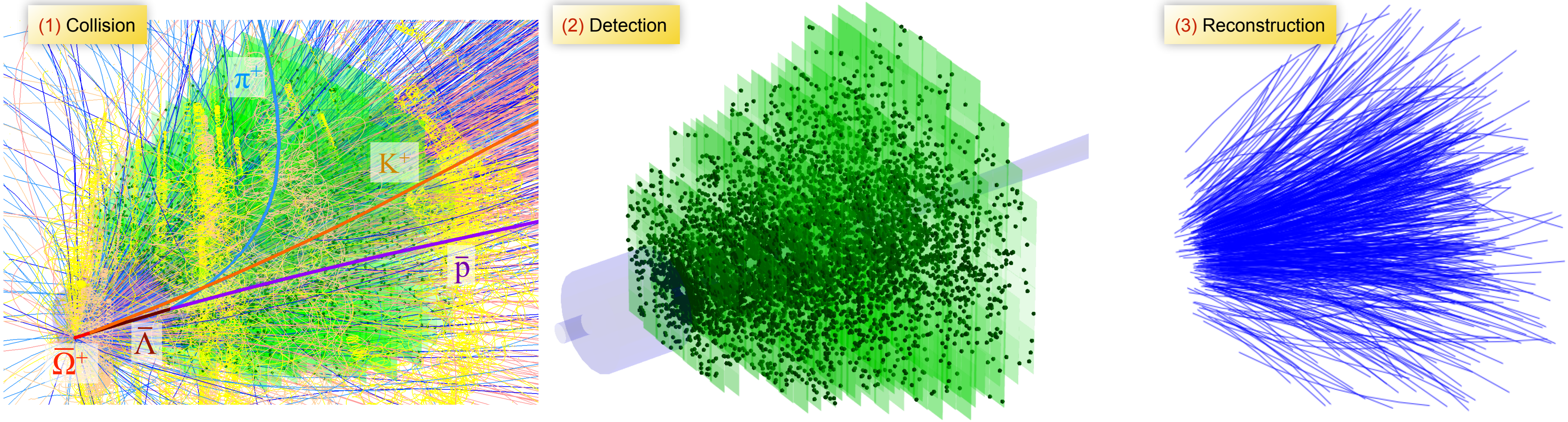
Online Event Reconstruction in the CBM Experiment at FAIR

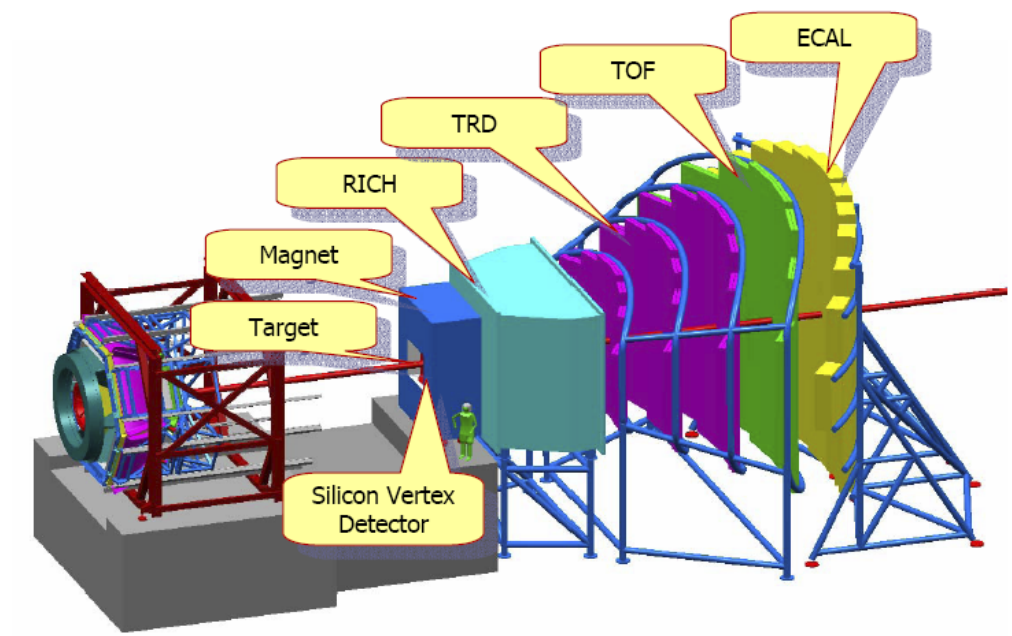
Valentina Akishina and Ivan Kisel
for the CBM Collaboration

Goethe-University Frankfurt am Main
FIAS Frankfurt Institute for Advanced Studies

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.

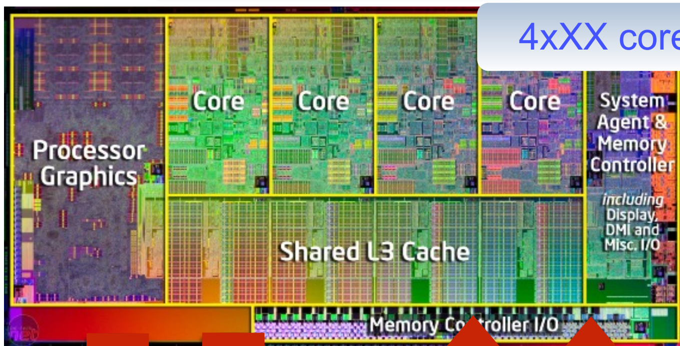
Cellular Automaton (CA) Track Finder
Kalman Filter (KF) Track Fitter
KF short-lived Particle Finder

All reconstruction algorithms are vectorized and parallelized.

Many-Core CPU/GPU Architectures

Intel/AMD CPU

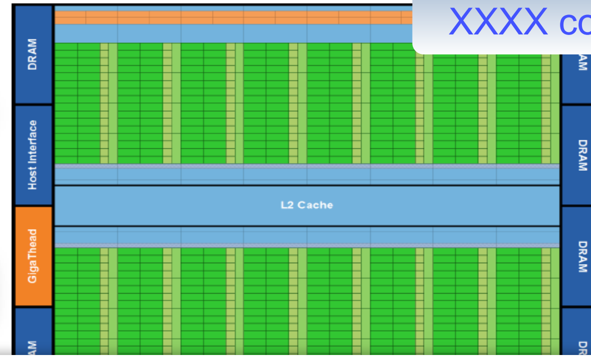
4xXX cores



- Optimized for low-latency access to cached data sets
- Control logic for out-of-order and speculative execution

Nvidia/ATI GPU

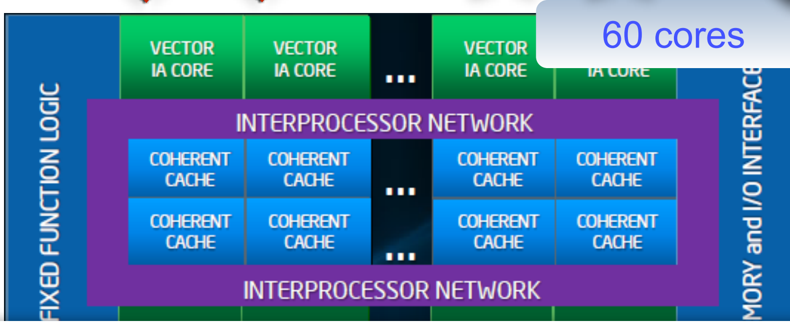
XXXX cores



- Optimized for data-parallel, throughput computation
- More transistors dedicated to computation

