MOKKA/ MARLIN

Ties Behnke, DESY

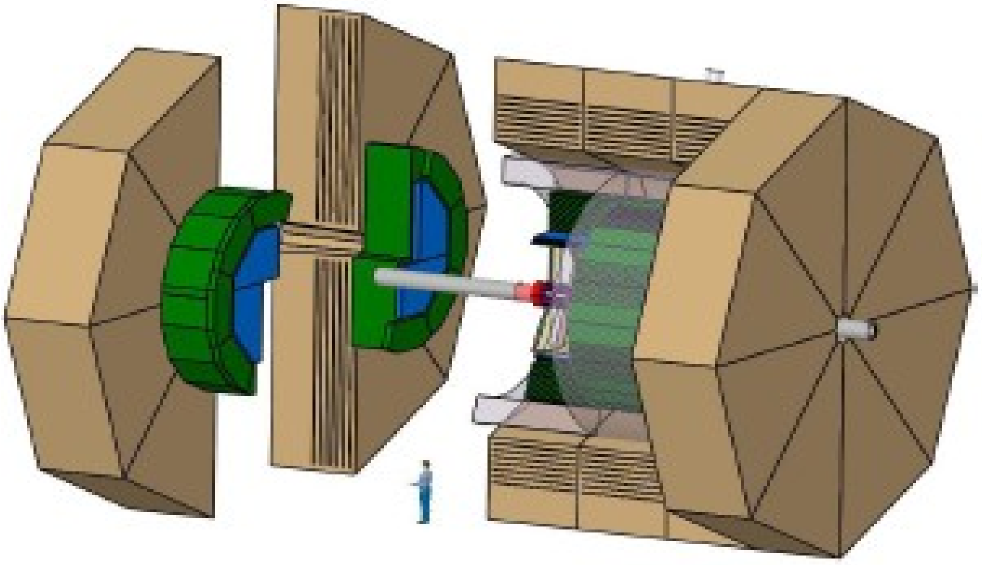
C++ based software system

for reconstruction and
analysis of ILC events

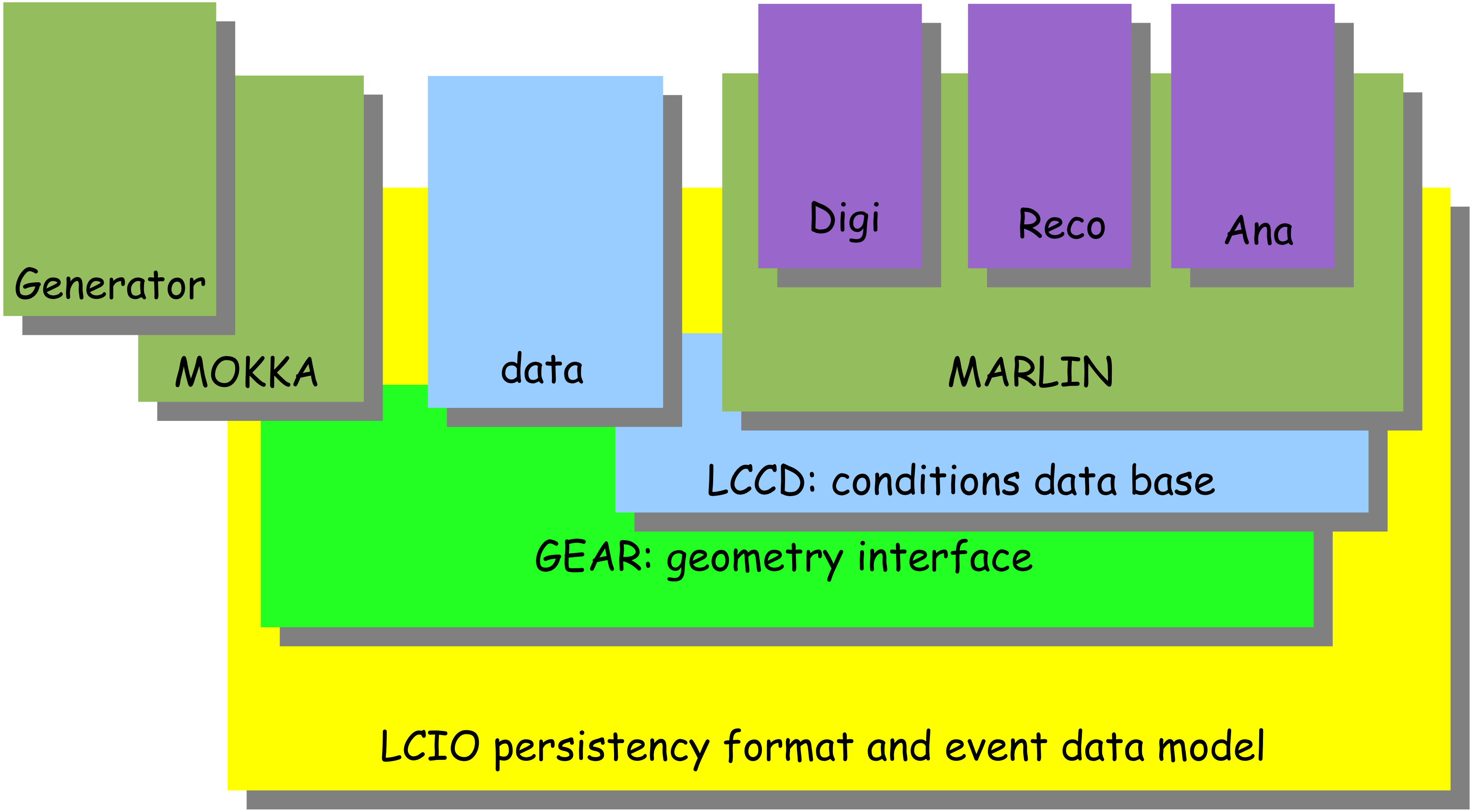
both simulated and real

prime focus so far: LDC concept

evolve towards ILD in the near future



Concept and Architecture


Generator
MOKKA
data
Digi
Reco
Ana
MARLIN
LCCD: conditions data base
GEAR: geometry interface
LCIO persistency format and event data model

MOKKA

Geant4 based full simulation program

- Wide range of detailed models available (LDC, SiD, several test beam installations)
- writes LCIO and GEAR geometries (recently significantly improved)
- detailed subdetectors are available
- recently new model LDC-V5 released
