ILCroot: Infrastructure for Large Colliders based on root (and add-ons for Muon Collider studies)

C. Gatto
INFN

A WORKING GROUP ON SIMULATION STUDIES FOR THE FEASIBILITY OF AN EXPERIMENT AT A MUON COLLIDER

- Should be based on:
  - A time frame of 1.5-2 years
  - Tools already existing
  - The time necessary for the accelerator people and the detector people to implement new configurations in the simulation
  - The goal of preparing a “Yellow Report”-like document
  - The people who have expressed their interest in such project

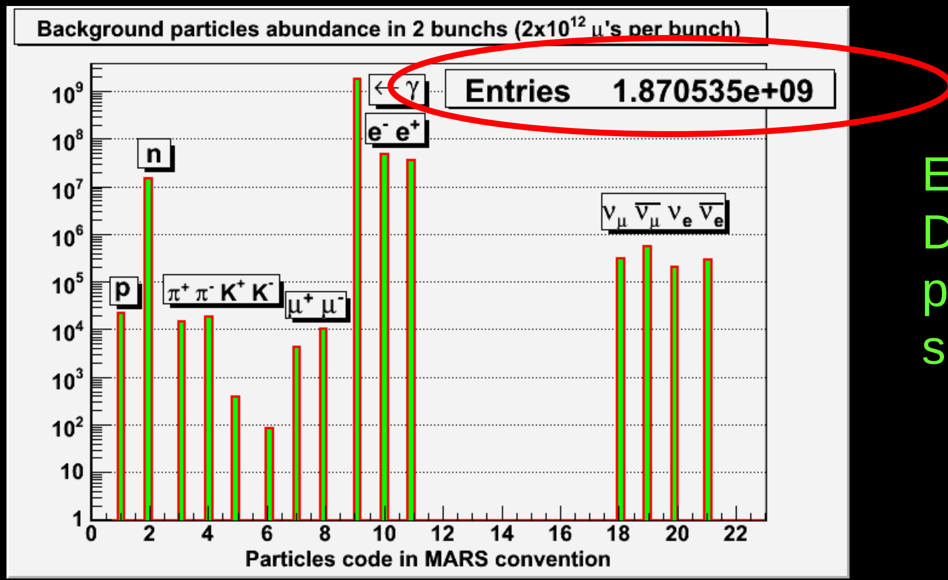
Three Phases Project

- Phase I: Tools preparation
  - Preparation of a list of few benchmark processes
  - A stable machine lattice configuration + Preliminary design of forward shielding (”cone”)
  - Simulation of detectors for subsequent phases

- Phase II: Optimization of detector and machine lattice wrt background
  - Preliminary plots of observables for the analysis of few Physics Benchmark processes
  - Freeze a lattice configuration for Phase III
  - freeze a detector layout for Phase III

- Phase III: Optimization of detector and forward shielding wrt Physics
  - freeze a forward shielding configuration and a detector layout/technology
  - perform final physics studies with such configuration

Four Relevant Issues Related to the Project



Expected Bkg in the Detector at MDI plane at z= 6m (no shield included)

1. Need full MC (Geant4 or Fluka) for phases II and III
2. A large number of background particles originate outside the detector region
3. MonteCarlo transport must start at an MDI plane usually belonging to the accelerator realm
4. The expected particles to be transported are unmanageable by existing Montecarlos

ILCroot: root Infrastructure for Large Colliders

- **Software architecture based on root, VMC & Aliroot**
  - All ROOT tools are available (I/O, graphics, PROOF, data structure, etc)
  - Extremely large community of users/developers
- **Re-allignement with latest Aliroot version every 1-2 years (v4.17 release)**
- **It is a simulation framework and an Offline Systems:**
  - **Single framework, from generation to reconstruction through simulation. Don't forget analysis!!!**
  - It naturally evolves into the offline systems of your experiment
  - It is immediatly usable for test beams (read stream and static data formats)
  - Six MDC have proven robustness, reliability and portability
- ***It is Publicly available at FNAL on ILCSIM since 2006***

ILCroot: main add-ons to Aliroot

1. Interface to external files from Event Generators in various format (STDHEP, text, ***MARS***, etc.)
2. Standalone VTX track fitter
3. Pattern recognition from VTX (for Si central trackers)
4. Track fitters for different trackers technologies (Si Pixels, Si Strips, Drift Chambers, Straw Tubes, TPC's) and a ombination of them
5. Full simulation of ***Dual Readout calorimeters***
6. Parametric beam background (# integrated bunch crossing chosen at run time)

Very important for detector and Physics studies of New Projects

Growing number of experiments have adopted it: Alice (LHC), Opera (LNGS), (Meg), CMB (GSI), Panda(GSI), 4th Concept, (SiLC ?) and **LHeC**

The Virtual Montecarlo Concept

- Virtual MC provides a virtual interface to Monte Carlo
- It allows to run the same user application with all supported Monte Carlo programs
- The concrete Monte Carlo (Geant3, Geant4, Fluka) is selected and loaded at run time

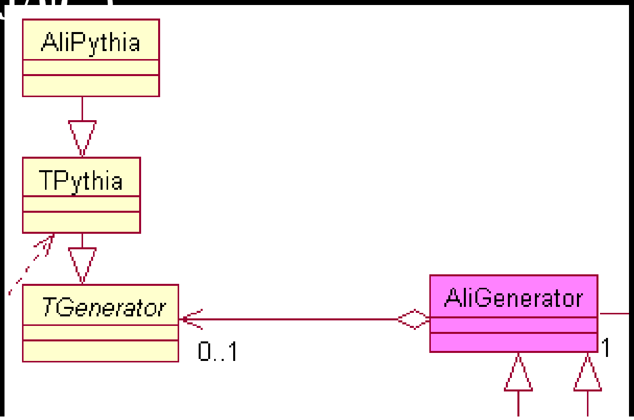
Generator Interface

- ***TGenerator*** is an abstract base class, that defines the interface of ROOT and the various event generators (thanks to inheritance)
- Provide user with
  - Easy and coherent way to study variety of physics signals
  - Testing tools
  - Background studies
- Possibility to study
  - Full events (event by event)
  - Single processes
  - Mixture of both (“Cocktail events”) with weigh
  - Mixture of signal and background
- **TGenerator interface to MARS added for MC studies**

An Example: Pythia and Jetset

- TPythia derived from TGenerator
  - Access to Pythia and Jetset common blocks via class methods
  - implements TGenerator methods

- MyPythia derived from TPythia
  - High level interface to Jetset and Pythia
  - Tailored to our special needs:
    - generation of hard processes (charm, beauty, J/ψ…)
    - selection of structure function
    - forced decay modes
    - particle decays … and more



One Step Up: GenCocktail

- Generation of Cocktail of different processes
  - Generation from parameterised transverse momentum and rapidity
  - Decays using JETSET
  - Rate and weighting control
  - Allow easy mixing of signal and background

ILCroot Strategy: modularity

The Detector Class

- **Both sensitive modules (detectors) and non-sensitive ones are described by this base class.**
- **This class must support:**
  - Geometry description
  - Event display
  - Simulation by the MC
  - Digitization
  - Pattern recognition
  - Local reconstruction
  - Local PiD
  - Calibration
  - QA
  - Data from the above tasks

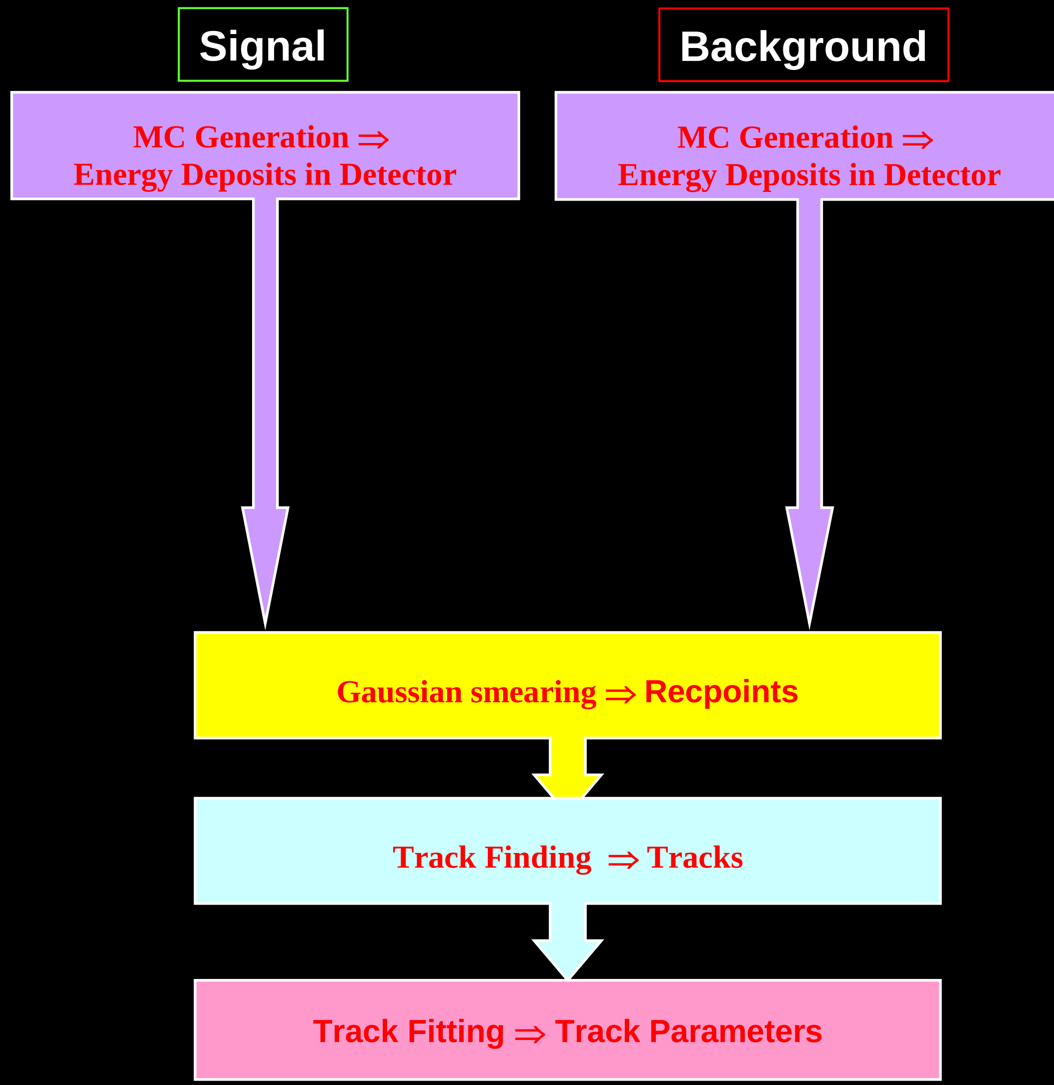
The geometry can be specified using:

- ***Root (TGeo)***
- **Geant3**
- **Geant4**
- **Fluka**
- **GDML**
- **XML**
- **Oracle**
- **CAD (semi-automatic)**

- **Several versions of the same detector are possible (choose at run time)**

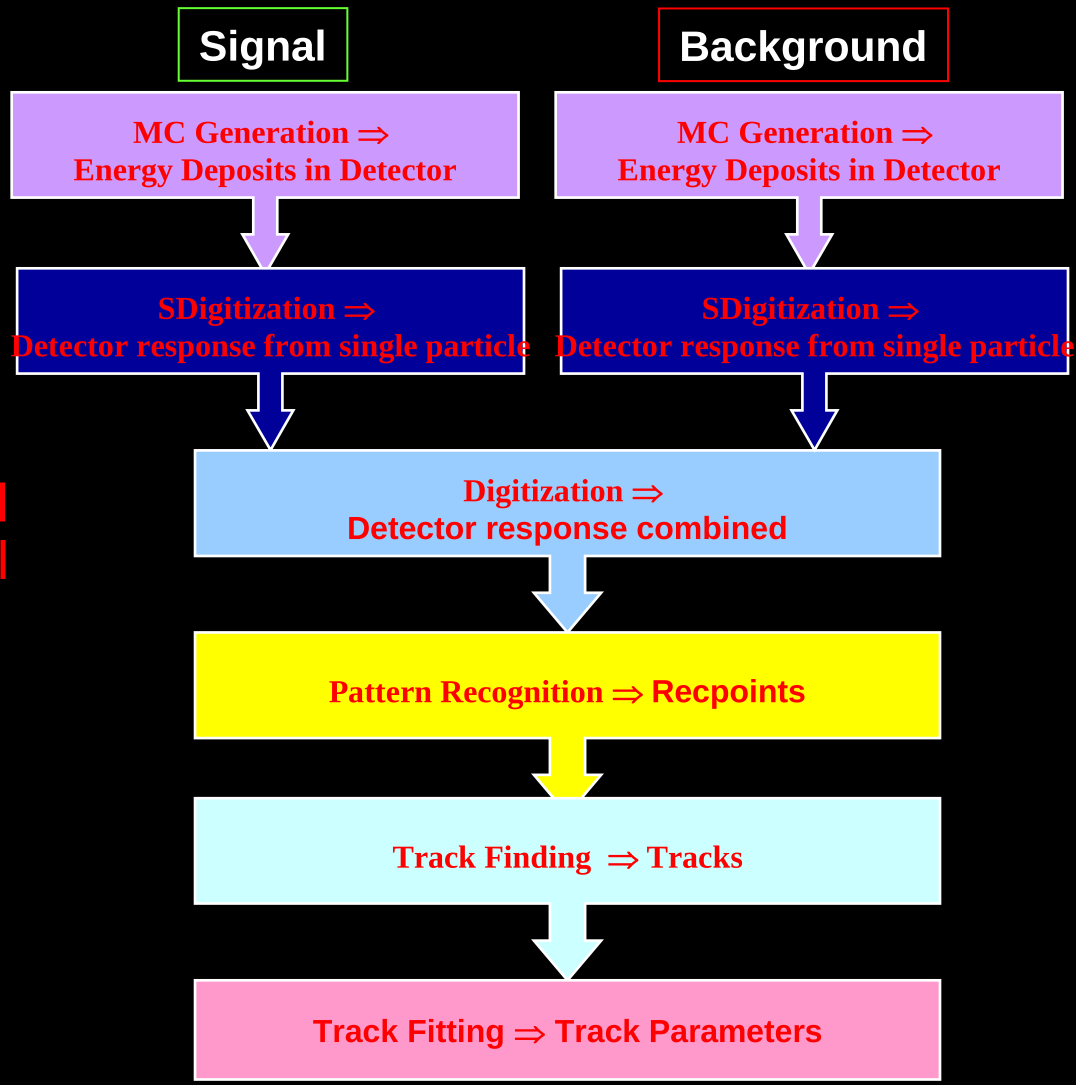
Fast Simulation: no digitization + reconstruction

- Hits: produced by MC (G3,G4,Fluka)
- FastRecpoints: gaussian smearing of hits
- Pattern recognition + track fit through full Parallel Kalman Filter