Intel Phi

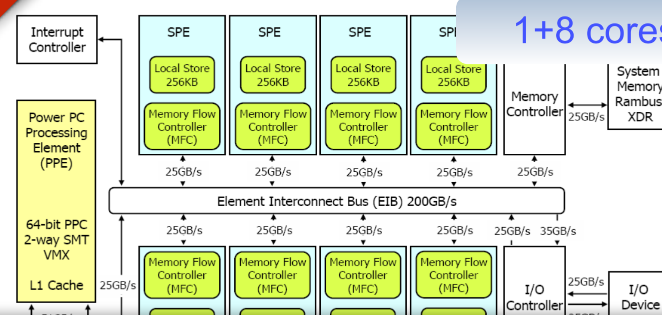
60 cores



- 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

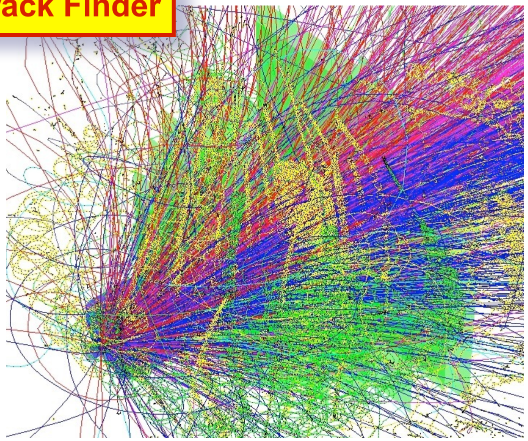


- General purpose RISC processor (PowerPC)
- 8 co-processors (SPE, Synergistic Processor Elements)
- 128-bit wide SIMD units

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

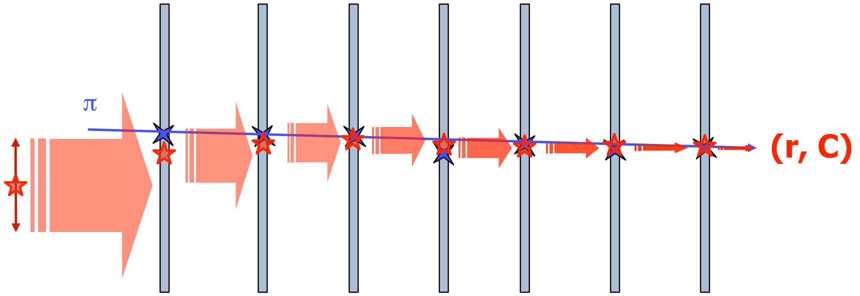
Stages of Event Reconstruction

// Track Finder



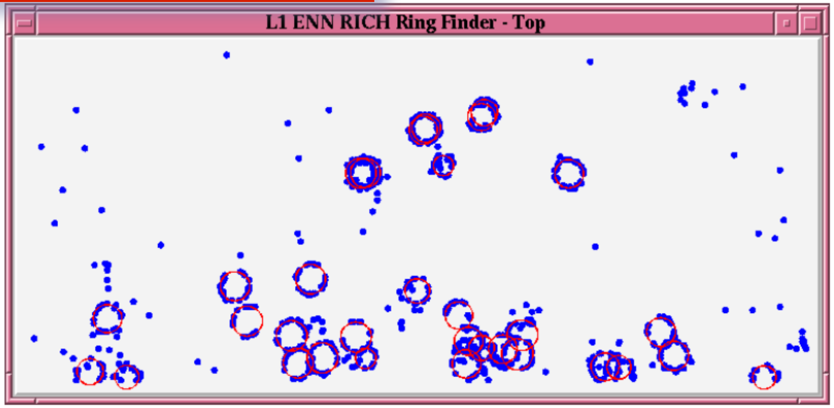
- Conformal Mapping
- Hough Transformation
- Track Following
- **Cellular Automaton**

// Track Fitter



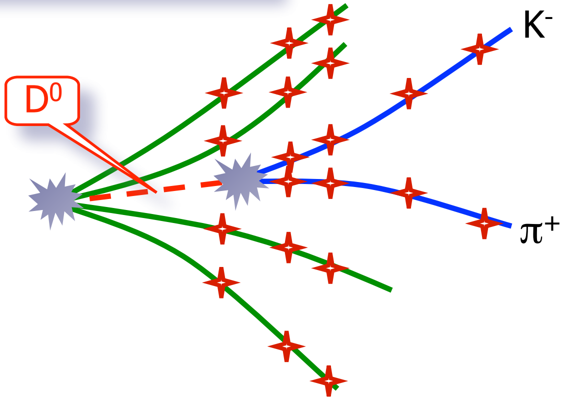
- **Kalman Filter**

// Ring Finder (Particle ID)



