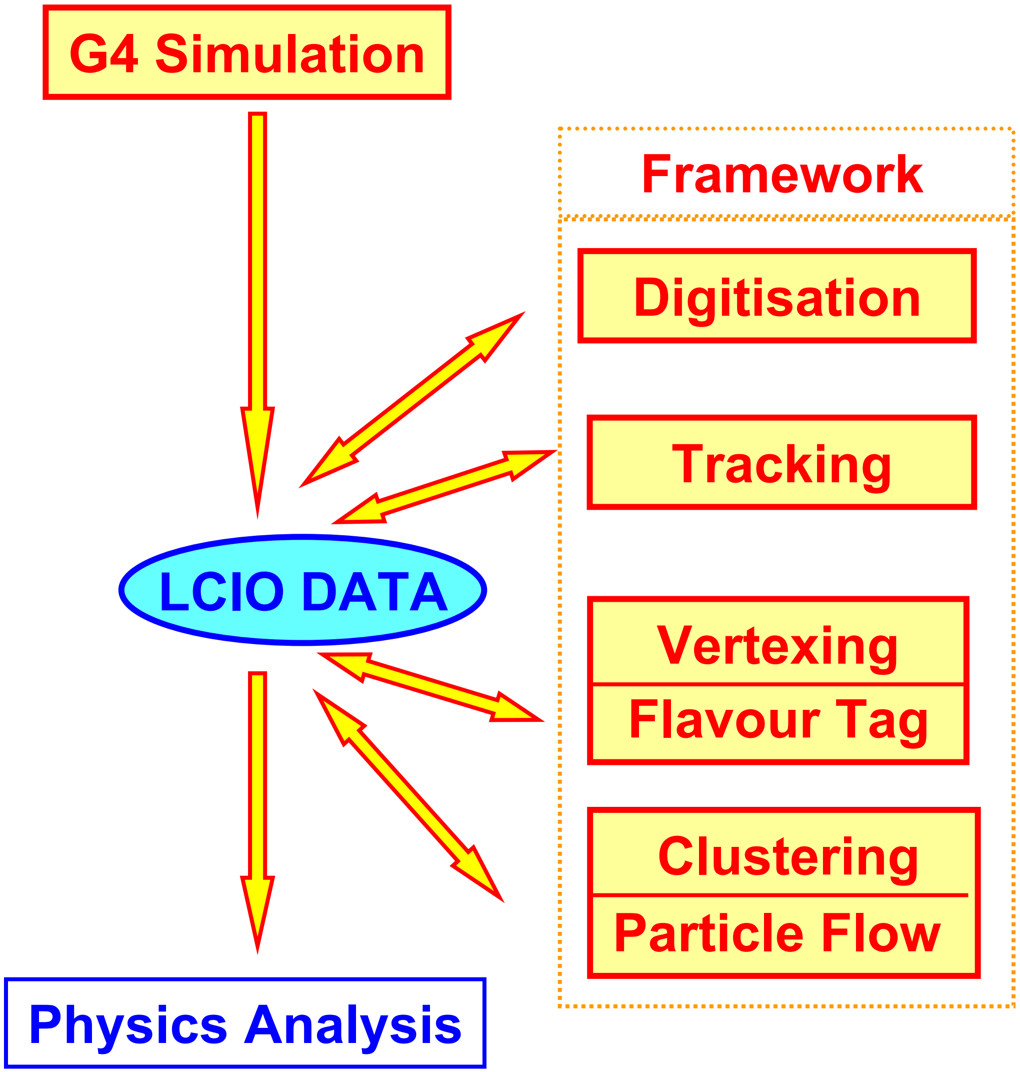
- ★ Optimisation of **nM$100** detectors needs to be based on physics performance – only way to study this is full simulation
- ★ 3 out of 4 detector concepts choose high granularity calorimetry (i.e. high cost) for particle flow. **NEED** to be convinced that the Particle Flow paradigm is the right way forward.

✚ In the not so distant future, we will produce detector CDRs/TDRs: full simulation of (at least) some processes will be needed

