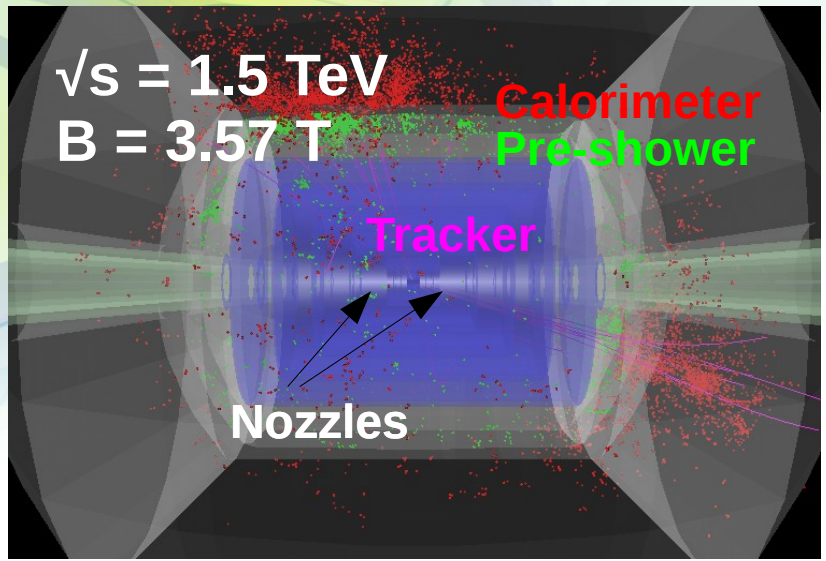
Jet reconstruction at Muon Collider

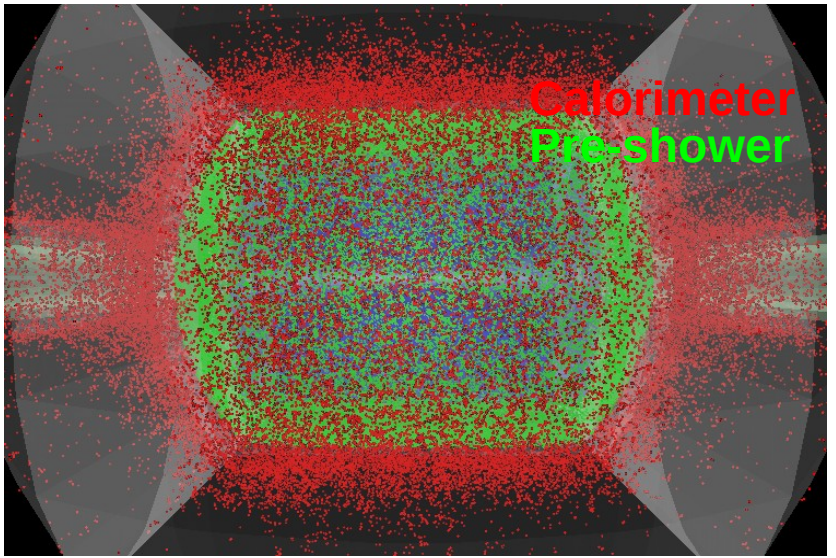
Lorenzo Sestini
INFN-Padova

Workshop on Detector and Physics simulation at a Muon Collider – Padova, 24-01-2020