- Hough Transformation
- Elastic Neural Net

// Short-Lived Particles Finder

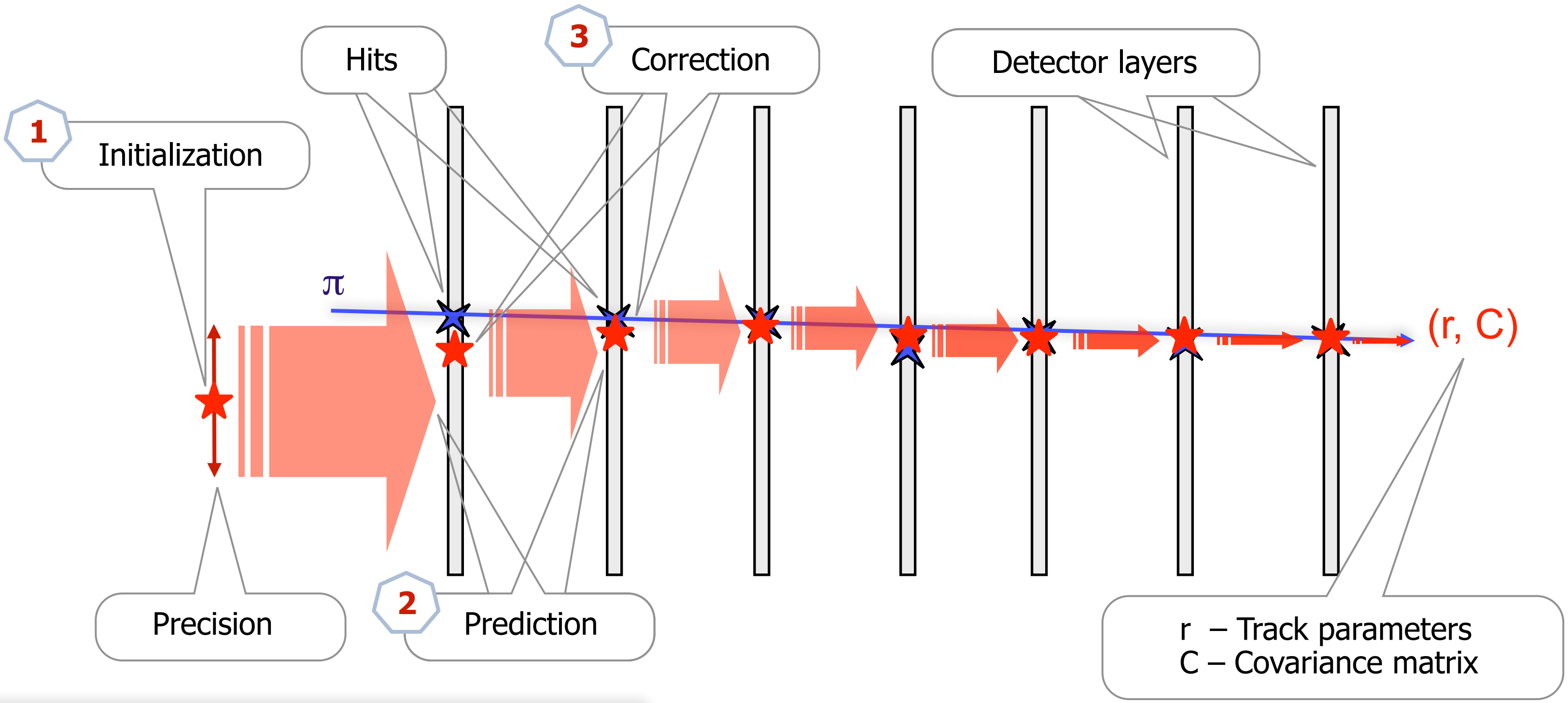


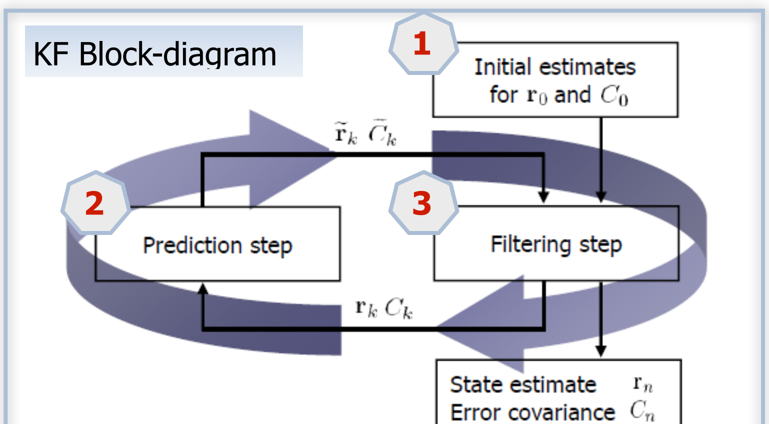
- **Kalman Filter**

1

Kalman Filter based Track Fit

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





KF as a recursive least squares method

State vector

Position, direction and momentum

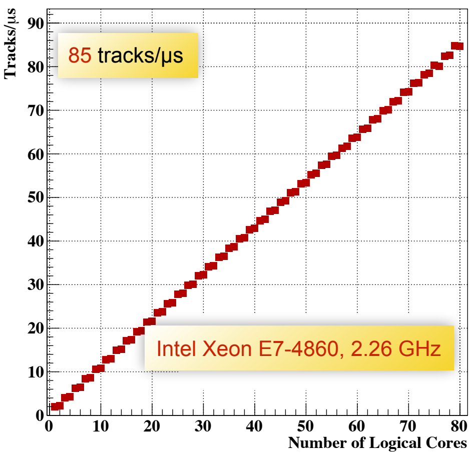
$r = \{ x, y, z, p_x, p_y, p_z \}$

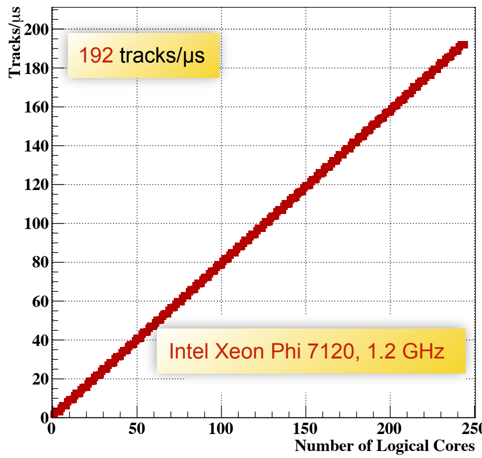
Kalman Filter:

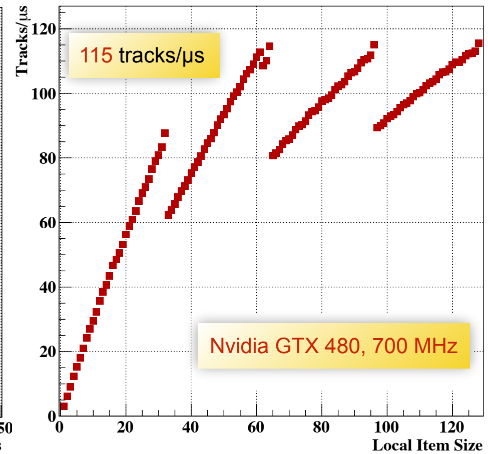
1. Start with an arbitrary initialization.
2. Add one hit after another.
3. Improve the state vector.
4. Get the optimal parameters after the last hit.

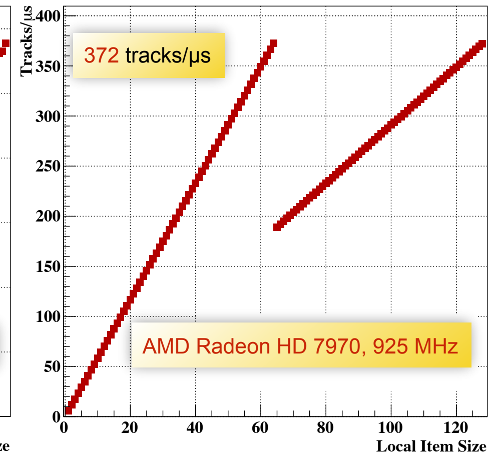
Nowadays the Kalman Filter is used in almost all HEP experiments

