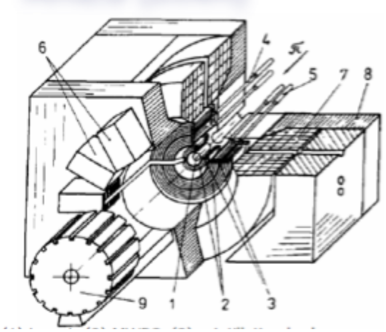
Real-time event reconstruction with Cellular Automaton and KF Particle Finder

Ivan Kisel

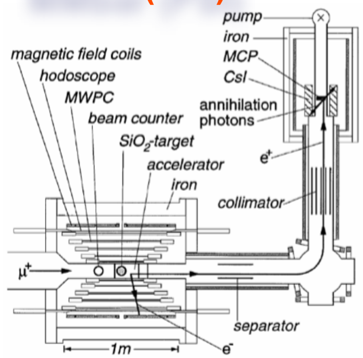
Goethe University Frankfurt am Main
FIAS Frankfurt Institute for Advanced Studies
GSI Helmholtz Center for Heavy Ion Research

Cellular Automaton for Tracking since 1990

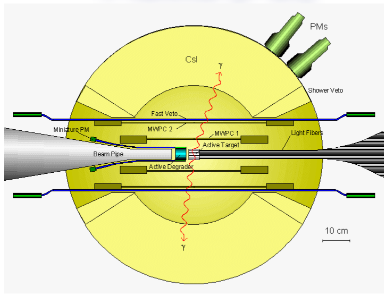
ARES (JINR)



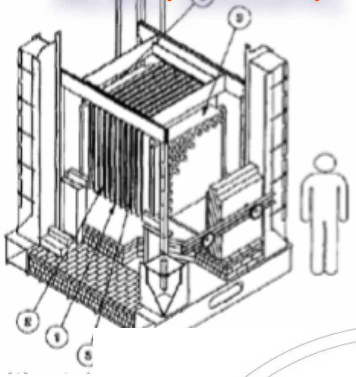
MMbar (PSI)



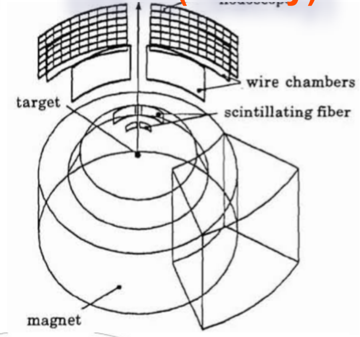
PiBeta (PSI)



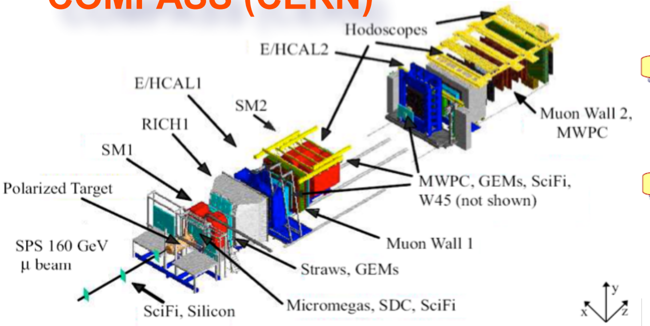
NEMO (Modane)



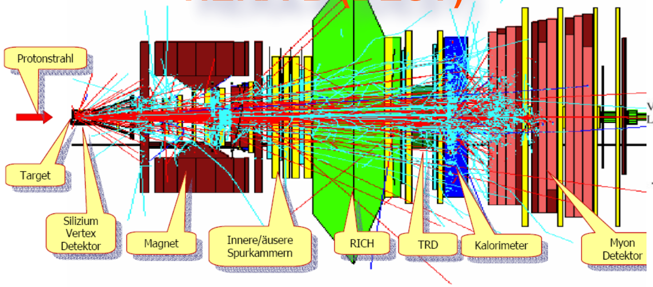
DISTO (Saclay)



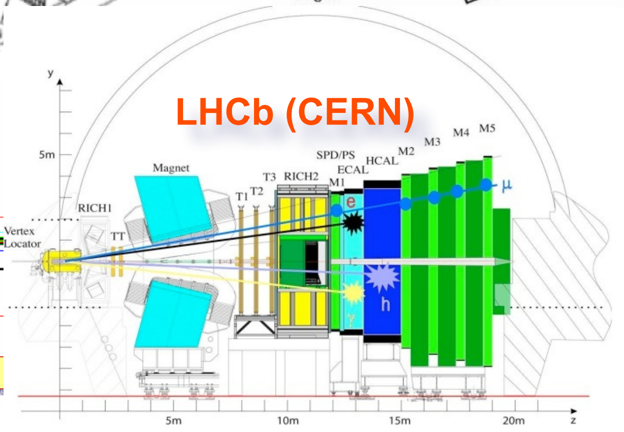
COMPASS (CERN)



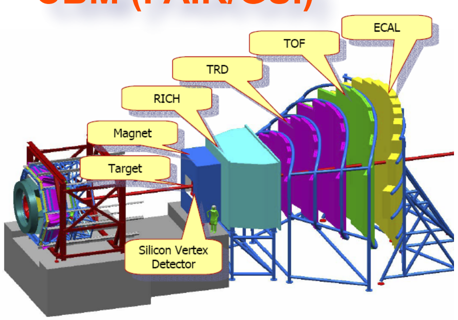
HERA-B (DESY)



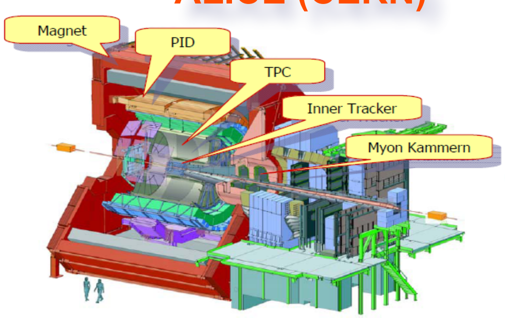
LHCb (CERN)



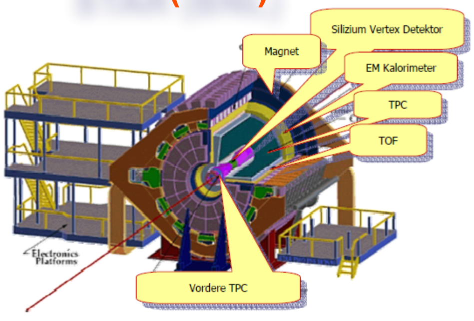
CBM (FAIR/GSI)



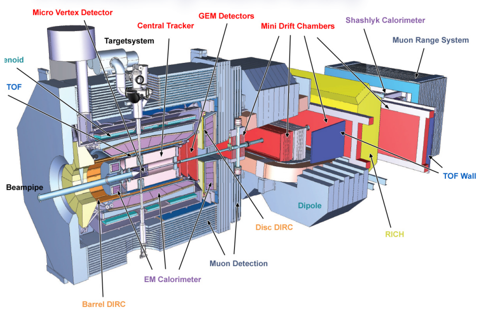
ALICE (CERN)



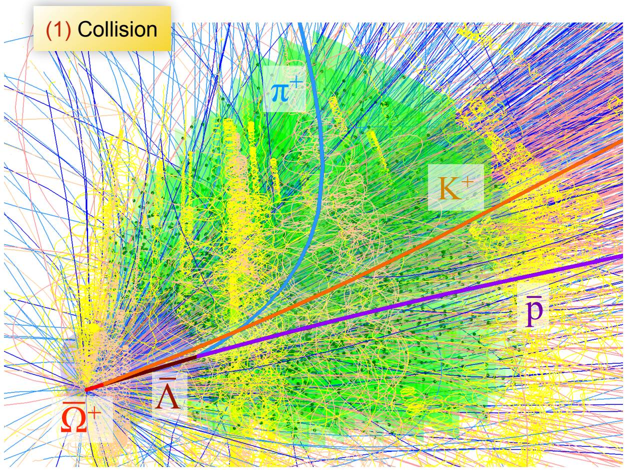
STAR (BNL)

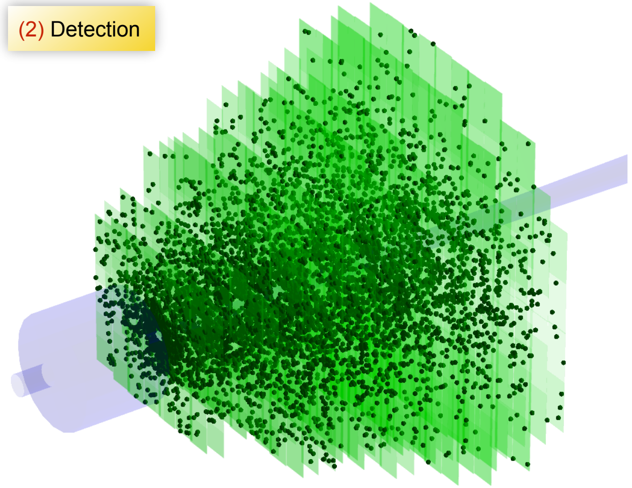


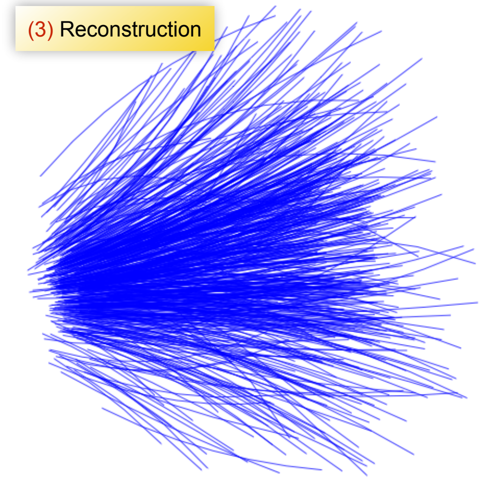
PANDA (FAIR/GSI)



Reconstruction Challenge in CBM





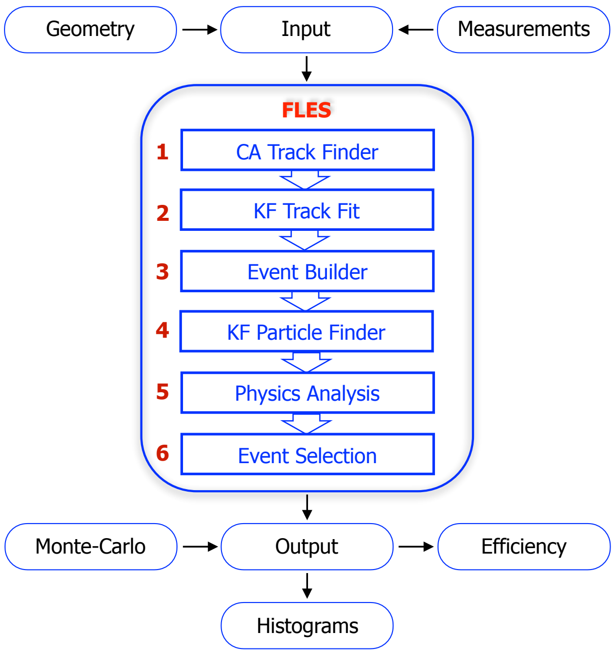


- Future fixed-target heavy-ion experiment at FAIR
- Explore the phase diagram at high net-baryon densities
- 10^7 Au+Au collisions/sec
- ~ 1000 charged particles/collision
- Non-homogeneous magnetic field
- Double-sided strip detectors
- 4D reconstruction of time slices.

