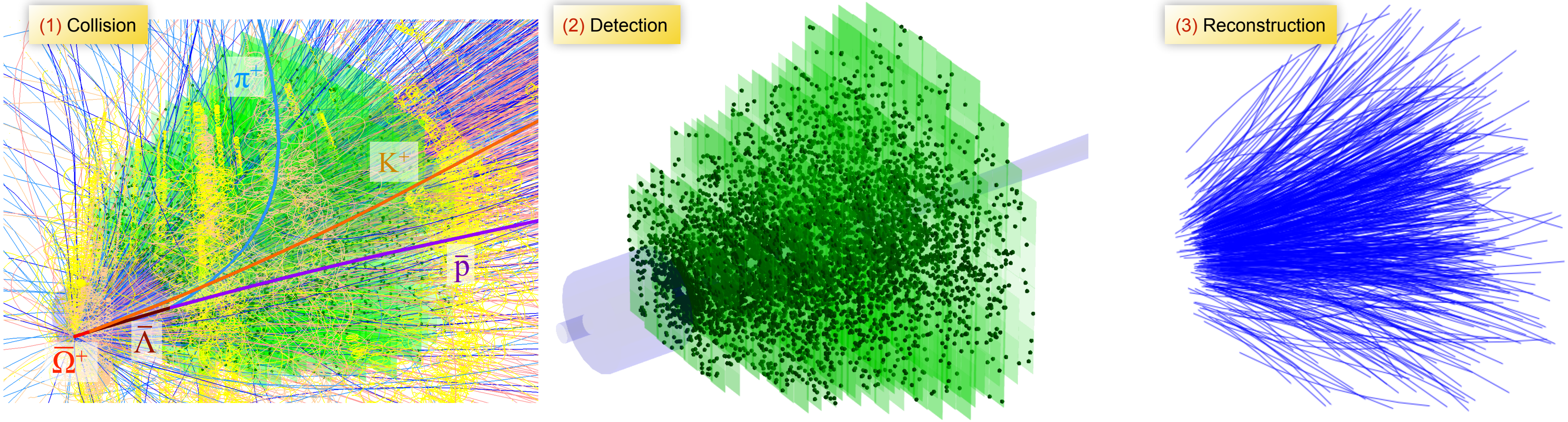
Real-time event reconstruction and analysis in the CBM experiment at FAIR using HPC

I. Kisel

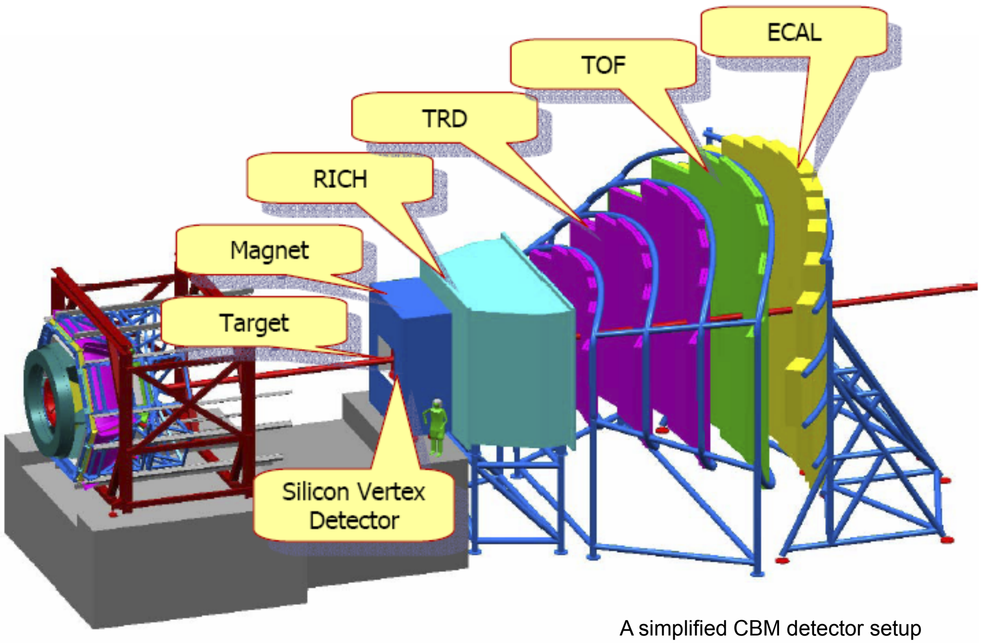
(for the CBM Collaboration)

Goethe University Frankfurt am Main
FIAS Frankfurt Institute for Advanced Studies
GSI Helmholtz Center for Heavy Ion Research

Reconstruction Challenge in CBM

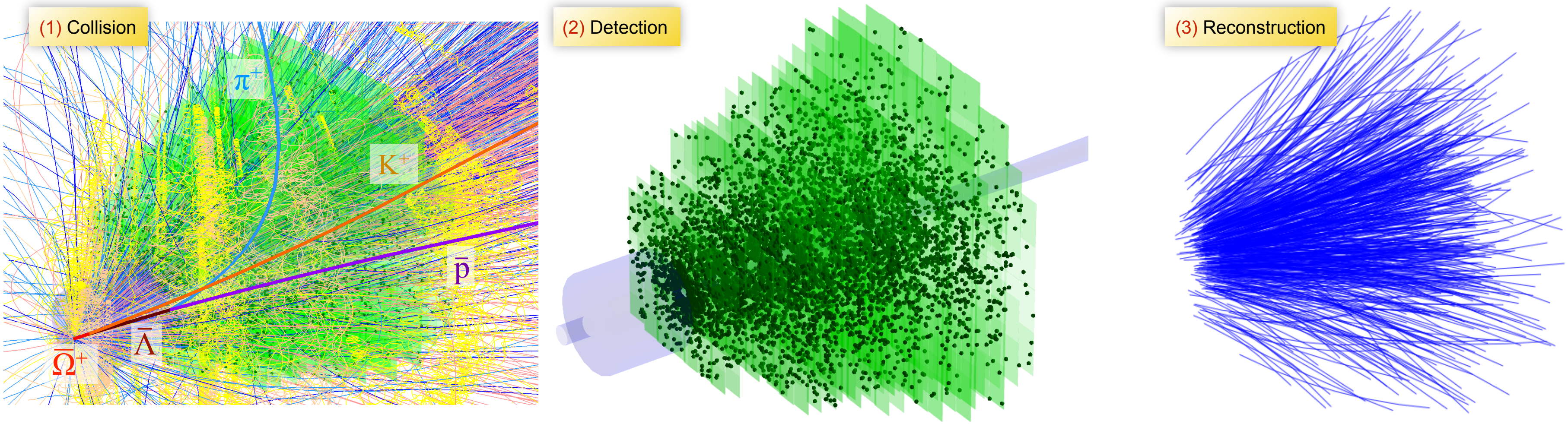


- Future fixed-target heavy-ion experiment at FAIR
- Explore the phase diagram at high net-baryon densities
- 10^7 Au+Au collisions/sec
- ~ 1000 charged particles/collision
- Non-homogeneous magnetic field
- Double-sided strip detectors
- 4D reconstruction of time slices.



A simplified CBM detector setup

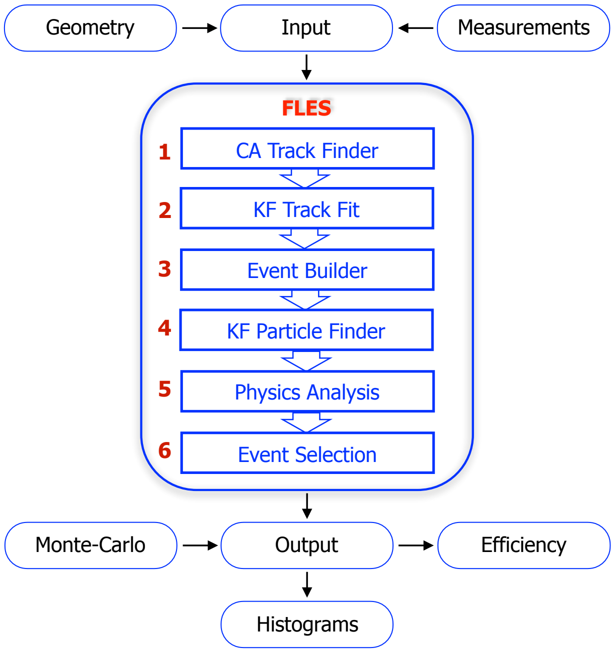
Reconstruction Challenge in CBM



The full event reconstruction will be done
on-line at the First-Level Event Selection (FLES) and
off-line using the same FLES reconstruction package.

- Cellular Automaton (CA) Track Finder
- Kalman Filter (KF) Track Fitter
- KF short-lived Particle Finder

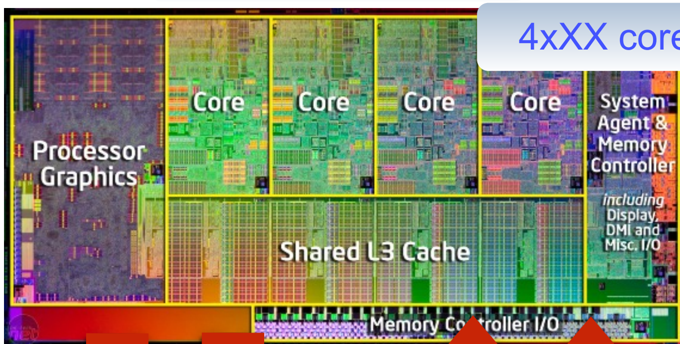
All reconstruction algorithms are vectorized and parallelized.



