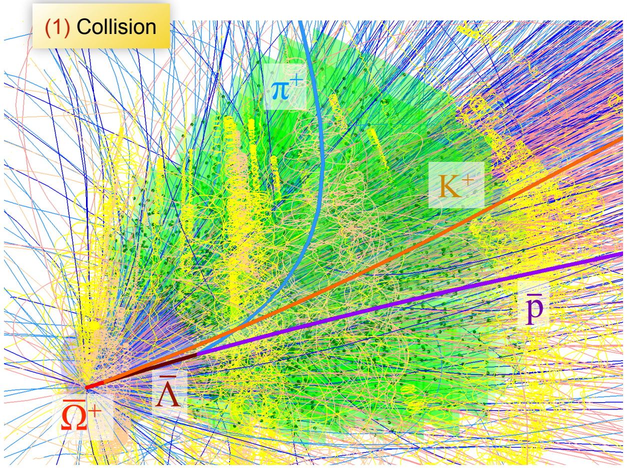
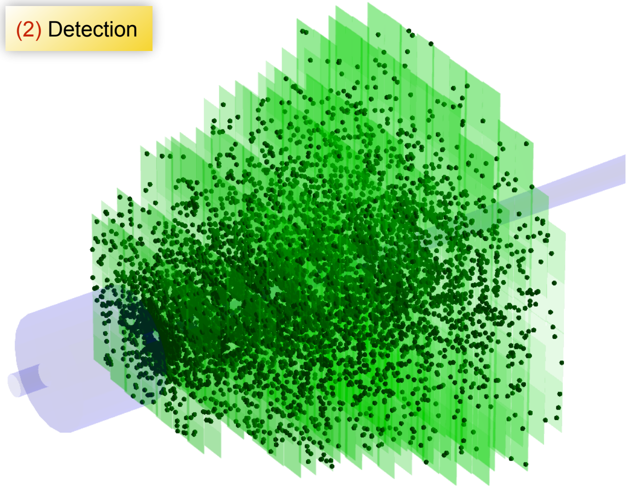
Event Topology Reconstruction in the CBM Experiment at FAIR

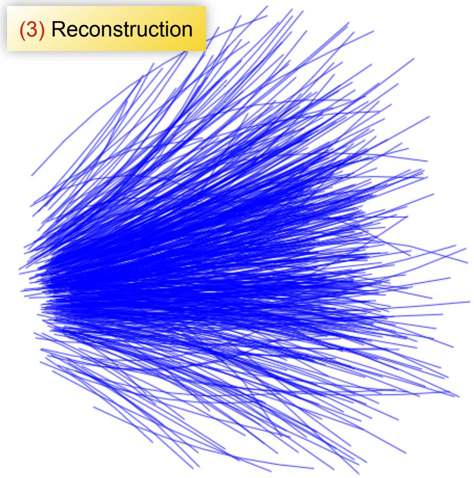
V. Akishina[1], *I. Kisel*[1,2,3], P. Kisel[1,3], P. Senger[3], I. Vassiliev[3], M. Zyzak[3]

[1] Goethe University Frankfurt am Main
[2] FIAS Frankfurt Institute for Advanced Studies
[3] GSI Helmholtz Center for Heavy Ion Research

Reconstruction Challenge in CBM



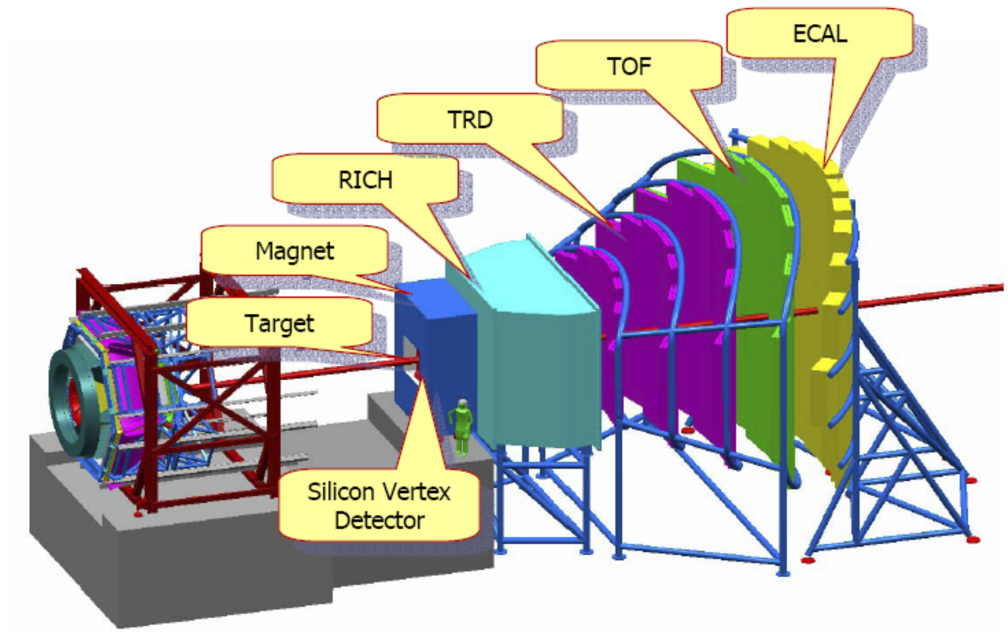




- Future fixed-target heavy-ion experiment
- 10^7 Au+Au collisions/sec
- ~ 1000 charged particles/collision
- Non-homogeneous magnetic field
- Double-sided strip detectors (85% fake space-points)

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

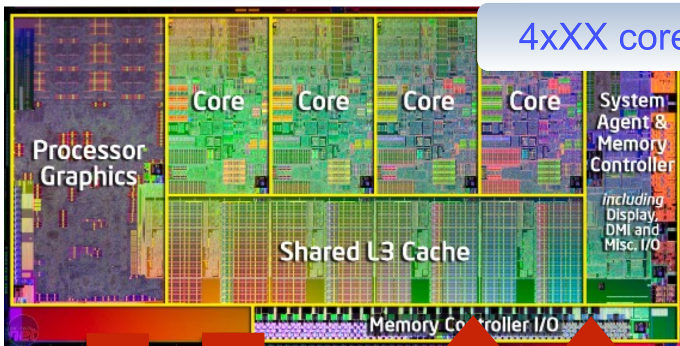


All reconstruction algorithms are vectorized and parallelized.

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

Math

Memory

Parallelism

Math

Memory

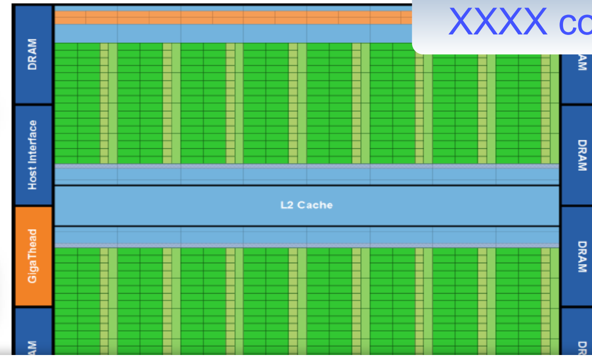
#Cores

Stability

Memory

Nvidia/ATI GPU

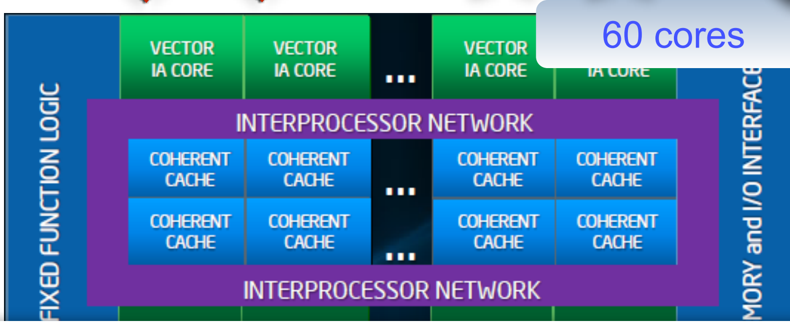
XXXX cores



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

Intel Phi

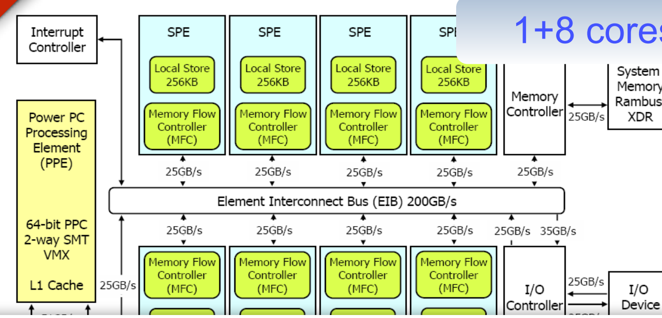
60 cores



- 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

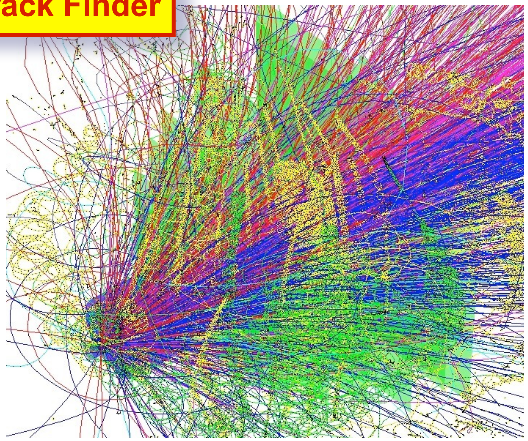


- General purpose RISC processor (PowerPC)
- 8 co-processors (SPE, Synergistic Processor Elements)
- 128-bit wide SIMD units

