ILCroot: the Software Framework of 4th Concept

Strategy

Architecture

Reconstruction Tools

4th Concept Software Strategy

- 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 CERN support
  - 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

Quite orthogonal to most of ILC software strategies

ILC software packages

	Description	Detector	Language	IO-Format	Region
Simdet	fast Monte Carlo	TeslaTDR	Fortran	StdHep/LCIO	EU
SGV	fast Monte Carlo	simple Geometry, flexible	Fortran	None (LCIO)	EU
Lelaps	fast Monte Carlo	SiD, flexible	C++	SIO, LCIO	US
Mokka	full simulation – Geant4	TeslaTDR, LDC, flexible	C++	ASCI, LCIO	EU
Brahms-Sim	Geant3 – full simulation	TeslaTDR	Fortran	LCIO	EU
SLIC	full simulation – Geant4	SiD, flexible	C++	LCIO	US
LCDG4	full simulation – Geant4	SiD, flexible	C++	SIO, LCIO	US
Jupiter	full simulation – Geant4	JLD (GDL)	C++	Root (LCIO)	AS
Brahms-Reco	reconstruction framework (most complete)	TeslaTDR	Fortran	LCIO	EU
Marlin	reconstruction and analysis application framework	Flexible	C++	LCIO	EU
hep.lcd	reconstruction framework	SiD (flexible)	Java	SIO	US
org.lcsim	reconstruction framework (under development)	SiD (flexible)	Java	LCIO	US
Jupiter-Satelite	reconstruction and analysis	JLD (GDL)	C++	Root	AS
LCCD	Conditions Data Toolkit	All	C++	MySQL, LCIC	EU
GEAR	Geometry description	Flexible	C++ (Java?)	XML	EU
LCIO	Persistency and datamodel	All	Java, C++, Fortran	-	AS,EU,US
JAS3/WIRED	Analysis Tool / Event Display	All	Java	xml,stdhep, heprep,LCIO,	US,EU

+ Event Generators

General Architecture: Guidelines

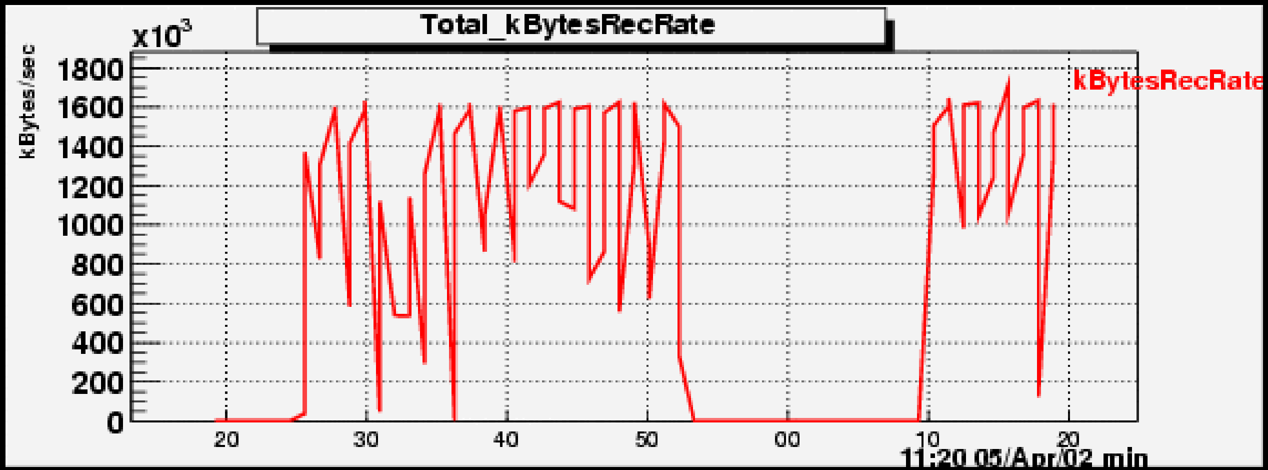
- Ensure high level of modularity (for easy of maintenance and development)
  - Absence of code dependencies between different detector modules (to C++ header problems)
  - Design the structure of every detector package so that static parameters (i.e. geometry and detector response parameters) are stored 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).