$H \rightarrow b\bar{b}$ and background $\sqrt{s}$ = 1.5 TeV



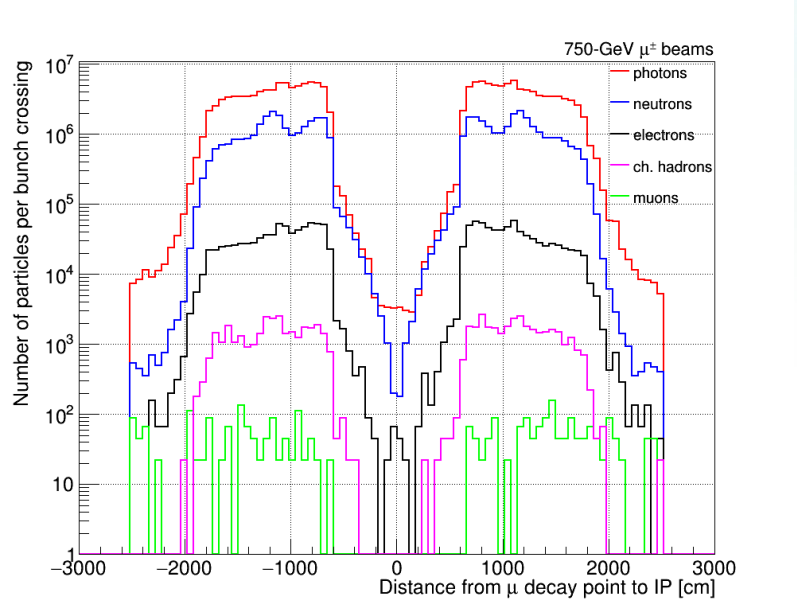
+

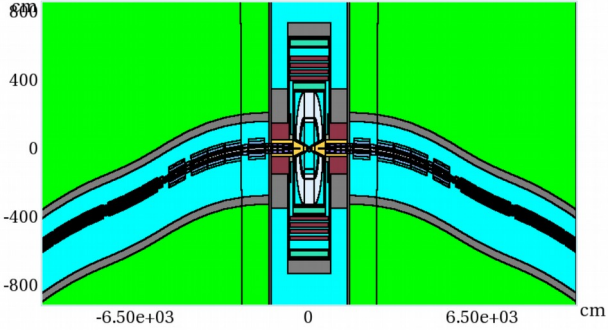


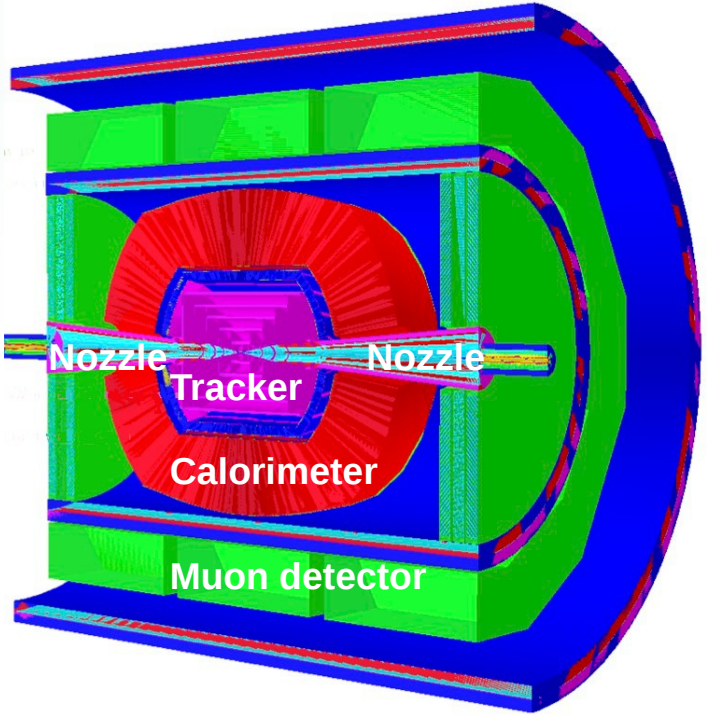
$H \rightarrow b\bar{b}$ signal