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

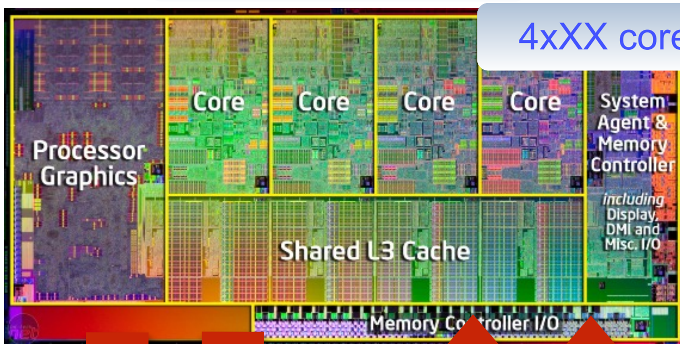
All reconstruction algorithms are vectorized and parallelized.



Many-Core CPU/GPU Architectures

Intel/AMD CPU

4xXX cores



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

Math

Memory

Parallelism

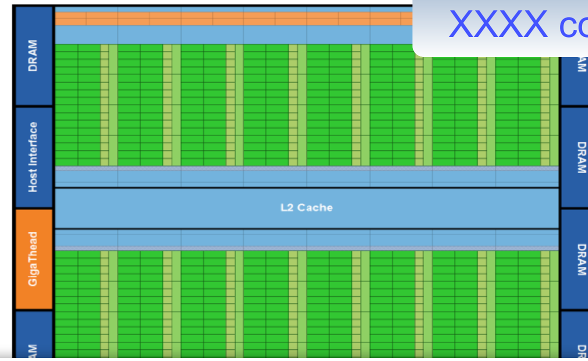
Math

Memory

#Cores

Nvidia/ATI GPU

XXXX cores



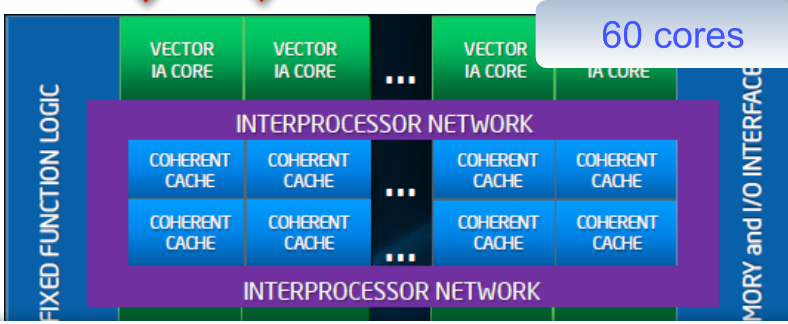
- ptimized for data-parallel, throughput computation
- lore transistors dedicated to computation

Stability

Memory

Intel Phi

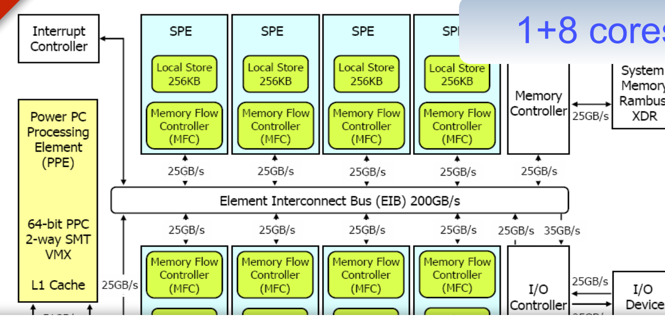
60 cores



- 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

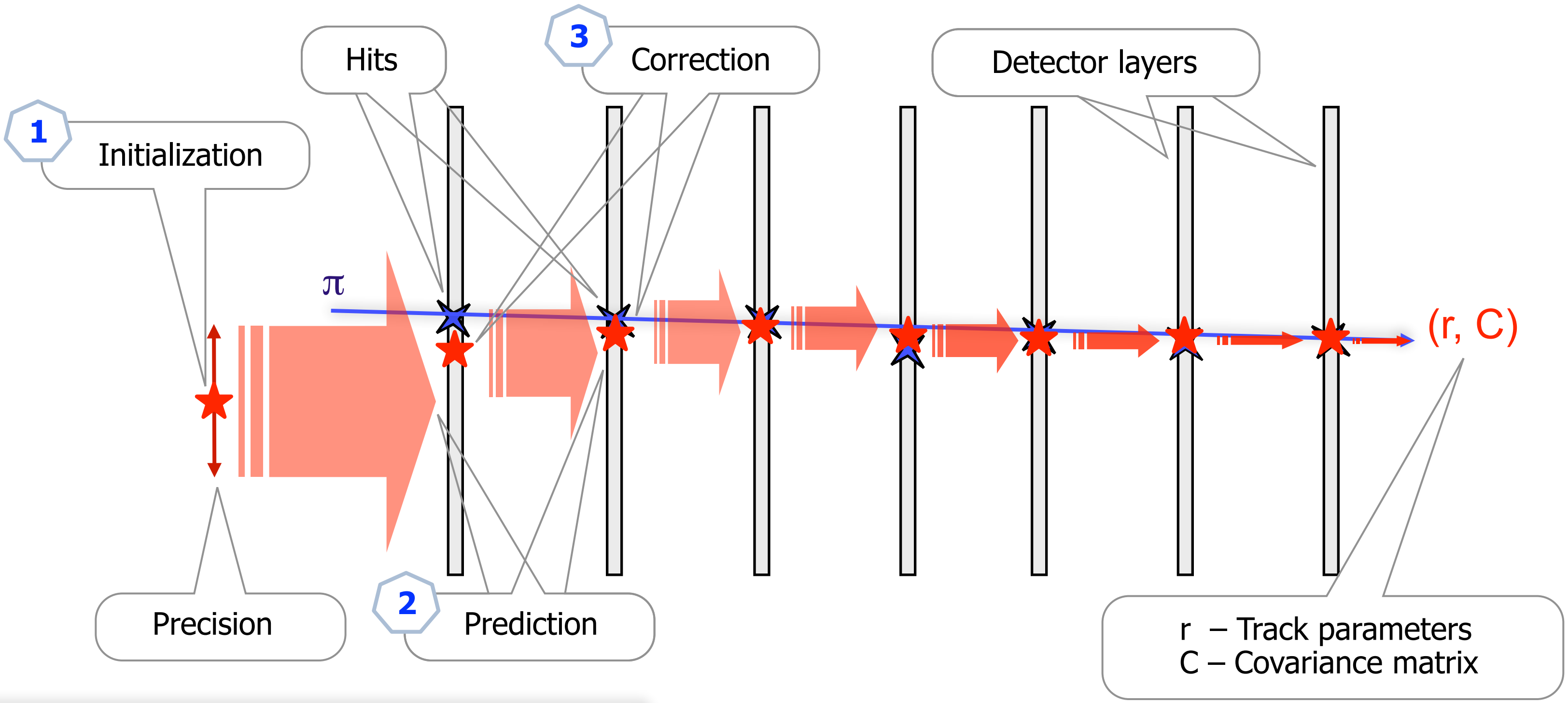


- General purpose RISC processor (PowerPC)
- 8 co-processors (SPE, Synergistic Processor Elements)
- 128-bit wide SIMD units

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

Kalman Filter (KF) based Track Fit

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

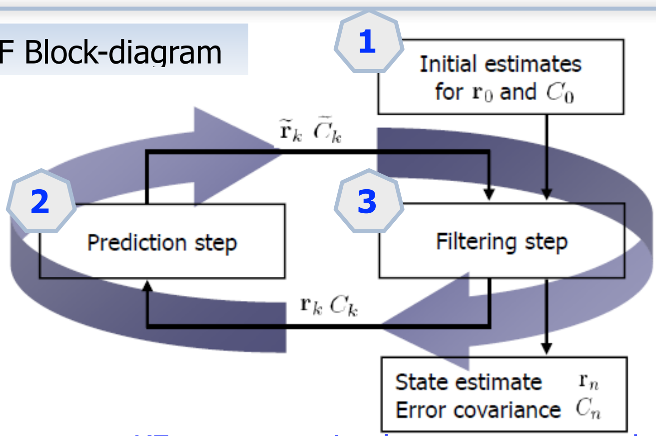


State vector

Position, direction and momentum

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

KF Block-diagram



KF as a recursive least squares method

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

Nowadays the Kalman Filter is used in almost all HEP experiments

Kalman Filter Track Fit on Cell

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

Intel (stages 1–3)

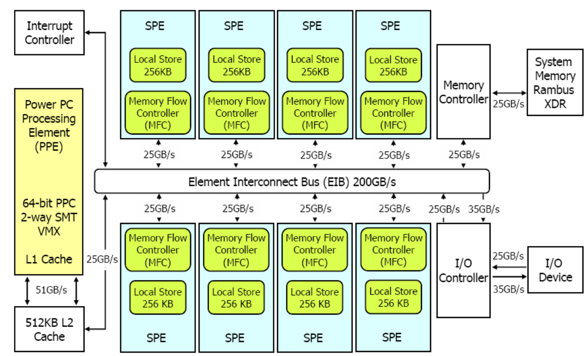
Cell (stages 4–5)

10000x faster on any PC

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

The KF speed was increased by 5 orders of magnitude

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



Motivated by, but not restricted to Cell !