ILCroot: a summary of features

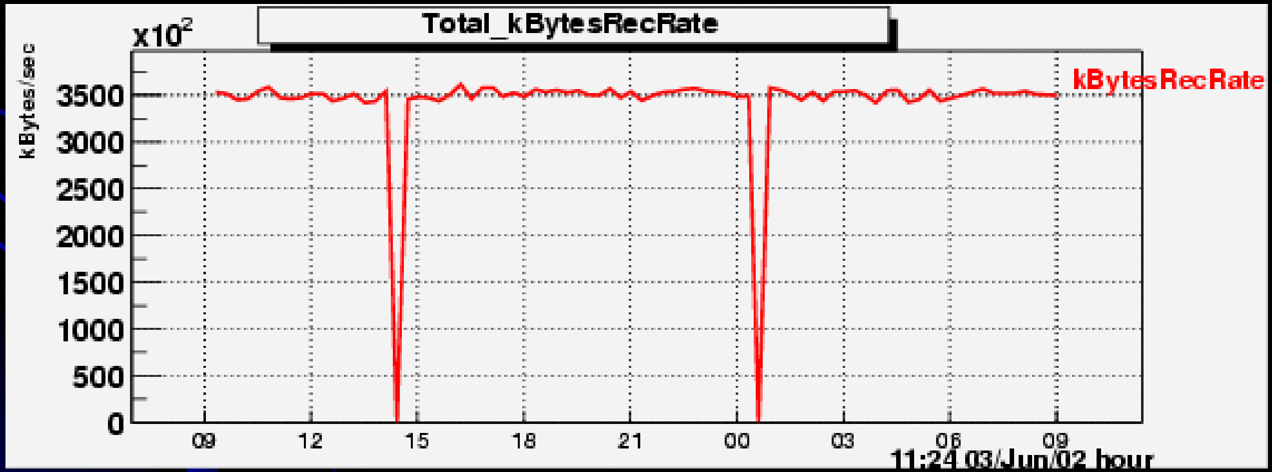
- CERN architecture (based on Alice's Aliroot)
- Full support provided by Brun, Carminati, Ferrari, et al.
- Uses ROOT as infrastructure
  - All ROOT tools are available (I/O, graphics, PROOF, data structure, etc)
  - Extremely large community of users/developers
  - ***Same framework as analysis***
- Six MDC have proven robustness, reliability and portability

Performance (Alice's IV MDC)

Data generation in LDC, event building, no data recording



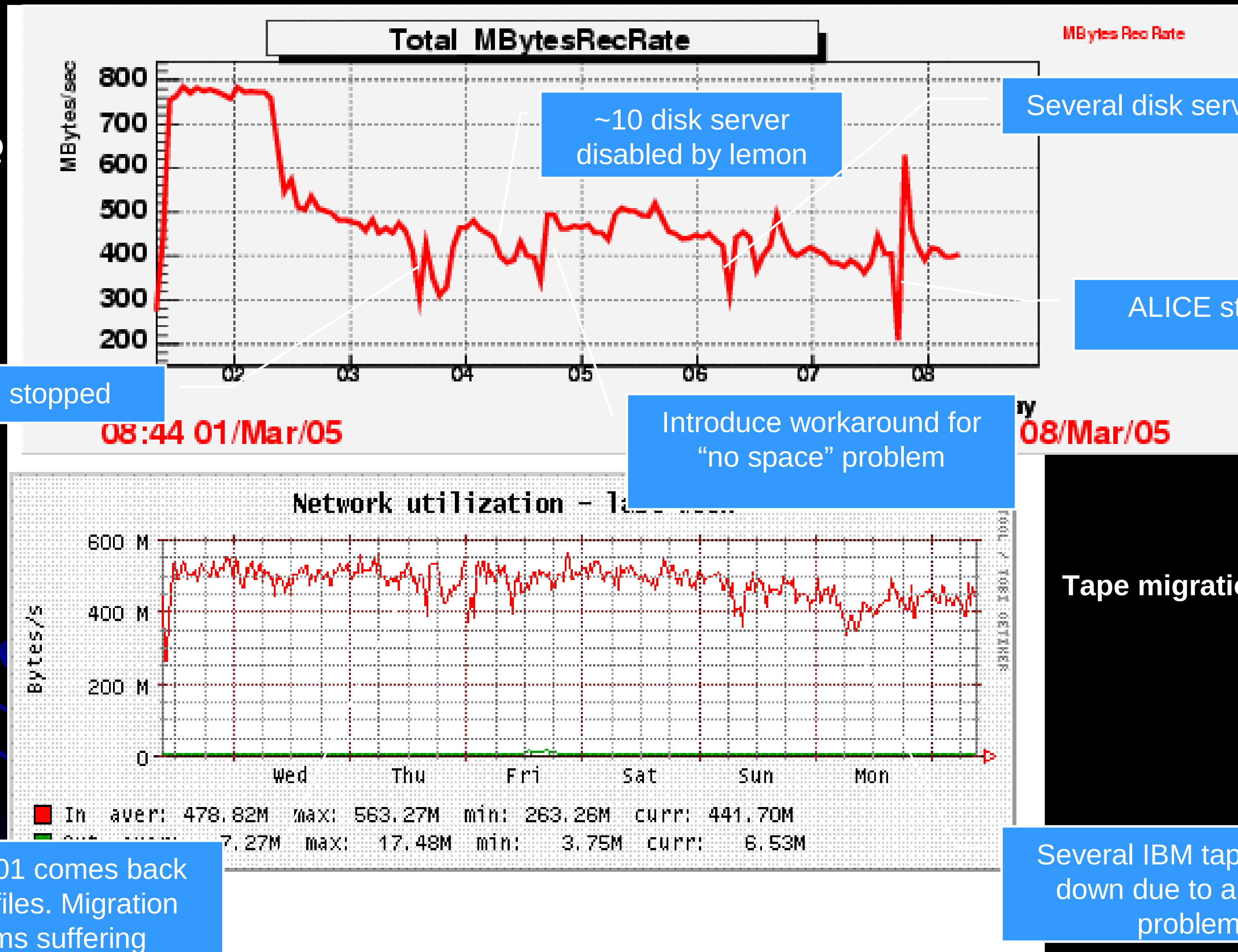
Data generation in LDC, event building, data recording to disk



