Alice Offline Software Package

A Software Tool

Shahid Aslam

First School on LHC Physics, 12-30 October 2009

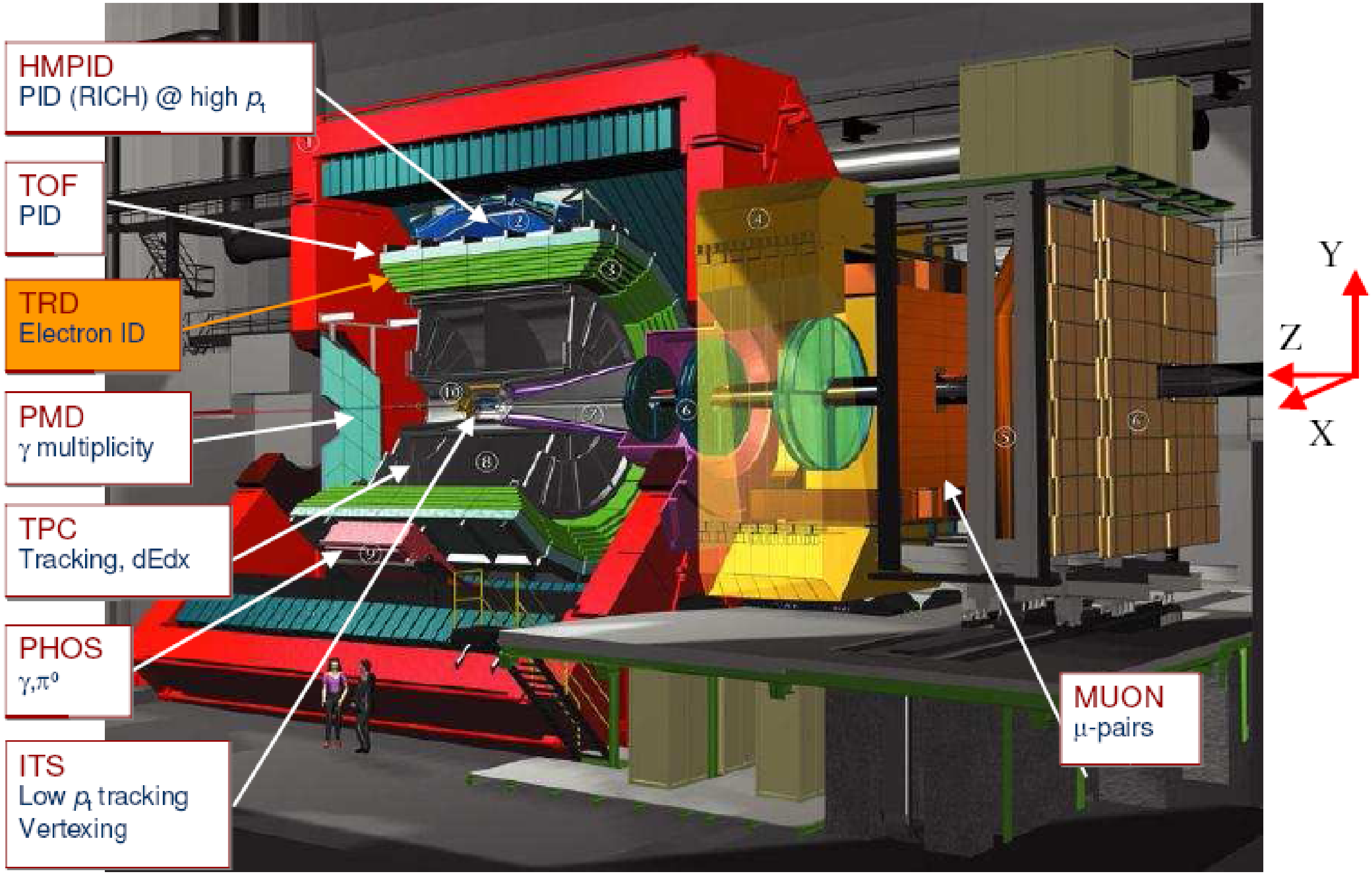
AliRoot Framwork

- The AliRoot framework is based on Object-Oriented technology and depends on the ROOT framework.
- Complexity:
- A large amount of work has been dedicated to reconstruct trajectories and identify particles. These tasks are particularly difficult in the context of heavy-ion collisions, as we expect that such collisions will produce a number of tracks an order of magnitude larger than in proton-proton (pp) collisions and that the occupancy can be as high as 40% in some regions.

The computing model

- The computing model for the **pp data** is similar to that of the other LHC experiments. Data are recorded at a rate of **100 MB/s**. They are reconstructed quasi on-line at the CERN Tier 0 facility. In parallel, data are exported to the different Tier 1s outside CERN (hereafter `external Tier 1s'), to provide two copies of the raw data, one stored at the CERN Tier 0 and another copy shared by all the external Tier 1s. All Tier 1s will have collectively enough resources to perform a second and third reconstruction pass.
- For **heavy-ion data** this model is not viable, as data are recorded at up to **1.25 GB/s**. Such a data rate would require a prohibitive amount of resources for quasi real-time processing. ALICE therefore requires that heavy-ion data be reconstructed at the CERN Tier 0 and exported during a period of four months after data taking.