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

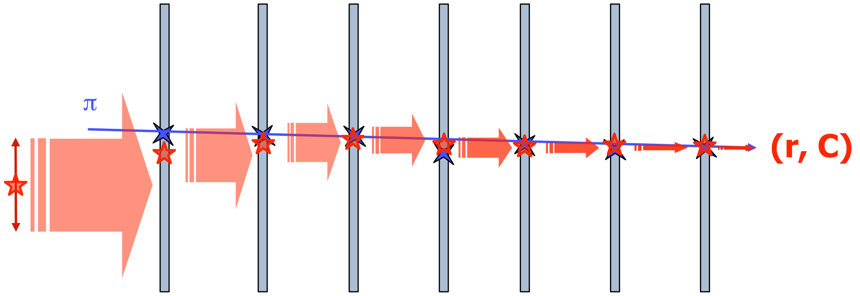
Stages of Event Reconstruction

// Track Finder



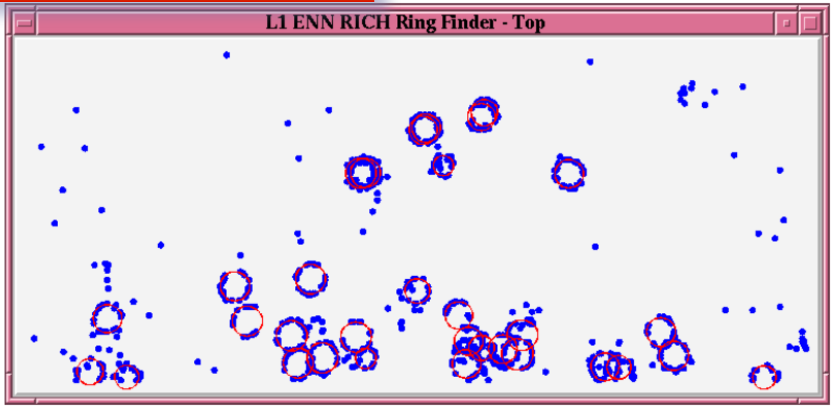
- Conformal Mapping
- Hough Transformation
- Track Following
- **Cellular Automaton**

// Track Fitter



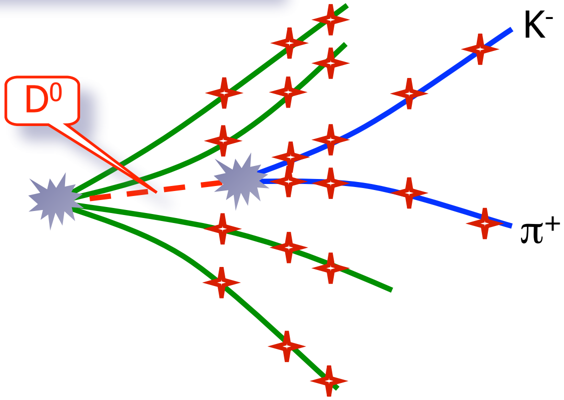
- **Kalman Filter**

// Ring Finder (Particle ID)



- Hough Transformation
- Elastic Neural Net

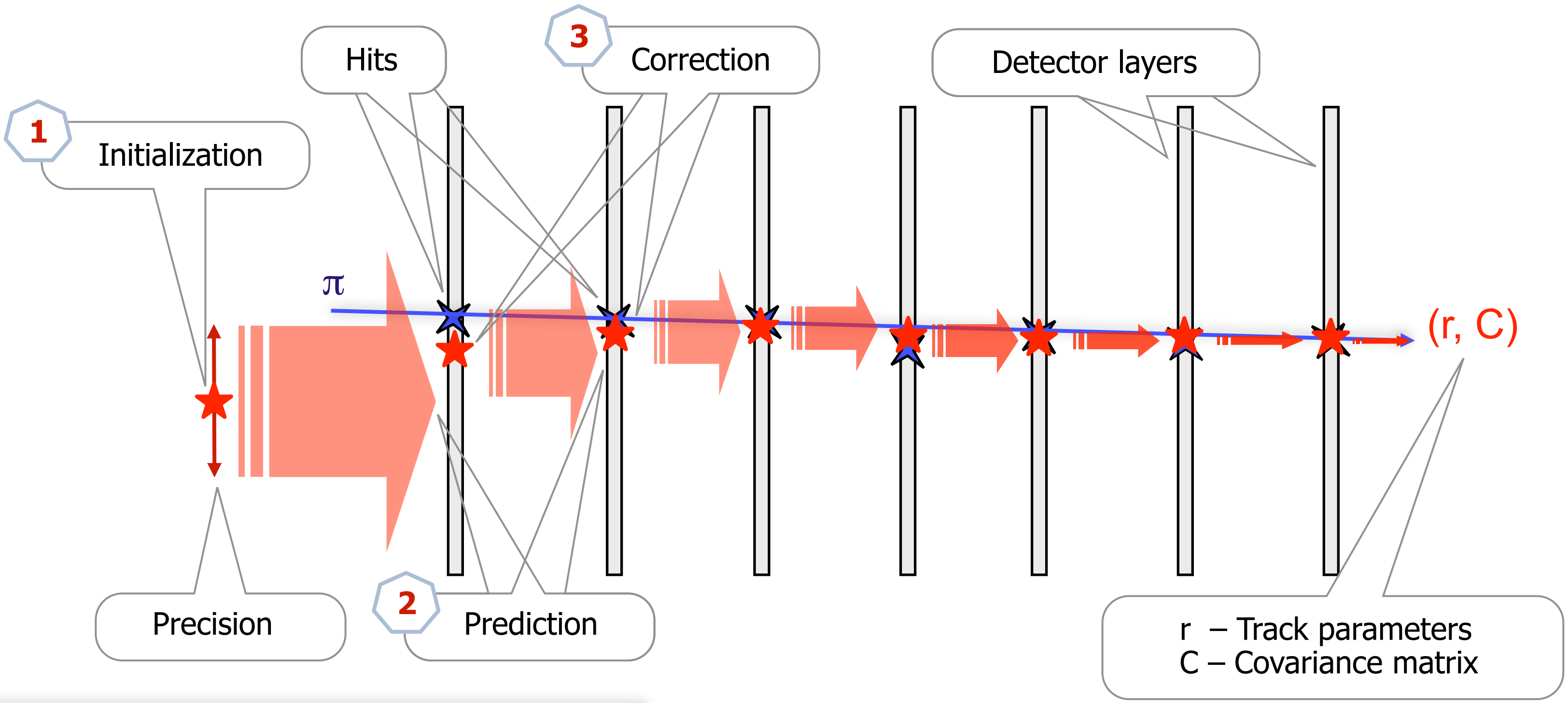
// Short-Lived Particles Finder



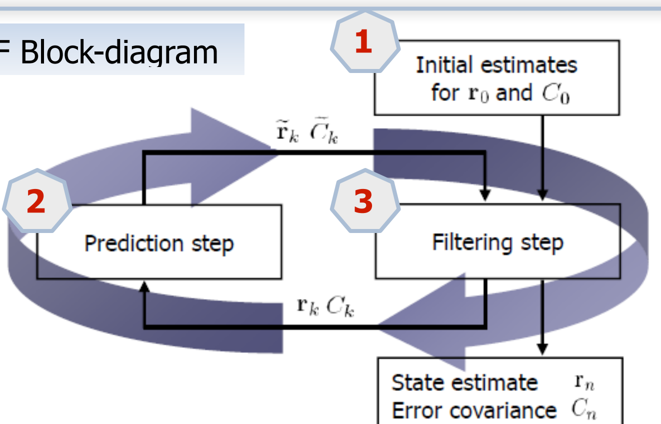
- **Kalman Filter**

Kalman Filter based Track Fit

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



KF Block-diagram



KF as a recursive least squares method

State vector

Position, direction and momentum

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

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 Library

Kalman Filter Methods

Kalman Filter Tools:
- KF Track Fitter
- KF Track Smoother
- Deterministic Annealing Filter

Kalman Filter Approaches:
- Conventional DP KF
- Conventional SP KF
- Square-Root SP KF
- UD-Filter SP
- Gaussian Sum Filter
- 3D (*x*,*y*,*z*) and 4D (*x*,*y*,*z*,*t*) KF

