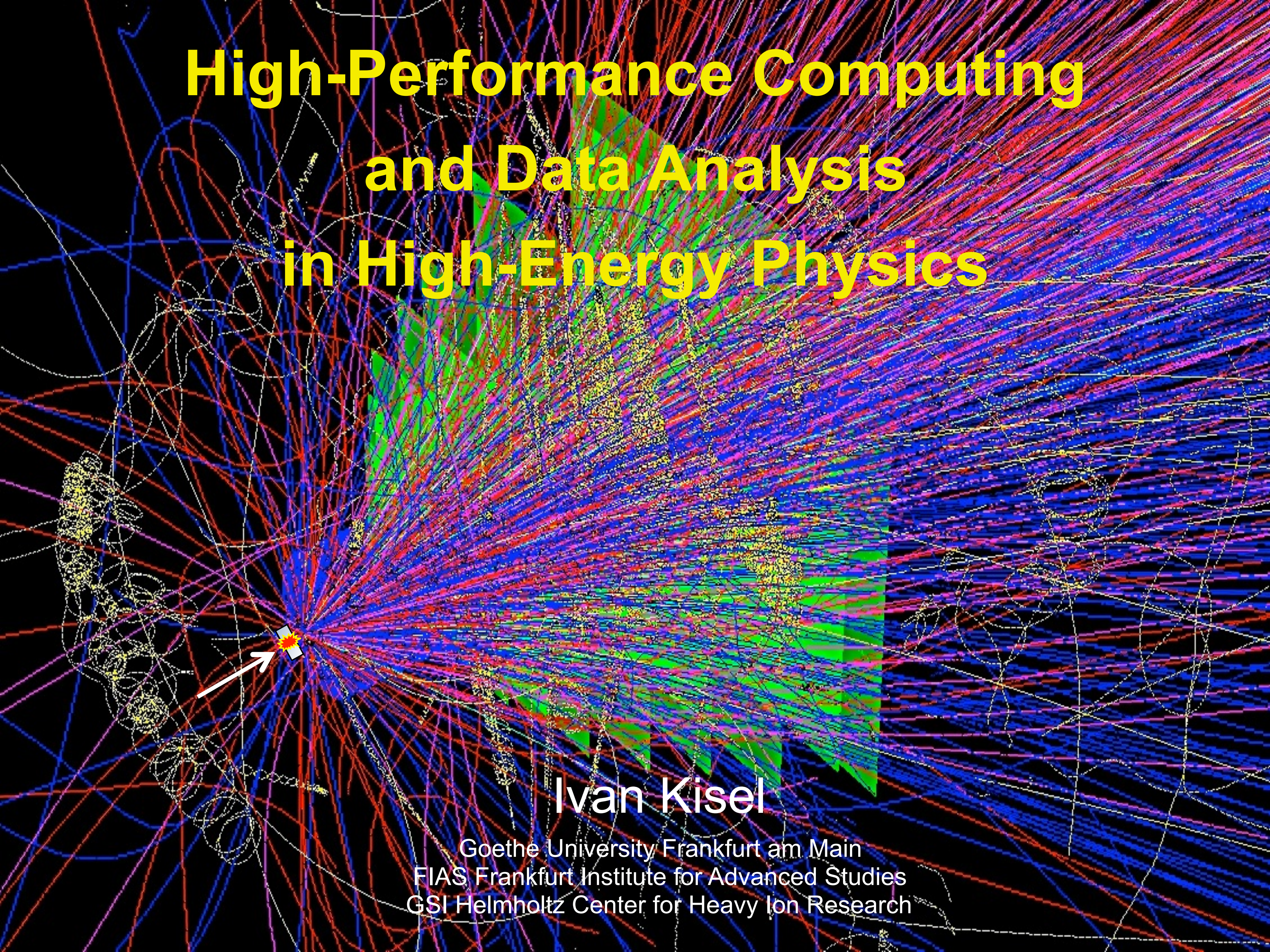


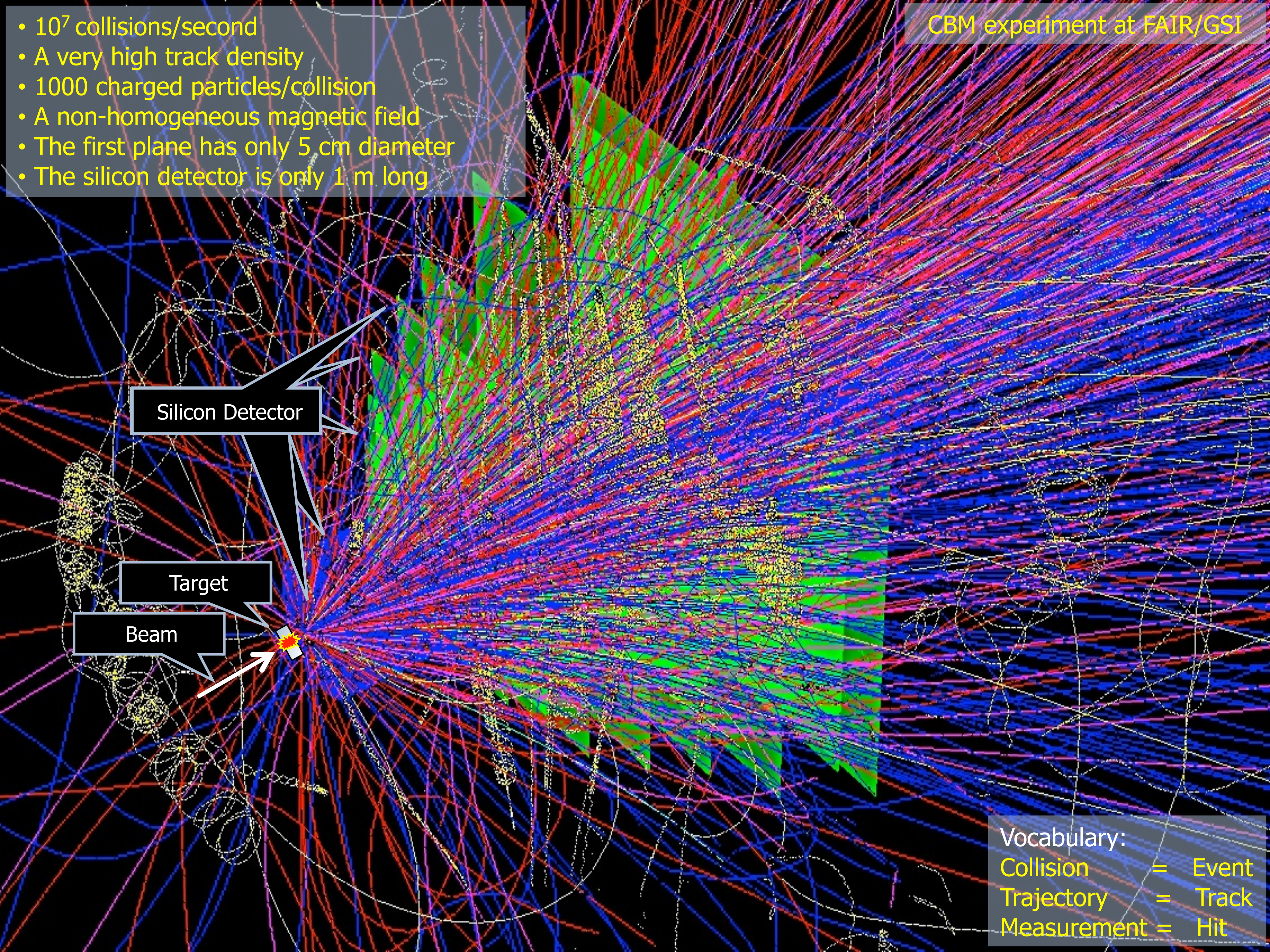
High-Performance Computing and Data Analysis in High-Energy Physics

Ivan Kisel

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

CBM experiment at FAIR/GSI

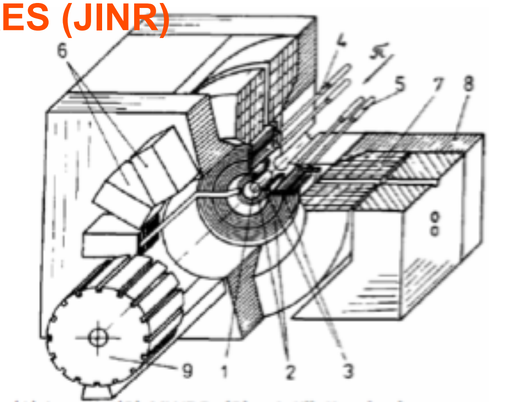
- 10^7 collisions/second
- A very high track density
- 1000 charged particles/collision
- A non-homogeneous magnetic field
- The first plane has only 5 cm diameter
- The silicon detector is only 1 m long



Vocabulary:
Collision = Event
Trajectory = Track
Measurement = Hit

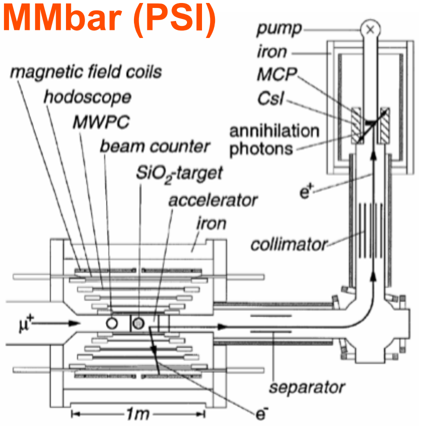
My Participation in HEP Experiments

ARES (JINR)

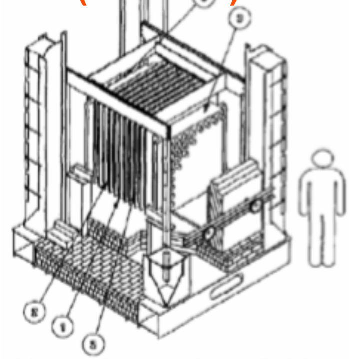


(1) target, (2) MWPC, (3) scintillation hodoscopes,
(4) lightguides, (5) photomultipliers, (6) electronics, (7)-(9) magnet.

MMbar (PSI)

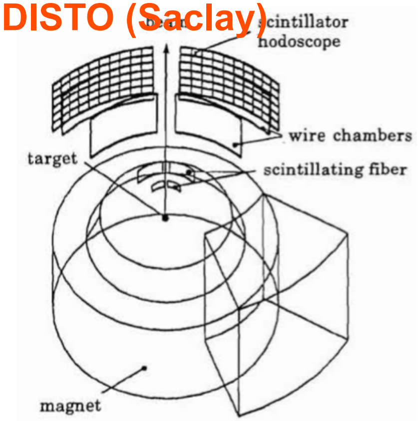


NEMO (Modane)

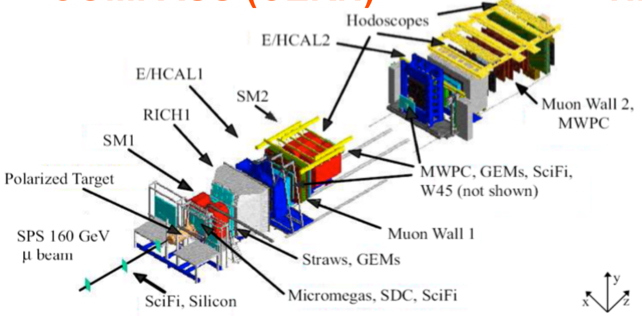


(1) central frame with the metallic foil,
(2) tracking device of 10 frames with 2x32 Geiger tubes each,
(3) scintillator walls of 5x5 counters.

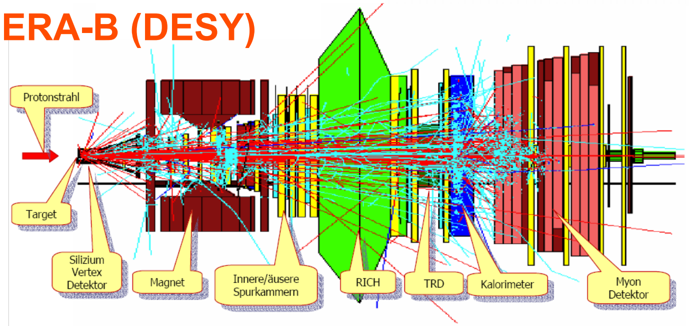
DISTO (Saclay)



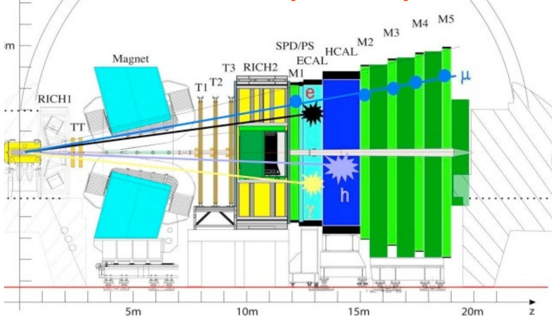
COMPASS (CERN)



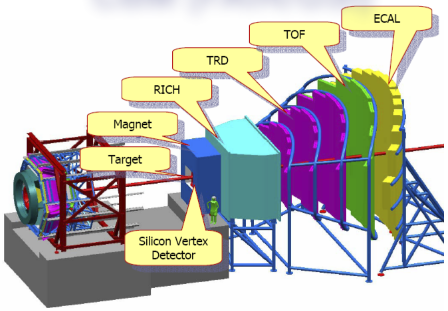
HERA-B (DESY)



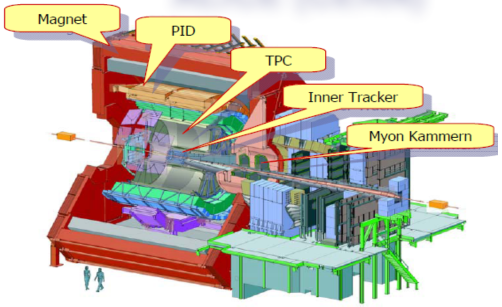
LHCb (CERN)



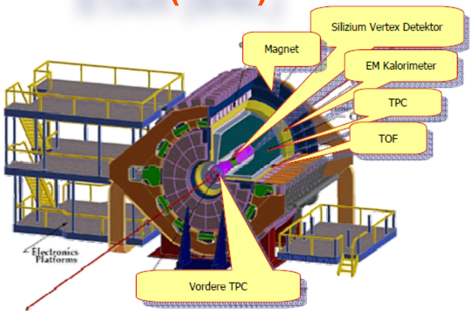
CBM (FAIR/GSI)



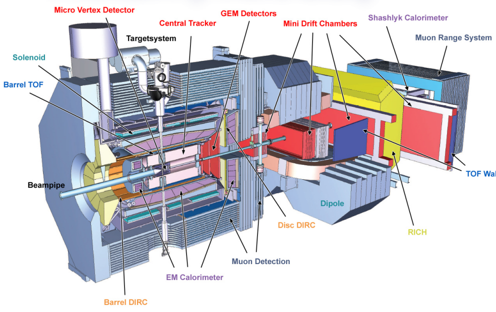
ALICE (CERN)



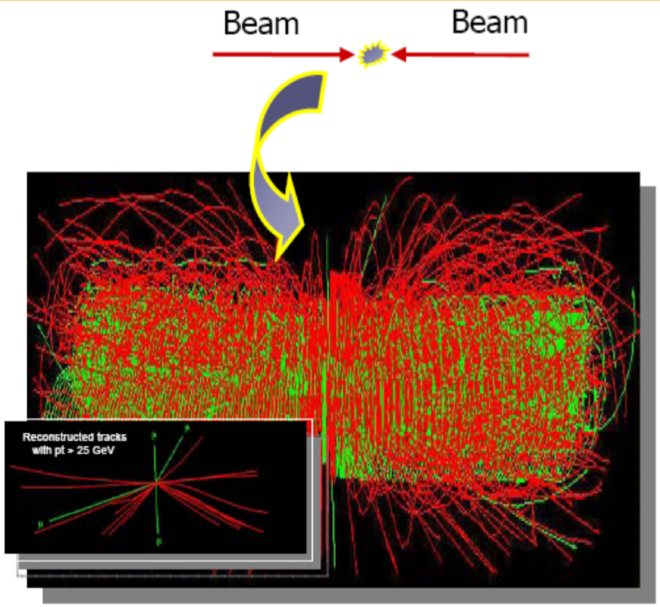
STAR (BNL)



PANDA (FAIR/GSI)



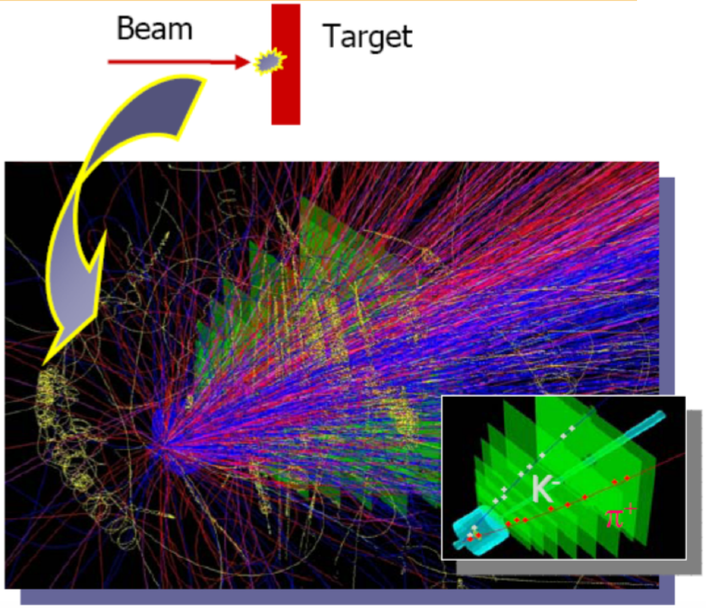
HEP Experiments: Collider and Fixed-Target



Inelastic collisions
$10^7 – 10^9$

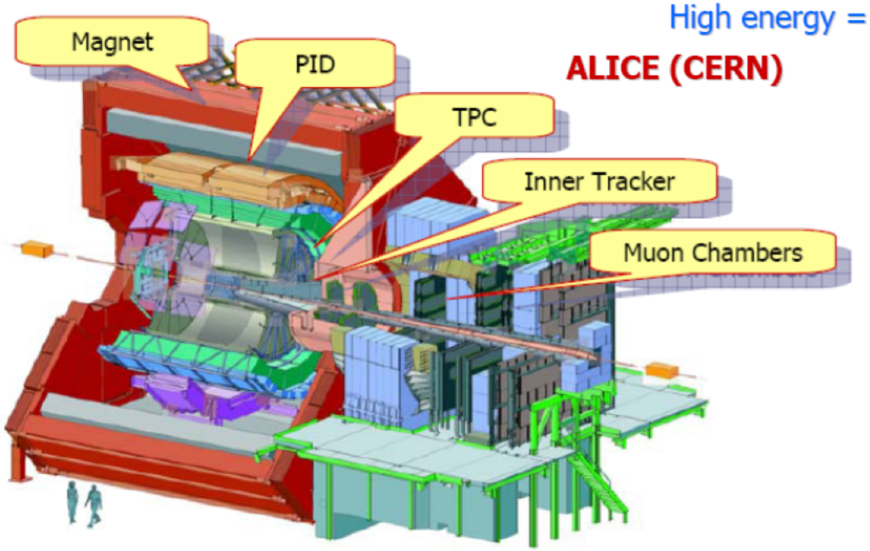
10^{11}

Signal events
$10^2 – 10^{-2}$

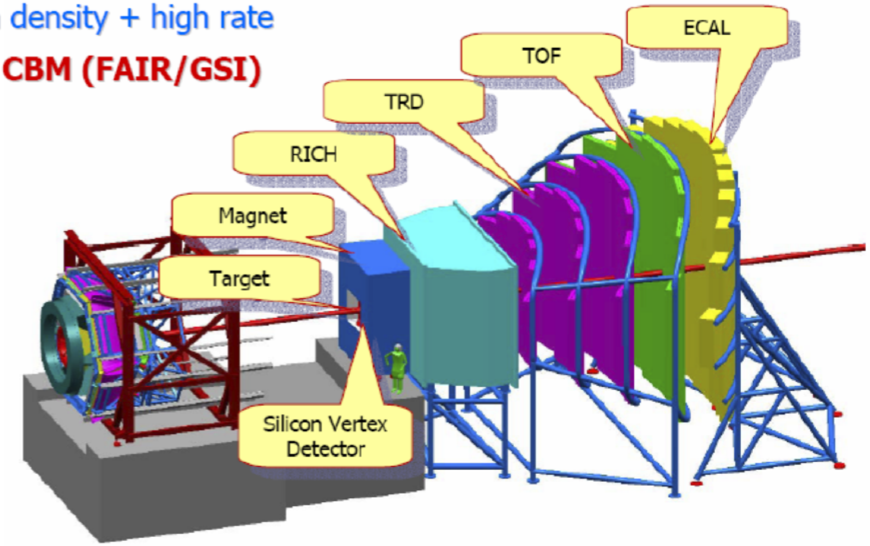


High energy = high density + high rate

ALICE (CERN)