**For software tools to develop – need to start using them now**

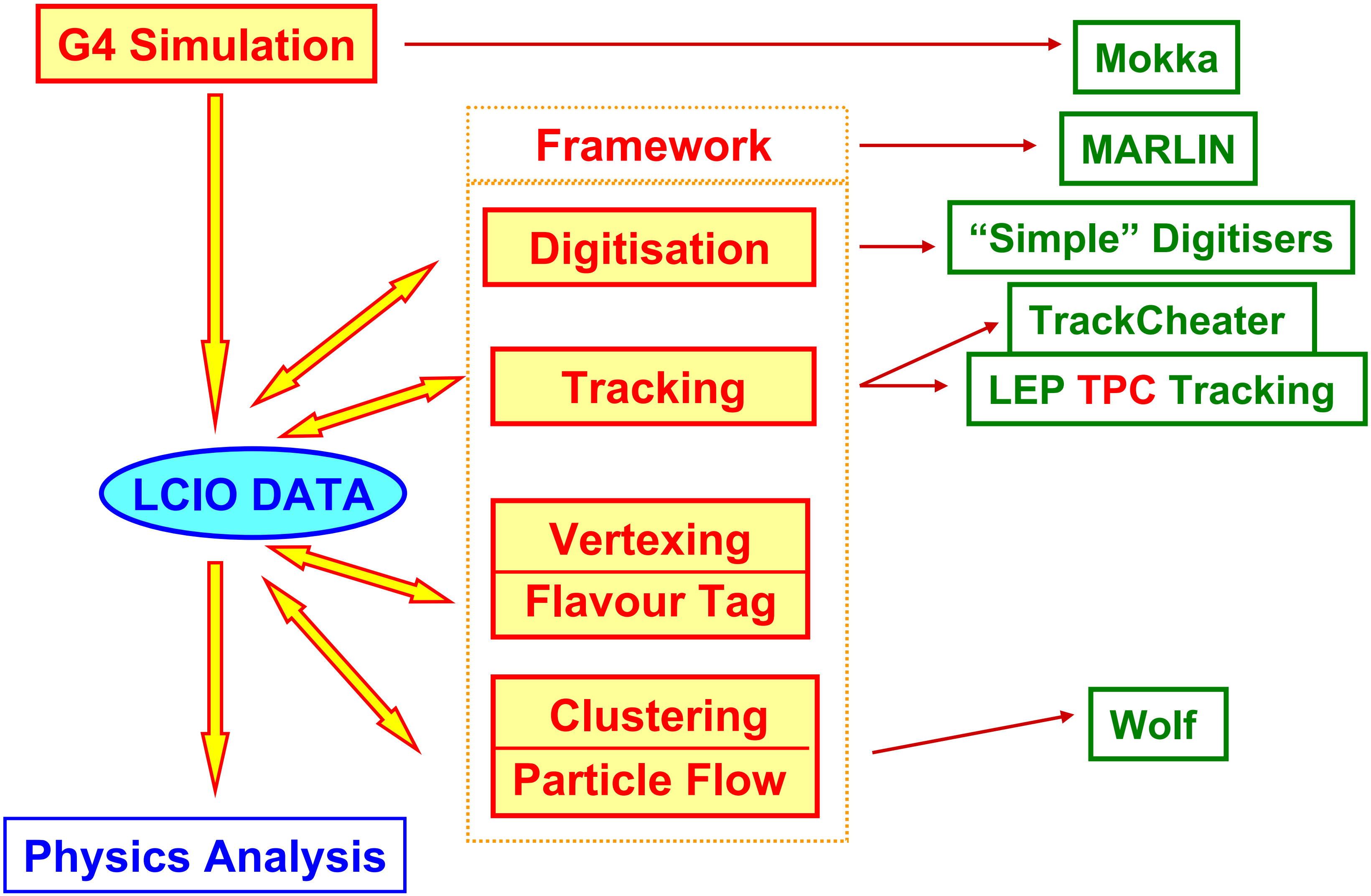
# ❷ Reasons to be Cheerful, 1

## What software is needed?



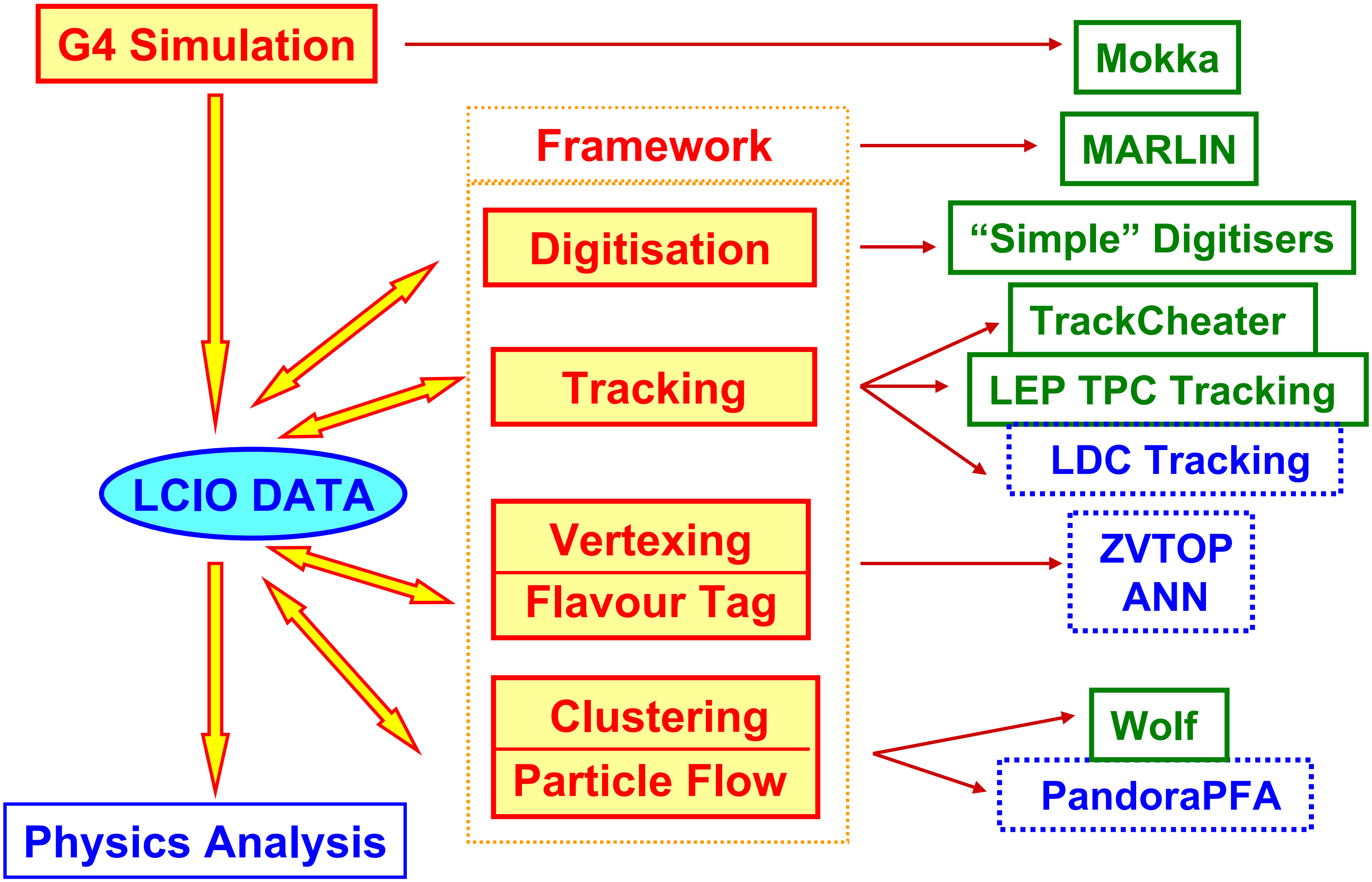
# Reasons to be Cheerful 1,2

**What software is needed?** | **What existed last year**



# Reasons to be Cheerful 1,2,3

**What software is needed?** **What exists now?**



★ A lot of progress over last year !

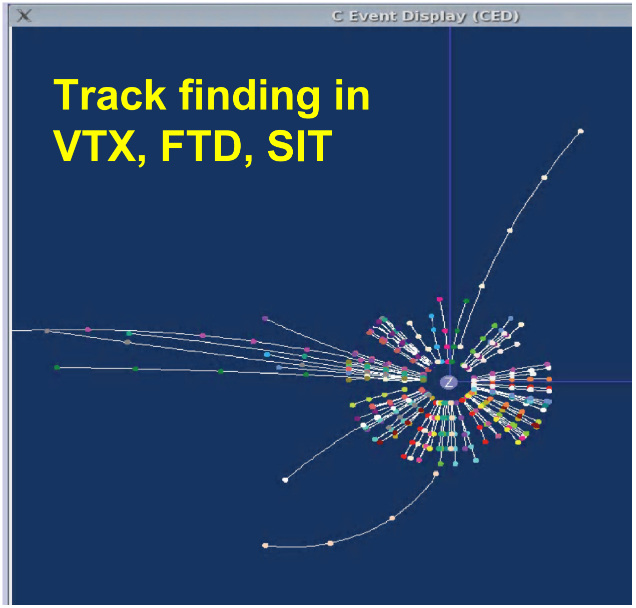
**What do we need for physics studies?**

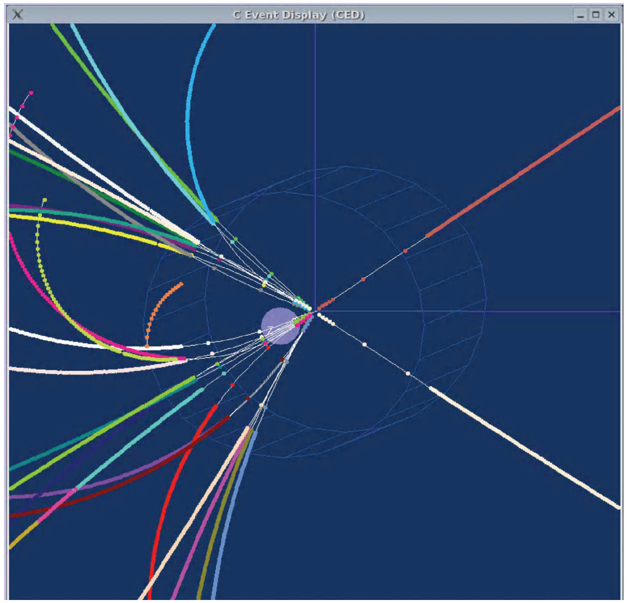
- need realistic Vertexing/Flavour Tag
- need realistic Particle Flow
- would like Full Tracking (VTX, FTD, SIT, TPC)
  – but cheated tracks probably OK
- ultimately would like realistic digitisation with noise etc.
  – but can wait

• In the following 230 slides will discuss recent progress with VERTEXING and PFA + status of tracking…

# ❸ Progress with Tracking

★ Previously only had TPC Pattern Recognition from LEP (S.Aplin)
★ Now have full PatRec in VTX, FTD, SIT (A. Raspereza)




★ + matching to TPC tracks (A. Raspereza)

⇨ **LDCTracking Processor**

- Pattern recognition finished !
- Working on full track parameter reco. taking into account energy loss
- Aim to finish by Christmas

**Very nice work - big step forward !**

# ❹ Progress with Vertexing

★ Many physics analyses at the ILC will rely on Heavy Flavour tagging
- either directly, e.g. BR(H→cc)
- + majority of studies will employ flavour tagging to select signal/reject background

★ To determine ultimate ILC physics sensitivity need "realistic", i.e. **sophisticated** vertex reconstruction software

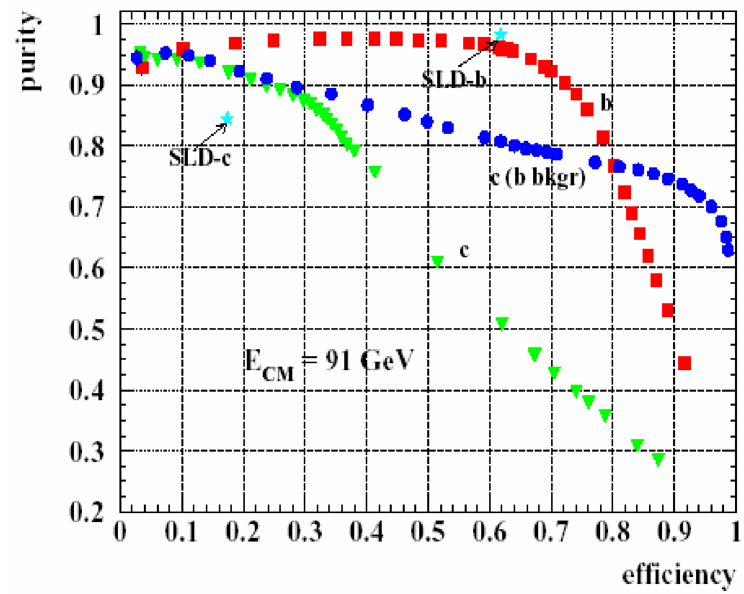
★ TESLA TDR studies used powerful techniques from **SLD/LEP**