Track Propagation:
- Runge-Kutta
- Analytic Formula

Detector Types:
- Pixel
- Strip
- Tube
- TPC

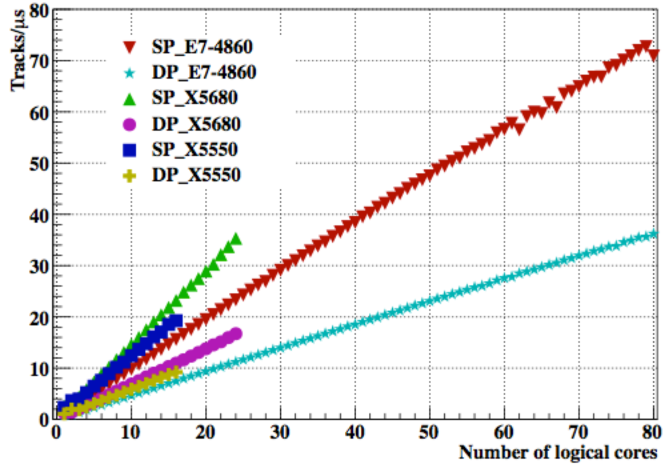
Implementations

Vectorization (SIMD):
- Header Files
- Vc Vector Classes
- ArBB Array Building Blocks
- OpenCL

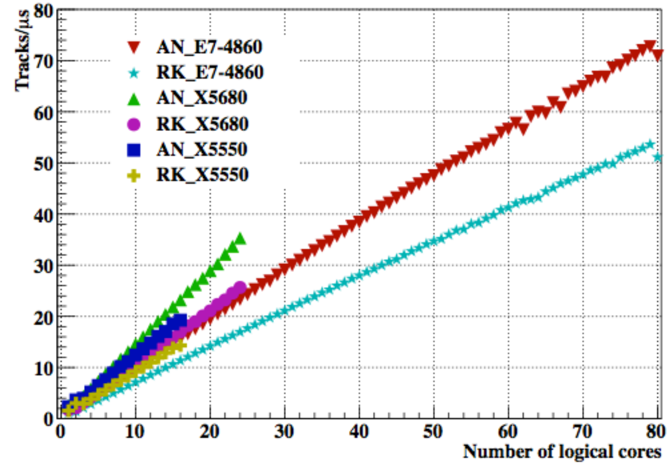
Parallelization (many-cores):
- Open MP
- ITBB
- ArBB
- OpenCL

Precision:
- single precision SP
- double precision DP

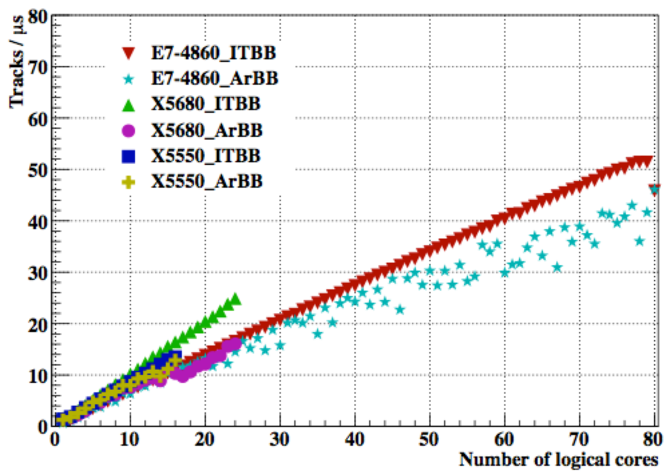
Conventional KF DP vs. SP



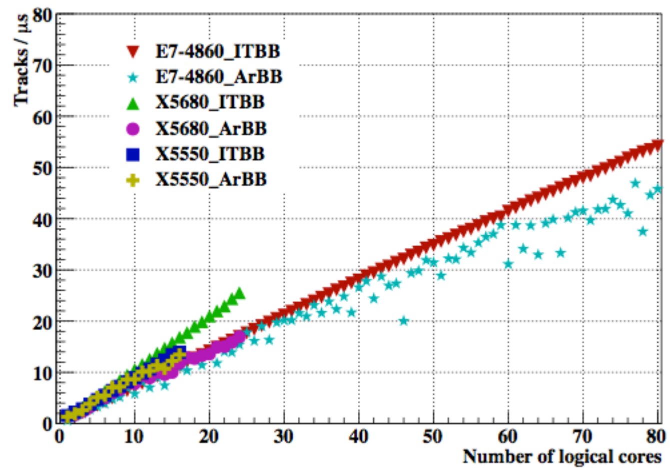
Conventional KF RK4 vs. Analytical



Square-Root KF



UD KF

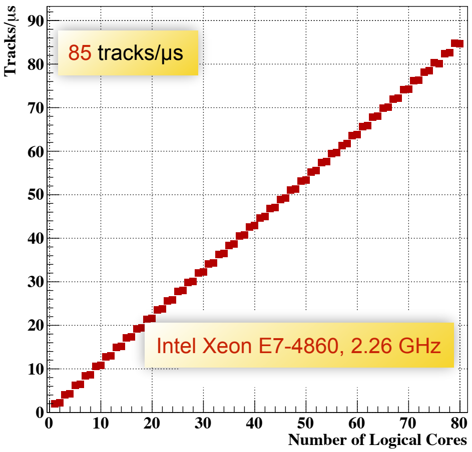


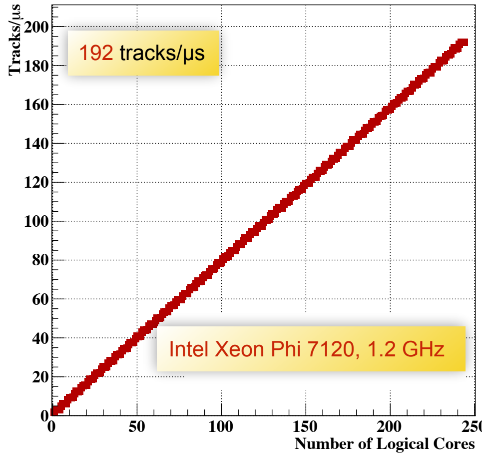
Strong many-core scalability of the Kalman filter library

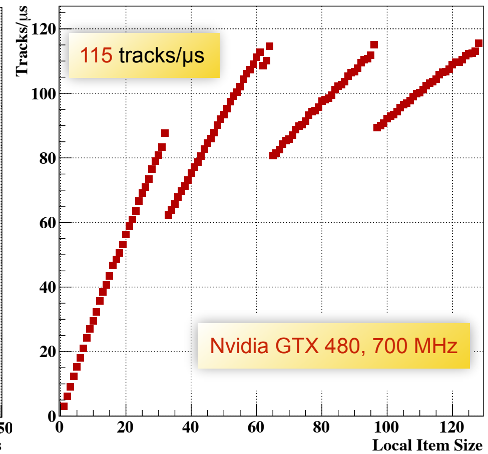
with I. Kulakov, H. Pabst* and M. Zyzak (*Intel)

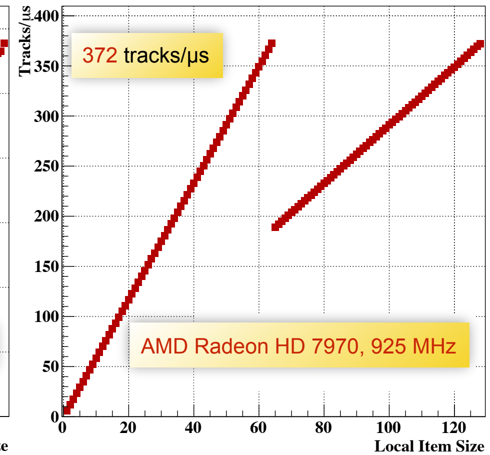
1

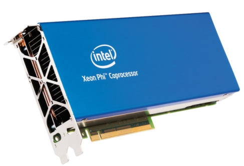
Full Portability of the KF Track Fit









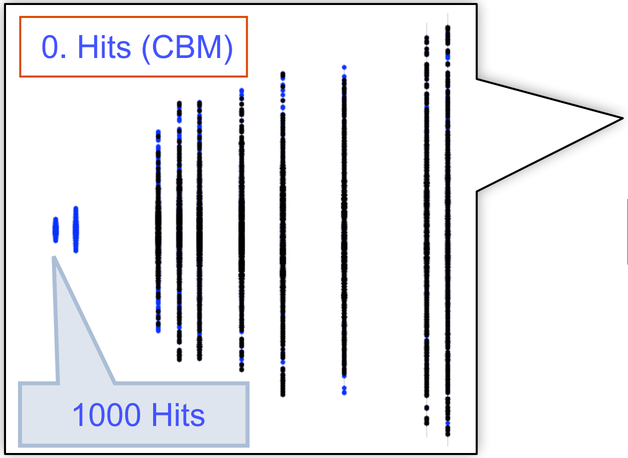


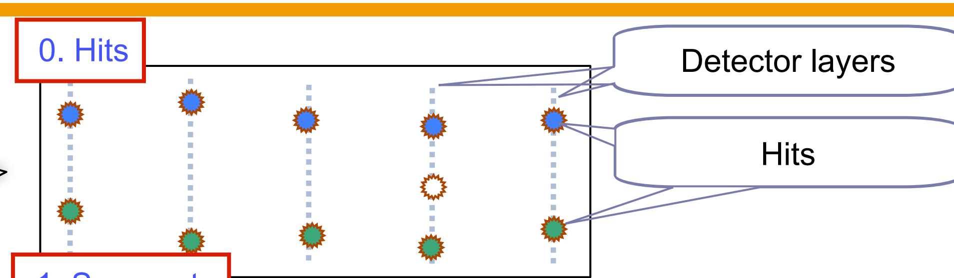
- Scalability with respect to the number of logical cores in a CPU is one of the most important parameters of the algorithm.
- The scalability on the Intel Xeon Phi coprocessor is similar to the CPU, but running four threads per core instead of two.
- In case of the graphics cards the set of tasks is 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.