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

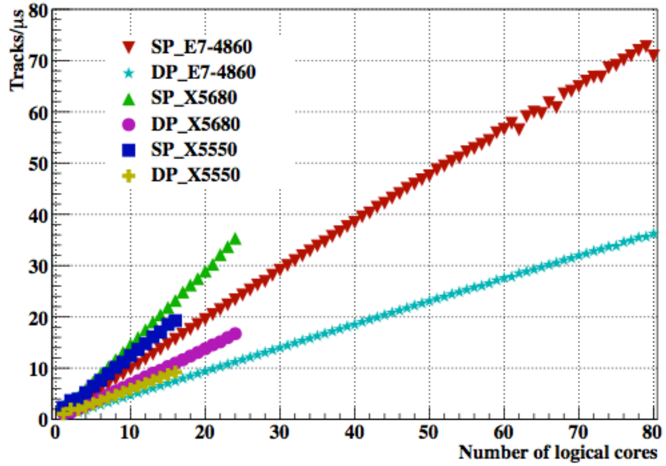
Implementations

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

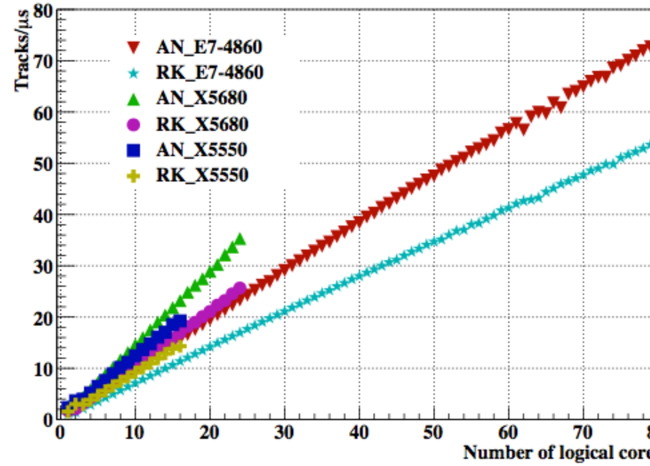
Parallelization (many-cores):
- Open MP
- ITBB
- ArBB
- OpenCL

Precision:
- single precision SP
- double precision DP

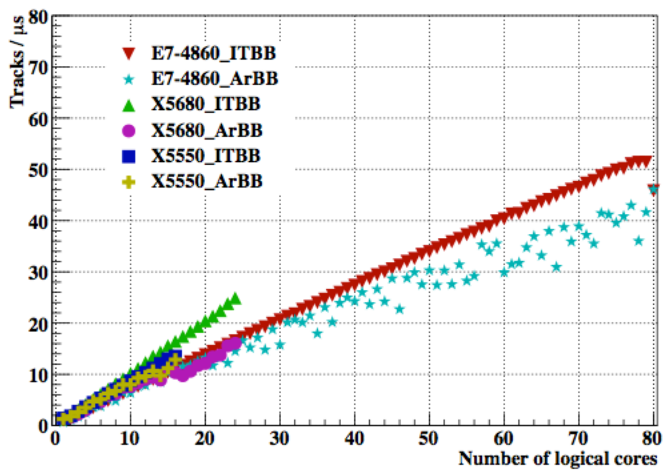
Conventional KF DP vs. SP



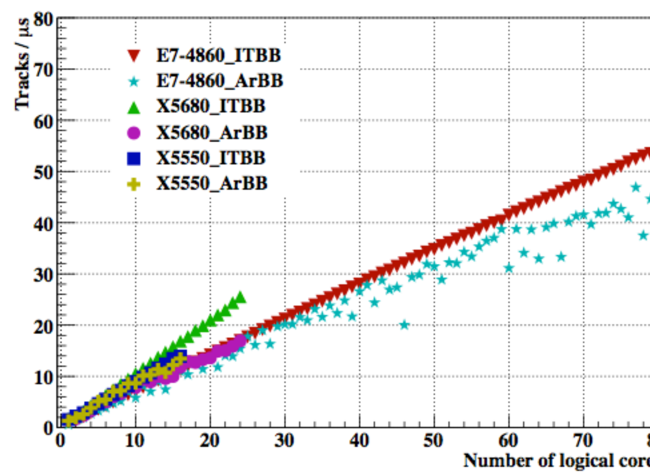
Conventional KF RK4 vs. Analytical



Square-Root KF



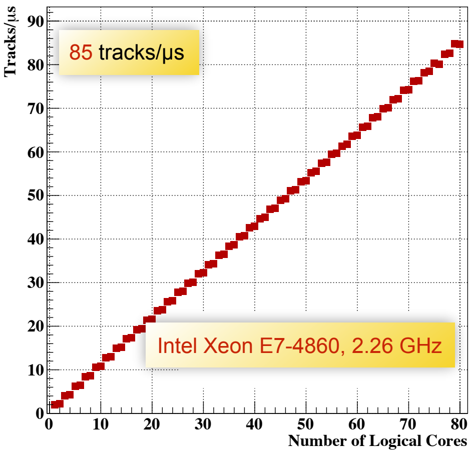
UD KF

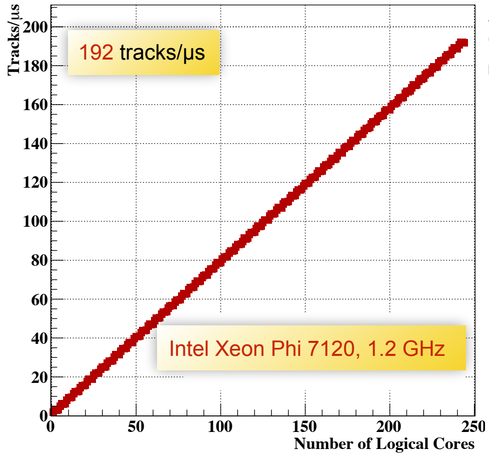


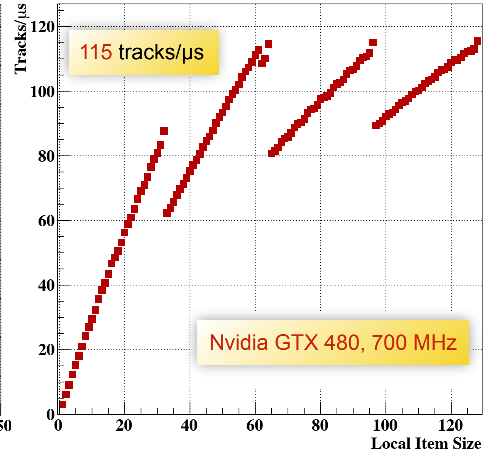
Strong many-core scalability of the Kalman filter library

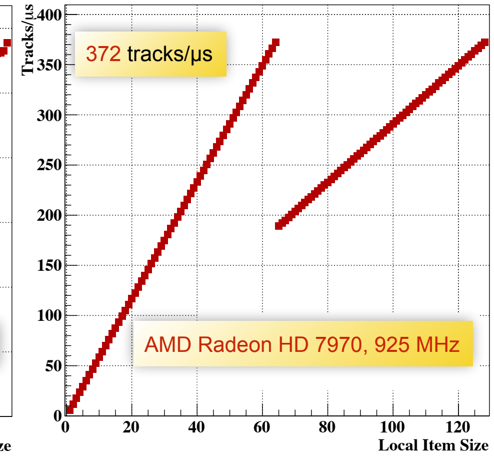
with I. Kulakov, H. Pabst* and M. Zyzak (*Intel)

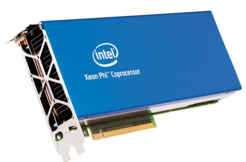
Kalman Filter (KF) Track Fit







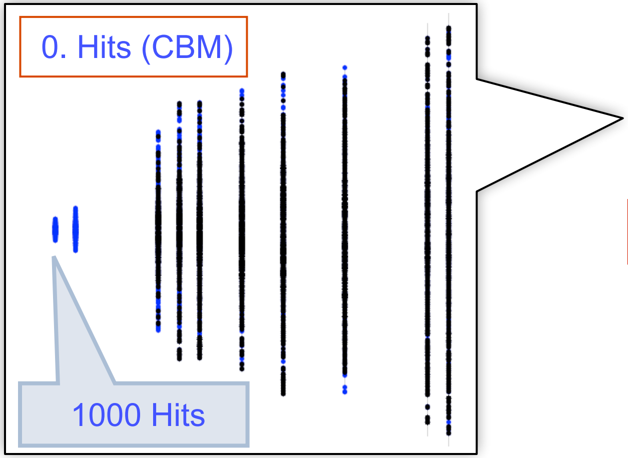


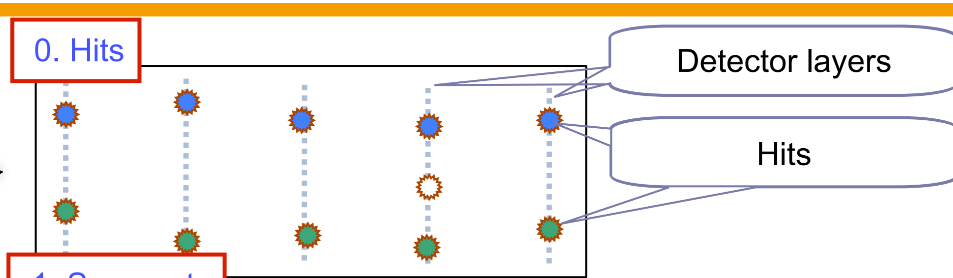


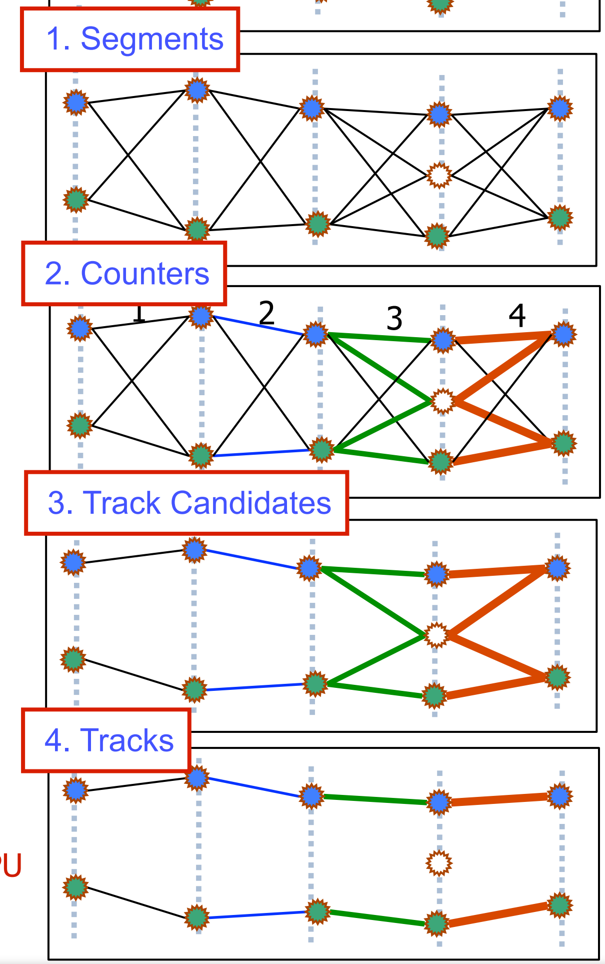
- Precise estimation of the parameters of particle trajectories is the core of the reconstruction procedure.
- Scalability with respect to the number of logical cores in a CPU is one of the most important parameters of the algorithm.
- The scalability on the Intel Xeon Phi coprocessor is similar to the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