- TDR vertex detector + code: achieved good performance, c.f. SLD

→ **Brave New World of C++/Marlin/…**

- Until recently – **nothing in C++**: **a major hole in our software**

*Klaus Desch/ Thorsten Kuhl*



# Current Status

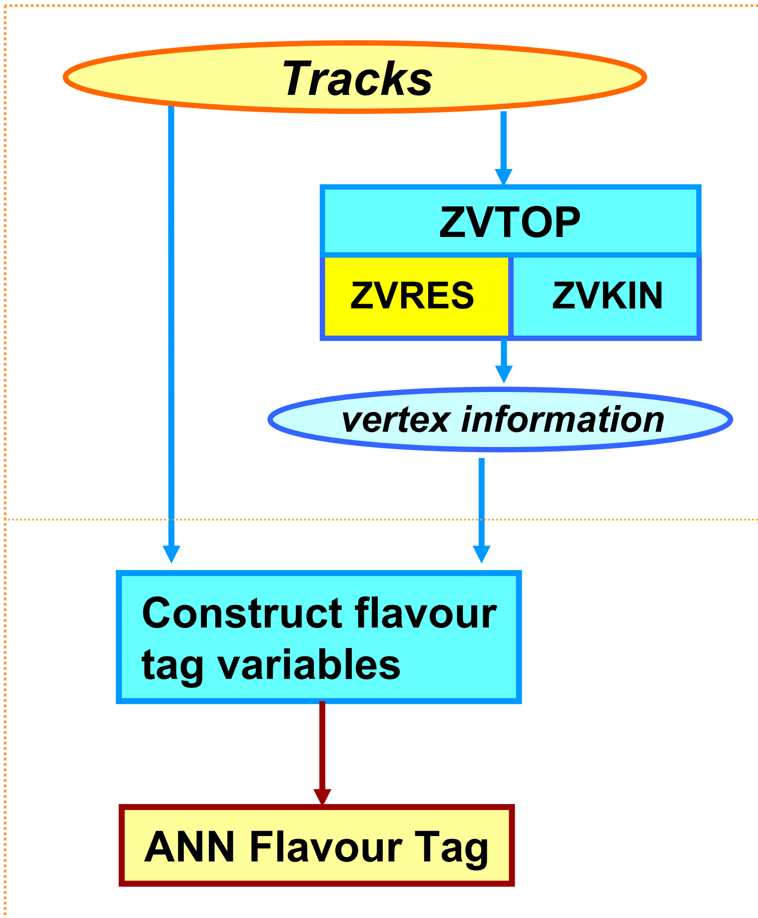
- ★ Real progress in the last year (two groups)
  - ✦ RAVE/Vertigo vertexing toolkit (Vienna group, Mitaroff, Waltenburg, et al.):
    - – potentially powerful generic "toolkit" approach
  - ✦ ZVTOP (LCFI group, Hillert et al.)
    C++/Marlin implementation of tried and tested Fortran algorithms
- ★ Today concentrate on ZVTOP approach
  [for full details see talks of S.Hillert and B.Jeffery in this afternoon's Simulation/Reconstruction session 1700-1840]

# ZVTOP (in Marlin)

*D. Jackson, NIM A 388 (1997) 247*

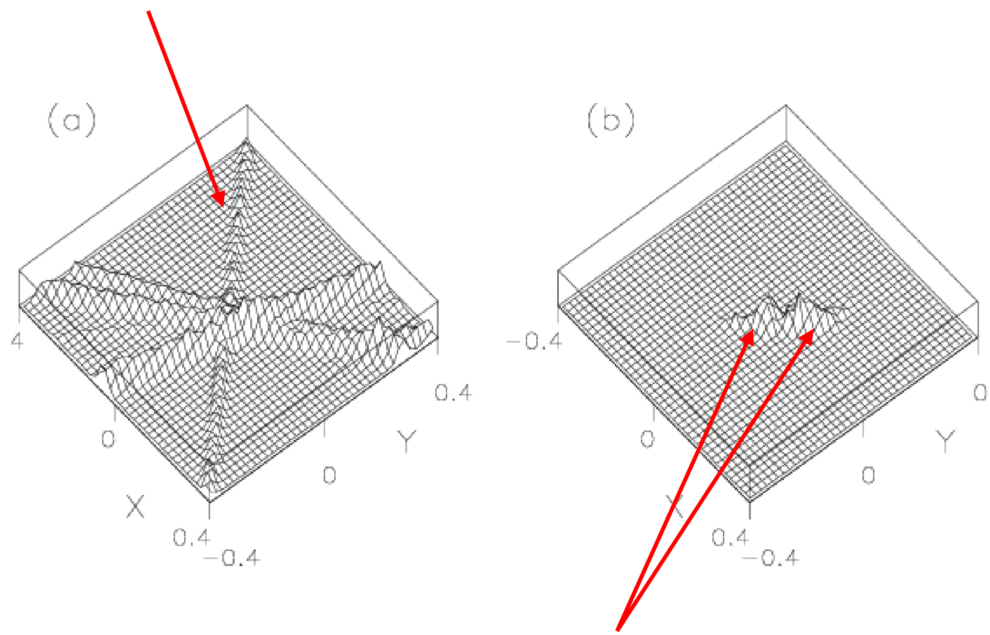
- ★ Brand new C++ implementation of ZVTOP
- ★ Major undertaking:
  - ▪ 16816 lines (!) of structured/documented code
  - ▪ Great care in validating each step against existing Fortran code

★ **Separate Marlin processors for Vertex finding and Flavour Tagging**
★ **Vertexing algorithm ZVTOP has two main parts**



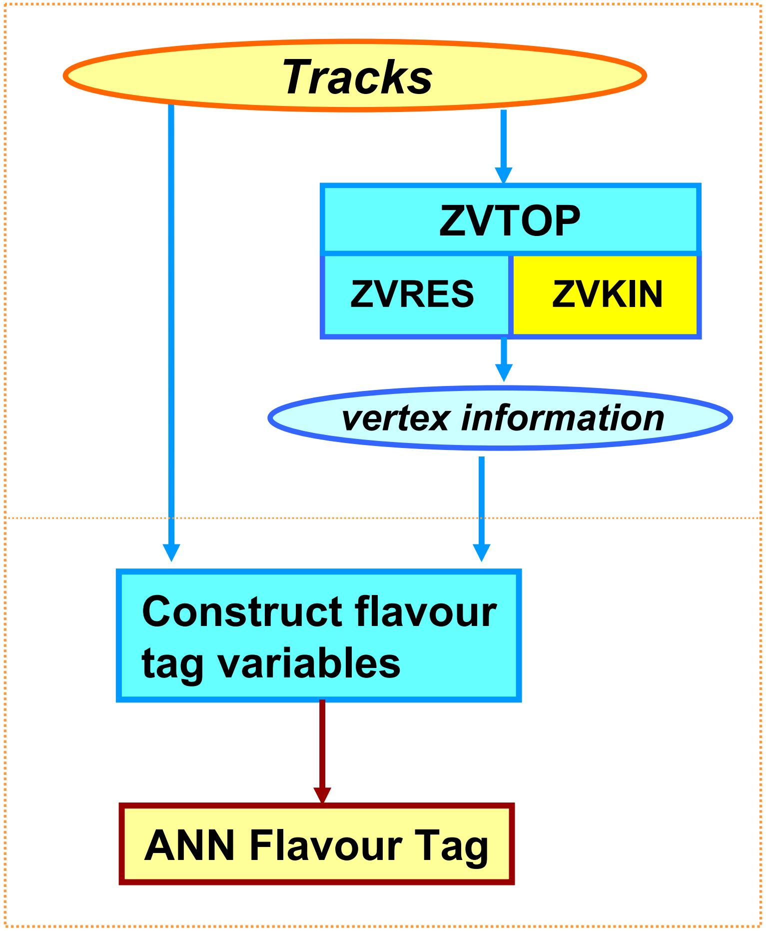

**ZVRES:**

- **Tracks represented as 3D Gaussian probability tubes**



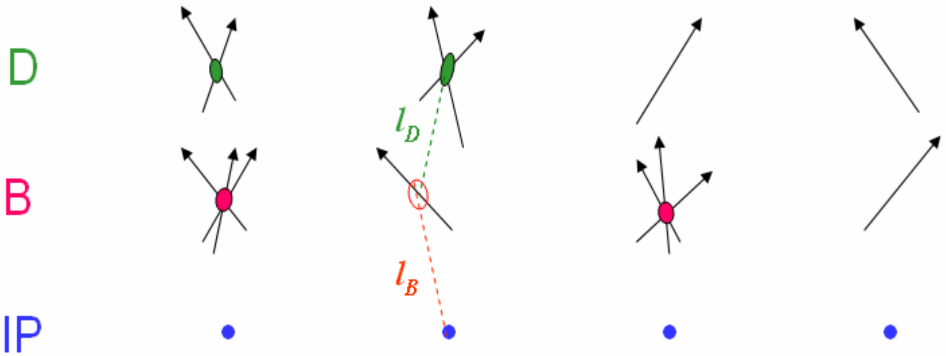

- **Overlaps used to identify vertices**

★ Separate Marlin processors for Vertex finding and Flavour Tagging
★ Vertexing algorithm ZVTOP has two main parts