Many-Core CPU/GPU Architectures

Intel/AMD CPU

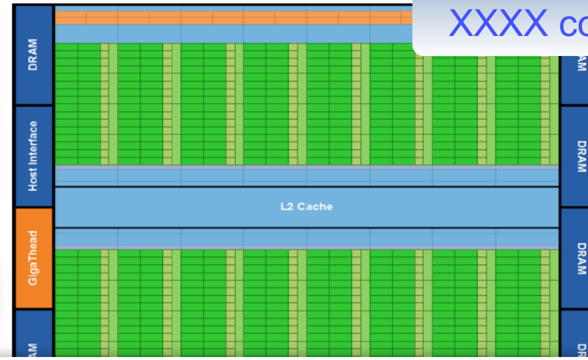
4xXX cores



- Optimiz or lo tency access ache ata sets
- Control c for -of-order and cula execution

Nvidia/ATI GPU

XXXX cores



- ptimized for data-parallel, throughput computation
- lore transistors dedicated to computation

Math

Memory

Parallelism

Math

Memory

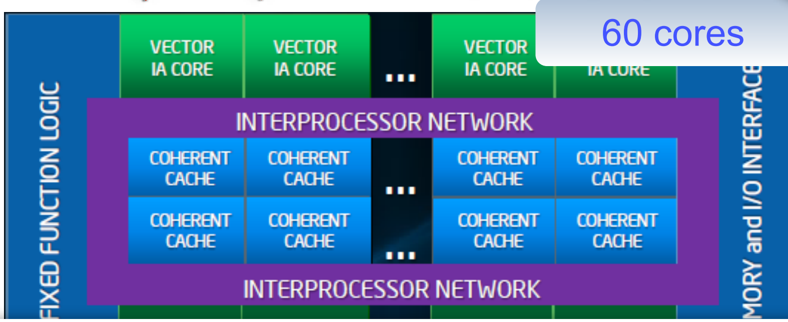
#Cores

Stability

Memory

Intel Phi

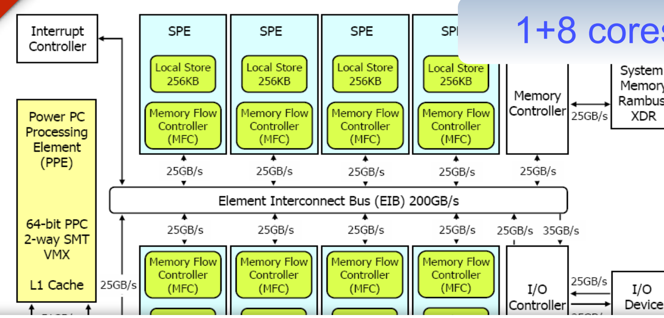
60 cores



- Many Integrated Cores architecture announced at ISC10 (June 2010)
- Based on the x86 architecture
- Many-cores + 4-way multithreaded + 512-bit wide vector unit

IBM Cell

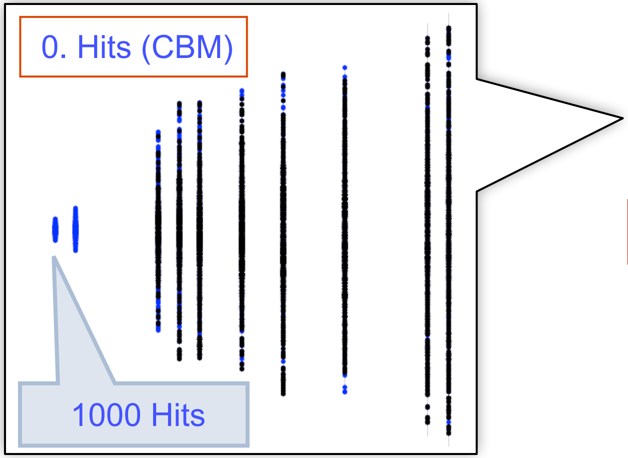
1+8 cores

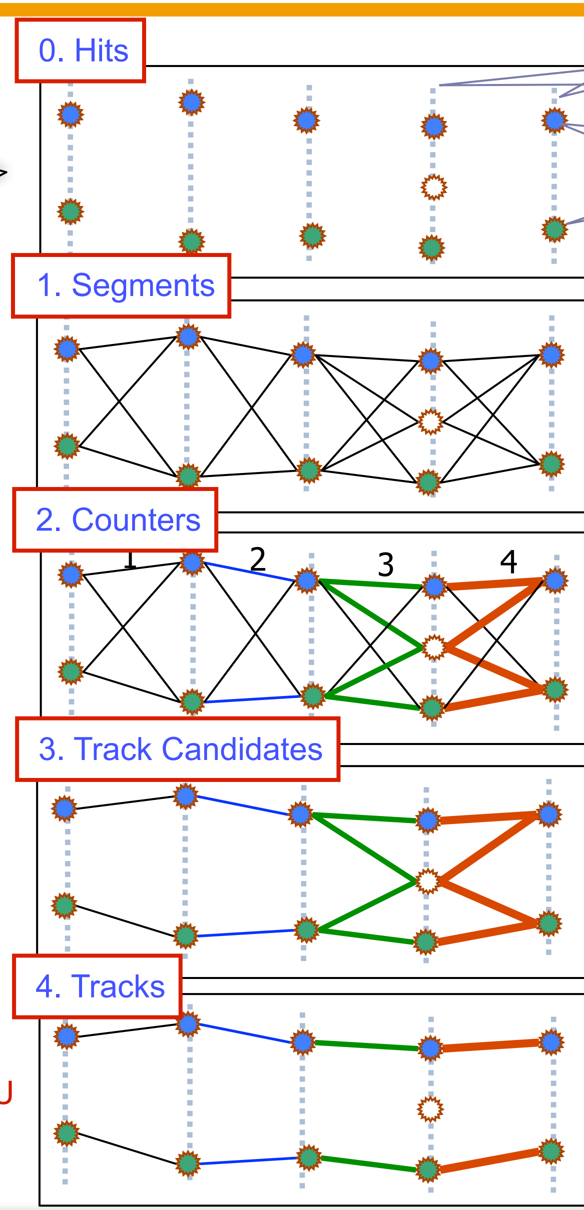


- General purpose RISC processor (PowerPC)
- 8 co-processors (SPE, Synergistic Processor Elements)
- 128-bit wide SIMD units

Future systems are heterogeneous. Fundamental redesign of traditional approaches to data processing is necessary

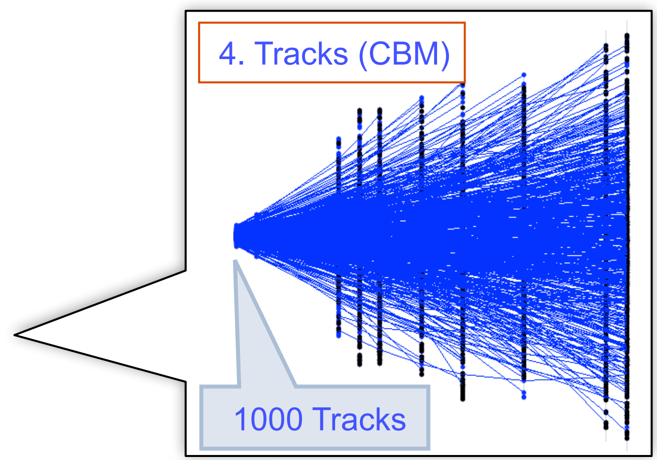
Cellular Automaton (CA) Track Finder





Cellular Automaton:

1. Build short track segments.
2. Connect according to the track model, estimate a possible position on a track.
3. Tree structures appear, collect segments into track candidates.
4. Select the best track candidates.



Cellular Automaton:

