Software Developments from CHEP

David Brown, LBNL

CHEP 2010

- First Post-LHC startup CHEP
- ~75% of conference related to LHC
  - subjects, presentations and participants
- No (major) new LHC developments
  - Major decisions made years ago
  - Technologies are mostly frozen (for now)
  - Operations phase issues not relevant for SuperB
- LHC computing discussed at Ferrara workshop
- ⇒ I will not discuss LHC issues today

CHEP 2010

- SuperB presentations
  - “Fast Simulation for SuperB” (poster)
  - “Distributed Production System for SuperB” (poster)
  - “Computing for Flavor Factories” (A. Fella plenary)
- Parallel streams I covered
  - “Event Processing”
  - “Software Engineering, datastore, databases”
- Topics of (personal) interest
  - Simulation and Reconstruction
  - Frameworks
  - New experiments: Fair (Panda), Belle II, STAR, ILC

The PandaRoot Framework for Simulation, Reconstruction and Analysis

Stefano Spataro

for the panda collaboration

UNIVERSITÀ DEGLI STUDI DI TORINO
ALMA UNIVERSITAS TAURINENSIS

ISTITUTO NAZIONALE DI FISICA NUCLEARE
Sezione di Torino

CHEP 2010 TAIPEI TAIWAN

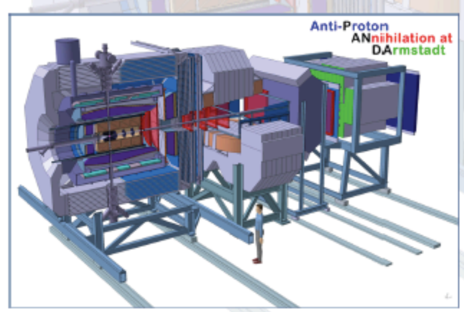
Tuesday, 19th October, 2010

The Panda experiment

AntiProton Annihilations at Darmstadt

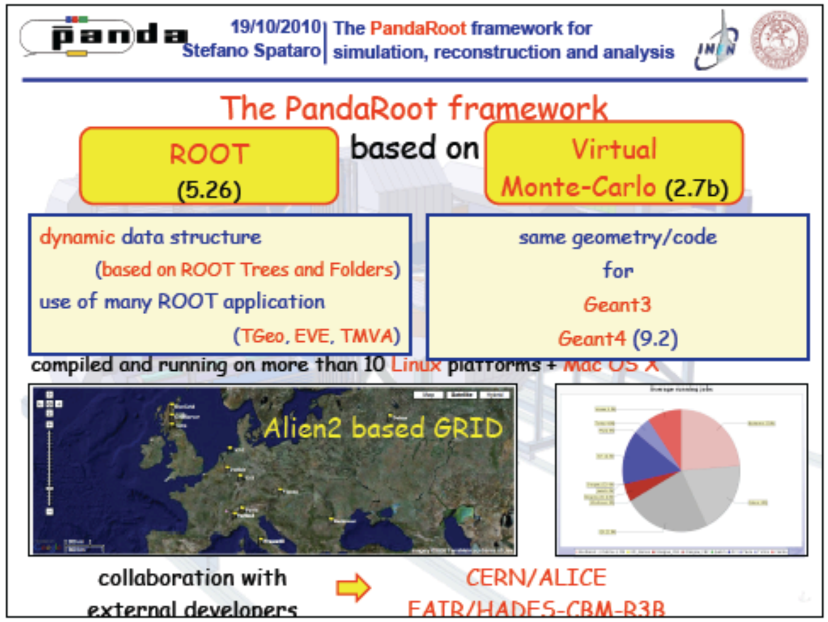
Multi purpose detector at FAIR

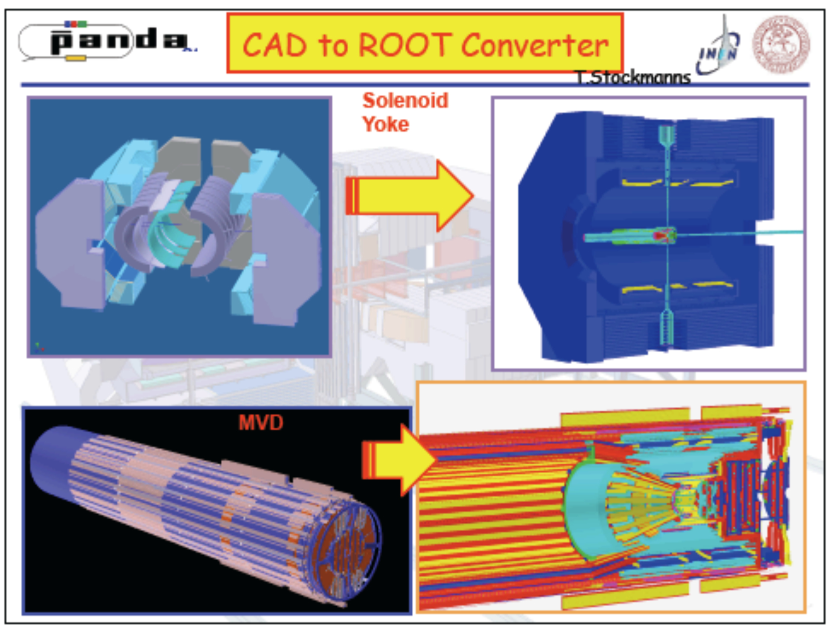
$\bar{p}$p, $\bar{p}$A collisions
1.5⇒15 GeV/c ($\bar{p}$ momentum)

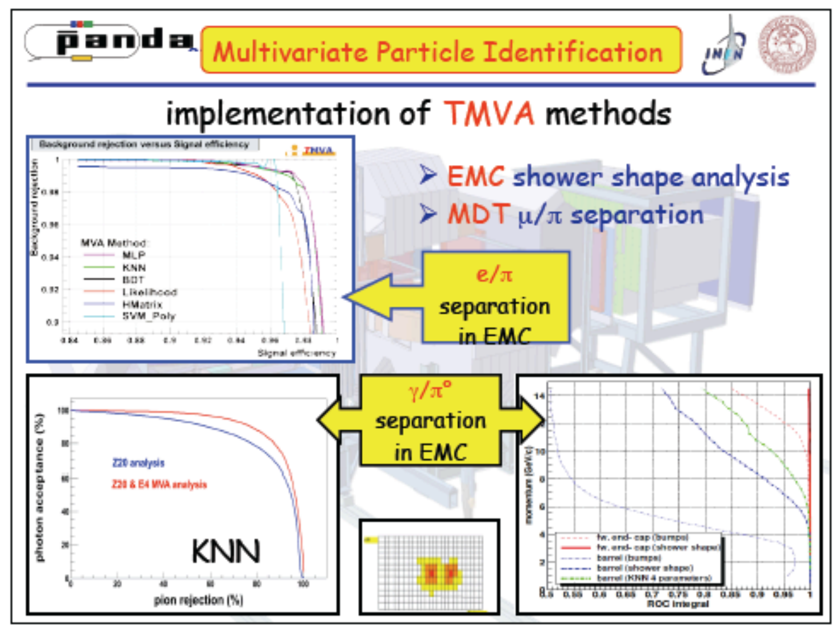


Physics program

- Charmonium ($c\bar{c}$) spectroscopy
- Open charm spectroscopy
- Search for gluonic excitations (hybrids - glueballs)
- Charmed hadrons in nuclei
- Drell-Yan
- Single and double Hypernuclei
- Other options (Parton Distrib., EM Form Factor...)


