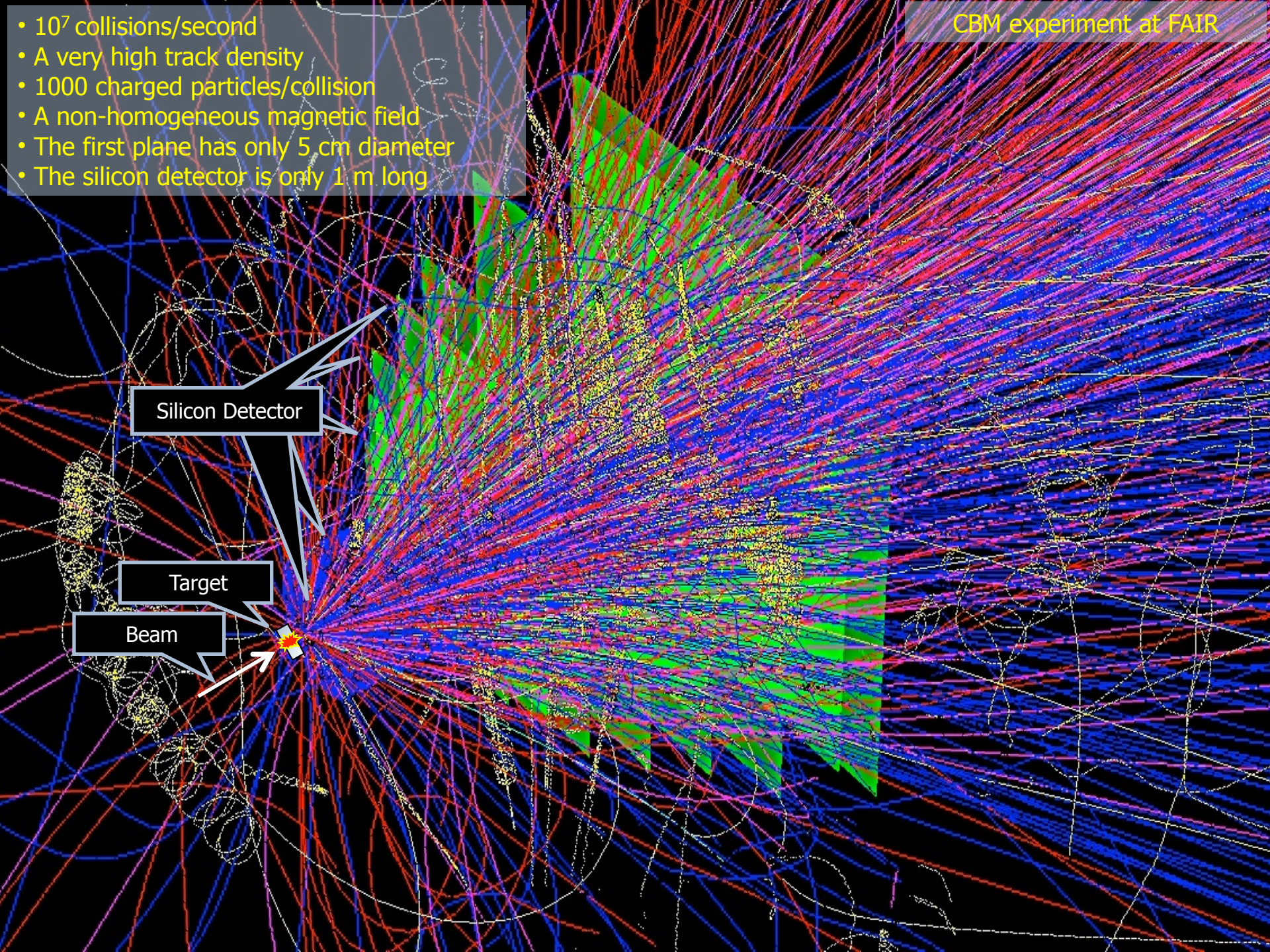
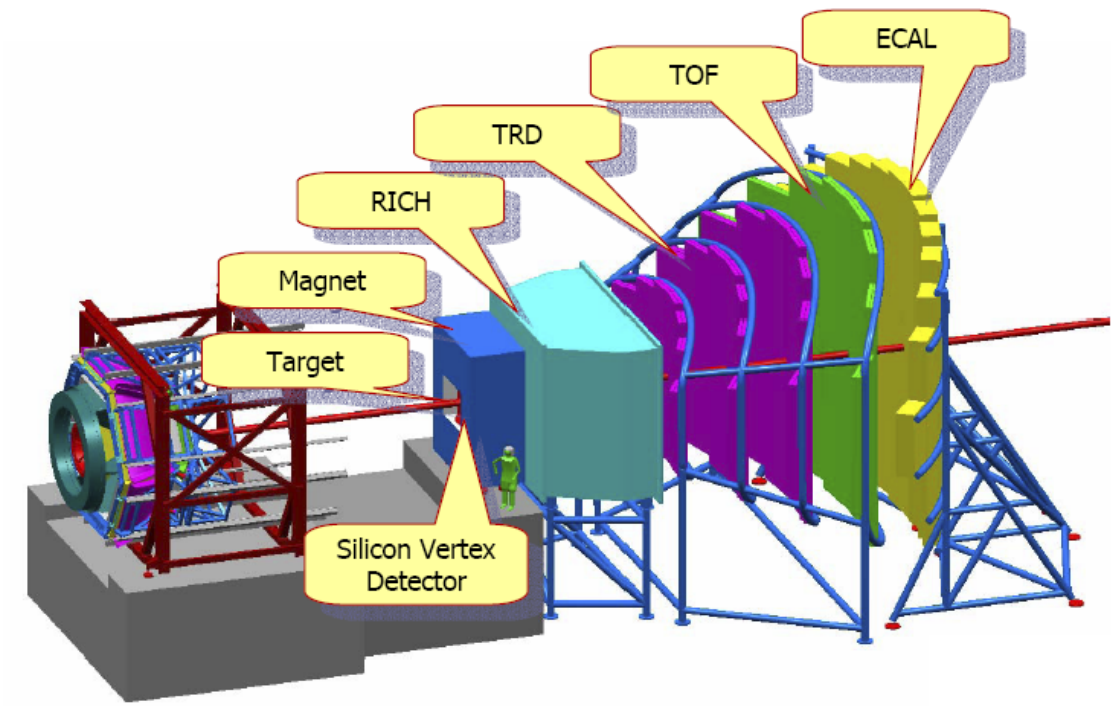
Scientific and High-Performance Computing at FAIR

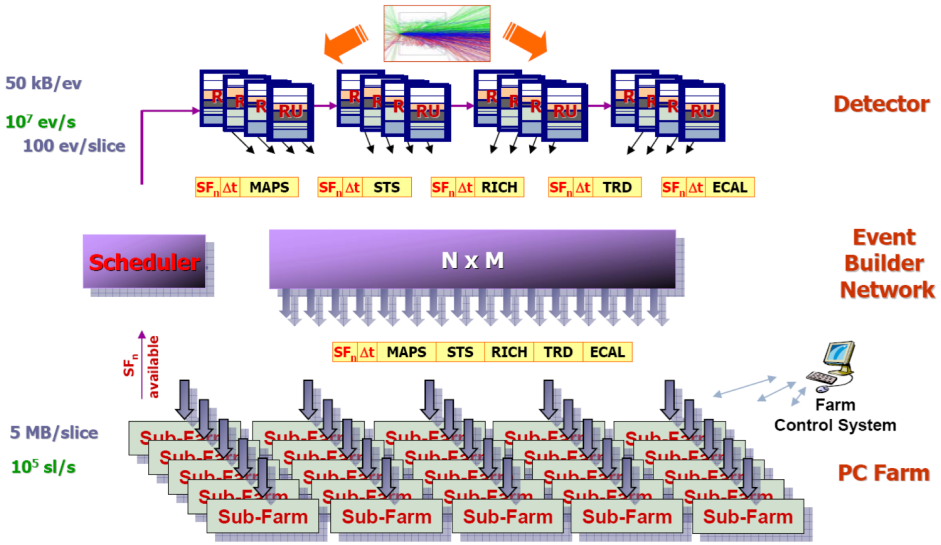
Ivan Kisel
Goethe University Frankfurt am Main
FIAS Frankfurt Institute for Advanced Studies


CBM experiment at FAIR
• 10^7 collisions/second
• A very high track density
• 1000 charged particles/collision
• A non-homogeneous magnetic field
• The first plane has only 5 cm diameter
• The silicon detector is only 1 m long
Silicon Detector
Target
Beam

Reconstruction Challenge in CBM



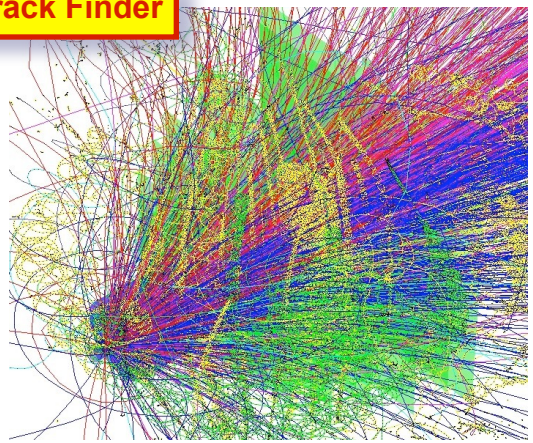
* Fixed-target heavy-ion experiment
* 10^7 Au+Au collisions/sec
* ~ 1000 charged particles/collision
* Non-homogeneous magnetic field
* Double-sided strip detectors (85% fake space-points)



First level event selection will be done in a processor farm with ≈ 60 000 CPU equivalent cores fed with data from the event (time slices) building network