The fastest implementation of the Kalman filter in the world

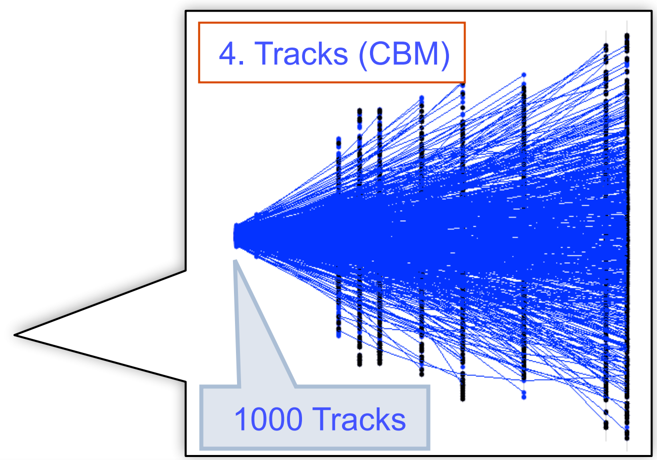
Cellular Automaton (CA) Track Finder







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.



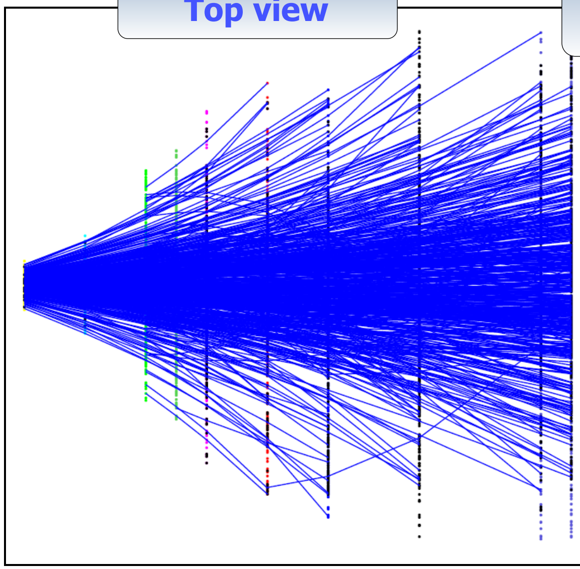
Cellular Automaton:
- local w.r.t. data
- no need in navigation
- intrinsically parallel
- extremely simple
- very fast

Deeply appropriate for many-core CPU/GPU

Useful for complicated event topologies with heavy combinatorics

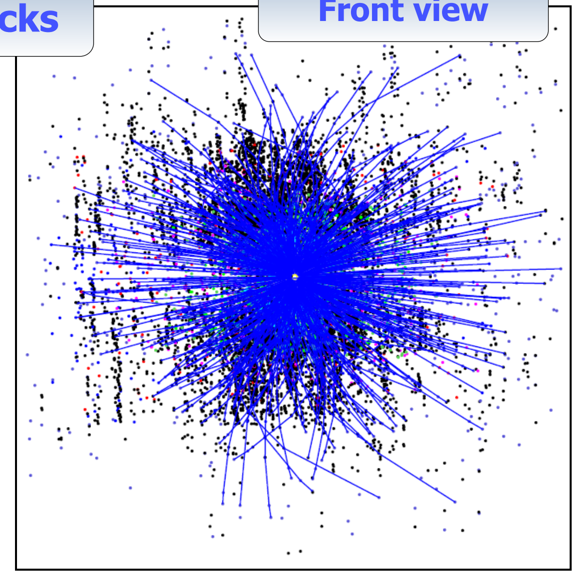
Cellular Automaton (CA) Track Finder

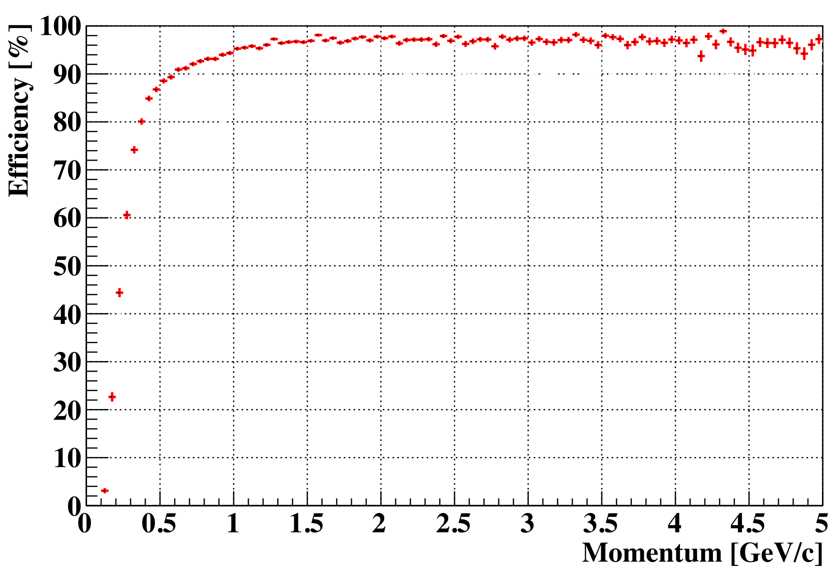
Top view



770 Tracks

Front view



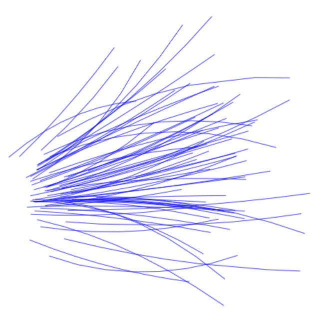


Track category	Eff, %
All tracks	90.9
Primary high-p	97.5
Primary low-p	92.6
Secondary high-p	91.1
Secondary low-p	63.8
Clone level	0.4
Ghost level	5.9
MC tracks found	134
Time, ms/ev	10

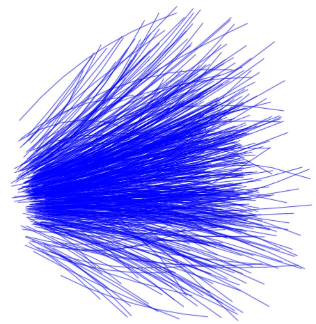
Fast and efficient track finder

CA Track Finder at High Track Multiplicity

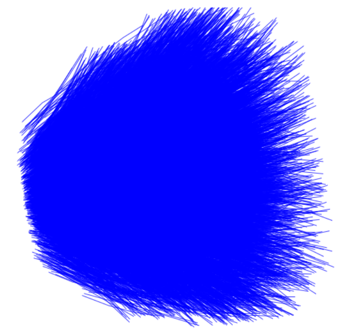
A number of minimum bias events is gathered into a group (super-event), which is then treated by the CA track finder as a single event.



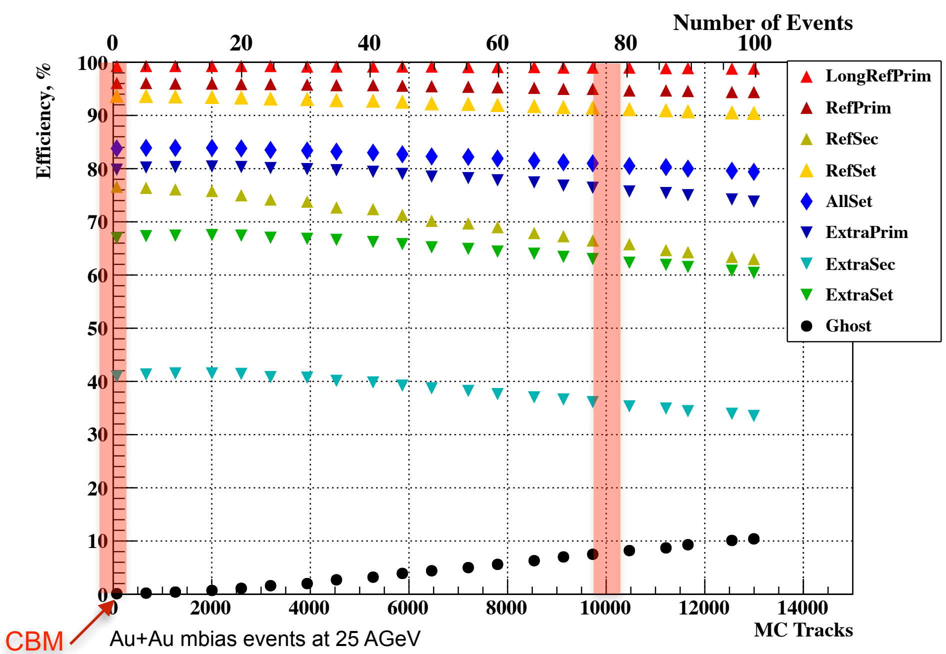
1 mbias event, $\langle N_{reco} \rangle$ = 109