  - Much improved LDC description
  - Improved interface to GEAR
  - New detailed calorimeter drivers
- First ILD model under development as starting point for common optimization study
- "stable" and ready for mass production (recently done 300k test events without problems (fully hadronic))

Getting ready for large scale GRID based central production

MARLIN

Modular **A**nalysis & **R**econstruction for the **L I N**ear Collider

C++ based application framework
based on LCIO as data model
configurable
extensible
main author Frank Gaede

Recent improvements:

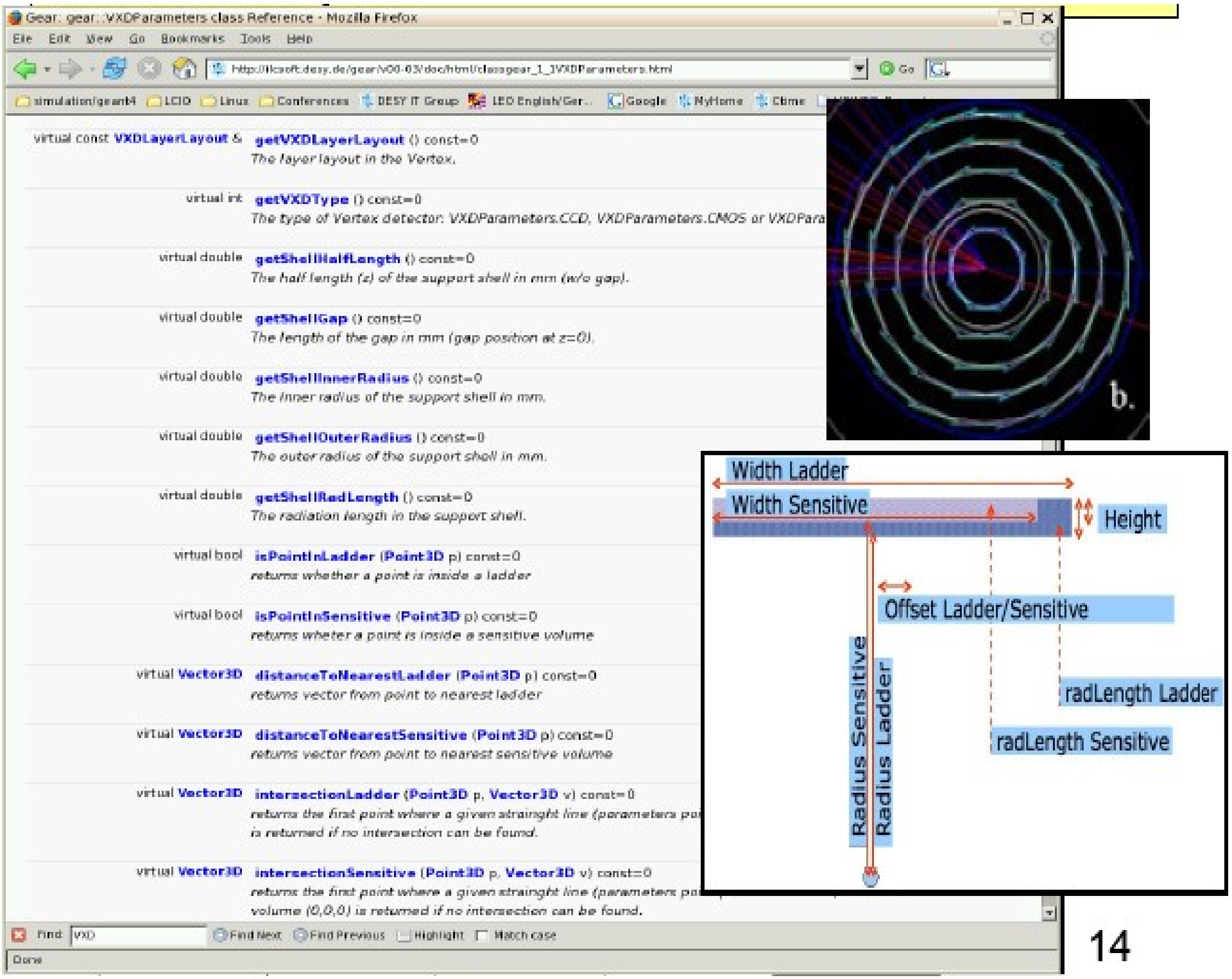
build system based on cmake
graphical user frontend to create and control program flow