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

Performance (Alice's VI MDC)

ALICE DAQ



~10 disk server disabled by lemon

Several disk servers lost

ALICE stopped

ALICE stopped

Introduce workaround for "no space" problem

Tape migration

Lxfsrk6101 comes back without files. Migration streams suffering

Several IBM tape drives down due to a library problem

The Virtual Montecarlo Concept

- Virtual MC provides a virtual interface to Monte Carlo
- It decouples the dependence of a user code on a concrete MC
- 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

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

→ digitization

TGenerator Concept

- ***TGenerator*** is an abstract base class
- It interfaces ROOT and the various event generators (thanks to inheritance)
- Possible to study
  - Full events (event by event)
  - Single processes
  - Mixture of both (“Cocktail events”)
  - Generation of Cocktail of different processes
  - Rate and weighting control
  - Allow easy mixing of signal and background
- More than a dozen Generators built-in

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

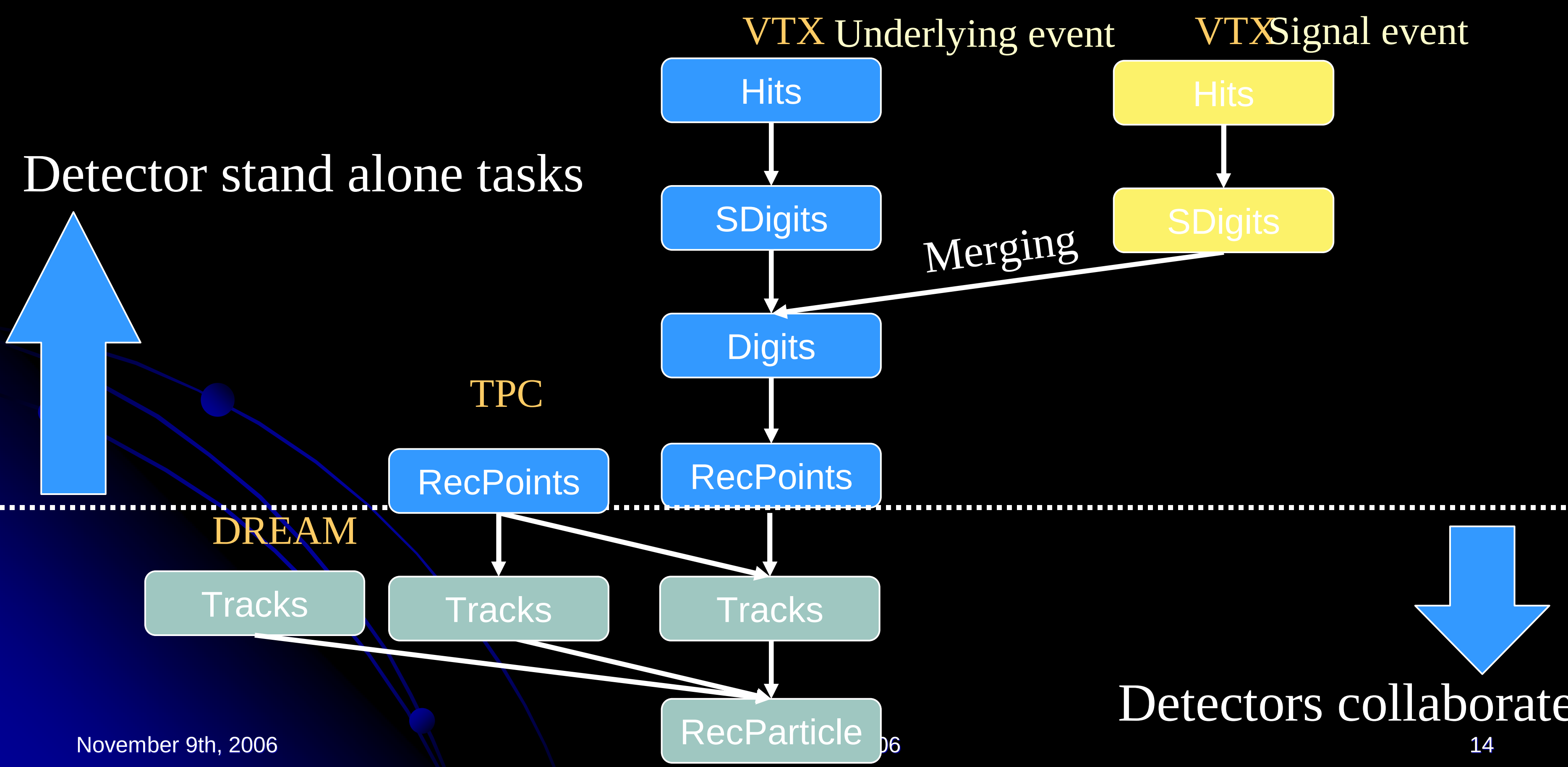
- **Both sensitive modules (detectors) and non-sensitive ones are described by this base class.**
- **This class must support:**
  - **Geometry description**
  - **Event display**
  - **Simulation by the MC**
  - **Digitization**
  - **Pattern recognition**
  - **Local reconstruction**
  - **Local PiD**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

Coordinating the Detectors

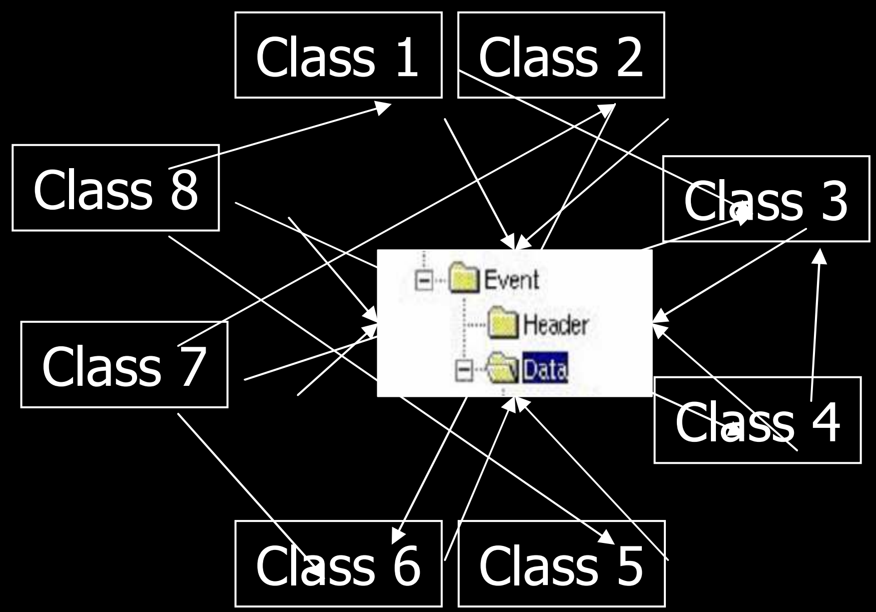
- Detector stand alone (Detector Objects)
  - Each detector executes a list of detector actions/tasks
  - On demand actions are possible but not the default
  - Detector level trigger, simulation and reconstruction are implemented as clients of the detector classes
- Detectors collaborate (Global Objects)
  - 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)