- local w.r.t. data
- intrinsically parallel
- extremely simple
- very fast

Deeply appropriate for many-core CPU/GPU

Useful for complicated event topologies with heavy combinatorics

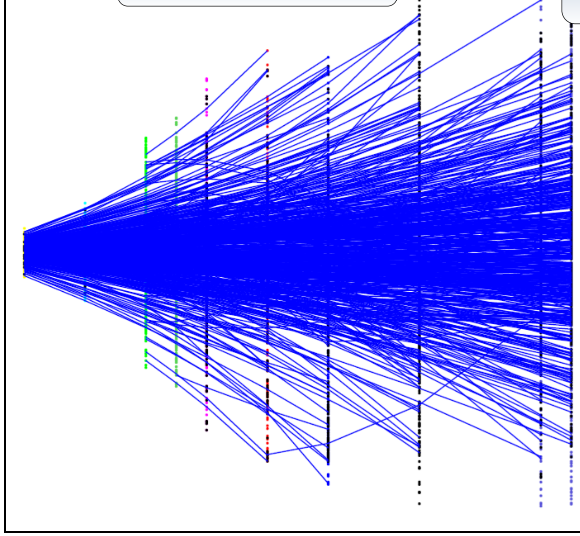
Cellular Automaton (CA) Track Finder

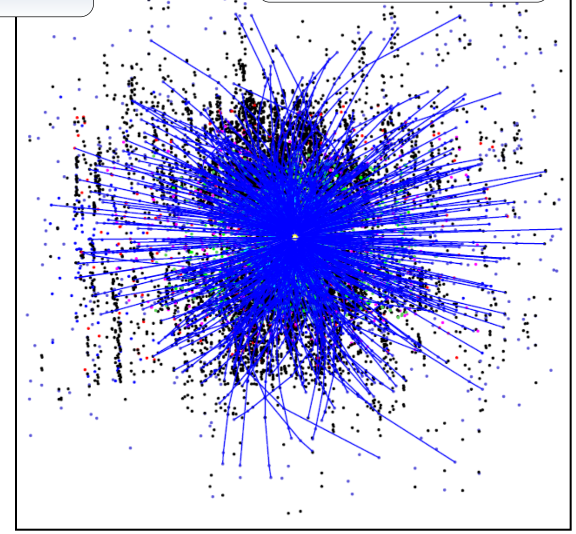
1

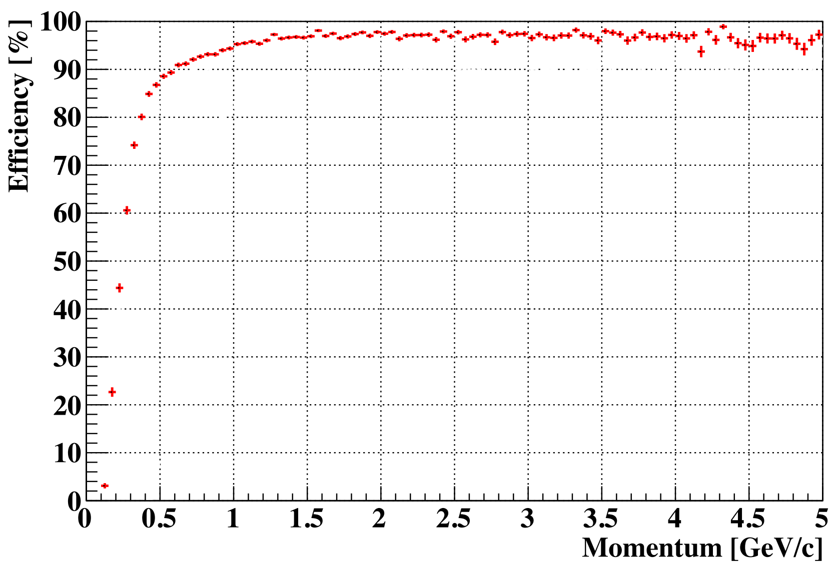
Top view

770 Tracks

Front view





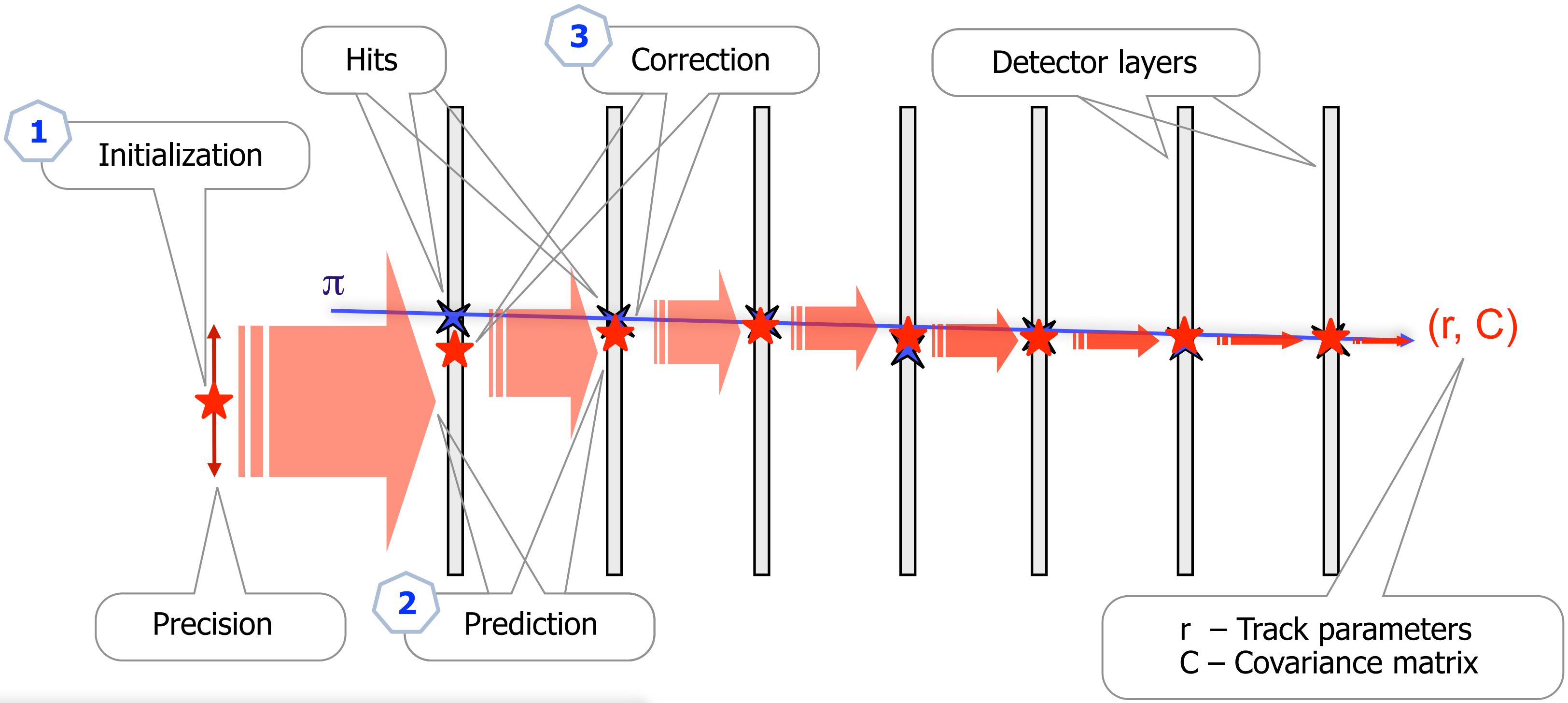


Track category	Eff, %
All tracks	90.9
Primary high-p	97.5
Primary low-p	92.6
Secondary high-p	91.1
Secondary low-p	63.8
Clone level	0.4
Ghost level	5.9
MC tracks found	134
Time, ms/ev	10

Fast and efficient track finder

Kalman Filter (KF) based Track Fit

Estimation of the track parameters at one or more hits along the track – Kalman Filter (KF)

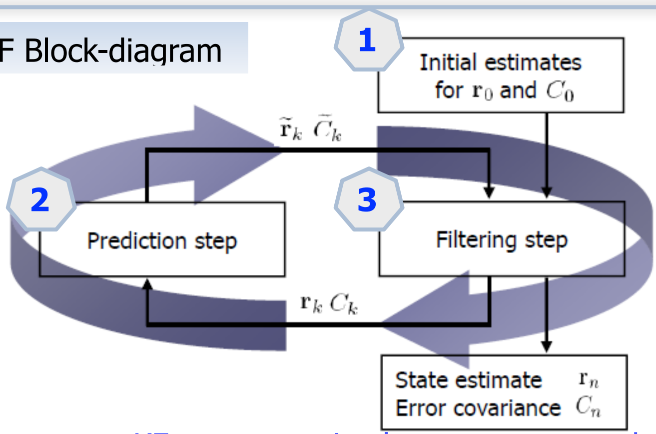


State vector

Position, direction and momentum

$r = \{ x, y, z, p_x, p_y, p_z \}$

KF Block-diagram



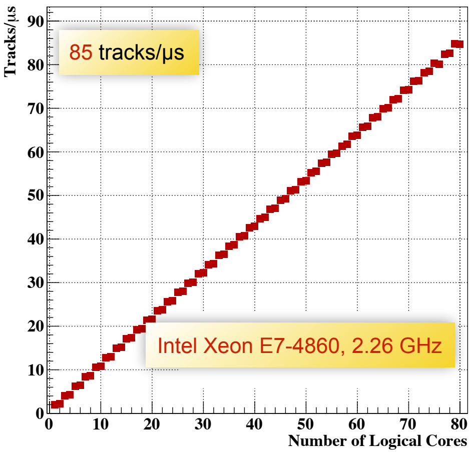
KF as a recursive least squares method

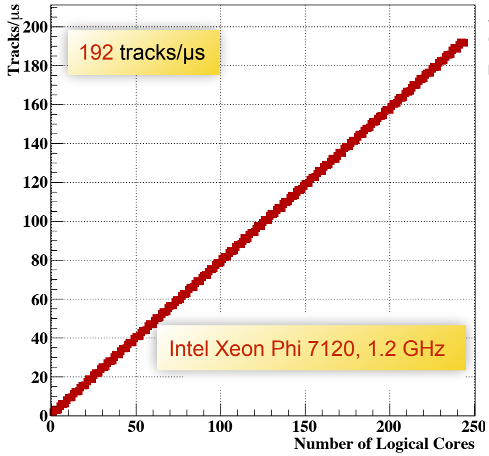
Kalman Filter:

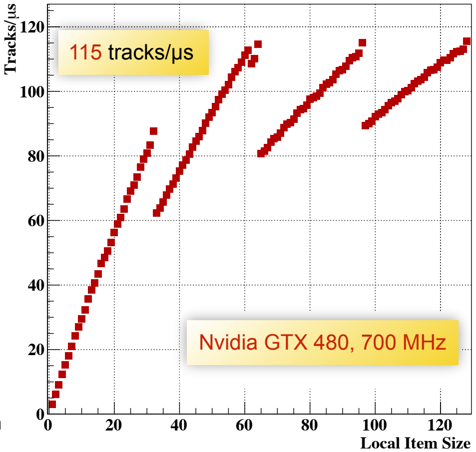
1. Start with an arbitrary initialization.
2. Add one hit after another.
3. Improve the state vector.
4. Get the optimal parameters after the last hit.