The ALICE Experiment



HMPID
PID (RICH) @ high p_t

TOF
PID

TRD
Electron ID

PMD
γ multiplicity

TPC
Tracking, dEdx

PHOS
γ,π^0

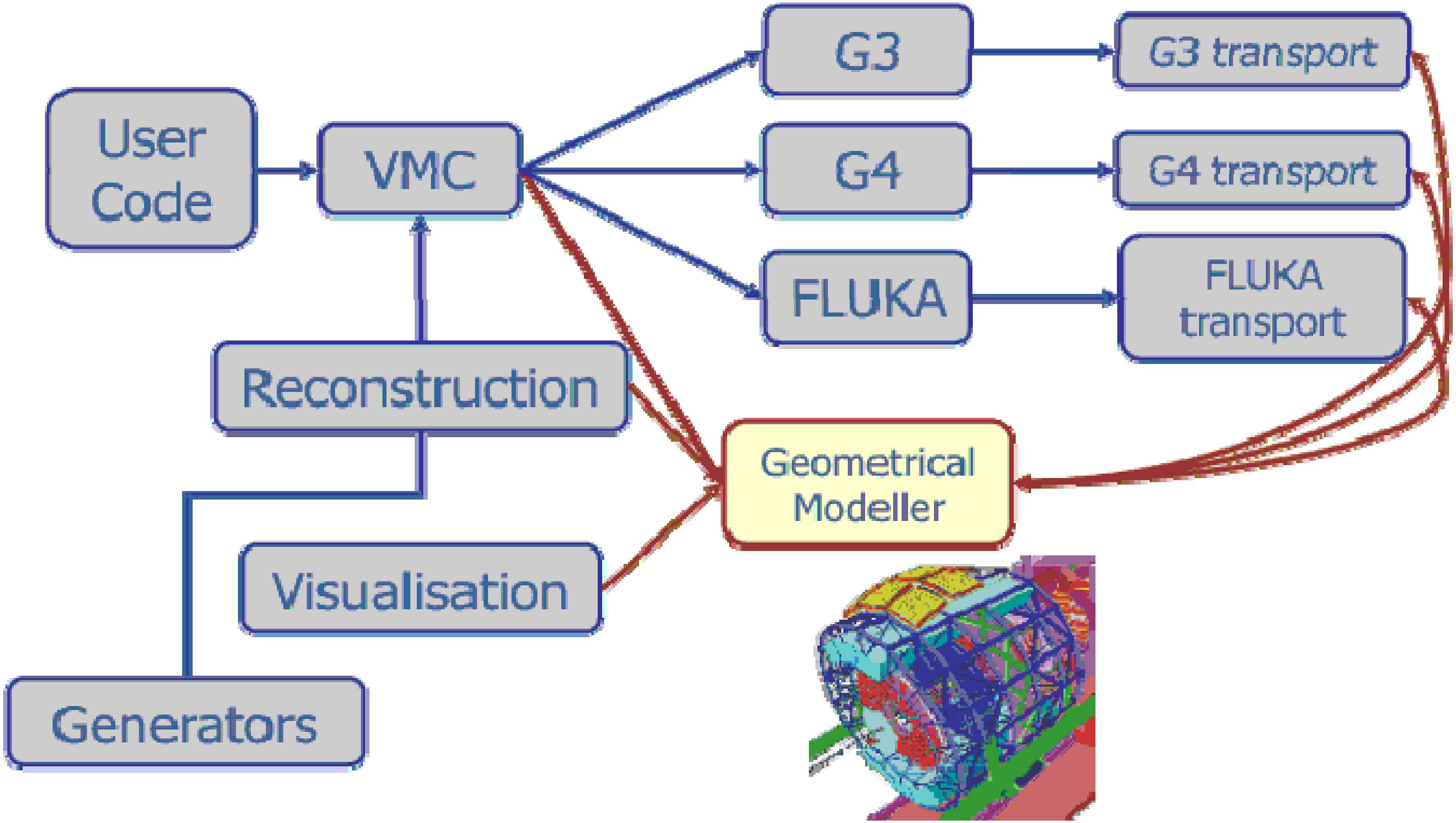
ITS
Low p_t tracking
Vertexing

MUON
μ-pairs

Detectors

- **Types:** Central and Forward Detectors
  - FMD Forward Multiplicity Detector
  - PHOS PHOton Spectrometer
  - TPC Time Projection Chamber
  - TRD Transition Radiation Detector
  - EMCAL Electromagnetic Calorimeter
  - ZDC Zero Degree Calorimeter
  - ACORDE ALICE Cosmic Ray Detector
  - Dimuon Muon Spectrometer
  - ITS Inner Tracking System
  - TOF Time of Flight Detector
  - TZERO or T0 detector with fast trigger capability. Measures the event time with very good precision (< 25 ps)
  - VZERO or V0 used as minimum bias trigger and for rejection of beam gas background
  - PMD Photon multiplicity detector
  - HMPID High Momentum Particle Identification Detector