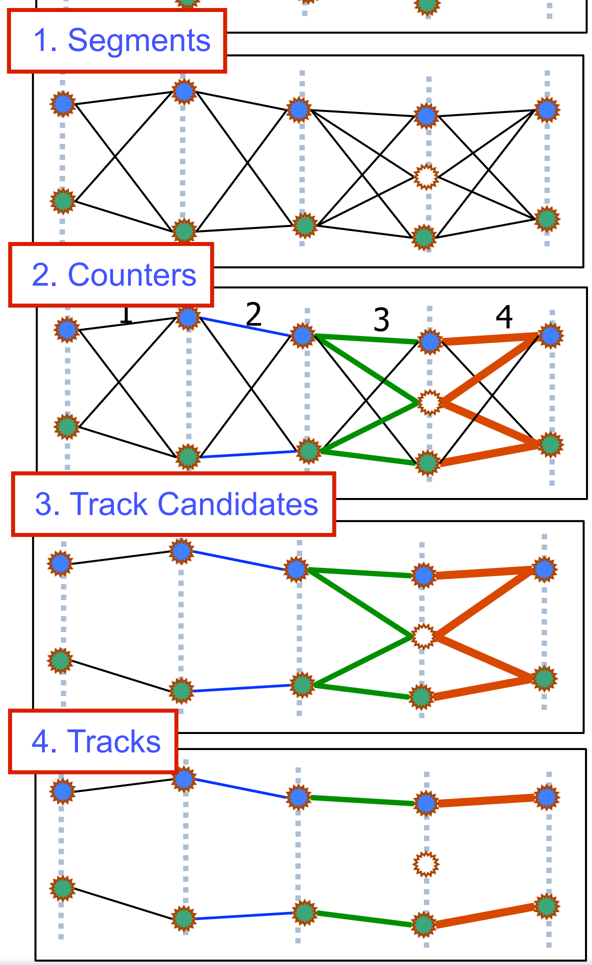
Single node KF Track Fit performance: 2*CPU+2*GPU = 10^9 tracks/s = (100 tracks/event)* 10^7 events/s = 10^7 events/s

Fast, precise and portable Kalman filter library

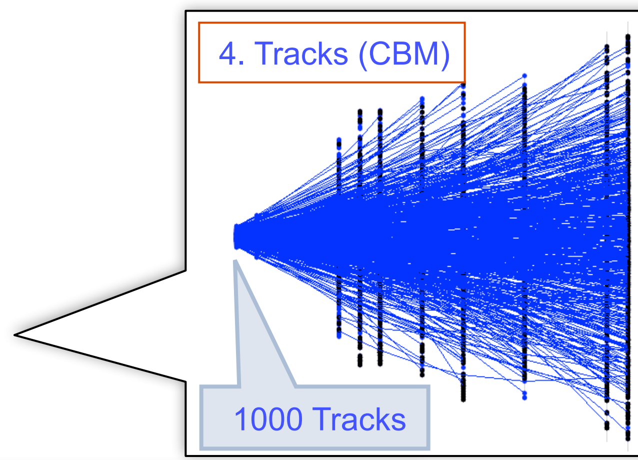
Cellular Automaton (CA) Track Finder







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.



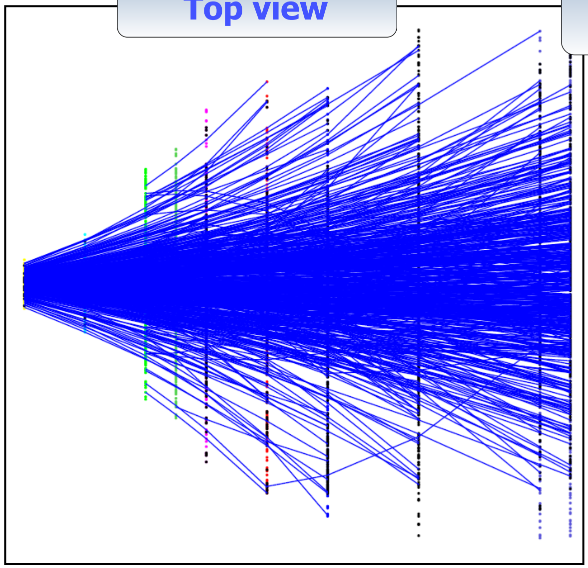
Cellular Automaton:
- local w.r.t. data
- intrinsically parallel
- extremely simple
- very fast

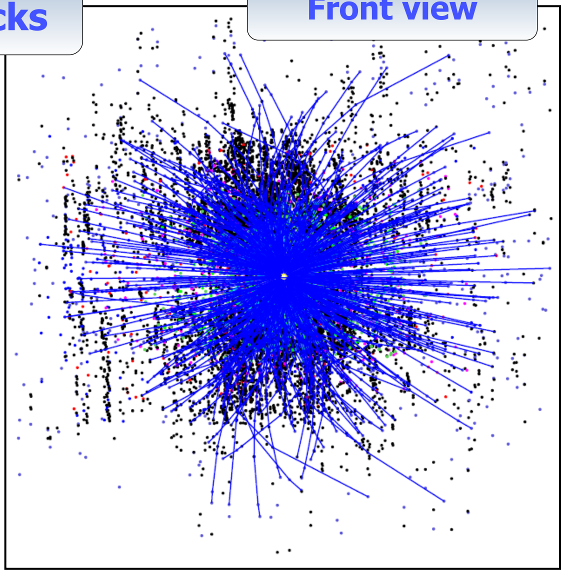
Perfect for many-core CPU/GPU !

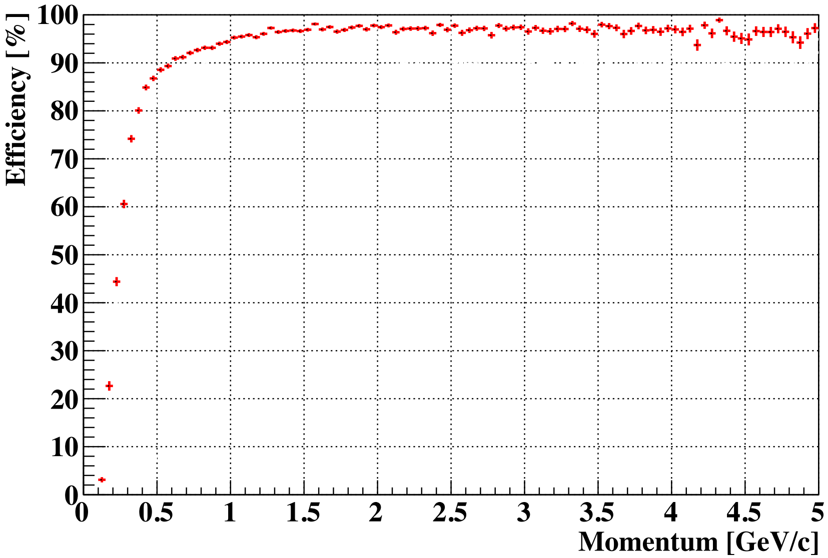
Useful for complicated event topologies with large combinatorics and for parallel hardware