Beam-induced background

Background characterization and detector



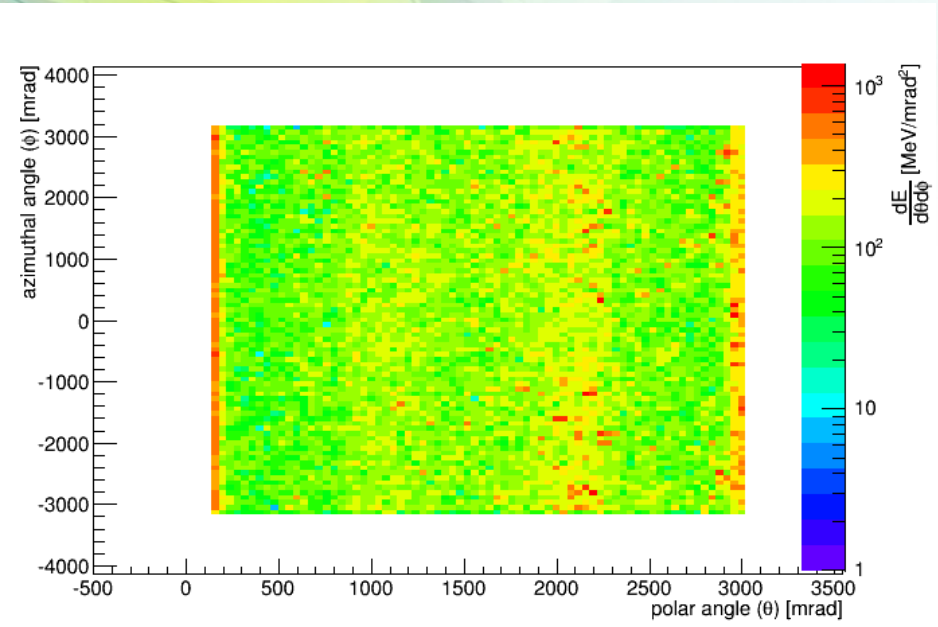




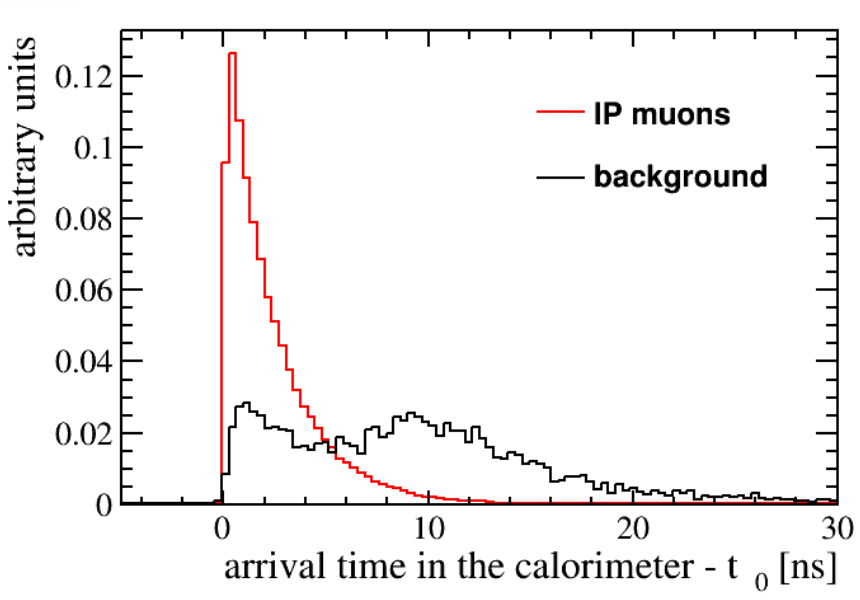
Detector designed by the MAP collaboration