MARLIN is designed to work with:

- GEAR: geometry interface package
- LCCD: conditions data tool set

MARLIN is used by
LDC
CALICE
LC-TPC
EUDET VTX
EUDET FCAL

GEAR



```
virtual const VXDLayerLayout &  getVXDLayerLayout () const=0
                                The layer layout in the Vertex.

virtual int                     getVXDType () const=0
                                The type of Vertex detector: VXDParameters.CCD, VXDParameters.CMOS or VXDPara

virtual double                  getShellHalfLength () const=0
                                The half length (z) of the support shell in mm (w/o gap).

virtual double                  getShellGap () const=0
                                The length of the gap in mm (gap position at z=0).

virtual double                  getShellInnerRadius () const=0
                                The inner radius of the support shell in mm.

virtual double                  getShellOuterRadius () const=0
                                The outer radius of the support shell in mm.

virtual double                  getShellRadLength () const=0
                                The radiation length in the support shell.

virtual bool                    isPointInLadder (Point3D p) const=0
                                returns whether a point is inside a ladder

virtual bool                    isPointInSensitive (Point3D p) const=0
                                returns wheter a point is inside a sensitive volume

virtual Vector3D                distanceToNearestLadder (Point3D p) const=0
                                returns vector from point to nearest ladder

virtual Vector3D                distanceToNearestSensitive (Point3D p) const=0
                                returns vector from point to nearest sensitive volume

virtual Vector3D                intersectionLadder (Point3D p, Vector3D v) const=0
                                returns the first point where a given straight line (parameters poi
                                is returned if no intersection can be found.

virtual Vector3D                intersectionSensitive (Point3D p, Vector3D v) const=0
                                returns the first point where a given straight line (parameters poi
                                volume (0,0,0) is returned if no intersection can be found.
```

Geometry interface package

slowly evolving to include most major sub-detectors

Plans to upgrade to LC-GEO as the next generation are on hold due to technical and manpower problems

The Test...