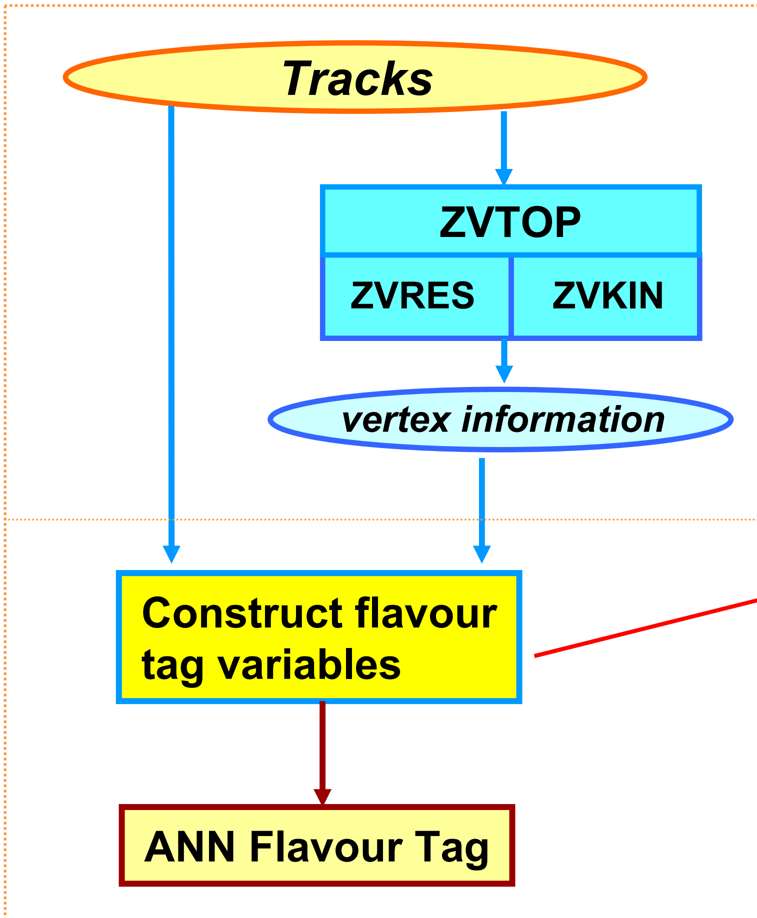

**ZVKIN:**

- Used by SLD
- Not in previous Brahms (fortran) code
- Should improve tagging for single prong B/D decays or where B decay not well resolved from vertex

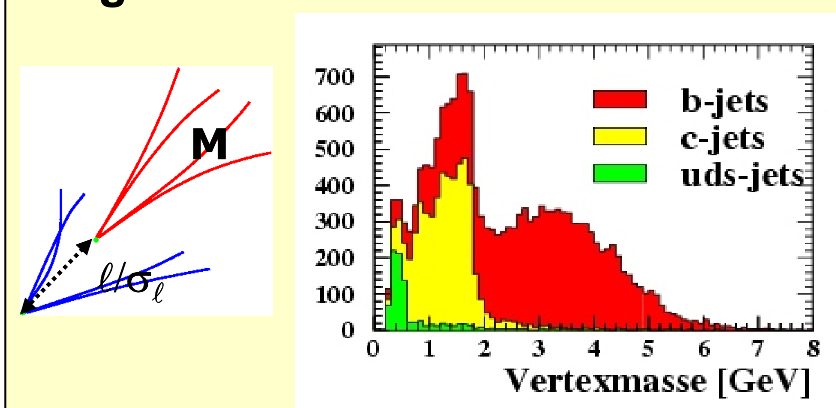


★ **Separate Marlin processors for Vertex finding and Flavour Tagging**

★ **Vertex information + tracking used for flavour tag variables**

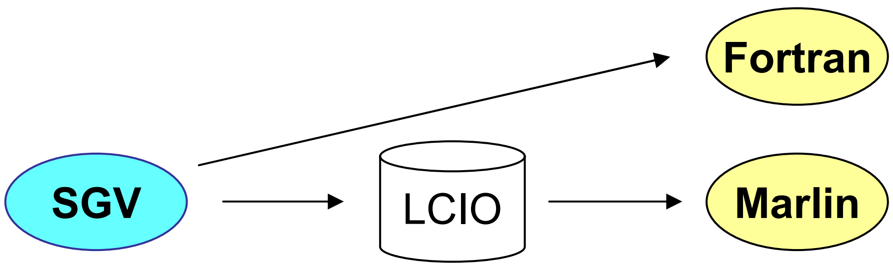



**e.g. vertex mass**




★ **Finally, apply Neural Network flavour tag using essentially the same method as earlier fortran code** (R.Hawkings, LC-PHSM-2000-021 )