Tungsten nozzles to reduce the electromagnetic background

Background in calorimeter



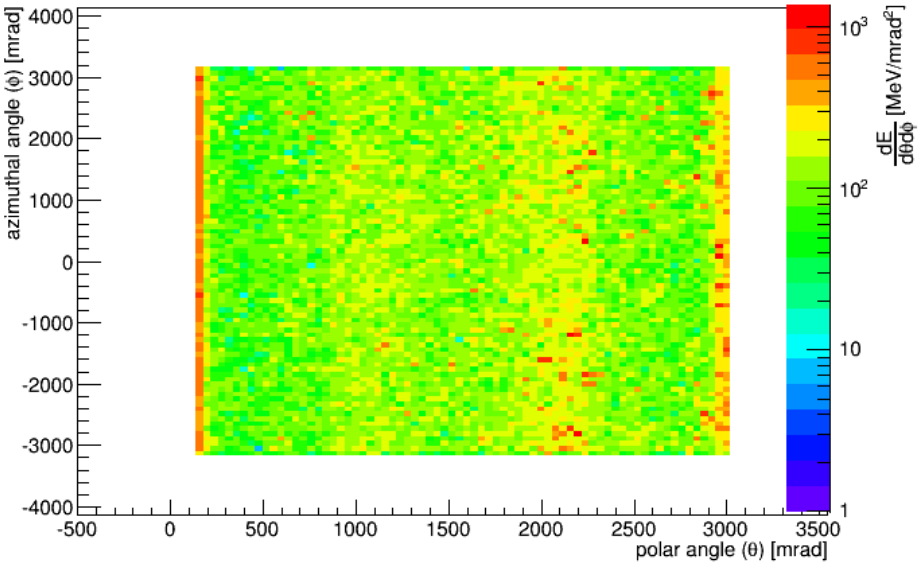
The background is **diffused**



Part of the background is **asynchronous with respect to the signal**

Jet reconstruction algorithm

- **Calorimeter clusters are reconstructed with the algorithm avalilable in ILCroot,** but we put a limit on the cluster size
- **Calorimeter clusters selection**:
  - The calorimeter phase space is divided in several (10) regions of equal pseudorapidity.
  - In each region the mean deposited energy <E> and the standard deviation σ_E are calculated.
  - If the cluster energy satisfies
    E_{cl} - <E> > 2 * σ_E
    then the cluster is used as input to the clustering but with energy E_{cl} - <E>
  - A cone algorithm with R=0.5 is used for clustering.