CBM CA Track Finder: Efficiency

Top view | **770 Tracks** | **Front view**





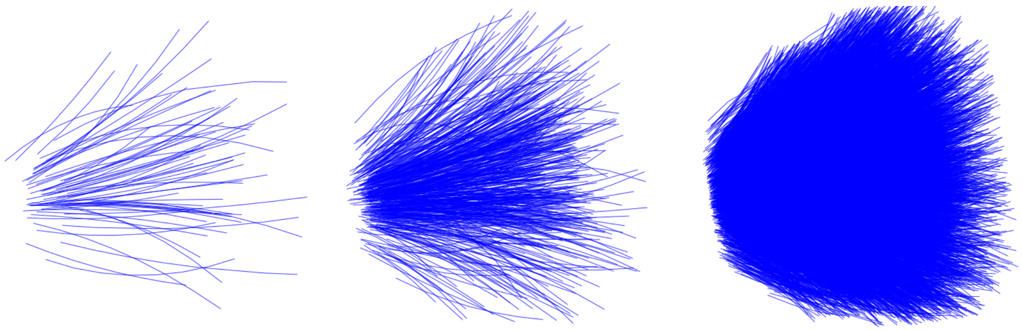


Track category	Eff, %
All tracks	90.9
Primary high-p	97.5
Primary low-p	92.6
Secondary high-p	91.1
Secondary low-p	63.8
Clone level	0.4
Ghost level	5.9
MC tracks found	134
Time, ms/ev	10

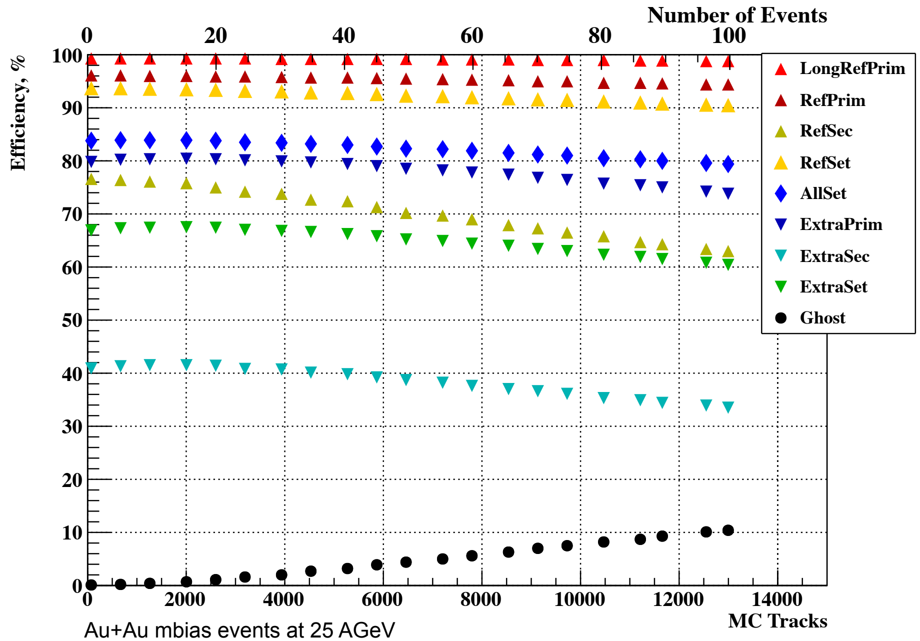
Efficient and stable event reconstruction

CA Track Finder at High Track Multiplicity

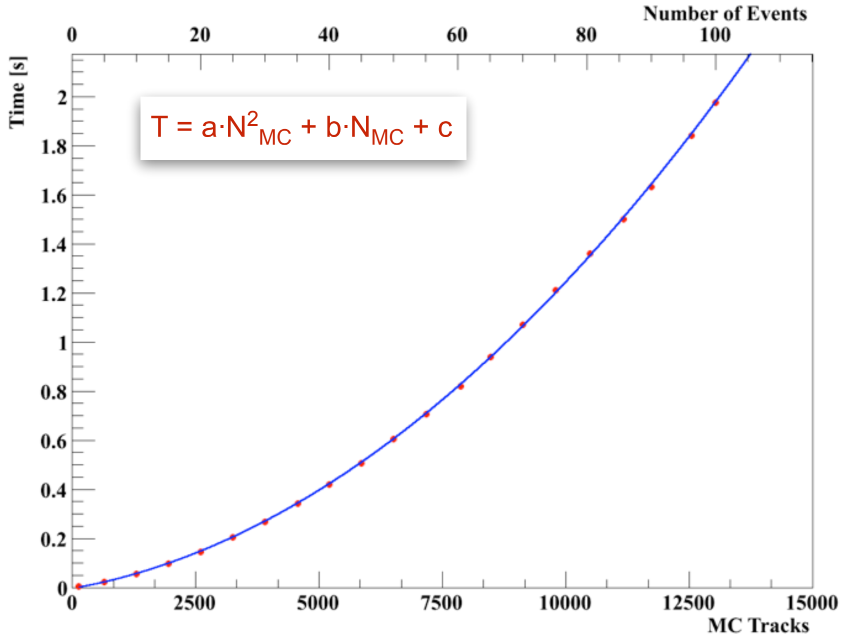
A number of minimum bias events is gathered into a group (super-event), which is then treated by the CA track finder as a single event



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

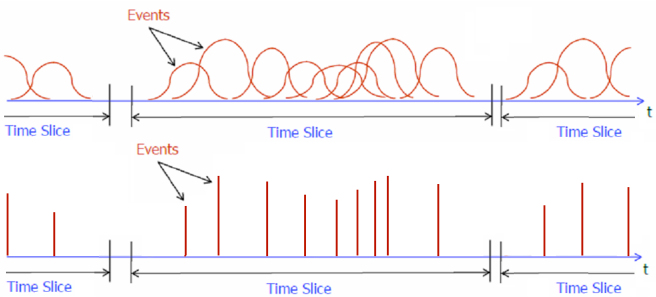


Au+Au mbias events at 25 AGeV



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

Time-based (4D) Track Reconstruction with CA Track Finder

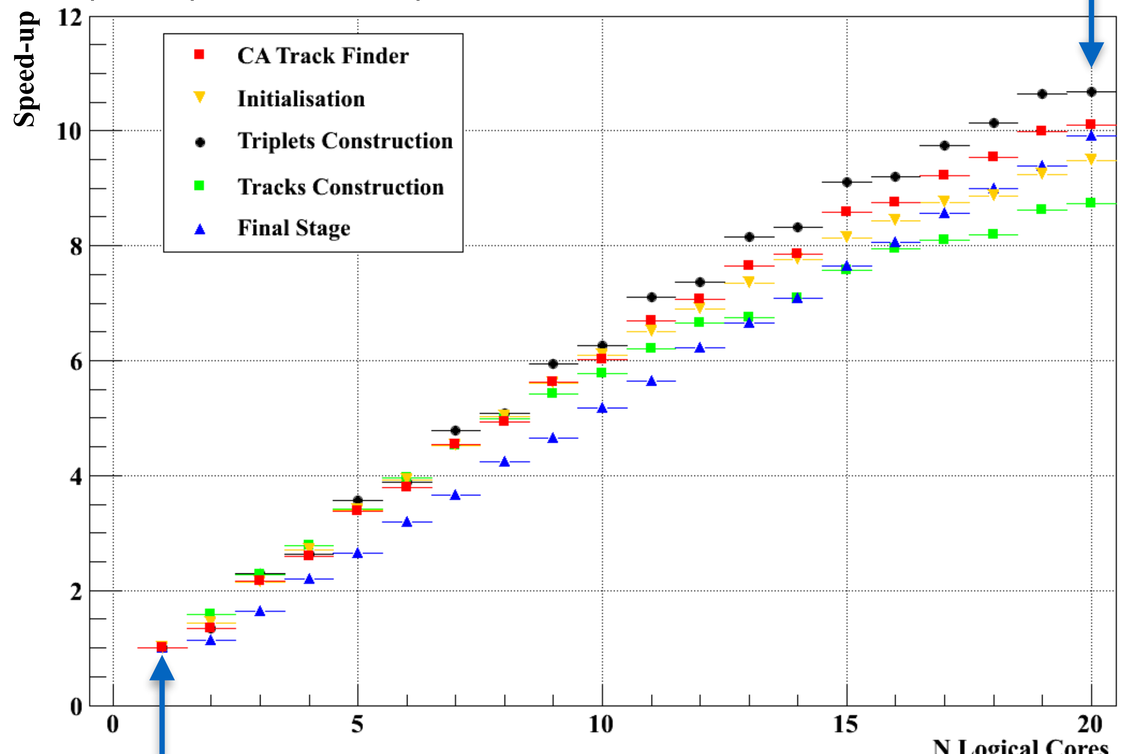


- The beam in the CBM will have no bunch structure, but continuous.
- Measurements in this case will be 4D (x, y, z, t).
- Significant overlapping of events in the detector system.
- Reconstruction of time slices rather than events is needed.