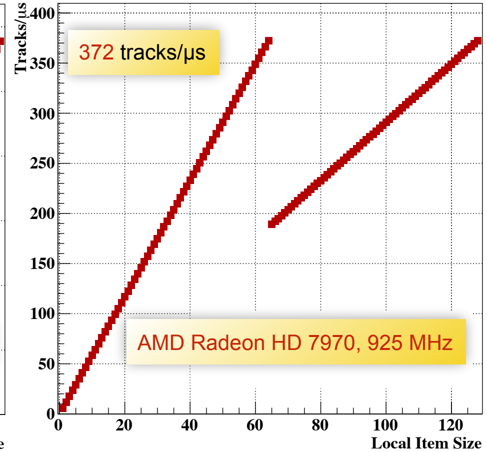
Nowadays the Kalman Filter is used in almost all HEP experiments

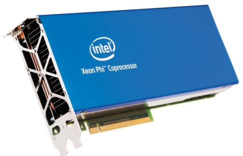
Kalman Filter (KF) Track Fit









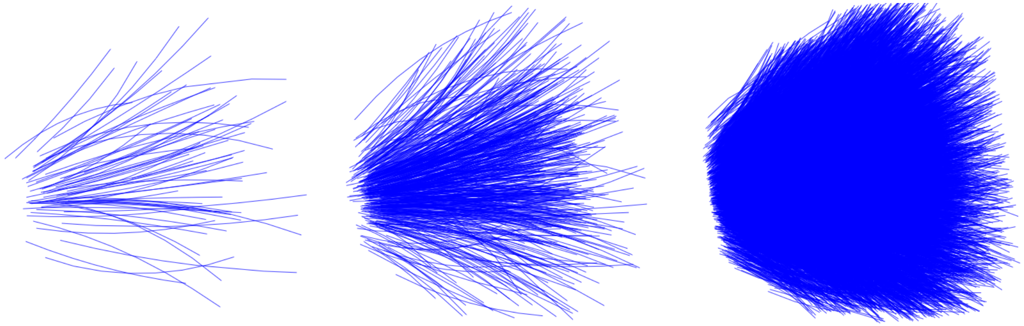


- Precise estimation of the parameters of particle trajectories is the core of the reconstruction procedure.
- Scalability with respect to the number of logical cores in a CPU is one of the most important parameters of the algorithm.
- The scalability on the Intel Xeon Phi coprocessor is similar to the CPU, but running four threads per core instead of two.
- In case of the graphics cards the set of tasks is divided into working groups of size *local item size* and distributed among compute units (or streaming multiprocessors) and the load of each compute unit is of the particular importance.
- The track fit performance on a single node: 2*CPU+2*GPU = 10^9 tracks/s = (100 tracks/event)* 10^7 events/s = 10^7 events/s.
- A single compute node is enough to estimate parameters of all particles produced at the maximum 10^7 interaction rate!

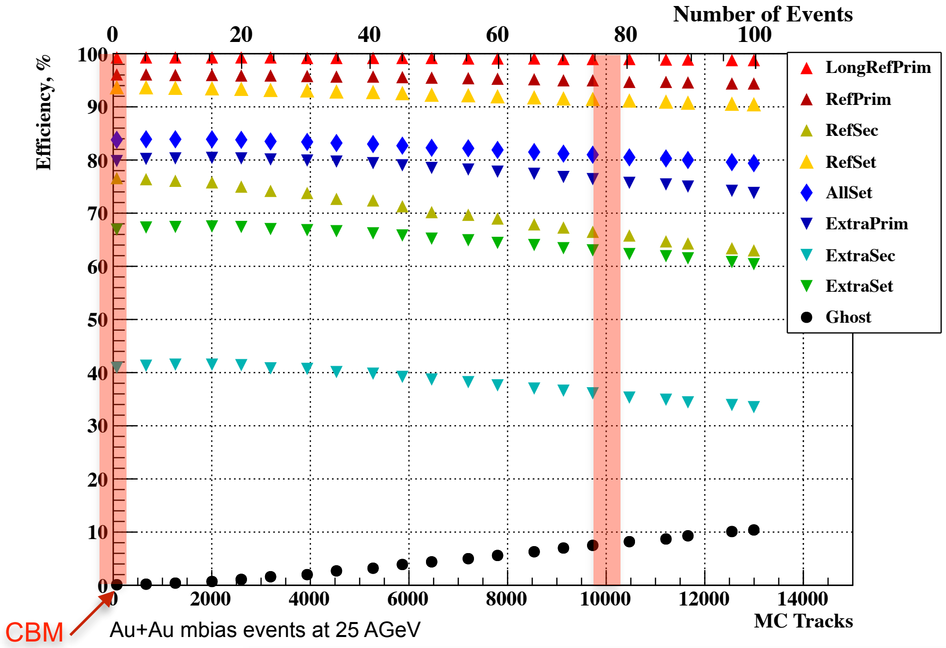
The fastest implementation of the Kalman filter in the world

CA Track Finder at High Track Multiplicity

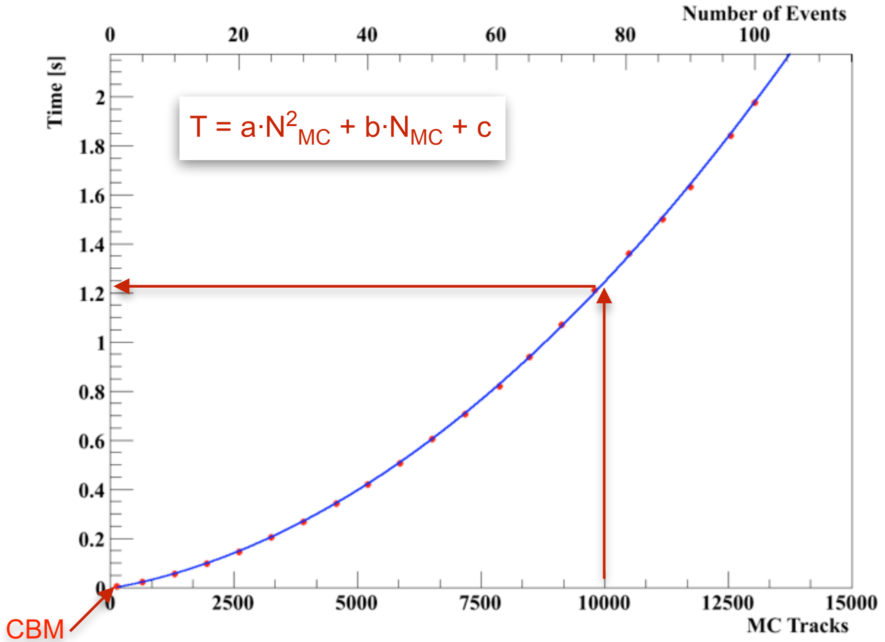
A number of minimum bias events is gathered into a group (super-event), which is then treated by the CA track finder as a single event.



1 mbias event, $\langle N_{reco} \rangle = 109$ 5 mbias events, $\langle N_{reco} \rangle = 572$ 100 mbias events, $\langle N_{reco} \rangle = 10340$

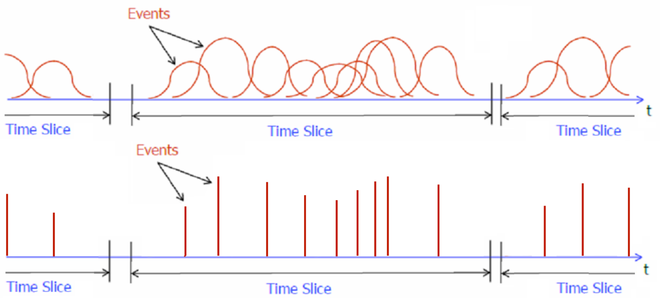


Au+Au mbias events at 25 AGeV



Reliable reconstruction efficiency and time as a second order polynomial w.r.t. to the track multiplicity

Time-based (4D) Track Reconstruction

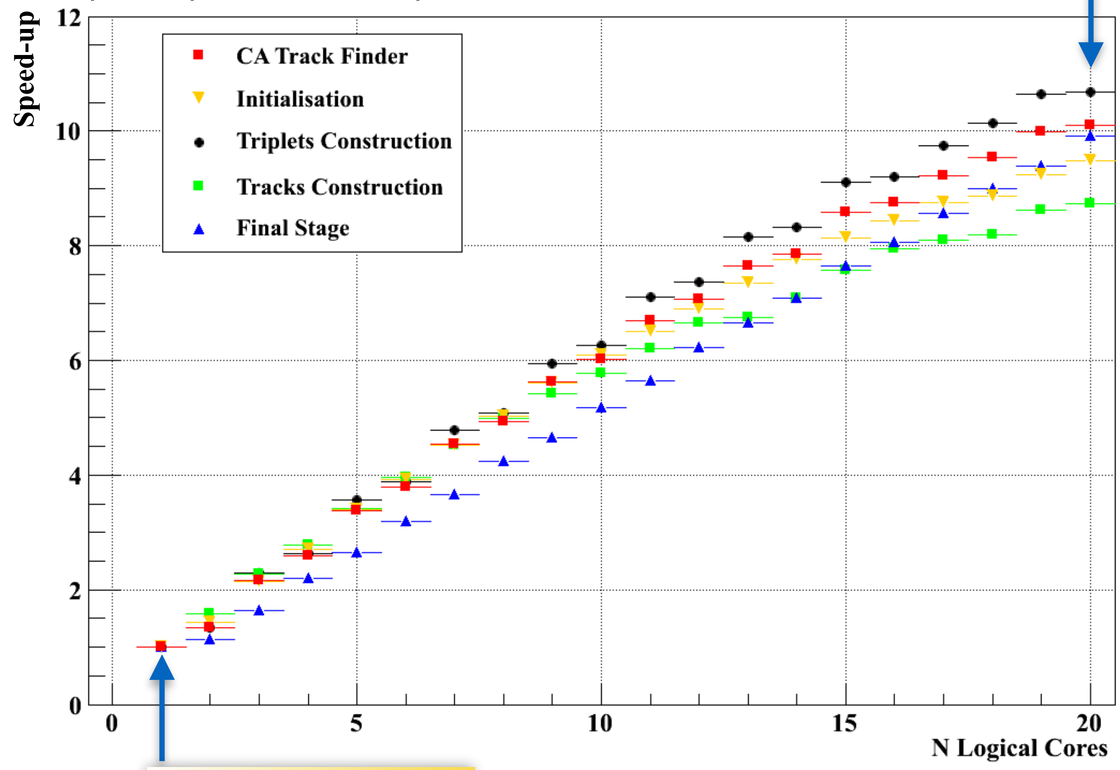


- The beam in the CBM will have no bunch structure, but continuous.
- Measurements in this case will be 4D (x, y, z, t).
- Significant overlapping of events in the detector system.
- Reconstruction of time slices rather than events is needed.