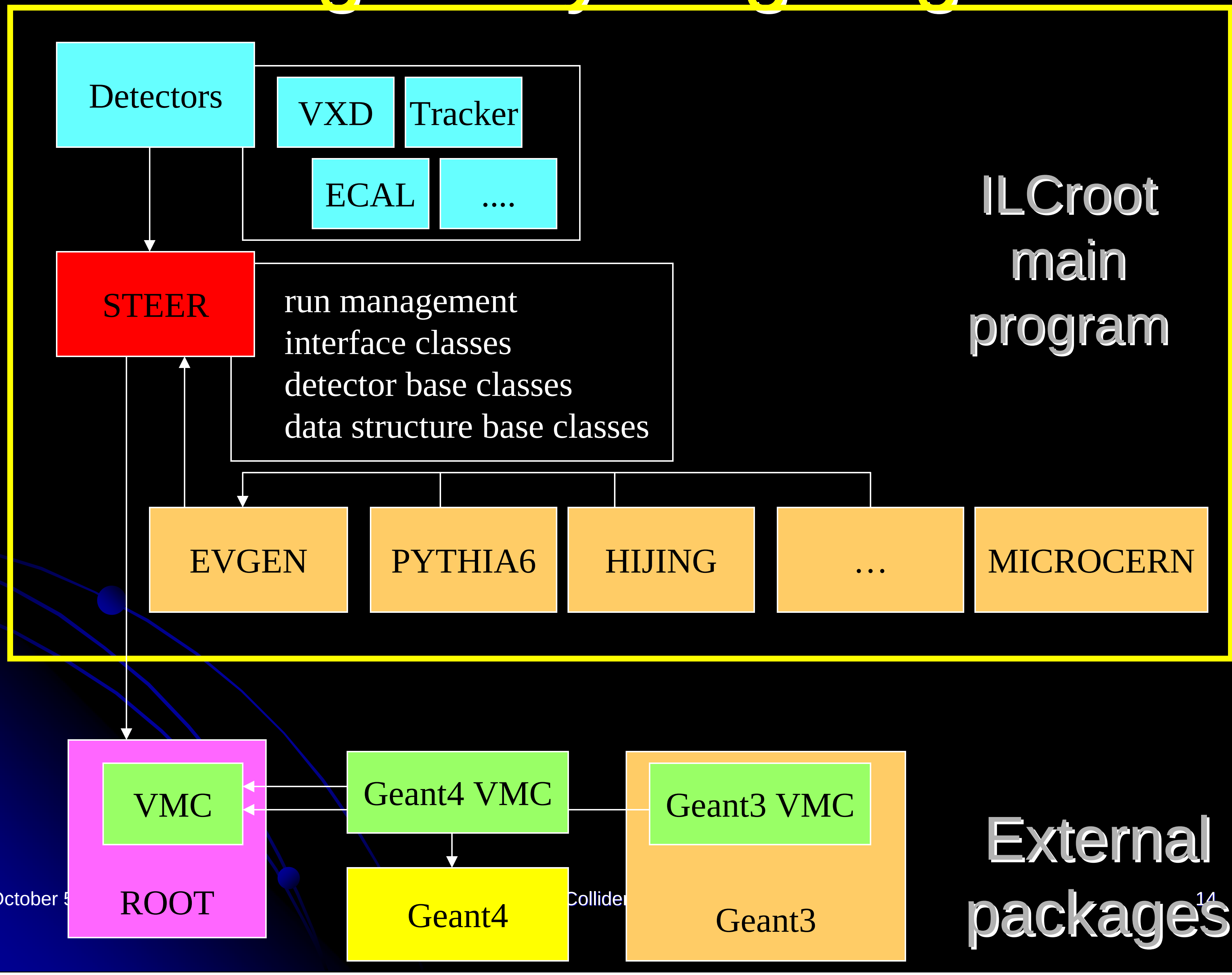


Full Simulation: digitization + reconstruction

- Hits: produced by MC (G3,G4,Fluka)
- SDigits: simulate detector response for each hit
- Digits: merge digit from several files of SDigits (example Signal + Beam Bkgnd)
- Recpoints: Clusterize nearby Digits
- Pattern recognition + track fit through full Parallel Kalman Filter

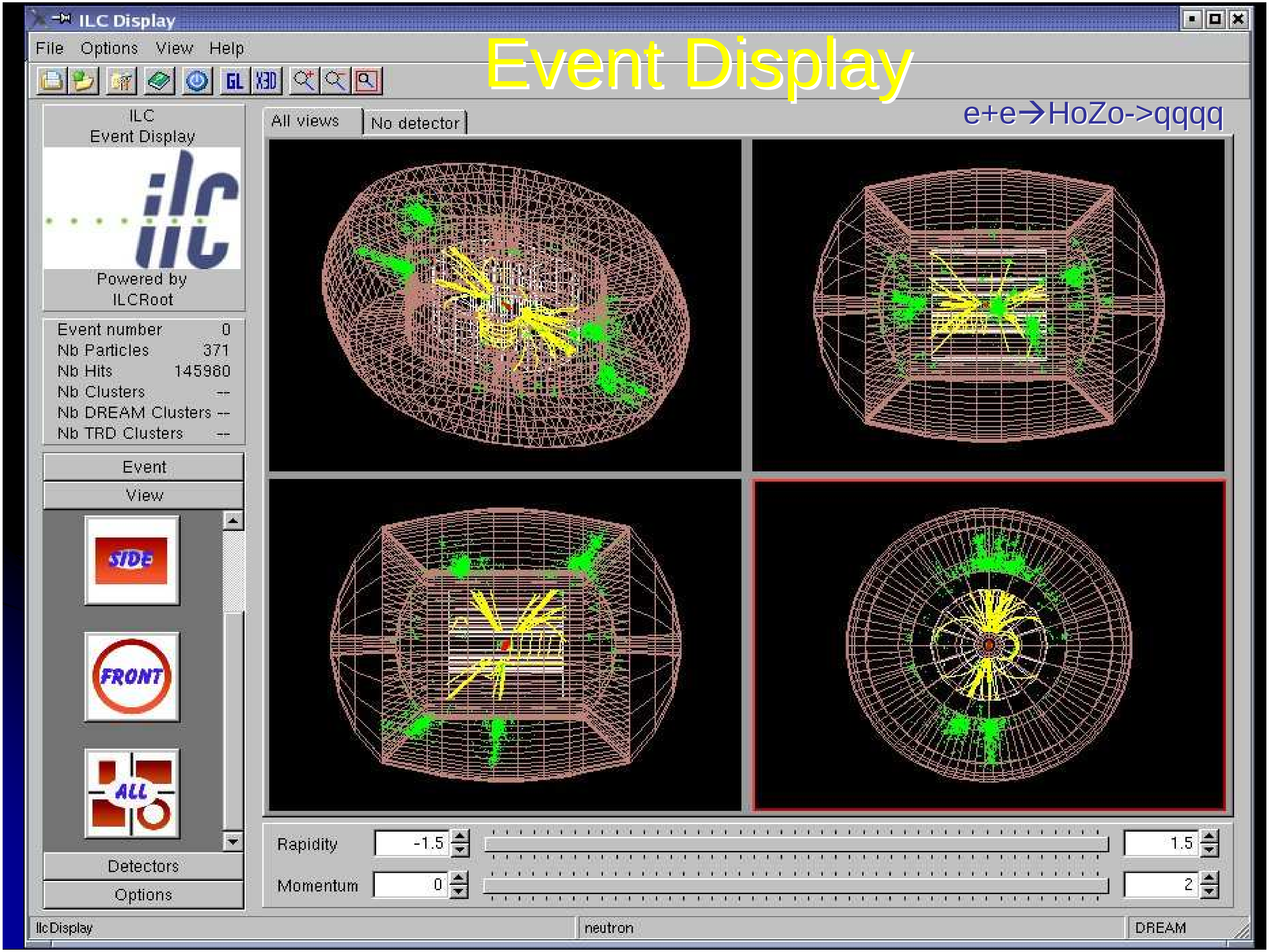


Putting Everything Together



Event Display

e+e→HoZo->qqqq



How long it takes to learn ILCroot?

Physics Analysis Study

- Specify the detector configuration in config.c
- Specify the digitization/clusterization type (full or gaussian smearing)
- Specify the Event Generator (and the channel) in config.c
- Run the job
- Get several .root files for doing your analysis (kinematics, reconstructed particles, trkref, etc.)

Few weeks are sufficient

Detector Simulation/Reconstruction

Expect 1-3 months lead time

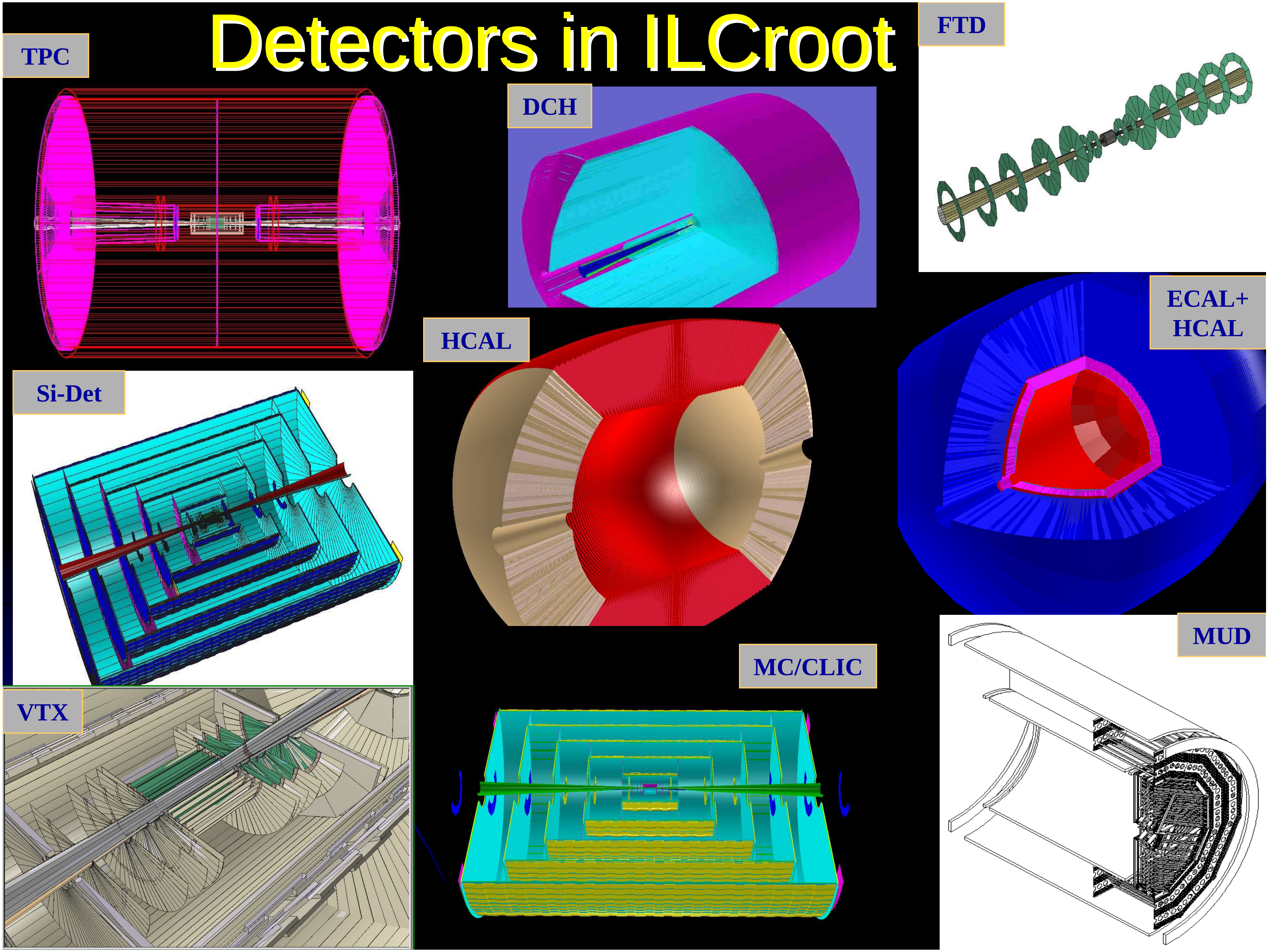
- Learn how the framework is structured
- Learn how to read/write your data into persistent objects
- Eventually, need to learn how to modify the geometry

14 Detectors in ILCroot + Alice

Detector	Layouts	Digit./Cluster.
VXD (SIDMay06)	1 (parametric)	Full
FTD (SiLC)	1	Full
DCH (CluCou)	2	Gauss Smear.
TPC (Hybrid readout)	1	Gauss. Smear.
Si-Tracker (SID01-Polyhedra)	1+1	Full
μCollider/CLIC Tracker	1	Full
Hadron Calorimeter	2	Full
ADRIANO Calorimeter	1	Full
EM Calorimeter	2	Full
Muon Spectrometer (straw tubes)	1	Gauss. Smear.

NEW

Detectors in ILCroot

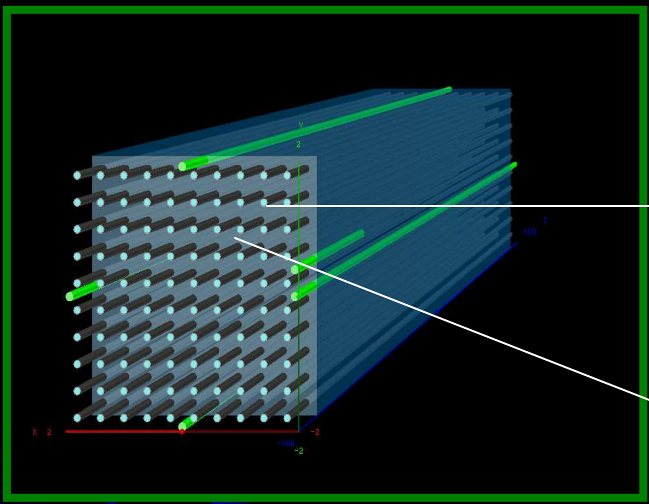


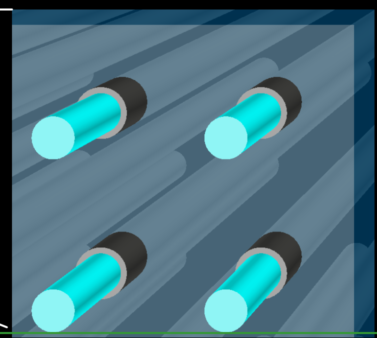
The ADRIANO calorimeter

- Optical SF glass as absorber, active Cerenkov radiator and mechanical structure
- Scintillating fibers matrix in glass structure

ADRIANO cell layout

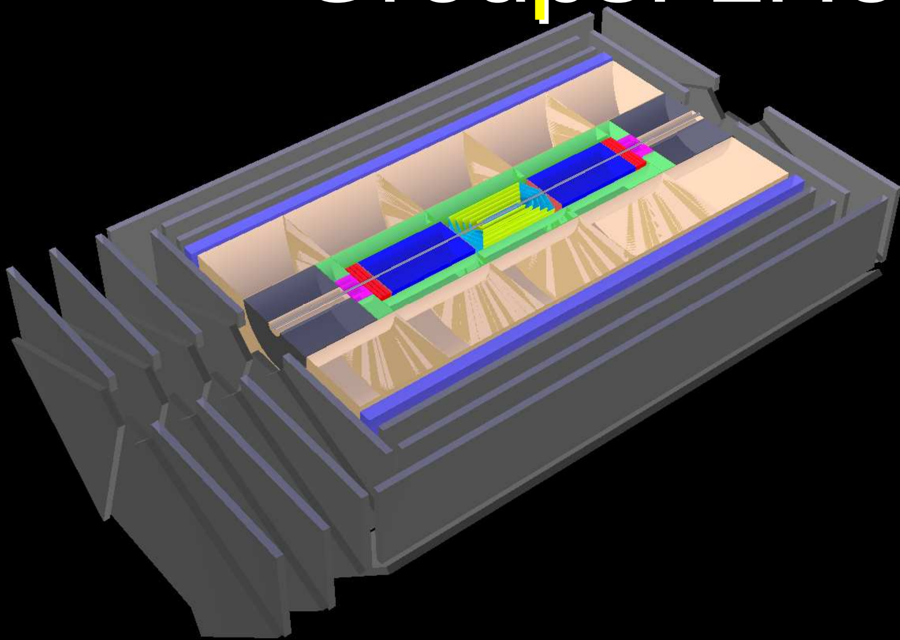
- Fully modular structure
- 2-D with longitudinal shower COG via light division techniques