panda
19/10/2010
Stefano Spataro
The PandaRoot framework for simulation, reconstruction and analysis
INFN
The PandaRoot framework
ROOT (5.26)
based on
Virtual Monte-Carlo (2.7b)
dynamic data structure
(based on ROOT Trees and Folders)
use of many ROOT application
(TGeo, EVE, TMVA)
same geometry/code
for
Geant3
Geant4 (9.2)
compiled and running on more than 10 Linux platforms + Mac OS X
Alien2 based GRID
collaboration with
external developers
CERN/ALICE
FAIR/HADES-CBM-R3B


panda
CAD to ROOT Converter
INFN
T.Stockmanns
Solenoid
Yoke
MVD


panda
Multivariate Particle Identification
INFN
implementation of TMVA methods
Background rejection versus Signal efficiency
TMVA
MVA Method:
MLP
KNN
BDT
Likelihood
HMatrix
SVM_Poly
Signal efficiency
EMC shower shape analysis
MDT μ/π separation
e/π
separation
in EMC
γ/π°
separation
in EMC
photon acceptance (%)
Z20 analysis
Z20 & E4 MVA analysis
KNN
pion rejection (%)
ROC Integral

CHEP 2010 Taipei Taiwan

2010/10/21

Andreas Moll

Max-Planck-Institut für Physik

The Software Framework of the Belle II Experiment

- The Belle II experiment
- Basic architecture
- Geometry and simulation

Max-Planck-Institut für Physik (Werner-Heisenberg-Institut)

MAX-PLANCK-GESELLSCHAFT

Belle II

TUM LMU Excellence Cluster Universe

From BASF to basf2

Software framework of BELLE: **BASF** (**B**elle **A**nalysi**S** **F**ramework)

Successfully used since over **10 years** for

Online:
- **HLT** (High level trigger)
- **DAQ** (Data acquisition)
- **DQM** (Data quality monitor)

Offline:
- BELLE detector **optimization**
- Physics **analysis**

\+ **Proven** concepts

\+ Combines a lot of knowledge (**treasure chest**)

− Highly **optimized** for BELLE (*hard to adopt it to Belle II*)

− Lack of **object persistency** (not object oriented)

Development of a new software framework for Belle II has started

basf2

Design choices

The **design** and **architecture** of the Belle II framework is driven by:

Resources:

- ✓ Take the best ideas/concepts from other frameworks *(BASF, ILC, Gaudi, CDF, ALICE)*
- ✓ Use proven third-party libraries *(ROOT, boost, libxml)*
- ✓ Modularization *(distribute work, keep maintenance work of the core low)*

boost C++ LIBRARIES

Technology choices:

- ✓ Framework core and libraries: **C++**
- ✓ Steering/Control: **Python**
- ✓ Parallel processing

Requirements:

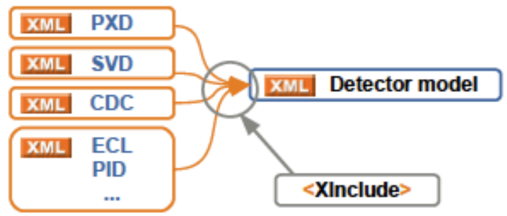
- ✓ Same framework used for both **online** (HLT, DQM) **offline** event processing

Geometry Parameter storage

Parameters describing the geometry: **XML**

Creation of the actual geometry (volumes): **C++**

- One **XML** file per subdetector
- One **XML** file describing a specific detector model



Parameter access:

- Library with **abstract interface** + different backend implementations *(currently XML, can be changed later to a database or a mixture of both)*
- Queries for parameters are carried out using **XPath**:

```
getParamLength("/Detector/Subdetector[@type='PXD']/Content/Laye
```

Generic concept: System can be used for any kind of paramete

Geant4 simulation

The ROOT **TGeo** geometry is converted to **Geant4** native volumes using **G4Root**

see PS02-1-223

Passive parts: done automatically by G4Root

Sensitive parts: by volume name prefix: e.g. "**SD_**"

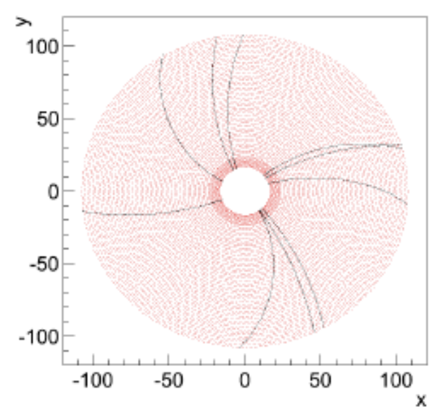
user connects prefix to **sensitive detector** in geometry creator class (C++ code)