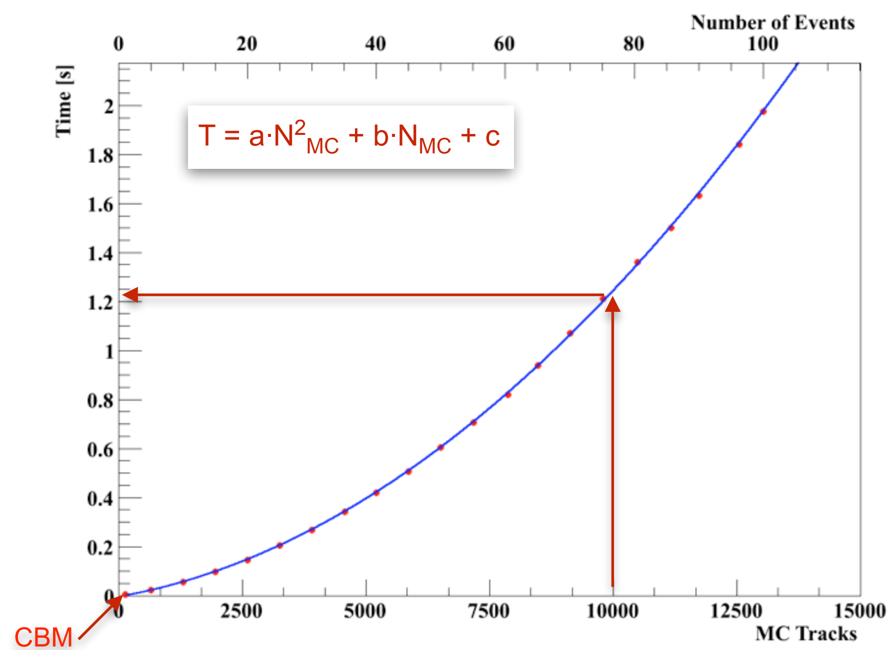


5 mbias events, $\langle N_{reco} \rangle$ = 572



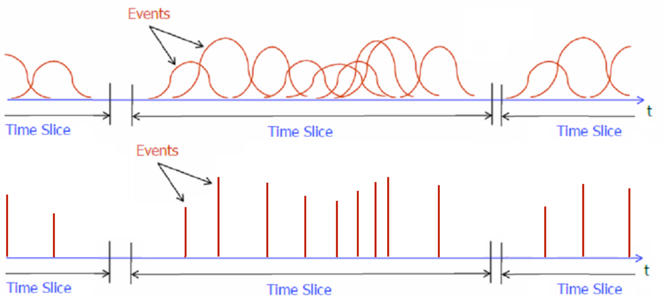
100 mbias events, $\langle N_{reco} \rangle$ = 10340





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

Time-based (4D) Track Reconstruction

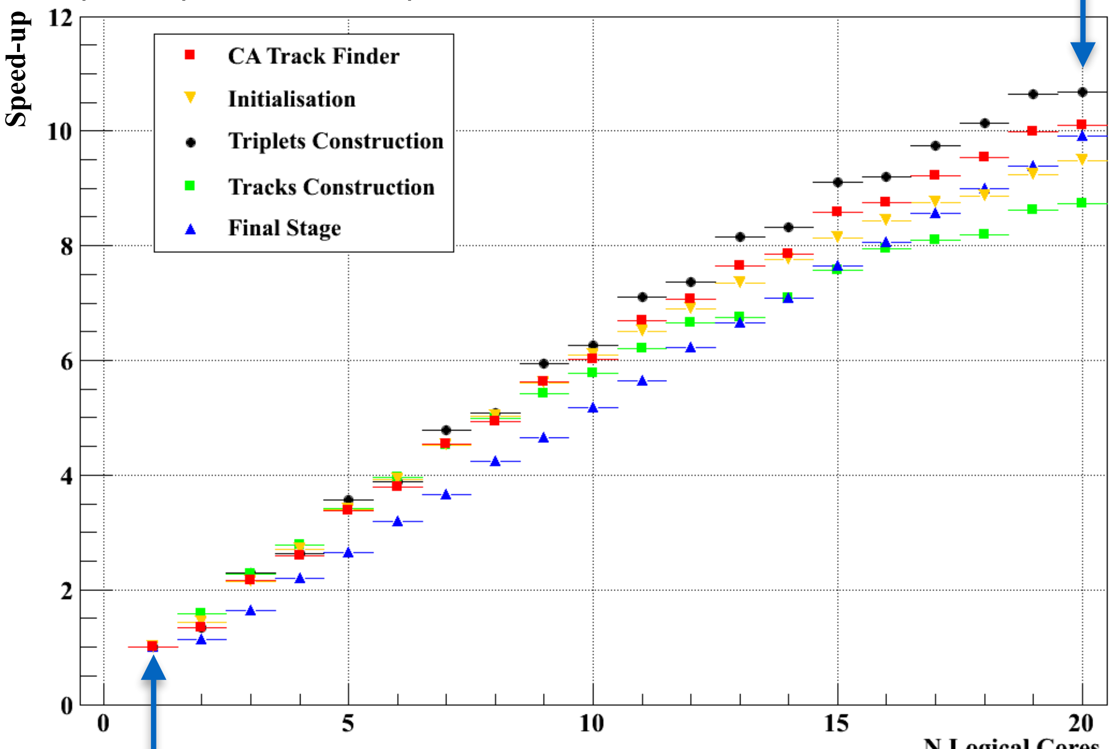


- The beam in the CBM will have no bunch structure, but continuous.
- Measurements in this case will be 4D (x, y, z, t).
- Significant overlapping of events in the detector system.
- Reconstruction of time slices rather than events is needed.

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

Total CA time = 84 ms

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



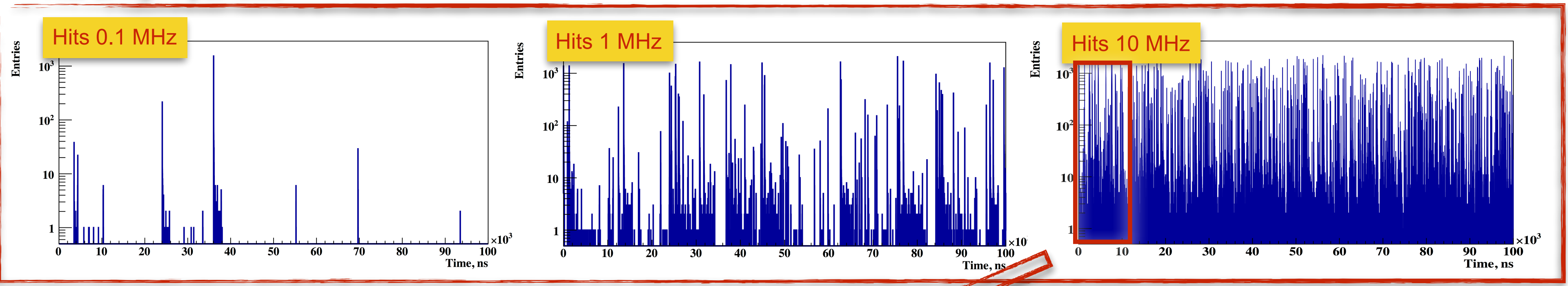
Total CA time = 849 ms

100 mbias events in a time-slice

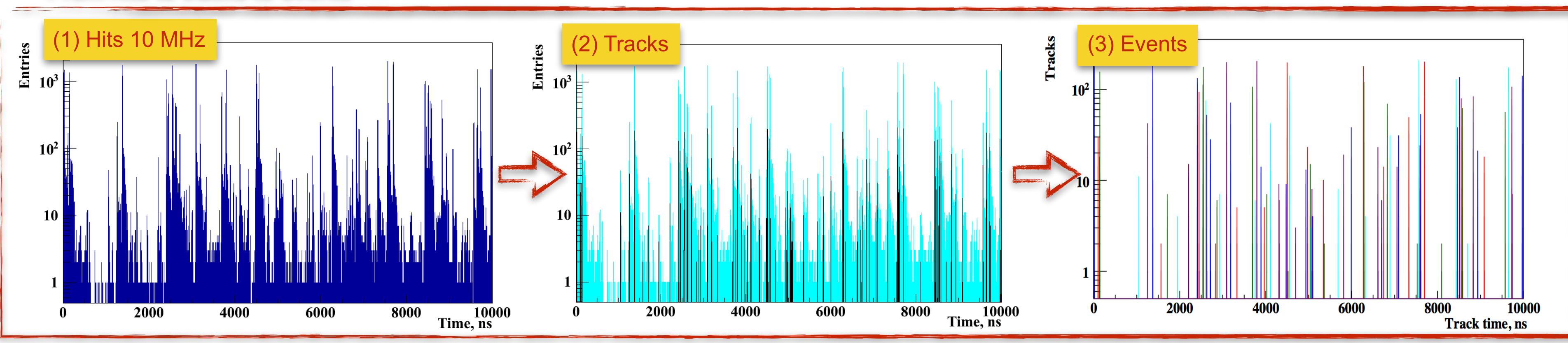
The reconstruction time 8.2 ms/event in 3D is recovered in 4D case as well