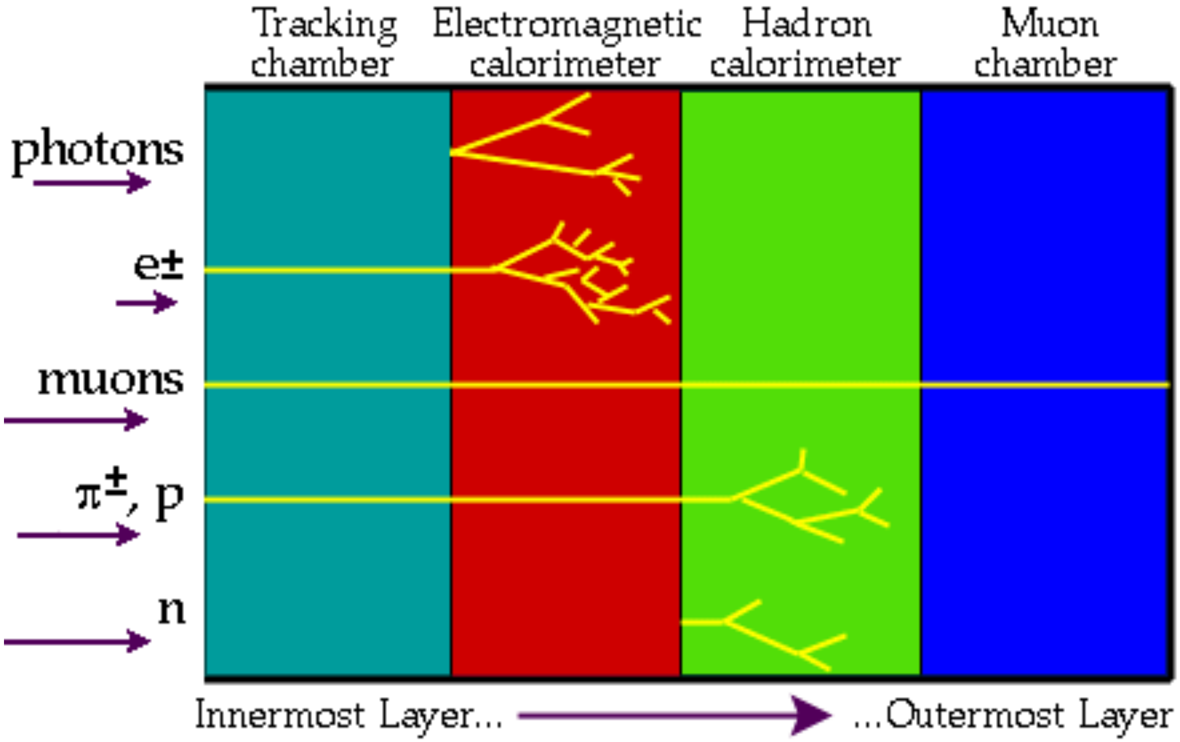
CBM (FAIR/GSI)



HEP Experiments: select interesting physics on-line

Typical Detector Components

The reason that detectors are divided into many components is that each component tests for a special set of particle properties. These components are stacked so that all particles will go through the different layers sequentially. A particle will not be evident until it either interacts with the detector in a measurable fashion, or decays into detectable particles.



- Charged particles, like electrons and protons, are detected both in the tracking chamber and the electromagnetic calorimeter.
- Neutral particles, like neutrons and photons, are not detectable in the tracking chamber; they are only evident when they interact with the detector. Photons are detected by the electromagnetic calorimeter, while neutrons are evidenced by the energy they deposit in the hadron calorimeter.
- Each particle type has its own "signature" in the detector. For example, if a particle is detected **only** in the electromagnetic calorimeter, then this is certainly a photon.

Detector Cross-Section with Particle Paths

Modern detectors consist of many different pieces of equipment which test for different aspects of an event. These many components are arranged in such a way that we can obtain the most data about the particles spawned by an event. In this way we can figure out the type of particle (PID) based on where that particle appeared in the detector.

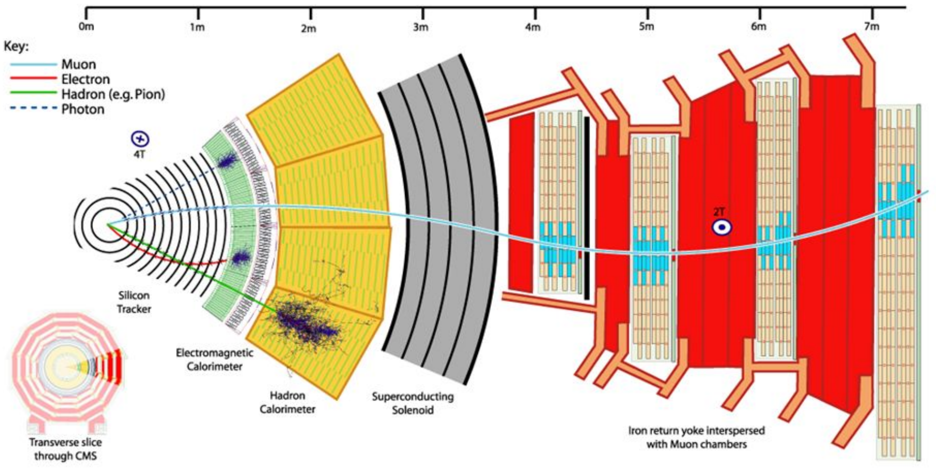
Tracking chamber: The inner region of the detector is filled with highly segmented sensing devices of various kinds, so that charged particle trajectories can be very accurately determined.

Electromagnetic Calorimeter: This device measures the total energy of e+, e-, and photons. These particles produce showers of e+/e- pairs in the material. The e-'s (or e+'s) are deflected by the electric fields of atoms, causing them to radiate photons. The photons then make e-/e+ pairs, which then radiate photons, etc. The number of final e+, e- pairs is proportional to the energy of the initiating particle.

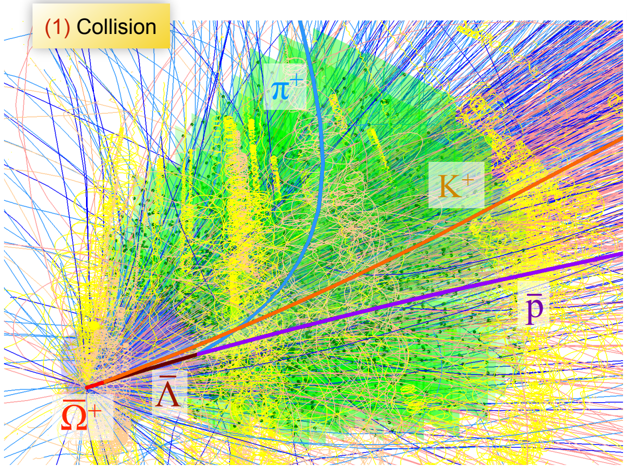
Hadron Calorimeter: This device measures the total energy of hadrons. The hadrons interact with the dense material in this region, producing a shower of charged particles. The energy that these charged particles deposit is then measured.

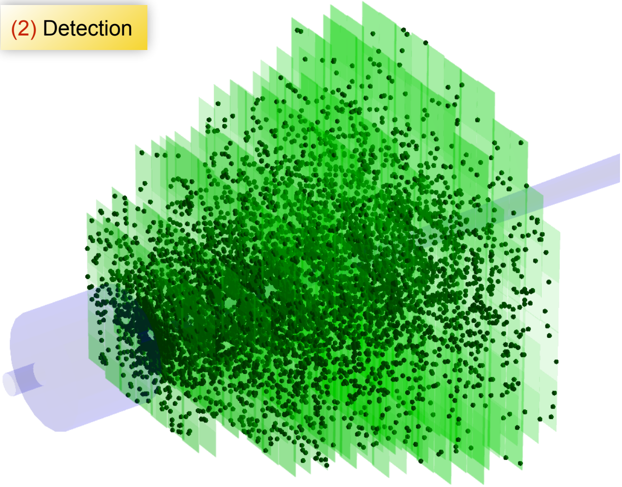
Muon Chambers: Only muons and neutrinos get this far. The muons are detected, but the weakly interacting neutrinos escape. The presence of neutrinos can be inferred by the "missing" energy.

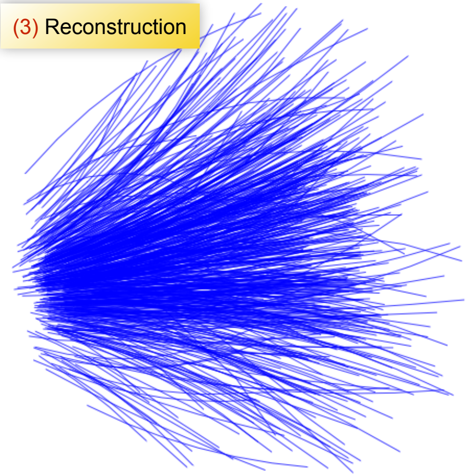
Magnet: The path of a charged particle curves in a magnetic field. The radius of curvature and direction tell the momentum and the sign of the charge.



Reconstruction Challenge in CBM at FAIR/GSI


(1) Collision

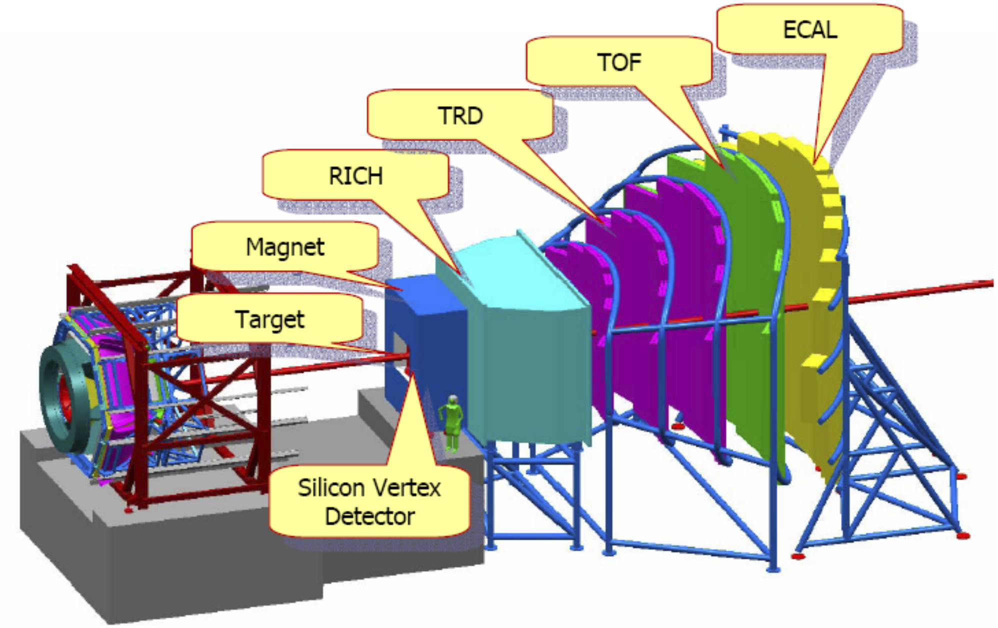

(2) Detection


(3) Reconstruction

- 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.

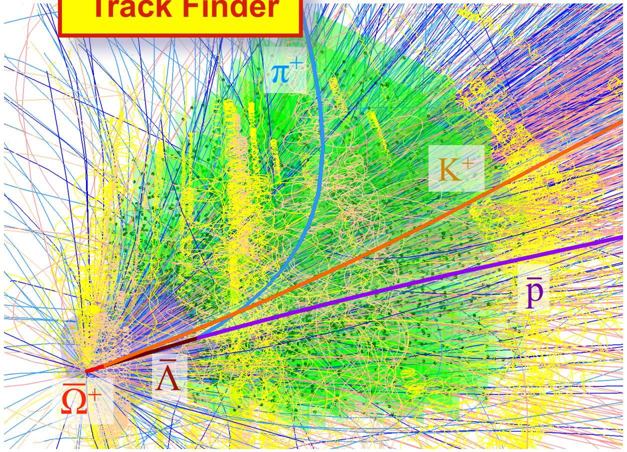
Cellular Automaton (CA) Track Finder
Kalman Filter (KF) Track Fitter
KF short-lived Particle Finder



All reconstruction algorithms are vectorized and parallelized.

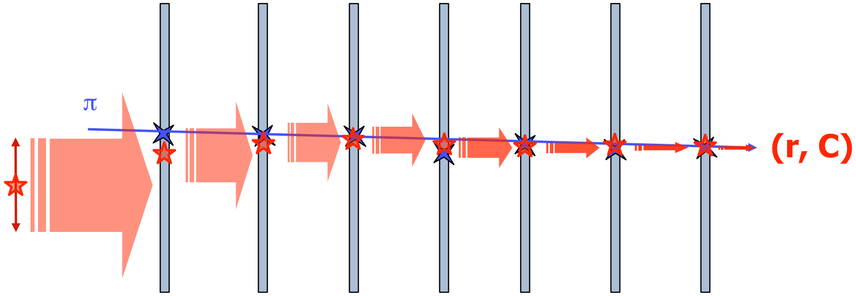
Stages of Event Reconstruction

1 Track Finder



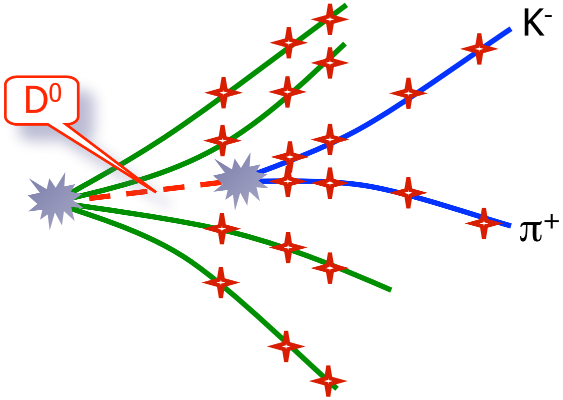
- Conformal Mapping
- Hough Transformation
- Track Following
- Cellular Automaton

2 Track Fitter



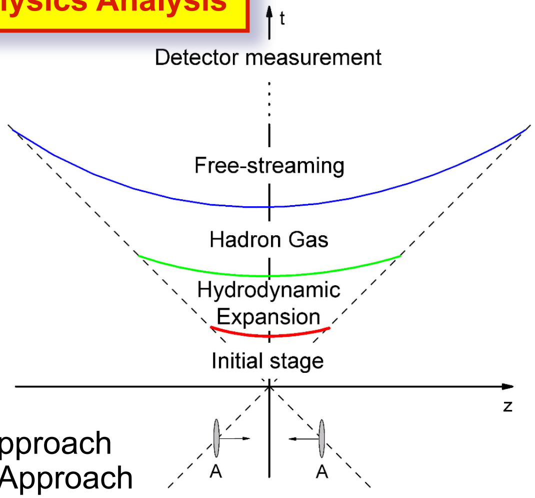
- Kalman Filter

3 Short-Lived Particles Finder



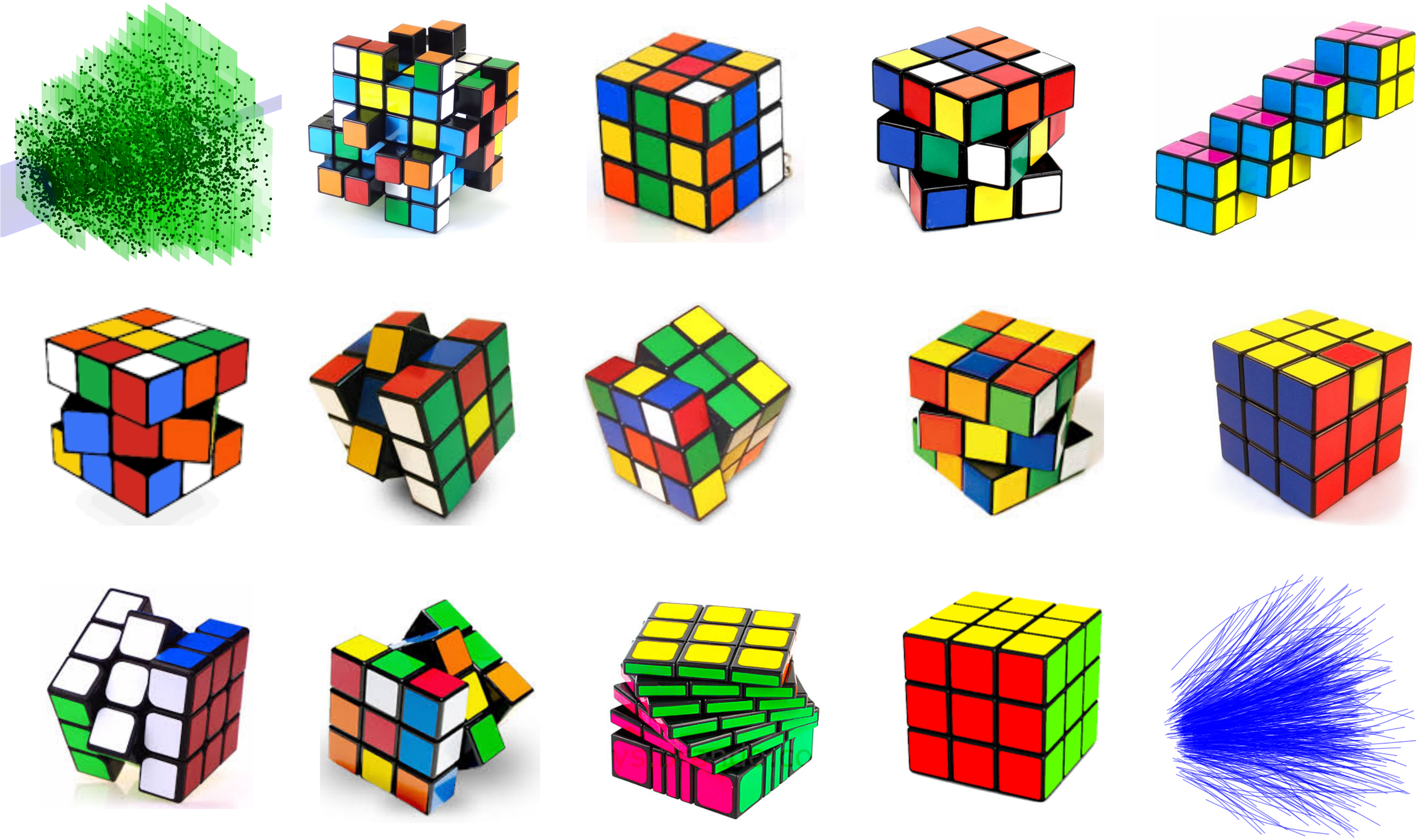
- Kalman Filter

4 Online Physics Analysis



- Direct Approach
- Inverse Approach

Track Finding: Rubik's Cube in CBM



The world record times are: 4.73 seconds/Rubik's cube by Feliks Zemdegs (Australia) and
0.0000045 seconds/CBM cube by FAIR-Russia Research Center HPC (Moscow)