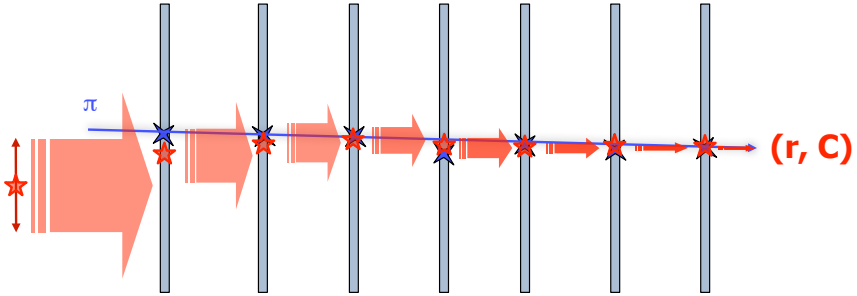
Stages of Event Reconstruction

1 // Track Finder



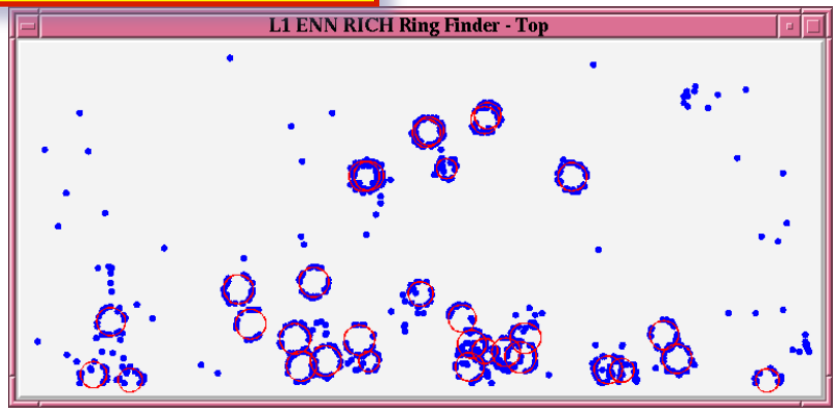
- Conformal Mapping
- Hough Transformation
- Track Following + Kalman Filter
- Cellular Automaton + Kalman Filter

2 // Track Fitter



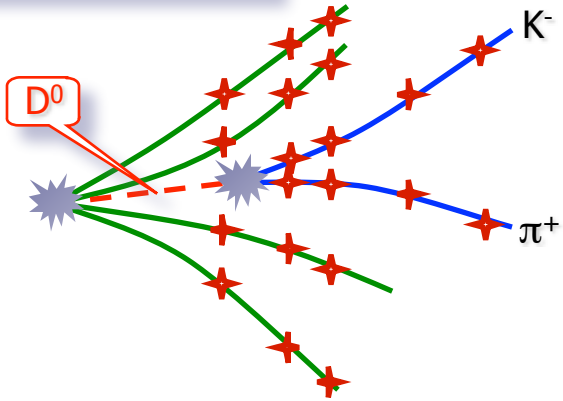
- Kalman Filter

3 // Ring Finder (Particle ID)



- Hough Transformation
- Elastic Neural Net

4 // Short-Lived Particle Finder



- Kalman Filter

Global Methods: Conformal Mapping + Histogramming

Global methods are especially suitable for fast tracking in projections

Example: Collider experiment with a solenoid, where tracks are circular trajectories

Conformal Mapping:

Transform circles into straight lines

$u = x/(x^2+y^2)$

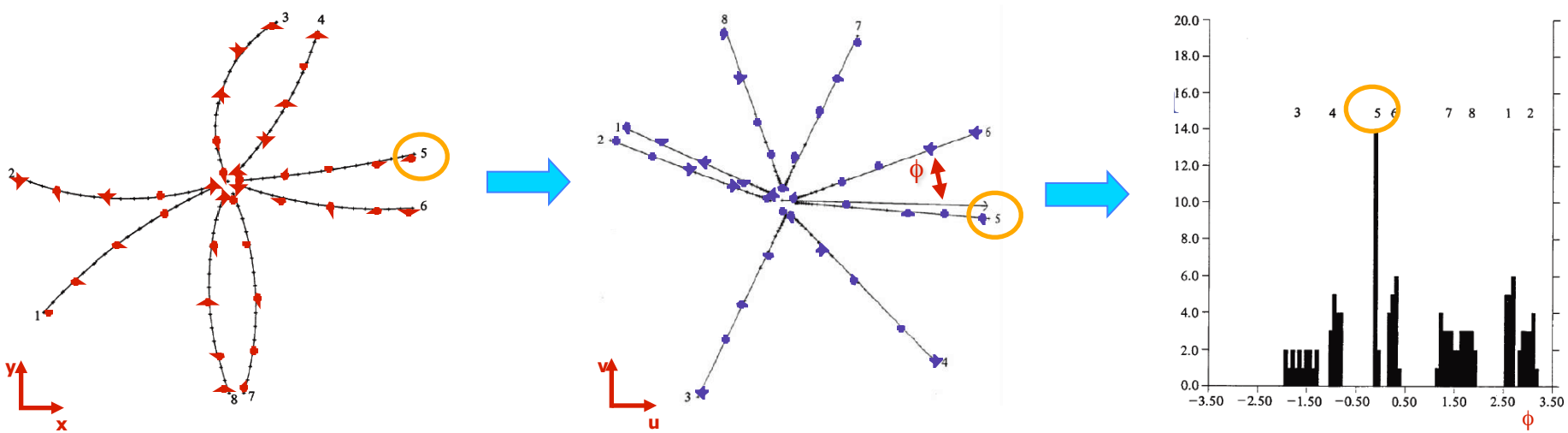
$v = -y/(x^2+y^2)$

Histogram:

Collect a histogram of azimuth angles ϕ

Find peaks in the histogram

Collect hits into tracks



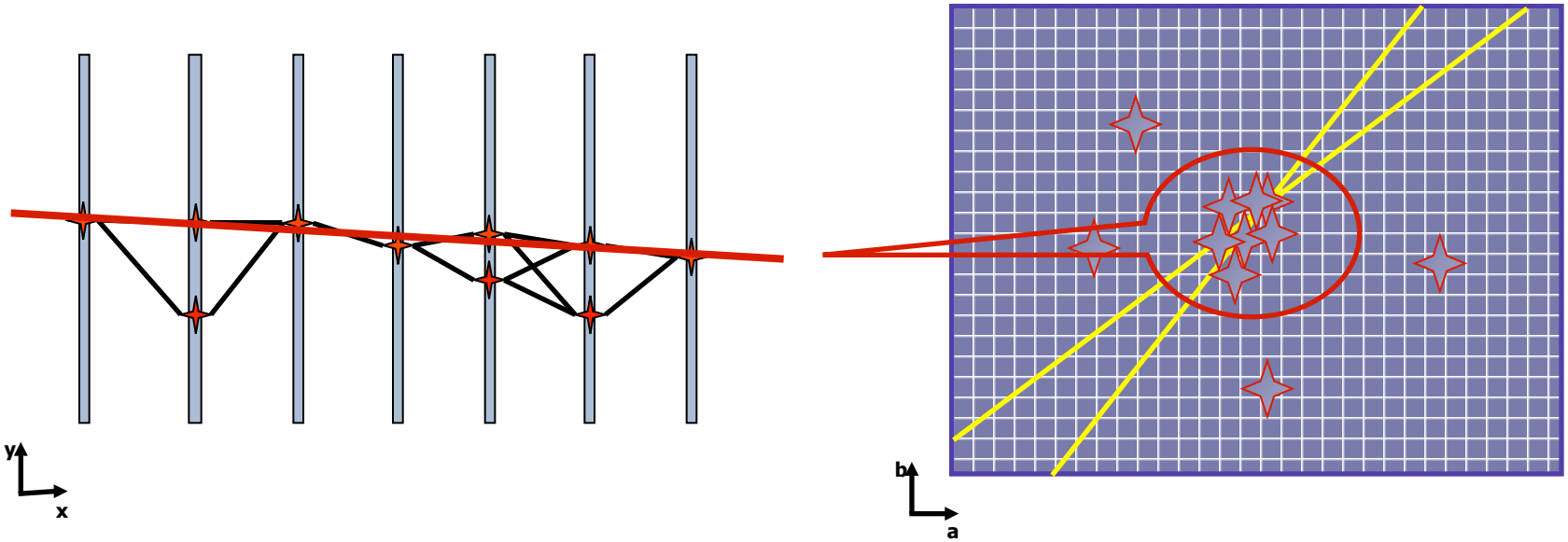
Global Methods: Hough Transformation

Measurement Space

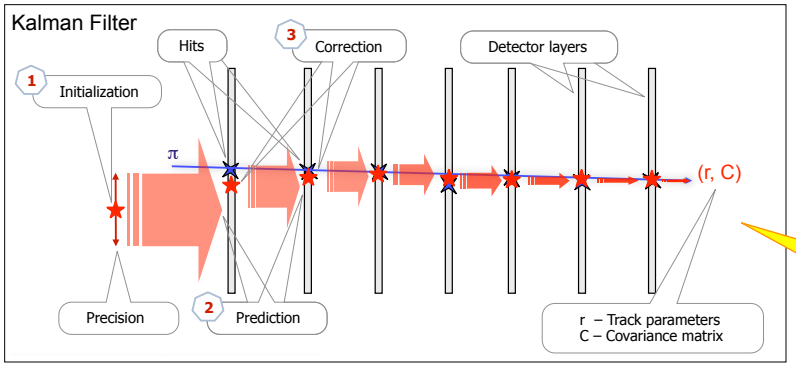
$$y = a*x + b$$

Parameter Space

$$b = -x*a + y$$

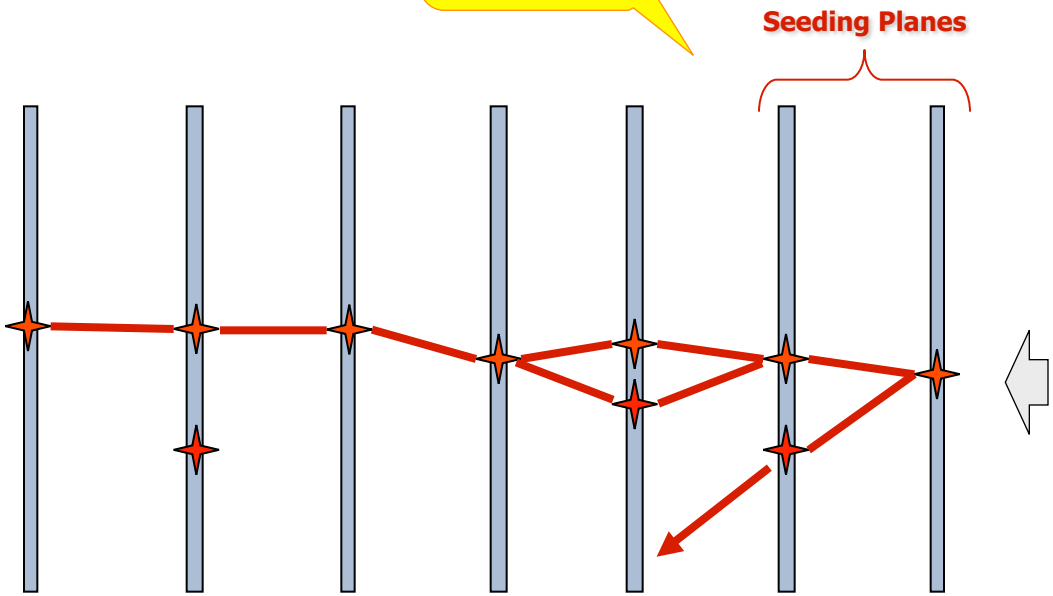


Local Methods: Kalman Filter for Track Following

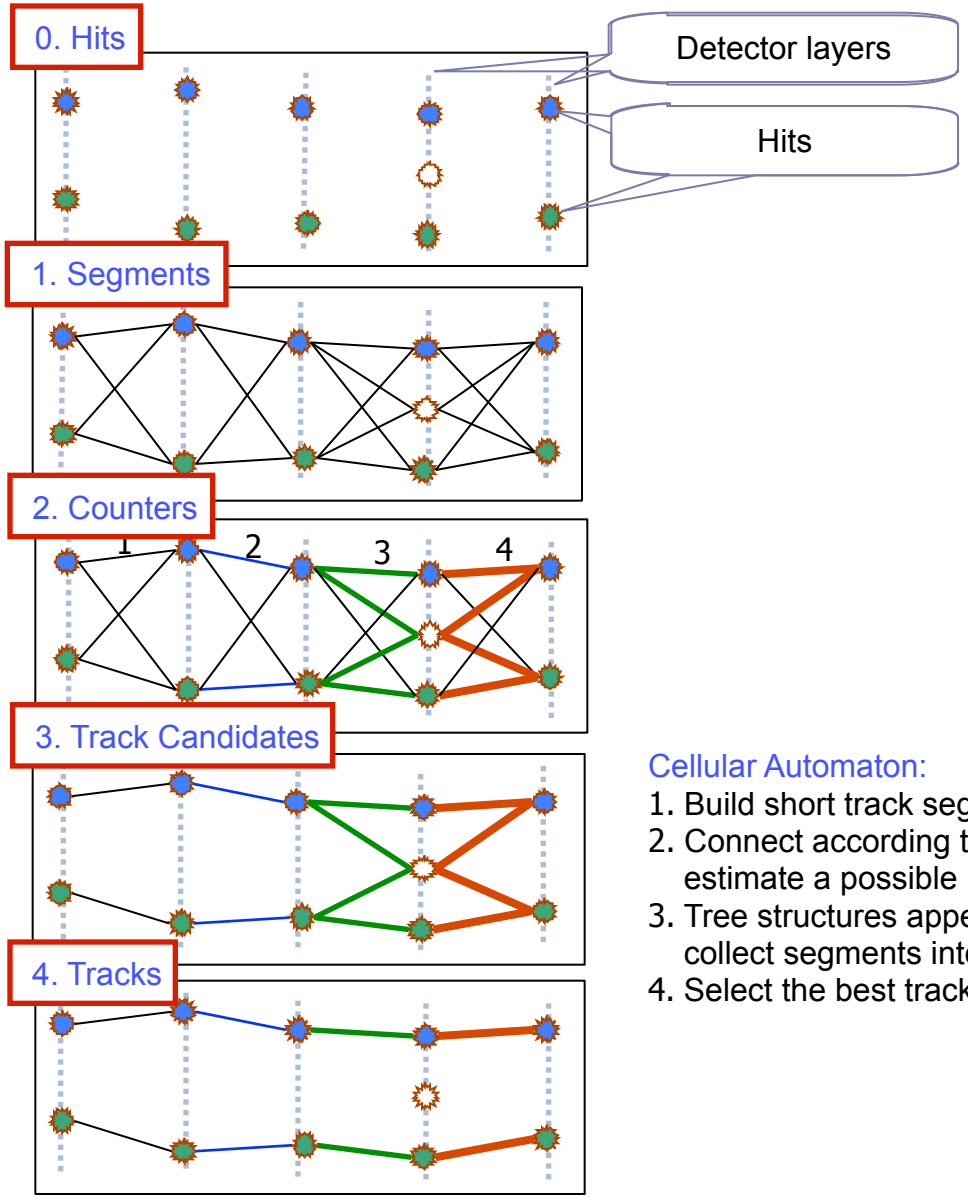


KF Track Fitter

KF Track Finder



Local Methods: Cellular Automaton as Track Finder



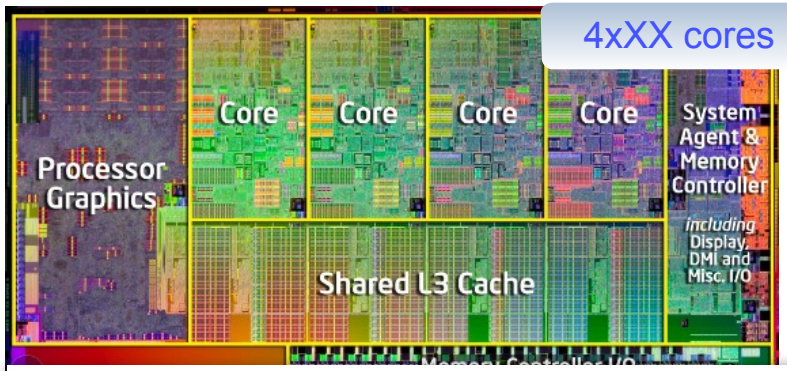
Cellular Automaton:

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.