4D Event Building at 10 MHz

Hits at high input rates

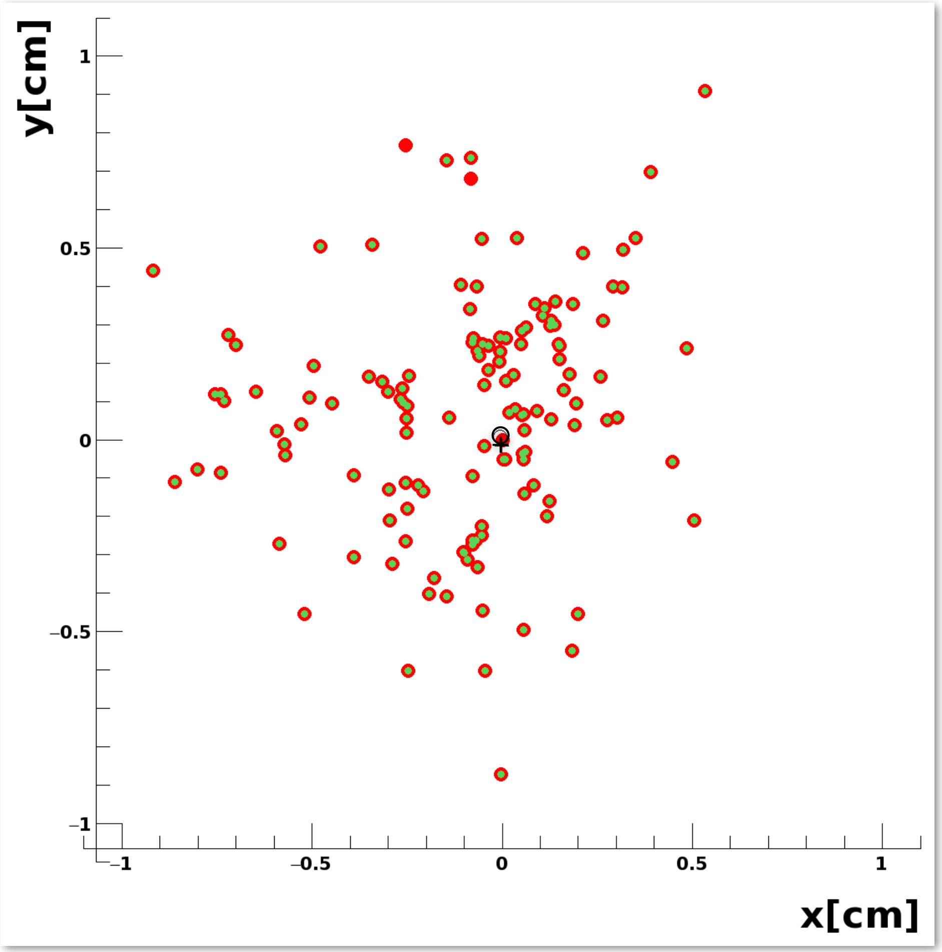


From hits to tracks to events

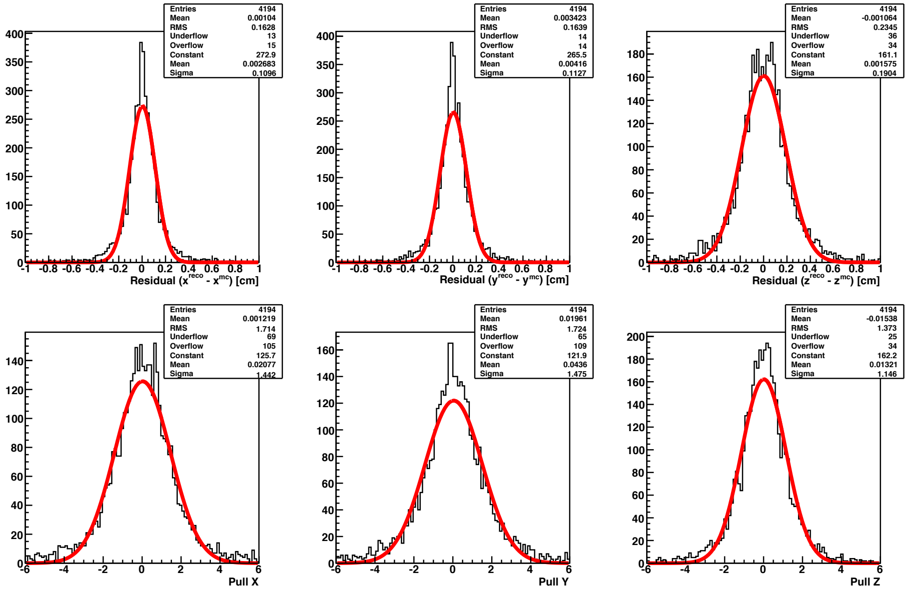


Reconstructed tracks clearly represent groups, which correspond to the original events

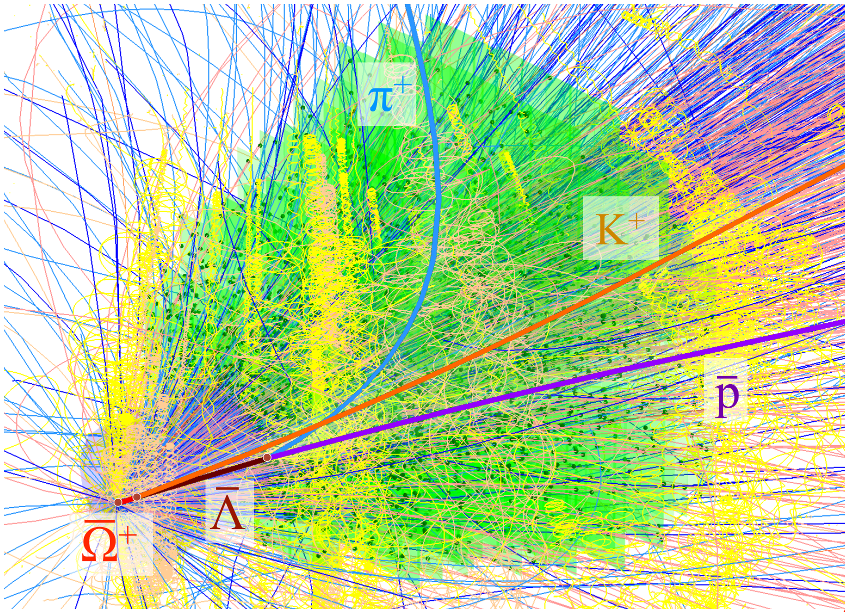
Primary Vertex Finder (à la KF Particle)



1. Choose 20 tracks with the largest momenta
2. Construct all possible 2-tracks vertices out of 20 tracks
3. Find a vertex with maximum number of neighbor vertices
4. Create a cluster of tracks from the chosen vertex and its neighbors
5. Use the chosen vertex position as an initial approximation
6. Fit the cluster of tracks with the Kalman Filter



KF Particle: Reconstruction of short-lived Particles



Simulated AuAu collision at 25 AGeV

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

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

State vector

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

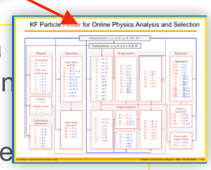
Covariance matrix

Features:

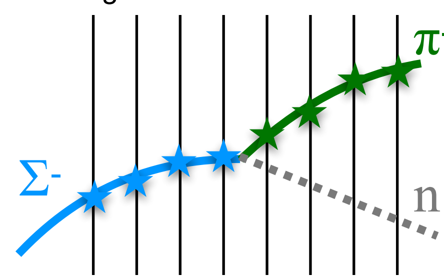
- KF Particle class describes particles by the state vector and the covariance matrix.
- The method for mathematically correct usage of covariance matrices is provided by the KF Particle package based on the Kalman filter (KF).
- Heavy mathematics of KF requires fast and vectorised algorithms.
- Mother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

- Developer
- Expert
- User

```
KFParticle Lambda(P, Pi);                  // construct anti Lambda
Lambda.SetMassConstraint(1.1157);          // improve momentum an
KFParticle Omega(K, Lambda);               // construct anti Omega
PV -= (P; Pi; K);                          // clean the primary verte
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
```

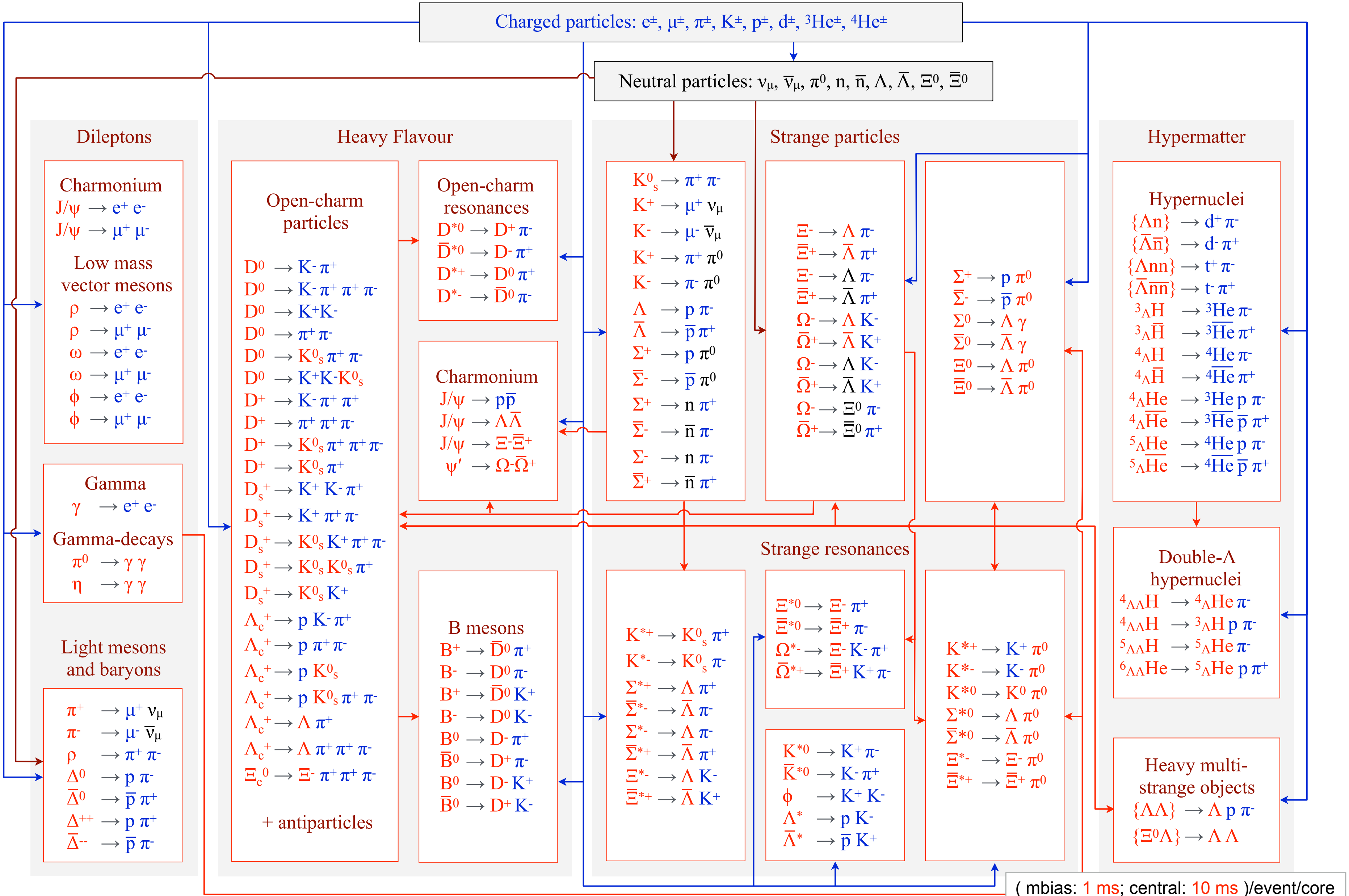


Reconstruction of decays with a neutral daughter
by the Missing Mass Method:

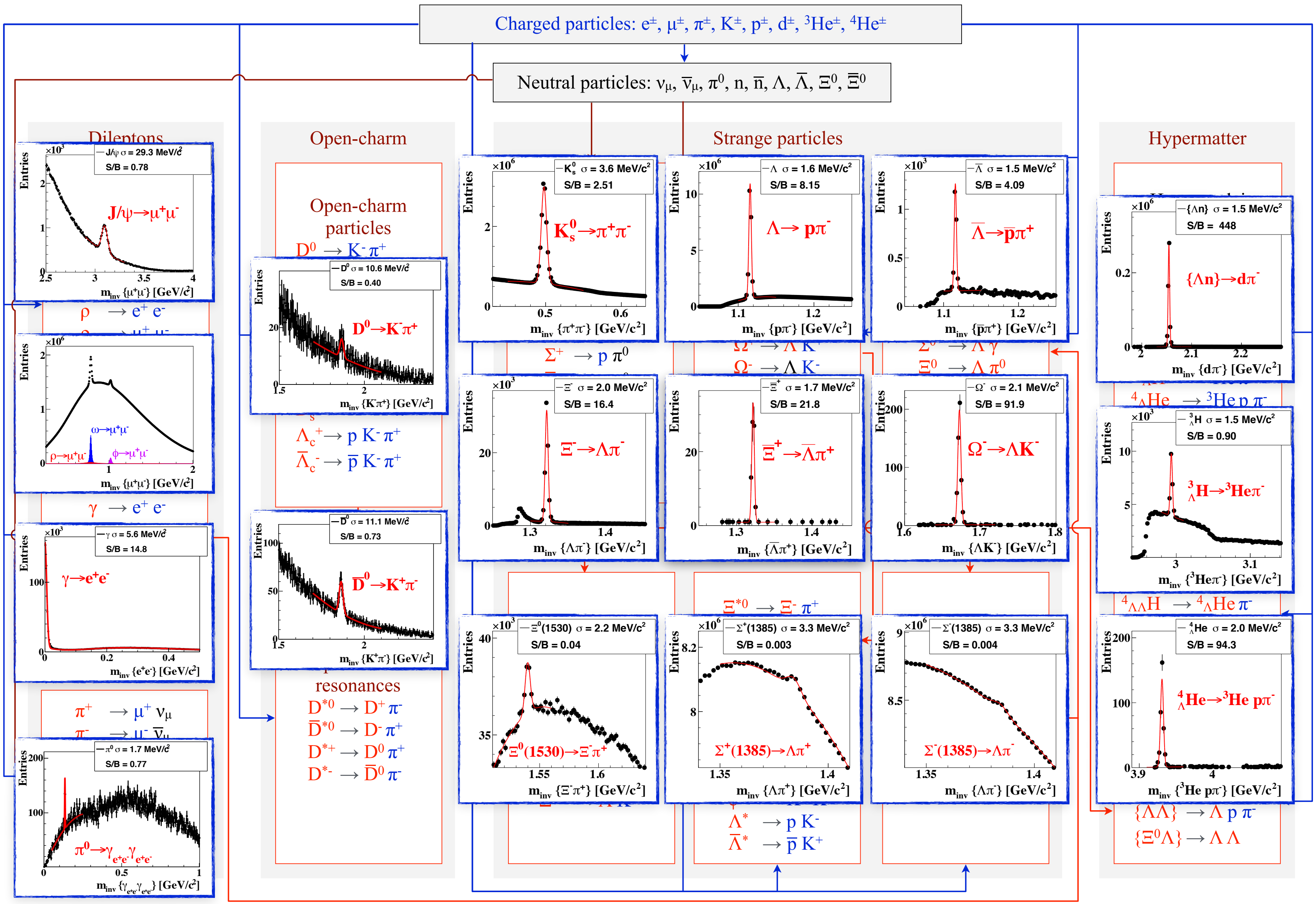


KF Particle provides a simple and very efficient approach to physics analysis

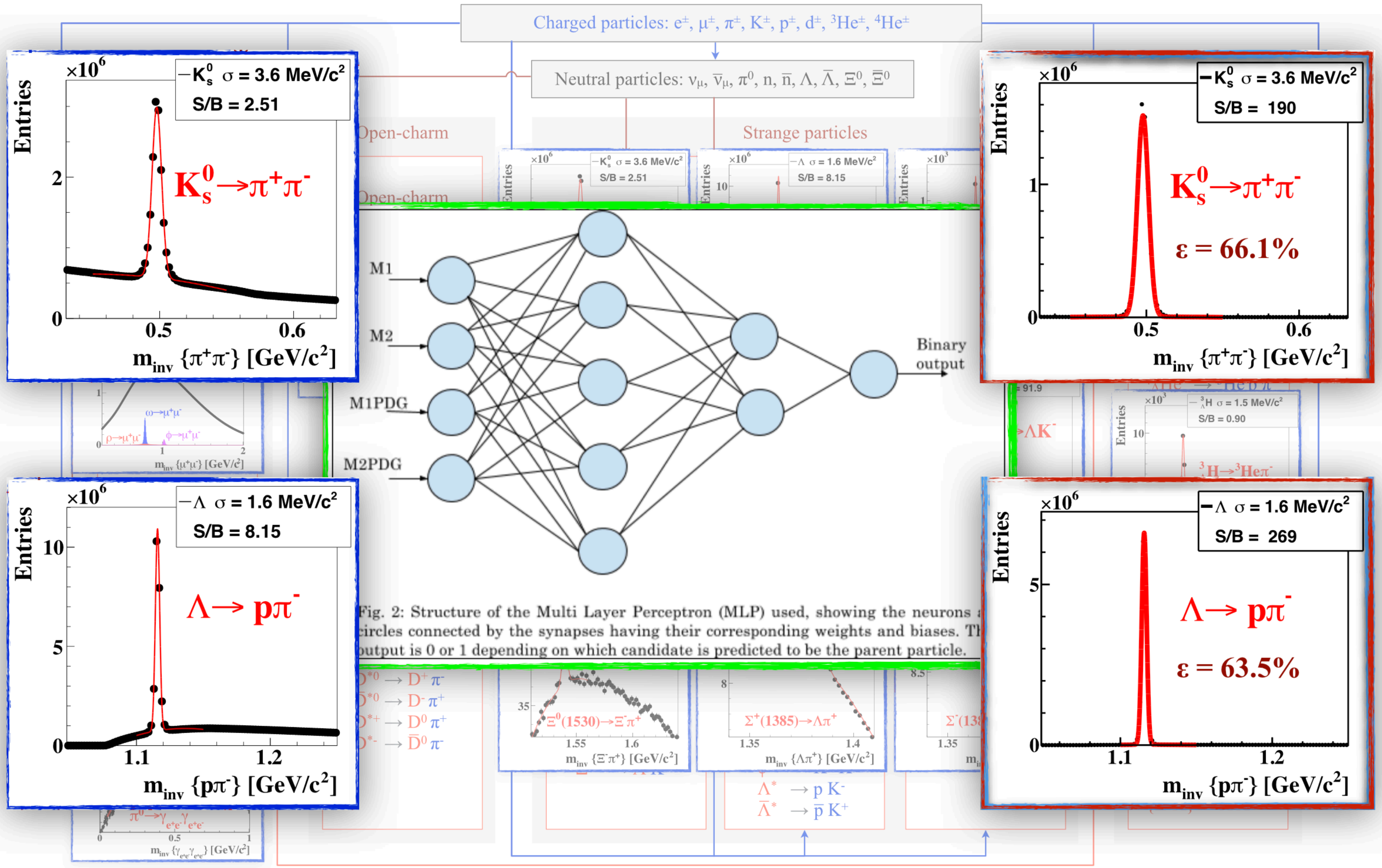
KF Particle Finder for Physics Analysis and Selection



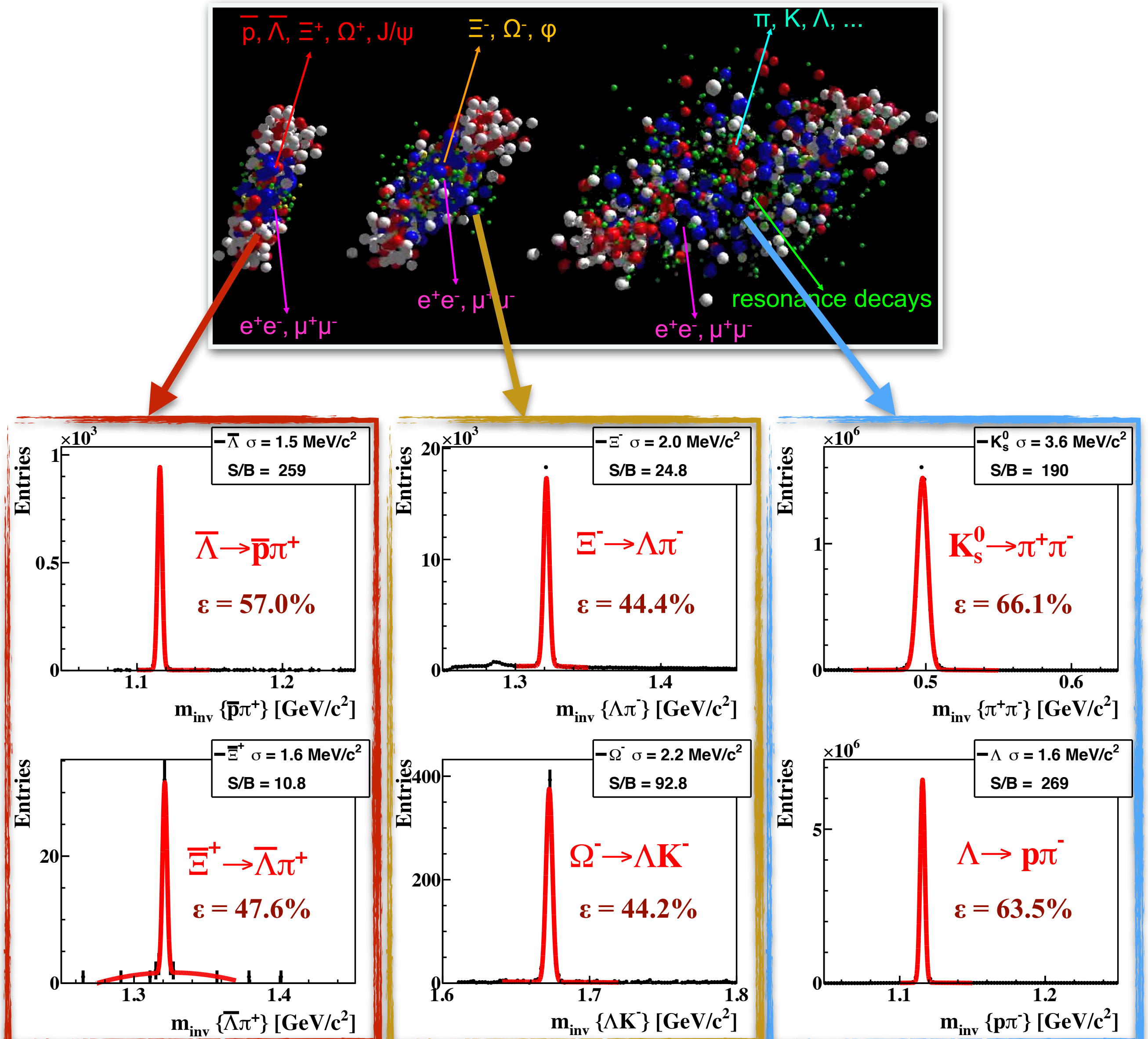
KF Particle Finder for Physics Analysis and Selection