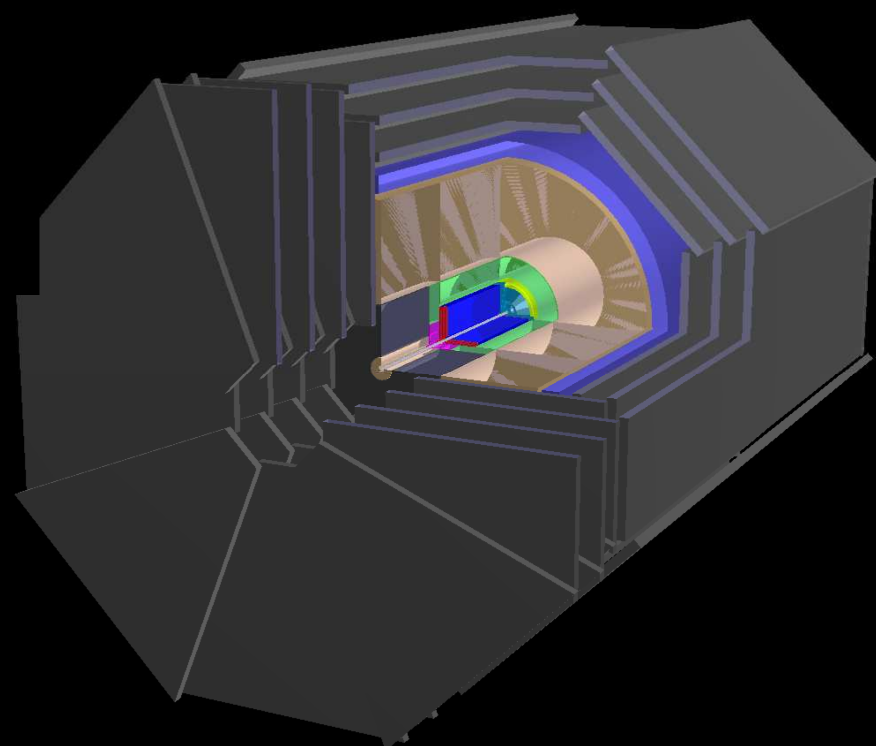




Cells dimensions:	4x4x180 cm^3
Absorber and Cerenkov radiator:	SF57HHT
Cerenkov light collection:	4 BCF92 fiber/cell
Scintillation region: inside 100µm thin steel capillary	SCSF81J fibers, 1mm Φ, 4mm pitch (total 100/cell)
Particle ID:	1 BCF92 fiber/cell (black painted except for foremost 20 cm)
Readout:	front and back SiPM
COG z-measurement:	light division applied to SCSF81J fibers

Sharing Detectors with Other Groups: LHeC in ILCroot





Preparing for a DCR in 2011

- Si VTX detector
- Si tracker
- LAr Calorimeter
- Muon Detector

MARS + ILCroot (Oct. 2009)

- The ingredients:
  - Final Focus descripted in MARS & ILCroot
  - Detector description in ILCroot
  - MARS-to-ILCroot interface (Vito Di Benedetto)

- How it works
  - The interface (ILCGenReaderMARS) is a *TGenerator* in ILCroot
  - MARS output is used as a config file
  - ILCGenReaderMARS create a STDHEP file with a list of particles entering the detector area at z = 7.5m
  - MARS weights are used to generate the particle multiplicity for G4
  - Threshold cuts are specified in Config.C to limit the particle list fed to G4
  - Geant4 takes over at 7.5m
  - Events are finally passed through the usual simulation (G4)-> digitization->reconstruction machinery

Dedicated ILCroot framework for MUX Physics and background studies
(in collaboration with N. Mokhov group) released on Feb. 2010

- Addresses the following issues
  - New "Nose" design
  - Higher Detector Simulation speed
  - More flexible weigth definition in MARS
  - Release of an easy detector configuration for initial Physics & Detector studies by non-experts

Still not the final version for detector studies
OK for Physics studies

Latest MARS interface

- New nose corresponds to minimal design (with correction of old mistakes)
- New definition of weights (N. Mokhov)
- Fine tuning of weight definition by separating muon decay from EM interactions (N. Mokhov)
- ILCGenReaderMARS with <u>**double**</u> option:
  - List of individual particles with unitary weight
  - List of weighted particles
- Feb 2010 background events by Vadim with latest E. Gianfelice lattice configuration

For tracking studies

For calorimetric studies

Status of ILCroot for Muon Collider

- Replaced EM and HAD calorimeters with homogeneous lauout mimicing dual-readout calorimeters

$$\sigma_{EM}/E = 2.8\%/\sqrt{E} \oplus 0.2\%/ \quad \sigma_{HAD}/E = 30\%/\sqrt{E} \oplus 1.7\%/$$

- Parametrized VXD description for tracking studies (unfinished)
- Insert quadrupoles for final focus (as MDI plane =7.5m)
- B-field = solenoid + quadrupole
- Faster simulation/reconstruction
- Tutorials for installation/running

Performance of ILCroot for Muon Collider

- Assuming present μColl background at E_{CM}=1.5 TeV and $2x10^{12}$ μ/bunch (no cut-offs)

- $4.5x10^9$ background particles/event (unweighted)
- $4.6x10^6$ background particles/event with <weight> ≈ $9.8x10^2$
- Processing time with 200 CPU on *fermiGrid*: 22 hrs/ evt with $4.6x10^6$ weighted or unweighted particles (no pre-cuts)

Few considerations on the updated framework

- Speed improvement is *O(10)* vs Nov. 2009
- Allows for 10^6-10^7/day background particles fully simulated and reconstructed with G4 in a real detector with 200 CPU
- Sufficient for calorimetric studies (which can digest weighted particles), but NOT for tracking studies
- At least *O(10)* too many particles for tracking studies

First Physics Process being Study in ILCroot

- Production of a single Z_o in a fusion process:

$$\mu^+\mu^- \to \nu_\mu \underline{\nu}_\mu + Z_o \to \text{2-jets}$$

- How well can the invariant mass of the Z_o be reconstructed from its decay into two jets?
- In particular, could the Z_o be distinguished from a $W^\pm$ decaying into two jets in the process

$$\mu^+\mu^- \to \mu^- \underline{\nu}_\mu + W^+$$

if the forward μ^- is not tagged?

- **Vito Di Benedetto Ph.D thesis**

Some optimization issues: background

- Split Background event from MARS into two components:
  1. Source term from 150m to 30m: mostly muons. Unlikely to be changed during the optimization phase of the beam delivery systems.
  2. Source term from 30m to MDI layer (7.5m)

- The two components are merged in ILCroot at run time (pretend they are incorrelated background sources)

Some optimization issues: background

- Need to reduce the event multiplicity by 90%. Two possible alternatives:
  1. Set cut-off values and use particles with weight = 1
     - from external fast-simulation (ex. S. Mrenna)
     - from ILCroot/G4 and force energy deposition within few steps
  2. Use no cut-off and use particles with weight *O(10)*

Some optimization issues: detectors

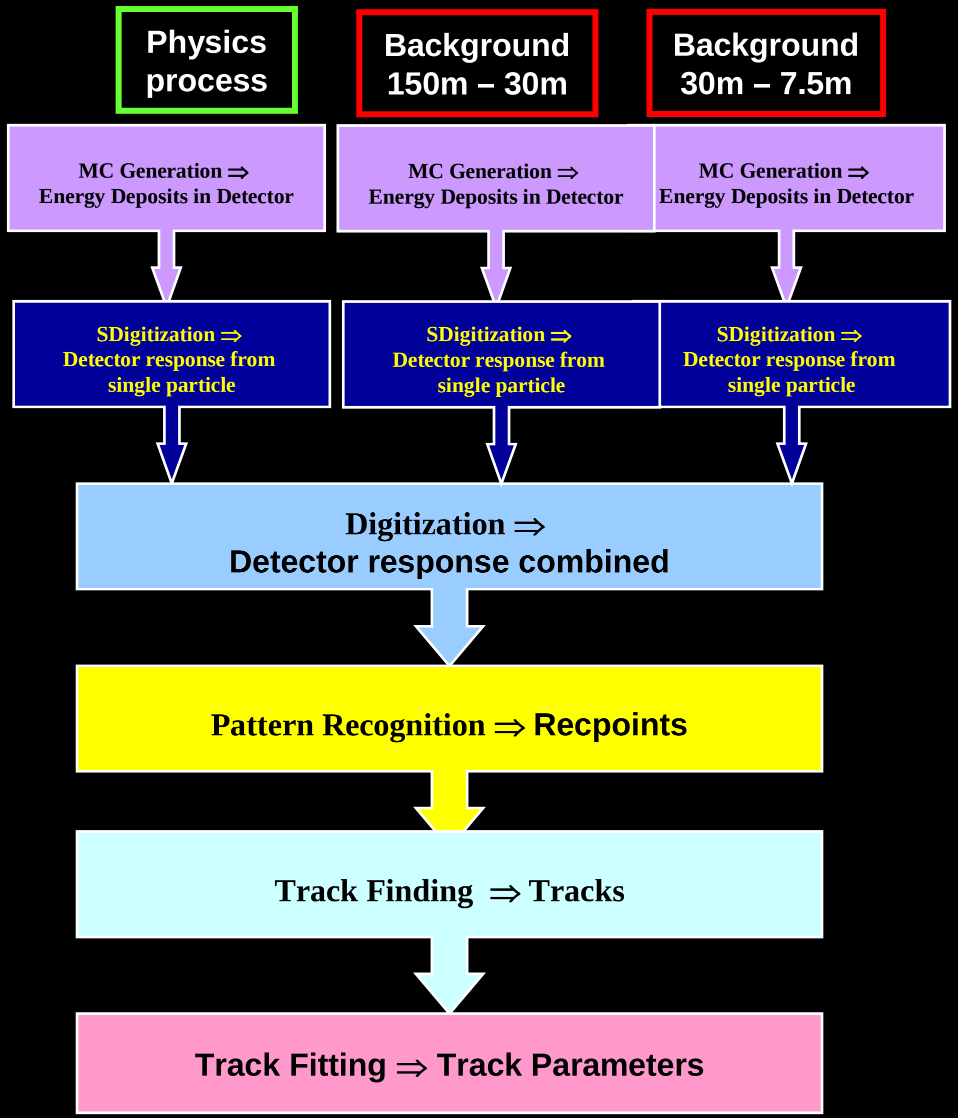
- Phase II would work better with two detector models:
  - Det-V1 for calorimetric studies: uses weighted particles from MARS (already implemented)
  - Det-V2 for tracking studies (occupancy, track separation, resolution, etc.): requires split of MARS weigthed particle into a list of unweighted particles
  - A compromise would be to reduce in ILCGenReaderMARS the weight and increase the particle multiplicity
  - Ex: a particle with weight 200 could be fed to G4 as:
    - 1 particle with weight 200
    - 200 particles, each with weight 1
    - ~20 particles, each with weight 10

(1 particle with weight 200; 200 particles, each with weight 1) → **Already implemented**

(~20 particles, each with weight 10) → **Not implemented**

Simulation + full digitization + reconstruction

1) Merge background with physics events or with jets
2) Reconstruct tracks with proper pattern recognition + Kalman Filter
3) Reconstruct Calorimetric Clusters
4) Study the effect on detector performance and measuremnts of Physics quantity



What's missing in ILCroot

- Complete parametrc description of VXD geometry (M. Peccarisi had to quit)
- Add variable weigth in ILCGenReaderMARS

About 3 weeks

- Implement VXD with two different technologies with fully depleted silicon (and an optional variant):
  1. 10 µm active for MAPS on 50 micron inactive base
  2. 50 µm for 3D - all active
  3. *Variant with double layers separated by D mm (D=100µm, 1 mm) (optional)*
- Define a final calorimeter for Phase III (Det-V3)

R. Lipton recommendation

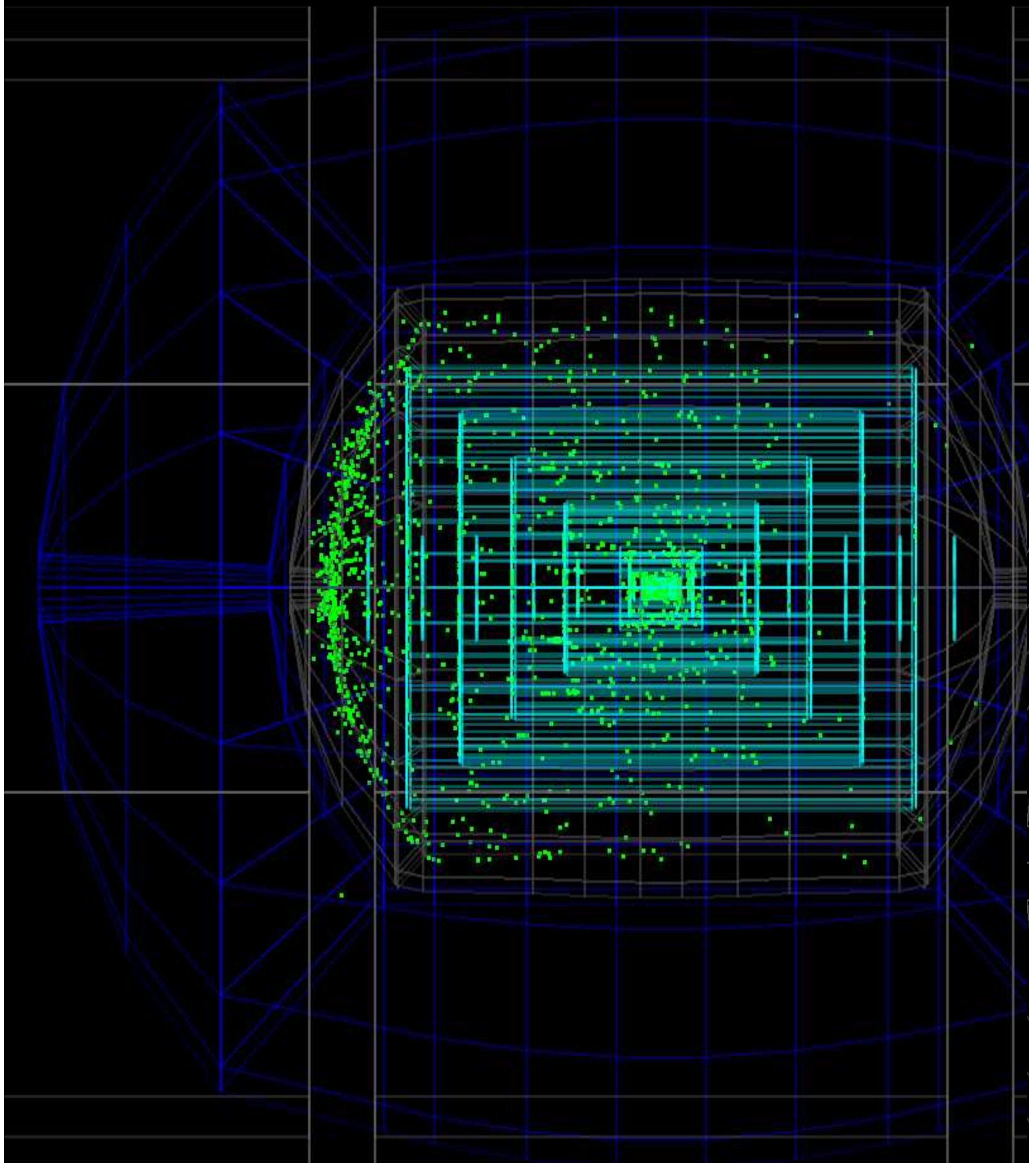
Requires a dedicated person

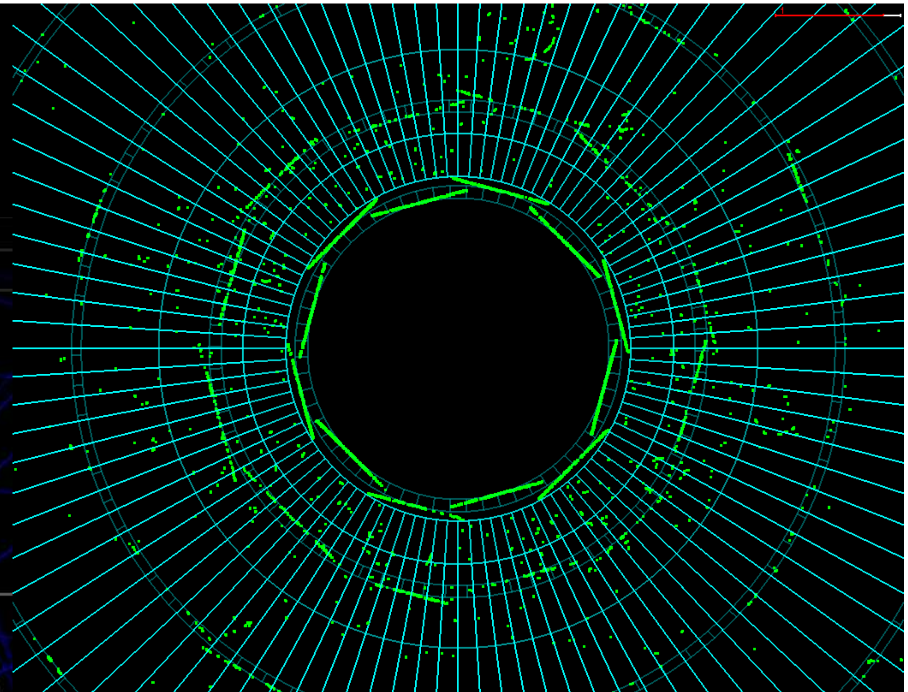
Major effort (~2 months)