Virtual MonteCarlo schema



User Code

VMC

G3

G4

FLUKA

G3 transport

G4 transport

FLUKA transport

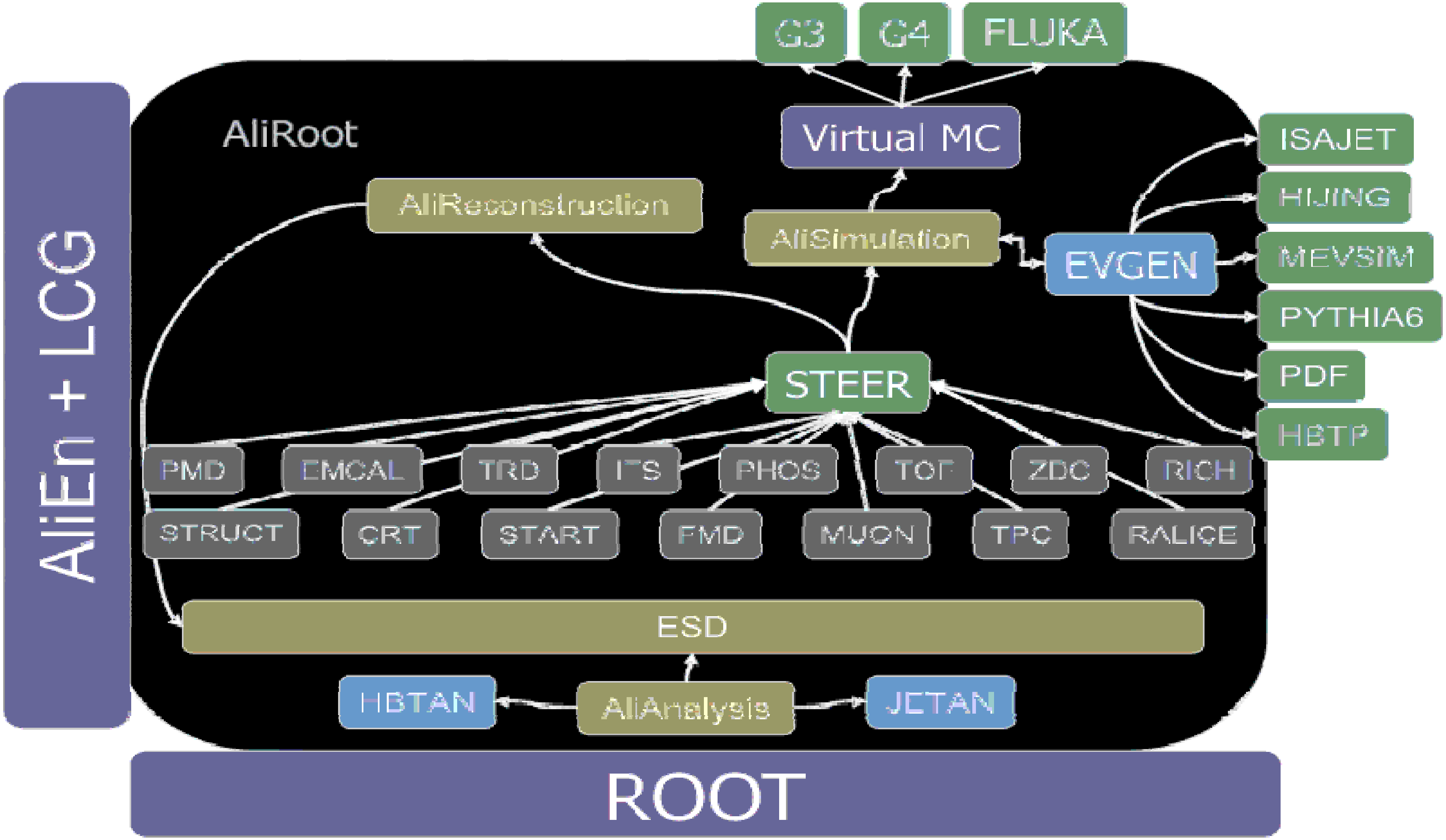
Reconstruction

Visualisation

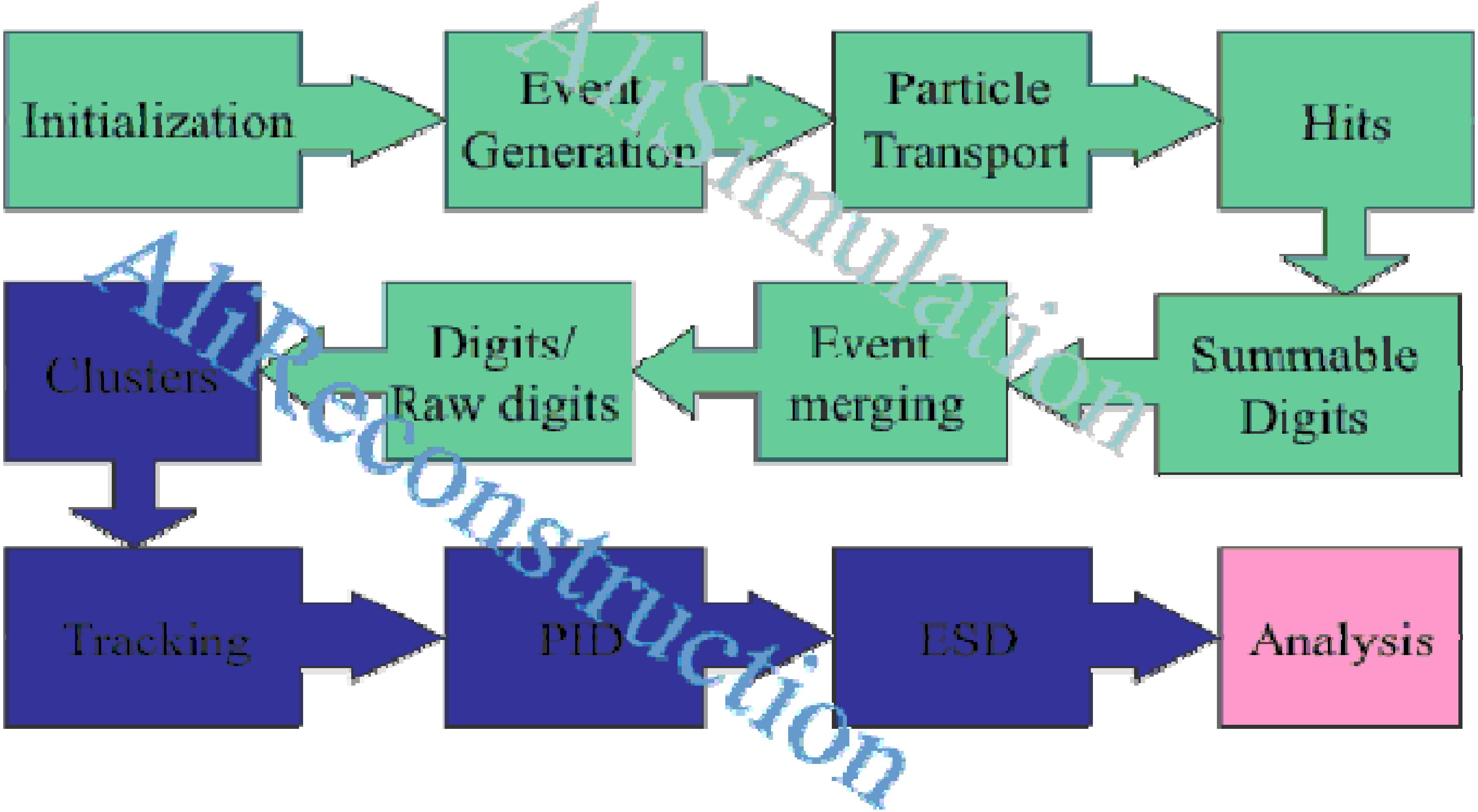
Generators

Geometrical Modeller