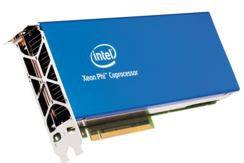
Full Portability of the KF Track Fit







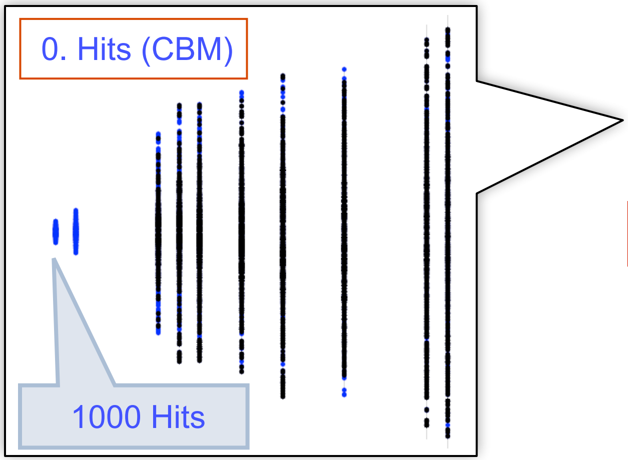


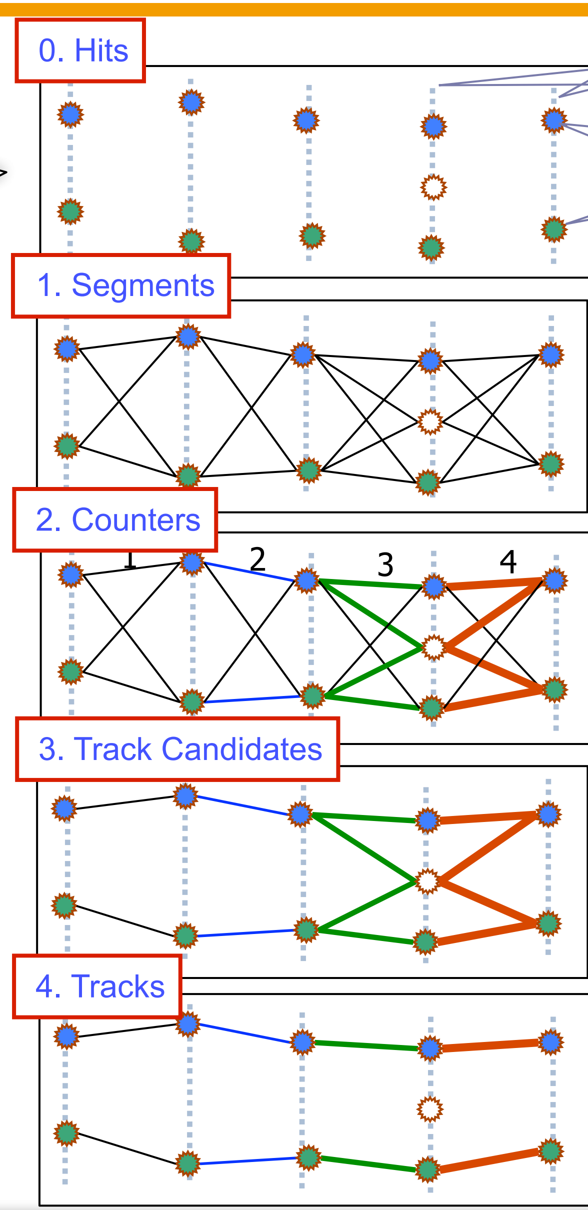


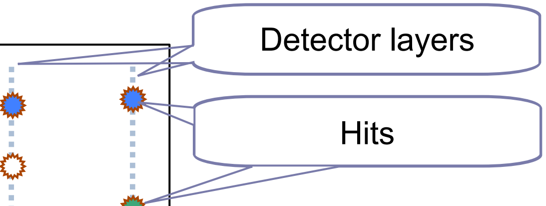
- Scalability with respect to the number of logical cores in a CPU is one of the most important parameters of the algorithm.
- The scalability on the Intel Xeon Phi coprocessor is similar to the CPU, but running four threads per core instead of two.
- In case of the graphic cards the set of tasks is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

Full portability of the Kalman filter library

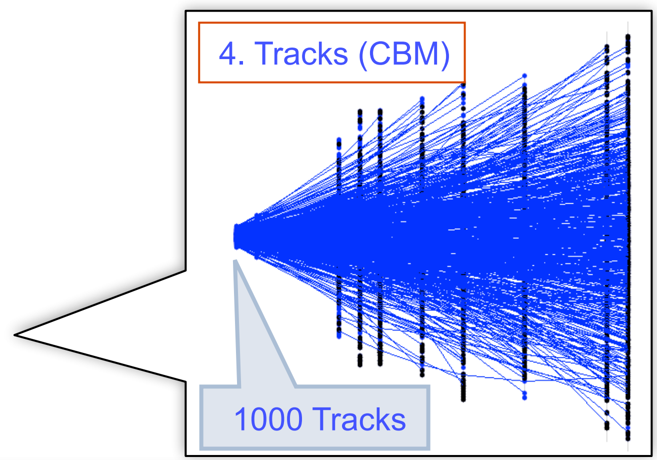
Cellular Automaton (CA) as Track Finder







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.