Processing Flow



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 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)

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):

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

- 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

50 µm

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

DREAM SDigitization

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

DREAM 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

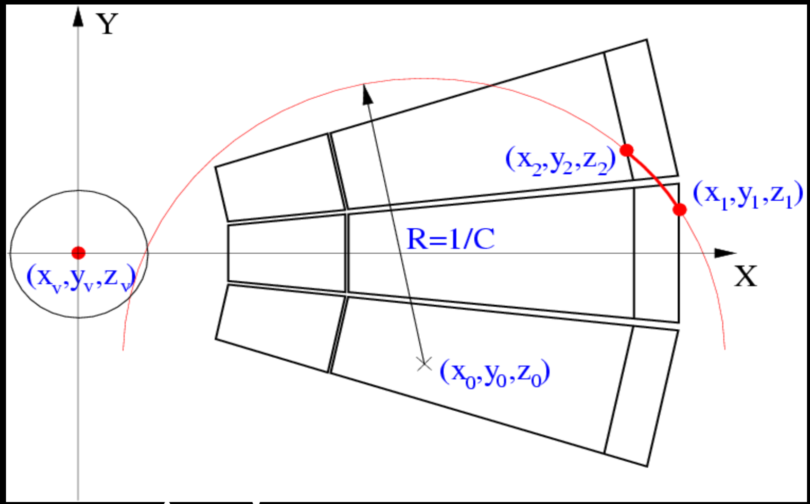
DREAM Clusterization

- 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

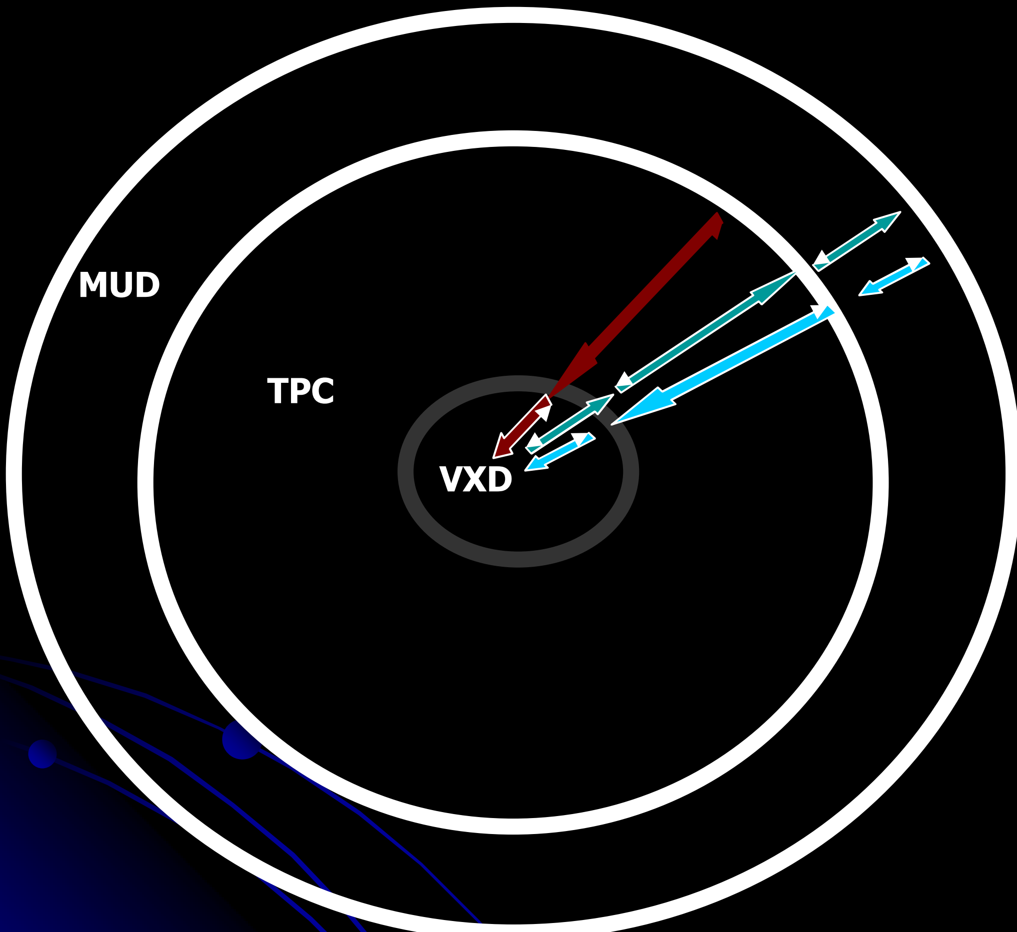
- Simple track follower
- Algorithm
  - Seeding between 3 pad-rows (with gaps 2 rows)
  - Check that nearest clusters available at prolongation
  - Find prolongation to inner radius to make 20 rows segment
  - Check quality of track segment