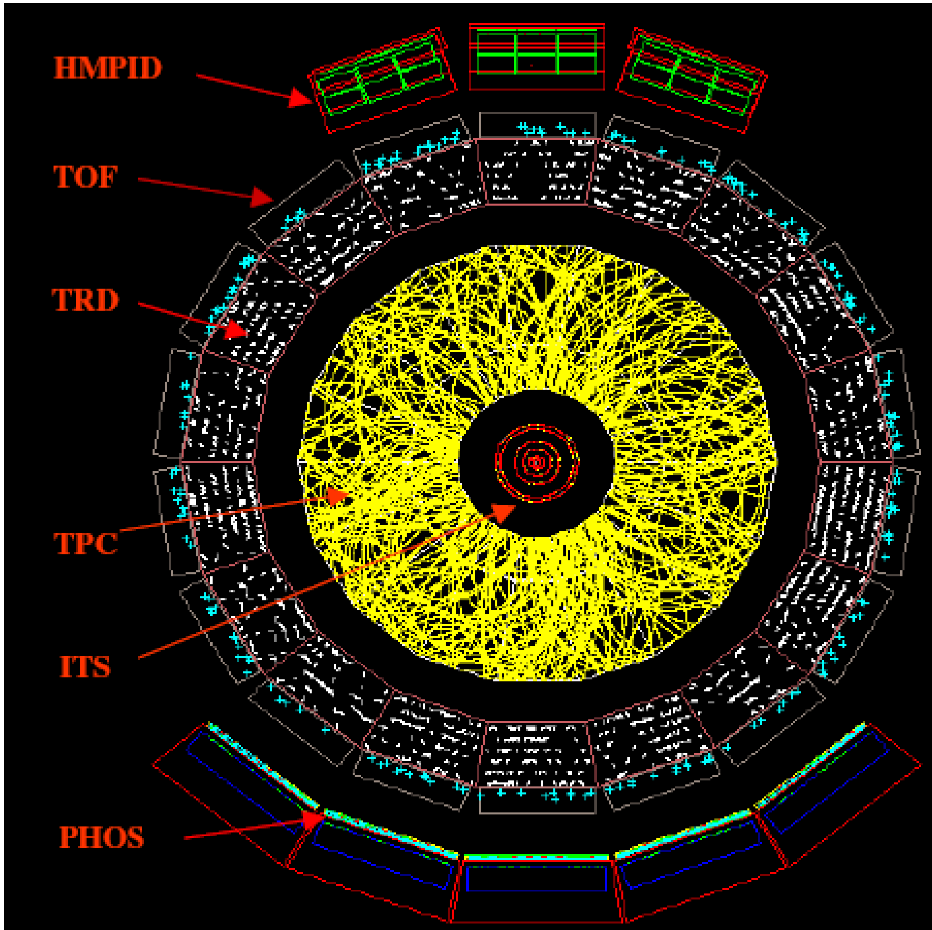
General AliRoot architecture



AliRoot flow of control



AliRoot Simulations

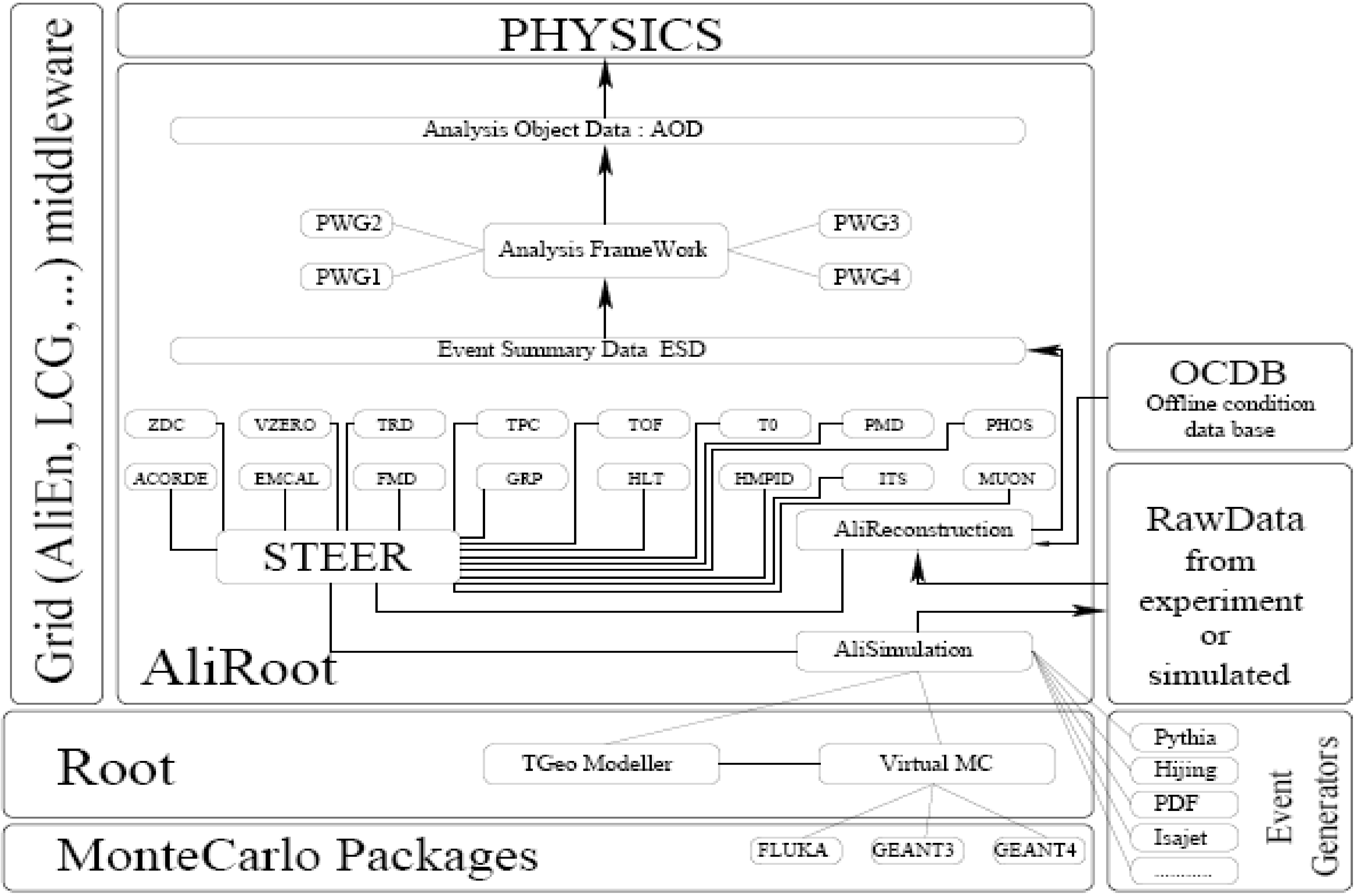


5000 PbPb events
at 5.5 TeV/nucleon pair

dNch/dη ~ 8000

(slice: 2° in θ)

Schematic view of the AliRoot Framework.