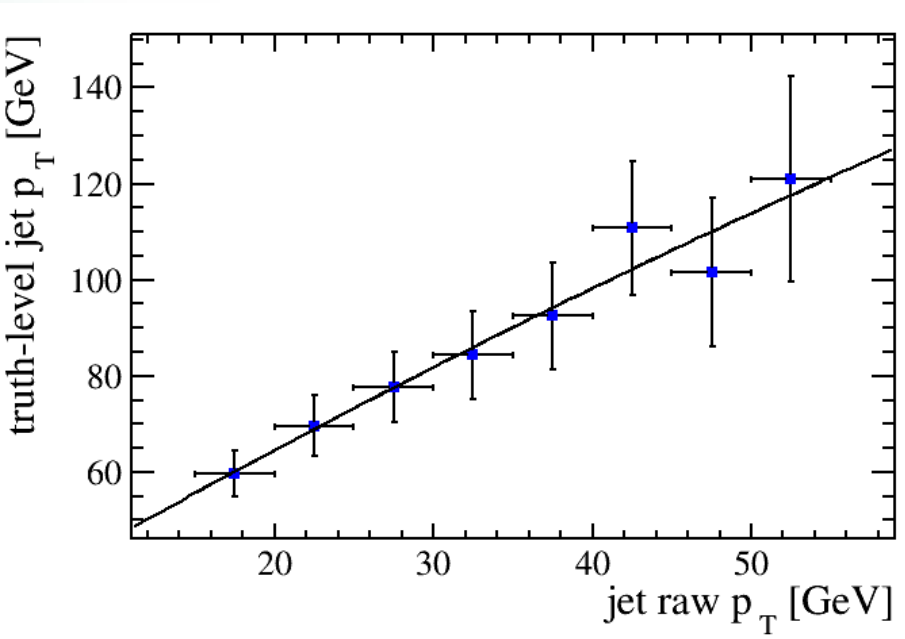


- **Clustering with a cone algorithm**: radius R=0.5, seeds with E_T>2 GeV

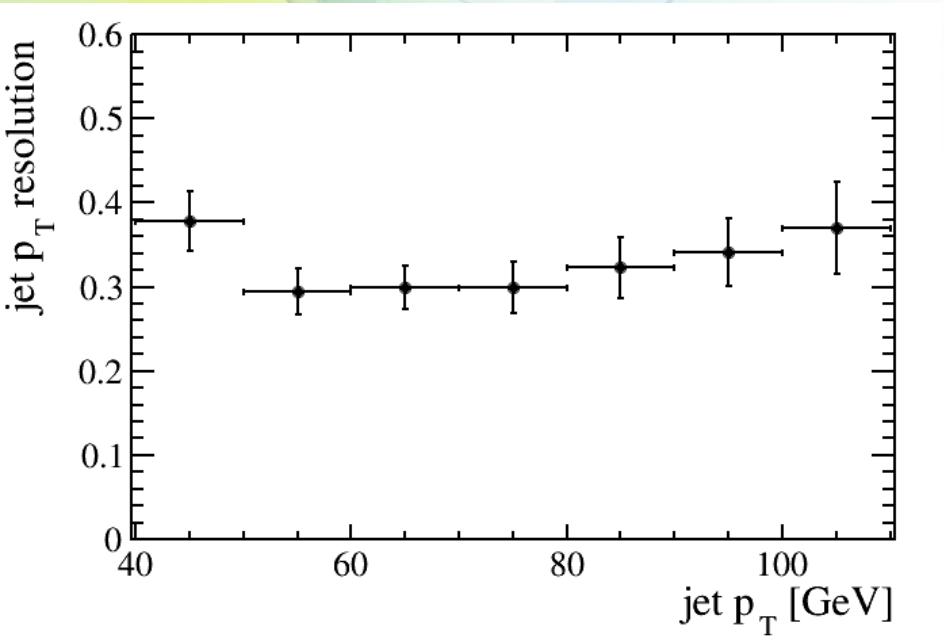
Jet energy correction

- **Truth-level jets are reconstructed,** using the cone algorithm with truth-level stable particles in input
- Reconstructed jets are matched with truth-level jets if ΔR <0.5
- For each interval of truth-level jet p_T, the average of the matched jets p_T is calculated
- **The raw p_T is proportional to the real p_T, therefore a calibration is possible**
- **A fit is performed to obtain the Jet Energy Correction function:**