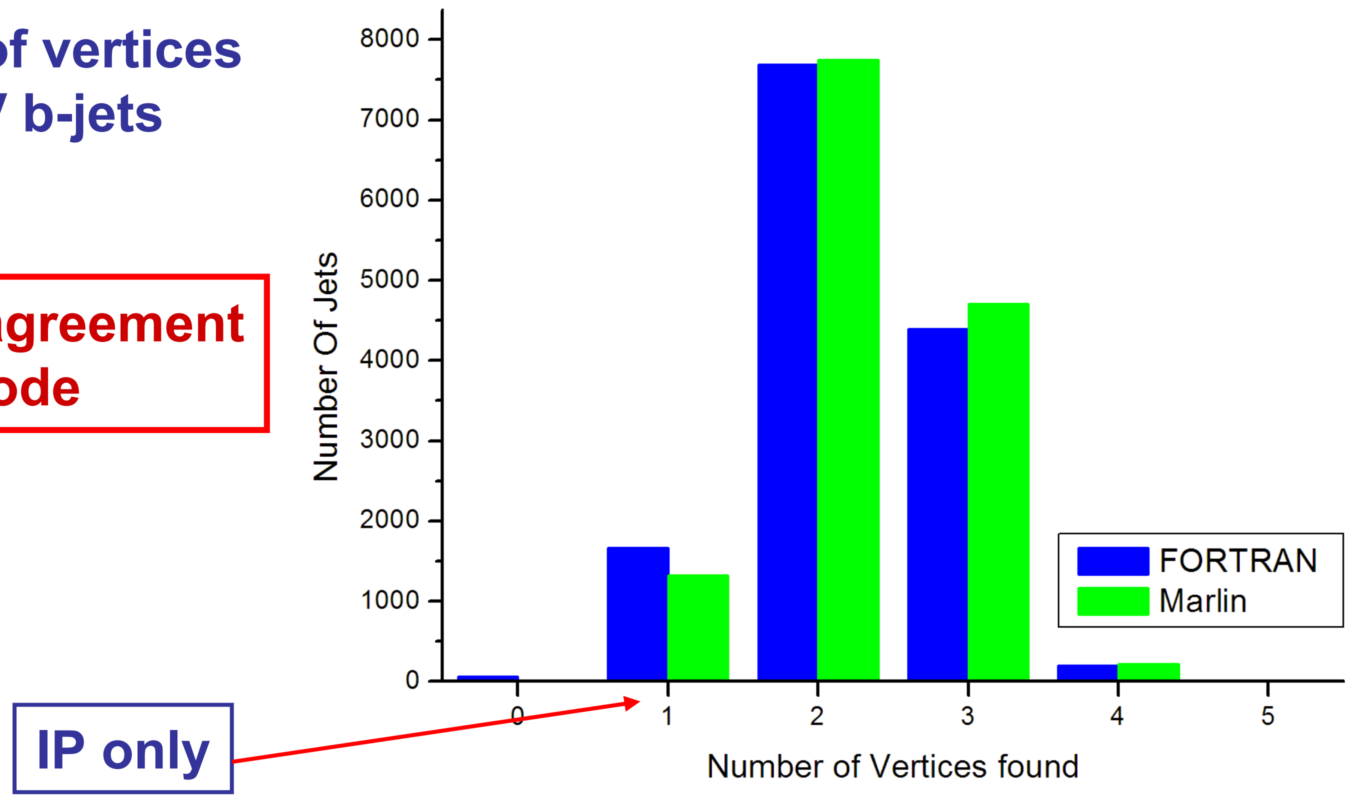
# Validation



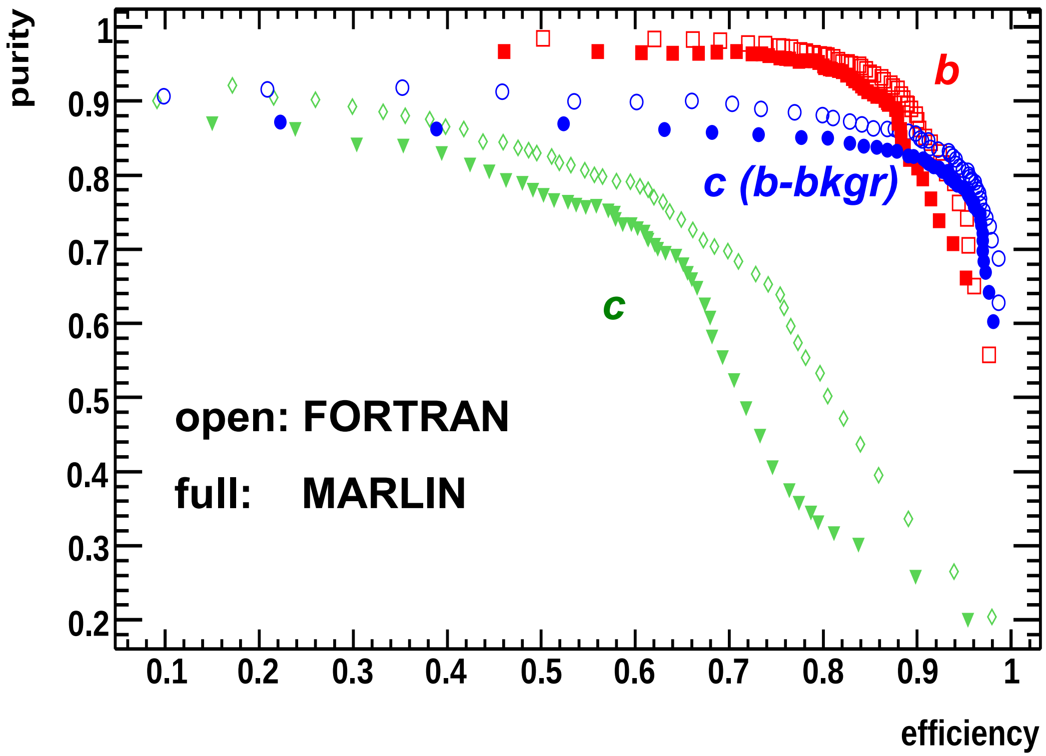

- ★ Tested using same events generated using fast MC (SGV)
- ★ Direct comparison of ZVTOP part

e.g. number of vertices in ~100 GeV b-jets

**ZVTOP: good agreement with Fortran code**



# Current Performance




- Performance is good
- But not yet quite as good as Fortran code (same events)
- very encouraging first attempt !

✦ Will soon be tested with Mokka (rather than SGV) tracks
✦ First release on timescale of a few weeks !
✦ Very likely to have first version before Christmas
✦ A big step forward !

# ❺ Progress with Particle Flow

- ★ Three of the four detector concepts based around the PFA approach to calorimetry.
- ★ Need sophisticated reconstruction software for detector optimisation and physics studies

## Where are we now ?

- ★ Within Marlin framework, 2 reasonably well developed algorithms:
  - **WolfPFA** (A. Raspereza et al.)
  - **PandoraPFA**
  - \+ current development (DESY group) of new structured "toolkit" approach (too early for results)
- ★ In terms of performance, PandoraPFA is currently leading the way.
- ★ Here will describe current status of PandoraPFA:
  - ♦ overview of algorithm
  - ♦ current performance
  - ♦ some example detector optimisation studies
  - ♦ use in physics studies…

# ① PandoraPFA Overview

★ ECAL/HCAL reconstruction and PFA performed in a single algorithm
★ Keep things fairly generic algorithm
  - applicable to multiple detector concepts
★ Use tracking information to help ECAL/HCAL clustering

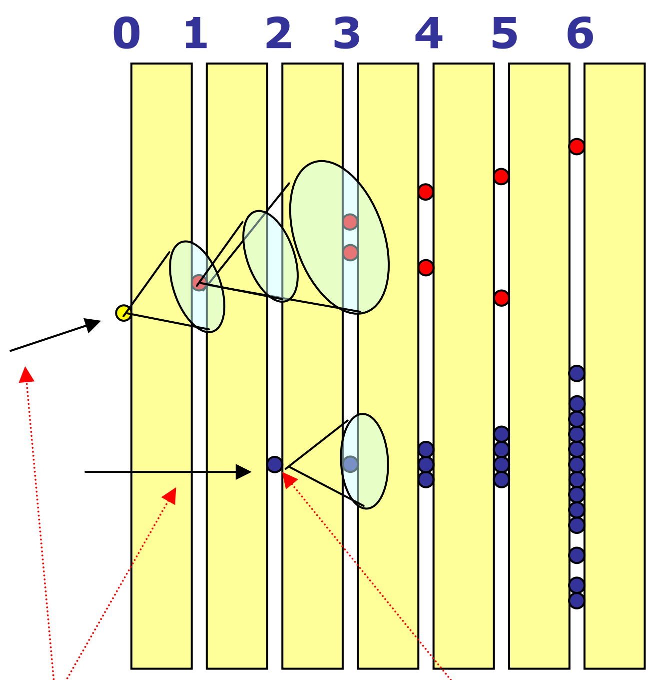
★ This is a fairly sophisticated algorithm : ~8000 lines of code

**Five Main Stages:**

i. Loose clustering in ECAL and HCAL
ii. Topological linking of clearly associated clusters
iii. Courser grouping of clusters
iv. Iterative reclustering
v. Formation of final Particle Flow Objects (reconstructed particles)

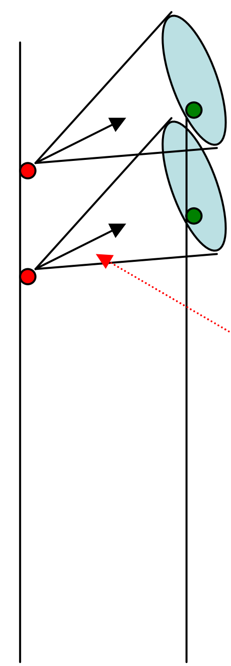
# i) ECAL/HCAL Clustering

- ★ Start at inner layers and work outward
- ★ Tracks can be used to “seed” clusters
- ★ Associate hits with existing Clusters
- ★ If no association made form new Cluster
- ★ Simple cone based algorithm




Initial cluster direction

Unmatched hits seeds new cluster



Simple cone algorithm based on current direction + additional N pixels

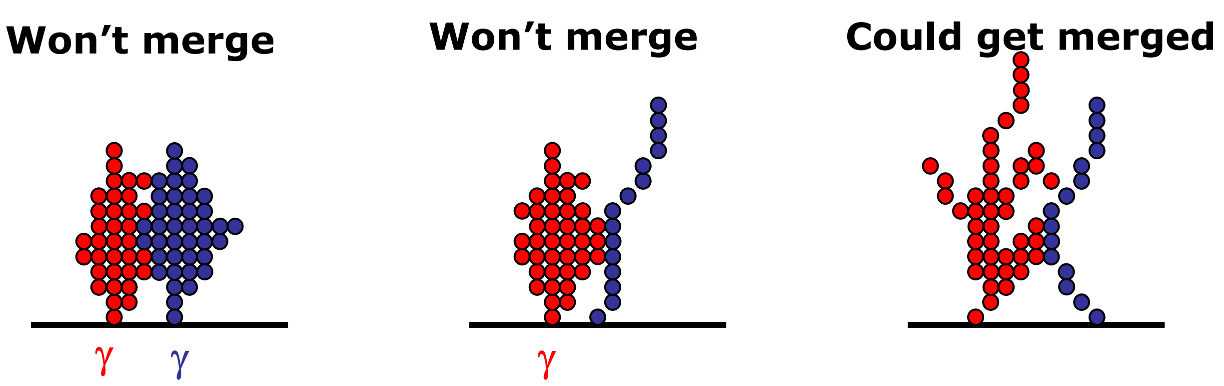
Cones based on either:
initial PC direction or
current PC direction

# ii) & iii) Cluster Association

- By design, clustering errs on side of caution
  i.e. clusters tend to be split
- Philosophy: easier to put things together than split them up
- Clusters are then associated together in two stages:
  - 1) Tight cluster association – clear topologies
  - 2) Loose cluster association – fix what's been missed

## Photon ID

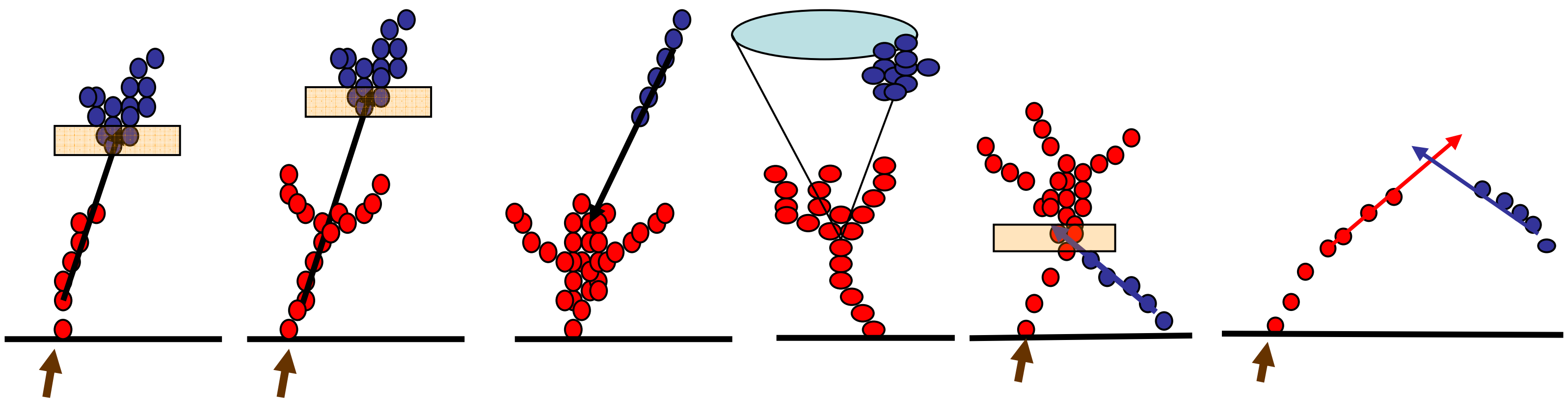
- Photon ID plays important role
- Simple "cut-based" photon ID applied to all clusters
- Clusters tagged as photons are immune from association procedure – just left alone