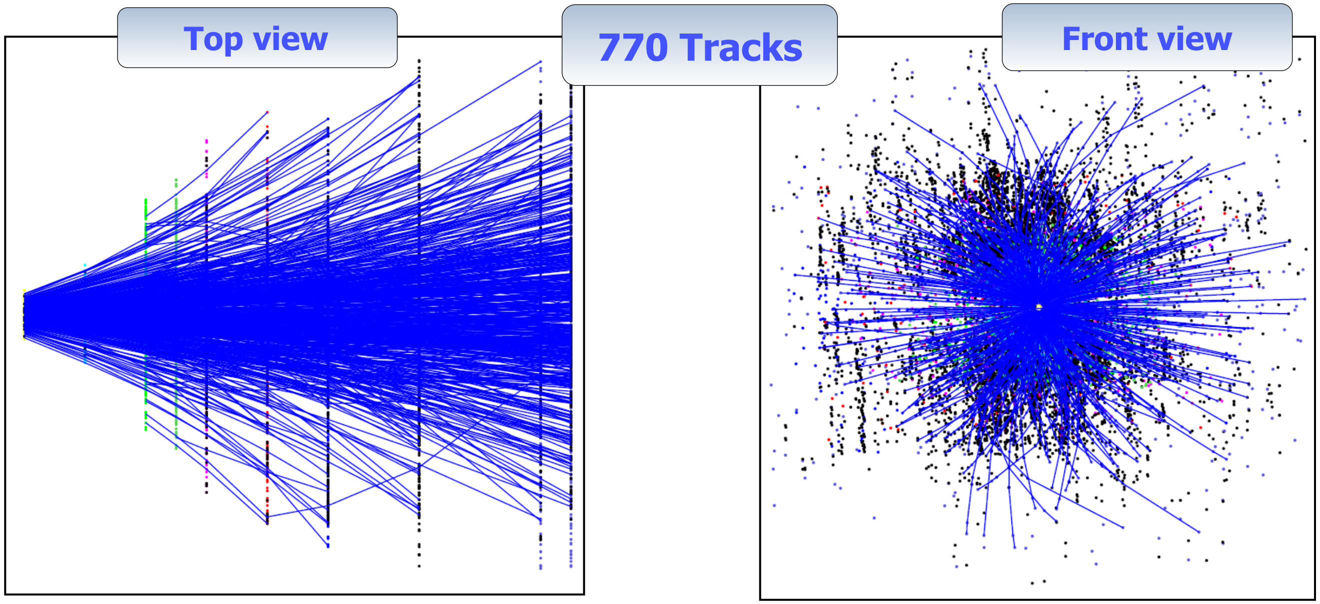


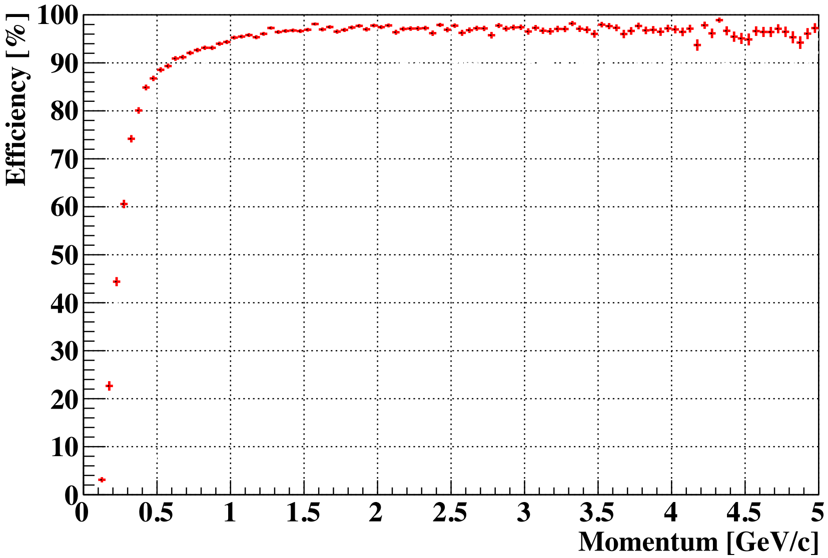
Cellular Automaton:
- local w.r.t. data
- intrinsically parallel
- extremely simple
- very fast

Perfect for many-core CPU/GPU !

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

CBM CA Track Finder: Efficiency



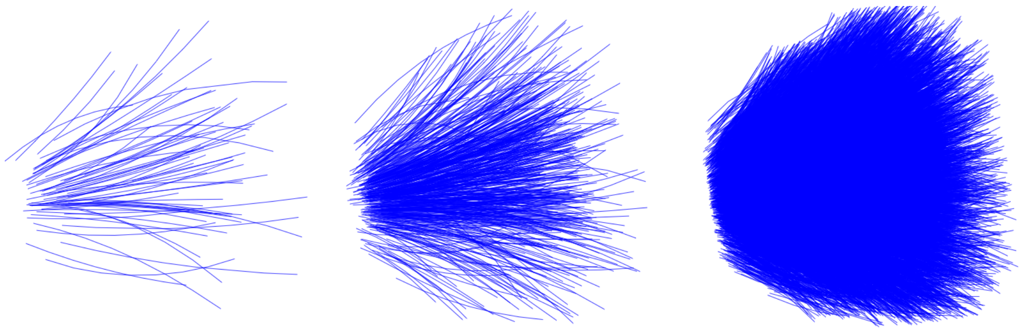


Track category	Eff, %
All tracks	90.9
Primary high-p	97.5
Primary low-p	92.6
Secondary high-p	91.1
Secondary low-p	63.8
Clone level	0.4
Ghost level	5.9
MC tracks found	134
Time, ms/ev	10

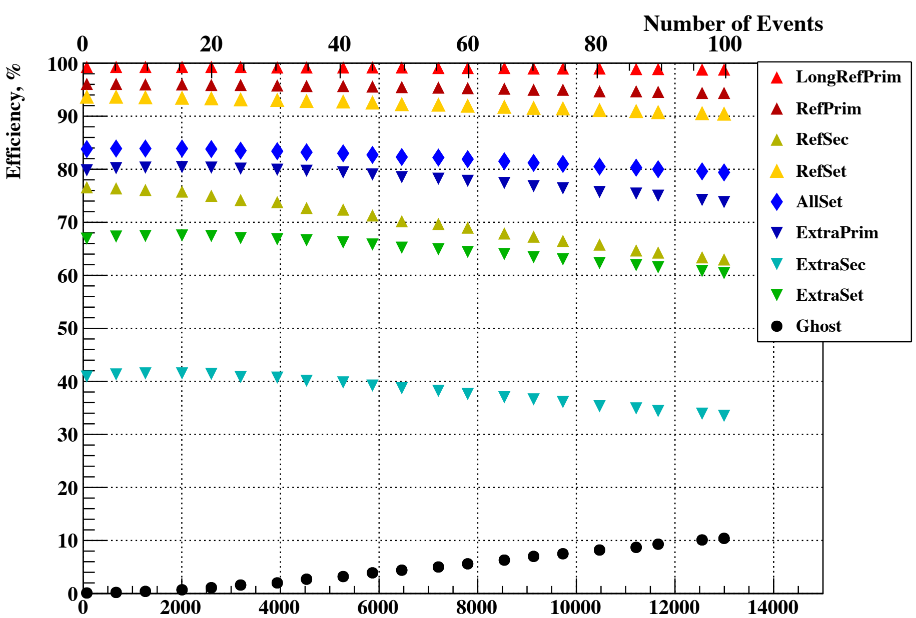
Efficient and stable event reconstruction

CA Track Finder at High Track Multiplicity

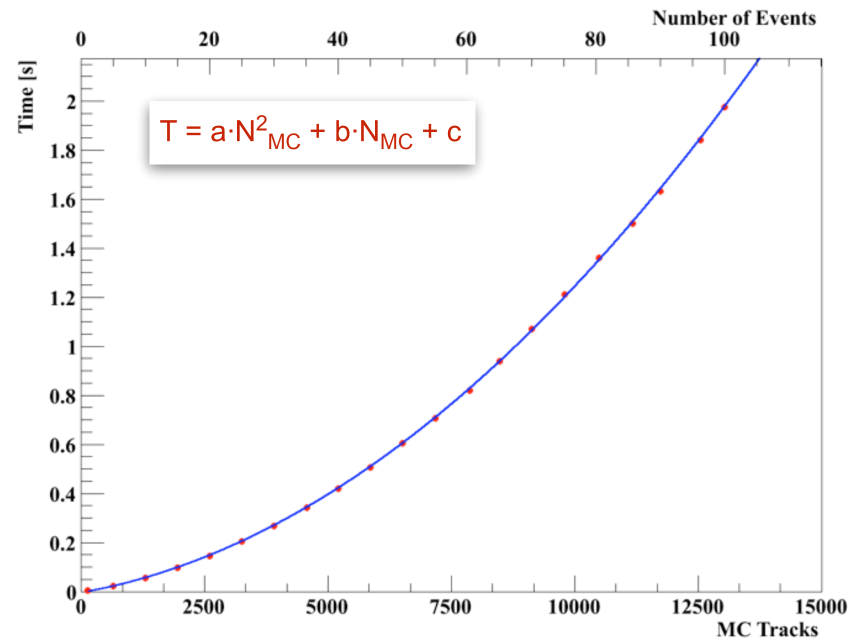
A number of minimum bias events is gathered into a group (super-event), which is then treated by the CA track finder as a single event