Many-Core CPU/GPU Architectures

Intel/AMD CPU

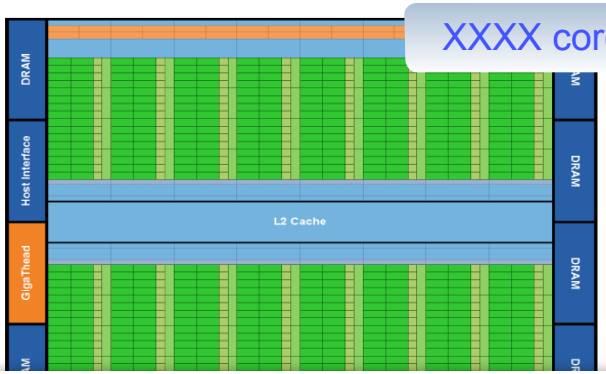
4xXX cores



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

ATI/NVIDIA GPU

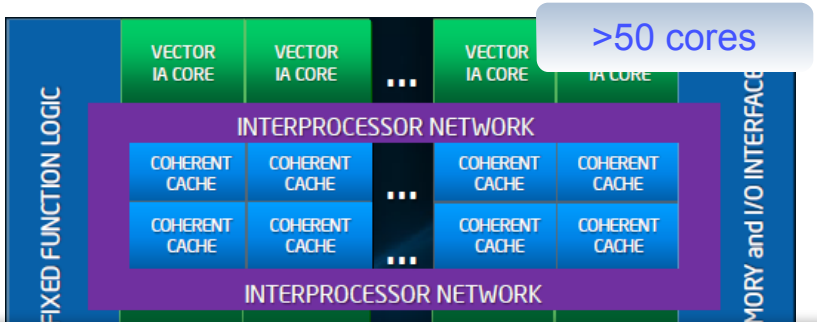
XXXX cores



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

Intel Xeon Phi

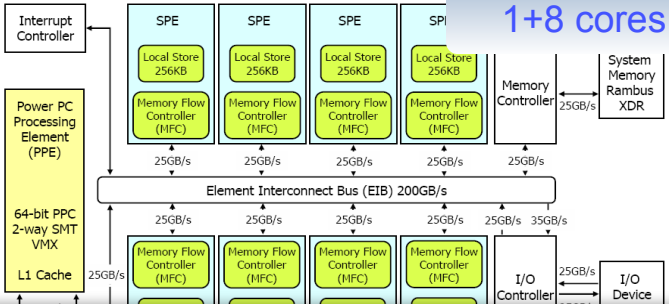
>50 cores



- 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



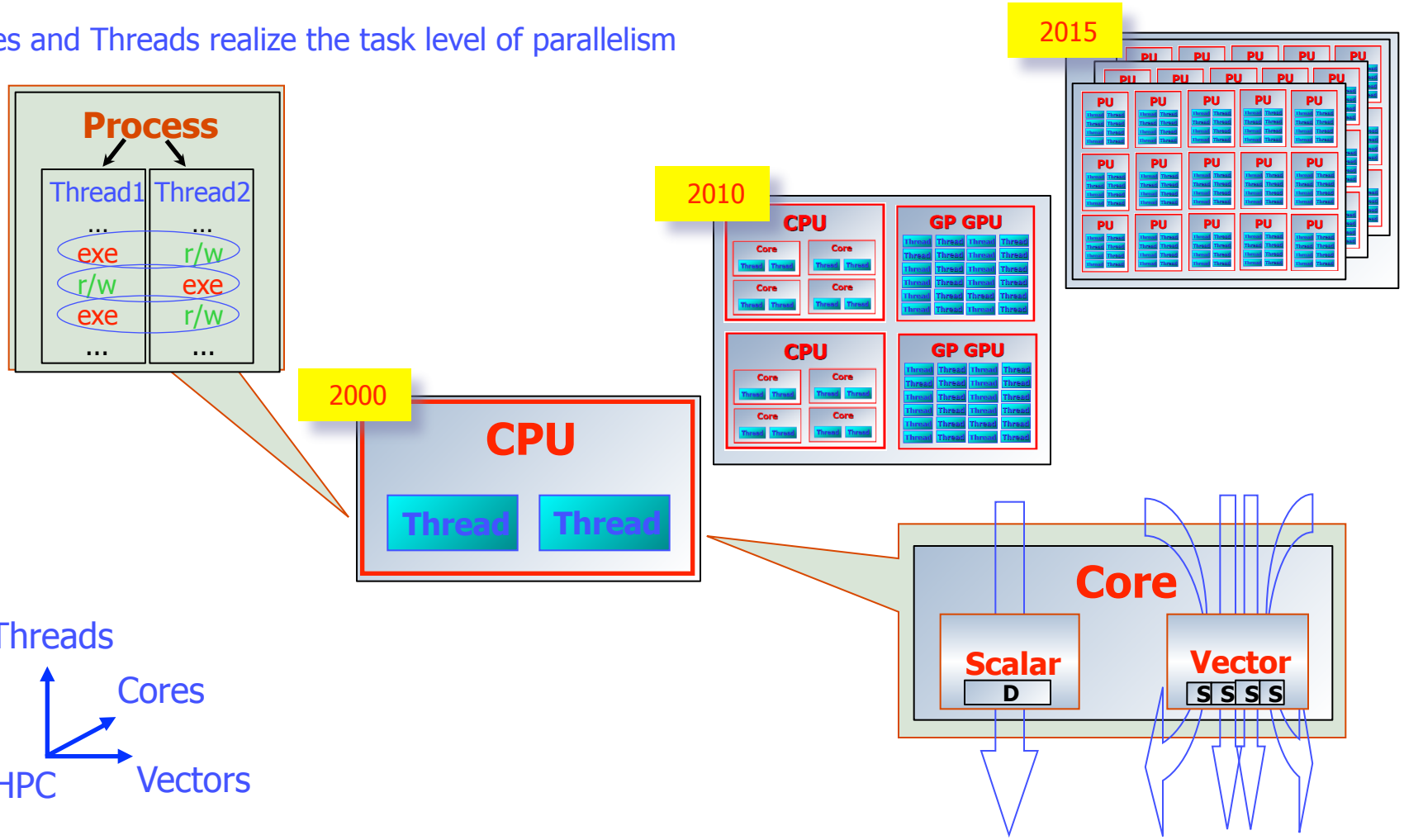
- General purpose RISC processor (PowerPC)
- 8 co-processors (SPE, Synergistic Processor Elements)
- 128-bit wide SIMD units

Future systems are heterogeneous

Many-core HPC: Cores, Threads and Vectors

HEP experiments work with high data rates, therefore need High Performance Computing (HPC) !

Cores and Threads realize the task level of parallelism



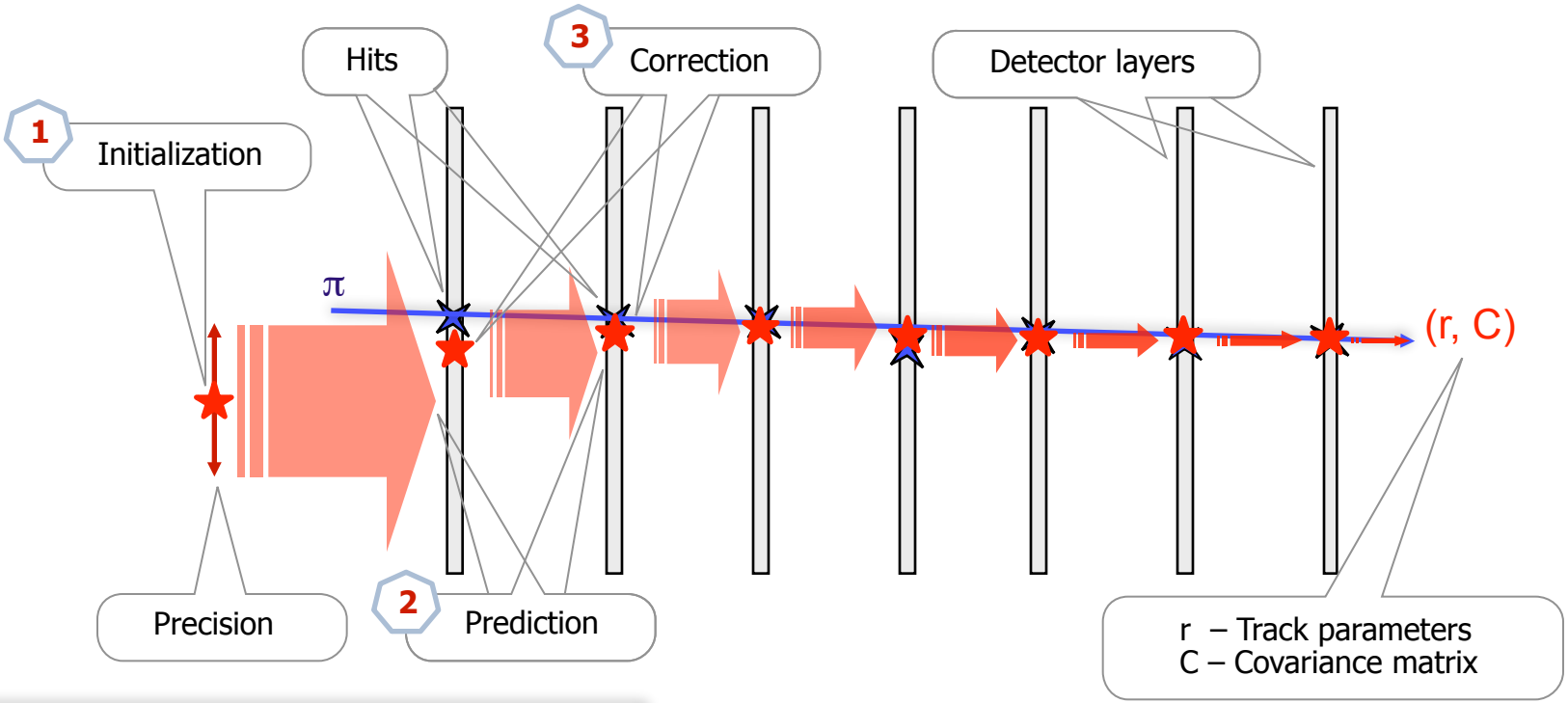
Vectors (SIMD) = data level of parallelism

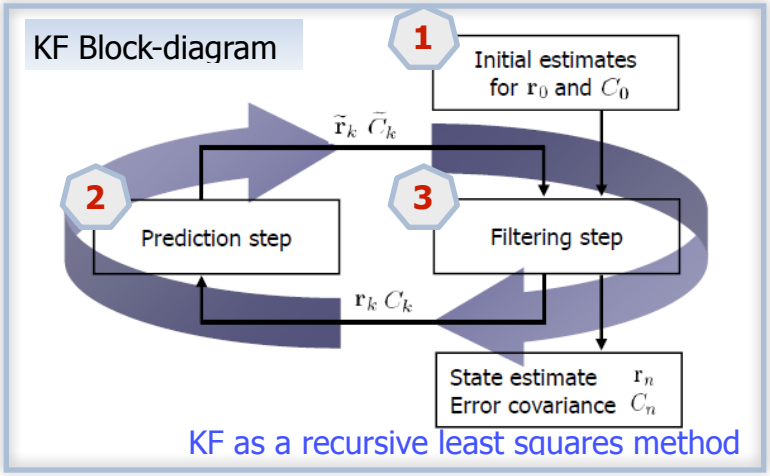
SIMD = Single Instruction, Multiple Data

Fundamental redesign of traditional approaches to data processing is necessary

Kalman Filter based Track Fit

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

CBM: Kalman Filter 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

Track Propagation:
- Runge-Kutta
- Analytic Formula

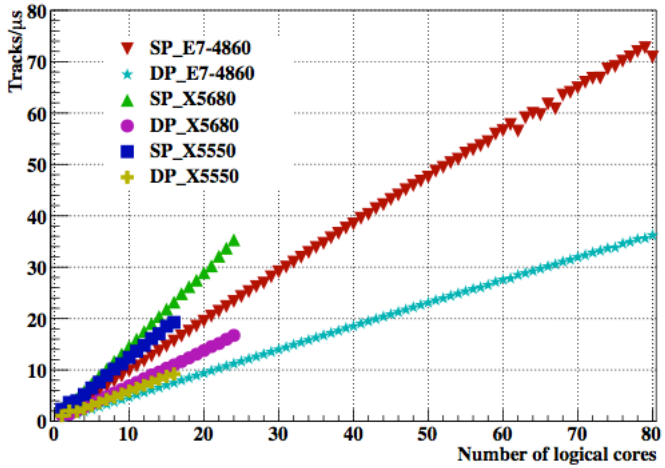
Implementations

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

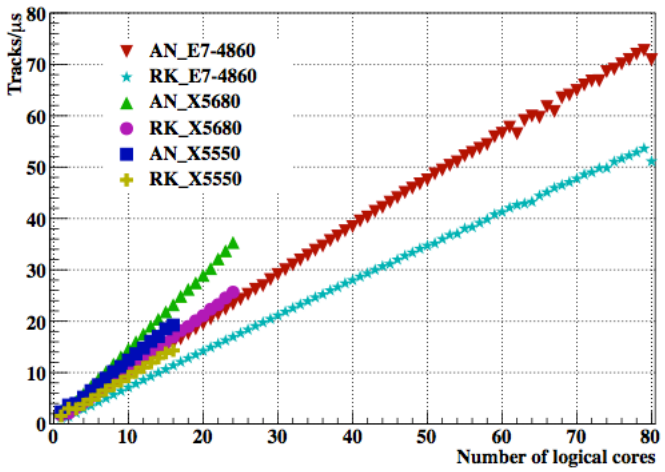
Parallelization (many-cores):
- Open MP
- ITBB
- ArBB
- OpenCL