Efficiency, %	3D	3+1 D	4D
All tracks	83.8	80.4	83.0
Primary high-p	96.1	94.3	92.8
Primary low-p	79.8	76.2	83.1
Secondary high-p	76.6	65.1	73.2
Secondary low-p	40.9	34.9	36.8
Clone level	0.4	2.5	1.7
Ghost level	0.1	8.2	0.3
Time/event/core, ms	8.2	31.5	8.5

Total CA time = 84 ms

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



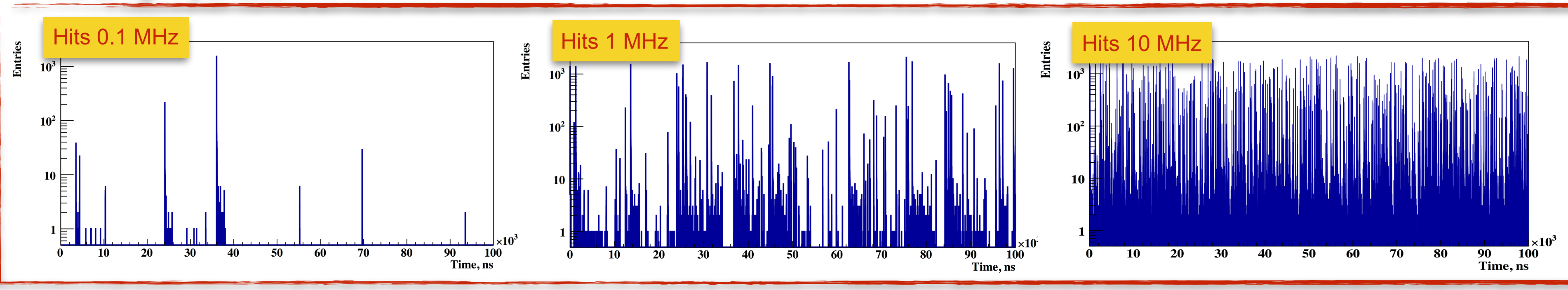
Total CA time = 849 ms

100 mbias events in a time-slice

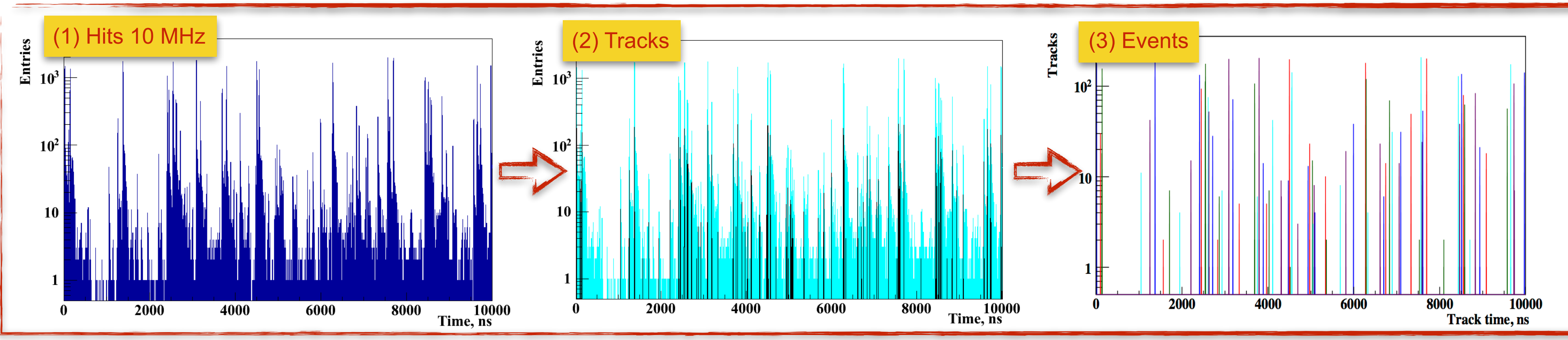
4D track reconstruction is scalable with the speed-up factor of 10.1; 3D reconstruction time 8.2 ms/event is recovered in 4D case

4D Event Building at 10 MHz

Hits at high input rates

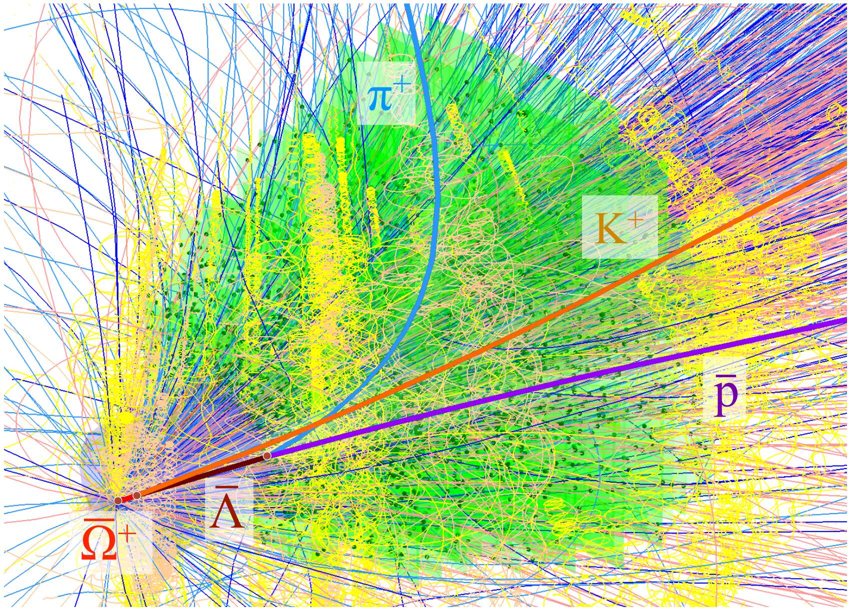


From hits to tracks to events



Reconstructed tracks clearly represent groups, which correspond to the original events
83% of single events, no splitted events, further analysis with TOF information at the vertexing stage

KF Particle: Reconstruction of Vertices and Decayed Particles



Simulated AuAu collision at 25 AGeV

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

```
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
```

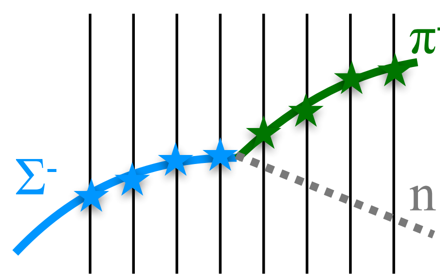
Concept:

- Mother and daughter particles have the same state vector and are treated in the same way
- Reconstruction of decay chains
- Kalman filter based
- Geometry independent
- Vectorized
- Uncomplicated usage

Functionality:

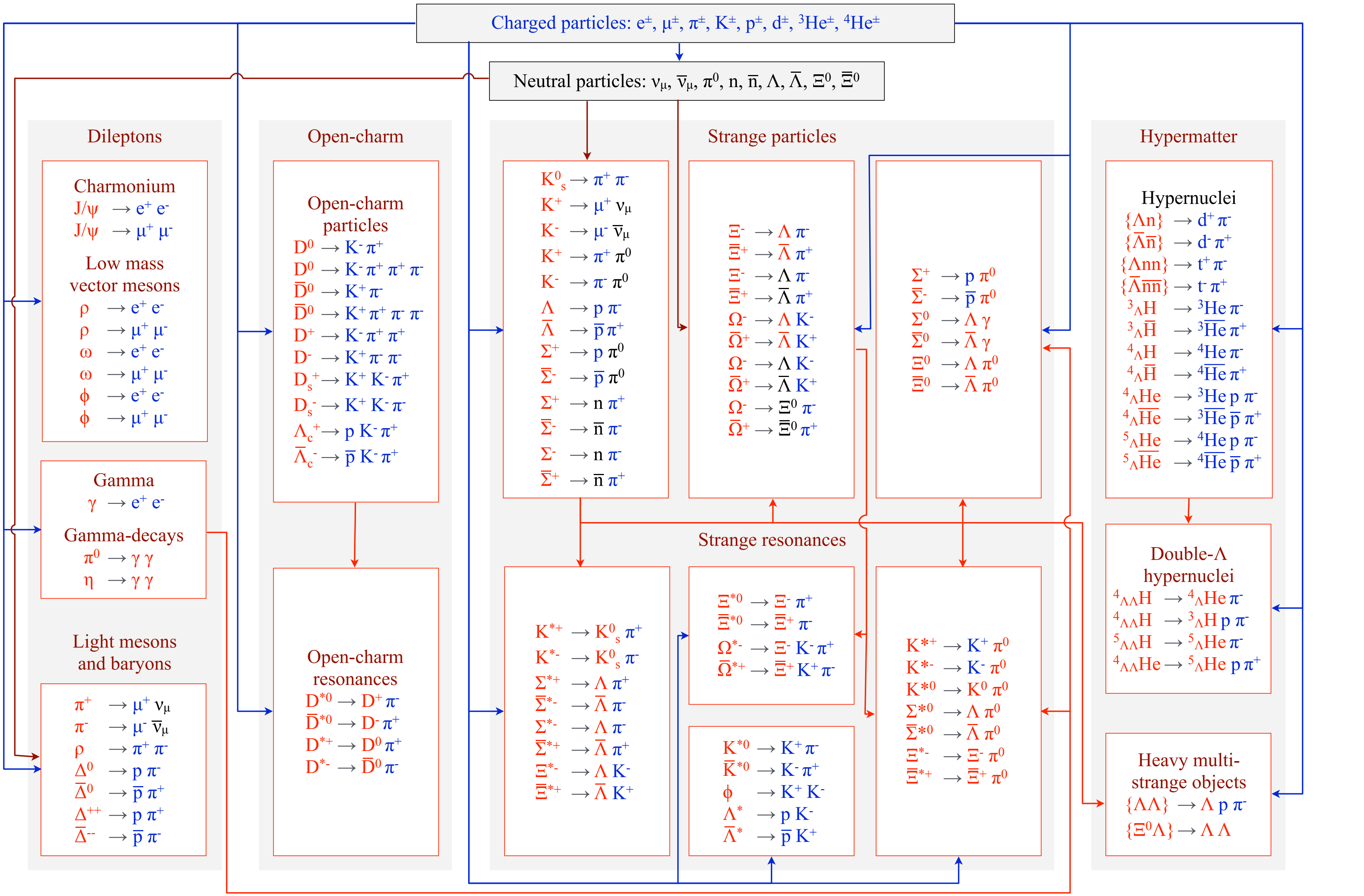
- Construction of short-lived particles
- Addition and subtraction of particles
- Transport
- Calculation of an angle between particles
- Calculation of distances and deviations
- Constraints on mass, production point and decay length
- KF Particle Finder