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

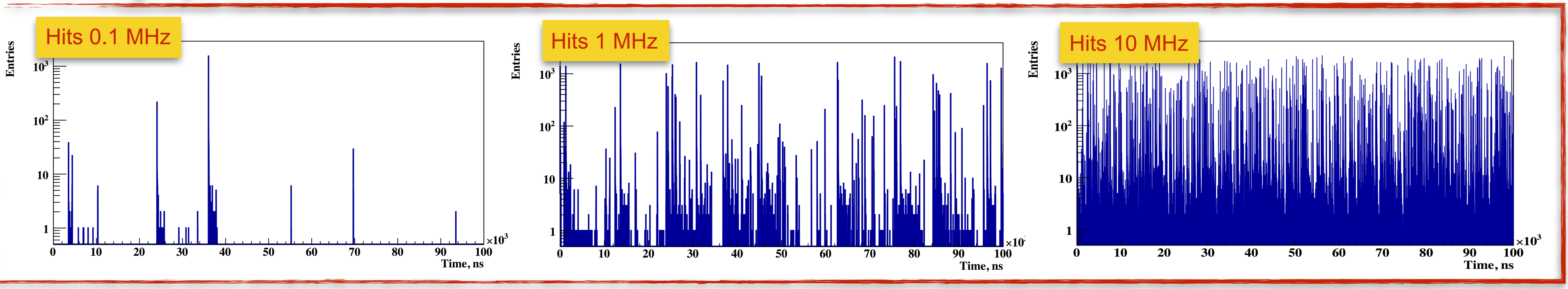


Stable reconstruction efficiency and time as a second order polynomial w.r.t. to track multiplicity