Efficiency, %	3D	4D
All tracks	83.8	83.0
Primary high-p	96.1	92.8
Primary low-p	79.8	83.1
Secondary high-p	76.6	73.2
Secondary low-p	40.9	36.8
Clone level	0.4	1.7
Ghost level	0.1	0.3
Time/event/core, ms	8.2	8.5

Total CA time = 84 ms

Speed-up factor due to parallelization within the time-slice



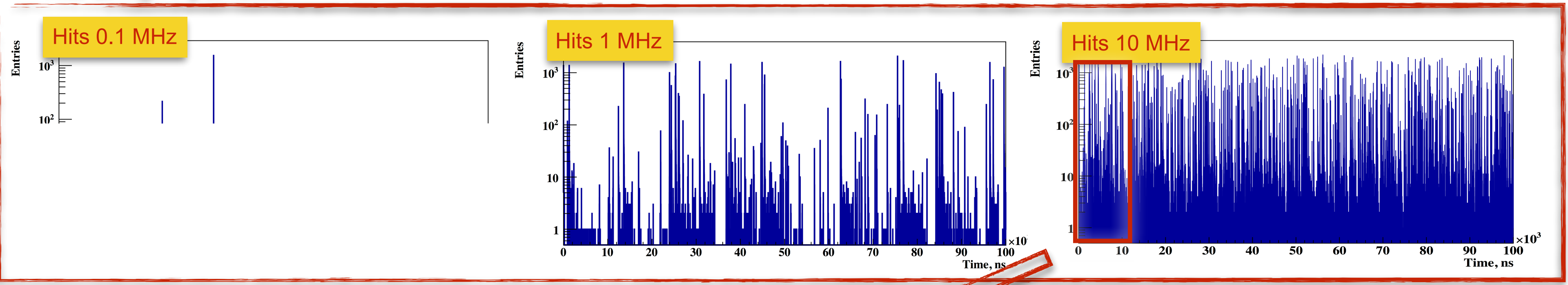
Total CA time = 849 ms

100 mbias events in a time-slice

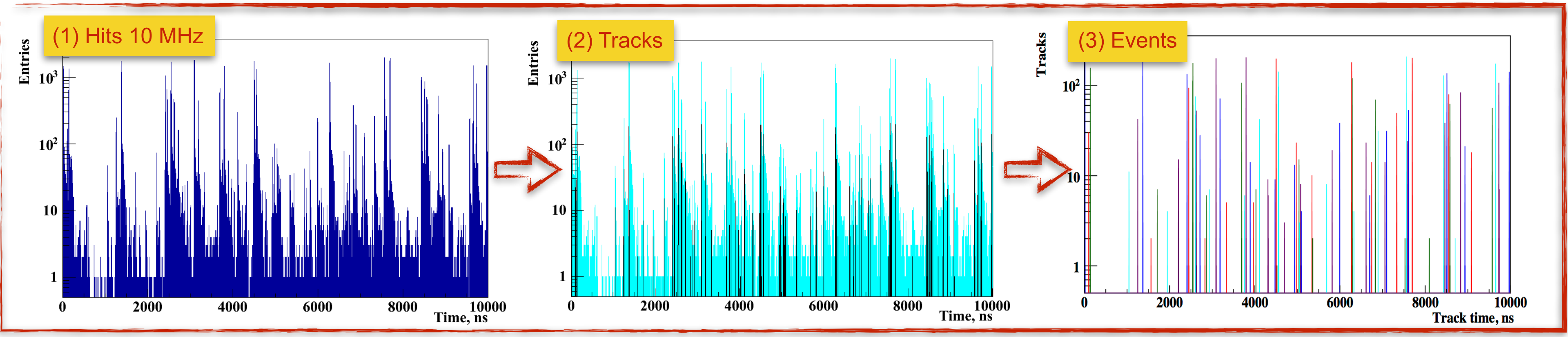
The reconstruction time 8.2 ms/event in 3D is recovered in 4D case as well

4D Event Building at 10 MHz

Hits at high input rates

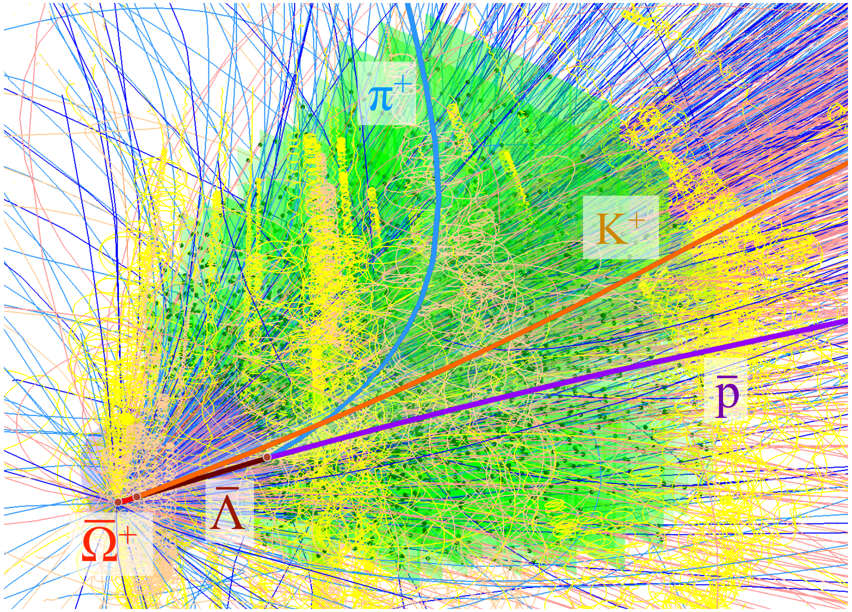


From hits to tracks to events



Reconstructed tracks clearly represent groups, which correspond to the original events

KF Particle: Reconstruction of short-lived Particles



Simulated AuAu collision at 25 AGeV

$$\bar{\Omega}^+ \leftarrow \bar{\Lambda}\, K^+$$
$$\hookrightarrow \bar{p}\, \pi^+$$

```
KFParticle Lambda(P, Pi);                 // construct anti Lambda
Lambda.SetMassConstraint(1.1157);         // improve momentum and mass
KFParticle Omega(K, Lambda);              // construct anti Omega
PV -= (P; Pi; K);                         // clean the primary vertex
PV += Omega;                              // add Omega to the primary vertex
Omega.SetProductionVertex(PV);            // Omega is fully fitted
(K; Lambda).SetProductionVertex(Omega);   // K, Lambda are fully fitted
(P; Pi).SetProductionVertex(Lambda);      // p, pi are fully fitted
```

$$\mathbf{r} = \{\, x, y, z, p_x, p_y, p_z, E \,\}$$

State vector

$$C = \langle \mathbf{r}\mathbf{r}^T \rangle = \begin{bmatrix} \sigma_x^2 & C_{xy} & C_{xz} & C_{xp_x} & C_{xp_y} & C_{xp_z} & C_{xE} \\ C_{xy} & \sigma_y^2 & C_{yz} & C_{yp_x} & C_{yp_y} & C_{yp_z} & C_{yE} \\ C_{xz} & C_{yz} & \sigma_z^2 & C_{zp_x} & C_{zp_y} & C_{zp_z} & C_{zE} \\ C_{xp_x} & C_{yp_x} & C_{zp_x} & \sigma_{p_x}^2 & C_{p_xp_y} & C_{p_xp_z} & C_{p_xE} \\ C_{xp_y} & C_{yp_y} & C_{zp_y} & C_{p_xp_y} & \sigma_{p_y}^2 & C_{p_yp_z} & C_{p_yE} \\ C_{xp_z} & C_{yp_z} & C_{zp_z} & C_{p_xp_z} & C_{p_yp_z} & \sigma_{p_z}^2 & C_{p_yE} \\ C_{xE} & C_{yE} & C_{zE} & C_{p_xE} & C_{p_yE} & C_{p_zE} & \sigma_E^2 \end{bmatrix}$$