Reconstruction of decays with a neutral daughter
by the missing mass method:



KF Particle provides a simple and direct approach to physics analysis (used in CBM, ALICE and STAR)

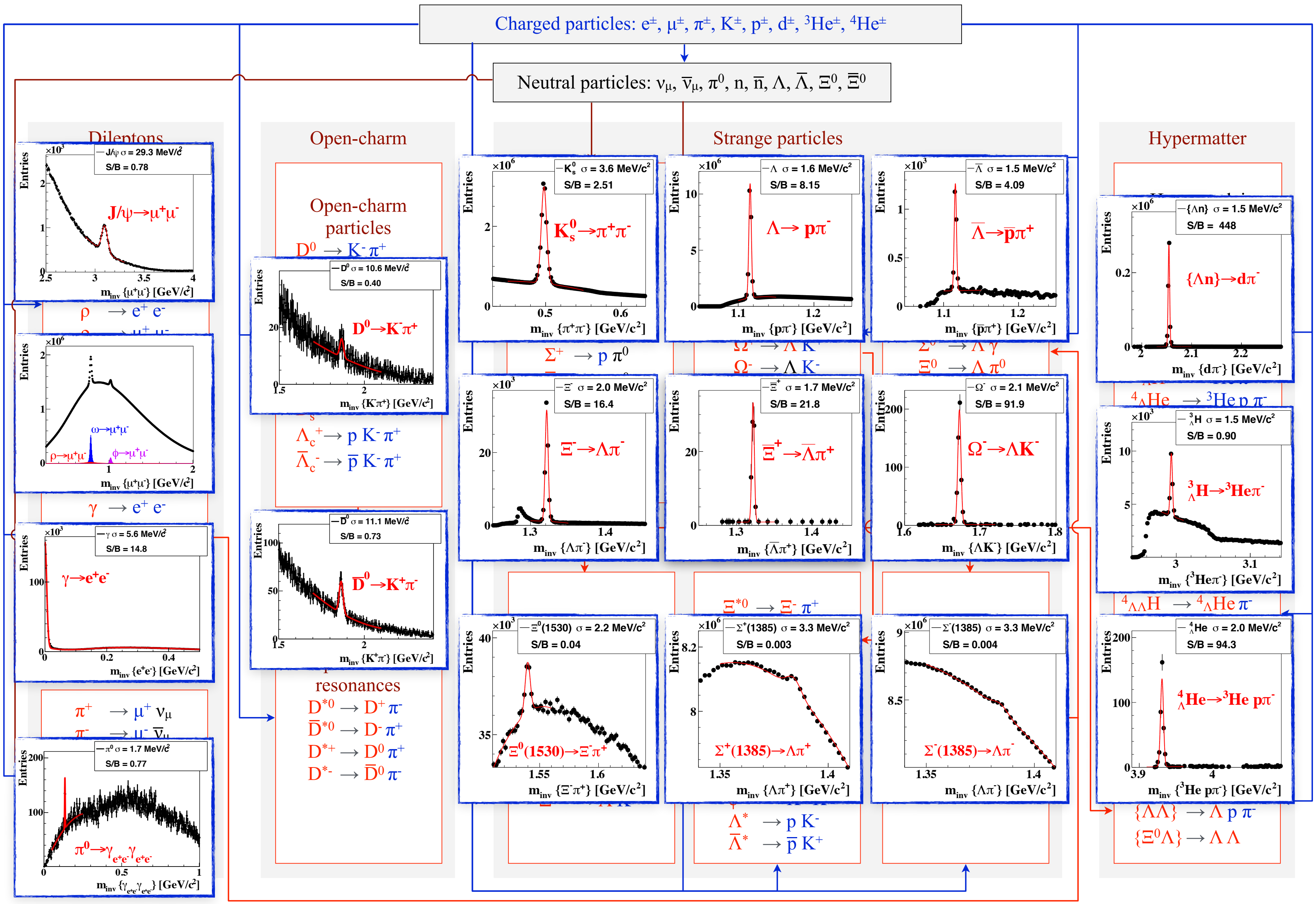
KF Particle Finder for Physics Analysis and Selection



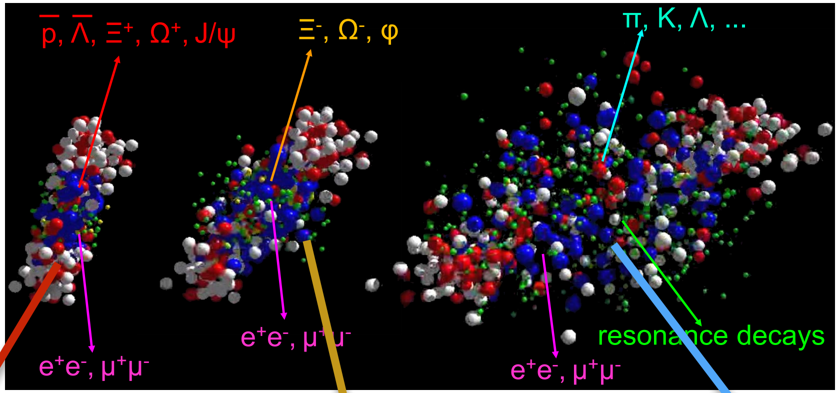
(mbias: 1.4 ms; central: 10.5 ms)/event/core