What is required to build and from where ?

- **Hardware**: Computer with at least 2GHz processor, Ram: Minimum 1 GB , Recommended 5 GB (more is always good)
- **Operating System:**
- Scientific Linux 4.6
- gcc 3.4.6
- g77 Fortran compiler
- SVN (subversion)

Current Software Check

- To check gcc version do: gcc -v

  gcc version 3.4.6 20060404 (Red Hat 3.4.6-10)

- To check g77 version do: g77 –v
- SVN –v

Update/Install of missing system software

Installation tool is yum

- yum install subversion
- yum install gfortran
- yum install gcc
- yum update –y
- yum install gcc4-gfortran

Second Method:

yum localinstall name-of-tool.rpm
OR
rpm –ivh name-of-tool.rpm

Status of System

Ready for AliRoot Deployment

Summary

(Versions check)

- installed directory : /home/[user name]/Alice

Root	: 5-23-02
pythia	: 6 and 8.1.2.0
CLHEP	: 2.0.4.2
geant 3	: v1-9
geant 4	:4.9.2.p01
aliroot	: v4-02-Release
Fluka	: 2008.3b.0
fluka_vmc	
xerces-c	: 3.0.1-x86-linux-gcc-3.4
alien	: /opt/alien

Deployment

- login as root
- useradd : [yourchoice] ali
- passwd : [what you like and remember]
- su – ali
- mkdir Alice
- cd Alice
- Download the software
- svn co http://root.cern.ch/svn/root/tags/v5-23-02 root

Compile root

- Add bin/ to PATH and lib/ to LD_LIBRARY_PATH.
- For the sh shell family do
  . bin/thisroot.sh
- chmod +x config.sh
- cd root
- export ROOTSYS=`pwd` if(.bashrc is not updated)
- ./configure –help
- ./configure --with-pythia6-uscore=SINGLE --enable-cern --enable-rfio --enable-mathmore --enable-mathcore --enable-roofit --enable-asimage --enable-minuit2 --enable-alien --with-alien-incdir=${ALIEN_ROOT}/api/include --with-alien-libdir=${ALIEN_ROOT}/api/lib
- (g)make
- (g)make install
- Ldconfig
- Root

- testing

- md test
- make
- ./stress

Pythia

- Download pythia from the site http://home.thep.lu.se/~torbjorn/Pythia.html
- tar -xvfz pythia8120.tgz
- cd pythia8120
- ./configure --enable-debug --enable-shared or
- ./configure

In configure file change

- export SHAREDLIBS=no to
- export SHAREDLIBS=yes
- make

Pythia Setting

- In .bashrc of your account :

```
export PYTHIA8=/home/physics/Alice/pythia8120
export PYTHIA8DATA=$PYTHIA8/xmldoc
export LHAPATH=$ALICE_ROOT/LHAPDF/PDFsets
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH\:$PYTHIA8/lib
```

Geant (for **GEometry ANd Tracking**)

- **Geant4** is a toolkit for the simulation of the passage of particles through matter. Its application areas include high energy, nuclear and accelerator physics, as well as studies in medical and space science.
- It provides a complete set of tools for all areas of detector simulation: geometry, tracking, detector response, run, event and track management, visualization and user interface. The multi-disciplinary nature of the toolkit requires that it supply an abundant set of Physics Processes to handle diverse interactions of particles with matter over a wide energy range. For many physics processes a choice of different models is available.
- A large set of utilities is provided, including a powerful set of random number generators. Other features, including a set of physics units and constants, particle management compliant with the Particle Data Group, interfaces to event generators, and object persistency solutions, complete the toolkit.

Geant3

- cd $ALICE
- svn co https://root.cern.ch/svn/geant3/tags/v1-9 geant3
- cd $ALICE/geant3
- export PLATFORM=`root-config --arch`
- gedit ~/Alice/root/bin/root-config
- changed from altf77="" to altf77="f77“ OR gfortran
- make

CLHEP library

- **It is intended to be a set of HEP-specific foundation and utility classes such as random generators, physics vectors, geometry and linear algebra.**
  File name : clhep-2.0.4.2.tgz
- tar zxvf clhep-2.0.3.1.tgz
- cd 2.0.4.2/CLHEP
- echo $ALICE
- ./configure --prefix=$ALICE/CLHEP
- Make
- make check
- make install

QT library and xerces

(necessary for Geant4)

- yum install qt qt-devel
- wget http://mirror.lums.edu.pk/apache/xerces/c/3/binaries/xerces-c-3.0.1-x86-linux-gcc-3.4.tar.gz
- tar -zxvf xerces-c-3.0.1-x86-linux-gcc-3.4.tar.gz   gunzip –d
- mv -r xerces-c-3.0.1-x86-linux-gcc-3.4 /home/[user name]/Alice
- chown -R username.usergroup /home/[user name]/Alice/xerces-c-3.0.1-x86-linux-gcc-3.4
    - set path in bashrc
- export PATH=$PATH\:/home/physics/Alice/xerces-c-3.0.1-x86-linux-gcc-3.4/bin
- export LD_LIBRARY_PATH=$LD_LIBRARY_PATH\:/home/physics/Alice/xerces-c-3.0.1-x86-linux-gcc-3.4/lib:$LD_LIBRARY_PATH

Geant 4

- http://geant4.web.cern.ch/geant4/support/download.shtml
- tar zxvf geant4.9.2.p01.tar.gz
- cd geant4.9.2.p01
- Download Data files from the above link which are :