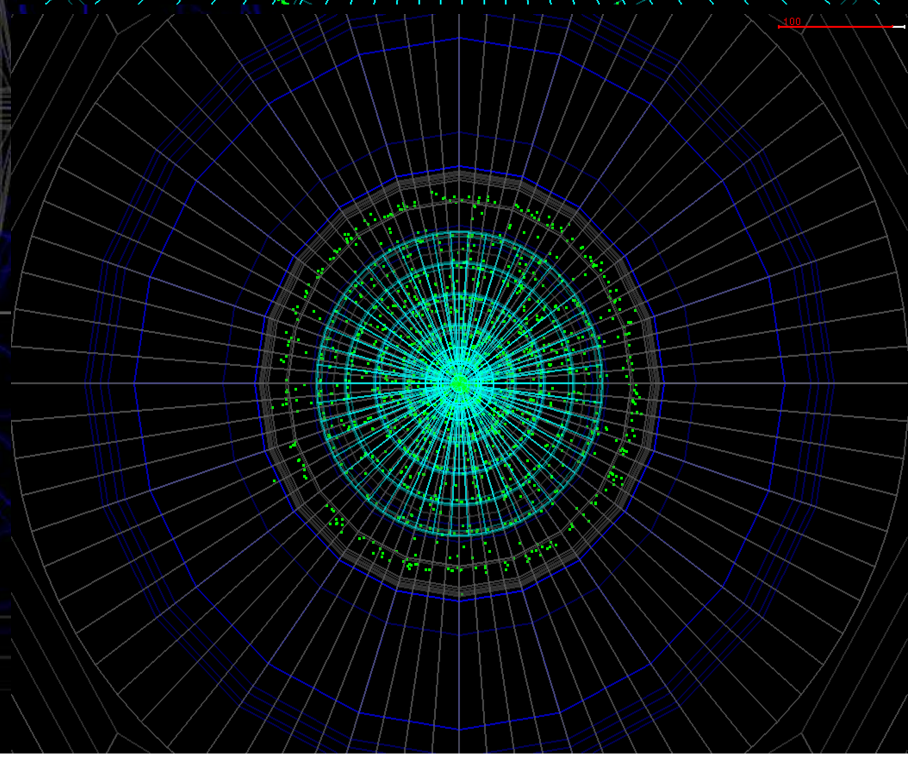
Conclusions

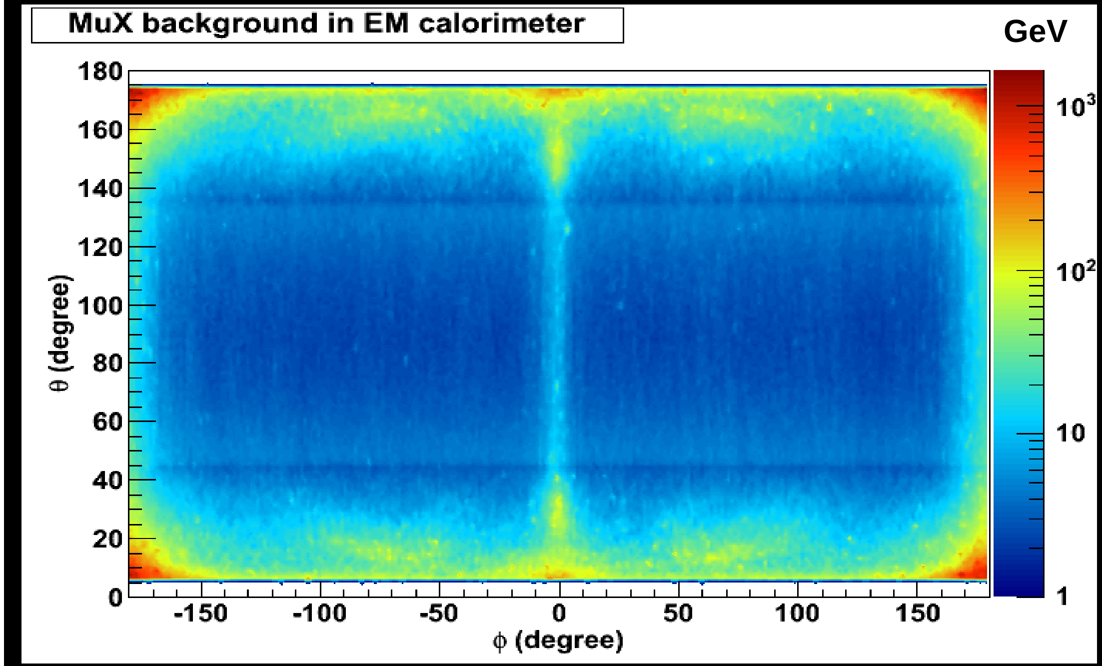
- ILCroot is a root based simulation, reconstruction and analysis framework available at Fermilab on ILCSIM cluster
- It allows merging of signal and background event very easily
- Reconstruction of tracks (**full Parallel Kalman Filter**) and showers in a conventional and/or **Dual Readout Calorimeter**
- It has run flawlessly along the benchmark process for LOI of the 4^{th} Concept collaboration at ILC (200-1000 CPU on Fermi-GRID almost no-stop since August 2008)
- Work on ILCroot specific for Muon Collider started in October 2009
- New interface to MARS is work smoothly for Muon Collider studies
- Ready to be used for a Task Force Simulation Group on Muon collider
- Detector-wise, needs few weeks to complete configuration for Phase I

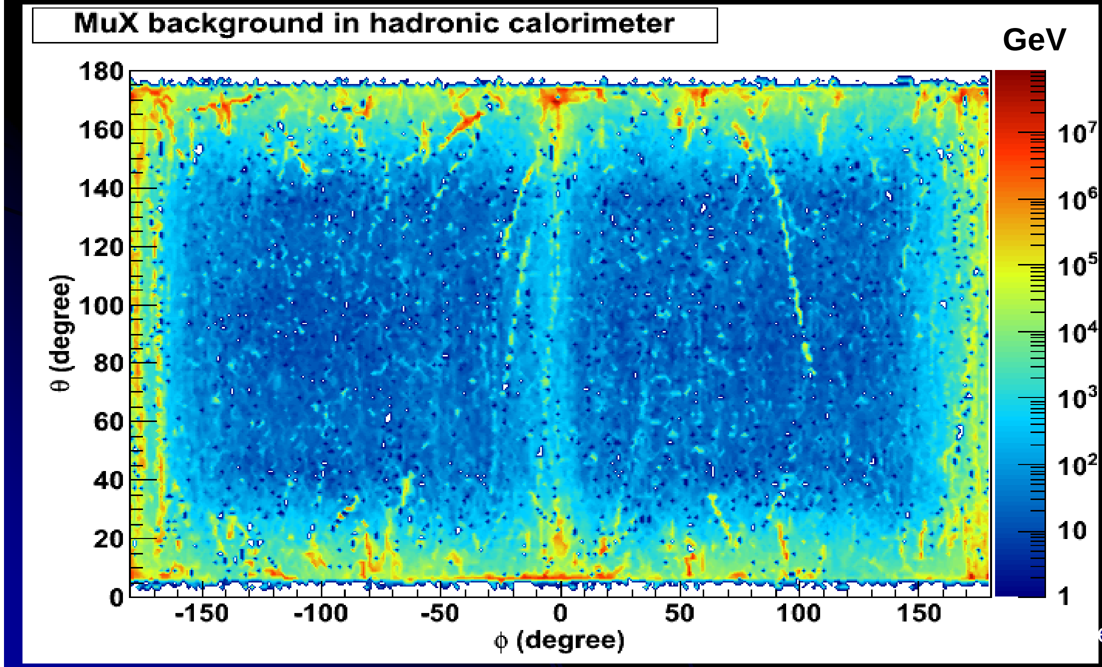
e^- Induced Background at Muon Collider







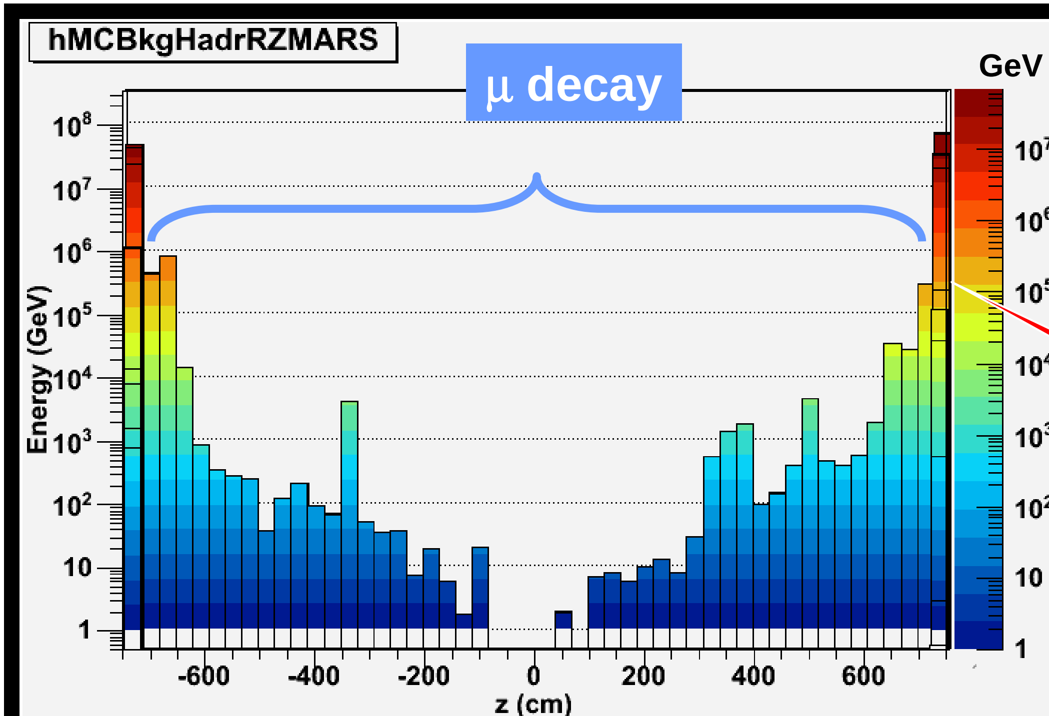




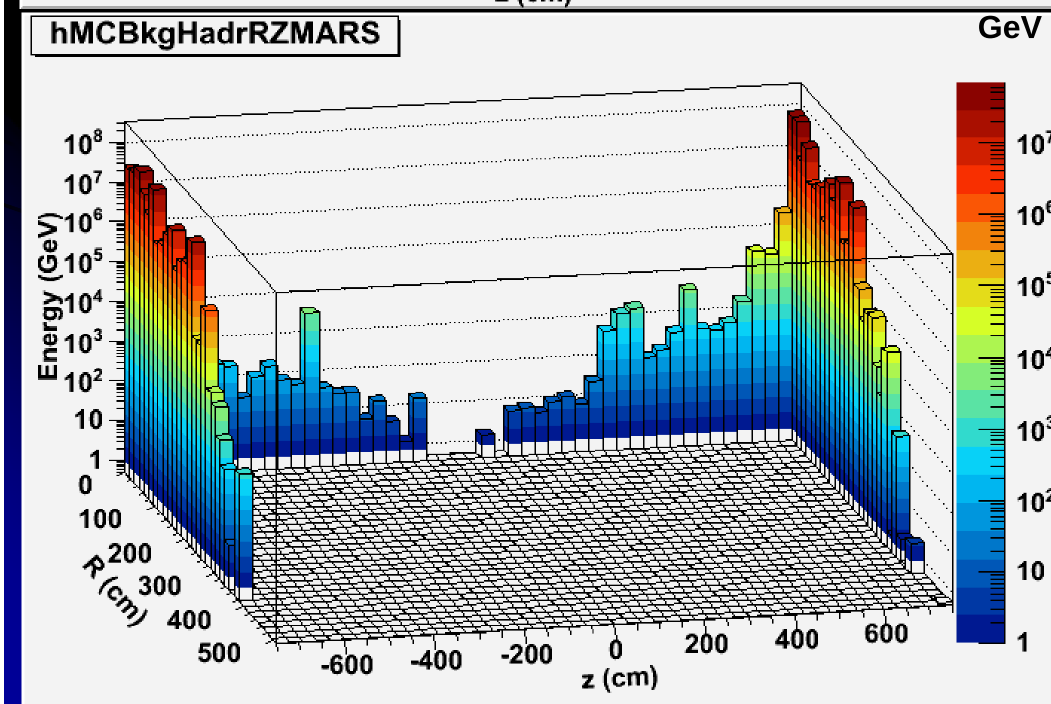
Integrated background in EM and Hadron calorimeters:

- Backround source from 1 collision (MARS - Dec. 2009)
- E_{CM} = 1.5 TeV
- Calorimeter coverage 6°<θ<174°
- Weighted particles method
- MDI separation plane: 7.5 m from I.P.
- No pre-cuts
- Full G4 simulation
- 1 bin = 4x4 cm² cell

Integrated background from MARS in HAD calorimeter vs origin of particle



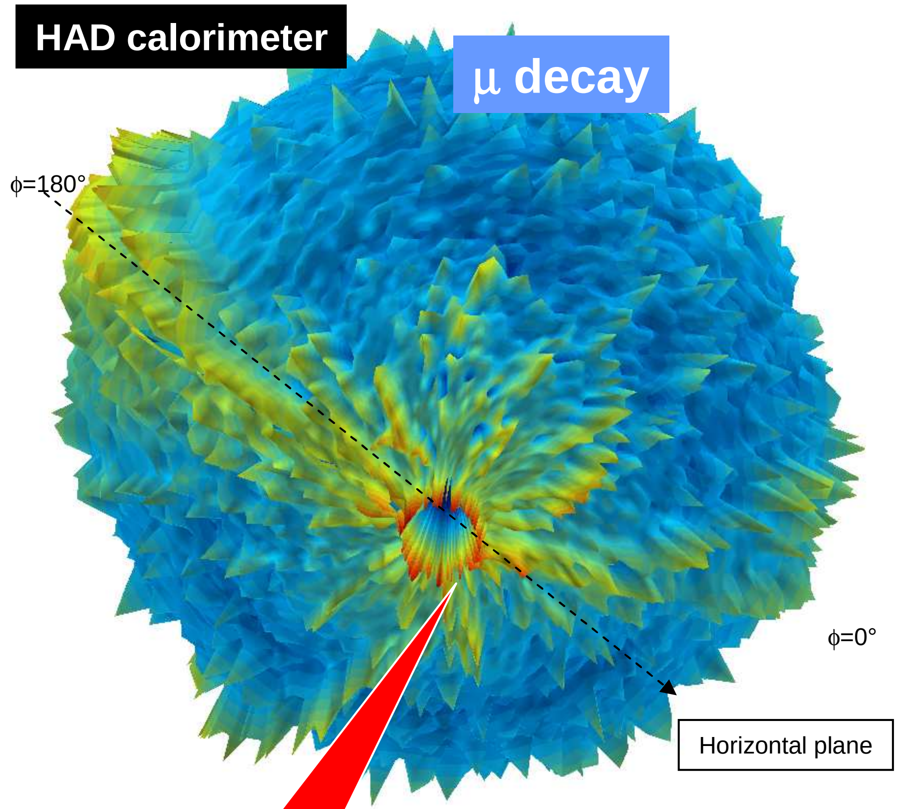
Entering detector area at 7.5 m

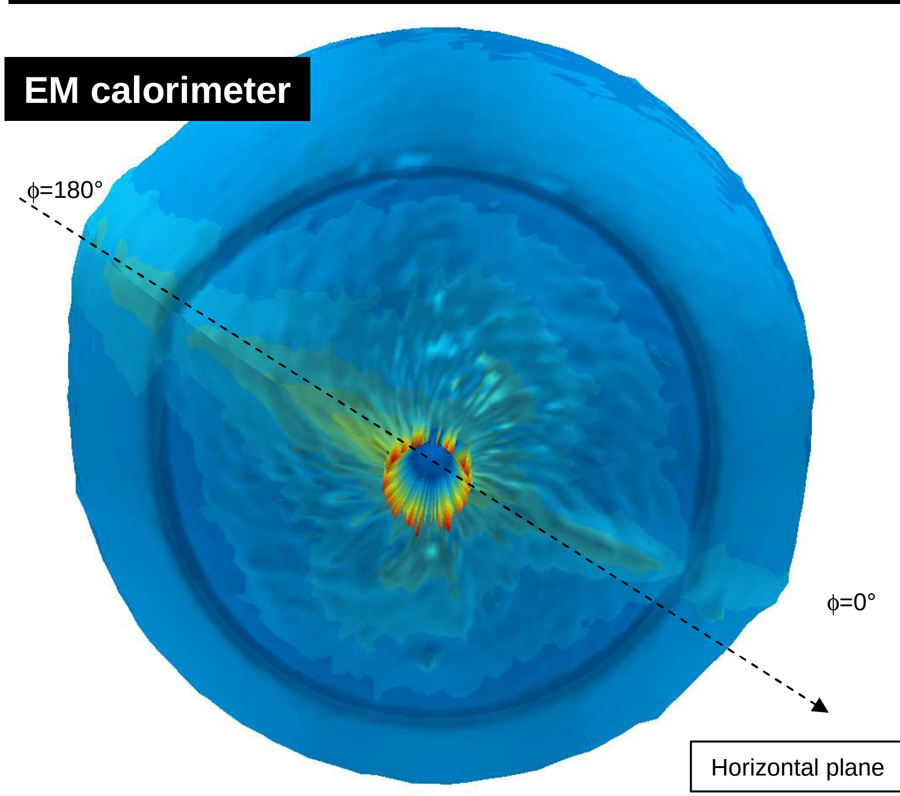


R = r_{xy} of particle origin (1bin= 30cm)