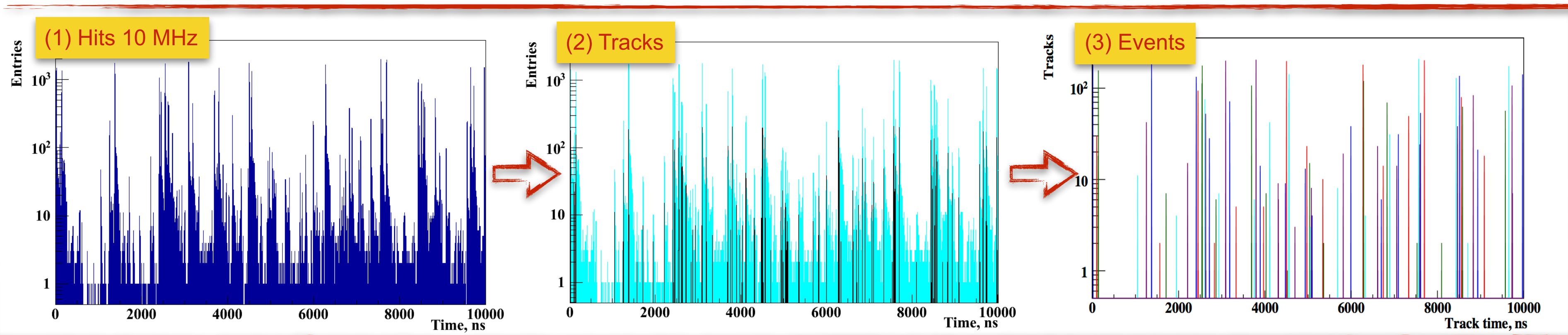
2

4D Event Building at 10 MHz

Hits at high input rates

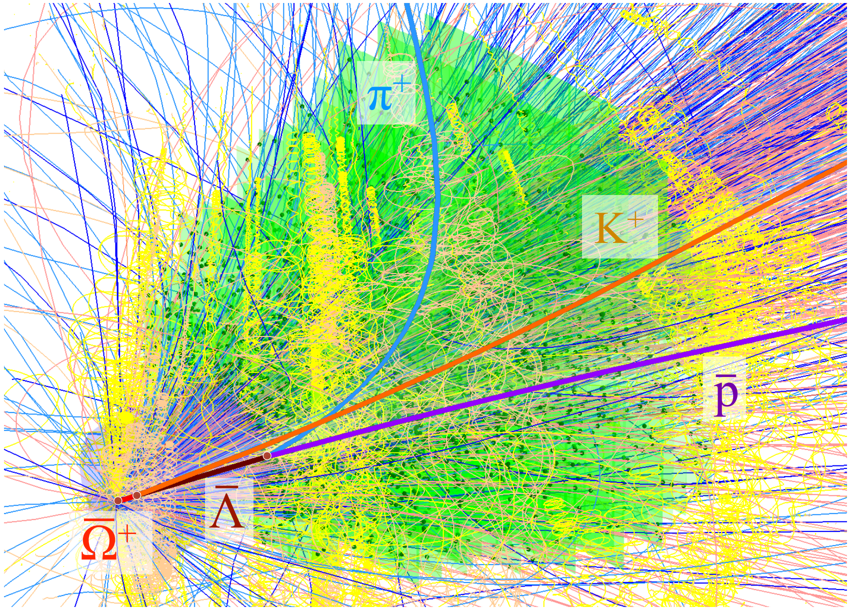


From hits to tracks to events



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

KFParticle: 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
```

State vector

Position, direction, momentum and energy

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

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:

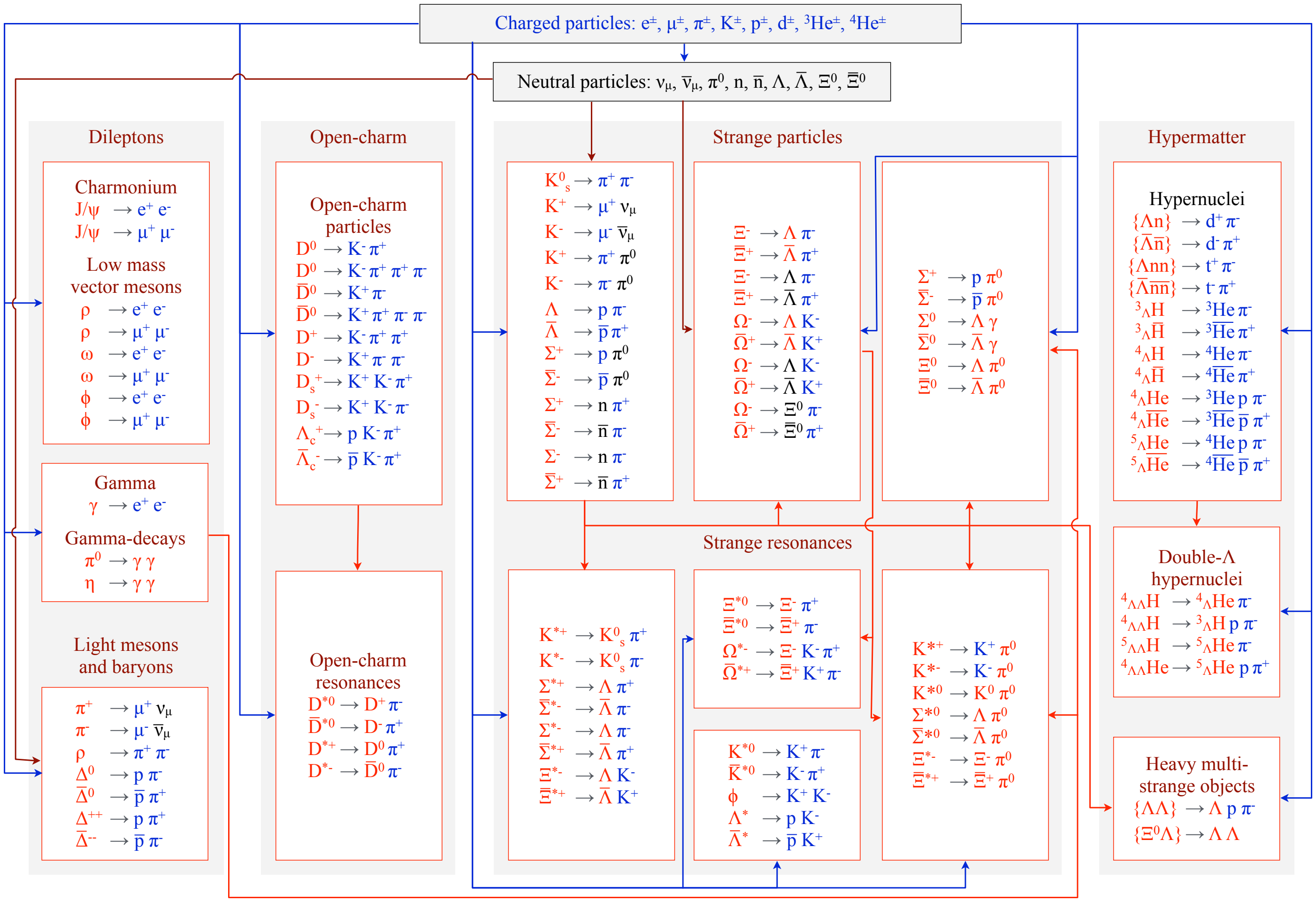
- Construction of short-lived particles
- Addition and subtraction of particles
- Transport
- Calculation of an angle between particles
- Calculation of distances and deviations
- Constraints on mass, production point and decay length
- KF Particle Finder

KFParticle provides uncomplicated approach to physics analysis (used in CBM, ALICE and STAR)

KF Particle Finder for Physics Analysis and Selection

The CBM experiment with 10^7 interactions/s requires full event reconstruction and selection online. This is a task of the First Level Event Selection (FLES) software module:

1. reconstruct ***charged particles*** observed directly in the detector;
2. identify ***short-lived particles*** via their hadronic or leptonic decay products;
3. select events with interesting particles for further offline physics analysis on a reduced (up to 10^4) data sample.

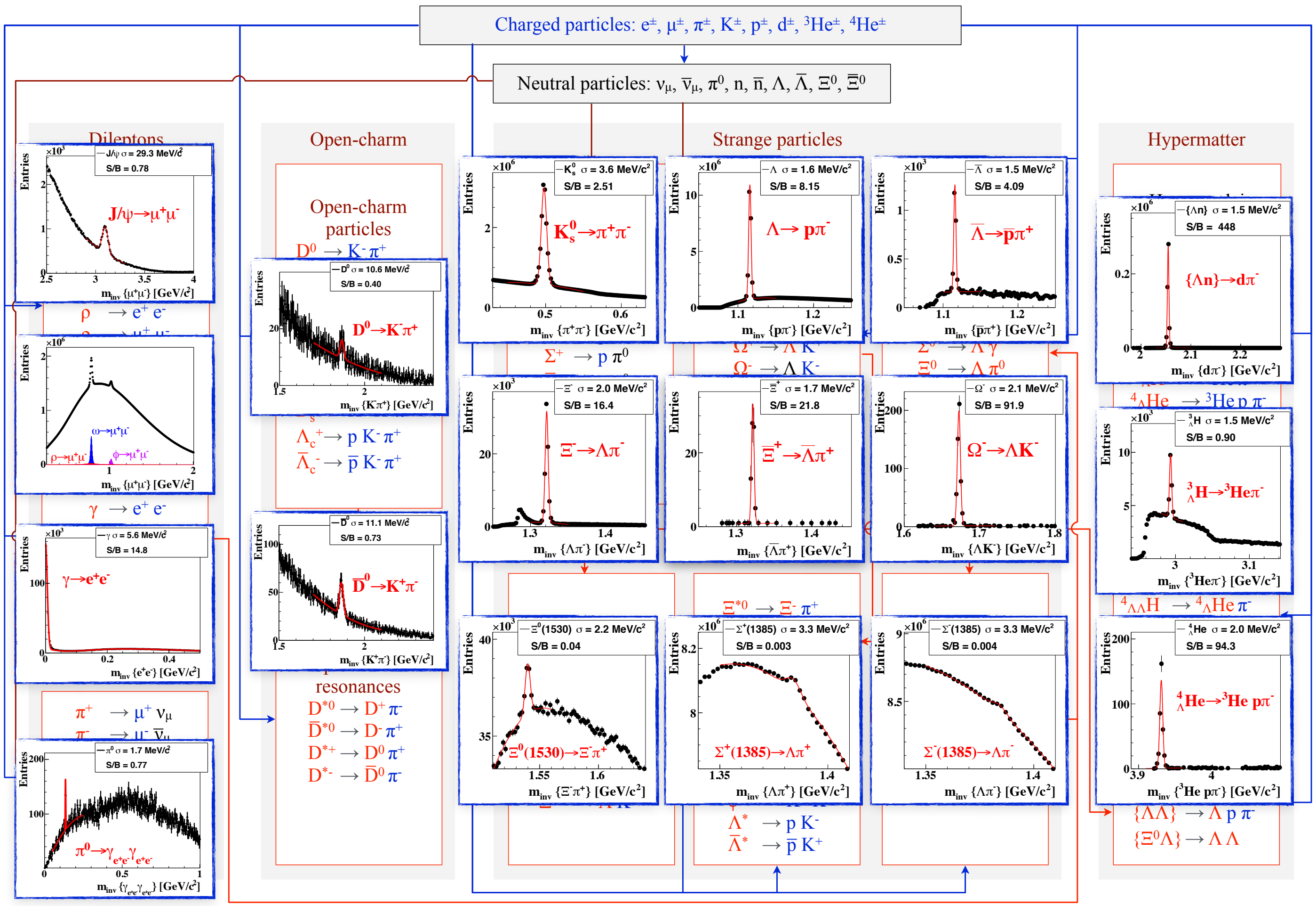


(mbias: 1.4 ms; central: 10.5 ms)/event/core