G4ABLA.3.0.tar.g	G4EMLOW.6.2.tar.gz	G4NDL.0.2.tar.gz
G4NDL.3.13.tar.gz	G4RadioactiveDecay.3.2.tar.gz	PhotonEvaporation.2.0.tar.gz

- mkdir data
- cd data
- tar zxvf ../../G4NDL.3.13.tar.gz
- tar zxvf ../../G4EMLOW.6.2.tar.gz
- tar zxvf ../../PhotonEvaporation.2.0.tar.gz
- tar zxvf ../../G4RadioactiveDecay.3.2.tar.gz
- tar zxvf ../../G4ABLA.3.0.tar.gz

Geant 4

- cd ..
- pwd
- ./Configure –build
  - If success
- Your Geant4 installation seems to be successful!
- ./Configure –install

Maintenance

clean-aliroot : Clean up all aliroot libraries

clean-MODULENAME : Clean everything from module MODULENAME

clean-all : Cleans up everything, including cern libraires

distclean : Like clean-all + clean all tgt_*'s

clean-modules : Clean all module.mk files in all modules

clean-libs : Clean all libraries (not object file

AliRoot

- http://aliceinfo.cern.ch/Offline/AliRoot/Releases.htm
- CD $ALICE
- svn co https://alisoft.cern.ch/AliRoot/trunk AliRoot
- cd $ALICE_ROOT
- make

Fluka

A transport tool

- make all-TFluka
- mkdir myFlukaArea
- mv fluka.tar.gz myFlukaArea/.
- cd myFlukaArea
- gunzip fluka.tar.gz
- tar -xf fluka.tar
- export FLUPRO=/somePath/myFlukaArea

Fluka vmc

- cd $ALICE
- svn co https://alisoft.cern.ch/fluka_vmc/trunk fluka_vmc
- cd fluka_vmc/source
- make
- export FLUKALIB=$ALICE/fluka_vmc/lib/tgt_$ALICE_TARGET
- export LD_LIBRARY_PATH=$FLUKALIB/\:$LD_LIBRARY_PATH

Visualization Software

- OpenGL or MesaGL (free software compatible with OpenGL)
- Qt , a powerfull toolkit
- Open Inventor (free software from SGI)
- Open Scientist (Interactive environment, including GUI)

Download links

- http://root.cern.ch/drupal/
- http://aliceinfo.cern.ch/Offline/AliRoot/Releases.html
- http://geant4.cern.ch/support/download.shtml
- http://proj-clhep.web.cern.ch/proj-clhep/manual/RefGuide/
- http://openscientist.lal.in2p3.fr/
- http://aliceinfo.cern.ch/Offline/AliRoot/Manual.html
- http://aliceinfo.cern.ch/Offline/Activities/EventGeneratorDoc.html
- root http://root.cern.ch
- aliroot https://alisoft.cern.ch/AliRoot/

Tansport packages:

- geant 3 http://cern.ch/geant3
- geant 4 http://cern.ch/geant4
- fluka http://www.fluka.org
- https://twiki.cern.ch/twiki/bin/view/ALICE/PWG2SpectraTopicalPythia8Setup
- http://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/InstallationGuide/html/ch01s03.html

Status of System

95 % Ready

Environment 1

- echo $ALICE_ROOT
  /home/physics/Alice/AliRoot
- **# ROOT and AliRoot**
- #export XERCESCROOT=/home/physics/Alice/xerces-c-3.0.1-x86-linux-gcc-3.4
- export XERCESCROOT=/usr/local
- export ROOTSYS=/home/physics/Alice/root
- export PATH=$PATH\:$ROOTSYS/bin
- export ALICE_TARGET=linux
- export ALICE_TARGET=`root-config --arch`
- export LD_LIBRARY_PATH=$LD_LIBRARY_PATH\:$ROOTSYS/lib
- export LD_LIBRARY_PATH=$LD_LIBRARY_PATH\:$XERCESCROOT/lib
- #export LD_LIBRARY_PATH=$XERCESCROOT/lib:$LD_LIBRARY_PATH
- export LD_LIBRARY_PATH=$LD_LIBRARY_PATH\:$ALICE_ROOT/ANALYSIS
- export LD_LIBRARY_PATH=$LD_LIBRARY_PATH\:$ALICE_ROOT/ANALYSIS/tgt_${ALICE_TARGET}
- #Unix.*.Root.IncludePath: -I$ALICE_ROOT/ANALYSIS/tgt_${ALICE_TARGET}
- **# AliRoot**
- export ALICE=/home/physics/Alice
- export ALICE_ROOT=$ALICE/AliRoot

Environment 2

- #export ALICE_TARGET=`root-config --arch`
- export PATH=$PATH\:$ALICE_ROOT/bin/tgt_${ALICE_TARGET}
- export LD_LIBRARY_PATH=$LD_LIBRARY_PATH\:$ALICE_ROOT/lib/tgt_${ALICE_TARGET}
- **# Geant3**
- #export PLATFORM=`root-config --arch`
- # Optional, defined otherwise in Geant3 Makefile
- export LD_LIBRARY_PATH=$LD_LIBRARY_PATH\:$ALICE/geant3/lib/tgt_${ALICE_TARGET}
- **#GEANT 4**
- ##. /home/physics/Alice/geant4.9.2.p01/src/geant4/.config/bin/Linux-g++/env.sh
- ##. /home/physics/Alice/geant4.9.2.p01/.config/bin/Linux-g++/env.sh
- #. /home/physics/Alice/test/.config/bin/Linux-g++/env.sh
- #. /home/physics/Alice/geant4/.config/bin/Linux-g++/env.sh
- source /home/physics/Alice/geant4/.config/bin/Linux-g++/env.sh

- **# FLUKA**
- export FLUPRO=$ALICE/fluka # $FLUPRO is used in TFluka
- export PATH=$PATH\:$FLUPRO/flutil
- **# Geant4: CLHEP**
- export CLHEP_BASE_DIR=$ALICE/CLHEP
- export LD_LIBRARY_PATH=$LD_LIBRARY_PATH\:/home/physics/Alice/CLHEP/lib