Z=7.5 means that the particle originated outside the MDI separation plane (1bin=16cm)

Integrated background from in EM & HAD calorimeters in spherical coordinates (log scale)

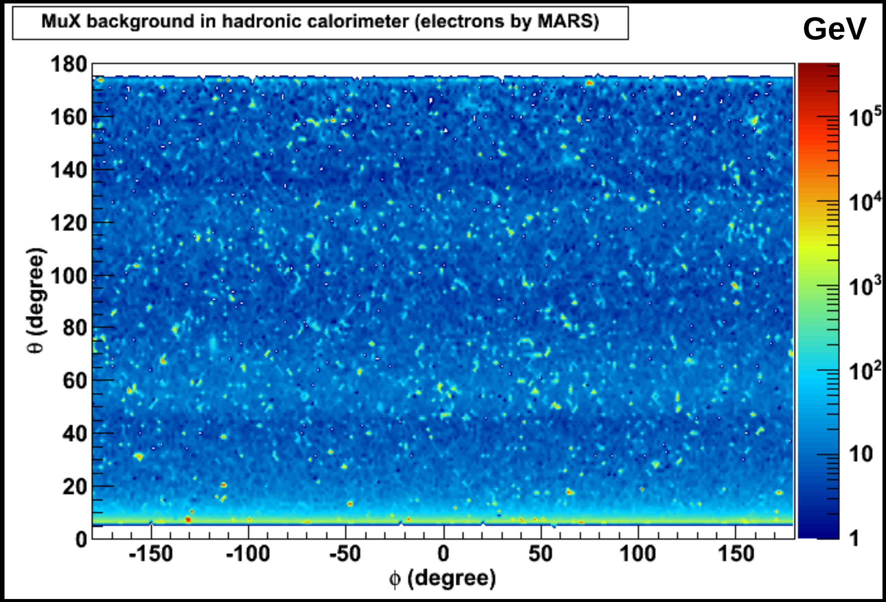


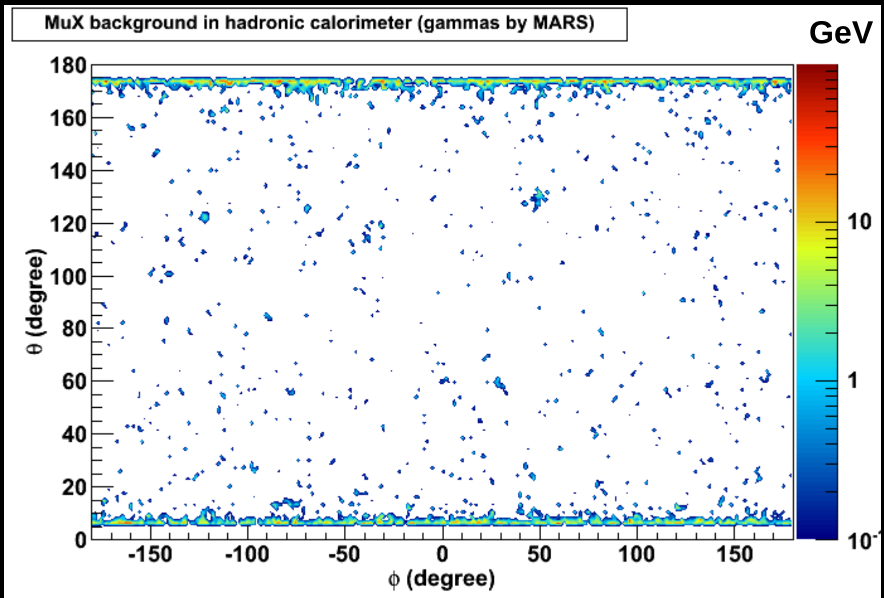


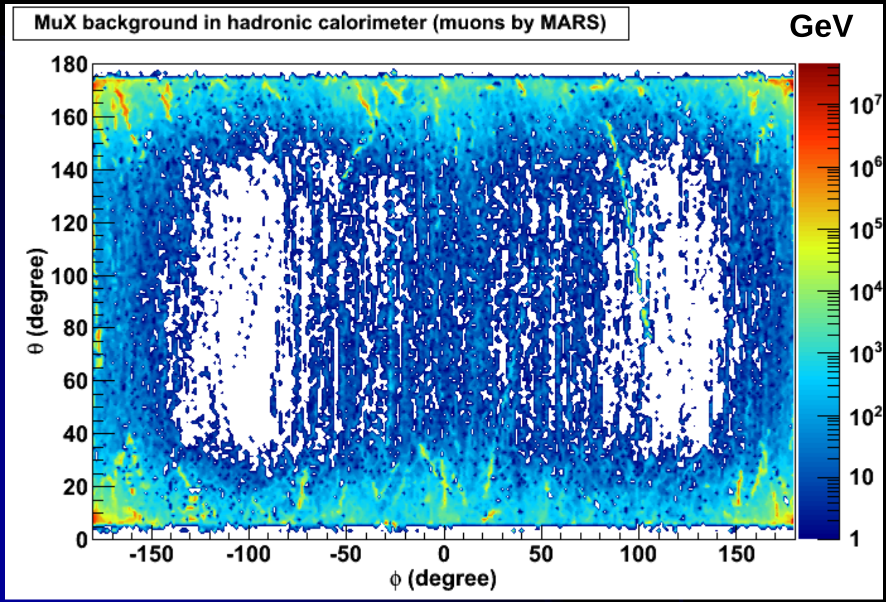
8 orders of magnitued between blue and red area)

Most of background source is located in well delimited regions

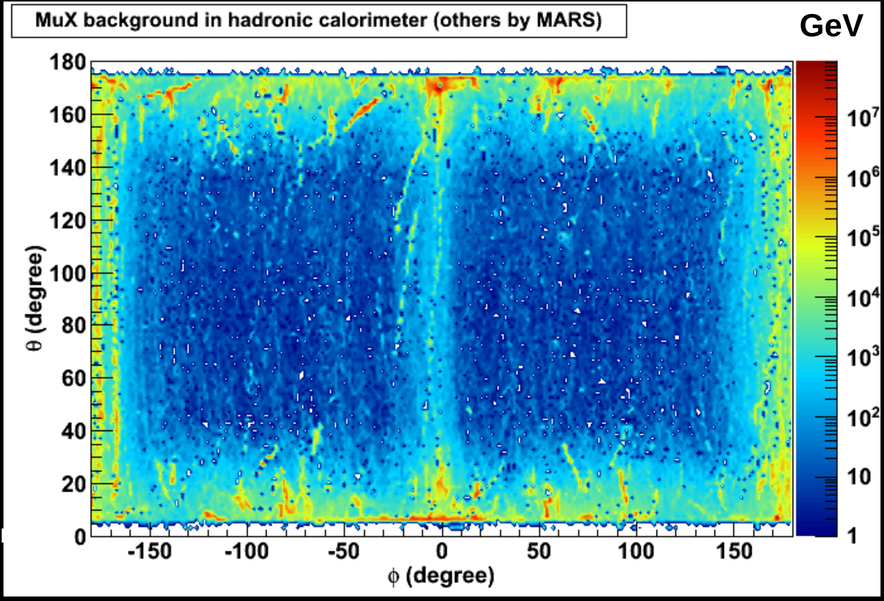
HAD Energy/4x4cm² vs MARS particle species (MDI separation plane=7.5 m from IP)



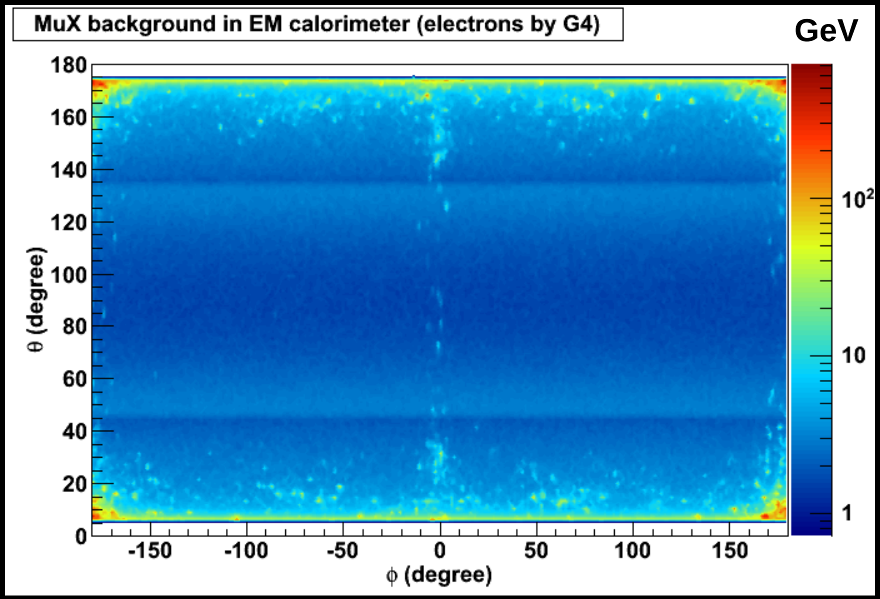


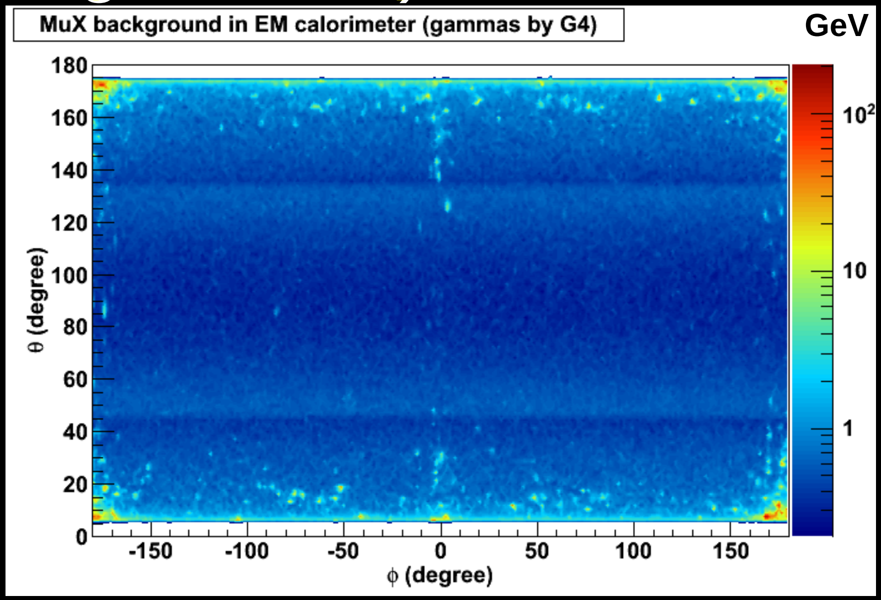


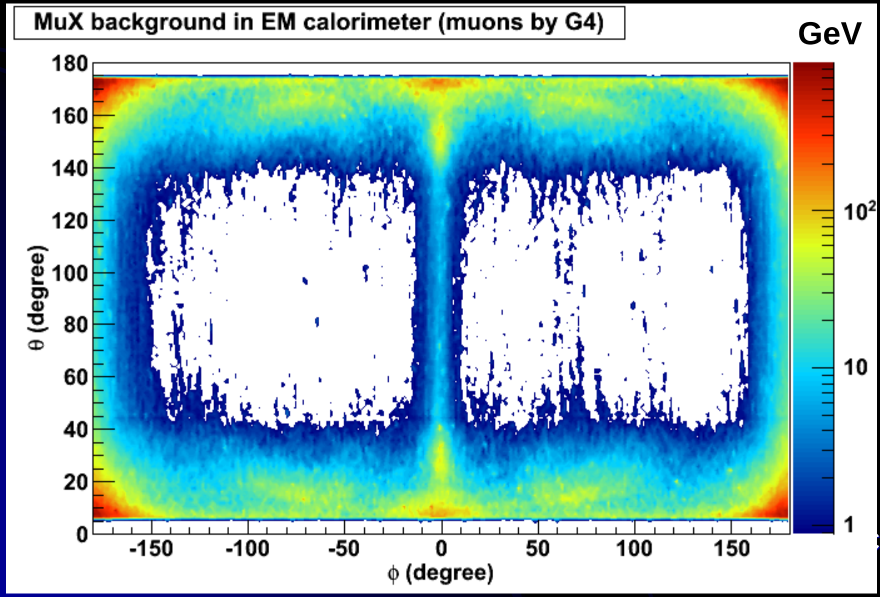
Collider

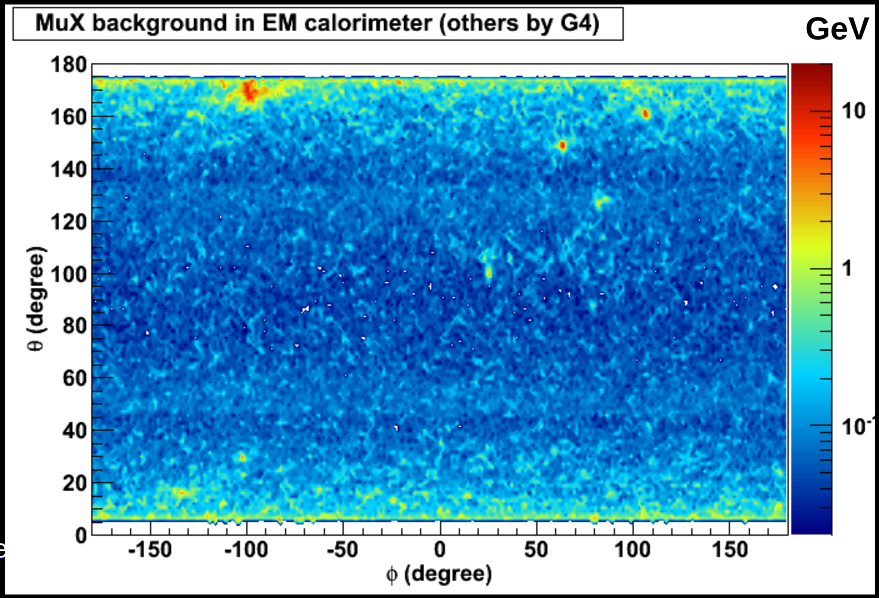


Energy/4x4cm^2 vs particles entering the EM calorimeter (G4 generator)