Example:

10 π^+ tracks

0.3 GeV < p < 1.3 GeV

Only wire drift chamber drawn



Clas12 Reconstruction and Analysis Framework

SOA based physics data processing (PDP)

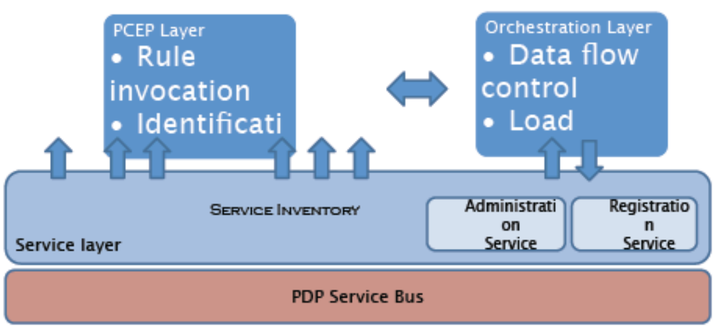
V. Gyurjyan†

11/15/10

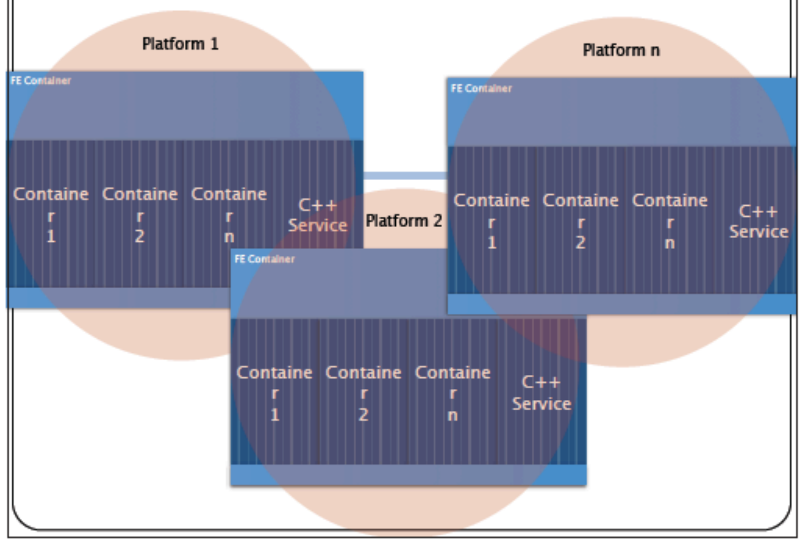
ClaRA

- SOA based physics data production application development framework, written in pure Java.
- Service development environment.
- Increase intrinsic interoperability of services by standardizing data exchange interface.
- Complex service composition.
- Clear separation between PDP application designer and service programmer.
  - Build and run PDP applications without an access to the source code of individual services.
- Increase federation.
  - Services and ClaRA based applications are united while maintaining their individual autonomy and self governance.
- Multi-Threaded event processing.
- Distributed event processing.
- Ease of application deployment.
- PDP application diversification and agility.

