PFA and digital calorimeter (ILD)

Gérald Grenier
IPN Lyon/Université Lyon 1

Calice09 meeting
Kyungpook National University
Daegu, Korea

The environment.

The ILD software installation is done using the ilcinstall package.
http://ilcsoft.desy.de/portal/software_packages/ilcinstall/

Current setup used : ilcinstall v01-06 with differences :
More recent Mokka version, HEAD of PandoraPFA and MarlinReco

Simulation done using Mokka.
Configuration = Steering file + macro file + standard input (at the same time)
Input = MySQL database for geometry.
Output = slcio file for the events and gear file for the geometry.

Reconstruction and PFA done within the Marlin framework.
Configuration = xml file.
Input = slcio file and gear file.
Output = root files, other slcio files.

For the moment, both Mokka and Pandora are mostly 'black boxes'.

The configurations.

Mokka :
•Steering file : 53 lines describing geometry, physics list, GEANT4 configuration.
•Macro file : 9 lines describing process to simulate
•Standard input : one line to exit Mokka

Small and manageable configuration files.
You have to be connected to internet for the MySQL database access
(or have installed the database on your laptop :-( ).

Marlin :
Xml file : 640 lines describing all the marlin processors and their parameters.
Parameters used in more than one processor need to be repeated.

A bit hard to manage :
using a personal version of writesteer.py script (from StandardConfig) to generate the config file of Marlin
http://www-zeuthen.desy.de/lc-cgi-bin/cvsweb.cgi/StandardConfig/scripts/?cvsroot=marlinreco

The geometry.

The gear file is an xml file describing some parameters of the database.
The file by itself is not sufficient to know what is the geometry.

Extract of gear file :

```
<detector name="HcalBarrel" geartype="CalorimeterParameters">
    <layout type="Barrel" symmetry="8" phi0="1.570796327e+00" />
    <dimensions inner_r="2.058000000e+03" outer_z="2.350000000e+03" />
    <layer repeat="48" thickness="2.600000000e+01" absorberThickness="2.000000000e+01"
cellSize0="1.005057471e+01" cellSize1="1.000000000e+01" />
    <parameter name="Hcal_barrel_number_modules" type="int" value="5" />
    <parameter name="N_cells_z" type="int" value="87" />
    <parameter name="Hcal_lateral_structure_thickness" type="double" value="1.000000000e+01" />
    <parameter name="Hcal_modules_gap" type="double" value="2.000000000e+00" />
    <parameter name="Hcal_outer_radius" type="double" value="3.387381227e+03" />
    <parameter name="Hcal_virtual_cell_size" type="double" value="1.000000000e+01" />
    <parameter name="InnerOctoSize" type="double" value="1.704903012e+03" />
    <parameter name="RPC_PadSeparation" type="double" value="5.000000000e-01" />
    <parameter name="TPC_Ecal_Hcal_barrel_halfZ" type="double" value="2.350000000e+03" />
</detector>
```

It's a barrel HCAL but which one ? AHCAL or DHCAL ?

The geometry.

How do PandoraPFA knows the HCAL geometry ?

```
const gear::CalorimeterParameters& pHcalBarrel = Global::GEAR->getHcalBarrelParameters();
....
.....
// set symmetry of HCALBarrel to be the polygon symmetry of outer edge
  try {
    _HCALsymmetry = pHcalBarrel.getIntVal("Hcal_outer_polygon_order");
  }
  catch(gear::UnknownParameterException &e){
    if(_HCALInnerSymmetry*2>_symmetry){
      _HCALsymmetry = _HCALInnerSymmetry*2;
      streamlog_out(WARNING) << "processRunHeader Setting HCAL symmetry at outer edge to "
                                          << _HCALsymmetry << std::endl;
    }
  }
  // Finally check if cyliderical or polygon
  bool foundOuterRadius=false;
  try {
    double rHcal = pHcalBarrel.getDoubleVal("Hcal_outer_radius");
    _outerRadiusOfBarrelHCAL = rHcal;
    foundOuterRadius=true;
    _HCALsymmetry = 0;
    streamlog_out(WARNING) <<
            "processRunHeader found Hcal_outer_radius : treat as DHCAL geometry" << std::endl;
  }
  catch(gear::UnknownParameterException &e){
  }
```