CALICE has developed a MARLIN based reconstruction system for the recent test beam experiment at CERN:

complete data chain established from DAQ to end – user
heavy usage of GRID infrastructure for data transfer, catalogue, reconstruction
time critical application of conditions data base

CALICE recorded successfully many millions of events

no conceptual problems found
many technical improvements and suggestions received

Learn from CALICE for the second generation test beam experiments:
MARLIN-TPC, EUDET VTX reconstruction framework

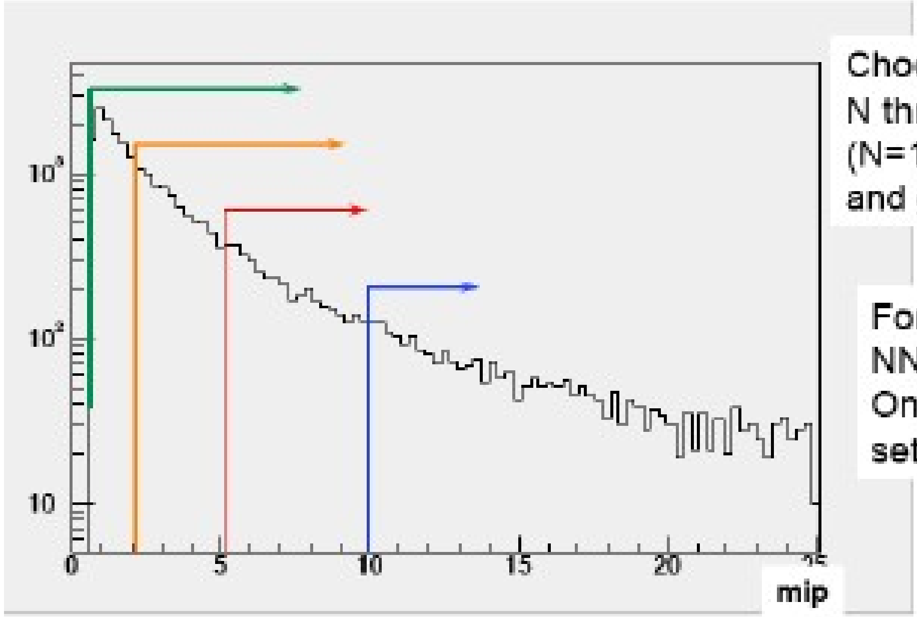
MARLIN modules: availability

First complete reconstruction version exists:

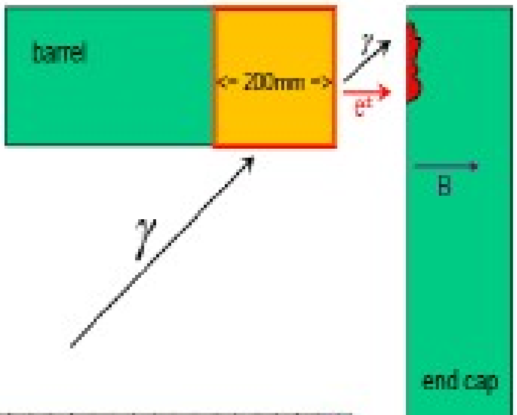
- full tracking: full solid angle, all sub-detectors, individual and combined
- vertexing: sophisticated secondary vertex reconstruction code
- jet finding, cluster finding, photon finding, etc
- three particle flow implementations:
  - WOLF
  - trackwise PFA
  - PandoraPFA (currently by far the best)