1

Global Methods: Conformal Mapping + Histogramming

Conformal Mapping:

Transform circles into straight lines

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

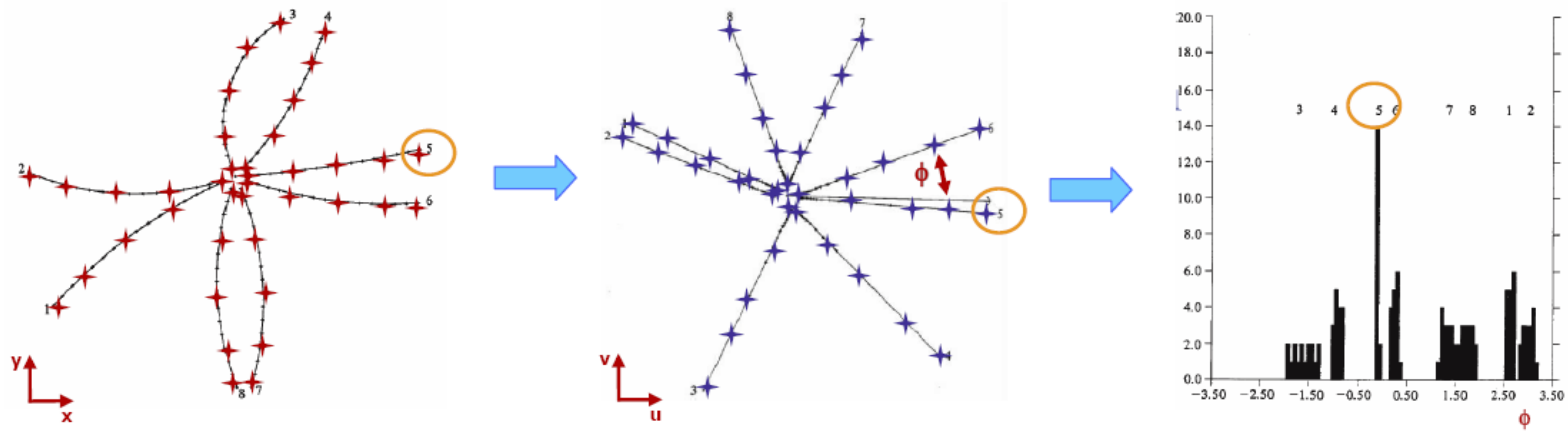
$v = -y/(x^2+y^2)$

Histogram:

Collect a histogram of azimuth angles ϕ

Find peaks in the histogram

Collect hits into tracks



Strong features:

- Impressive visual simplification of the problem
- Each step is easy to implement in hardware
- This results in a fast algorithm
- ...

Weak points:

- Non-obvious complications of the problem
- No tracks found, but only approximate track parameters
- No hits grouping into track candidates
- Therefore, no possibility to refit tracks
- Needs to know exact position of the interaction point
- Finds only primary tracks
- Reorder the hits (last <-> first)
- Measurement errors are now no more uniform
- Geometry of the setup must be transformed
- The same for the (non-uniform) magnetic field map
- What with the Lorentz force: $\mathbf{F} = q(\mathbf{E}+\mathbf{v}\times\mathbf{B})$?
- ...

Useful implemented in hardware and for very simple event topologies only

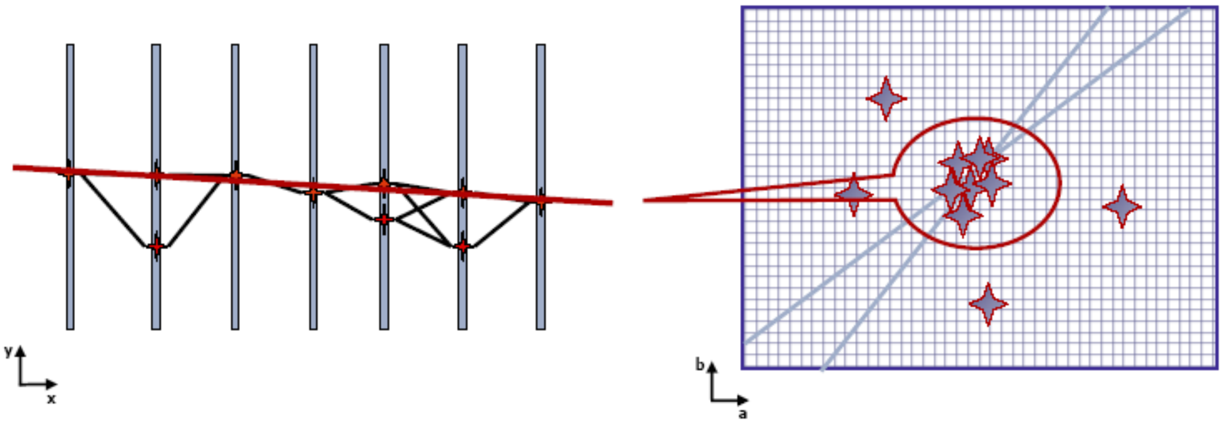
Global Methods: Hough Transformation

Measurement Space

$y = a*x + b$

Parameter Space

$b = -x*a + y$



Strong features:

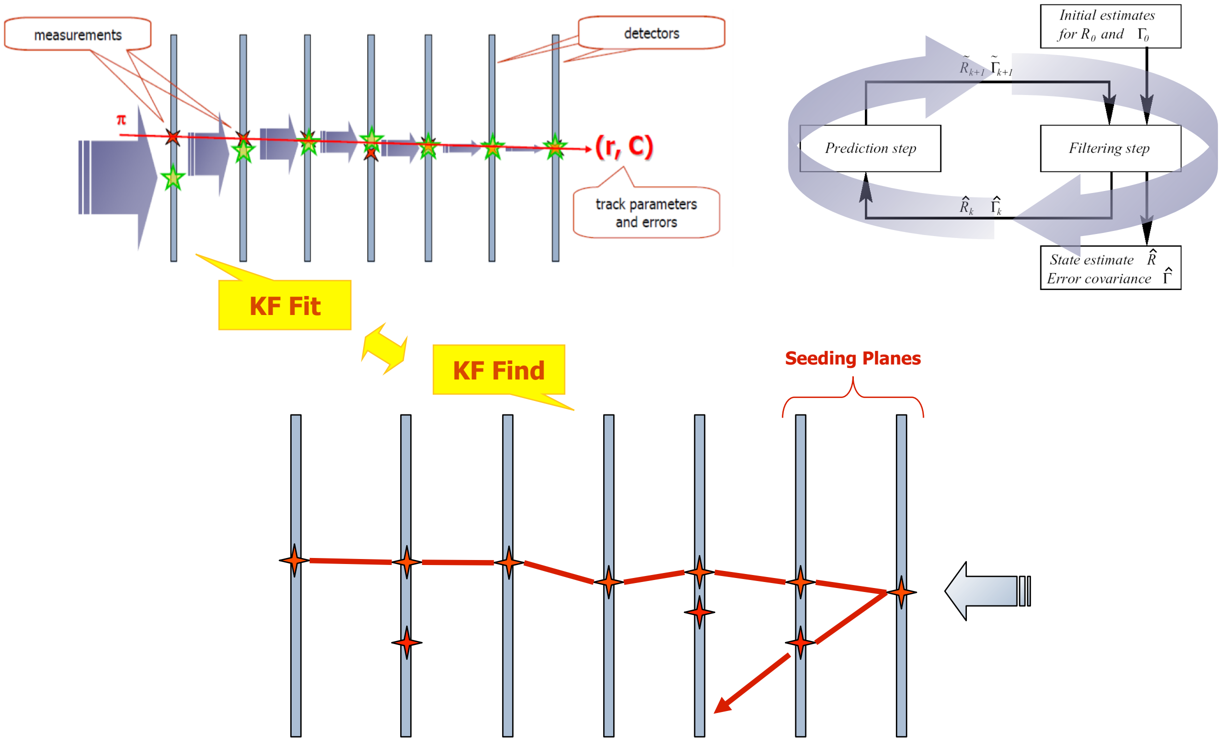
- Generalization of the histogramming method
- Easy to implement in hardware
- This results in a fast algorithm
- ...

Weak points:

- Needs a global track model
- Therefore, appropriate for simple magnetic fields only
- Does not include multiple scattering -> only fast tracks
- Histogramming provides only track parameters etc.
- Not possible competition between track candidates
- Histogramming needs access to main memory -> slow
- 5D histogramming (x, y, tx, ty, q/p) needs a lot of memory
- Precise tracking requires even more memory
- ...

Useful implemented in hardware and for simple event and trigger topologies

Local Methods: Kalman Filter for Track Following



Strong features:

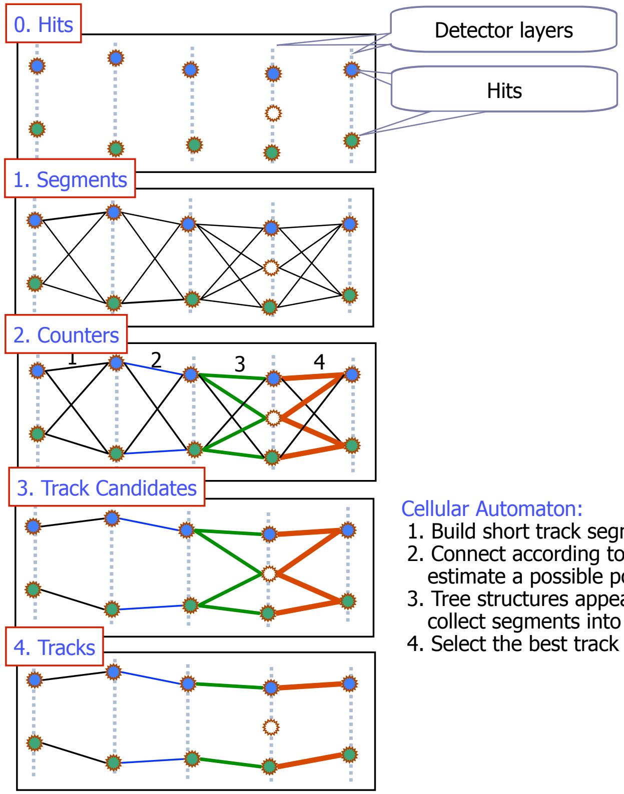
- Psychologically easy to accept hit by hit track finding
- Combined track finder and fitter based on KF
- Development of a new experiment starts with an ideal MC track finder and a realistic KF track fitter, therefore the next step to a realistic track finder is obvious – KF
- ...

Weak points:

- Based on a single track approach
- Needs seeding (starting short track segments)
- Final efficiency is always limited by seeding efficiency
- It is limited also by the efficiency of the seeding chambers
- Works at the hits level, searching for hits within a region
- Repeats calculations, when discarding track candidates
- Therefore needs a lot of seeds -> even larger combinatorics
- How many inefficient detectors can be tolerated in general ?
- Too early competition between track candidates
- ...

Useful for relatively simple event topologies and as a second after the ideal 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.

Strong features:

- Local relations -> simple calculations
- Local relations -> parallel algorithm
- Staged implementation: hits -> segments -> tracks
- Polynomial (2nd order) combinatorics
- Track competition at the global level
- Includes the KF fitter, if necessary, for high track densities
- Detector inefficiency problem outside the combinatorics
- ...

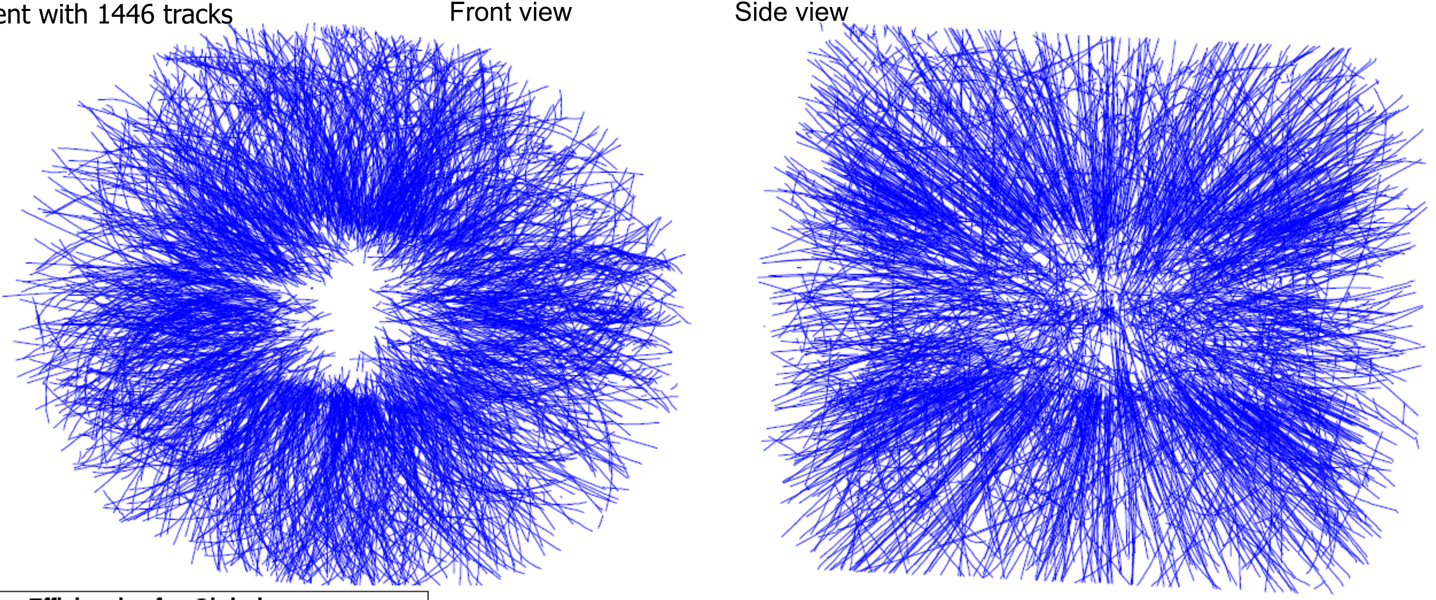
Weak points:

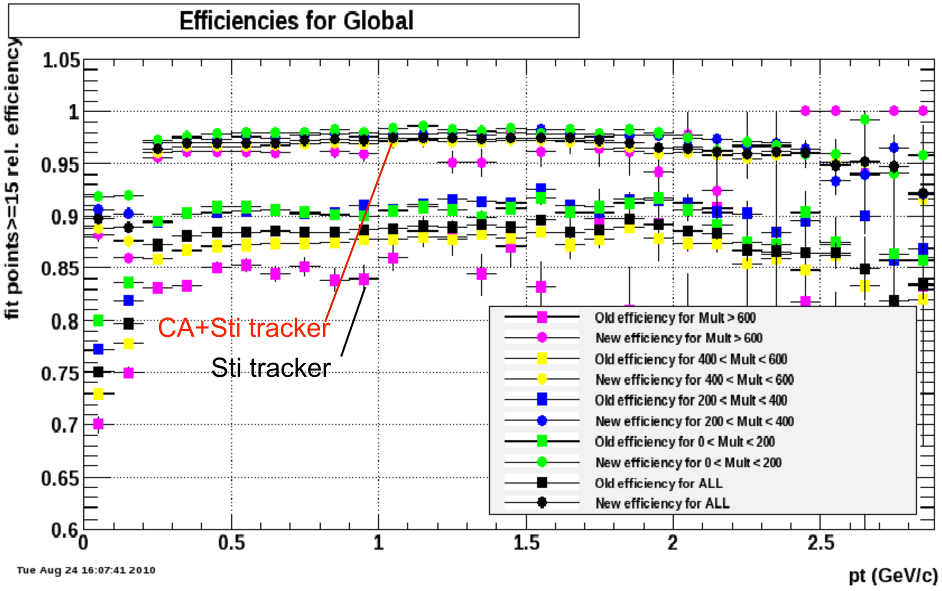
- Not easy to understand a parallel algorithm (Game of Life)
- Currently implementations on sequential computers
- Parallel hardware is coming now
- ...

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

STAR TPC CA Track Finder

Au-Au event with 1446 tracks





Efficiency and ratio, %	
Ref Set	96.6
All Set	88.6
Clone	10.6
Ghost	12.6
Tracks/ev	659
Time/ev, ms	47

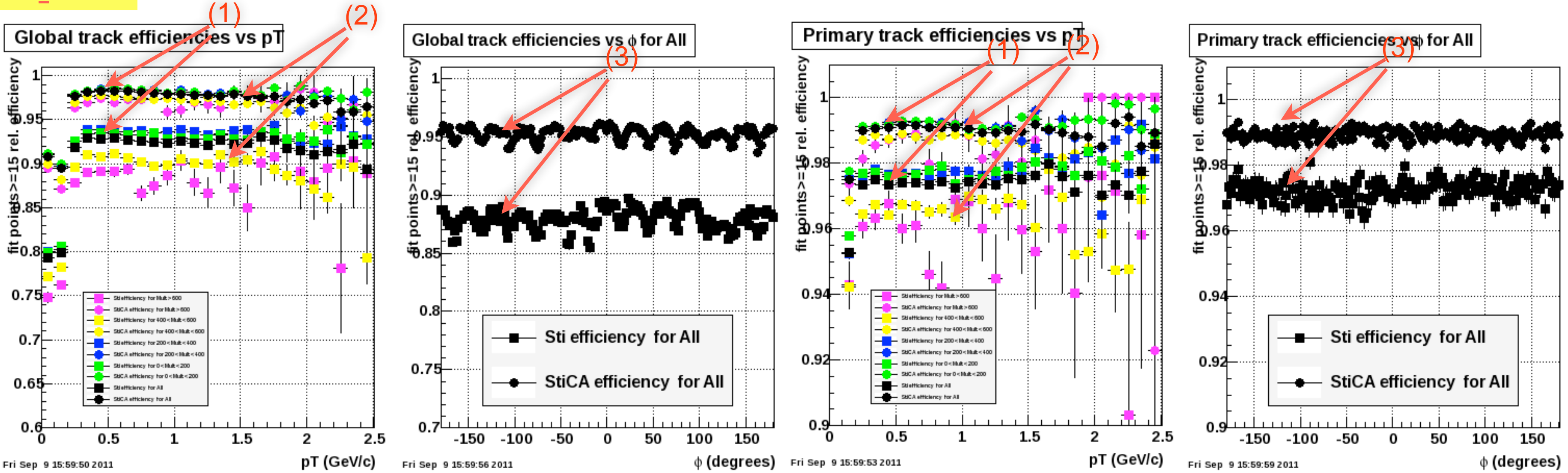
All set: $p \geq 0.05$ GeV/c
Reference set: $p \geq 1$ GeV/c
Ghost: purity < 90%

The CA track finder is more stable w.r.t. track multiplicity and is ~10 times faster than the TF based Sti track finder.