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

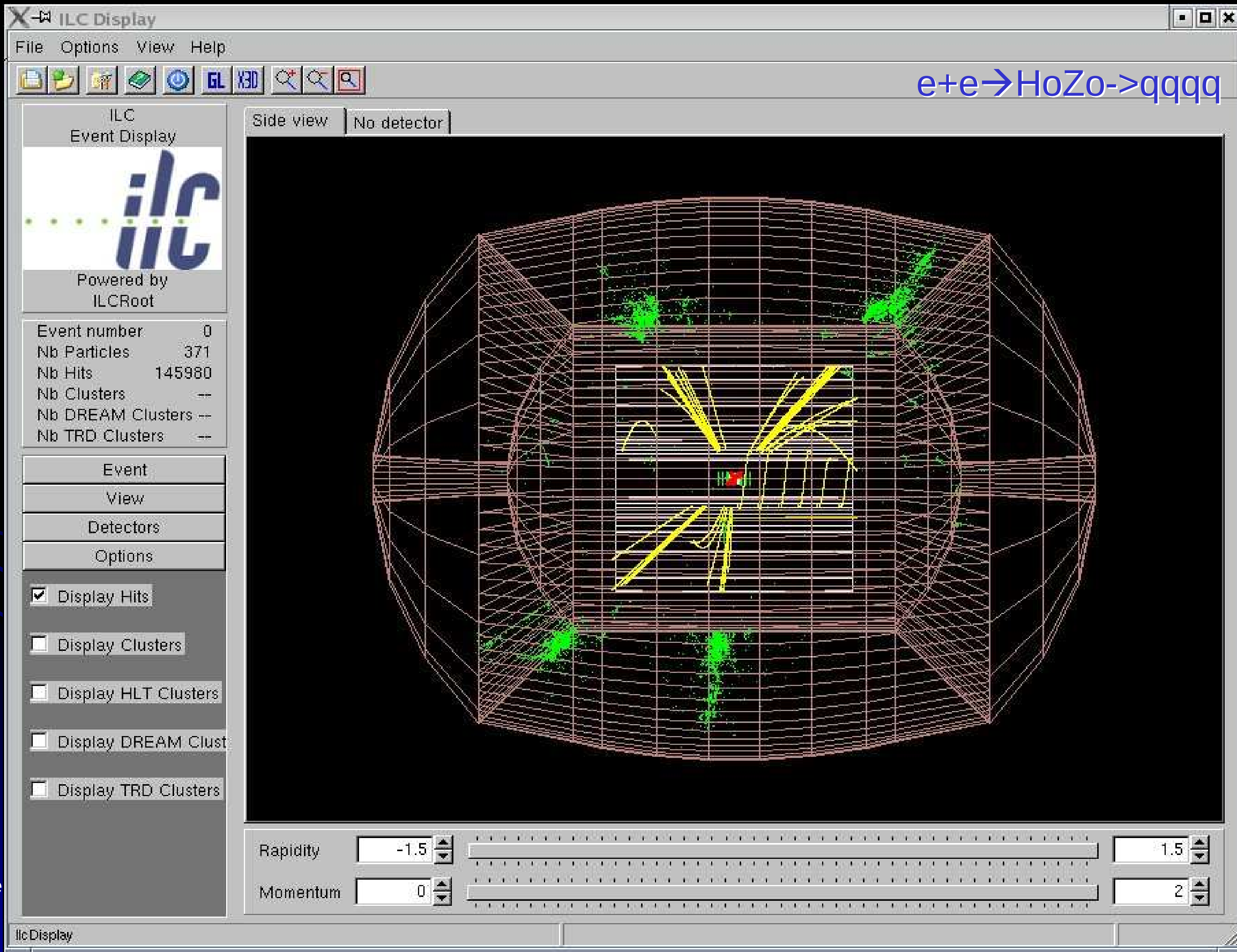
Tracking strategy – Primary tracks



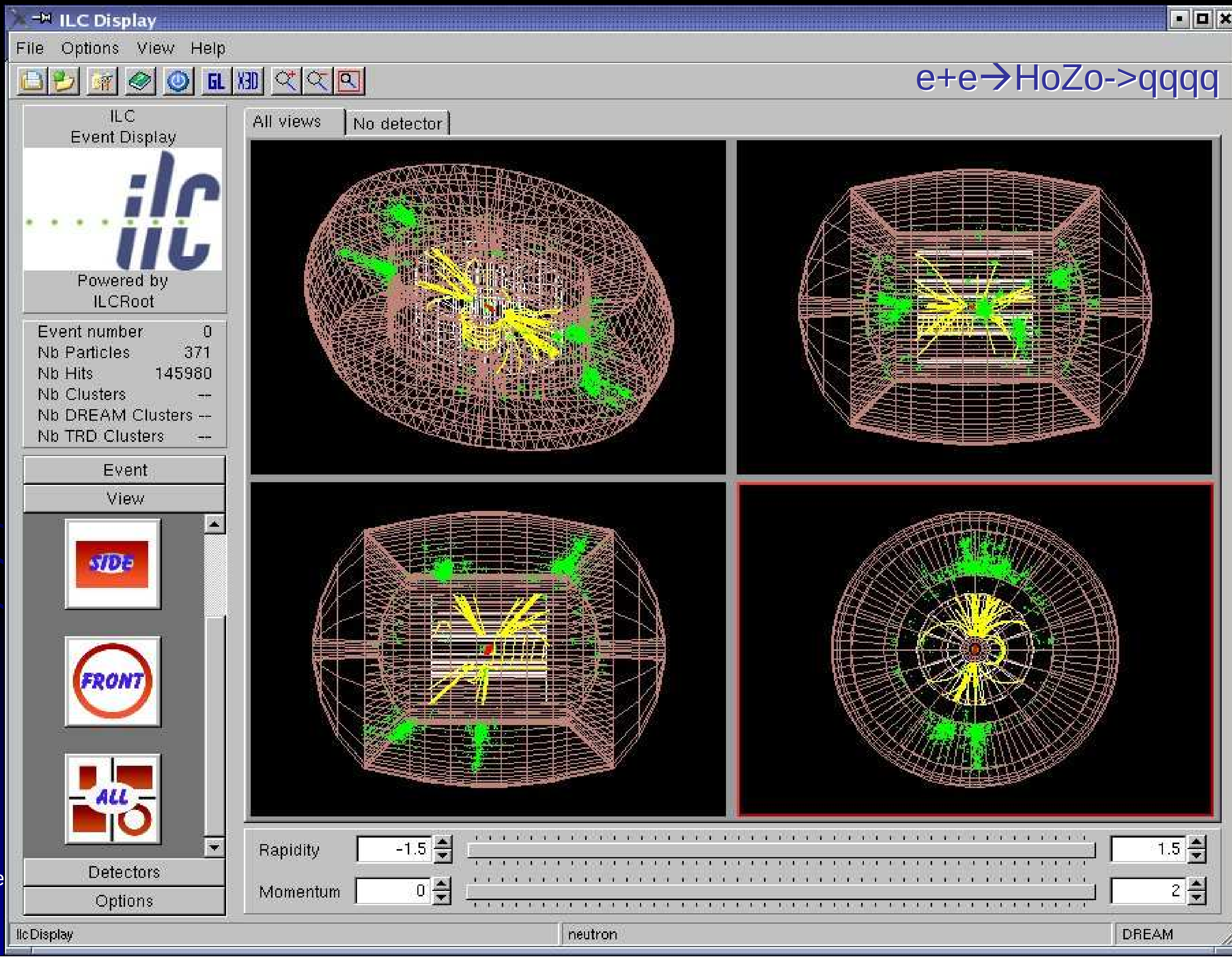
- Iterative process
  - Forward propagation towards to the vertex –TPC-ITS
  - Back propagation –VXD-TPC-MUD
  - Refit inward MUD-TPC-VXD
- Continuous seeding – track segment finding in all detectors
- Try to find standalone tracks in MUD and VXD from leftover clusters