Example: Photon Finder

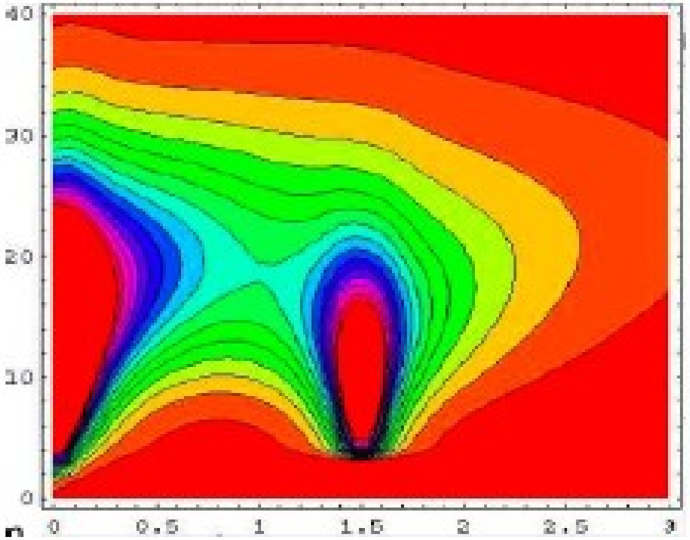
P.Krstonosic



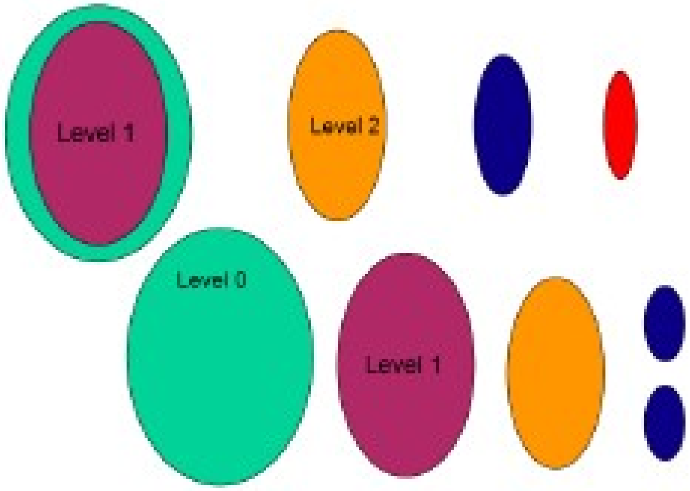
Choose
N threshold levels
(N=10 at the moment)
and get N sets of hits



For each set do a
NN clustering
Only in particular
set!!



Single photon



?!

- sophisticated photon ID
- not yet incl. in PFA algorithms

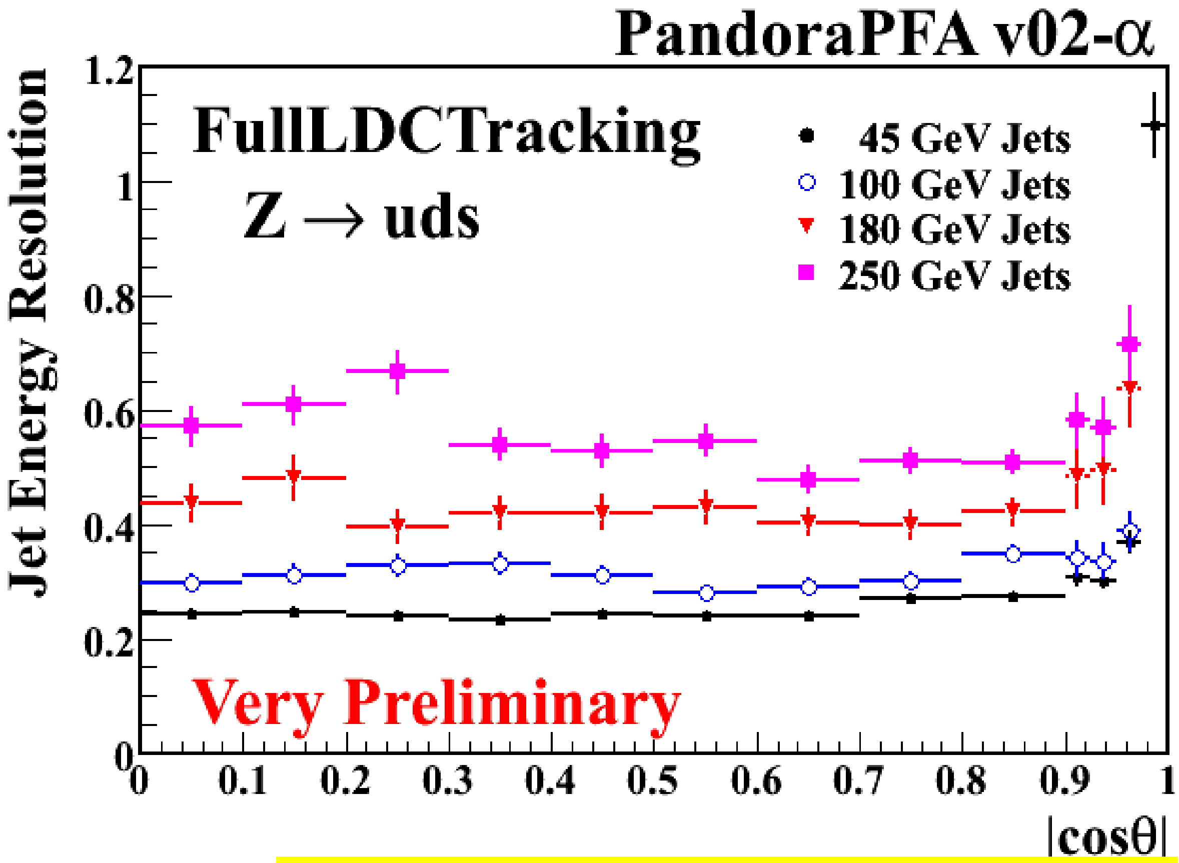
Particle Flow: Performance

Particle flow (PandoraPFA)
including full realistic tracking

Mark Thomson, Cambridge

There is still room
for significant improvement

but performance is
good enough to
start real physics analyses



25%/sqrt(E) at 45 GeV at the moment
including full LDC tracking!