AHCAL geometry

DHCAL geometry

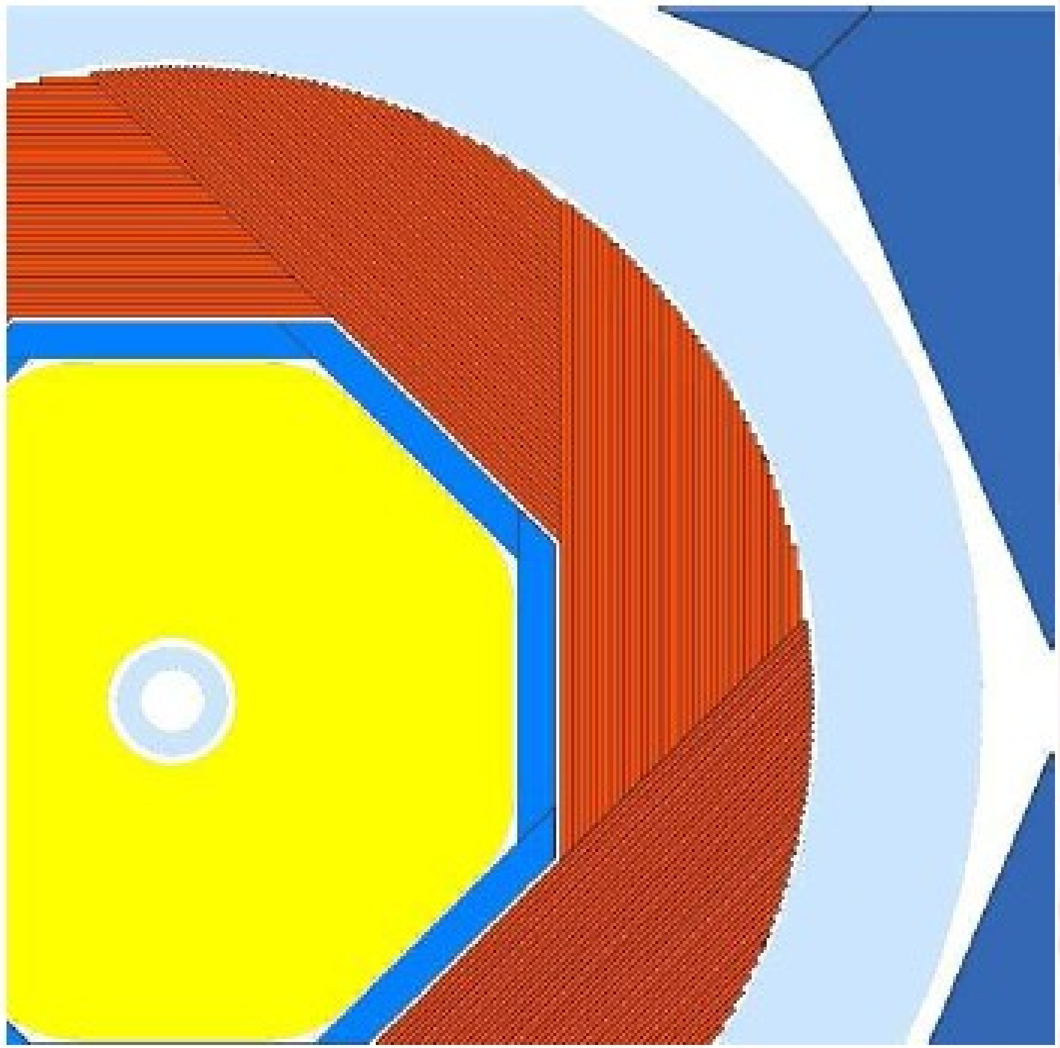
The geometry.

Extract of gear file :


```
<detector name="HcalBarrel" geartype="CalorimeterParameters">
    <layout type="Barrel" symmetry="8" phi0="1.570796327e+00" />
    <dimensions inner_r="2.058000000e+03" outer_z="2.350000000e+03" />
    <layer repeat="48" thickness="2.600000000e+01" absorberThickness="2.000000000e+01"
cellSize0="1.005057471e+01" cellSize1="1.000000000e+01" />
    <parameter name="Hcal_barrel_number_modules" type="int" value="5" />
    <parameter name="N_cells_z" type="int" value="87" />
    <parameter name="Hcal_lateral_structure_thickness" type="double" value="1.000000000e+01" />
    <parameter name="Hcal_modules_gap" type="double" value="2.000000000e+00" />
    <parameter name="Hcal_outer_radius" type="double" value="3.387381227e+03" />
    <parameter name="Hcal_virtual_cell_size" type="double" value="1.000000000e+01" />
    <parameter name="InnerOctoSize" type="double" value="1.704903012e+03" />
    <parameter name="RPC_PadSeparation" type="double" value="5.000000000e-01" />
    <parameter name="TPC_Ecal_Hcal_barrel_halfZ" type="double" value="2.350000000e+03" />
</detector>
```


So it is DHCAL geometry

The geometry.



DHCAL geometry à la Videau is octogonal inside and circular outside.

Parameter name Hcal_outer_radius

It is not yet clear for me if PandoraPFA deals correctly with the outer side of the DHCAL

The HCAL endcap geometry has just been changed but pandoraPFA doesn't know.

Geometry wishes.

I don't know which hypothesis on geometry have been made for PandoraPFA.

However, I would like to be able to use the same reconstruction in full LDC simulation and in test beam data.

The PandoraPFA clustering reconstruct clusters layer by layer.
The fact that the layers are in a cylinder or in a cube shouldn't be a problem.
In practice, the current code can't do it.

Both PandoraPFA and the geometry management need to be updated to be able to do so.

Ideally, the geometry information should be part of the run header (and not provided as a separate file). The geometry information should contain detector shape information (cubic, cylindrical, octogonal, …) in addition to the information contained in the gear file.

Will dive soon into Pandora PFA code to understand how it manages geometry and hit positions.