currently is implemented TPC+VXD barrel propagation and Bangalore version of MUD

Present Status: VXD+TPC+DREAM



Present Status: VXD+TPC+DREAM

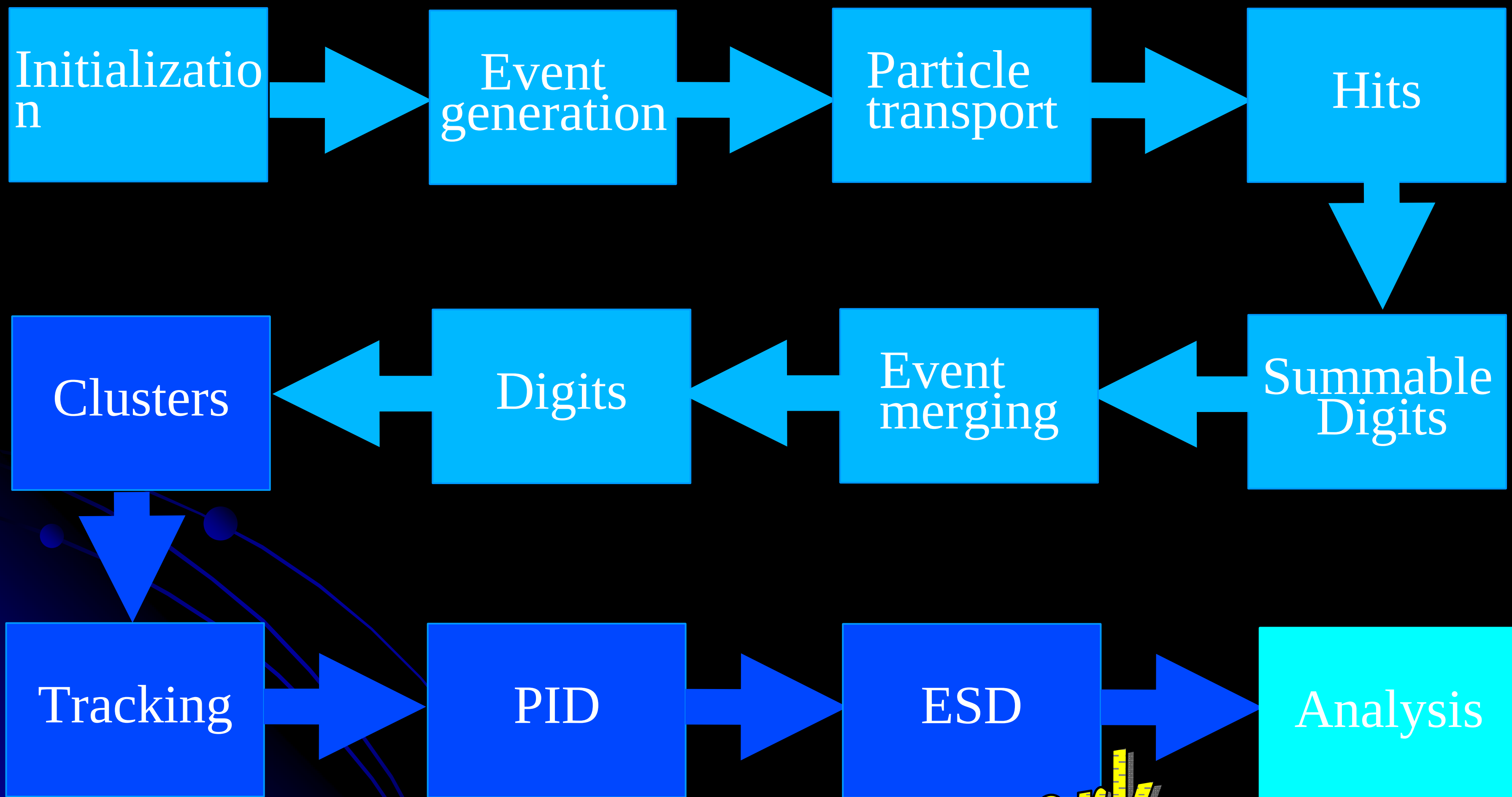


Conclusions

- ILCroot machinery is in place and running
- It is proving extremely stable (20 cpu's x 2 weeks with no crash)
- Still few steps to complete (about 1-2months):
  - Full Digitization + clusterization in the TPC (pads only)
  - Reconstruction in EMCAL (waiting for results from CERN test beam)
  - Reconstruction in new MUD
  - Kalman Filter in VXD amd MUD Endcaps
- Physics analyses already running
- Publicly available next week on the FNAL repository

Backup slides

ILCroot Flow Control



All in one framework

LCIO vs MONARC