Environment 3

- ##############**pythia8120**
- export PYTHIA8=/home/physics/Alice/pythia8120
- export PYTHIA8DATA=$PYTHIA8/xmldoc
- export LHAPATH=$ALICE_ROOT/LHAPDF/PDFsets
- export LD_LIBRARY_PATH=$LD_LIBRARY_PATH\:$PYTHIA8/lib
- ###################
- export FLUKALIB=$ALICE/**fluka_vmc/**lib/tgt_${ALICE_TARGET}
- export LD_LIBRARY_PATH=$FLUKALIB/\:$LD_LIBRARY_PATH
- ######**path xml par xerces**########
- export PATH=$PATH\:/usr/local/bin
- export LD_LIBRARY_PATH=$LD_LIBRARY_PATH\:/usr/local/lib

Environment 4

- #export ALICE_TARGET=linux
- export PLATFORM=linux
- export ALIEN_ROOT=/opt/alien
- export ALIEN=$ALIEN_ROOT
- #export X509_CERT_DIR=/opt/alien/globus/share/certificates/
- export PATH=$PATH\:/opt/alien/bin:/opt/alien/api/bin
- export GSHELL_ROOT=$ALIEN/api
- export GLOBUS_LOCATION=/opt/globus
- export GSOAP_LOCATION=/opt/alien/gsoap
- export CGSI_GSOAP_LOCATION=/opt/alien
- #export CLASSAD_LOCATION=/opt/alien
- export PATH=$PATH\:$ALIEN/bin:$GSHELL_ROOT/bin:$GLOBUS_LOCATION/bin:/opt/alien/gsoap/bin:/usr/local/Trolltech/Qt-4.5.2/bin
- ######################**Qt-4.5.2**#############
- #export PATH=/usr/local/Trolltech/Qt-4.5.2/bin:$PATH
- #export LD_LIBRARY_PATH=/usr/local/Trolltech/Qt-4.5.2/lib
- export LD_LIBRARY_PATH=$LD_LIBRARY_PATH\:/usr/local/Trolltech/Qt-4.5.2/lib
- **###For visualization one hast to find the system libraries for x-windows and so on #####**
- export OGLHOME=/usr/X11R6
- export OGLHOME1=/usr

Environment 5

- export OGLFLAGS="-I$OGLHOME/include"
- export OGLLIBS="-L$OGLHOME1/lib -lGL -lGLU -L$OGLHOME/lib -lXp -lXm -lXext -lXt -lX11 -lXmu"
- export TCL_LIBRARY=/usr/lib
- export TK_LIBRARY=/usr/lib
- export IVROOT=/usr/local/src
- export LD_LIBRARY_PATH=$LD_LIBRARY_PATH\:$IVROOT/usr/lib
- export LD_LIBRARY_PATH=$LD_LIBRARY_PATH\:$IVROOT/usr/lib/InventorDebug
- export PATH=$PATH\:$IVROOT/usr/bin

Status of System

All Done.

I/O Design 1

Based on loaders

- Loader responsibility: opening/closing files; posting/accessing data
- Run loader: work with central data (AliRun object, event header, kinematics, stack, track references, AliRunLoader object itself)
- Detector Loader: works with detector specific data(hits, sdigits, digits, rec. points, tracks)

I/O Design 2

Architecture

- AliRunLoader has a list of AliLoaders
- Each detetor registers its Loader to the list
- Each detetors sees pointer to its Loader for fast access but does not own the Loader. This pointer is restored automatically if one reads the detector from a file.
- Each data type (hits, digits etc.) is defined completely by instane of AliLoaderDataInfo class
- **AliRunLoader** : you call some of the *core classes* of AliRoot to write the file galice.root that will know everything about the architecture of the output files. The *galice.root* is a sort of manager of the output files.
- **TVirtualMCDecayer** : it's an interface class (virtual class) for the Decayer you want to use.

Opening an aliroot session (gAlice, gMC, gRandom,gSystem)

- **gSystem** allows an interaction beetween AliRoot and your operative system, for instance to load more libraries. loads the library of Geant3, Geant4 it is one of the transport softwares the offline group chose to simulate the interaction of the particles whan they pass through materials.
- **gRandom** allows you to use random generators for the distribution you need (flat,gaussian). starts the random generators with a seed not defined (if you want to simulate more events it's important not to write a fixed number for the seed).
- **gMC** is necessary for the Monte Carlo simulation. this is the messenger to whom you have to say which physical processes you want to include in your simulation and which cuts you want to introduce.
- **gAlice** is involved in the global management of the input and output files of AliRoot.
- gAlice->Init()
- gAlice Run(3)

AliRoot initialization

- AliRoot (as Root) is entirely based on object oriented C++.
- Functions, Macros and Commands must follow the standard C++ sintax (Cint) .
- The event generation is handled via a configuration file (that is also a C++ script) Config.C which

- Initializes Geant3,
- Initializes the Generator
- Switch on/off detectors and magnetic. field
- Set detectors parameters

Simulation (1 event)

- How to do simulation and reconstruction:
- AliSimulation

- Simulate 1 event with Config.C (macro for generator + detector setup):

- AliSimulation sim;
- sim.Run();

Simulation (5 event)

- Simulate 5 events with ConfigPPR.C:
- AliSimulation sim("ConfigPPR.C");
- sim.Run(5);

Reconstruction

- AliReconstruction
- Reconstruct events in galice.root:
- AliReconstruction rec;
- rec.Run();

Running Simulation/Reconstruction

- root [0]AliSimulation *pSim = new AliSimulation(''Config.C''); root [1] pSim->Run(1);
- root [2]delete *pSim;
- root [3]AliReconstruction *pRec=new AliReconstruction;
- root [4]pRec->SetRunLocalReconstruction(“PHOS");
- root [5]pRec->Run();
- root [6]delete *pSim;
- root [7]AliReconstruction *pRec=new AliReconstruction;
- root [8] pRec->SetRunLocalReconstruction("ITS TPC TRD TOF");
- root [9]pRec->Run();

PDG code

- root [0] kProton (const enum PDG_t)2212
- root [1] kProtonBar (const enum PDG_t)(-2212)
- root [2] kKPlus (const enum PDG_t)321
- root [3] kKMinus (const enum PDG_t)(-321)
- root [4] kK0 kK0Long kK0Short kK0 kK0Bar
- root [5] kK0Bar (const enum PDG_t)(-311)
- The mesons have 3-numbers pdg code, while the barions have 4-numbers pdg code.
  Nevertheless there is the *kPhiKK* that corresponds to a number different from the *pdg-code* one, that is *333*

Simulate ? With ?