$$JES(p_T) = \frac{a + b \cdot p_T + c \cdot p_T^2}{p_T}$$



Jet p_T resolution



~30% p_T resolution

Without the beam-induced background the jet resolution is ~20%

Jet reconstruction efficiency

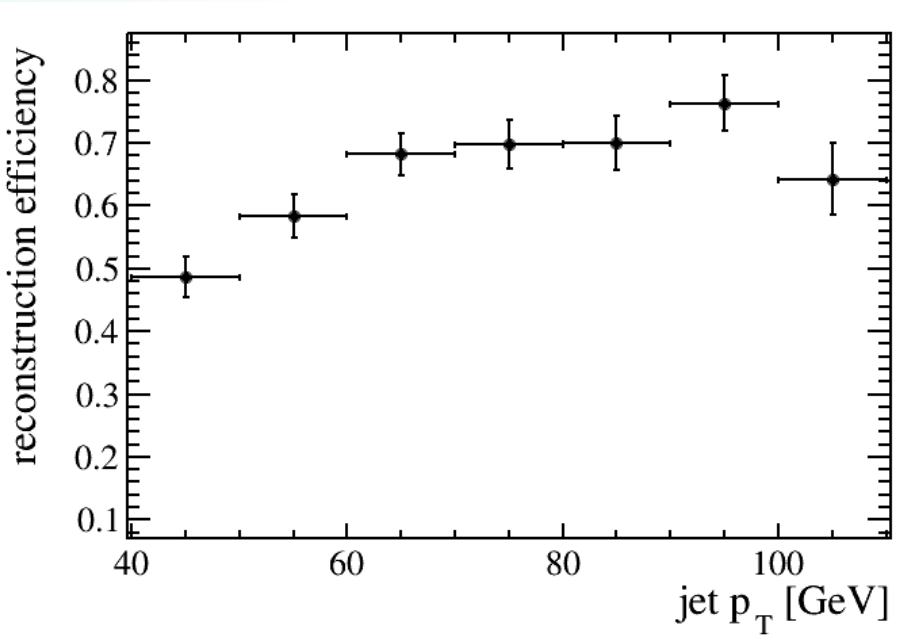
- **Jet reconstruction efficiency:**

$$\epsilon = \frac{N^{reco}_{matched}}{N^{truth}}$$

- **Fake jets rate:**

$$\omega = \frac{N^{reco}_{not-matched}}{N^{reco}_{matched}}$$

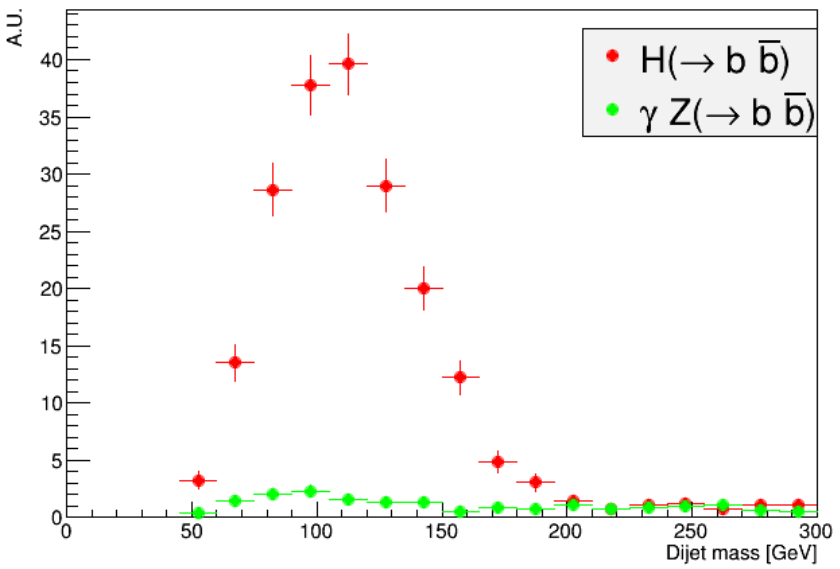
Jet reconstruction efficiency



Fake jets
~25% for p_T > 40 GeV

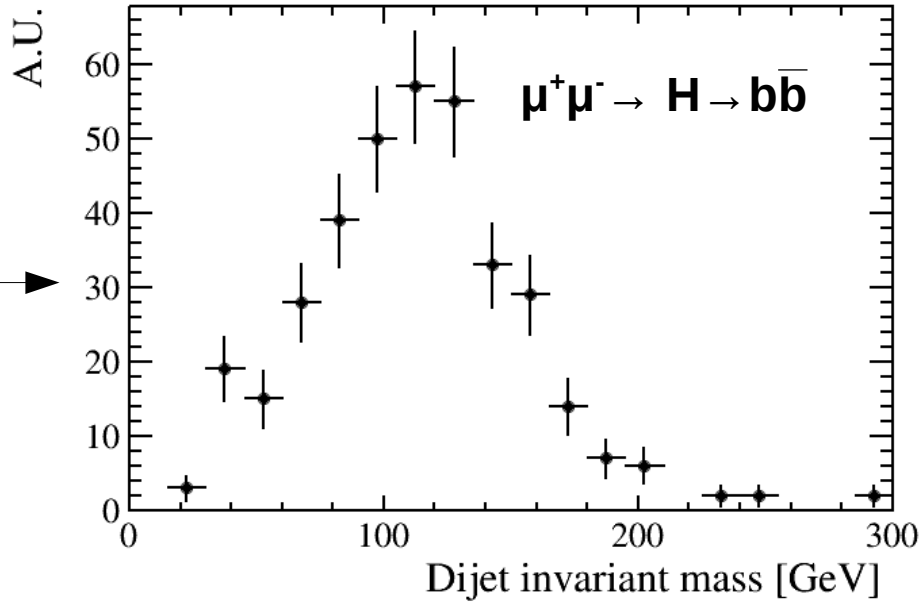
Higgs invariant mass

Without beam-induced background



~30% mass resolution

With beam-induced background



~40% mass resolution

Implementation

- In **ILCroot** a dedicated jet algorithm for Muon Collider didn't exist
- An ordinary cone algorithm is implemented
- We modified the IlcJetFinder class to change the input selection
- JEC has been calculated with a root macro and applied directly in the analysis

```
IlcJetFinder*  ConfigJetAnalysis()
{
    //
    // Configuration goes here
    //
    printf("ConfigJetAnalysis() \n");
    IlcJetESDReaderHeader *jrh = new IlcJetESDReaderHeader();
    jrh->SetComment("Testing");
    //jrh->SetFirstEvent(201);
    //jrh->SetLastEvent(1000);
    jrh->SetPtCut(0.5);
    jrh->SetDetector(0);
    jrh->SetDebug(11);
    jrh->SetFiducialEta(-5,5);
    jrh->SetReadSignalOnly(kFALSE);
    // Define reader and set its header
    IlcJetESDReader *er = new IlcJetESDReader();
    er->SetReaderHeader(jrh);



    // Define jet header
    IlcUA1JetHeaderV1 *jh=new IlcUA1JetHeaderV1();
    jh->SetComment("UA1 jet code with  parameters similar to PYCELL during sim.");
    jh->BackgMode(0);
    jh->SetRadius(0.5);
    jh->SetEtSeed(2);
    jh->SetLegoNbinPhi(432);
    jh->SetLegoNbinEta(274);
    jh->SetLegoEtaMin(-5);
    jh->SetLegoEtaMax(+5);
    jh->SetJetEtaMax(5);
    jh->SetJetEtaMin(-5);
    jh->SetMinJetEt(5.);


    // Define jet finder. Set its header and reader
    jetFinder = new IlcUA1JetFinderV1();
    jetFinder->SetJetHeader(jh);
    jetFinder->SetJetReader(er);
    return jetFinder;
}
```