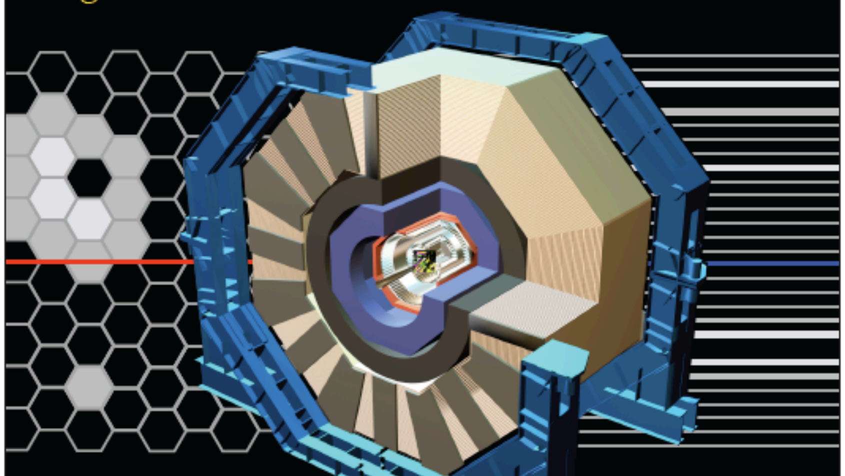
ClaRA Design Architecture



ClaRA Cloud



org.lcsim: Event Reconstruction in Java



Norman Graf (SLAC)
For the ALCPG Physics & Detector
Simulation & Reconstruction Group

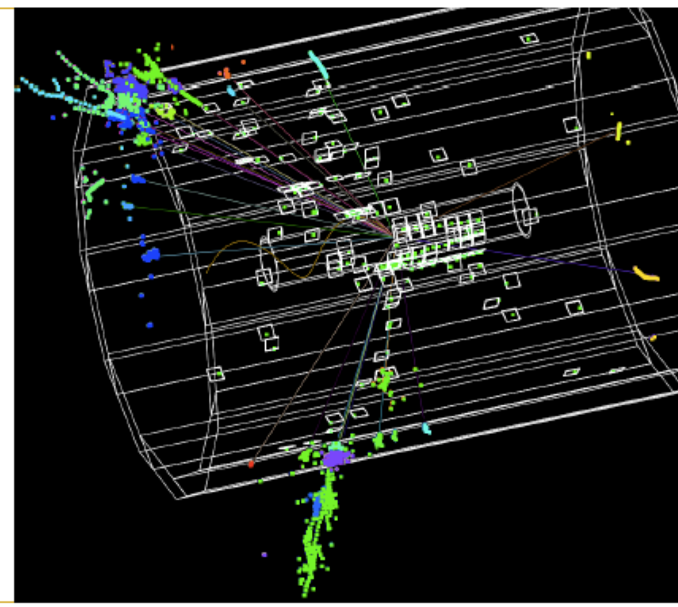
CHEP 2010
Taipei, Oct. 21, 2010

LCD Simulation Mission Statement

- Provide full simulation capabilities for Linear Collider physics program:
  - Physics simulations
  - Detector designs
  - Reconstruction and analysis
- Need flexibility for:
  - New detector geometries/technologies
  - Different reconstruction algorithms
- Limited resources demand efficient solutions, focused effort.
  - Strong connections between university groups, national labs, international colleagues.

Fast Detector Response Simulation

- Covariantly smear tracks with matrices derived from geometry, materials and point resolution using Billoir's formulation.
  - Analytically from geometry description.
- Smear neutrals according to expected calorimeter resolution (EM for γ, HAD for neutral hadrons)
  - Derived from full Geant4 simulations
- Create reconstructed particles from tracks and clusters (γ, e, μ from MC, $\pi^{+/-}$, K^0_L for others)
- Can also dial in arbitrary effective jet energy resolution.
  - Derived from full simulation, reconstruction & analysis.



Mu2e-doc-1127-v4

art: A Framework For New, Small Experiments at Fermilab

Rob Kutschke,
Fermilab CD and the Mu2e Collaboration
CHEP 2010, Academia Sinica, Taipei
October 21, 2010

The Candidates

- Existing FNAL CD supported products:
  - D0, CDF, CMS, MiniBoone, MINOS …
- Third party products
  - FMWK (ROOT based; MIPP, early NOvA)
  - ALIROOT / ILCROOT family
  - GAUDI
  - JAS
- With O(2 FTE), CD cannot support a third party product.
- CD recommended evolving the CMS framework
  - Most modern of the 5 FNAL candidates.
  - C++ based.
  - Remove/replace features if of little benefit to small experiments and hard to use and/or maintain.