Marlin setup change.

To study PFA withDHCAL, 2 marlin processors are important
NewLDCCaloDigi (makes hits from hits) and Pandora.

NewLDCCaloDigi parameters :

- **CalibrHCal** **0.0509 0.59**
- **HCALThreshold** **6.5e-8 6.5e-6 (this simulates DHCAL 2 thresholds 0.1 MIP and 10 MIP)**
- **IfDigitalHcal** **1**

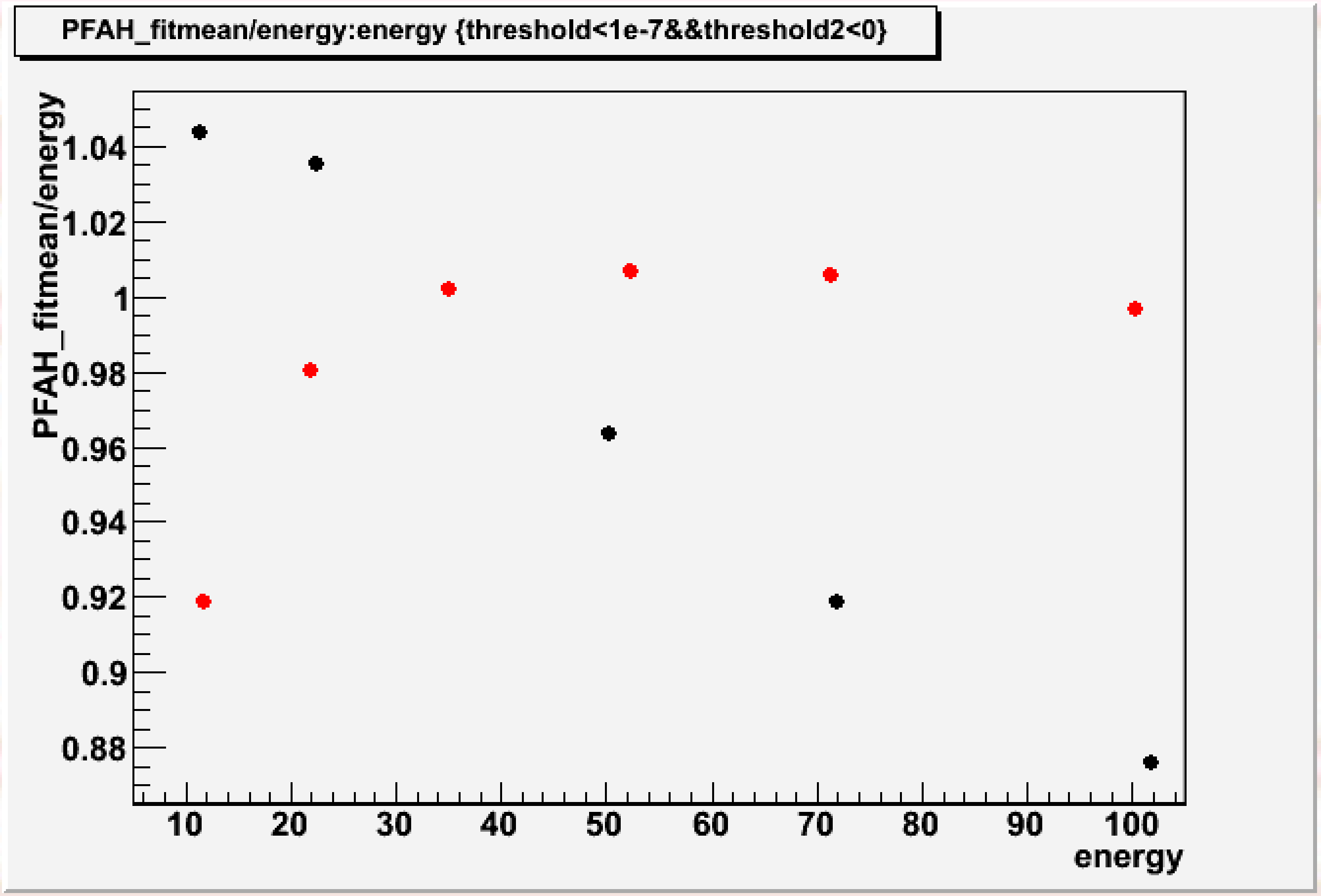
PandoraPFA parameters changes :

- **HCALMIPCalibration** **8.1**
- **HCALEMMIPToGeV** **0.1238**
- **HCALHadMIPToGeV** **0.1238**
- **TypeOfOrderingInLayer** **1**
  (0 = PseudoEnergy,
  1 = density within layer,
  2=local density over layers)

Values in red are correlated and have been tuned for 35 GeV single K_long.
Underlined values have been also tuned on 100 GeV single K_long.

Example of result.

Ratio of reconstructed mean PFA energy over true energy as a function of true energy for mono-kinetic Klong interacting in the HCAL only



Black dots : Single threshold (0.1 MIP) DHCAL

Red dots : 2 thresholds (0.1 MIP and 10 MIP) DHCAL

Plot done with an extra program processing root files produced by PandoraPFA