Precision:
- single
- double

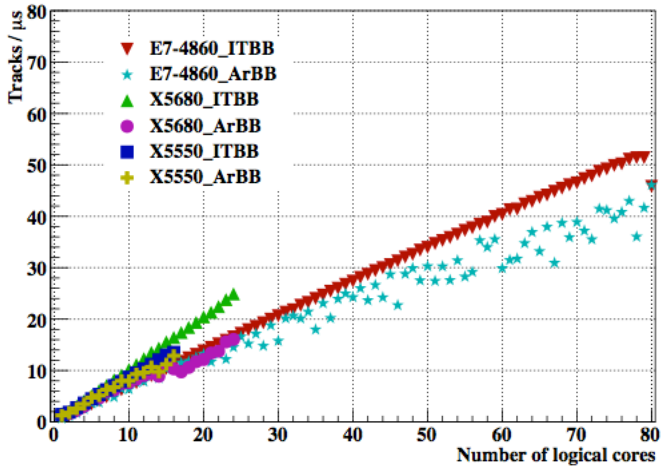
Conventional KF DP vs. SP



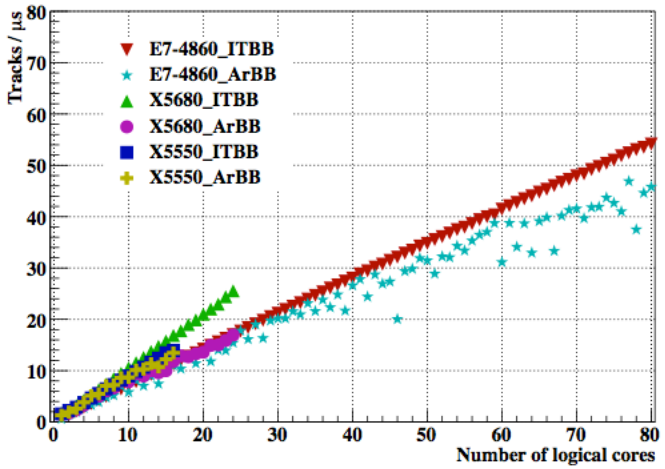
Conventional KF RK4 vs. Analytical



Square-Root KF

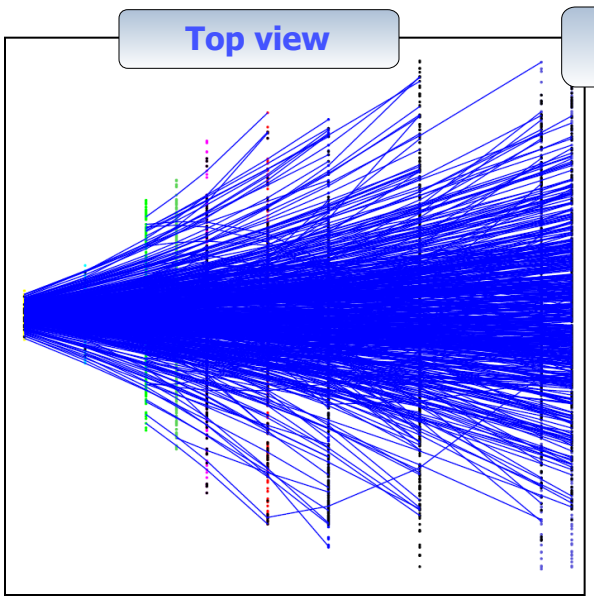


UD KF

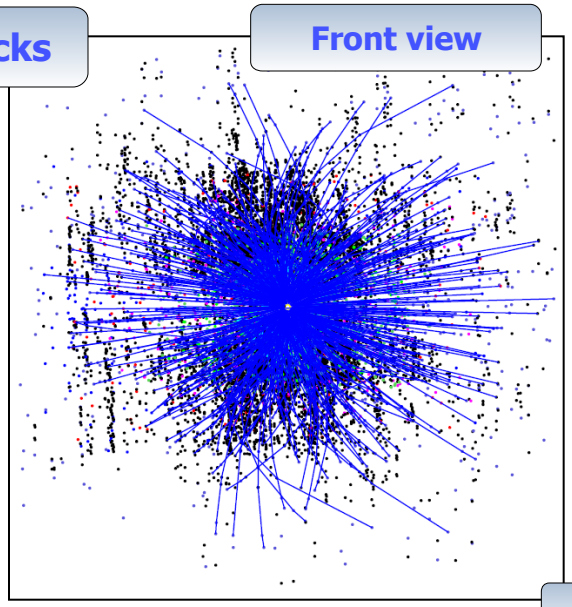


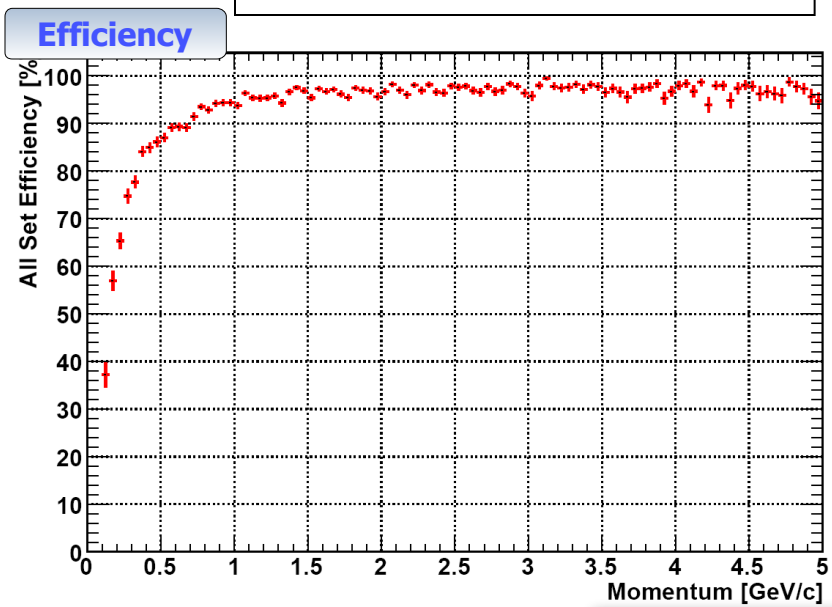
Strong many-core scalability of the Kalman filter library

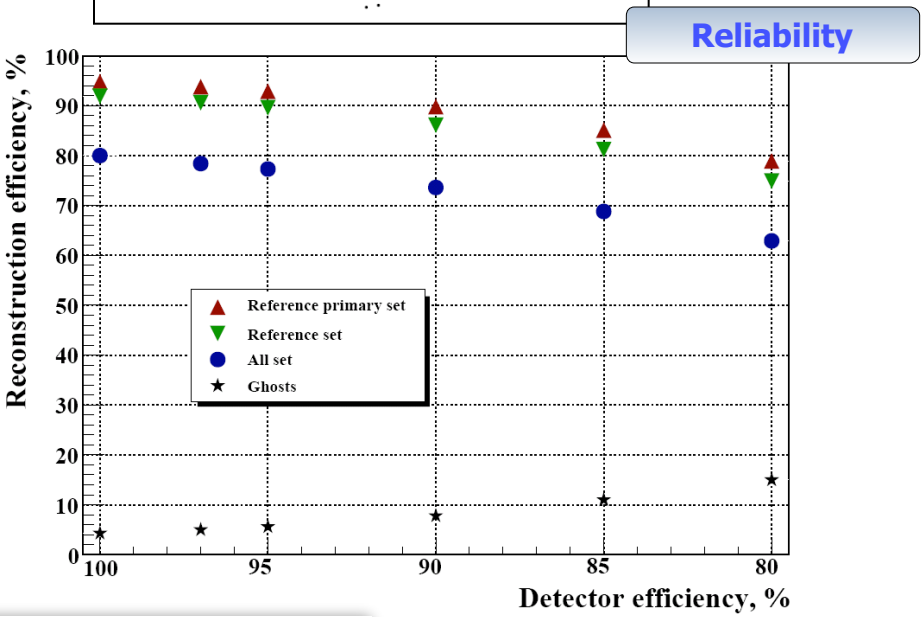
CBM: Cellular Automaton Track Finder



770 Tracks



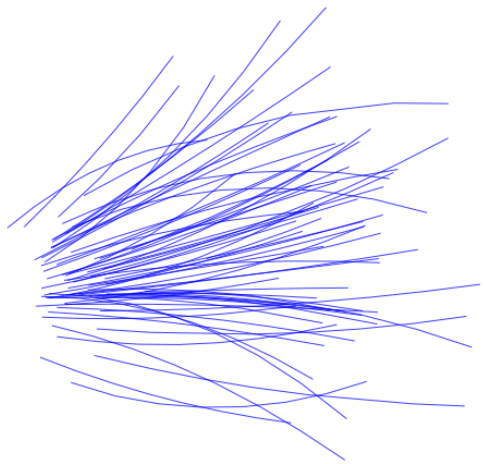




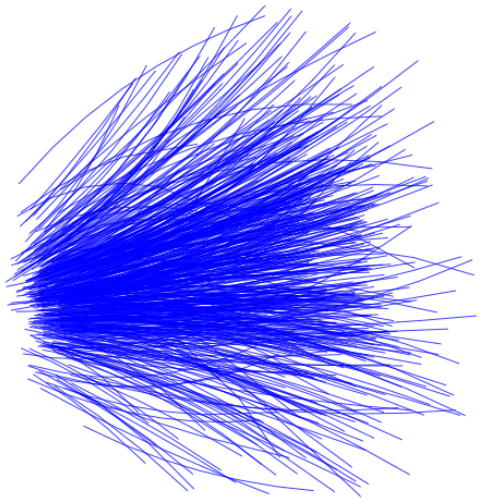
Efficient and reliable event reconstruction