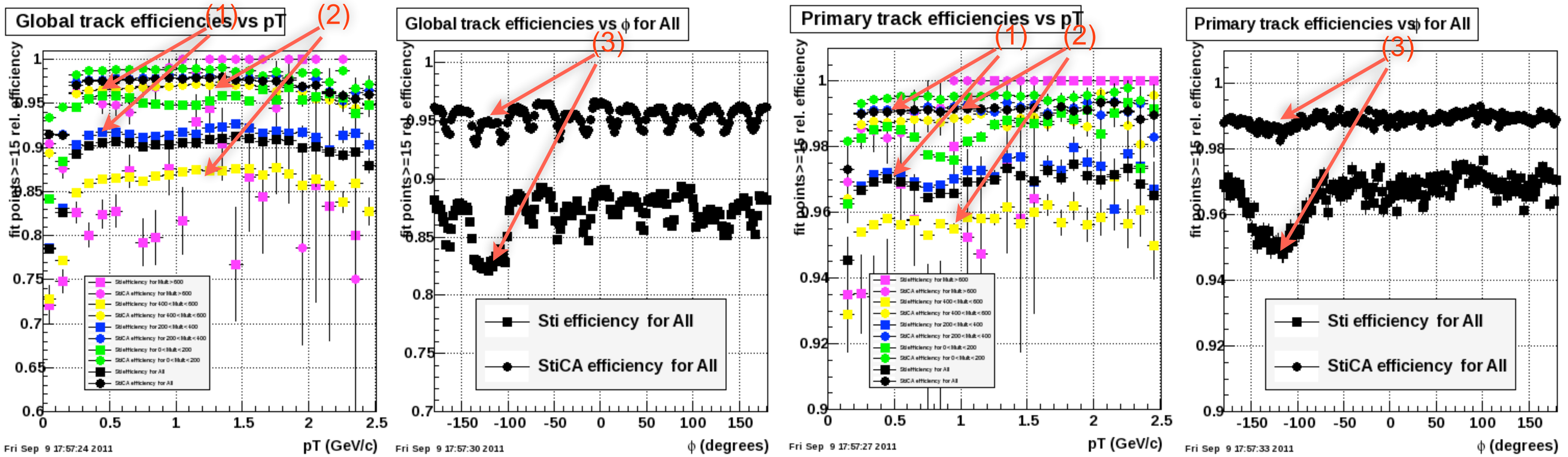
1

CA+Sti vs. Sti

Y2004_AuAu200

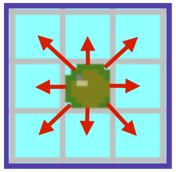


Y2010_AuAu200



CA+Sti has (1) a higher efficiency, is (2) stable w.r.t. track multiplicity and (3) robust w.r.t. detector inefficiency.

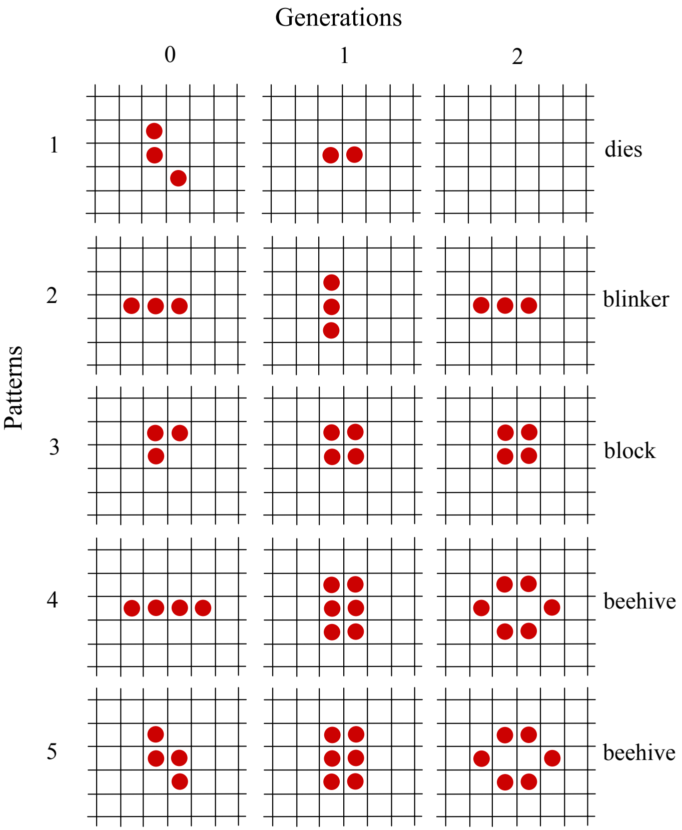
Cellular Automaton - Game “Life”

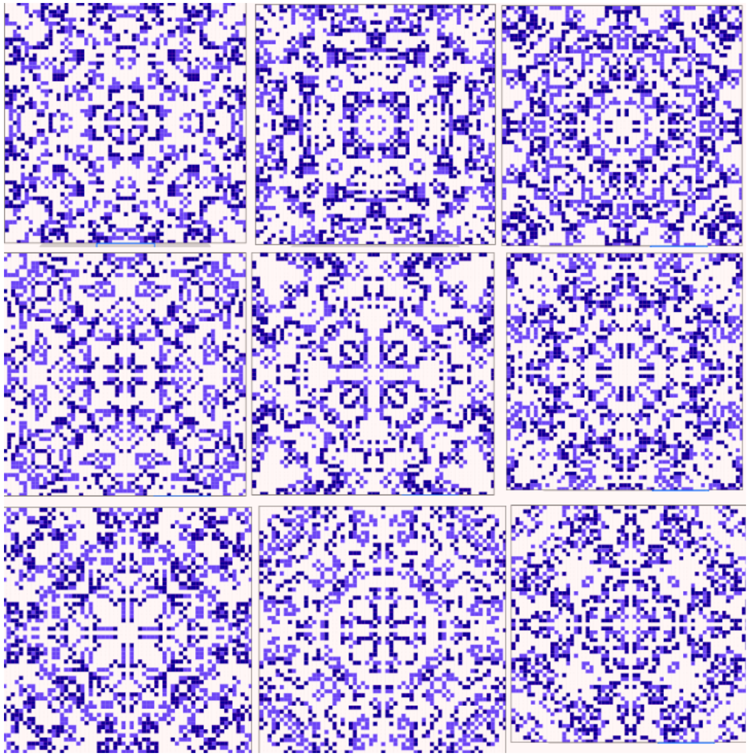


Each **cell** has 8 neighboring cells, 4 adjacent orthogonally, 4 adjacent diagonally. The **rules** are:

Survival: Every counter with 2 or 3 neighboring counters survives for the next generation.
Death: Each counter with 4 or more neighbors dies from overpopulation, with 1 neighbor or none dies from isolation.
Birth: Each empty cell adjacent to exactly 3 neighbors is a birth cell.

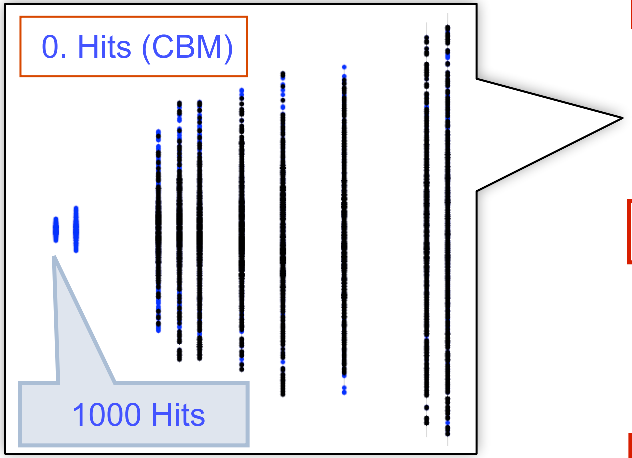
It is important to understand that all births and deaths occur *simultaneously*.

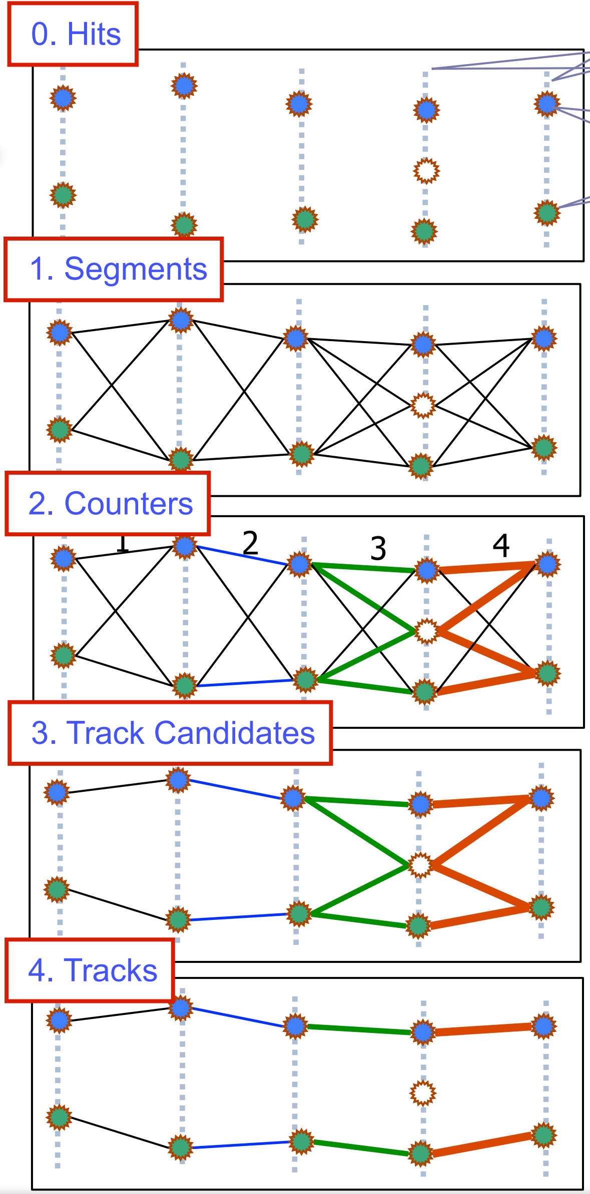




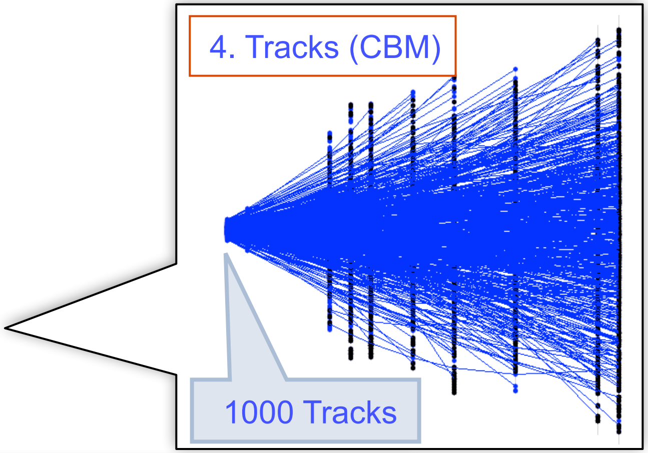
Martin Gardner, Mathematical games: The fantastic combinations of John Conway‘s new solitaire game „Life“, Sci. Amer., 223 (1970) 120-123

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.



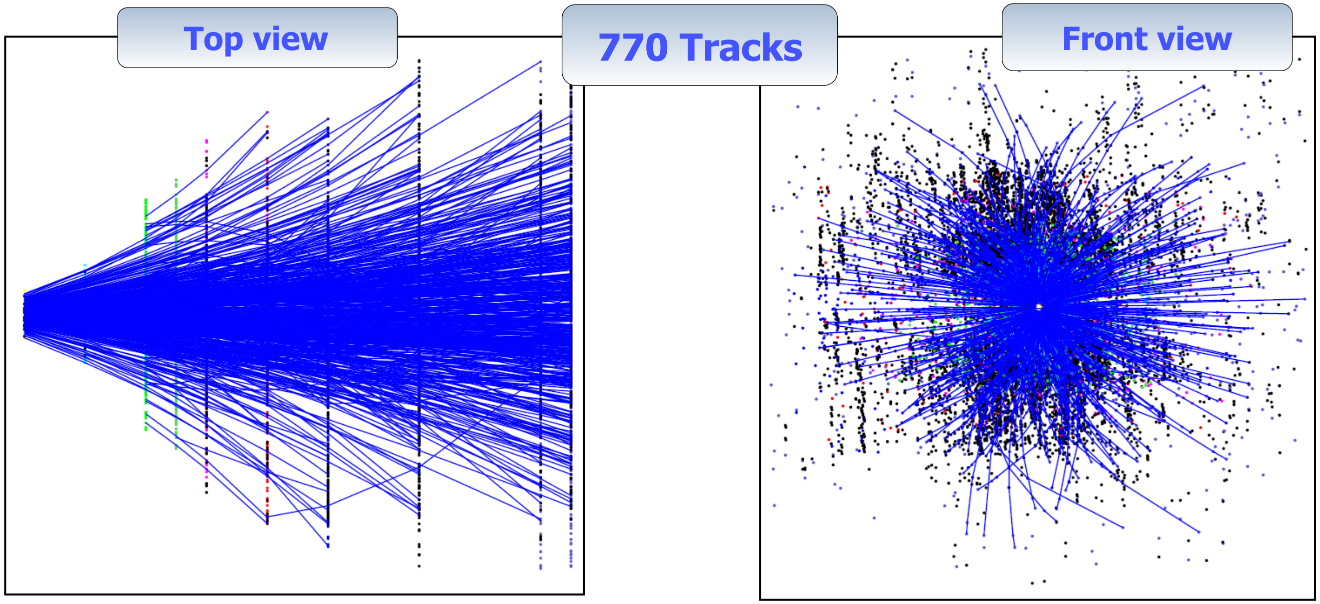
Cellular Automaton:
- local w.r.t. data
- intrinsically parallel
- extremely simple
- very fast

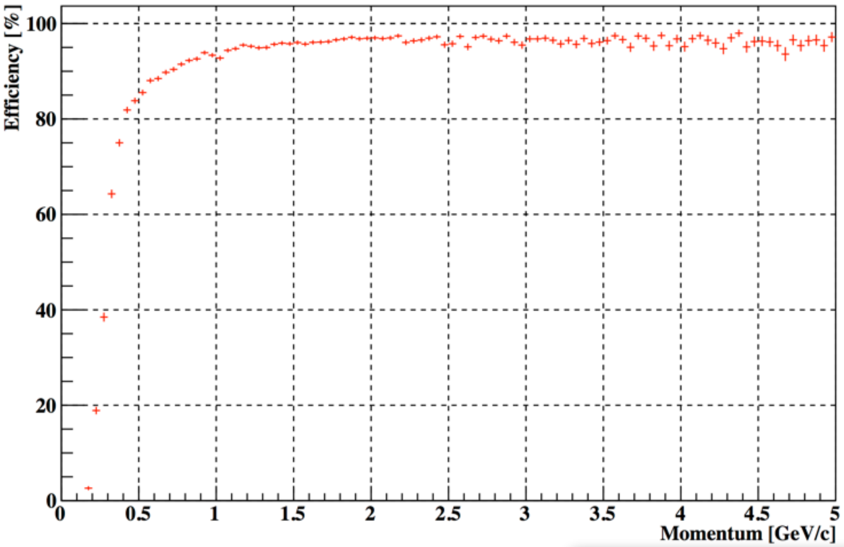
Perfect for many-core CPU/GPU !

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

1

CBM CA Track Finder



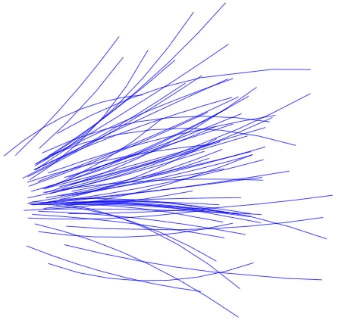


	Efficiency, %	
	mbias	central
Primary high-p tracks	97.1	96.2
Primary low-p tracks	90.4	90.7
Secondary high-p tracks	81.2	81.4
Secondary low-p tracks	51.1	50.6
All tracks	88.5	88.3
Clone level	0.2	0.2
Ghost level	0.7	1.5
Reconstructed tracks/event	120	591
Time/event/core	8.2 ms	57 ms

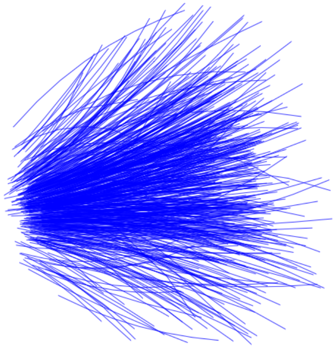
Efficient and stable event reconstruction

CA Track Finder at High Track Multiplicity

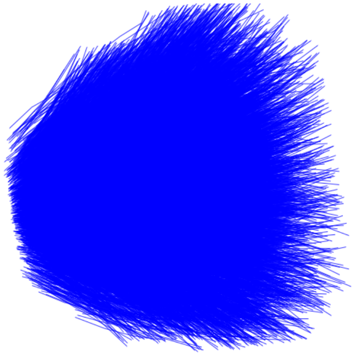
A number of minimum bias events is gathered into a group (super-event), which is then treated by the CA track finder as a single event



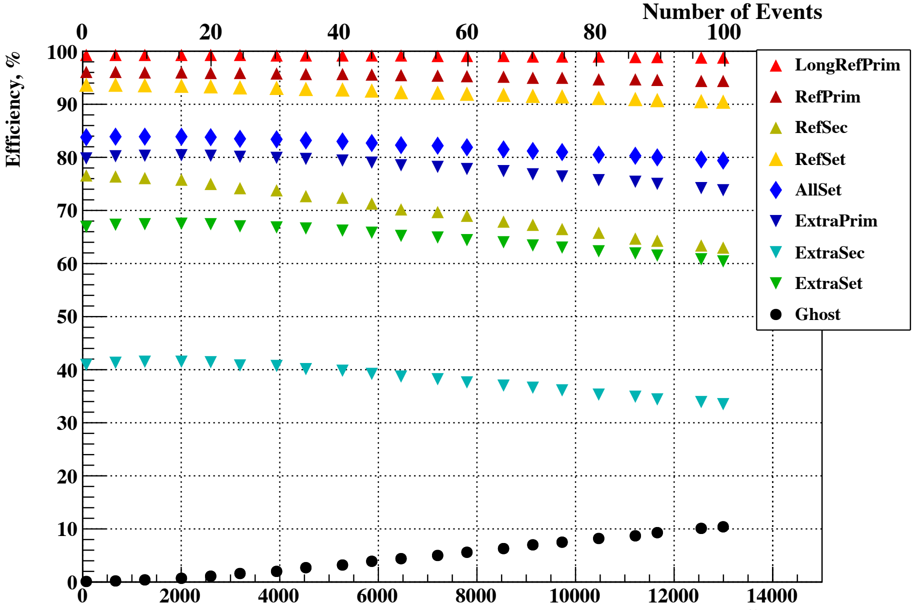
1 mbias event, $\langle N_{reco} \rangle = 109$