★ Clusters associated using a number of topological rules

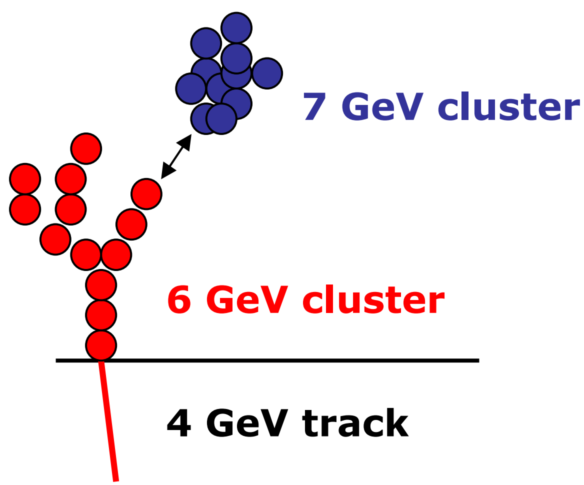
## Clear Associations:

- Join clusters which are clearly associated making use of high granularity + tracking capability: very few mistakes



## Less clear associations:

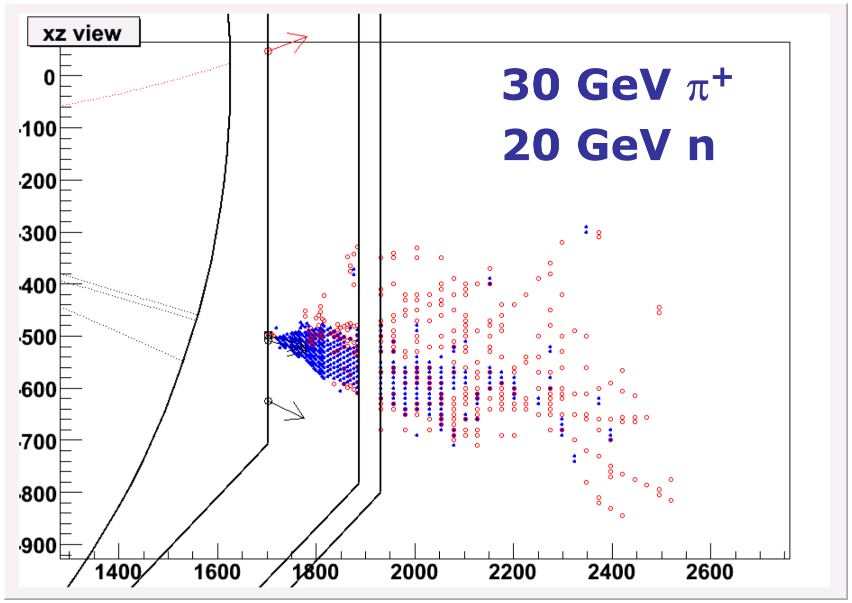
e.g. **Proximity**




**Use E/p consistency to veto clear mistakes**

# iv) Iterative Reclustering

★ Generally performance is good – but some difficult cases…




★ At some point hit the limit of "pure" particle flow
- just can't resolve neutral hadron in hadronic shower

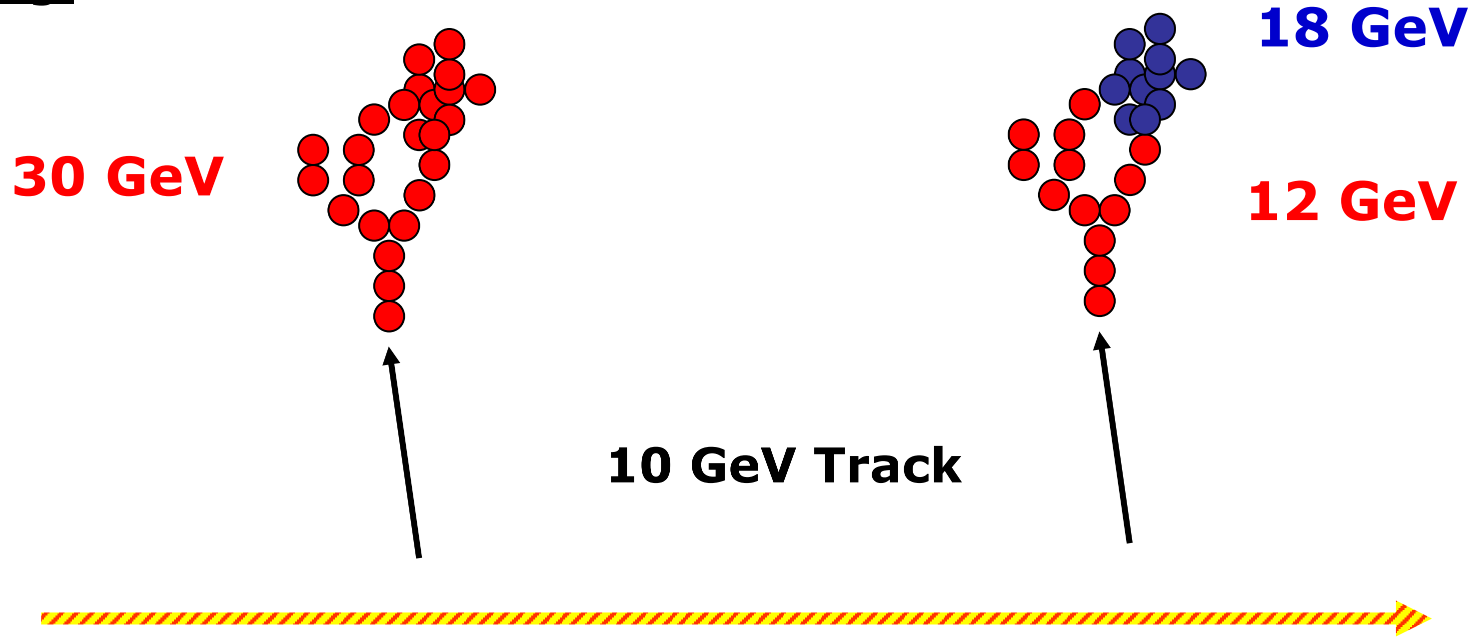
The ONLY(?) way to address this is "statistically"

→

e.g. if have 30 GeV track pointing to 20 GeV cluster
SOMETHING IS WRONG

★ **If track momentum and cluster energy inconsistent : RECLUSTER**

**e.g.**




**Change clustering parameters until cluster splits and get sensible track-cluster match**

**NOTE:** **NOT FULL PFA as clustering driven by track momentum**

**This is very important for higher energy jets**

# ② Current Performance

**Figures of Merit:**

## $rms_{90}$

★ Find smallest region containing 90 % of events

★ Determine rms in this region

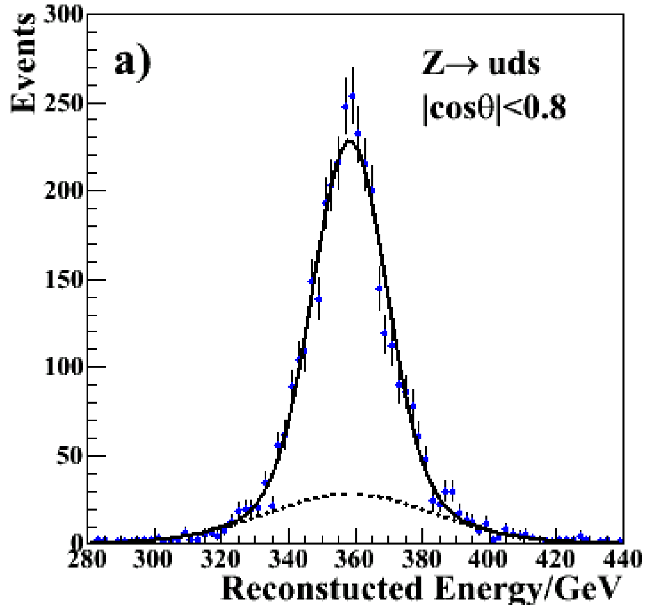
| $E_{JET}$ | $\sigma_E/E = \alpha\sqrt{(E/GeV)}$ $\lvert\cos\theta\rvert<0.8$ |
|---|---|
| 45 GeV | 0.30 |
| 100 GeV | 0.37 |
| 180 GeV | 0.57 |
| 250 GeV | 0.75 |

**For jet energies < 100 GeV performance is probably good enough for physics studies**

## $\sigma_{75}$

★ Fit sum of two Gaussians with same mean. The narrower one is constrained to contain 75% of events

★ Quote $\sigma$ of narrow Gaussian




It is found that $rms_{90} \approx \sigma_{75}$

# ③ Recent Detector Optimisation Studies

★ From point of view of detector design – what do we want to know ?