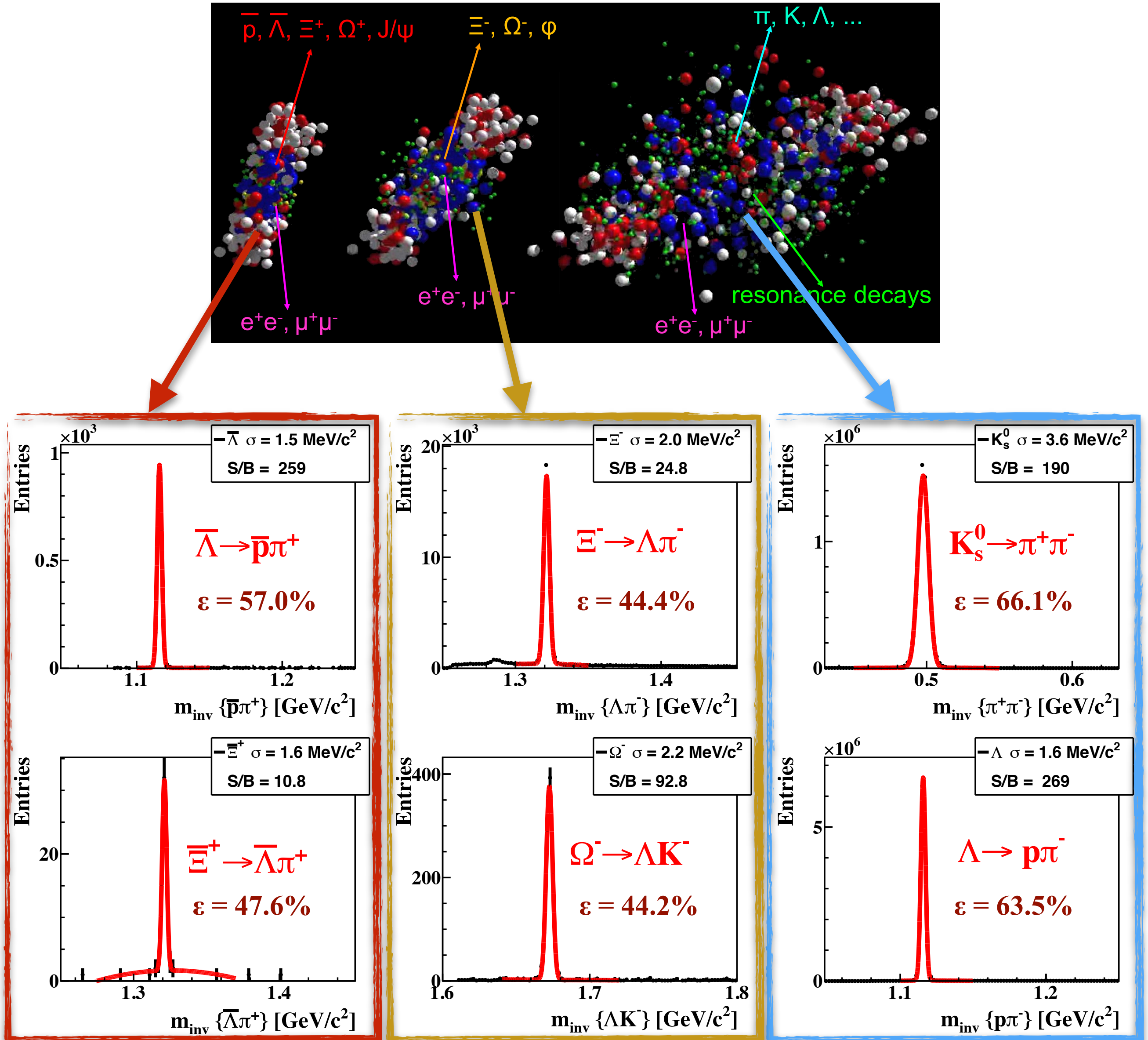
KF Particle Finder for Physics Analysis and Selection

The online and offline event reconstruction will be done by the same software on the same hardware with the same quality, otherwise the FLES online selection will be not possible.

This provides us an opportunity also to make the physics analysis on the full data set with all decay channels in real time online with the experiment.



Clean Probes of Collision Stages

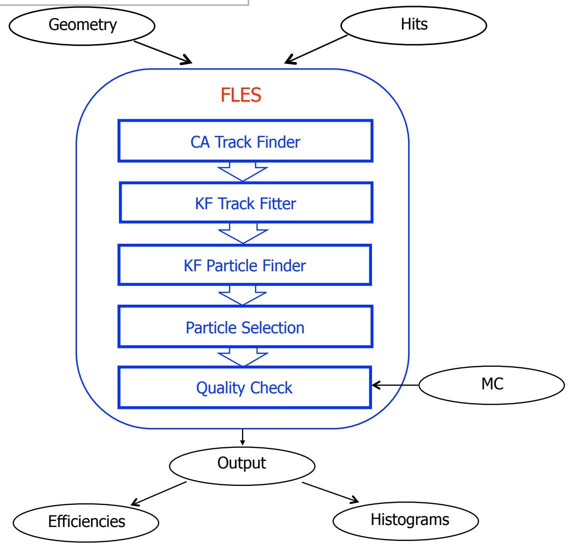


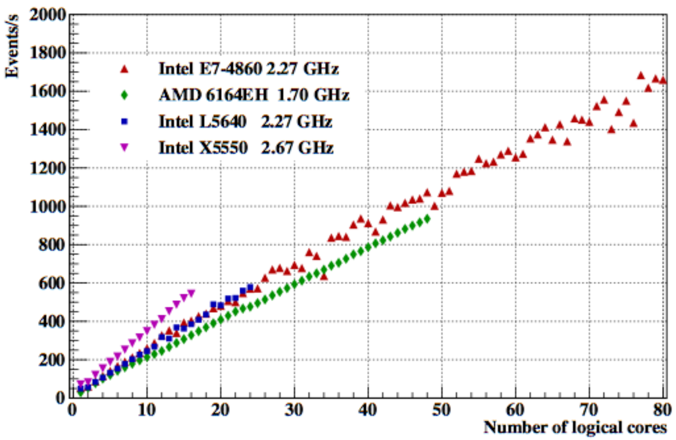
AuAu, 10 AGeV, 3.5M central UrQMD events, MC PID

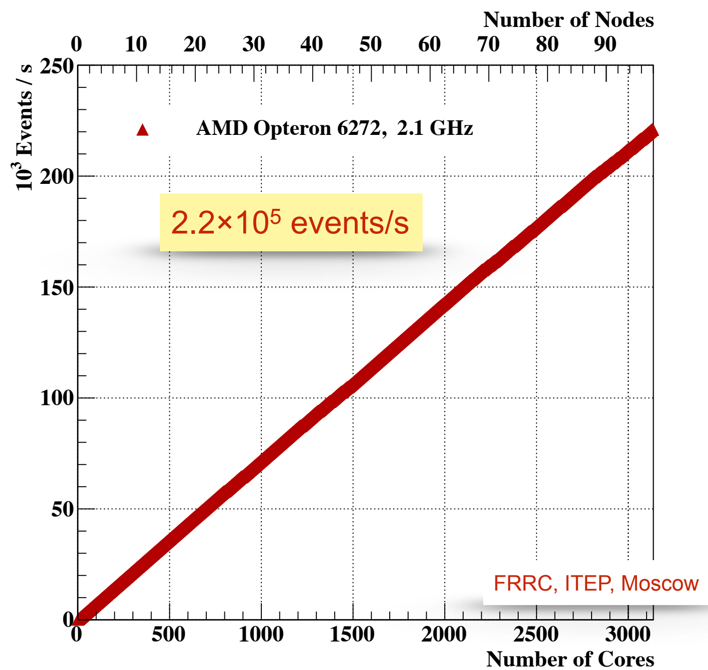
3

CBM Standalone First Level Event Selection (FLES) Package

with Prof. Dr. Volker Lindenstruth







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

Summary

We develop reconstruction algorithms:

- CA Track Finder
- KF Track Fitter
- KF Particle Finder

(a) in a simple and general form to be universal

(b) for on-line event reconstruction and physics analysis

(c) in CBM, STAR, ALICE and PANDA experiments

For efficient use of many-core architectures
we need to combine experience of physicists, computer scientists and CPU/GPU/Phi developers.

More details:

- V. Akishina, 4D event reconstruction in the CBM experiment, PhD Thesis, Uni-Frankfurt, 2017
- M. Zyzak, Online selection of short-lived particles on many-core computer architectures in the CBM experiment at FAIR, PhD Thesis, Uni-Frankfurt, 2016