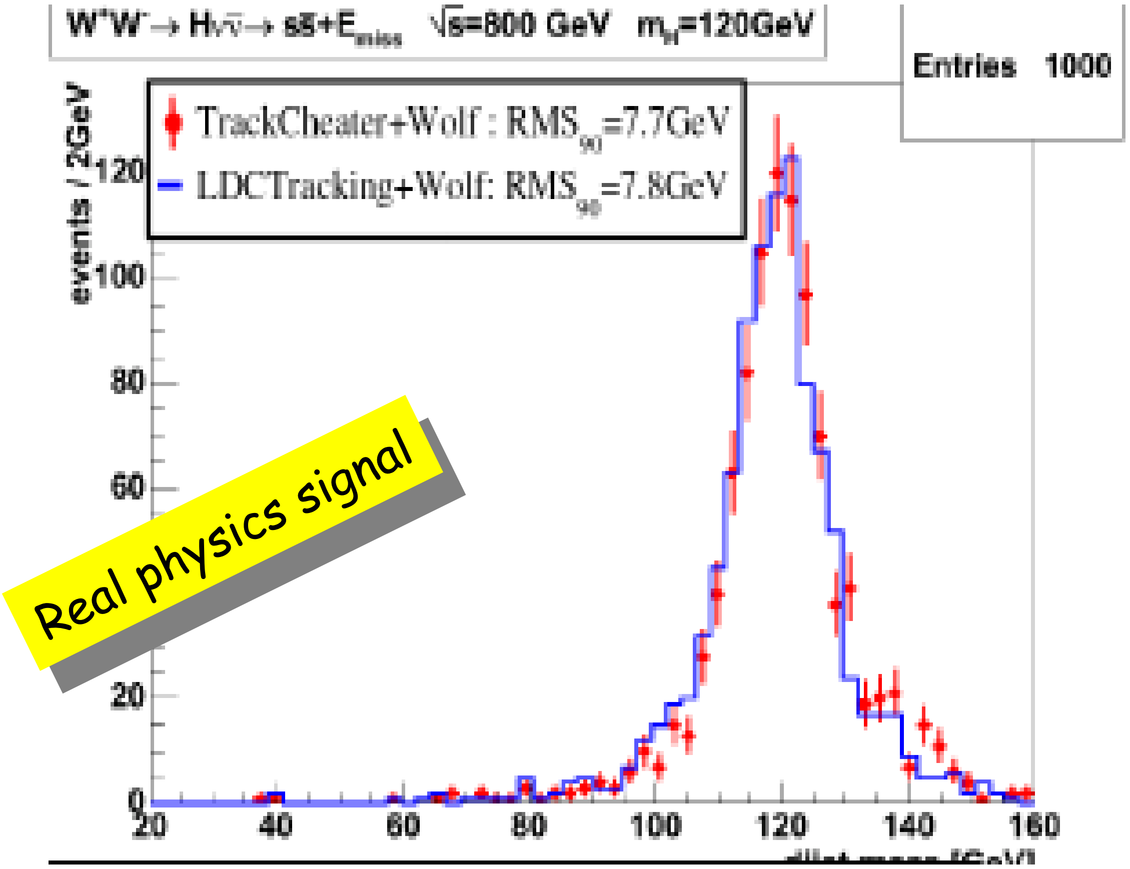
Particle Flow: Performance

Wolf Particle flow
including full realistic tracking

Alexei Rasperieza, Munich

There is still room
for significant improvement

but performance is
good enough to
start real physics analyses



Real physics signal

Plans

Finalize a complete reconstruction chain:

need to improve some digitization routines

Start a large scale simulation and reconstruction run

several 100k events, optimized for detector optimization

This is done within ILD to understand the new ILD concept

Plans: Software

Continue to improve the software framework:

- Improve LCIO data model
- develop “online” LCIO
- Develop level-2 LCIO (more user friendly data model, transient, not persistent
  - (example trajectory class, example extended cluster class))
- improve conditions data base

Performance issue (LCIO)

Persistency layer: stay with SIO? Move to something else (maybe root IO?)

Improve interface for user analysis: n-tuple, root tree support, etc

We think we are in reasonable shape for a LOI focussed simulation and reconstruction effort
and to support the ongoing test beam effort