Retained Features

- The state machine.
- Module (Producer/Analyzer/Filter/IO) and Service base classes.
- Three part event ID
- EDM: in memory and ROOT Tree based persistency.
- Persistent objects: Event/Run/SubRun (SubRun=LuminosityBlock).
- Four part data product ID.
- ParameterSet mechanism.
- Python runtime config – will change.
- Reconstruction on demand.
  - Scheduled reconstruction also retained for now.
- Data product provenance.
- Exception handling strategy: action orthogonal to throw
- TFileService
- Message logger

Features (to be) Removed

- EventSetup
  - Dummied out and will be removed.
  - Conditions data can be adequately managed via Services.
- Event merging/overlay
  - Removed from Source modules; now done in producers.
  - Factor into a bookkeeping problem and a physics problem.
- References across data products
  - Will develop something similar to CLEO III Lattice.
  - This puts the complexity in the right place.
- Matching module names to .so file name
  - We have a much smaller problem that will admit a solution based on the relative of the source file (relative to root of a release).

Track Finding in a High-Rate Time Projection Chamber Using GPUs

Felix Böhmer
for the GEM-TPC Collaboration

Physik Department E18
Technische Universität München
Germany

CHEP 2010, October 19,
Taipei, Taiwan

Maximum Search in 5 Dimensions

H. Li, M. A. Lavin, R. J. Le Master: *"Fast Hough Transform: ..."*, 1986

Technische Universität München

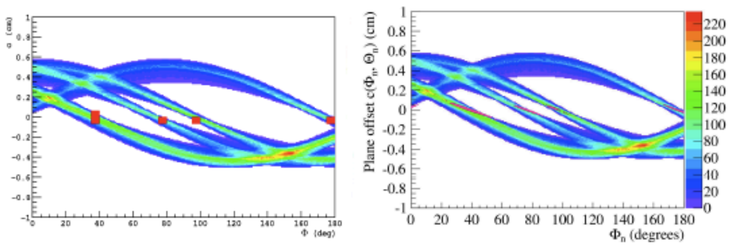
... is not trivial:

- Conventional histogramming methods fail (memory consumption!)

Instead: trade memory consumption for processing load

- Subdivide the parameter space into subvolumes *"nodes"*
- Perform an iterative tree search
- Example of 5 simulated tracks:

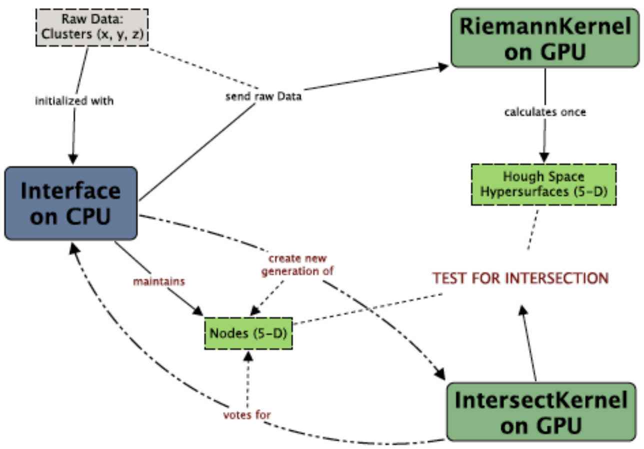
10 iterations



Monte Carlo Truth   (m, c) projection

Implementation on a GPU using CUDA^{TM}

Technische Universität München



Track Reconstruction in the STAR TPC with a Cellular Automaton Based Approach

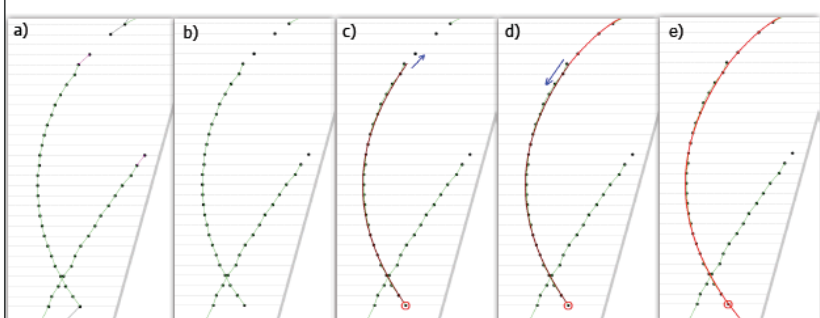
Y. Fisyak[1], I. Kisel[2], I. Kulakov[2], J. Lauret[1], M. Zyzak[2]

[1]BNL, Brookhaven, USA; [2]GSI, Darmstadt, Germany

CHEP-2010
Taipei, October 18-22, 2010

Cellular Automaton (CA) Track Finder Algorithm in STAR TPC

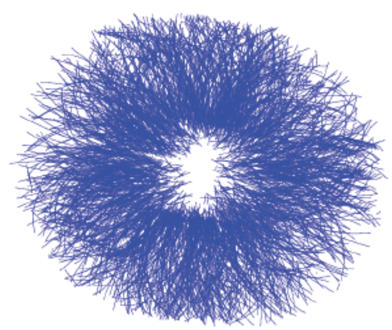
1. Reconstruction of track segments in each TPC sector:
   a) Find and link neighbors hits
   b) Clean links
   c) Create segments by fitting chains and adding outer hits
   d) Refit tracks and add inner hits
   e) Selection of tracks
6. Merge sector tracks into TPC global tracks.



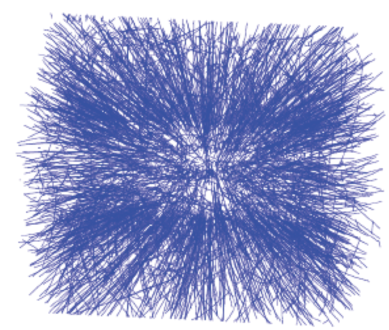
Based on the ALICE HLT TPC CA Track Finder

CA Tracker: Au-Au Event with 1446 Tracks

Front view



Side view



GENFIT

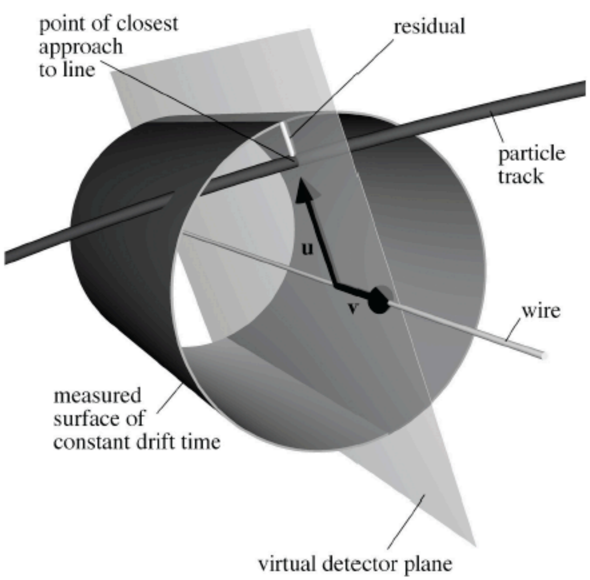
A novel generic framework for track fitting in complex detector systems

Christian Höppner - Technische Universität München

International Conference on Computing in High Energy and Nuclear Physics, Taipei, October 2010

Virtual Detector Plane - Drift Chamber

fit minimizes distance to surface of constant drift time



Fitting Algorithms

- Standard algorithm in GENFIT: extended Kalman Filter
  [*R. Frühwirth, NIMA 262 (1987) 444-450*]
- Recent implementation in GENFIT: Deterministic Annealing Filter
  [*R. Frühwirth & A. Strandlie, CPC 120 (1999) 197-214*]

  rejection of noise hits in planar detectors and outliers in TPC works well
- Other algorithms like Gaussian Sum Filters can be easily implemented