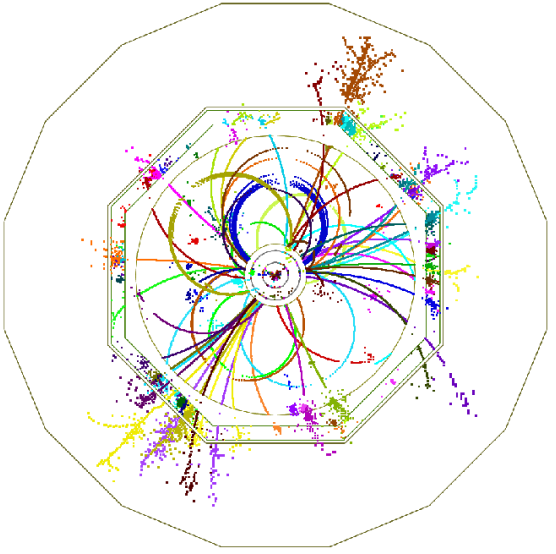
Jet finder configuration

Particle Flow reconstruction at linear colliders

- linear collider detectors are optimized for PFA:
  - high granular calorimeters
  - high hermiticity
  - excellent tracking efficiency

PFA

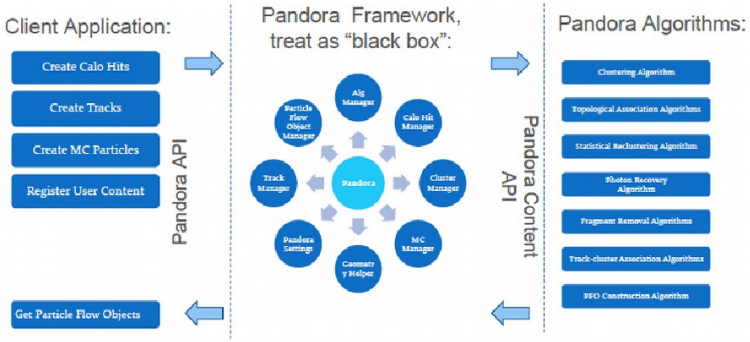
- reconstruct every single particle
- use tracks for all charged particles
- use Ecal for photons
- use Hcal for neutral hadrons



ttbar @ 500 GeV in ILD

PandoraPFA

- generic framework for pattern recognition in calorimeters
  - originally developed for ILC and CLIC (U.Cambridge)
  - glue code for Marlin: **DDMarlinPandora**
- state of the art particle flow algorithm for highly granular calorimeters
- **AIDA2020** project
  - application to LAr-TPC at neutrino experiments
  - application to HL-LHC