**Optimise performance vs. cost**

★ Main questions (the major cost drivers):

- Size : performance vs. radius
- Granularity (longitudinal/transverse): ECAL and HCAL
- B-field : performance vs. B

★ To answer them use MC simulation + PFA algorithm

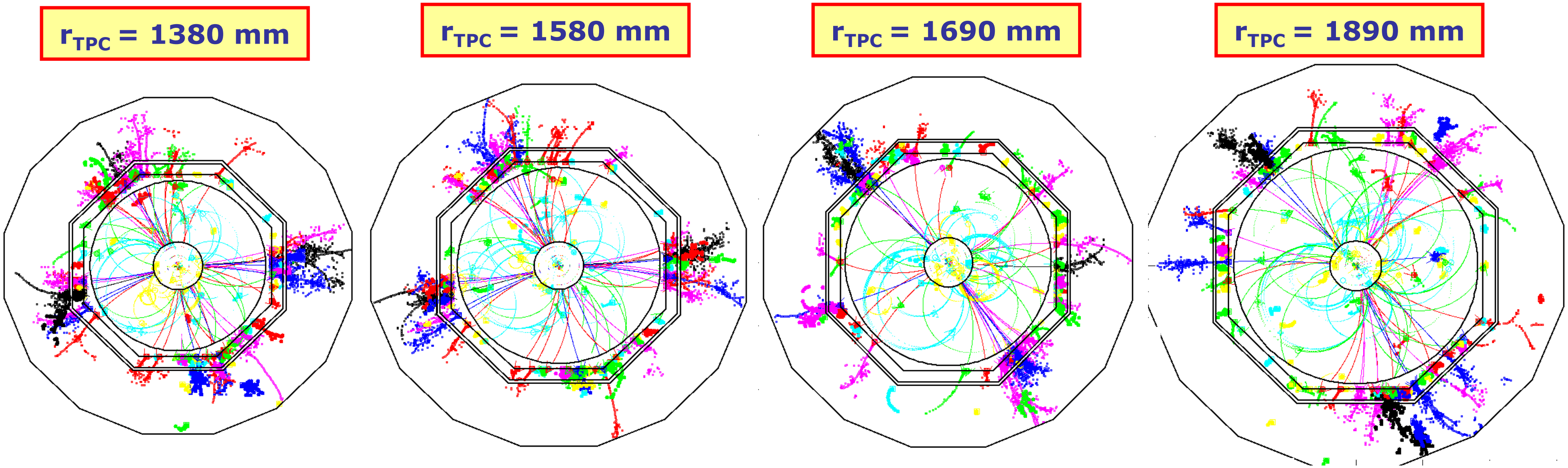
⚠

- Need a good simulation of hadronic showers
- Need realistic PFA algorithm
  (want/need results from multiple algorithms)

★ This is important – significant impact on overall design of *xxx* M$ detector !

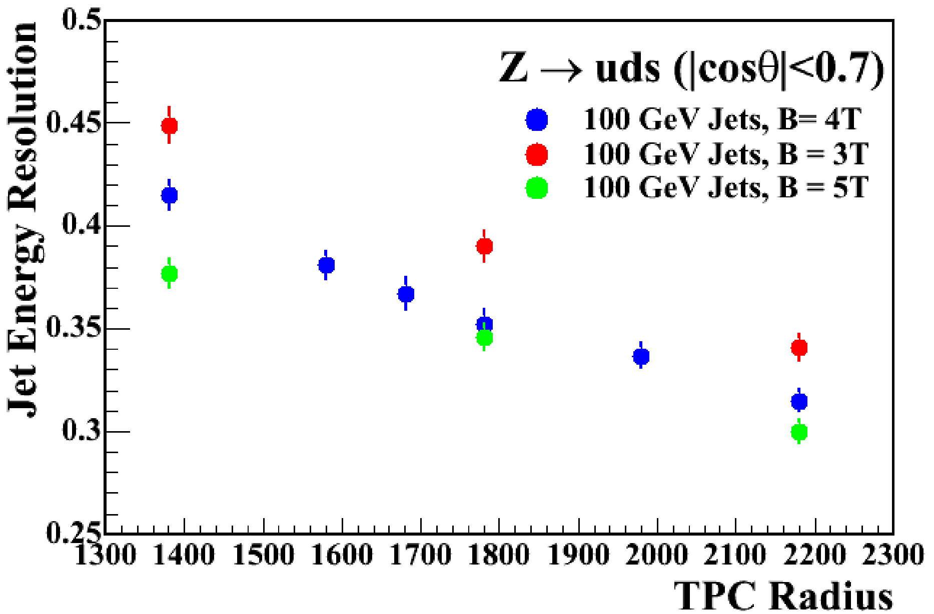
★ Interpretation of results needs care – observe effects of detector + software

# e.g. Radius/Field

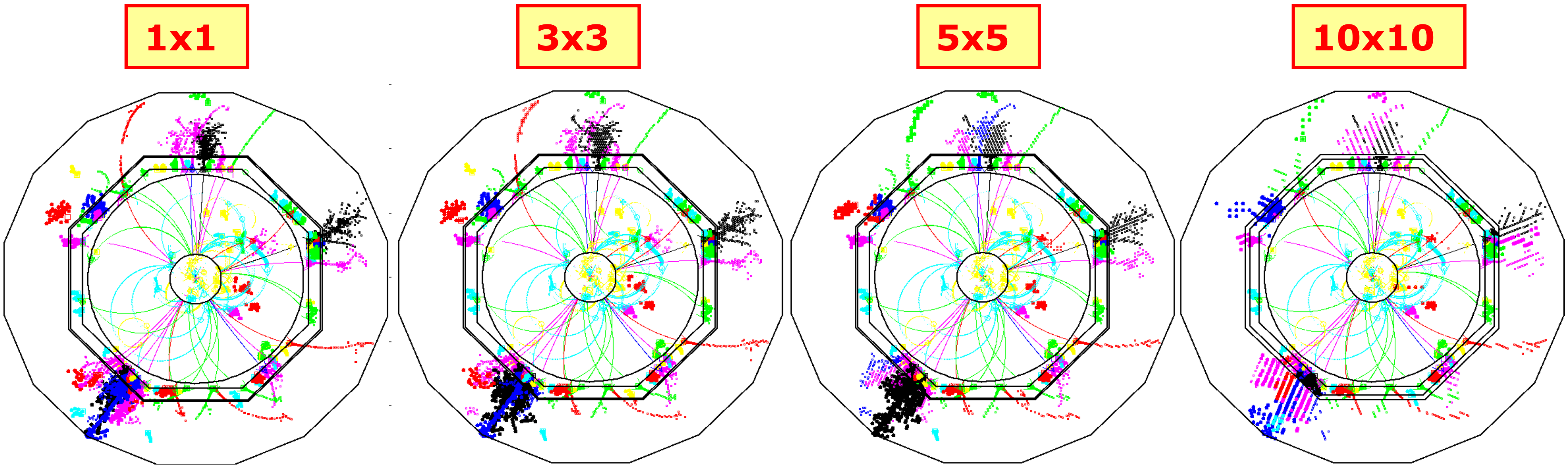



**e.g. 100 GeV uds Jets in Barrel**

- ★ Performance vs. radius/B (Tesla TDR detector)
- ★ Argues for large high field
- ★ With a reasonable cost model for ECAL+HCAL and Solenoid could identify "optimal" parameters



# e.g. HCAL Transverse Granularity




| Detector Model | $\sigma_{Evis}/E = \alpha\sqrt{(E/GeV)}$ | |
|---|---|---|
| | Z @91 GeV | tt@500 GeV |
| LDC00Sc 1cm x 1cm | 31.4 ± 0.3 % | 42 ± 1 % |
| LDC00Sc 3cm x 3cm | 30.6 ± 0.3 % | 45 ± 1 % |
| LDC00Sc 5cm x 5cm | 31.3 ± 0.3 % | 48 ± 1 % |
| LDC00Sc 10cm x 10cm | 33.7 ± 0.3 % | 56 ± 1 % |