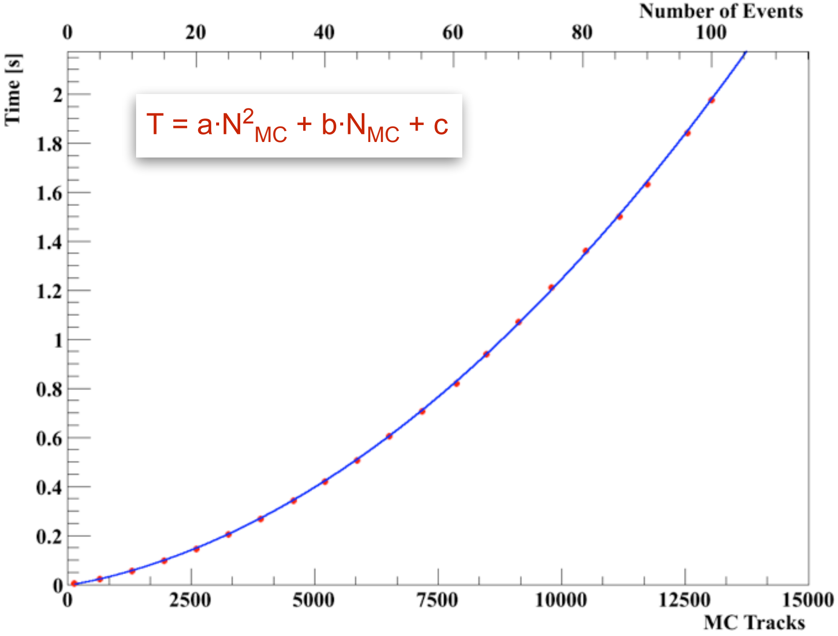
5 mbias events, $\langle N_{reco} \rangle = 572$



100 mbias events, $\langle N_{reco} \rangle = 10340$



Au+Au mbias events at 25 AGeV

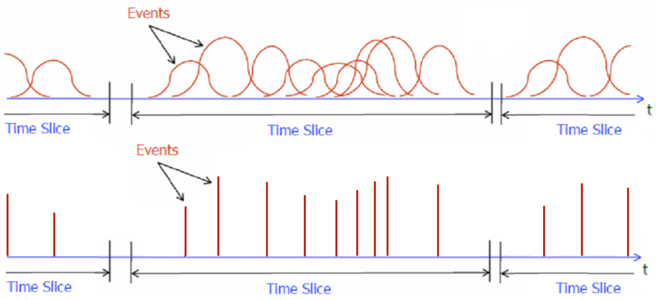


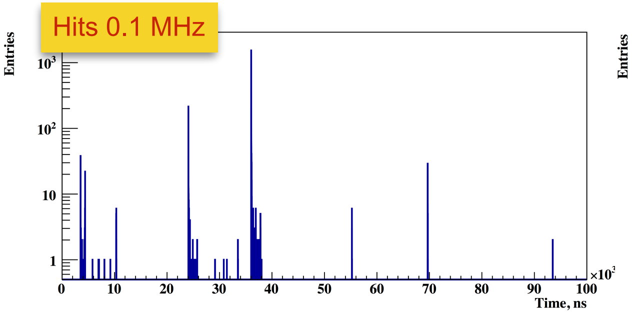
$T = a \cdot N^2_{MC} + b \cdot N_{MC} + c$

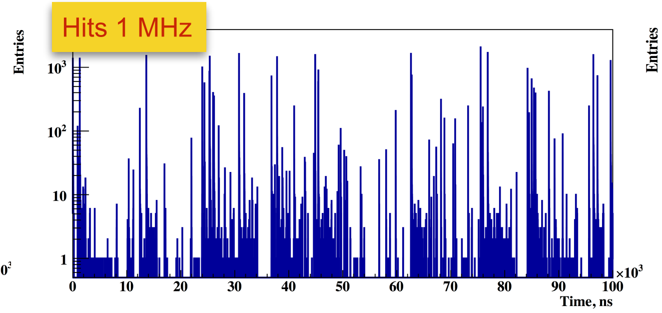
Stable reconstruction efficiency and time as a second order polynomial w.r.t. to track multiplicity

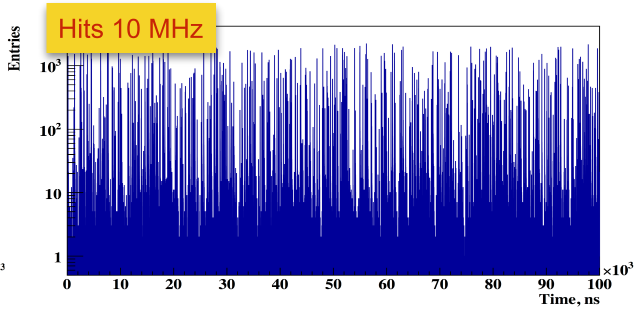
4D Track Reconstruction and Event Building at 10 MHz

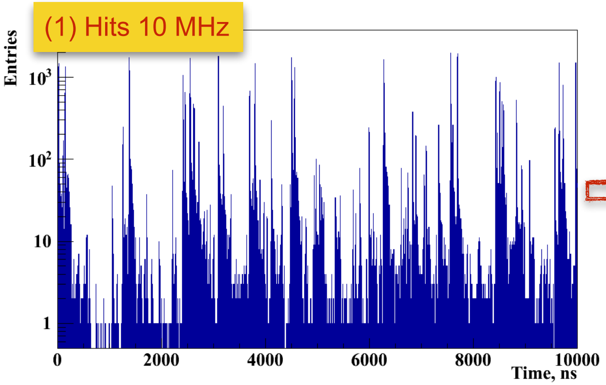
- The beam in the CBM will have no bunch structure, but continuous.
- Measurements in this case will be 4D (x, y, z, t).
- Reconstruction of time slices rather than events will be needed.

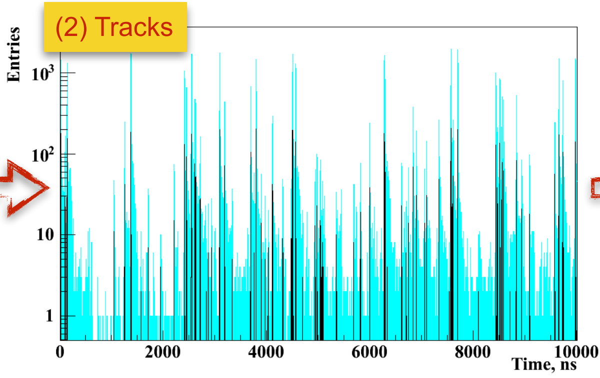


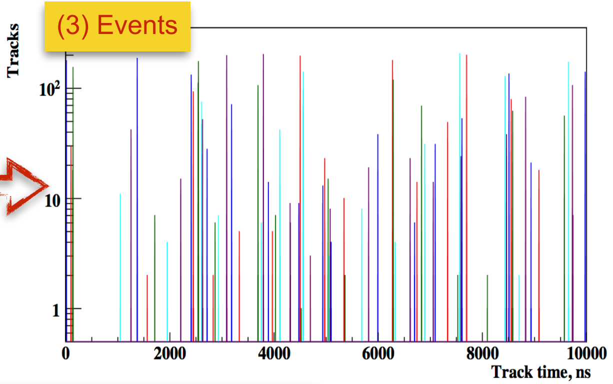

Hits 0.1 MHz


Hits 1 MHz


Hits 10 MHz


(1) Hits 10 MHz


(2) Tracks

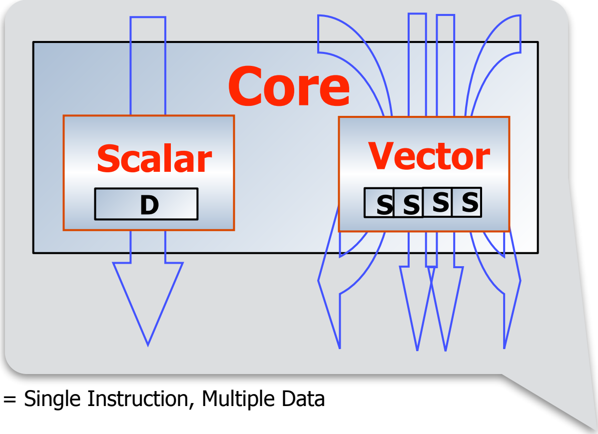

(3) Events

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

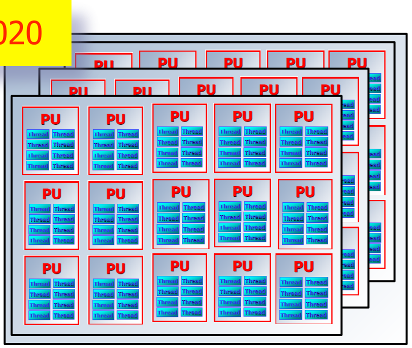
Many-core HPC: Cores, Threads and Vectors

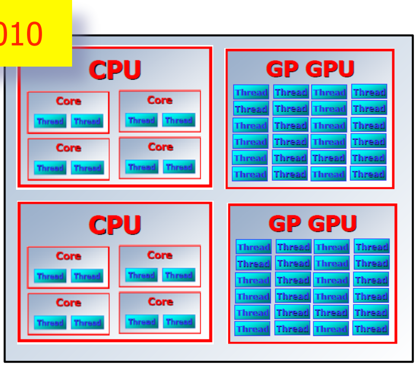
HEP experiments work with high data rates, therefore need High Performance Computing (HPC) !

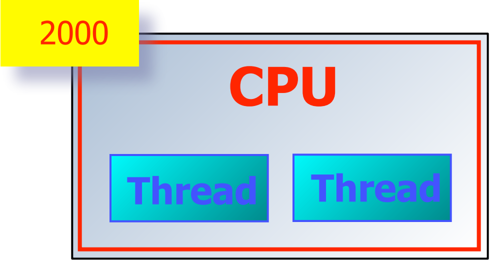
Vectors (SIMD) = data level parallelism

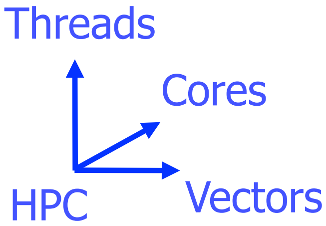


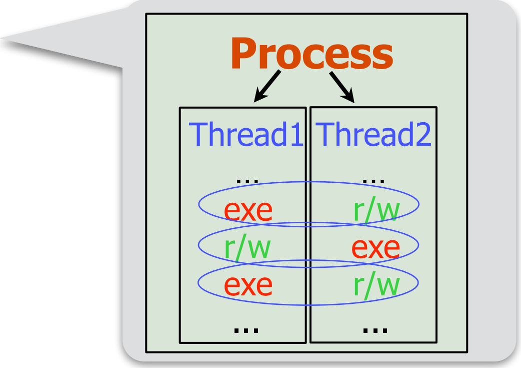
SIMD = Single Instruction, Multiple Data











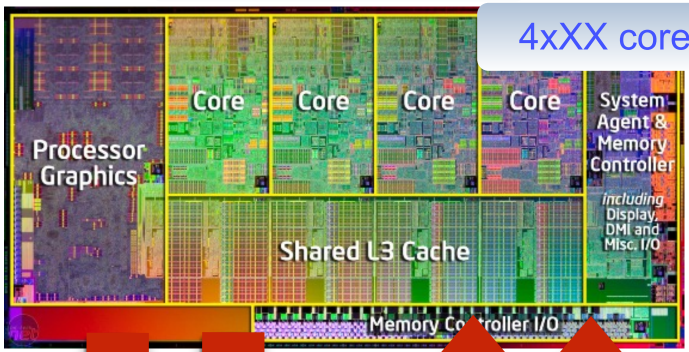
Cores and Threads = task level parallelism

Fundamental redesign of traditional approaches to data processing is necessary

Many-Core CPU/GPU Architectures

Intel/AMD CPU

4xXX cores



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

Math

Memory

Parallelism

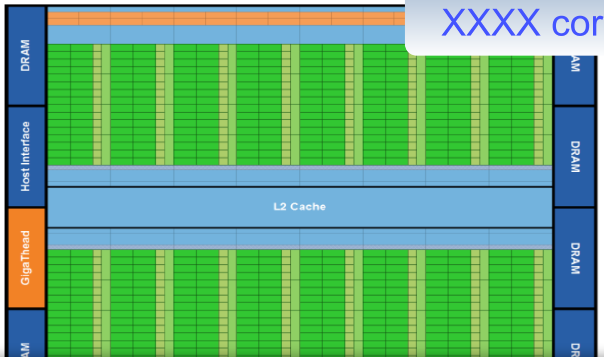
Math

Memory

#Cores

Nvidia/ATI GPU

XXXX cores



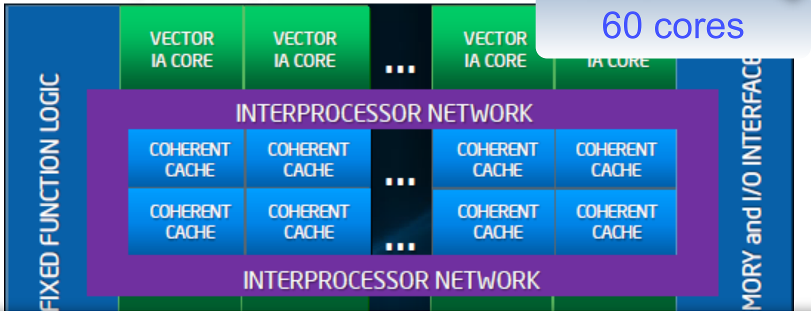
- Optimized for data-parallel, throughput computation
- More transistors dedicated to computation

Stability

Memory

Intel Phi

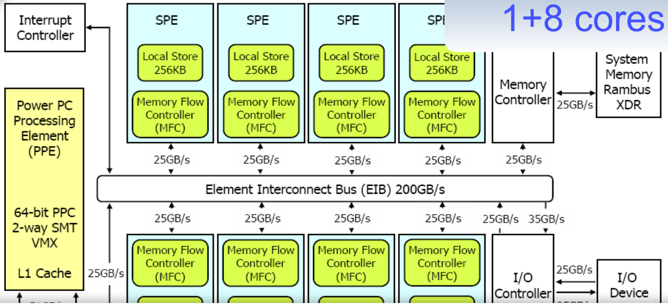
60 cores



- 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



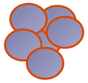
- General purpose RISC processor (PowerPC)
- 8 co-processors (SPE, Synergistic Processor Elements)
- 128-bit wide SIMD units

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

Kalman Filter Algorithm

The Kalman filter is a recursive estimator – only the estimated state from the previous time step and the current measurement are needed to compute the estimate for the current state.

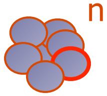
mean value over n measurements



$$\mu_n = \frac{1}{n}\sum_{i=1}^{n} x_i$$

previous estimation

new measurement



$$\mu_{n+1} = \frac{1}{n+1}\sum_{i=1}^{n+1} x_i = \frac{n}{n+1}\left(\frac{1}{n}\sum_{i=1}^{n} x_i + \frac{1}{n}x_{n+1}\right) = \frac{n}{n+1}\mu_n + \frac{1}{n+1}x_{n+1} = \mu_n + \frac{1}{n+1}\underbrace{(x_{n+1}-\mu_n)}$$

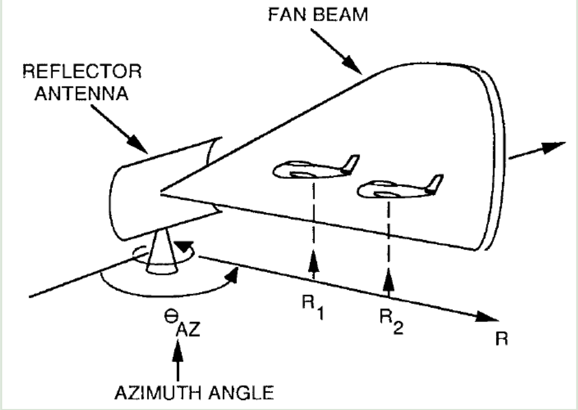
mean value over n+1 measurements

weight

correction

For this work, U.S. President Barack Obama rewarded Rudolf Kálmán with the National Medal of Science on October 7, 2009.

Radar applications



state vector:

$$\mathbf{r} = \{\, x, y, z, v_x, v_y, v_z \,\}$$

covariance matrix:

$$\mathbf{C} = \left\{\begin{matrix} \sigma^2_x & & & & & \\ & \sigma^2_y & & & \dots & \\ & & \sigma^2_z & & & \\ & & & \sigma^2_{vx} & & \\ & \dots & & & \sigma^2_{vy} & \\ & & & & & \sigma^2_{vz} \end{matrix}\right\}$$

Apollo Flight Journal

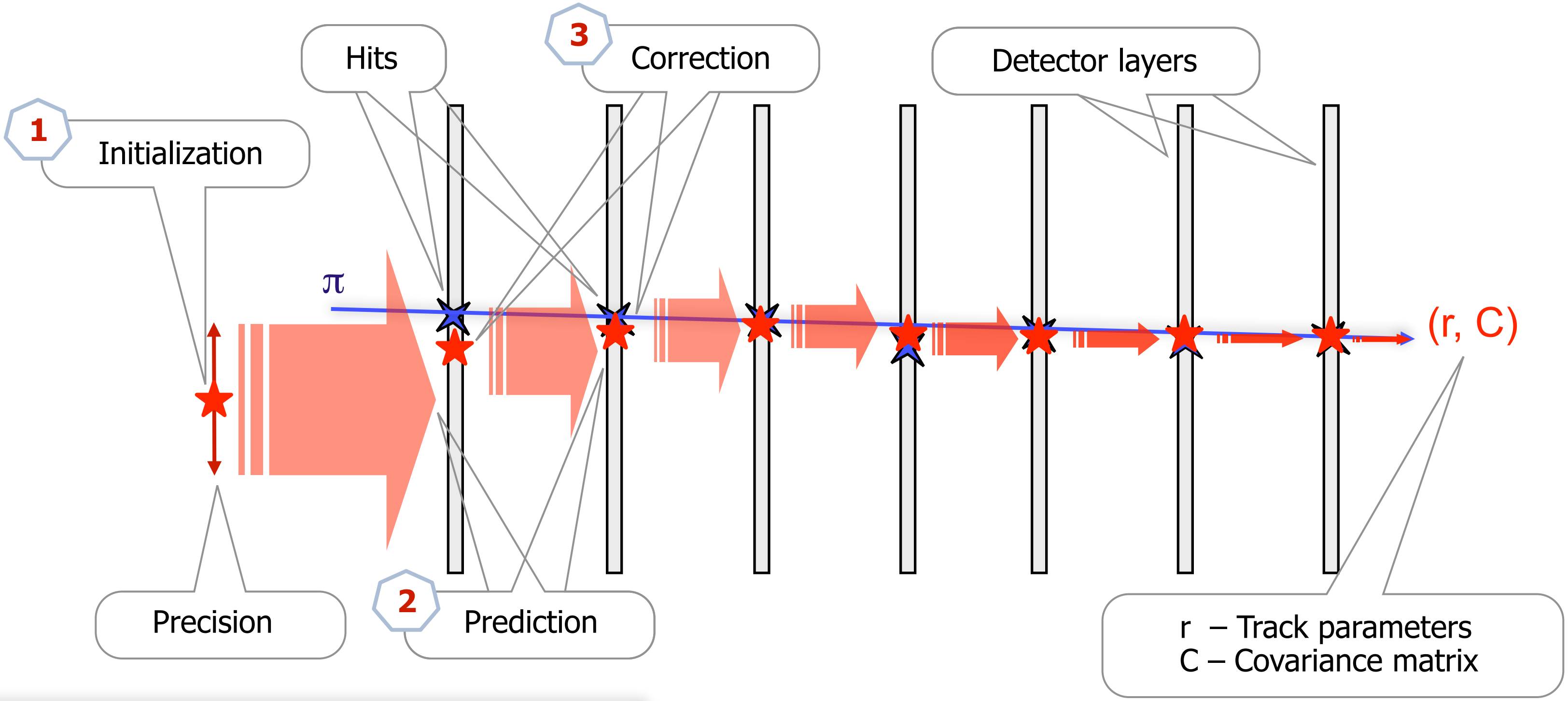
December 21, 1968. The Apollo 8 spacecraft has just been sent on its way to the Moon.