- **Events** : Pb-Pb collisions or p-p collisions
- **particles** : one particle, more than one particle of the same type, or a flux of different particles
- Pythia or Hijing : Pythia for p-p collisions, Hijing for Pb-Pb collisions
- generate *particles* $ALICE_ROOT/EVGEN .
  - *AliGenBox.h* : generates particles of the same type in a box-type momentum interval;
- *AliGenFixed.h* : generates particles of the same type with fixed momentum;
- *AliGenLib.h* : generates particles of the same type using their properties (meanlife, etc.), it is a sort of register that knows everything about particles.;
- *AliGenRICHlib.h* : generates particles useful just for the RICH detector analysis;
- *AliGenCocktail.h* : generate a cocktail of different particles.

Hijing

Generator

- It generates a cocktail of 2 protons (box-type momentum) going to each detector module and 510 particles generated by Hijing as if they were coming from a Pb-Pb collision. This is useful when it's not necessary to simulate a full Hijing event, but just a sort of *background* for the identification of particles.
- AliGenHIJINGpara *pHijingPara=new AliGenHIJINGpara(510);
-   pHijingPara->SetMomentumRange(0,99);
-   pHijingPara->SetThetaRange(60,120);
-   pHijingPara->SetPhiRange(0,60);
-   pHijingPara->SetOrigin(0,0,0);
-   pHijingPara->SetSigma(0,0,0);
-   pCocktail->AddGenerator(pHijingPara,"HijingPara",1);
-   pCocktail->Init();

Pythia

Generator

- AliGenPythia *pPythia = new AliGenPythia(-1);
- pPythia->SetMomentumRange(0,999999);
- pPythia->SetPhiRange(0,360);
- pPythia->SetThetaRange(0,180);
- pPythia->SetYRange(-12,12);
- pPythia->SetPtRange(0,1000);
- pPythia->SetStrucFunc(kCTEQ4L);
- pPythia->SetProcess(kPyMb);
- pPythia->SetEnergyCMS(14000);
- pPythia->SetTrackingFlag(0);
- pPythia->Init();

Thanks

Extra Slides

Exercise

- ssh –p 3145 alice01@
- mkdir [yourname]
- cd [yourname]

Wait for instructions !!!!

Sim-Reco

- 2. Aliroot Simulation and Reconstuction:

- Simulation Reconstruction

- Monte Carlo:

Particles --> Hits -->SDigits -->Digits -->

- Clusters--> Tracks--> ESD
- Real Data:

Detector --> DAQ -------->Raw Data -->

Problem
Configuring with Pythia

- touch lib/libPythia.so lib/libPythia6.so
- ./configure linux --with-pythia-libdir=$ROOTSYS/lib
- make
- Error message: make: *** [lib/libEGPythia.so] Error 1
- then execute
- g++ -shared -Wl,-soname,libEGPythia.so -O -o lib/libEGPythia.so pythia/src/TMCParticle.o pythia/src/TPythia.o pythia/src/G__Pythia.o
- make
- ./configure linux --with-pythia6-libdir=$ROOTSYS/lib
- make

Problem
Configuring with Pythia

- Error message: make: *** [lib/libEGPythia6.so] Error 1
- then execute
- g++ -shared -Wl,-soname,libEGPythia6.so -O -o lib/libEGPythia6.so pythia6/src/TMCParticle6.o pythia6/src/TPythia6.o pythia6/src/G__Pythia6.o
- make

How to read .root files

- The time sequence of the creation of the files :
  galice.root -> Kinematics.root -> Hits.root -> SDigits.root -> Digits.root -> Tracking Algorithm -> AliESDs.root
  Before the AliESDs.root files, all the informations about particles, hits, digits and tracks are divided into 8 categories.
  The categories are : Kinematics, Hits, SDigits, Digits, Reconstructed Points, Tracks, Reconstructed Particles.
  For instance a cluster of pads in some detector is not a SDigit or a Digit, it is something *reconstructed*.
  So, first af all you have to focus on which kind of information you want to retrieve :

- Do you need the momentum of a particle at the interaction point?

- Do you need to know which are the daughters of a particle has decayed in? => You need informations from Kinematics.root

- Do you need to know if the particle is a daughter or a mother?

  Do you need the energy loss in some device of the detector?

- Do you need the pads involved? => You need informations from Hits.root

  Do you need the the charge response of the detector? => You need informations from Digits.root

  Do you want to know how many pads a cluster is made of?

- Do you want to know which is the charge of the cluster? => You need informations from RecPoints

  Do you need informations about the reconstructed tracks?

- Do you need informations about each detector response? => You need informations from AliESD.root

AliEn

- ALIEN_ROOT=/opt/alien   if(.bashrc is not updated)
- create dir as root:
- ls -ld /opt/alien/
- drwxr-xr-x 2 root root 4096 Apr 3 09:25 /opt/alien/
- chmod 777 /opt/alien
- cd /opt/alien

AliEn Cont!

- Installation as user [you created : say ali]:
- su - ali
- wget http://alien.cern.ch/alien-installer
- chmod +x alien-installer
- ./alien-installer
- alein version v2-16
- default workspace location : /home/physics/.alien/cache
- platform:
- i686-pc-linux-gnu
- installation location: /opt/alien

Tags

- http://aliceinfo.cern.ch/Offline/Activities/Reconstruction/RunEventTagSystem/EventTagsCreation.html

Script

- #!/bin/sh
- #Compilation of ROOT
- make
- make install
- Save above as anyname.sh
- chmod +x anyname.sh
- ./anyname.sh

System commands

- To unzip any file
- tar -zxvf [filename].tar.gz
- gunzip fluka.tar.gz
- tar -xf fluka.tar

Links

- http://morsch.web.cern.ch/morsch/AliGenerator.html
- http://aliceinfo.cern.ch/static/aliroot-new/html/roothtml/ClassIndex.html