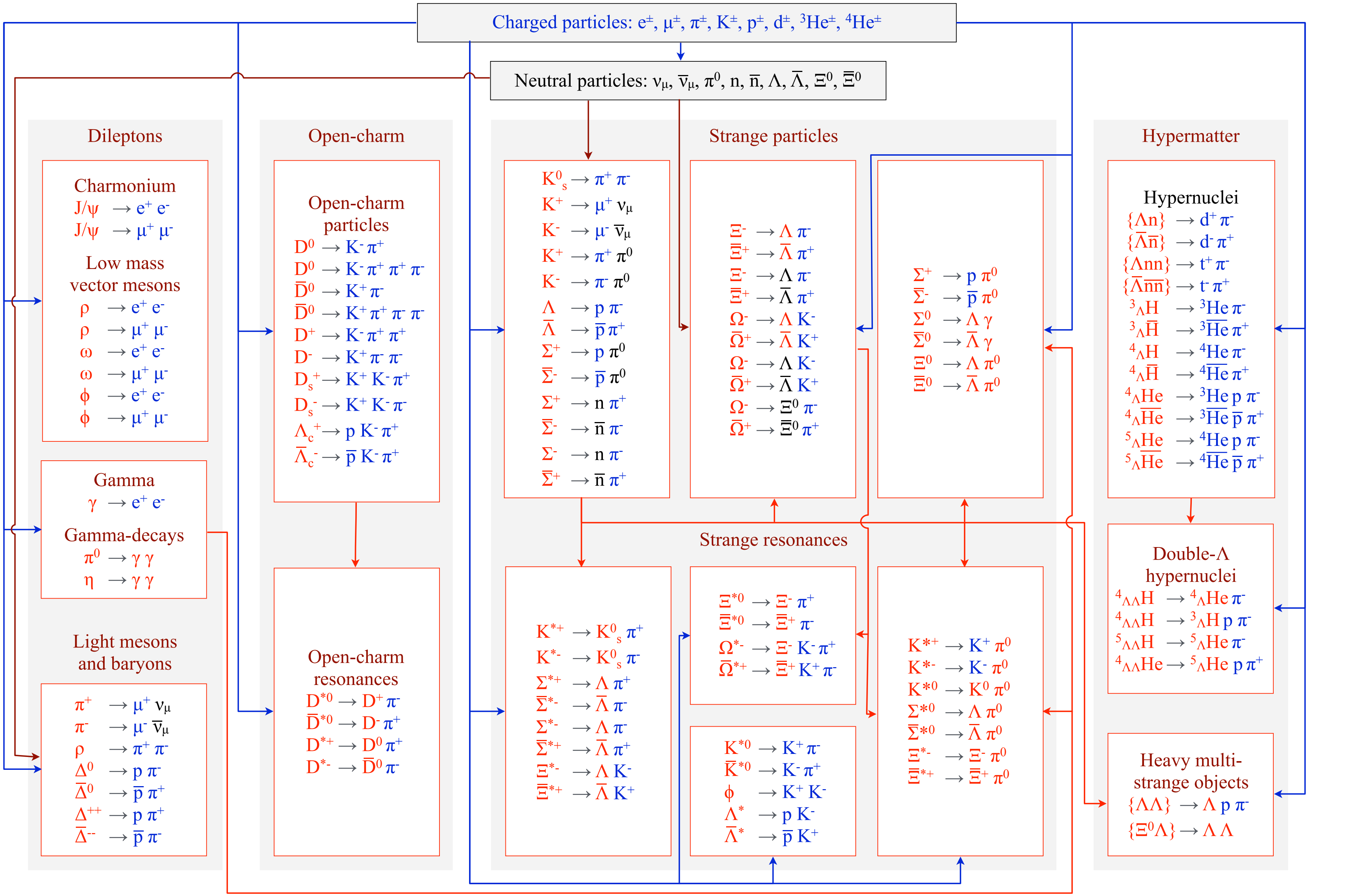
Covariance matrix

Features:

- KF Particle class describes particles by the state vector and the covariance matrix.
- Covariance matrix contains essential information about tracking and detector performance.
- The method for mathematically correct usage of covariance matrices is provided by the KF Particle package based on the Kalman filter (KF).
- Heavy mathematics of KF requires fast and vectorised algorithms.
- Mother and daughter particles are treated in the same way.
- The natural and simple interface allows two reconstruct easily complicated decay chains.
- The package is geometrically independent and can be adapted to different experiments (CBM, ALICE, STAR).

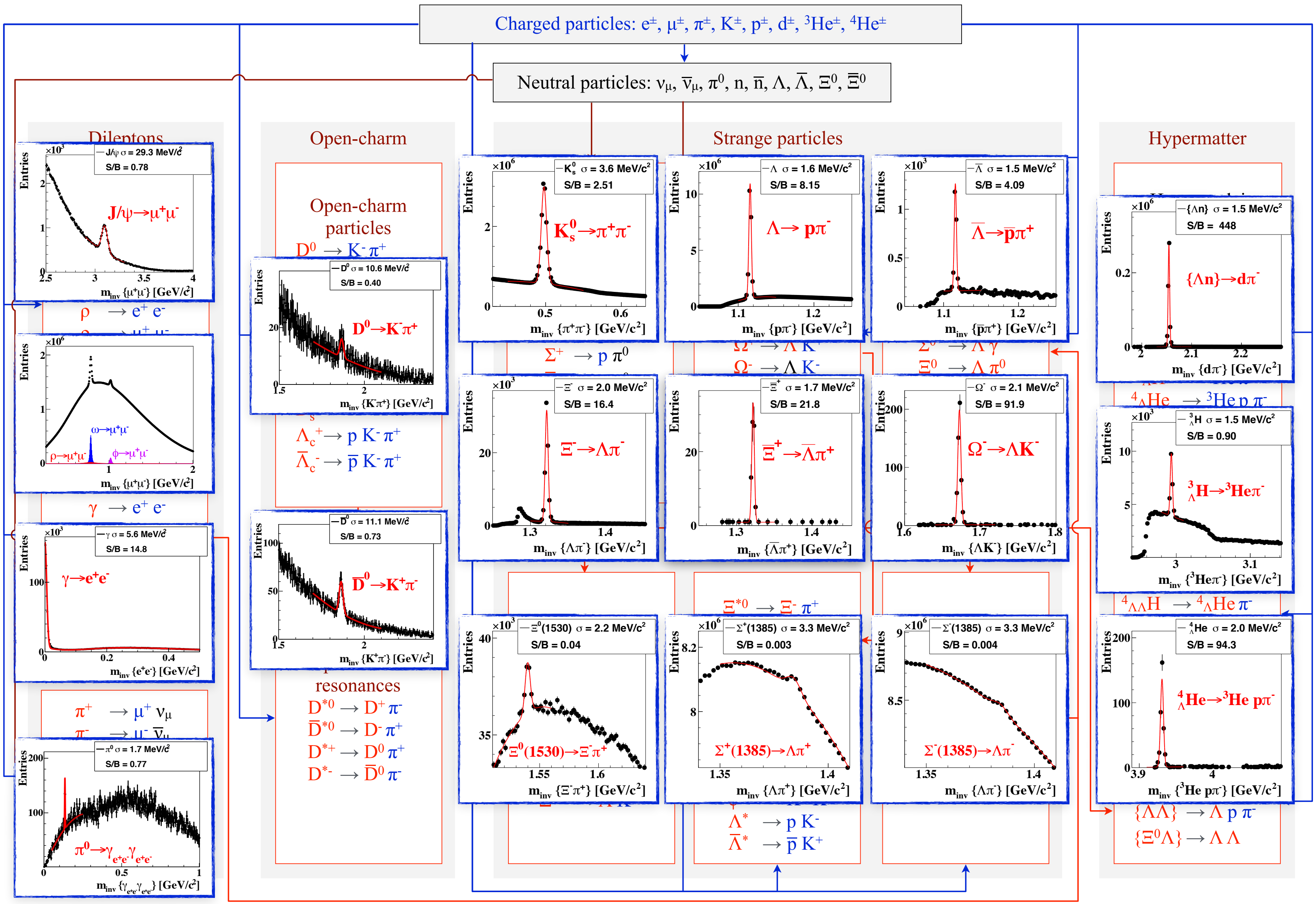
KF Particle provides a simple and very efficient approach to physics analysis