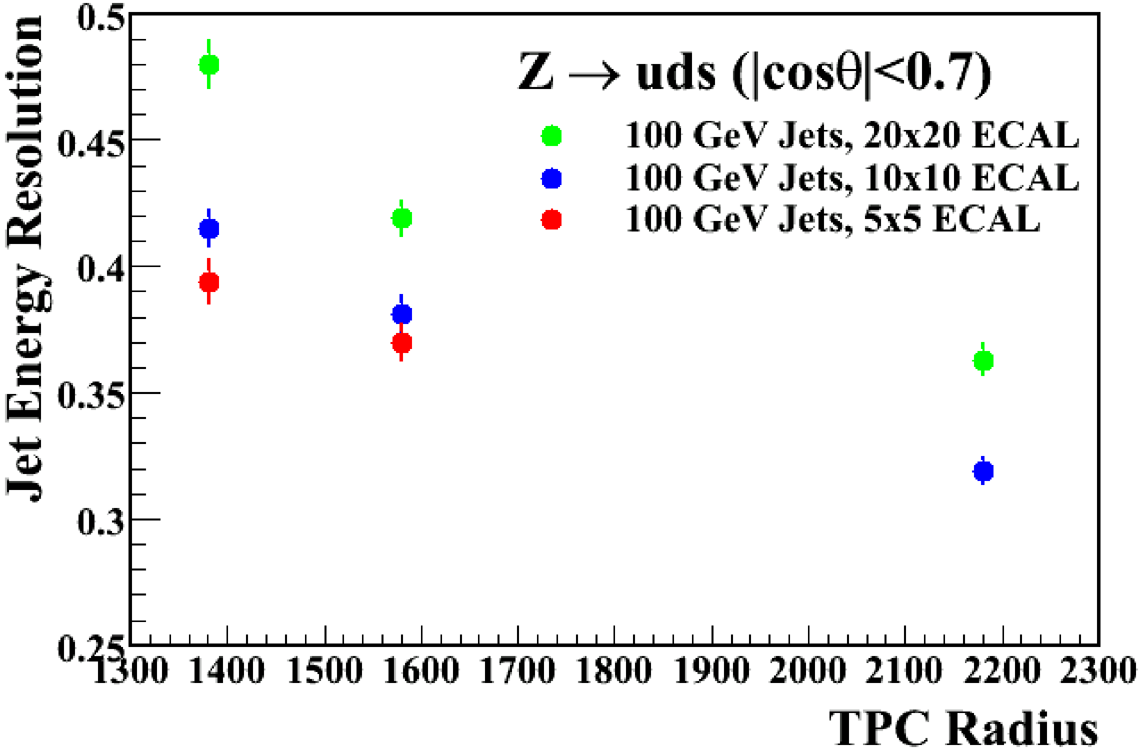
Visible energy resolution

- ★ 10x10 too coarse (can be seen clearly from display)
- ★ Finer granularity helps(?) somewhat at higher energies

# e.g. ECAL Transverse Granularity

- Use Mokka to generate Z → uds events @ 200 GeV with different ECAL segmentation: 5x5, 10x10, 20x20 [mm$^2$]

- Detector model: LDC00Sc (~Tesla TDR)
- B = 4 Tesla
- 30x30mm$^2$ HCAL




## With PandoraPFA

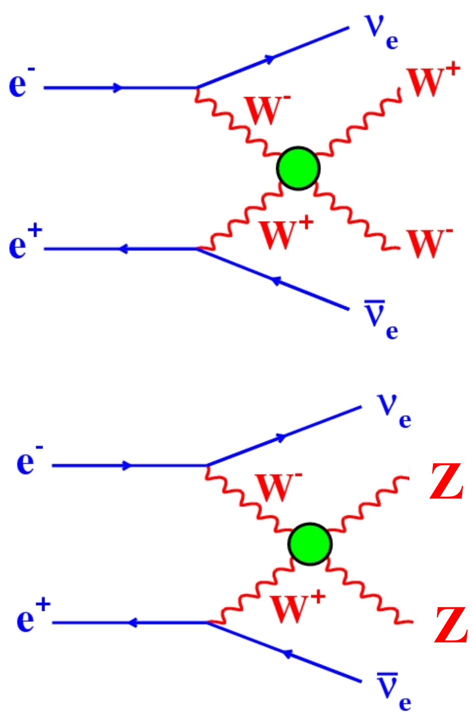
- 20x20 segmentation looks too coarse
- For 100 GeV jets, not a big gain going from 10x10 → 5x5mm$^2$

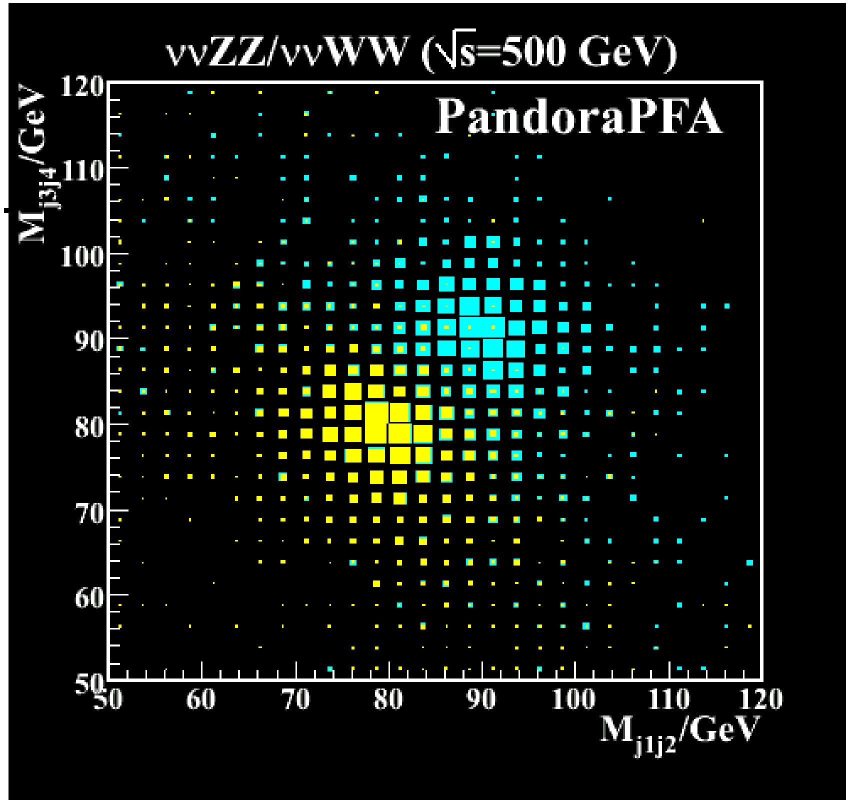
  [for these jet energies the contributions from confusion inside the ECAL is relatively small]

# ④ "Physics Studies"

- ★ PandoraPFA is not perfect, but does a reasonable job
- ★ Can start to use it for full simulation physics studies
- ★ Using Marlin jet finders + reconstructed PandoraPFA reconstructed PFOs, it took me about 1 hour to produce the "classic" PFA plot

$e^+e^- \rightarrow \nu\nu WW \rightarrow \nu\nu qqqq$ , $e^+e^- \rightarrow \nu\nu ZZ \rightarrow \nu\nu qqqq$



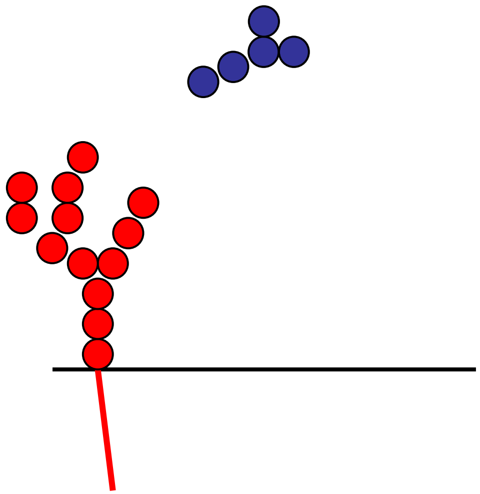
# PandoraPFA : Outlook

★ Code was "released" in the Summer:
  **http://www.hep.phy.cam.ac.uk/~thomson/pandoraPFA**

★ Shortly after this meeting will go into Marlin CVS repository

★ Still some work to do…

★ Currently performance almost entirely limited by low energy "hadronic fragments"



- No attempt to identify these in current code
- Has been on "to do" list for some time
- Hope to address this before Christmas

✦ Also working to make code compatible with SLIC and Jupiter events – clustering works well, some issues with tracking

# ❻ Where are we now?

- Significant progress in all main areas of ILC reconstruction software
- Can currently perform full simulation physics analysis with:

**Digitisers** + **TrackCheater** + **PandoraPFA**

- Hopefully, by Christmas will be able to do it all !

**Digitisers** + **LDCTracking** + **ZVTOP** + **PandoraPFA**

**Time to start full simulation physics studies !**