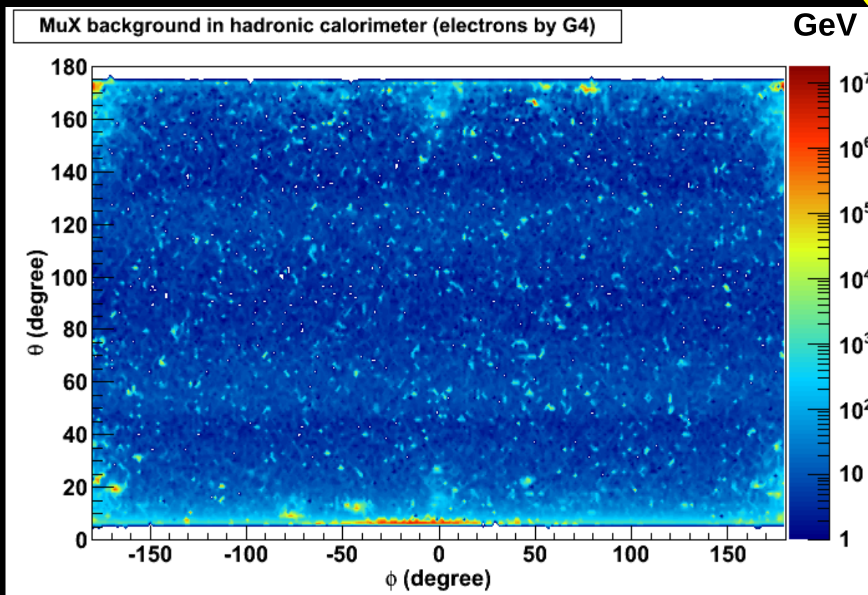


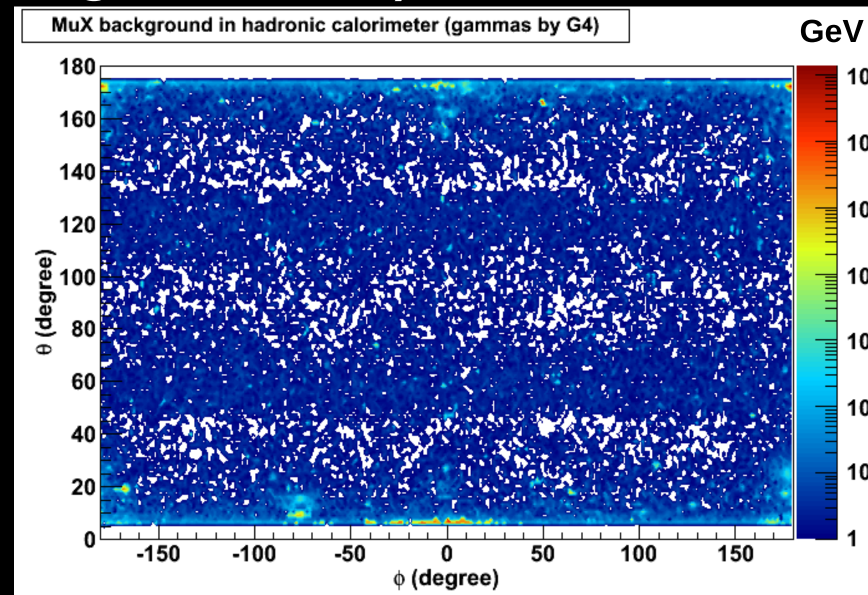


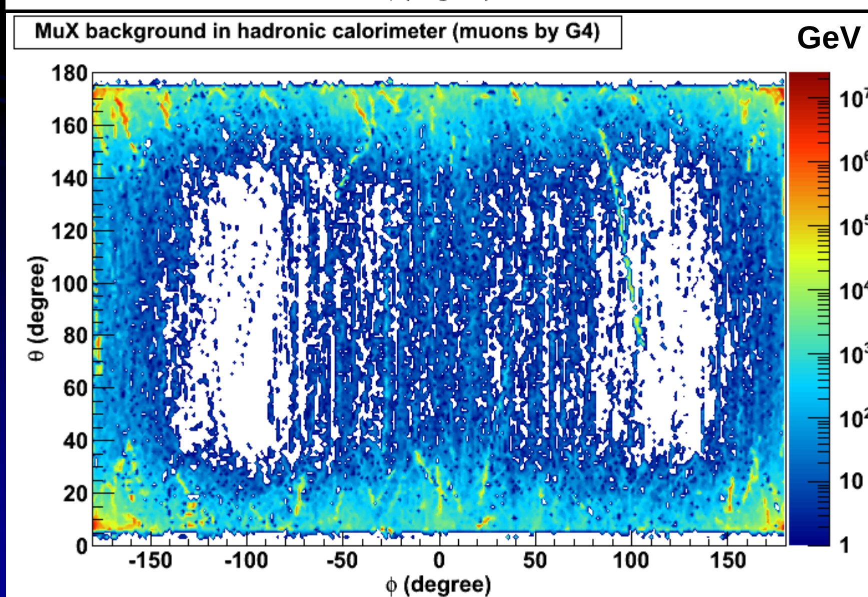


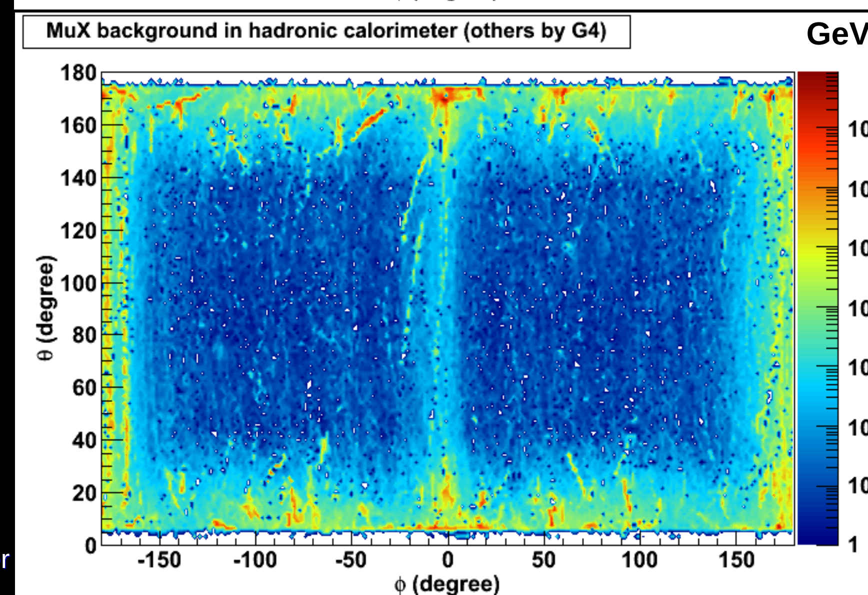


Energy/4x4cm^2 vs particles entering the HAD calorimeter (G4 generator)

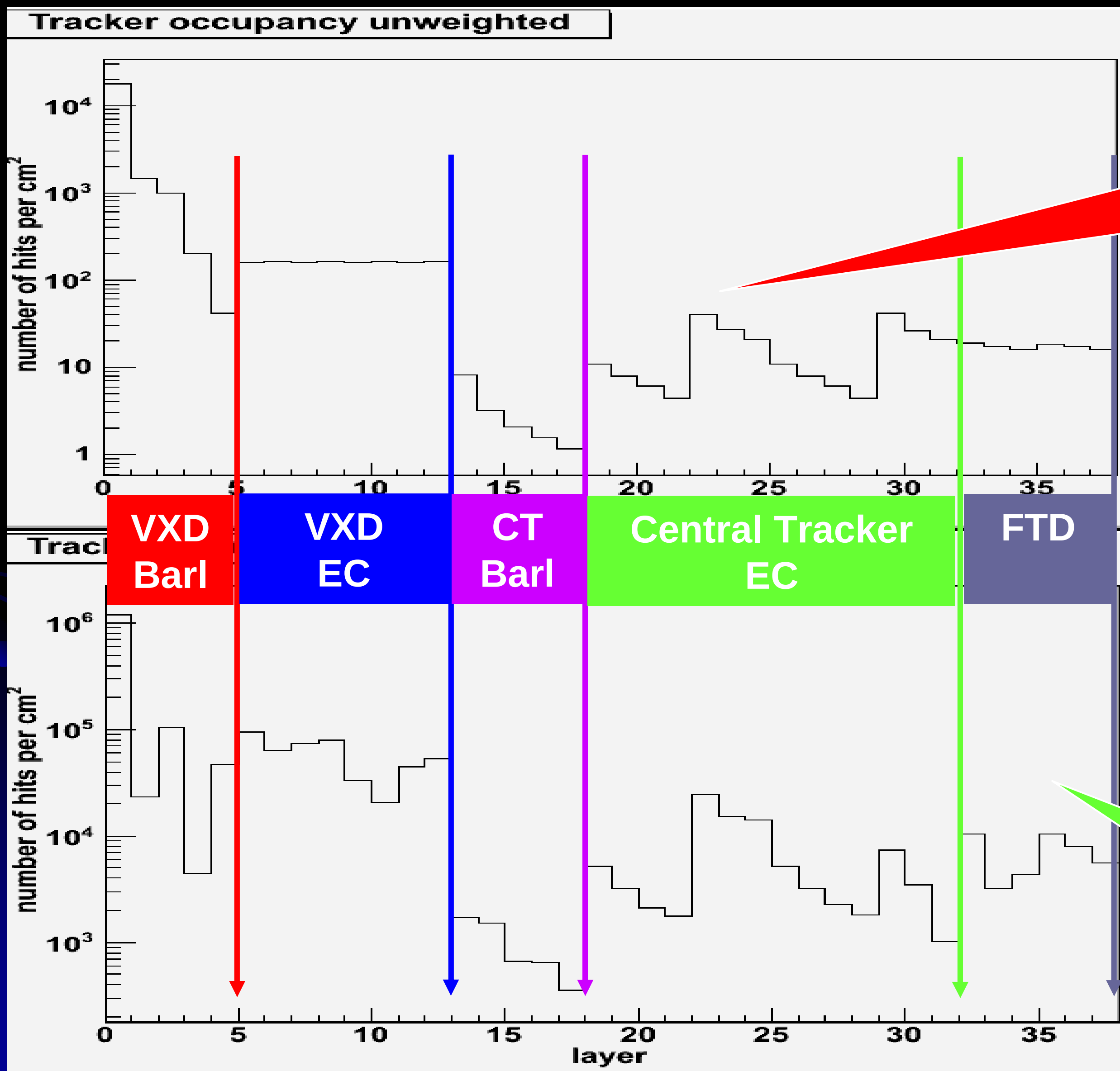








Occupancy in the Tracking Systems (from MARS+G4)



Unweighted:

MARS output sent to G4 disregarding the weights

None of the two plots is fully correct

Dedicated approach is required (see later)

Correct occupancy is in the middle

Weighted:

MARS output sent to G4 multiplying by its weight

Phase I: Tools preparation (on going)

- Physics
  - Preparation of a list of few benchmark processes for their implementation in the Physics event generators
- Accelerator:
  - A machine lattice configuration optimized in MARS15 for 0.75 and 1.5 TeV muon beams, at the machine-detector interface plane (z=7.5) to
  - Preliminary design of forward shielding (”cone”) inside the detector in proximity of the IP.
- Detector:
  - Implementation in ILCroot of detector models to be used for various studies to be performed in Phase II and Phase III (see later).
  - Determination of the threshold parameters (”pre-cuts”) in MARS15 simulation in order to reduce the contributions to the background only to the sources effectively affecting detector performance.

Goal is to have a stable lattice configuration and three detector configurations for Phase II

Phase II: Optimization of detector and machine lattice wrt background. Initial Physics studies - 9 months

- Physics
  - Preliminary plots of observables for the analysis of few Physics Benchmark processes, using a simplified detector simulation (Det-V0, already distributed)

- Accelerator:
  - A consistent design of the Interaction Region (IR), based on superconducting magnets capable of handling 0.5 kW/m loss rate, verified thorough MARS15 calculations, with inner liners, masks and shielding against Bethe-Heitler muons for 1.5-TeV muon beams.
  - A source term, calculated with MARS15 for 1.5-TeV muon beams, at the machine detector interface plane for the optimal IR and shielding configuration.

- Detector:
  - Optimization of detector wrt background for various lattice configurations (see later)

Goal is to freeze a lattice configuration and a detector layout for Phase III

Phase III: Optimization of detector and forward shielding wrt Physics. Full Physics studies - 12 months

- Physics, Accelerator, Detector:
  - Perform the Physics analysis of the benchmark channels
  - All analysis will be performed for few different forward shielding ("nose").

Goals are:

1) to freeze a forward shielding configuration
and a detector layout/technology

2) perform final physics studies with such configuration

Requirements for a software framework for detector studies at large colliders

- Easy interface with existing packages:
  - Geant3 , Geant4, Fluka, Event generators, HPSS, grid, etc
- Simple structure to be used by non-computing experts
- Portability
- Scalability
- Experiment-wide framework
- Use a world-wide accepted framework, if possible

Collaboration-specific framework is less likely to survive in the long term

Strategies for a Software Framework

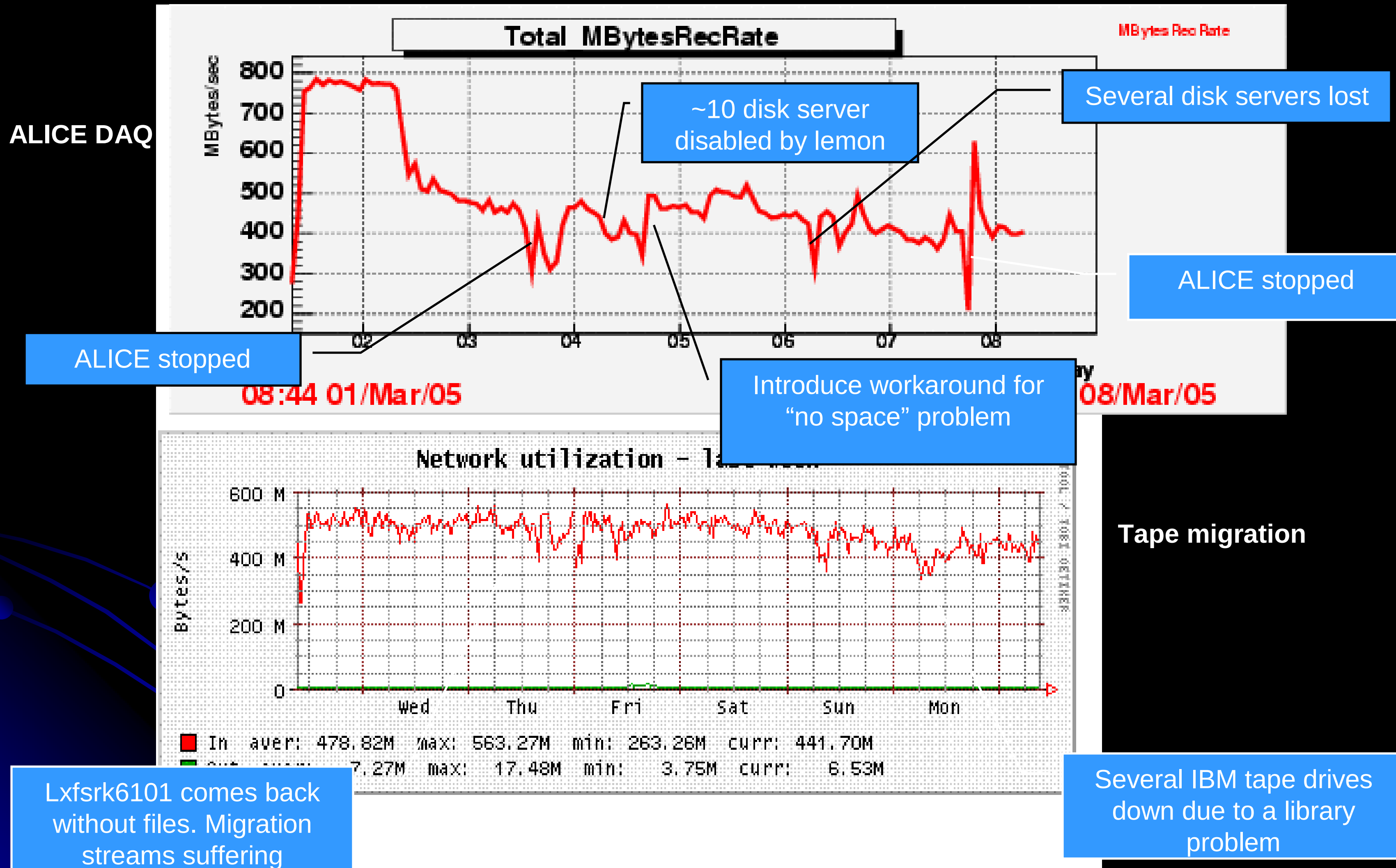
- Use of public domain common tools
- Adopt the ROOT framework
  - all needed functionalities present (from data taking to final plots)
  - reconstruction & analysis are naturally developing in the same framework
  - Extensive support by large labs
  - Unprecedented Large contributing HEP Community
  - Open Source project
  - Multiplatforms
  - Support multi-threading and asynchronous I/O
  - Optimised for different access granularity (Raw data, DST's, NTuple analysis)
- *Impose* a single framework
  - Provide central support, documentation and distribution
  - Train users in the framework