003:46:31 Collins: Roger. At your convenience, would you please go P00 and Accept? We're going to update to your W-matrix.

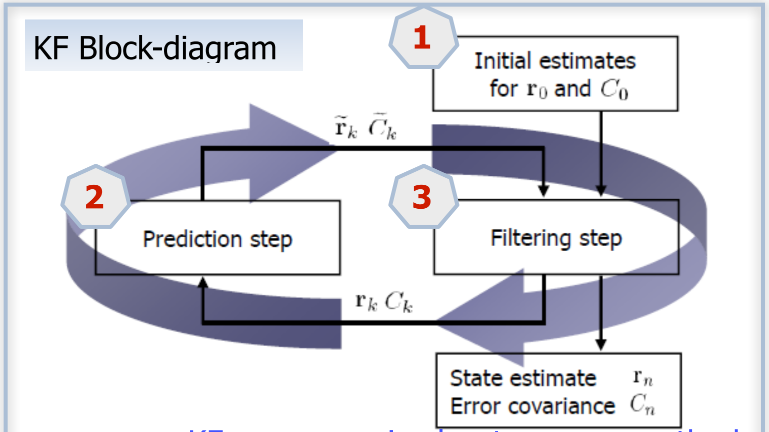
2

Kalman Filter based Track Fit

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



r – Track parameters
C – Covariance matrix



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 Track Fit Quality

CBM track parameters: $r = \{ x, y, t_x, t_y, q/p \}$

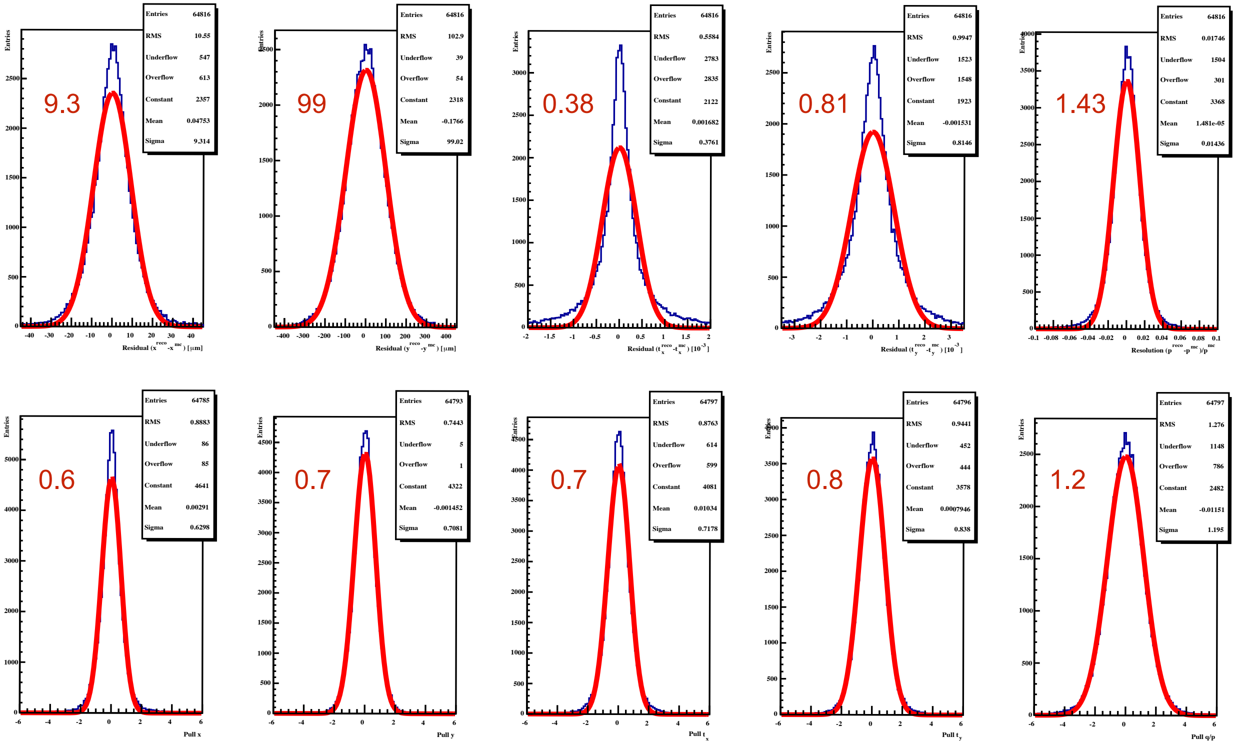
position, tg of slopes, charge over momentum

residuals

$$\rho_x = x_{reco} - x_{mc}$$

pulls

$$P(x) = \frac{\rho_x}{\sqrt{C_{xx}}}$$



resolution					pull widths				
x, μm	y, μm	t_x, 10^{-3}	t_y, 10^{-3}	p, %	x	y	t_x	t_y	q/p
9.3	99	0.38	0.81	1.43	0.6	0.7	0.7	0.8	1.2

Input hits parameters: ρ x = 8.4, ρ y = 91, Pull x = 0.63, Pull y = 0.69

Illustrative slide, 2012!

Kalman Filter Track Fit on Cell

Intel: stages 1–3

Cell: stages 4–5

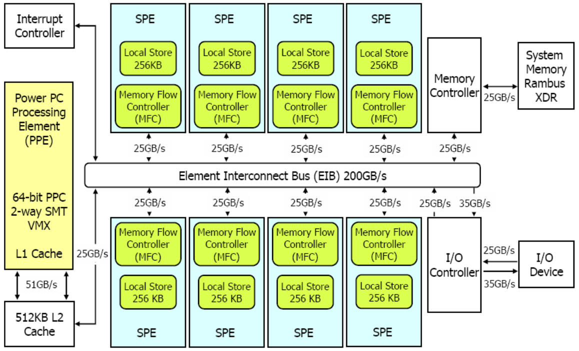
Stage	Description	Time/track	Speedup
	Initial scalar version	12 ms	–
1	Approximation of the magnetic field	240 μs	50
2	Optimization of the algorithm	7.2 μs	35
3	Vectorization	1.6 μs	4.5
4	Porting to SPE	1.1 μs	1.5
5	Parallelization on 16 SPEs	0.1 μs	10
	Final simdized version	0.1 μs	120000

10000x faster on any PC

Comp. Phys. Comm. 178 (2008) 374-383

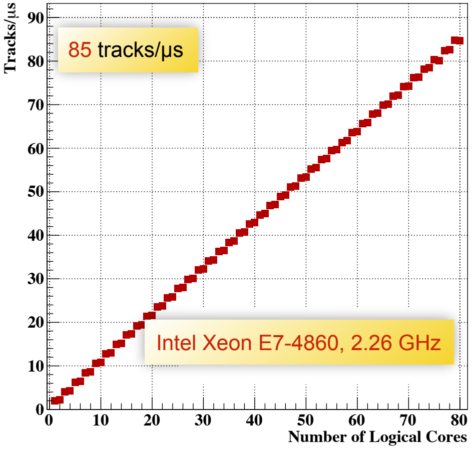
blade11bc4 @IBM, Böblingen:
2 Cell Broadband Engines, 256 kB LS, 2.4 GHz

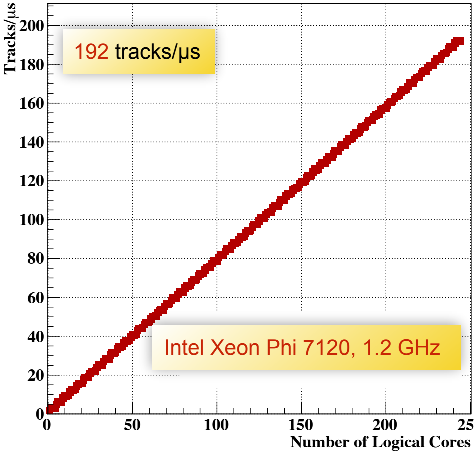
The KF speed was increased by 5 orders of magnitude

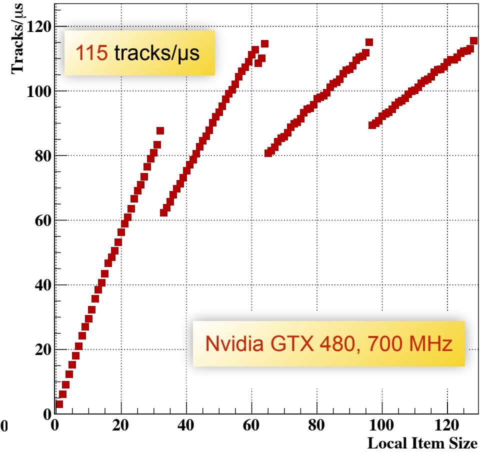


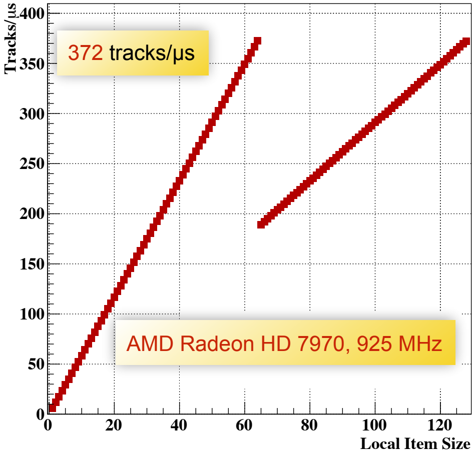
Motivated by, but not restricted to Cell !

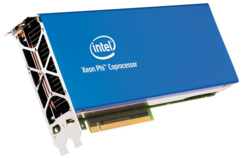
Full Portability of the KF Track Fit











- 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

Parallelization Challenge in the CBM Event Reconstruction

Location		Architecture	(Nodes·)sockets·cores·threads·SIMD	Data streams
CERN	Switzerland	AMD 6164HE	4·12·1·4	192
GSI	Germany	Intel E7-4860	4·10·2·4	320
ITEP	Russia	AMD 6272	100·(2·16·1·4)	12 800
FIAS	Germany	Intel E5-2600+Intel Phi 7120	2·8·2·8+2·61·4·16	256+7 808
BNL	USA	Intel E5-2680+Intel Phi 5110P	22·(2·12·2·8+2·60·4·16)	8 448+168 960

8 064

177 408

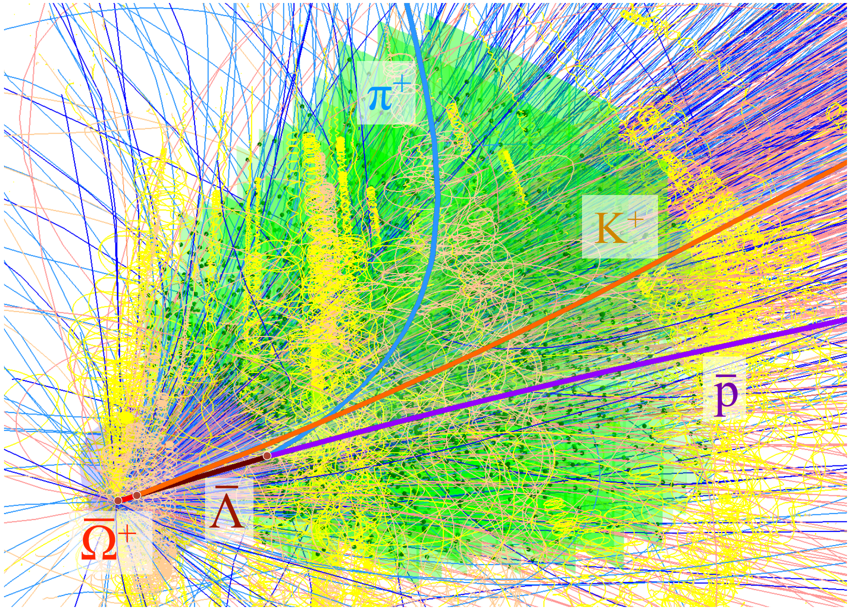
List of some heterogeneous HPC nodes, used in our investigations