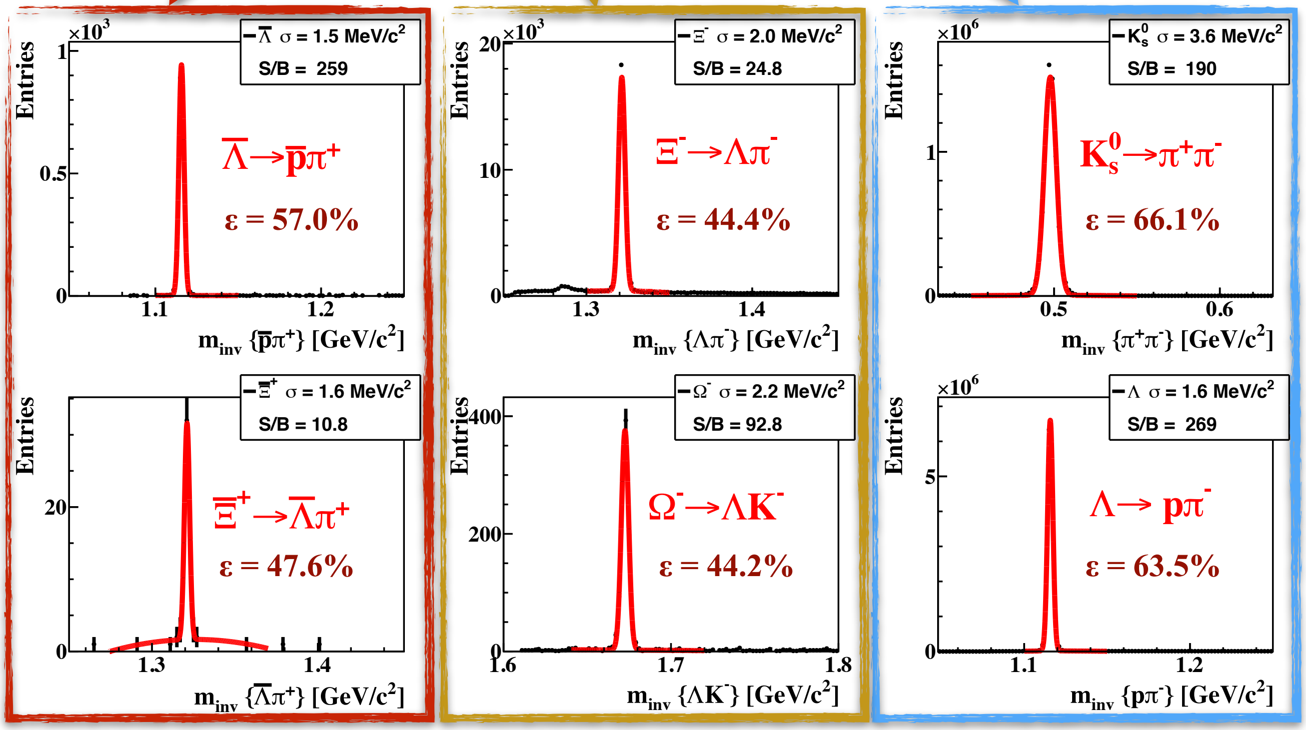
3

KF Particle Finder for Physics Analysis and Selection



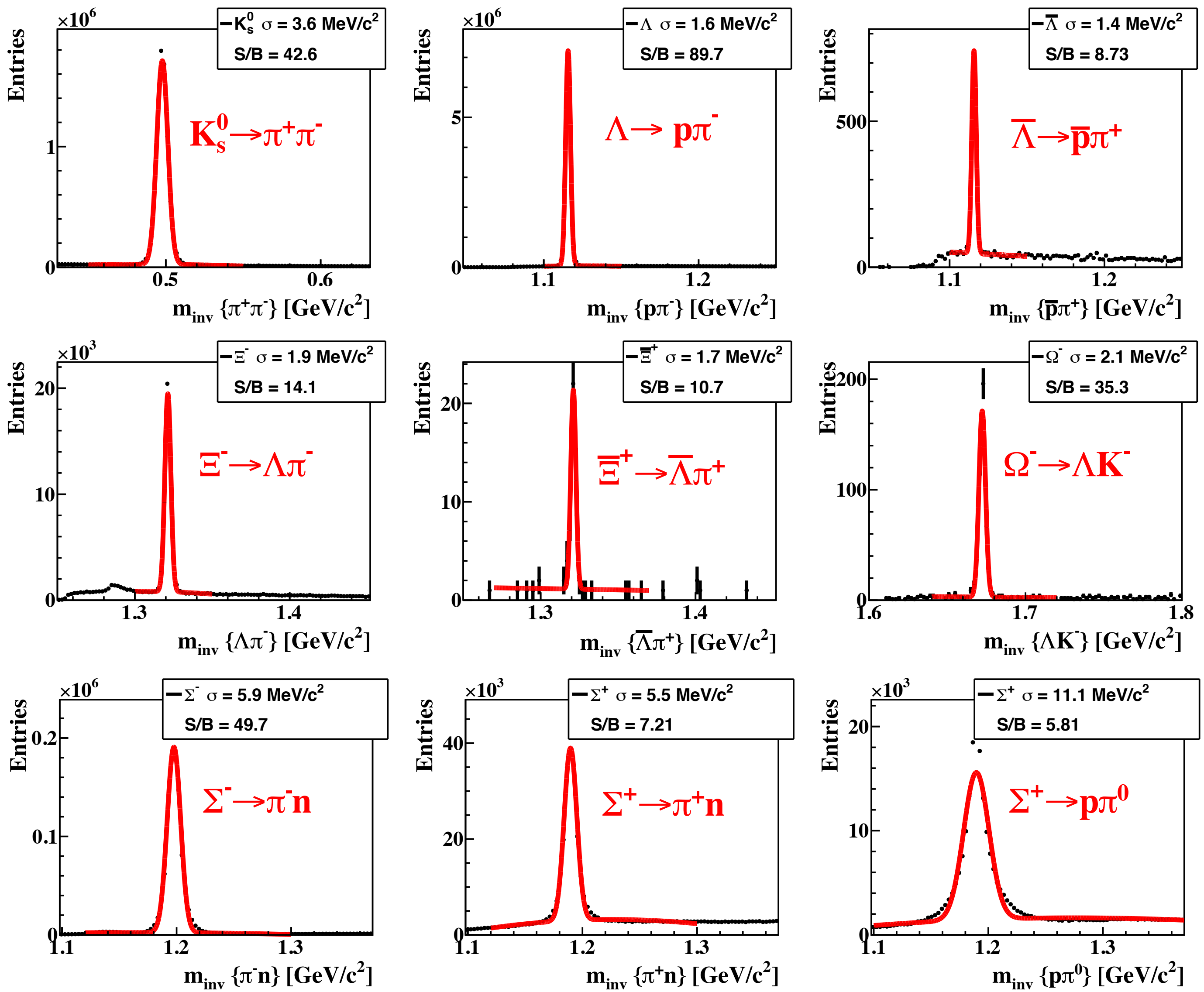
Clean Probes of Collision Stages





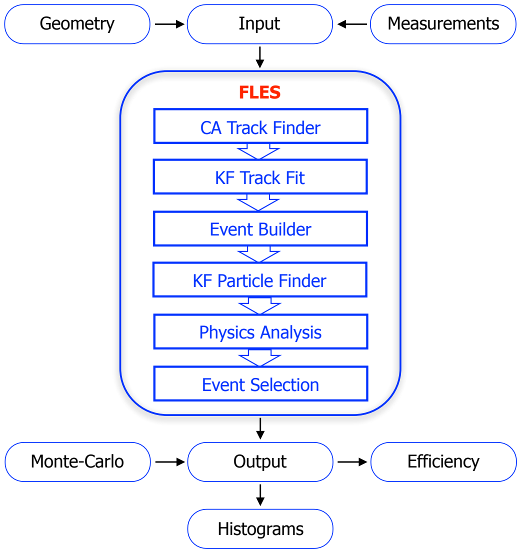
AuAu, 10 AGeV, 3.5M central UrQMD events, MC PID

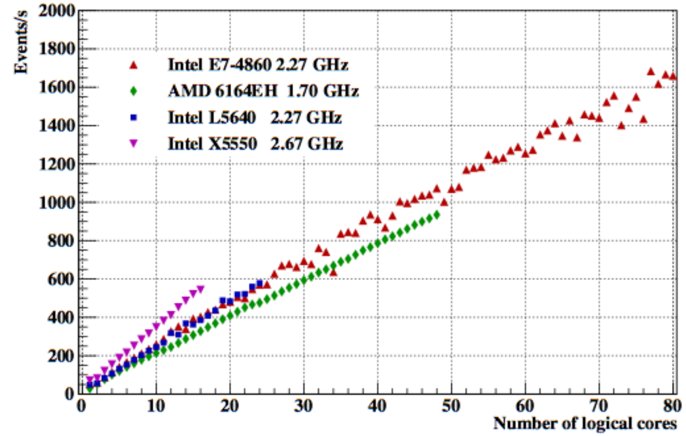
Clean Probes of Collision Stages

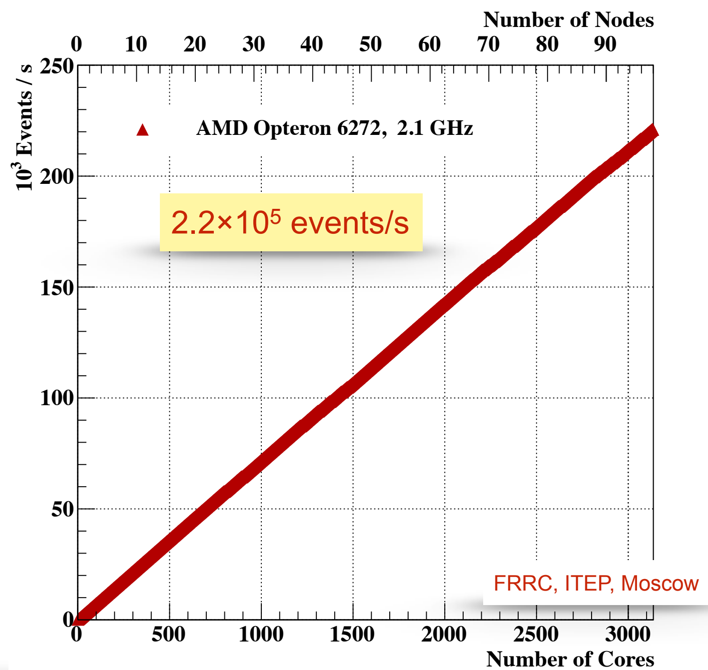


5M central AuAu UrQMD events at 10 AGeV with realistic PID

CBM Standalone First Level Event Selection (FLES) Package







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

Summary

- ✓ CBM will explore the QCD phase diagram in the region of high net baryon densities
- ✓ Efficient and clean reconstruction of long-lived primary particles with the CA track finder
- ✓ KF particle finder is a universal platform for short-lived particles reconstruction and physics analysis in on- and off-line modes
- ✓ Clean reconstruction of long- and short-lived particles produced at different stages of heavy-ion collisions
- ✓ Reconstruction is highly parallelized and vectorized for use on many-core CPU/Phi/GPU computer architectures
- ✓ Towards a common package for full event topology reconstruction in the CBM, ALICE and STAR heavy ion experiments