a) Do not re-invent the wheel
b) Be productive immediately

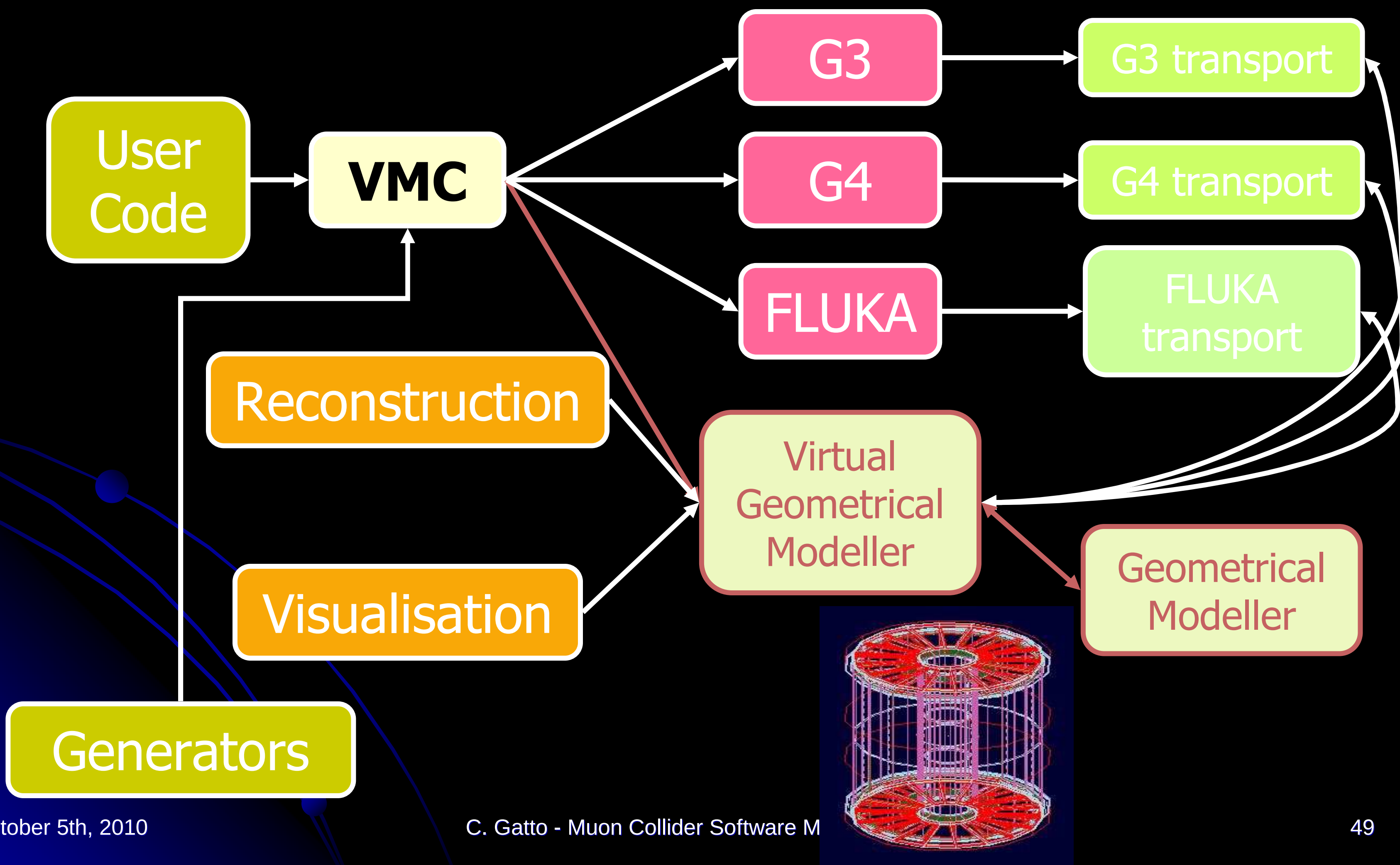
General Architecture: Guidelines

- Ensure high level of modularity (for easy of maintenance and development)
  - Absence of code dependencies between different detector modules (to C++ header problems)
  - Design the structure of every detector package so that static parameters (i.e. geometry and detector response parameters) are stored in distinct objects
- The data structure to be built up as ROOT TTree-objects
  - Access either the full set of correlated data (i.e., the event) or only one or more sub-sample (one or more detectors).

Performance (Alice's VI MDC)



The Virtual MC Concept



The VMC Advantage

- It decouple the user from keeping up with MC code updates (just update VMC lib)
- It allows the comparison between Geant3 Geant4 and Fluka using the same geometry and data structure (QA)
- You can generate and simulate different events with different MC's and merge the digits
- Example:
  - Geant4 for signal event
  - Fluka for beam background

→ **digit**

ILCroot Strategy: Modularity

- ILCroot Building Block: The Detector Class
- Detector-centric approach (vs Processor-centric)
- Main policy: each detector is responsible for its code & data
- Cross-modules calls are not allowed

↓

- Easy to work for groups across many countries
- Allow for several versions of the same detector or several detector of the same kind (ex. TPC & DCH)

The Detector Class

- **Both sensitive modules (detectors) and non-sensitive ones are described by this base class.**
- **This class must support:**
  - **Geometry description**
  - **Event display**
  - **Simulation by the MC**
  - **Digitization**
  - **Pattern recognition**
  - **Local reconstruction**
  - **Local PiD**
  - **Calibration**
  - **QA**
  - **Data from the above tasks**
- **Several versions of the same detector are possible (choose at run time)**

- The geometry can be specified using:
  - ***Root (TGeo)***
  - Geant3
  - Geant4
  - Fluka
  - GDML
  - XML
  - Oracle
  - CAD (semi-automatic)

Geometry Interface

- Detector Geometry can be exchanged with systems developed by other groups trough the VGM
- The Virtual Geometry Modeller (VGM) has been developed as a generalization of the existing convertors roottog4, g4toxml provided within Geant4 VMC
- VGM can provide format exchange among several systems (Geant4, Root, XML AGDD, GDML)
- The implementation of the VGM for a concrete geometry model represents a layer between the VGM and the particular native geometry.
- At present this implementation is provided for the Geant4 and the Root TGeo geometry models.
- Not an issue if the Geometry is described using Root TGeo
- At present, digitization of geometry imported in ILCRoot needs to be coded within the system
- SLAC's extended GDML (for digitization) could be implemented for automatic digitization

Coordinating the Detectors

- Detector stand alone (Detector Objects)
  - Each detector executes a list of detector actions/tasks
  - On demand actions are possible but not the default
  - Detector level trigger, simulation and reconstruction are implemented as clients of the detector classes
- Detectors collaborate (Global Objects)
  - One or more Global objects execute a list of actions involving objects from several detectors
  - Data are exchanged using a whiteboard techinque
- The Run Manager
  - executes the detector objects in the order of the list
  - Global trigger, simulation and reconstruction are special services controlled by the Run Manager class
- The Offline configuration is built at run time by executing a ROOT macro (Configuration file)

Example: VXD SDigitization

- Define Segmentation (at run-time)
- Define Model: Silicon Pixel, Silicon Strip, Silicon Drift (at run-time)
- Add background hits from file (optional)
- Step into materials (min. Step = 1μm)
  - Convert energy deposited by MC into charge
  - Spread charge in asymmetric way (ExB effect)

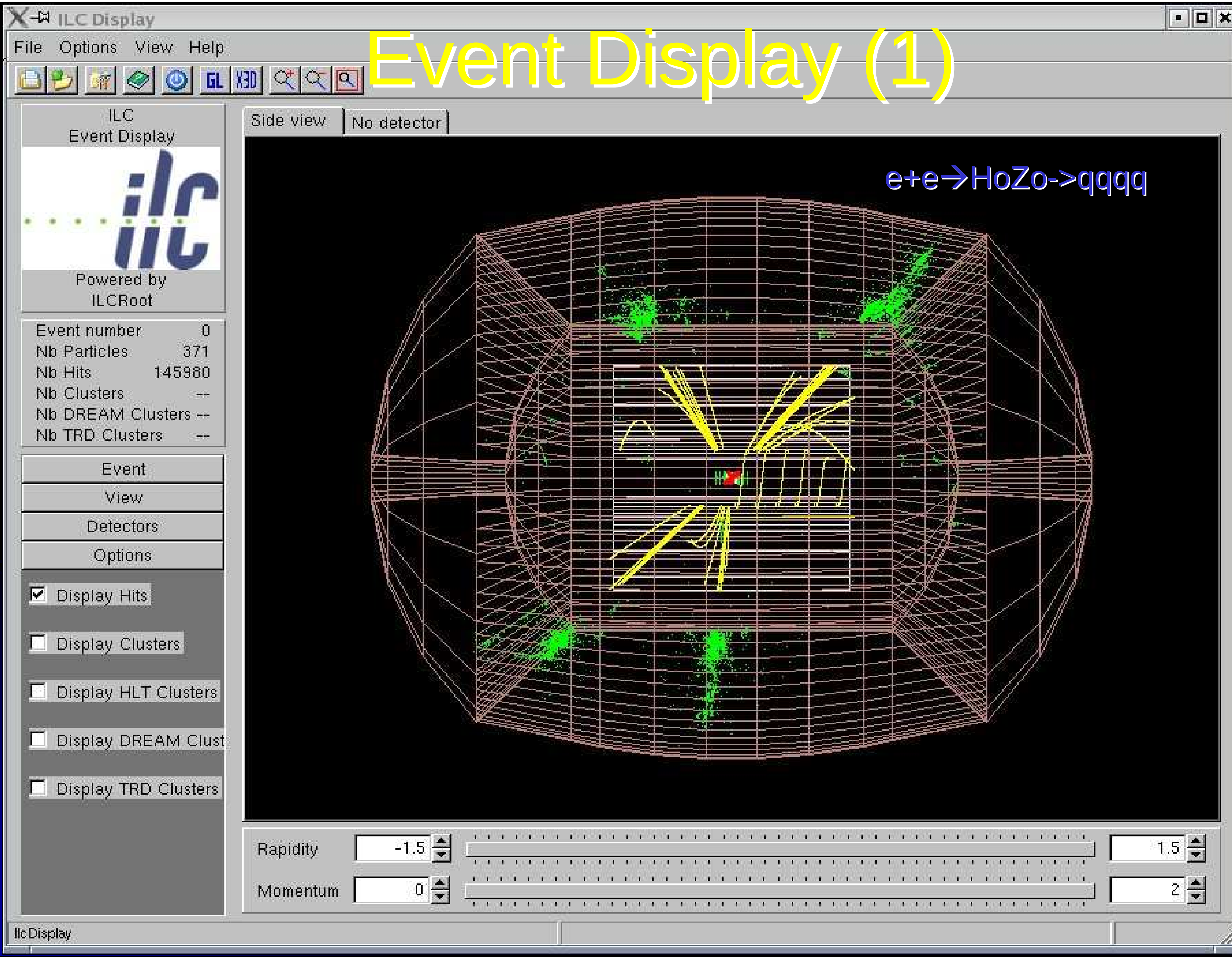
    $D(x,z)=Erfc(x,z,\sigma_z,\sigma_x)$

    $\sigma_z = sqrt(2k/e \times T^\circ \times (thickness/bias\ V) \times step)$

    $\sigma_x = asymm \times \sigma_z$
  - Add pixels to list
- Add coupling between nearby pixels
- Remove dead pixels (optional)

It allows to study and optimize a detector technology including effects as: beam background, magnetic field, integration time, etc.

Event Display (1)



Config.C example: generators

```
ILCGenCocktail *gener = new ILCGenCocktail();
gener->SetMomentumRange(0, 999999.);
gener->SetPhiRange(-180., 180.); // Set pseudorapidity range from -8 to 8.
gener->SetThetaRange(thmin,thmax); // Underlying Event // //
ILCGenHIJINGparaBa *bg = new ILCGenHIJINGparaBa(82534); ILCGenHIJINGparaBa *bg = new
    ILCGenHIJINGparaBa(10);
// // Jets from Pythia //
ILCGenPythia *jets = new ILCGenPythia(-1); // Centre of mass energy
jets->SetEnergyCMS(5500.);
// Process type
jets->SetProcess(kPyJets);
// final state kinematic cuts
jets->SetJetEtaRange(-0.3, 0.3);
// Decay type (semielectronic, semimuonic, nodecay)
jets->SetForceDecay(kAll);

// // Add all to cockail ... //
gener->AddGenerator(jets,"Jets",1);
gener->AddGenerator(bg,"Underlying Event", 1);
```

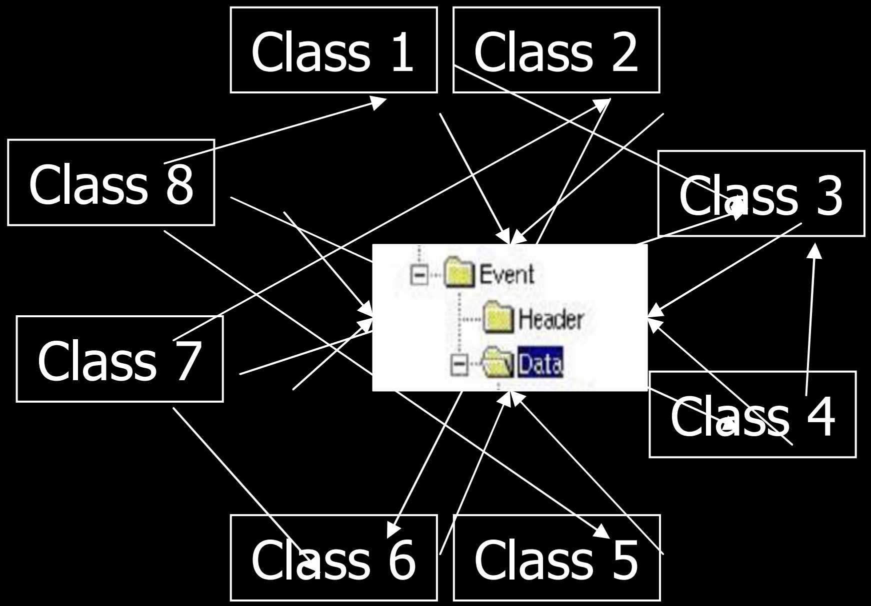
Config.C example: processes and cuts

```
// ************* STEERING parameters FOR ILC SIMULATION
      **************
// --- Specify event type to be tracked through the ILC setup
// --- All positions are in cm, angles in degrees, and P and E in
      GeV
gMC->SetProcess("DCAY",1);
gMC->SetProcess("PAIR",1);
gMC->SetProcess("COMP",1);
gMC->SetProcess("PHOT",1);
gMC->SetProcess("PFIS",0);
gMC->SetProcess("DRAY",0);
gMC->SetProcess("ANNI",1);
gMC->SetProcess("BREM",1);
gMC->SetProcess("MUNU",1);
gMC->SetProcess("CKOV",1);
gMC->SetProcess("HADR",1);
gMC->SetProcess("LOSS",2);
gMC->SetProcess("MULS",1);
gMC->SetProcess("RAYL",1);
```

```
Float_t cut = 1.e-3;
// 1MeV cut by default
Float_t tofmax = 1.e10;
gMC->SetCut("CUTGAM", cut);
gMC->SetCut("CUTELE", cut);
gMC->SetCut("CUTNEU", cut);
gMC->SetCut("CUTHAD", cut);
gMC->SetCut("CUTMUO", cut);
gMC->SetCut("BCUTE", cut);
gMC->SetCut("BCUTM", cut);
gMC->SetCut("DCUTE", cut);
gMC->SetCut("DCUTM", cut);
gMC->SetCut("PPCUTM", cut);
gMC->SetCut("TOFMAX", tofmax);
```

Run-time Data-Exchange

- Post transient and persistent data to a white board
- Structure the whiteboard according to detector sub-structure & tasks results
- Each detector is responsible for posting its data
- Tasks access data from the white board
- Detectors cooperate through the white board



Reconstruction in ILCroot

VXD Digitization

- Read SDigits from several files (produced by different generators and/or MC)
- Add electronic noise
- Cut signal + electronic noise < threshold
- Zero suppression

VXD Cluster Finding

- Create a initial cluster from adiacent pixels (no for diagonal)
- Subdivide the previous cluster in smaller NxN clusters (default 3x3)
- Kalman filter picks up the best clusters

TPC Simulation

- Pads simulation. Gaussian smearing according to:

Sigma of cluster COG position determination

- σ_t of cluster center (not systematic (threshold)effect):

$$\sigma_{tCOG} = \sqrt{\frac{\sigma_L^2(z_{max} - z)}{N_{ch}} G_g + \frac{tan(\alpha)^2 l_{pad}^2 G_{Landau}(N_{prim})}{12 N_{chprim}} + \sigma_{noise}^2} \quad (7)$$

50 µm

- σ_p of cluster center (not systematic (threshold) effect):

$$\sigma_{pCOG} = \sqrt{\frac{\sigma_T^2(z_{max} - z)}{N_{ch}} G_g + \frac{tan(\beta)^2 l_{pad}^2 G_{Landau}(N_{prim})}{12 N_{chprim}} + \sigma_{noise}^2} \quad (8)$$

N_{ch} - total number of electrons in cluster
N_{chprim} - number of primary electrons in cluster
G_g - gas gain fluctuation factor
G_{Landau} - secondary ionization fluctuation factor

- Digital readout:
  - Simulate gas transport
  - illuminate each pixel using cluster statistics and ε = 90%/electron.