CBM: CA Track Finder at High Track Multiplicity

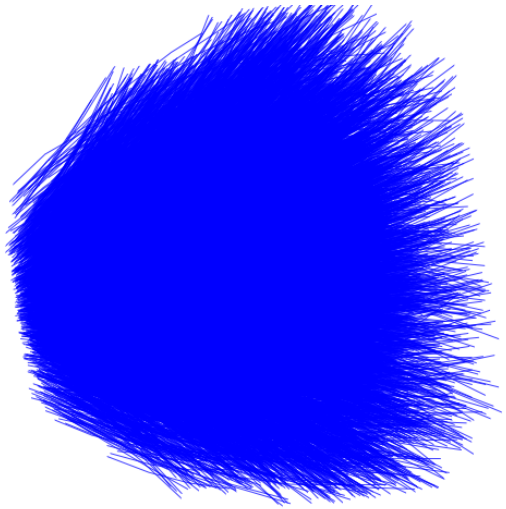
A number of minimum bias events is gathered into a group, which is then treated by the track finder as one event



1 minimum bias event
$\langle N_{reco} \rangle = 109$



5 minimum bias events
$\langle N_{reco} \rangle = 572$

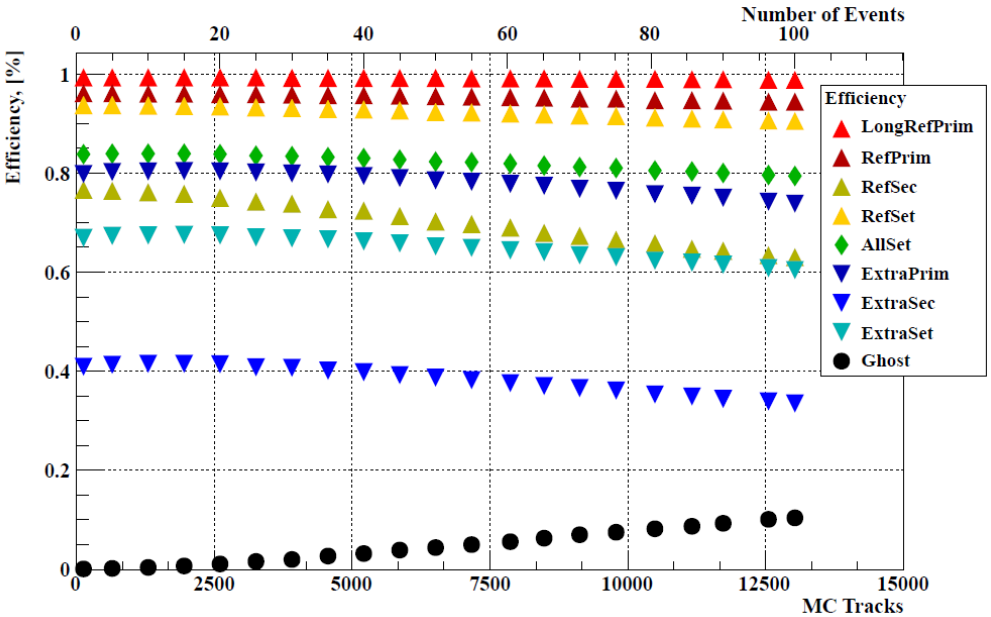


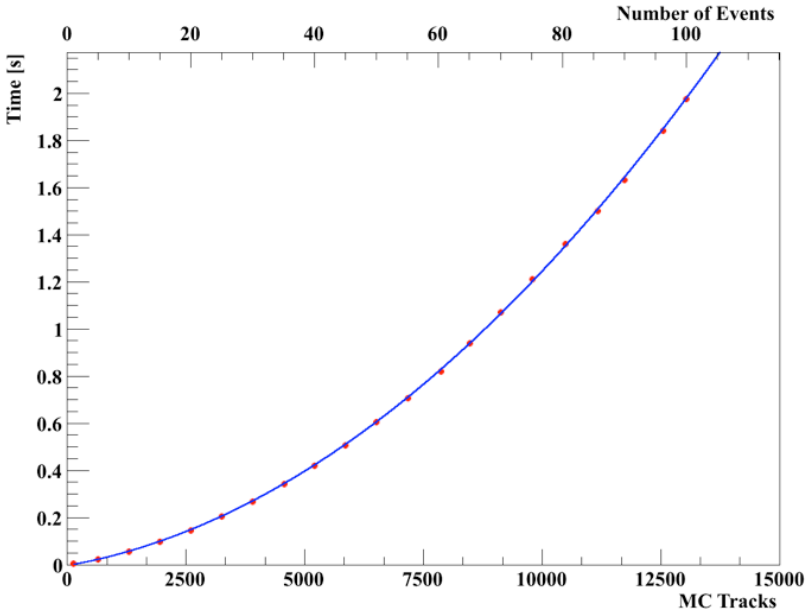
100 minimum bias events
$\langle N_{reco} \rangle = 10340$

Au+Au mbias events at 25 AGeV, 8 STS, 0 x 7,5 strip angles

Towards 4D (space+time) event reconstruction

CBM: CA Track Finder Efficiency and Time vs. Track Multiplicity

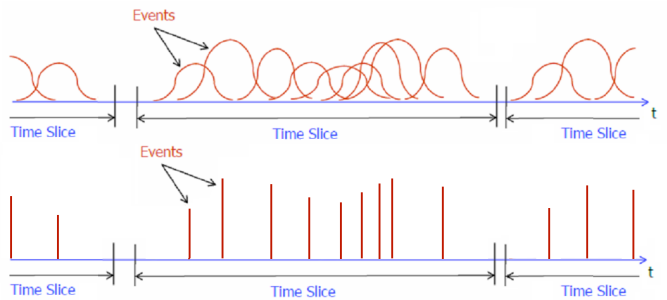


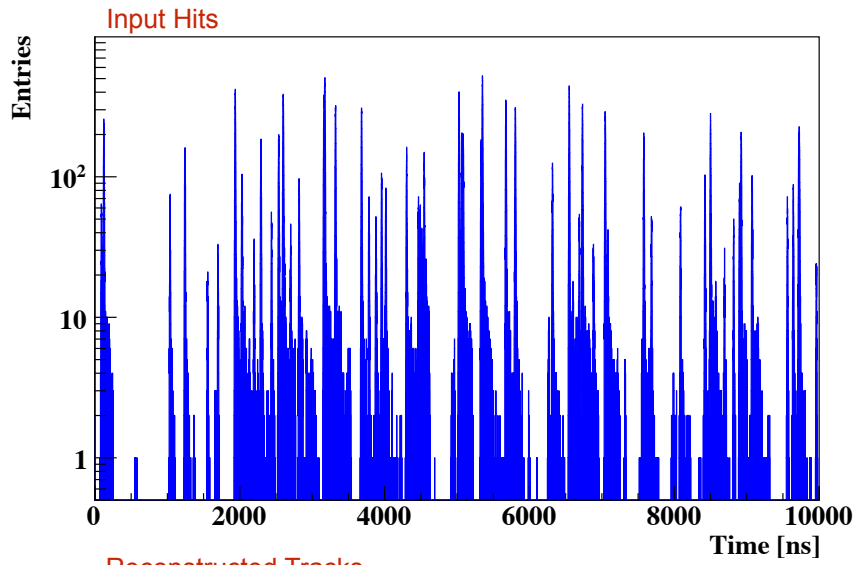


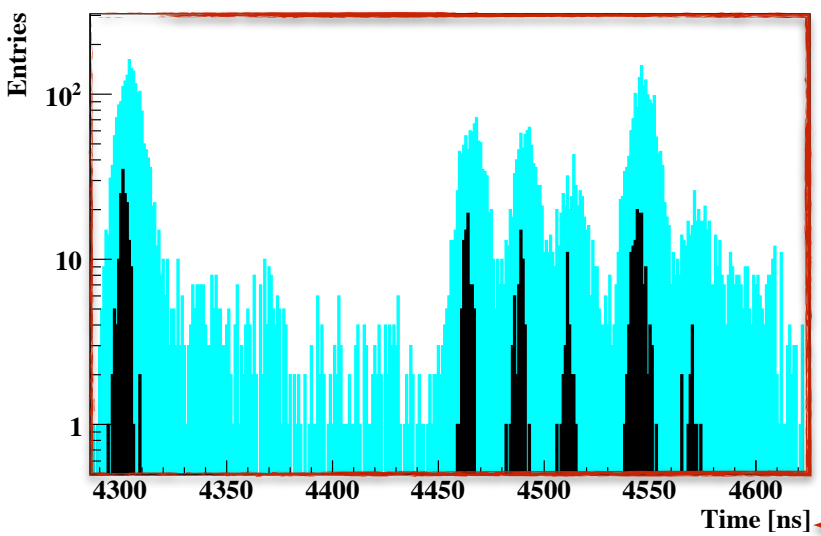
Stable reconstruction efficiency and time as a second order polynomial up to 100 minimum bias events in a group

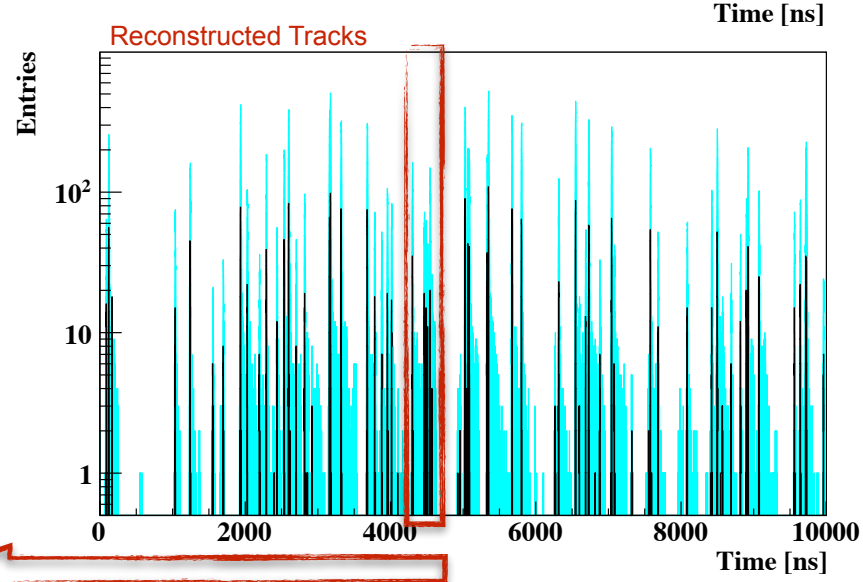
CBM: 4D Event Building with CA Track Finder