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%

Parallelization becomes a standard in the CBM experiment

KF Particle: Reconstruction of 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
```

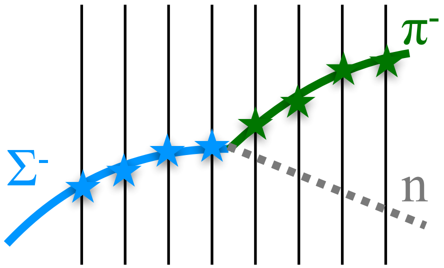
Concept:

- 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

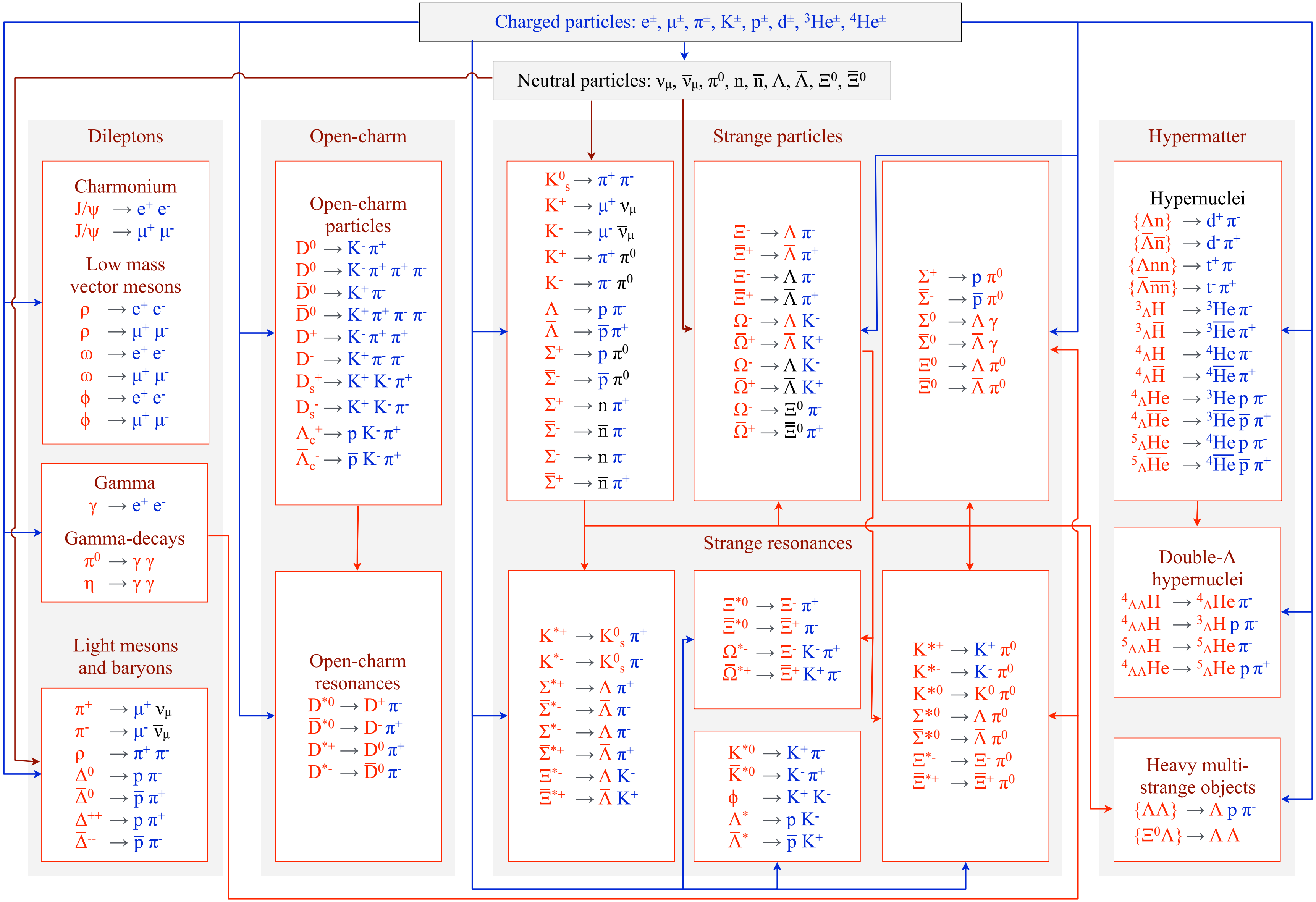
Reconstruction of decays with a neutral daughter
by the missing mass method:



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

3

KF Particle Finder for Physics Analysis and Selection



Charged particles: $e^\pm$, $\mu^\pm$, $\pi^\pm$, $K^\pm$, $p^\pm$, $d^\pm$, ${}^3He^\pm$, ${}^4He^\pm$

Neutral particles: ν_μ, $\bar{\nu}_\mu$, π^0, n, $\bar{n}$, Λ, $\bar{\Lambda}$, Ξ^0, $\bar{\Xi}^0$

Dileptons

Charmonium

$J/\psi \longrightarrow e^+ e^-$

$J/\psi \longrightarrow \mu^+ \mu^-$

Low mass vector mesons

$\rho \longrightarrow e^+ e^-$

$\rho \longrightarrow \mu^+ \mu^-$

$\omega \longrightarrow e^+ e^-$

$\omega \longrightarrow \mu^+ \mu^-$

$\phi \longrightarrow e^+ e^-$

$\phi \longrightarrow \mu^+ \mu^-$

Gamma

$\gamma \longrightarrow e^+ e^-$

Gamma-decays

$\pi^0 \longrightarrow \gamma\, \gamma$

$\eta \longrightarrow \gamma\, \gamma$

Light mesons and baryons

$\pi^+ \longrightarrow \mu^+ \nu_\mu$

$\pi^- \longrightarrow \mu^- \bar{\nu}_\mu$

$\rho \longrightarrow \pi^+ \pi^-$

$\Delta^0 \longrightarrow p\, \pi^-$

$\bar{\Delta}^0 \longrightarrow \bar{p}\, \pi^+$

$\Delta^{++} \longrightarrow p\, \pi^+$

$\bar{\Delta}^{--} \longrightarrow \bar{p}\, \pi^-$

Open-charm

Open-charm particles

$D^0 \longrightarrow K^- \pi^+$

$D^0 \longrightarrow K^- \pi^+ \pi^+ \pi^-$

$\bar{D}^0 \longrightarrow K^+ \pi^-$

$\bar{D}^0 \longrightarrow K^+ \pi^+ \pi^- \pi^-$

$D^+ \longrightarrow K^- \pi^+ \pi^+$

$D^- \longrightarrow K^+ \pi^- \pi^-$

$D_s^+ \longrightarrow K^+ K^- \pi^+$

$D_s^- \longrightarrow K^+ K^- \pi^-$

$\Lambda_c^+ \longrightarrow p\, K^- \pi^+$

$\bar{\Lambda}_c^- \longrightarrow \bar{p}\, K^- \pi^+$

Open-charm resonances

$D^{*0} \longrightarrow D^+ \pi^-$

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

$D^{*+} \longrightarrow D^0 \pi^+$

$D^{*-} \longrightarrow \bar{D}^0 \pi^-$

Strange particles

$K^0_s \longrightarrow \pi^+ \pi^-$

$K^+ \longrightarrow \mu^+ \nu_\mu$

$K^- \longrightarrow \mu^- \bar{\nu}_\mu$

$K^+ \longrightarrow \pi^+ \pi^0$

$K^- \longrightarrow \pi^- \pi^0$

$\Lambda \longrightarrow p\, \pi^-$

$\bar{\Lambda} \longrightarrow \bar{p}\, \pi^+$

$\Sigma^+ \longrightarrow p\, \pi^0$

$\bar{\Sigma}^- \longrightarrow \bar{p}\, \pi^0$

$\Sigma^+ \longrightarrow n\, \pi^+$

$\bar{\Sigma}^- \longrightarrow \bar{n}\, \pi^-$

$\Sigma^- \longrightarrow n\, \pi^-$

$\bar{\Sigma}^+ \longrightarrow \bar{n}\, \pi^+$

$\Xi^- \longrightarrow \Lambda\, \pi^-$

$\bar{\Xi}^+ \longrightarrow \bar{\Lambda}\, \pi^+$

$\Xi^- \longrightarrow \Lambda\, \pi^-$

$\bar{\Xi}^+ \longrightarrow \bar{\Lambda}\, \pi^+$

$\Omega^- \longrightarrow \Lambda\, K^-$

$\bar{\Omega}^+ \longrightarrow \bar{\Lambda}\, K^+$

$\Omega^- \longrightarrow \Lambda\, K^-$

$\bar{\Omega}^+ \longrightarrow \bar{\Lambda}\, K^+$

$\Omega^- \longrightarrow \Xi^0\, \pi^-$

$\bar{\Omega}^+ \longrightarrow \bar{\Xi}^0\, \pi^+$

$\Sigma^+ \longrightarrow p\, \pi^0$

$\bar{\Sigma}^- \longrightarrow \bar{p}\, \pi^0$

$\Sigma^0 \longrightarrow \Lambda\, \gamma$

$\bar{\Sigma}^0 \longrightarrow \bar{\Lambda}\, \gamma$

$\Xi^0 \longrightarrow \Lambda\, \pi^0$

$\bar{\Xi}^0 \longrightarrow \bar{\Lambda}\, \pi^0$

Strange resonances

$K^{*+} \longrightarrow K^0_s\, \pi^+$

$K^{*-} \longrightarrow K^0_s\, \pi^-$

$\Sigma^{*+} \longrightarrow \Lambda\, \pi^+$

$\bar{\Sigma}^{*-} \longrightarrow \bar{\Lambda}\, \pi^-$

$\Sigma^{*-} \longrightarrow \Lambda\, \pi^-$

$\bar{\Sigma}^{*+} \longrightarrow \bar{\Lambda}\, \pi^+$

$\Xi^{*-} \longrightarrow \Lambda\, K^-$

$\bar{\Xi}^{*+} \longrightarrow \bar{\Lambda}\, K^+$

$\Xi^{*0} \longrightarrow \Xi^-\, \pi^+$

$\bar{\Xi}^{*0} \longrightarrow \bar{\Xi}^+\, \pi^-$

$\Omega^{*-} \longrightarrow \Xi^-\, K^- \pi^+$

$\bar{\Omega}^{*+} \longrightarrow \bar{\Xi}^+\, K^+ \pi^-$

$K^{*0} \longrightarrow K^+ \pi^-$

$\bar{K}^{*0} \longrightarrow K^- \pi^+$

$\phi \longrightarrow K^+ K^-$

$\Lambda^* \longrightarrow p\, K^-$

$\bar{\Lambda}^* \longrightarrow \bar{p}\, K^+$

$K^{*+} \longrightarrow K^+ \pi^0$

$K^{*-} \longrightarrow K^- \pi^0$

$K^{*0} \longrightarrow K^0 \pi^0$

$\Sigma^{*0} \longrightarrow \Lambda\, \pi^0$

$\bar{\Sigma}^{*0} \longrightarrow \bar{\Lambda}\, \pi^0$

$\Xi^{*-} \longrightarrow \Xi^-\, \pi^0$

$\bar{\Xi}^{*+} \longrightarrow \bar{\Xi}^+\, \pi^0$

Hypermatter

Hypernuclei

$\{\Lambda n\} \longrightarrow d^+ \pi^-$

$\{\bar{\Lambda}\bar{n}\} \longrightarrow d^- \pi^+$

$\{\Lambda nn\} \longrightarrow t^+ \pi^-$

$\{\bar{\Lambda}\overline{nn}\} \longrightarrow t^- \pi^+$

${}^3_\Lambda H \longrightarrow {}^3He\, \pi^-$

${}^3_\Lambda \bar{H} \longrightarrow {}^3\overline{He}\, \pi^+$

${}^4_\Lambda H \longrightarrow {}^4He\, \pi^-$

${}^4_\Lambda \bar{H} \longrightarrow {}^4\overline{He}\, \pi^+$

${}^4_\Lambda He \longrightarrow {}^3He\, p\, \pi^-$

${}^4_\Lambda \overline{He} \longrightarrow {}^3\overline{He}\, \bar{p}\, \pi^+$

${}^5_\Lambda He \longrightarrow {}^4He\, p\, \pi^-$

${}^5_\Lambda \overline{He} \longrightarrow {}^4\overline{He}\, \bar{p}\, \pi^+$

Double-Λ hypernuclei

${}^4_{\Lambda\Lambda} H \longrightarrow {}^4_\Lambda He\, \pi^-$

${}^4_{\Lambda\Lambda} H \longrightarrow {}^3_\Lambda H\, p\, \pi^-$

${}^5_{\Lambda\Lambda} H \longrightarrow {}^5_\Lambda He\, \pi^-$

${}^4_{\Lambda\Lambda} He \longrightarrow {}^5_\Lambda He\, p\, \pi^+$

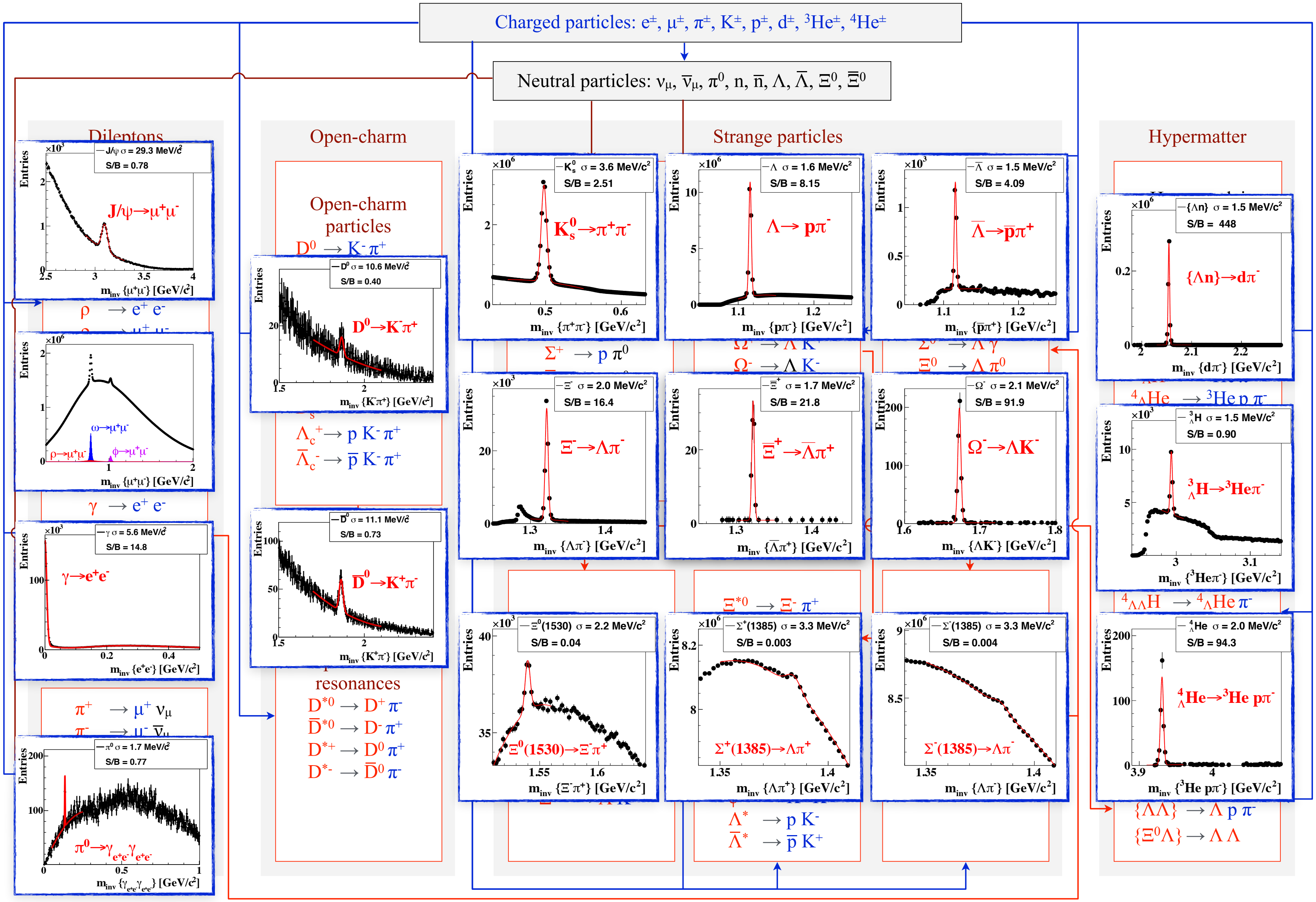
Heavy multi-strange objects

$\{\Lambda\Lambda\} \longrightarrow \Lambda\, p\, \pi^-$

$\{\Xi^0\Lambda\} \longrightarrow \Lambda\, \Lambda$

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

KF Particle Finder for Physics Analysis and Selection



Charged particles: $e^\pm$, $\mu^\pm$, $\pi^\pm$, $K^\pm$, $p^\pm$, $d^\pm$, $^3He^\pm$, $^4He^\pm$

Neutral particles: ν_μ, $\bar{\nu}_\mu$, π^0, n, $\bar{n}$, Λ, $\bar{\Lambda}$, Ξ^0, $\bar{\Xi}^0$

Dileptons

$\rho \rightarrow e^+ e^-$

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

$\pi^+ \rightarrow \mu^+ \nu_\mu$

Open-charm

Open-charm particles

$D^0 \rightarrow K^- \pi^+$

$\Lambda_c^+ \rightarrow p K^- \pi^+$

$\bar{\Lambda}_c^- \rightarrow \bar{p} K^- \pi^+$

resonances

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

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

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

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

Strange particles

$\Sigma^+ \rightarrow p \pi^0$

$\Omega^- \rightarrow \Lambda K^-$

$\Xi^0 \rightarrow \Lambda \pi^0$

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

$\Lambda^* \rightarrow p K^-$

$\bar{\Lambda}^* \rightarrow \bar{p} K^+$

Hypermatter

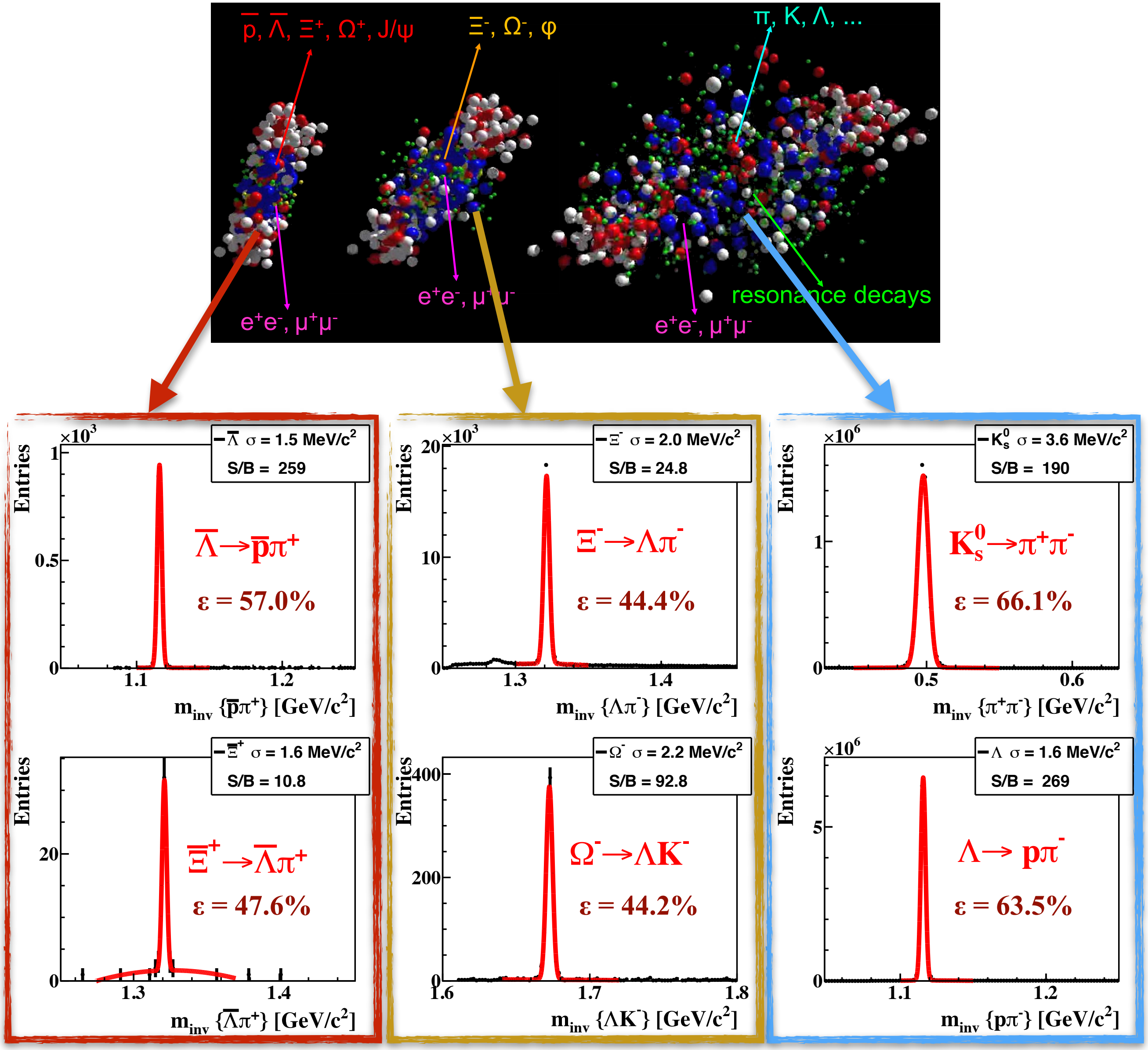
$^4_\Lambda He \rightarrow {}^3He\, p\, \pi^-$

$^4_{\Lambda\Lambda}H \rightarrow {}^4_\Lambda He\, \pi^-$

$\{\Lambda\Lambda\} \rightarrow \Lambda p \pi^-$

$\{\Xi^0\Lambda\} \rightarrow \Lambda \Lambda$

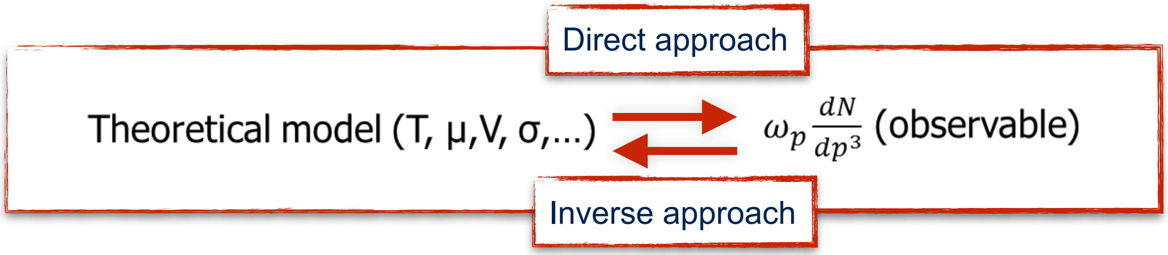
Clean Probes of Collision Stages



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

CBM Online Physics Analysis

Online physics analysis = online extraction of medium properties in heavy-ion collisions



Direct approach

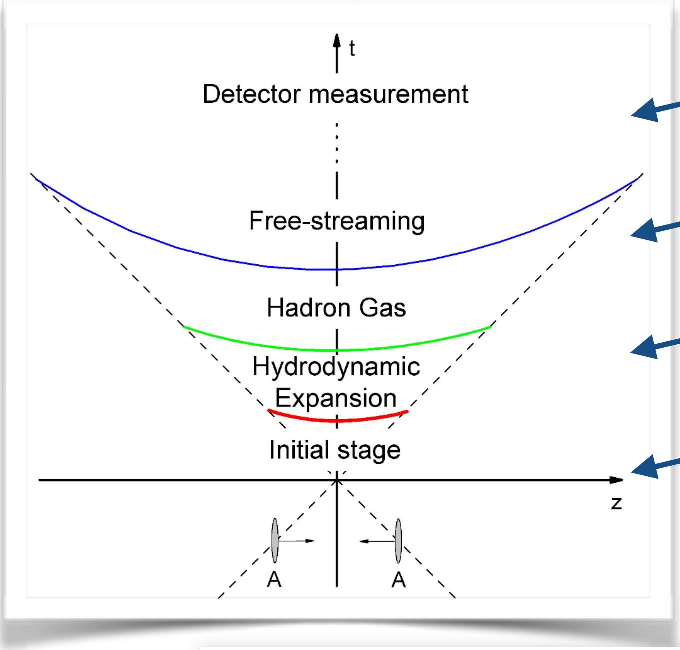
Theoretical model (T, μ,V, σ,…) $\rightleftarrows$ $\omega_p \frac{dN}{dp^3}$ (observable)

Inverse approach

Motivation:

- determination of physical properties of QCD matter created in HIC (temperature, flow, phase transitions, …),
- obtain limits of applicability of different models

Stages of collision



Models for different stages

Final momentum spectrum (Blast-Wave, Tsallis, …)

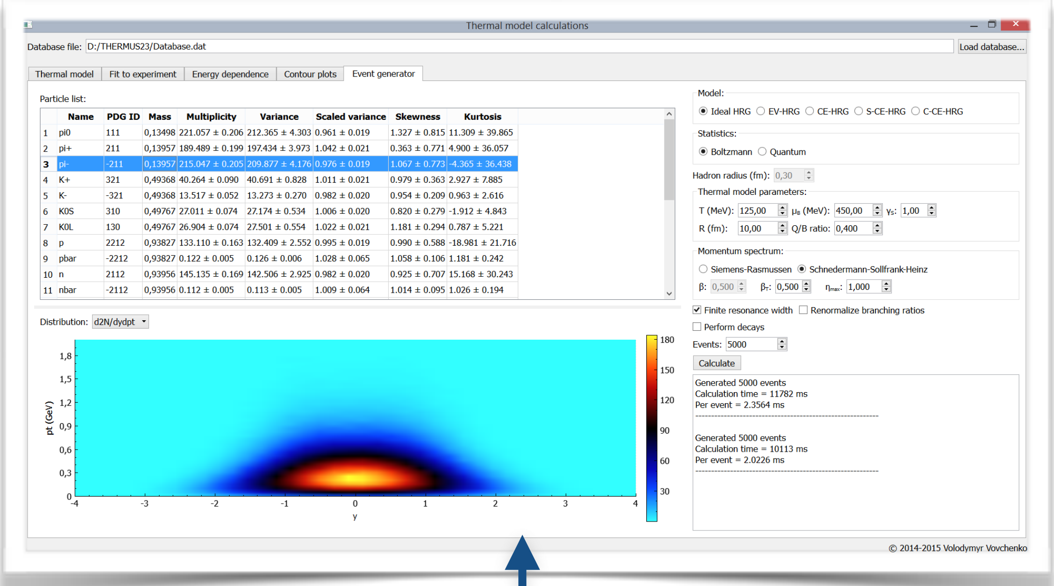
Statistical-thermal models for chemical freeze-out (ideal hadron gas, Van der Waals hadron gas, Hagedorn states, …)

Relativistic hydrodynamics (ideal, viscous; (0+1)D, (1+1)D, (3+1)D, …)

Initial stage (Glauber, CGC, …)

A package to extract the parameters of theoretical models in CBM experiment is implemented

CBM Online Physics Analysis



Thermal model calculations

Database file: D:/THERMUS23/Database.dat

Particle list:

	Name	PDG ID	Mass	Multiplicity	Variance	Scaled variance	Skewness	Kurtosis
1	pi0	111	0,13498	221.057 ± 0.206	212.365 ± 4.303	0.961 ± 0.019	1.327 ± 0.815	11.309 ± 39.865
2	pi+	211	0,13957	189.489 ± 0.199	197.434 ± 3.973	1.042 ± 0.021	0.363 ± 0.771	4.900 ± 36.057
3	pi-	-211	0,13957	215.047 ± 0.205	209.877 ± 4.176	0.976 ± 0.019	1.067 ± 0.773	-4.365 ± 36.438
4	K+	321	0,49368	40.264 ± 0.090	40.691 ± 0.828	1.011 ± 0.021	0.979 ± 0.363	2.927 ± 7.885
5	K-	-321	0,49368	13.517 ± 0.052	13.273 ± 0.270	0.982 ± 0.020	0.954 ± 0.209	0.963 ± 2.616
6	K0S	310	0,49767	27.011 ± 0.074	27.174 ± 0.534	1.006 ± 0.020	0.820 ± 0.279	-1.912 ± 4.843
7	K0L	130	0,49767	26.904 ± 0.074	27.501 ± 0.554	1.022 ± 0.021	1.181 ± 0.294	0.787 ± 5.221
8	p	2212	0,93827	133.110 ± 0.163	132.409 ± 2.552	0.995 ± 0.019	0.990 ± 0.588	-18.981 ± 21.716
9	pbar	-2212	0,93827	0.122 ± 0.005	0.126 ± 0.006	1.028 ± 0.065	1.058 ± 0.106	1.181 ± 0.242
10	n	2112	0,93956	145.135 ± 0.169	142.506 ± 2.925	0.982 ± 0.020	0.925 ± 0.707	15.168 ± 30.243
11	nbar	-2112	0,93956	0.112 ± 0.005	0.113 ± 0.005	1.009 ± 0.064	1.014 ± 0.095	1.026 ± 0.194

Distribution: d2N/dydpt

Model:
● Ideal HRG ○ EV-HRG ○ CE-HRG ○ S-CE-HRG ○ C-CE-HRG

Statistics:
● Boltzmann ○ Quantum

Hadron radius (fm): 0,30

Thermal model parameters:
T (MeV): 125,00 μ_B (MeV): 450,00 γ_S: 1,00
R (fm): 10,00 Q/B ratio: 0,400

Momentum spectrum:
○ Siemens-Rasmussen ● Schnedermann-Sollfrank-Heinz
β: 0,500 β_T: 0,500 η_{max}: 1,000

☑ Finite resonance width ☐ Renormalize branching ratios

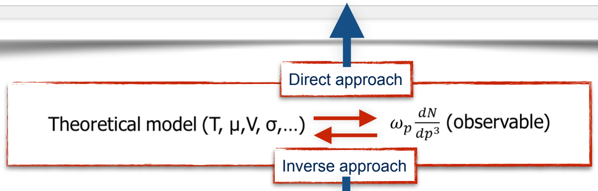
☐ Perform decays

Events: 5000