KF Particle Finder for Physics Analysis and Selection

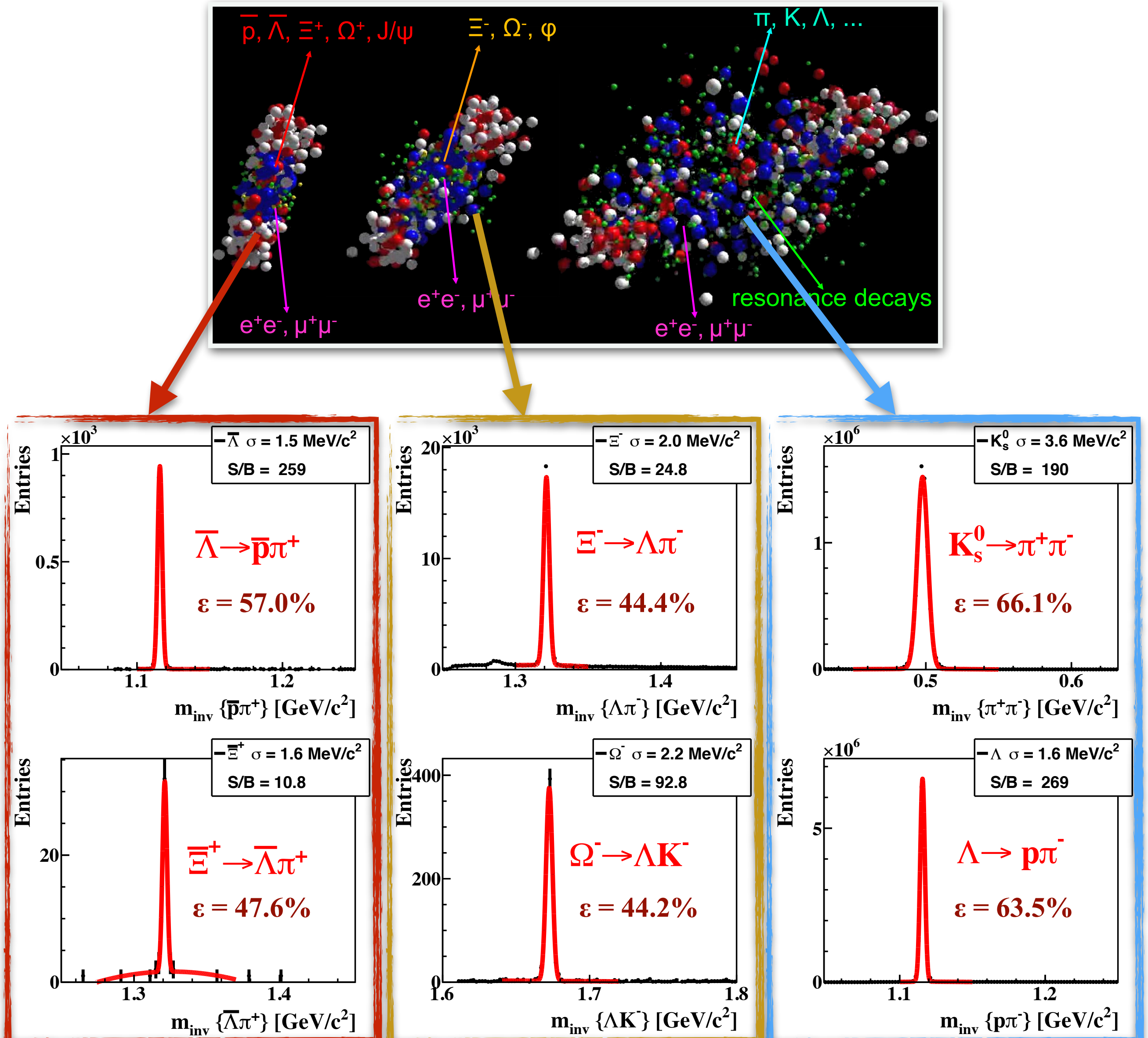


(mbias: 1 ms; central: 10 ms)/event/core

KF Particle Finder for Physics Analysis and Selection

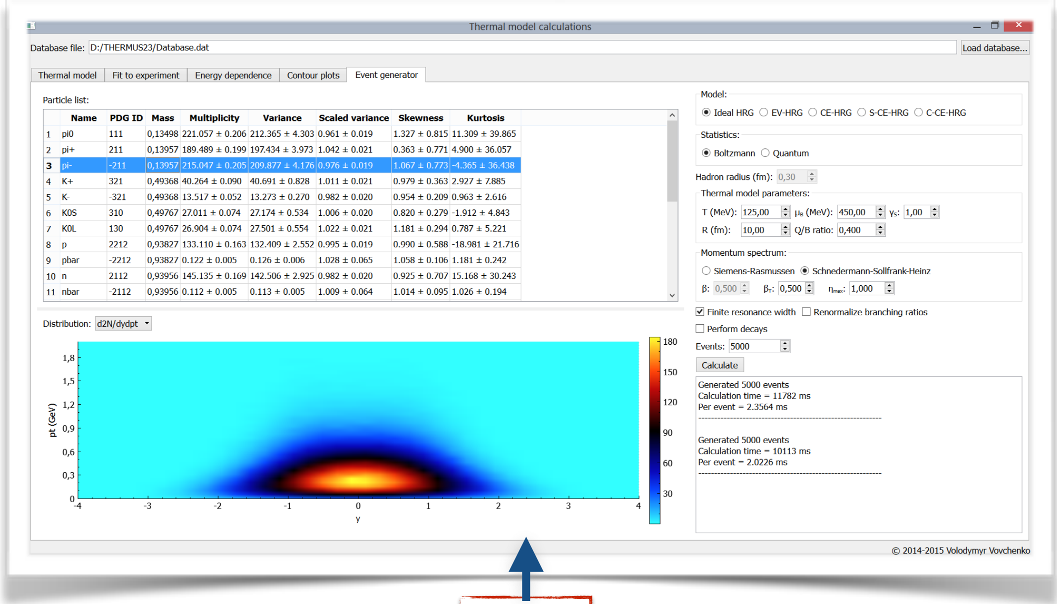


Very Clean Probes of Collision Stages



AuAu, 10 AGeV, 3.5M central UrQMD events, **MC PID**

Inverse Approach in Real-Time Physics Analysis

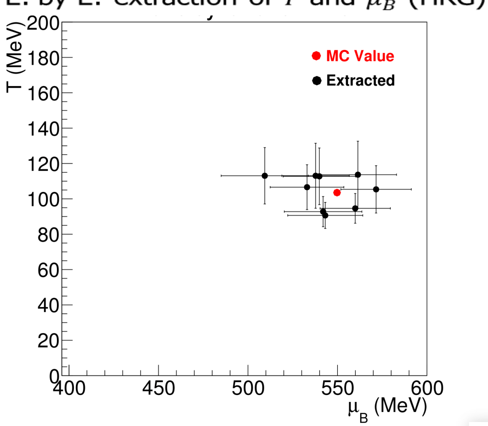


Direct approach

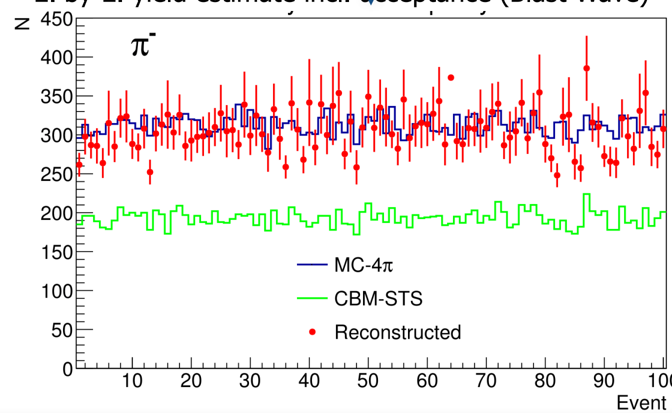
Theoretical model (T, μ,V, σ,…) ⇄ $\omega_p \frac{dN}{dp^3}$ (observable)

Inverse approach

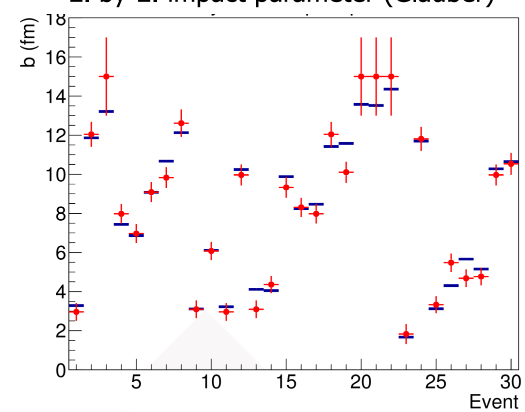
E.-by-E. extraction of T and μ_B (HRG)



E.-by-E. yield estimate incl. acceptance (Blast-Wave)



E.-by-E. impact parameter (Glauber)



A package to estimate the medium parameters is implemented

Efficient Parallelization in Event Reconstruction

CPU - Full reconstruction

CPU - Tracking

All - Benchmark

GPU/Phi - Selection

Algorithm	SIMD	ITBB, OpenMP	CUDA	OpenCL CPU/GPU	Phi	ArBB
Hit Producers						
STS KF Track Fit	✓	✓	✓	✓/✓	✓	✓
STS CA Track Finder	✓	✓				
MuCh Track Finder	✓	✓	✓			
TRD Track Finder	✓	✓	✓			
RICH Ring Finder	✓	✓		✓/✓		
KF Particle Finder	✓	✓		✓/✓	✓	
Off-line Physics Analysis	✓					
FLES Analysis and Selection	✓	✓				

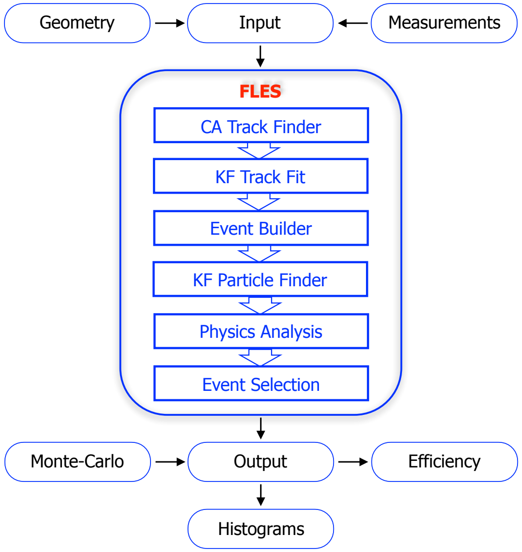
Andrzej Nowak (OpenLab, CERN) by Hans von der Schmitt (ATLAS) at GPU Workshop, DESY, 15-16 April 2013

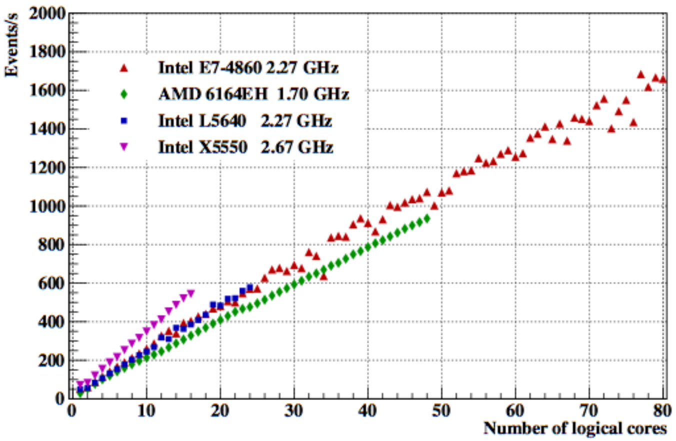
	SIMD	Instr. Level Parallelism	HW Threads	Cores	Sockets	Factor	Efficiency
MAX	4	4	1.35	8	4	691.2	100.0%
Typical	2.5	1.43	1.25	8	2	71.5	10.3%
HEP	1	0.80	1	6	2	9.6	1.4%
CBM@FAIR	4	3	1.3	8	4	499.2	72.2%