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









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

CBM: Particle Finder for Physics Analysis and Selection

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

Tracks: $e^{\pm}$, $\mu^{\pm}$, $\pi^{\pm}$, $K^{\pm}$, $p^{\pm}$
secondary and primary

Open-charm:

$D^0 \rightarrow \pi^+ K^-$
$D^0 \rightarrow \pi^+ \pi^+ \pi^- K^-$
$\bar{D}^0 \rightarrow \pi^- K^+$
$\bar{D}^0 \rightarrow \pi^- \pi^- \pi^+ K^+$
$D^+ \rightarrow \pi^+ \pi^+ K^-$
$D^- \rightarrow \pi^- \pi^- K^+$
$D_s^+ \rightarrow \pi^+ K^+ K^-$
$D_s^- \rightarrow \pi^- K^+ K^-$
$\Lambda_c \rightarrow \pi^+ K^- p$

Strange particles:

$K^0_s \rightarrow \pi^+ \pi^-$
$\Lambda \rightarrow p\, \pi^-$
$\bar{\Lambda} \rightarrow \pi^+ p^-$

Gamma

$\gamma \rightarrow e^- e^+$

Strange resonances

$K^{*0} \rightarrow K^+ \pi^-$
$\bar{K}^{*0} \rightarrow \pi^+ K^-$
$\Lambda^* \rightarrow p\, K^-$
$\bar{\Lambda}^* \rightarrow p^- K^+$

Light vector mesons:

$\rho \rightarrow e^- e^+$
$\rho \rightarrow \mu^- \mu^+$
$\omega \rightarrow e^- e^+$
$\omega \rightarrow \mu^- \mu^+$
$\varphi \rightarrow e^- e^+$
$\varphi \rightarrow \mu^- \mu^+$
$\varphi \rightarrow K^- K^+$

Charmonium:

$J/\Psi \rightarrow e^- e^+$
$J/\Psi \rightarrow \mu^- \mu^+$

Multi-strange hyperons:

$\Xi^- \rightarrow \Lambda\, \pi^-$
$\Xi^+ \rightarrow \bar{\Lambda}\, \pi^+$
$\Omega^- \rightarrow \Lambda\, K^-$
$\Omega^+ \rightarrow \bar{\Lambda}\, K^+$

Strange and multi-strange resonances:

$\Sigma^{*+} \rightarrow \Lambda\, \pi^+$
$\bar{\Sigma}^{*+} \rightarrow \bar{\Lambda}\, \pi^-$
$\Sigma^{*-} \rightarrow \Lambda\, \pi^-$
$\bar{\Sigma}^{*-} \rightarrow \bar{\Lambda}\, \pi^+$
$K^{*-} \rightarrow K^0_s\, \pi^-$
$K^{*+} \rightarrow K^0_s\, \pi^+$
$\Xi^{*-} \rightarrow \Lambda\, K^-$
$\Xi^{*+} \rightarrow \bar{\Lambda}\, K^+$

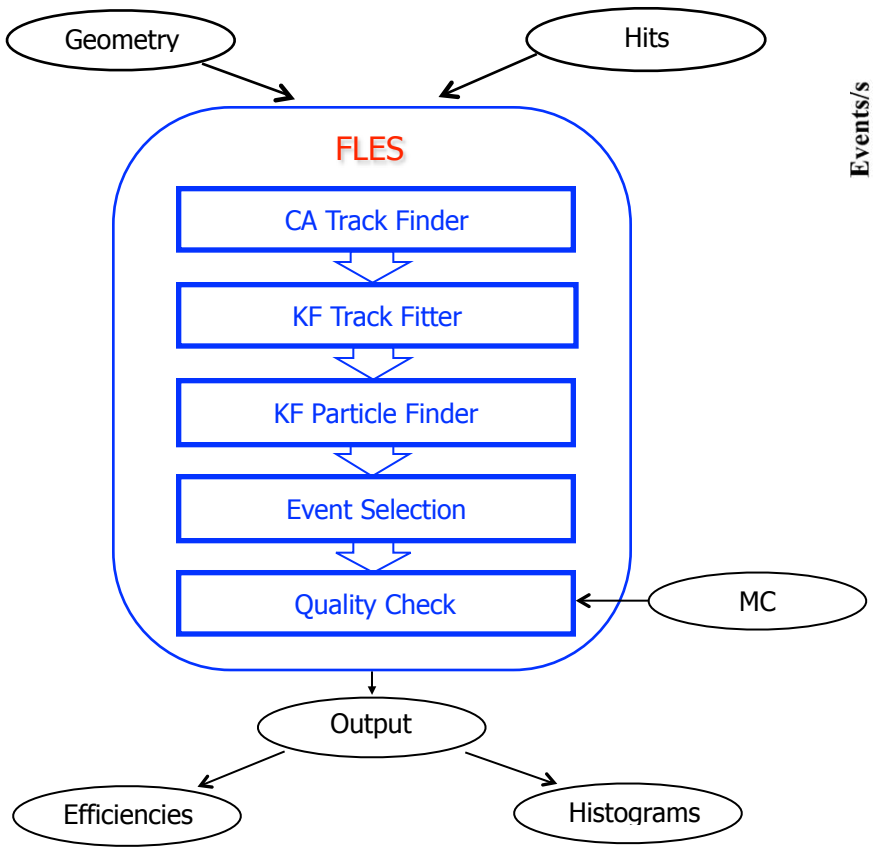
Open-charm resonances:

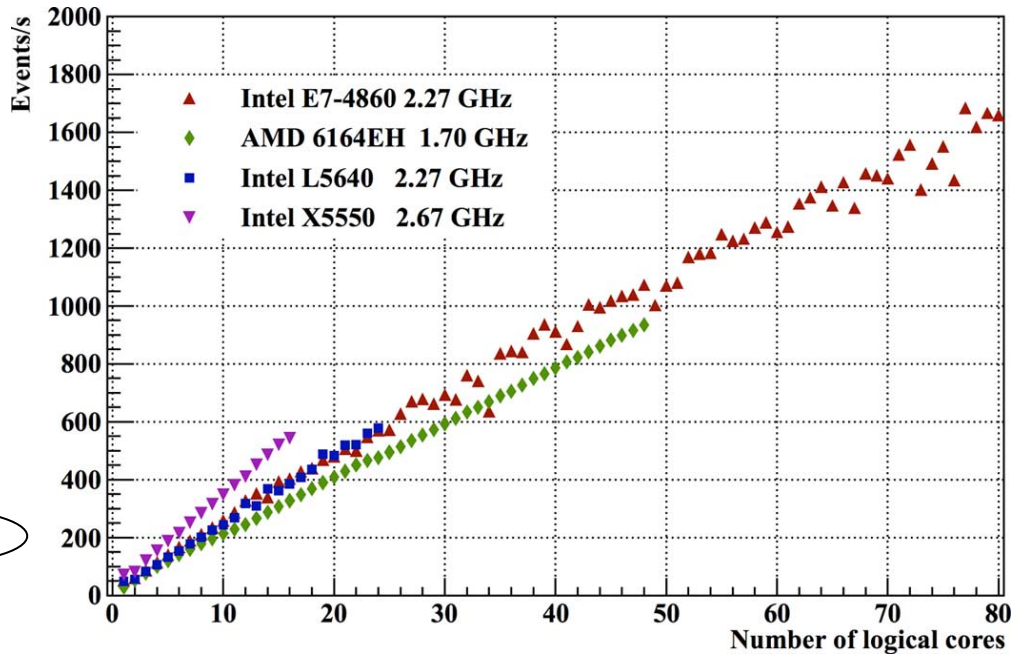
$D^{*0} \rightarrow D^+ \pi^-$
$\bar{D}^{*0} \rightarrow D^- \pi^+$
$D^{*+} \rightarrow D^0 \pi^+$
$D^{*-} \rightarrow \bar{D}^0 \pi^-$

Multi-strange resonances:

$\Xi^{*0} \rightarrow \Xi^- \pi^+$
$\bar{\Xi}^{*0} \rightarrow \Xi^+ \pi^-$
$\Omega^{*-} \rightarrow \Xi^- \pi^+ K^-$
$\Omega^{*+} \rightarrow \Xi^+ \pi^- K^+$

CBM Standalone First Level Event Selection (FLES) Package

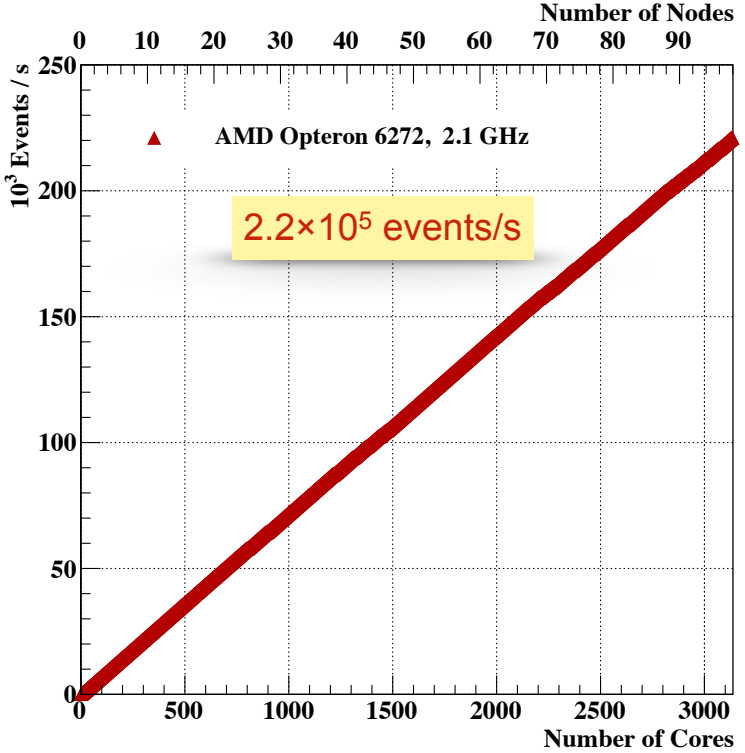




Given n threads each filled with 1000 events, run them on specified n cores, thread/core.