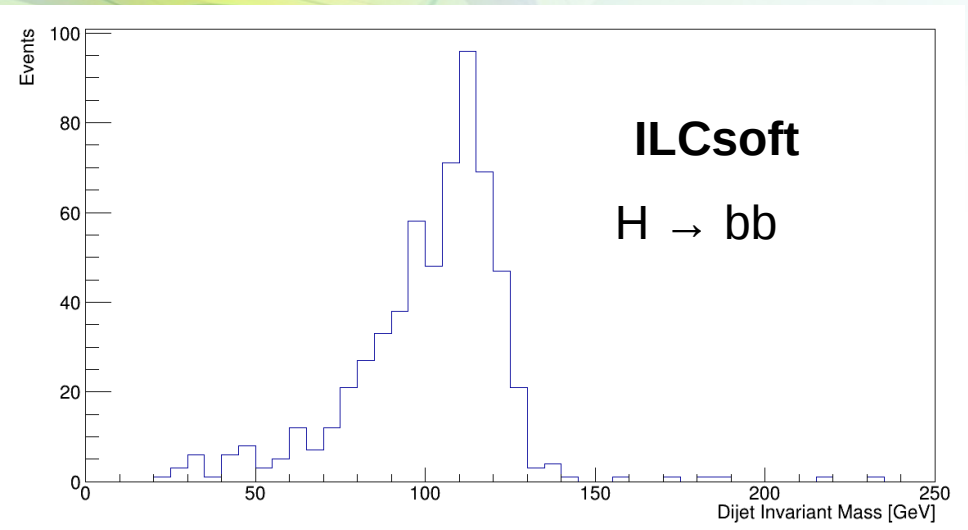
J.Marshall

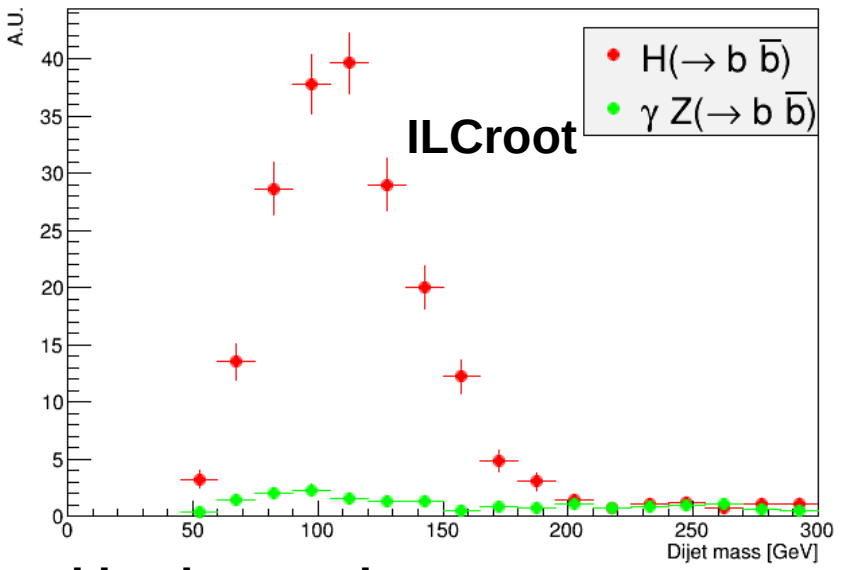
the PandoraSDK provides a nice blue print for how to handle cross-experiment software tools and algorithms

Working plan

- **Study the Pandora PFA for calorimeter clustering**
  - Is it possible to act only on configuration parameters?
  - Do we need to modify the source code?
- **Test the Particle Flow in the MC environment** → Tracking is needed!
- **Re-do the analysis of the jet performance**

ILCsoft vs ILCroot

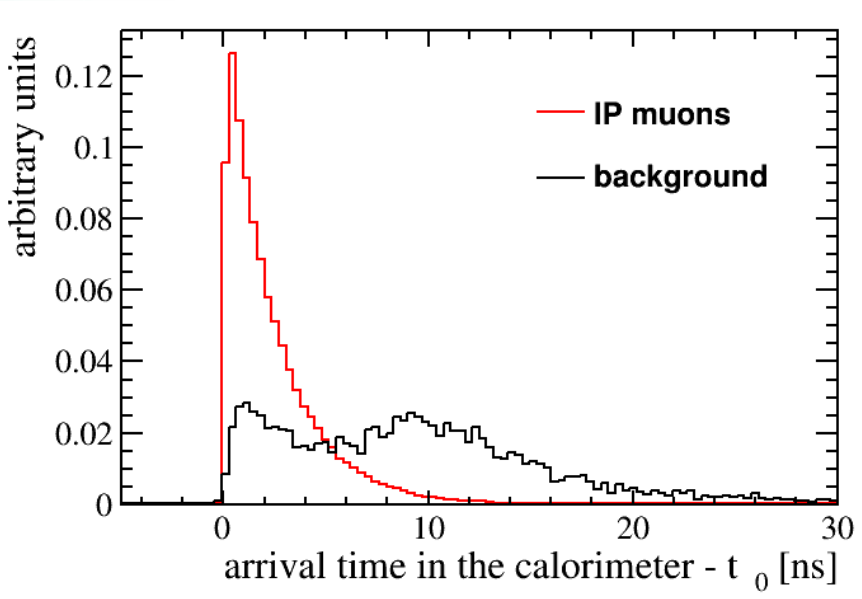




Without beam-induced background

New calorimeters design?

- **Optimized geometry**
- **Test different materials?**
- **Exploit the advantage of a time measurement (how many layers? How to implement?)**



Backup slides