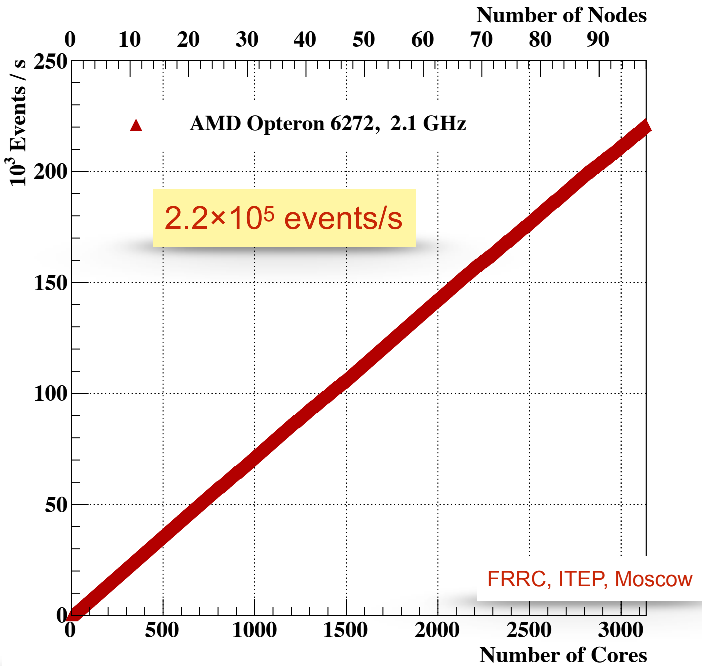
x Algorithm
x Memory

Parallelization becomes a standard in the CBM experiment

Running FLES on HPC Node/Farm





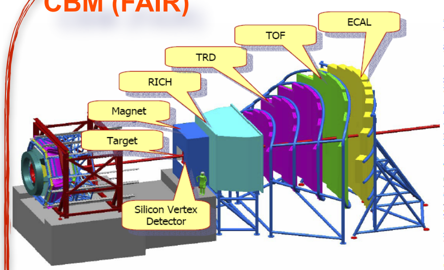


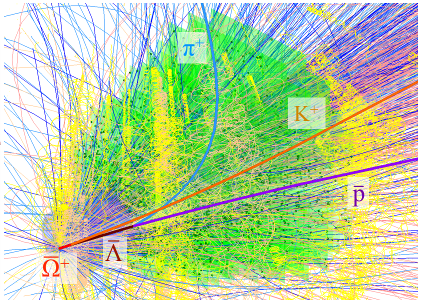
The FLES package is vectorized, parallelized, portable and scalable up to 3 200 CPU cores

CBM -> STAR: Reconstruction and Analysis Software

Within the FAIR Phase-0 program the CBM KF Particle Finder has been adapted to STAR and applied to real data of 2014, 2016 and BES-I.

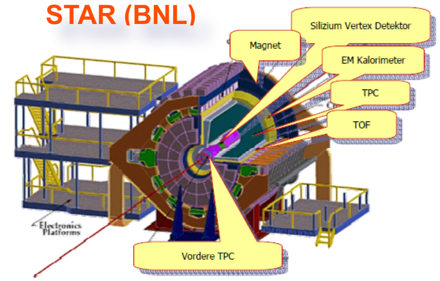
CBM (FAIR)

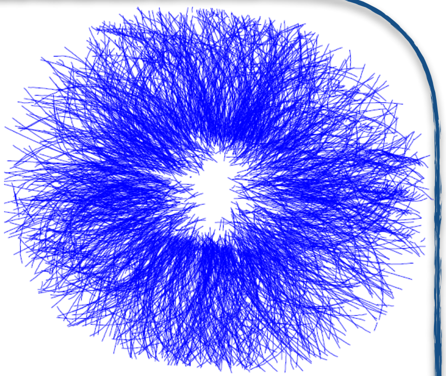




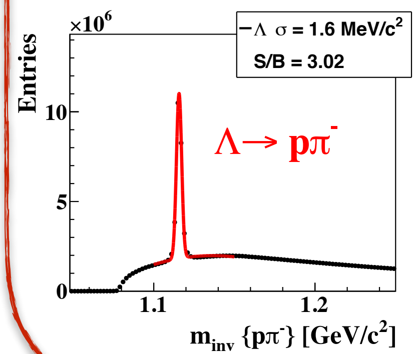
$\bar{\Omega}^+ \rightarrow \bar{\Lambda} K^+$
$\hookrightarrow \bar{p}\, \pi^+$

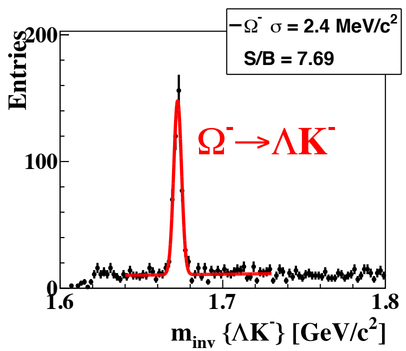
STAR (BNL)



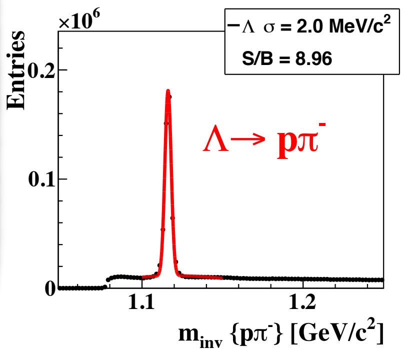


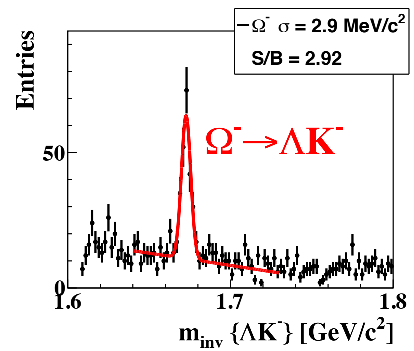
CBM, 5M central Au+Au, 10 AGeV, PHSD





STAR, 1.3M mbias Au+Au, 200 AGeV, Run 2016

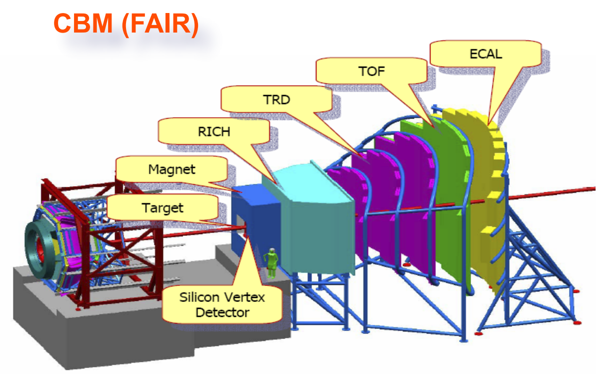


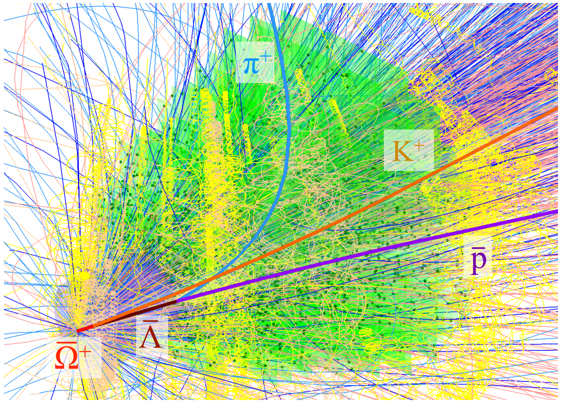


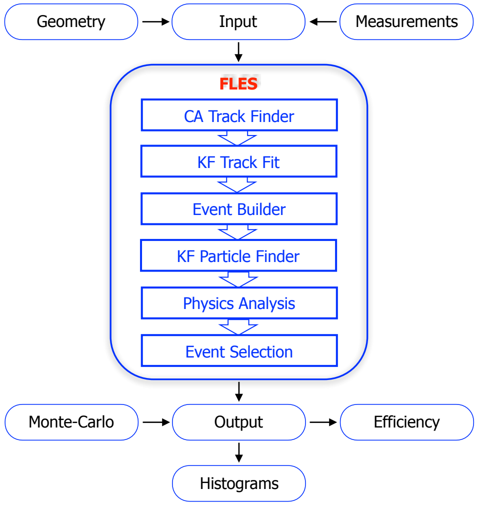
- ✓ Since 2013 (online) and 2016 (offline) the CA track finder is the standard STAR track finder for data production. Use of CA provides 25% more D^0 and 20% more W.
- ✓ The KF particle finder provides a factor 2 more signal particles than the standard approach in STAR. The integration of the KF particle finder into the official STAR repository for use in physics analysis is currently in progress.

Preparing for the real-time express physics analysis during the BES-II runs (2019-2020)

Conclusion







1. The next generation of HEP and HI experiments with very high input rates will require the full reconstruction and physics analysis of the experimental data in real time.
2. Errors and insufficient accuracy in online data processing, physics analysis or selection of interesting collisions will lead to complete loss of all data, since only the (incorrectly) selected data will be stored in this case.
3. This requires to redesign all offline algorithms for their fast and reliable online operation, as it is already partially done on some of HPC High-Level Trigger farms, like in ALICE (CERN, Switzerland) and STAR (BNL, US).
4. The Cellular Automaton for searching for particle trajectories and the Kalman Filter to estimate their parameters have a high level of intrinsic parallelism for their efficient implementation on modern and future many-core HPC architectures.