The first version of the FLES package is vectorized, parallelized, portable and scalable

Towards CBM FLES Demonstrator

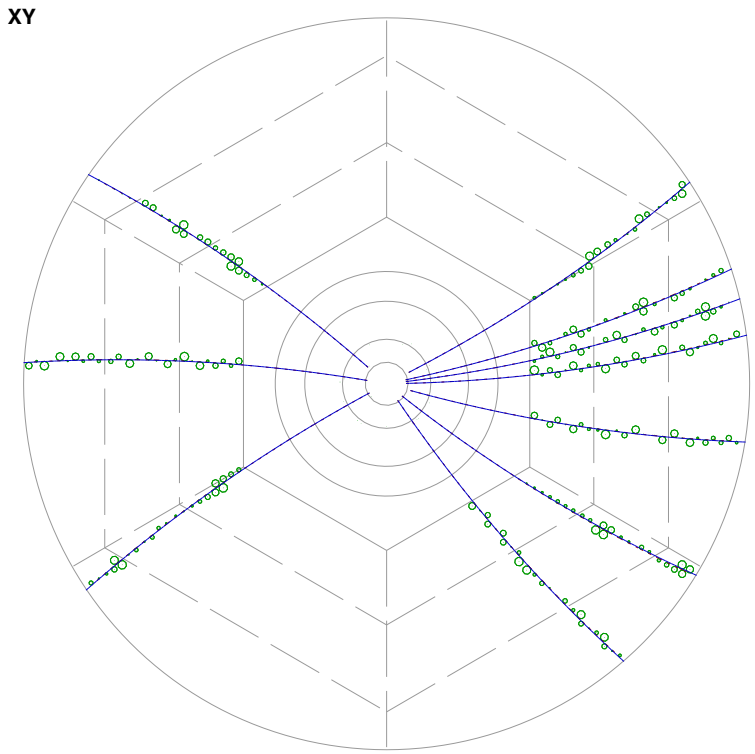


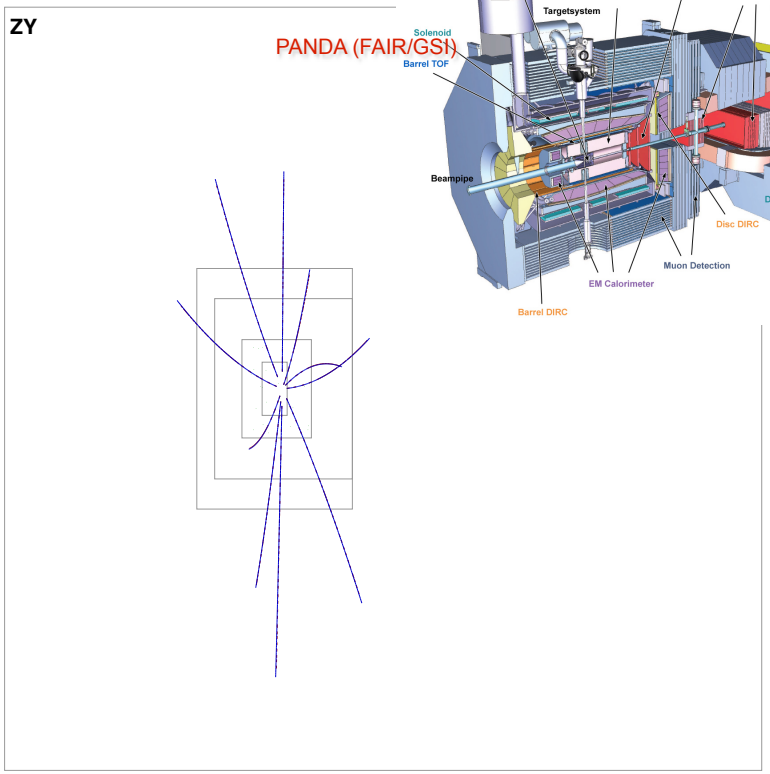
- LOEWE CSC (FIAS, Frankfurt)
- Green Cube (GSI, Darmstadt)
- FAIR-Russia HPC Cluster (ITEP, Moscow)

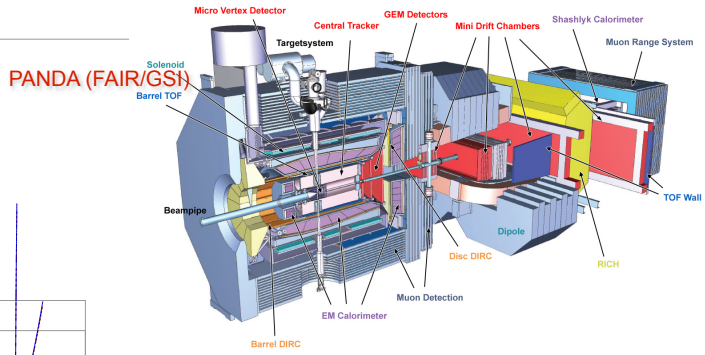
From cores to the CBM FLES farm with 60 000 cores

PANDA (FAIR): CA Track Finder in STT and MVD

10 primary tracks with pt = 1 GeV/c



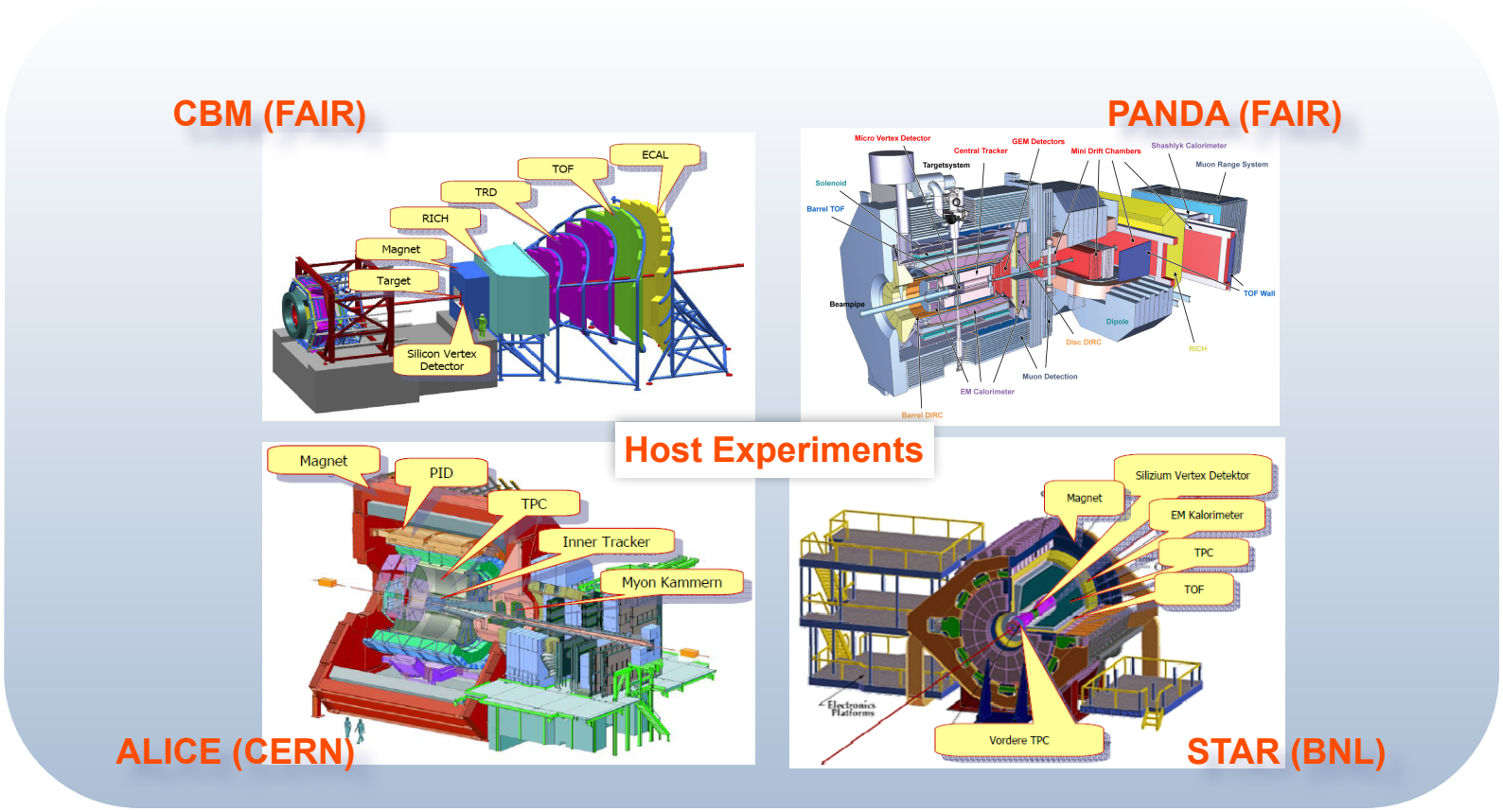




	STT	STT+MVD
Efficiency	97.2	99.3
Clone	1.8	9.2
Ghost	2.5	2.5
Tracks/event	10	10
Time, ms/event	5	7

Combine experience of CBM and PANDA

Consolidate Efforts: Common Reconstruction Package



- Create a common FAIR reconstruction package
- Use also in running experiments (ALICE, STAR, …)
- Optimize for many-core CPU/GPU/Phi architectures
- Hide parallelism and keep the traditional form of physics analysis

Summary

With increasing complexity of online data analysis

HPC computing becomes a detector sub-system in HEP experiments