New Developments in the ROOT Mathematical Libraries

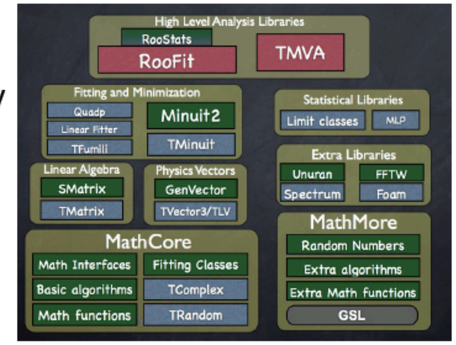
L. Moneta, C. Gumpert, B. Rabacal
(CERN, PH-SFT)

ROOT
An Object-Oriented Data Analysis Framework

CHEP
TAIPEI TAIWAN 2010

Outline

- Recent developments in ROOT libMathCore
  - numerical algorithms interfaces
- Fitting improvements
  - `TFitResult`
- New classes in *Hist* library
  - `TEfficiency` class
  - `TKDE` class for density estimation
- Goodness of Fit
  - new `GoFTest` class
- Conclusions



CHEP
TAIPEI TAIWAN 2010

New Data Libraries and Physics Data Management Tools

Mincheol Han[1], Chan-Hyeung Kim[1], Hee Seo[1]
Lorenzo Moneta[2]
Maria Grazia Pia[3]

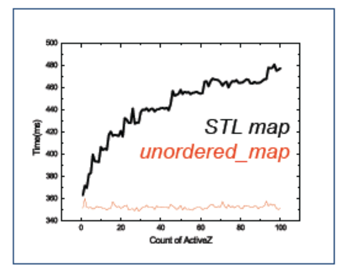
[1] *Hanyang University, Seoul, Korea*
[2] *CERN, Geneva, Switzerland*
[3] *INFN Sezione di Genova, Italy*

CHEP 2010
Taipei, 18-22 October 2010

Use forthcoming C++ features

- Current implementation uses STL *map* for most data, STL *vector* for a few data types
- Evaluated ***unordered_map*** (AKA hash map)
- Included in **C++0x TR1**
- *<tr1/unordered_map>* gcc 4.3.x
- *<unordered_map>* in MSVC

Pair production cross sections



time (ms) to **load** data
vs. number of elements present in the experimental set-up

ROOT I/O: The Fast and Furious

CHEP 2010: Taipei, October 19.
Philippe Canal/FNAL,
Brian Bockelman/Nebraska,
René Brun/CERN,

Without Prefetching Baskets

- Default was for all buffers to have the same size.
- Branch buffers are not full at the same time.
  - A branch containing one integer/event and with a buffer size of 32Kbytes will be written to disk every 8000 events.
  - while a branch containing a non-split collection may be written at each event.
- Without *TTreeCache*:
  - Many small reads.
- When reading with prefetching there were still inefficiencies:
  - Backward seeks.
  - Gap in reads.
- Hand tuning the baskets which was feasible with a dozens branches became completely impracticable for *TTree* with more than 10000 branches.

Solution: *TTreeCache*

- Prefetches and caches a set of baskets (from several branches).
- Designed to reduce the number of file reads (or network messages) when reading a *TTree* (by a factor of 10000).
- Configuration

```
T->SetCacheSize(cachesize);
if (cachesize != 0) {
   T->SetCacheEntryRange(efirst,elast);
   T->AddBranchToCache(data_branch,kTRUE); // Request all the sub branches too
   T->AddBranchToCache(cut_branch,kFALSE);
   T->StopCacheLearningPhase();
}
```

Conclusions

- Other new experiments are dealing with the same (software) issues as SuperB
  - Legacy code
  - Need improved performance for large data sets
  - Want to exploit recent developments and trends
- They are ahead of us
  - we should learn/collaborate/steal from them
- Root continues to take over HEP software
  - resistance is futile
- Improving standards, tools (boost) means less HEP-specific software needed