```
Generated 5000 events
Calculation time = 11782 ms
Per event = 2.3564 ms
-----------------------------------------------------

Generated 5000 events
Calculation time = 10113 ms
Per event = 2.0226 ms
-----------------------------------------------------
```

© 2014-2015 Volodymyr Vovchenko

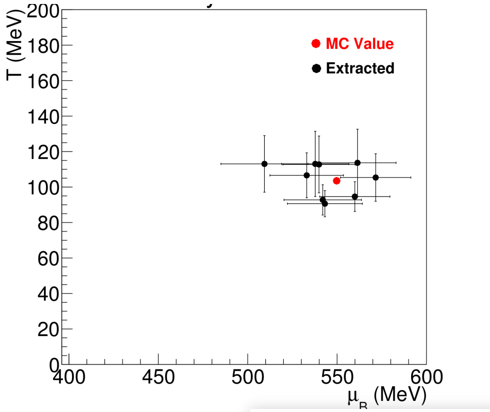


Direct approach

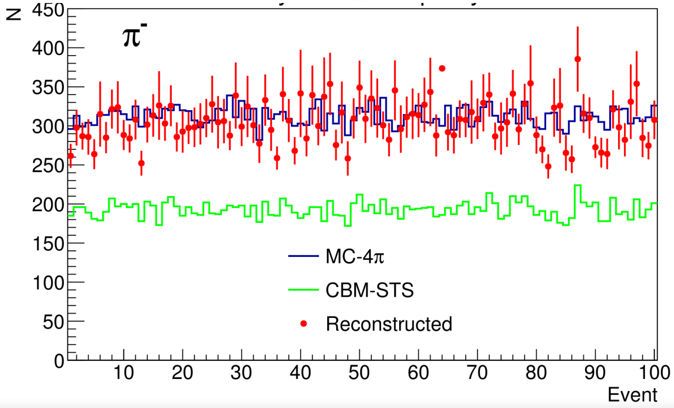
Theoretical model (T, μ,V, σ,…) ⇄ $\omega_p \frac{dN}{dp^3}$ (observable)

Inverse approach

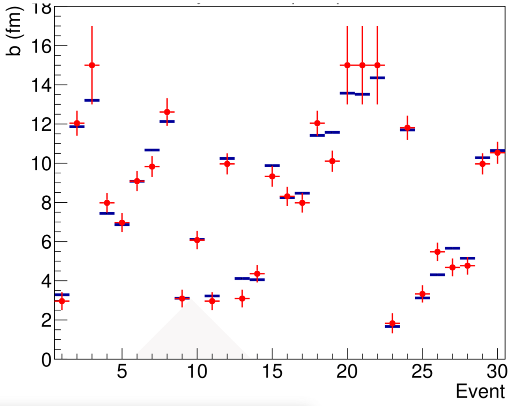
E.-by-E. extraction of T and μ_B (HRG)



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

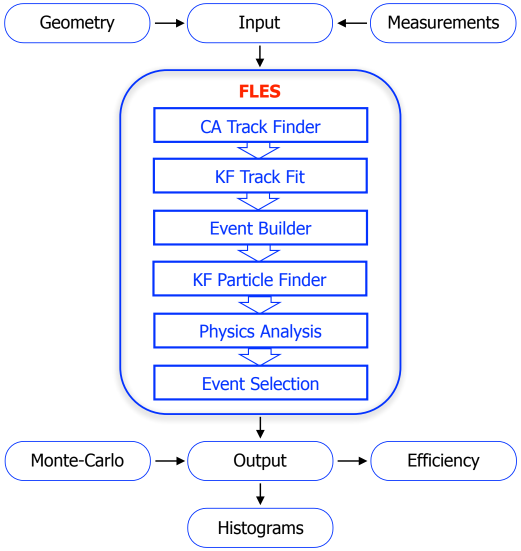


E.-by-E. impact parameter (Glauber)



A package to extract the parameters of theoretical models in CBM experiment is implemented

CBM Standalone First Level Event Selection (FLES) Package


Geometry → Input ← Measurements

FLES

CA Track Finder

KF Track Fit

Event Builder

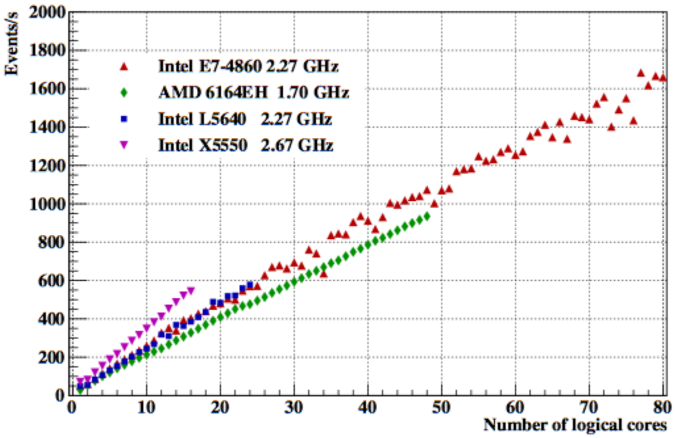
KF Particle Finder

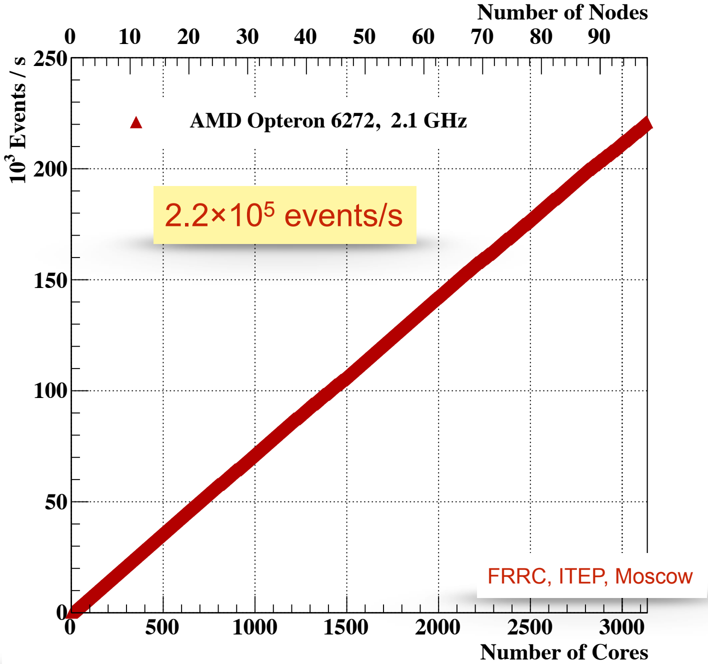
Physics Analysis

Event Selection

Monte-Carlo → Output → Efficiency

Histograms





2.2×10^5 events/s

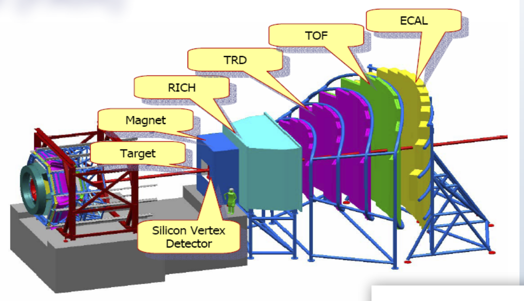
FRRC, ITEP, Moscow

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

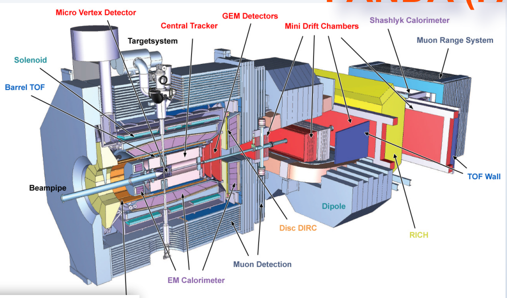
Summary: to be HPC Efficient - Consolidate Efforts

Common
Reconstruction and Analysis
Package

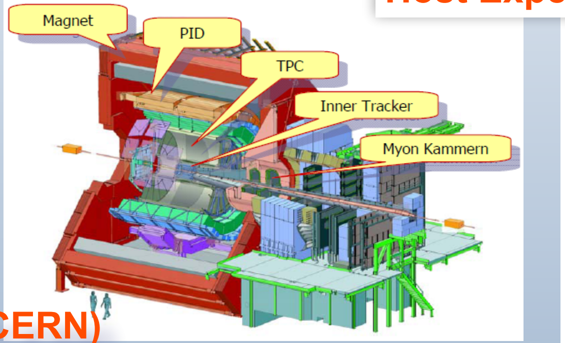
CBM (FAIR)



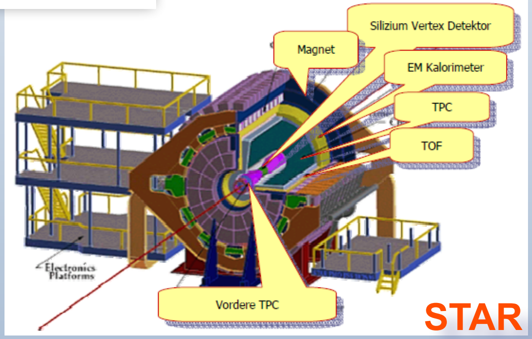
PANDA (FAIR)



Host Experiments



ALICE (CERN)



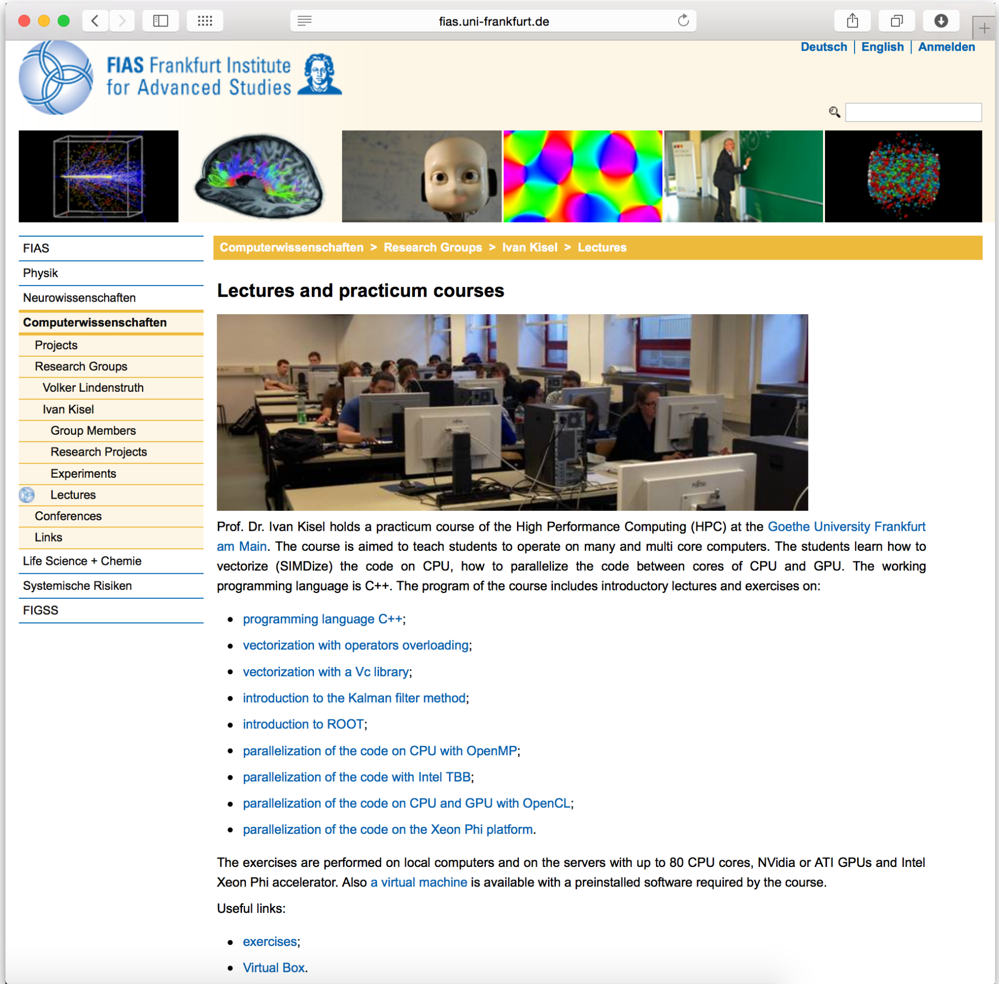
STAR (BNL)

Consolidate efforts of:

- Physicists
- Mathematicians
- Computer scientists
- Developers of // languages
- Many-core CPU/GPU manufacturers

HPC Practical Course at the Goethe University Frankfurt

http://fias.uni-frankfurt.de/de/cs/kisel/lectures/



- FIAS
- Physik
- Neurowissenschaften
- **Computerwissenschaften**
  - Projects
  - Research Groups
    - Volker Lindenstruth
    - Ivan Kisel
      - Group Members
      - Research Projects
      - Experiments
      - Lectures
    - Conferences
    - Links
- Life Science + Chemie
- Systemische Risiken
- FIGSS

Computerwissenschaften > Research Groups > Ivan Kisel > Lectures

Lectures and practicum courses

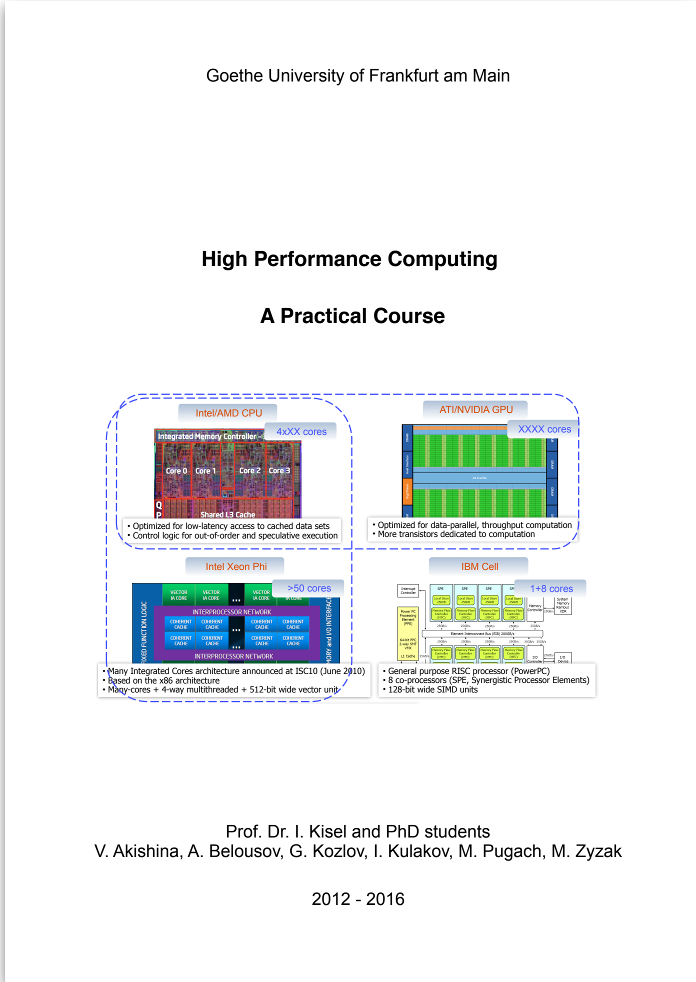
Prof. Dr. Ivan Kisel holds a practicum course of the High Performance Computing (HPC) at the Goethe University Frankfurt am Main. The course is aimed to teach students to operate on many and multi core computers. The students learn how to vectorize (SIMDize) the code on CPU, how to parallelize the code between cores of CPU and GPU. The working programming language is C++. The program of the course includes introductory lectures and exercises on:

- programming language C++;
- vectorization with operators overloading;
- vectorization with a Vc library;
- introduction to the Kalman filter method;
- introduction to ROOT;
- parallelization of the code on CPU with OpenMP;
- parallelization of the code with Intel TBB;
- parallelization of the code on CPU and GPU with OpenCL;
- parallelization of the code on the Xeon Phi platform.

The exercises are performed on local computers and on the servers with up to 80 CPU cores, NVidia or ATI GPUs and Intel Xeon Phi accelerator. Also a virtual machine is available with a preinstalled software required by the course.

Useful links:

- exercises;
- Virtual Box.



Goethe University of Frankfurt am Main

High Performance Computing

A Practical Course

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

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

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

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

Prof. Dr. I. Kisel and PhD students
V. Akishina, A. Belousov, G. Kozlov, I. Kulakov, M. Pugach, M. Zyzak

2012 - 2016

Conclusion

The future is parallel.

The future is parallel.
The future is parallel.
The future is parallel.
The future is parallel.
The future is parallel.