Calorimeter SDigitization

- Simulate light production in each quartz and plastic fiber with ad hoc algorithms (includes light transport)
- Add PM efficiency
- Ad random background

Calorimeter Digitization

- Read SDigits from several files (produced by different generators and/or MC)
- Extract E from E_s and E_c (for use with jet-finders)
- Zero suppression

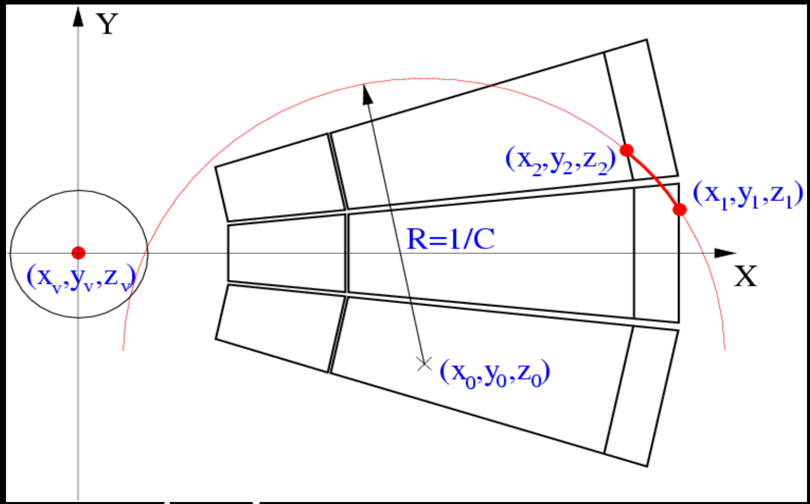
Calorimeter Clusterization

- Add togheter adjacent cells with signal in a large cluster
- Look for peaks in the shape of the signals
- Perform cluster unfolding via a Minuit fit
- Attempt to associate the final clusters to a track from the Kalman filter (succesful for isolated tracks/clusters only)

Global Tracking: seeding

Primary Seeding with vertex constrain

- Take 2 pad-rows with gap 20 rows
- Check quality of track segment:
  - chi2
  - number of founded clusters
  - number of shared clusters



Secondary Seeding without vertex constrain

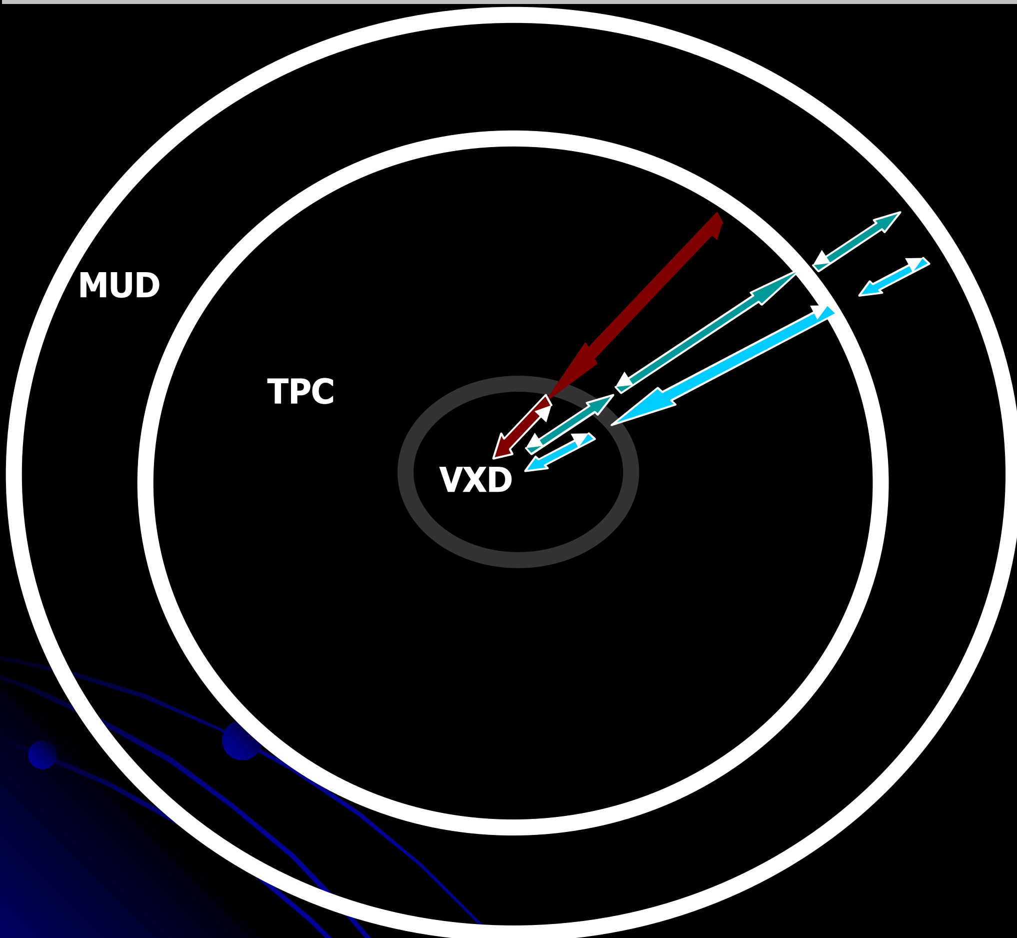
- Simple track follower
- Algorithm
  - Seeding between 3 pad-rows (with gaps 2 rows)
  - Check that nearest clusters available at prolongation
  - Find prolongation to inner radius to make 20 rows segment
  - Check quality of track segment

Track Efficiency limited by efficiency of seeding!

Parallel Kalman Filter

- seedings with constraint + seedings without constraint at different radii from outer to inner
-
- Tracking
  - Find for each track the prolongation to the next pad-row
  - Estimate the errors
  - Update track according current cluster parameters
  - (Possible refine clusters parameters with current track)
- Track several track-hypothesis in ***parallel***
  - Allow cluster sharing between different track
  - Find kinks
  - Find V0
- Remove-Overlap

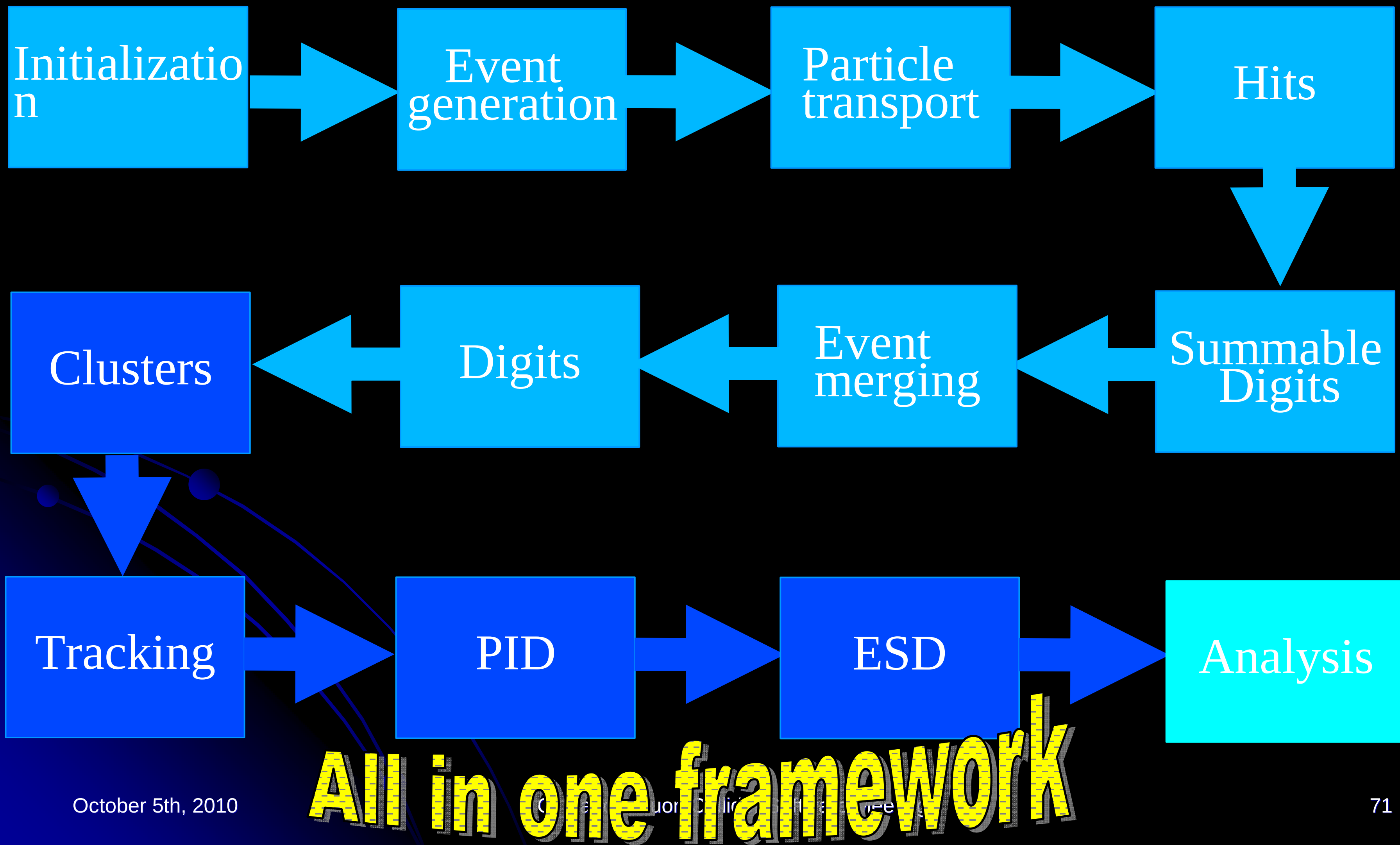
Tracking strategy – Primary tracks



- Iterative process
  - Forward propagation towards to the vertex –TPC-ITS
  - Back propagation –VXD-Central Tracker-MUD
  - Refit inward MUD-CT-VXD
- Continuous seeding – track segment finding in all detectors
- Try to find standalone tracks in MUD and VXD from leftover clusters

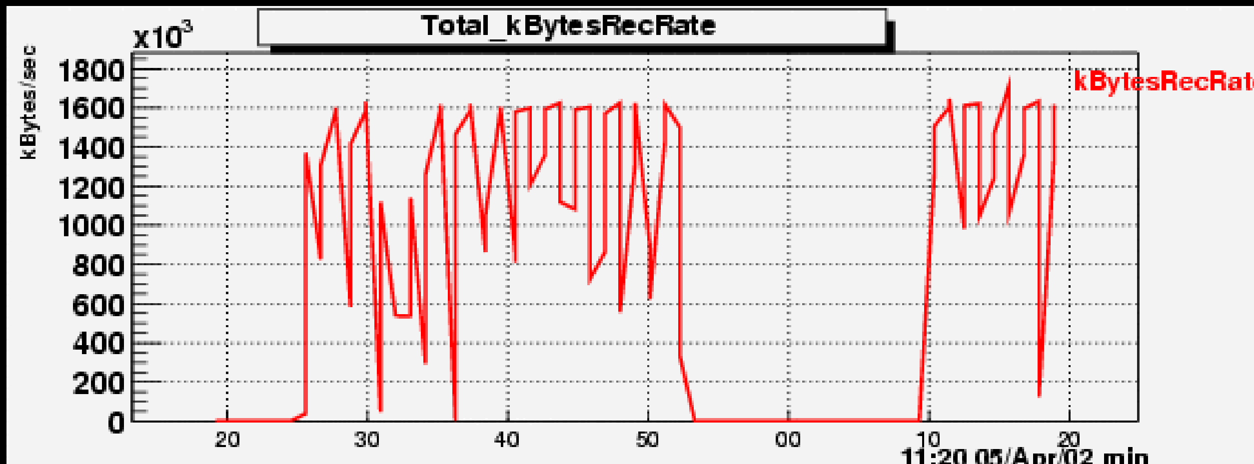
Backup slides

ILCroot Flow Control

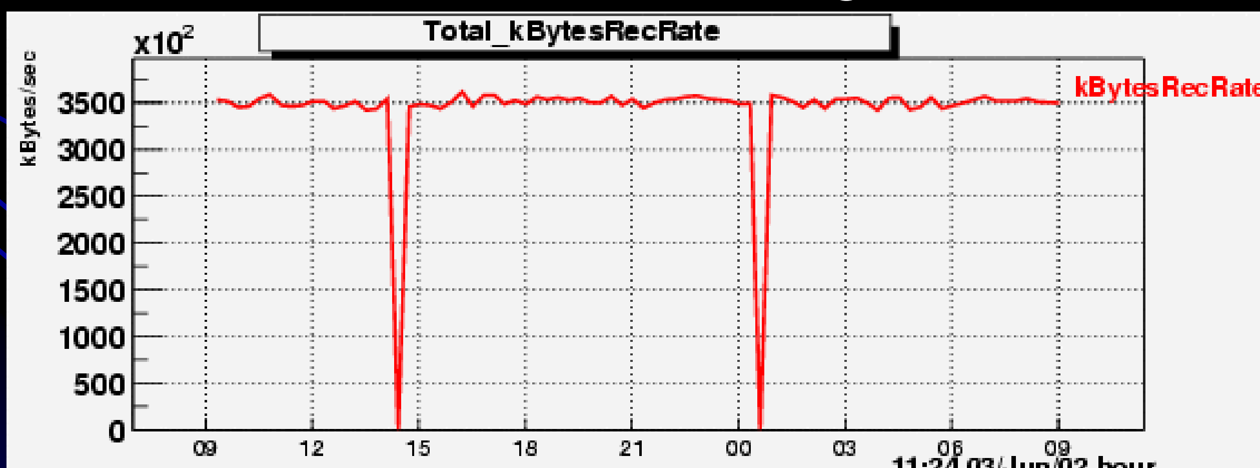


Performance (Alice's IV MDC)

Data generation in LDC, event building, no data recording



Data generation in LDC, event building, data recording to disk



Total: 192 CPU servers (96 on Gbe, 96 on Fe), 36 DISK servers, 10 TAPE servers

Processing Flow