ANN for Decay Classification

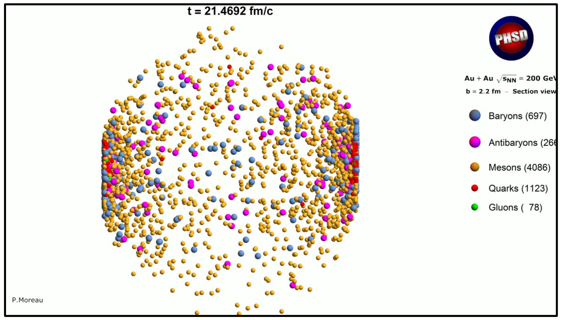


Very Clean Probes of Collision Stages



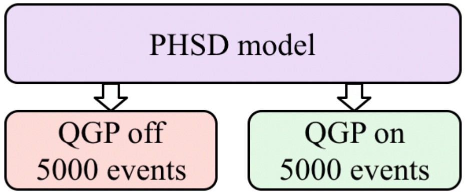
AuAu, 10 AGeV, 3.5M central UrQMD events, **MC PID**

ANN for Event Classification



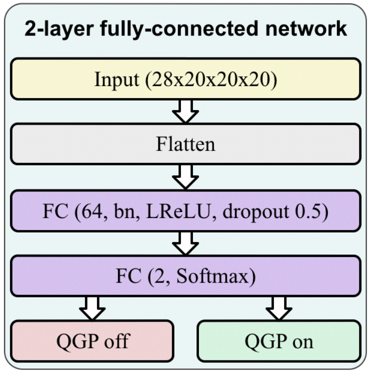
PHSD simulation of Au+Au collision
theory.gsi.de/~ebratkov/phsd-project/

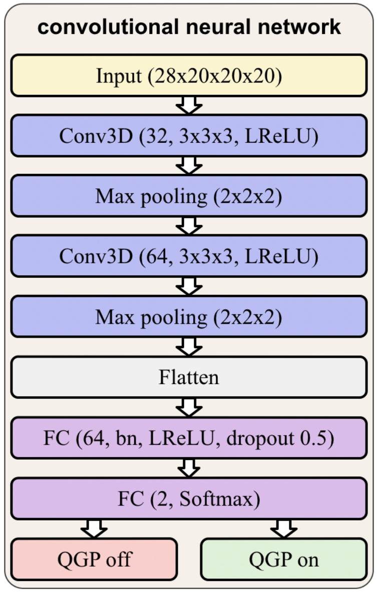
Central Au+Au 31.2A GeV

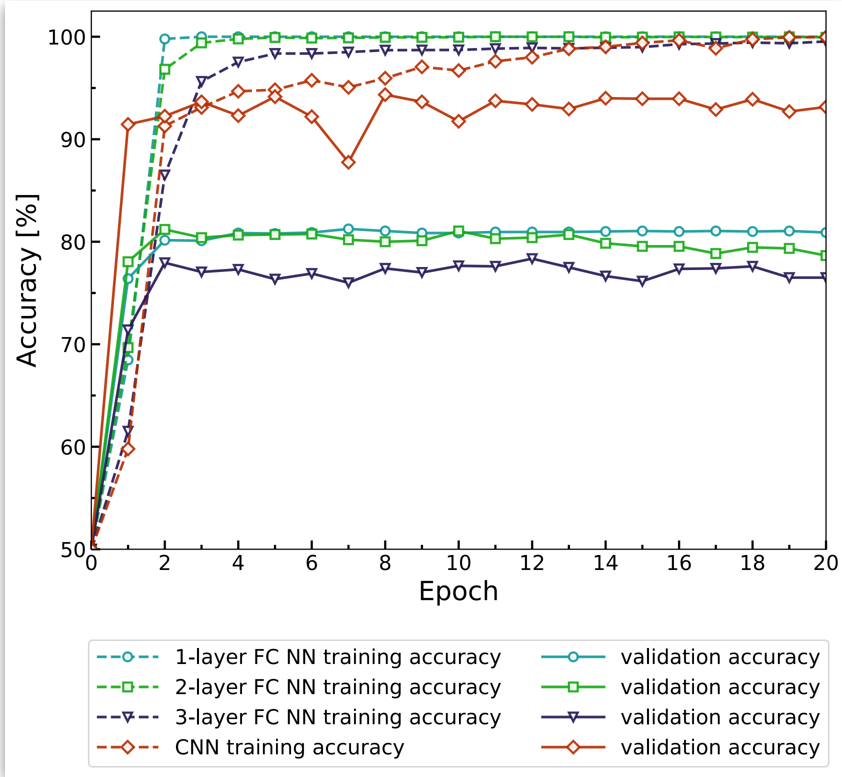


How to classify an event?

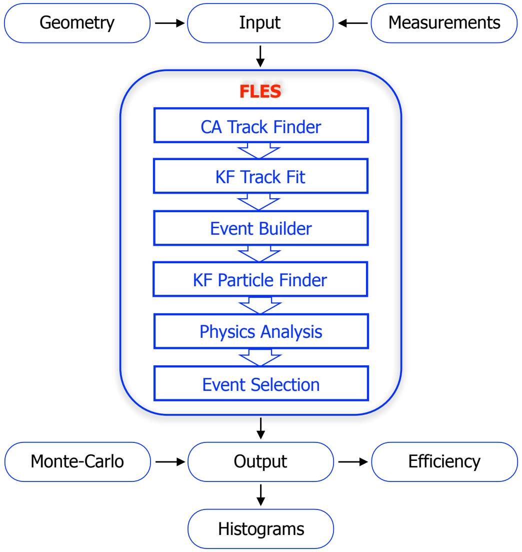
Architecture		Accuracy
FC NN	1-layer	~80%
	2-layer	~80%
	3-layer	~75%
CNN		**>90%**

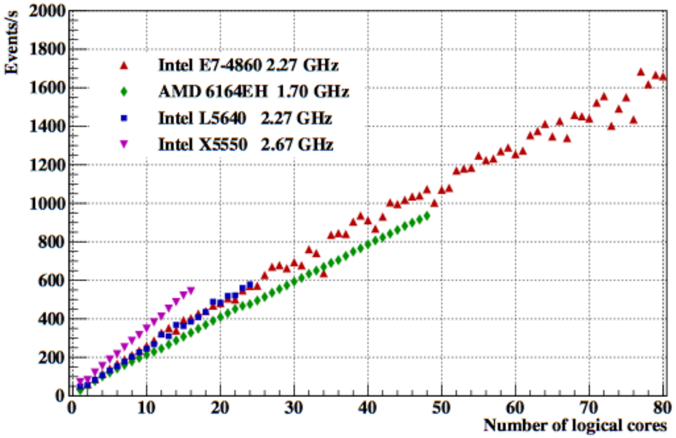


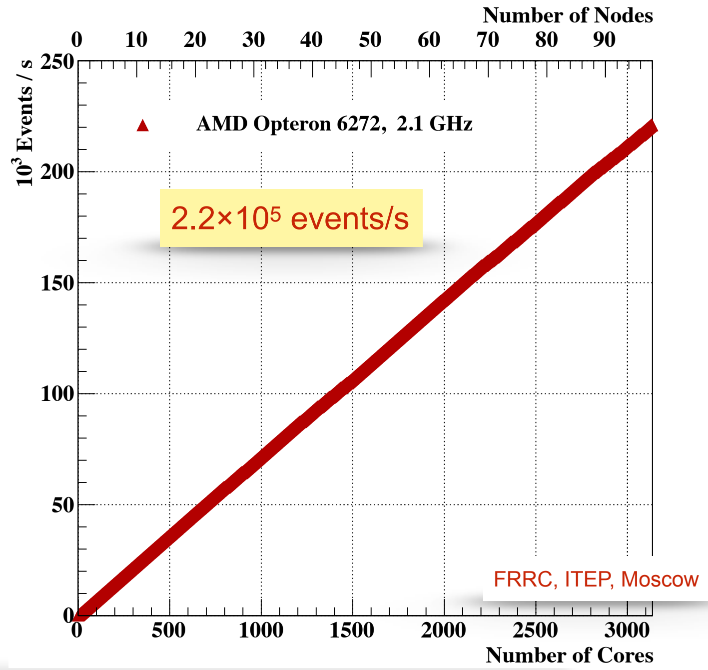




Running FLES on HPC Node/Farm



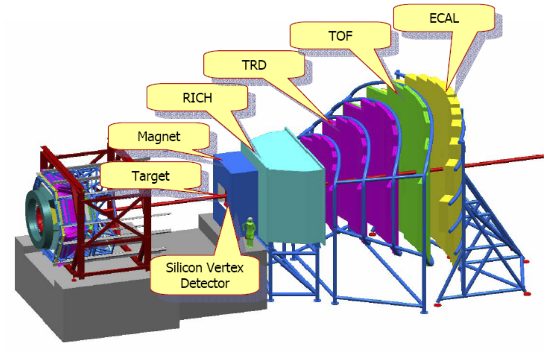




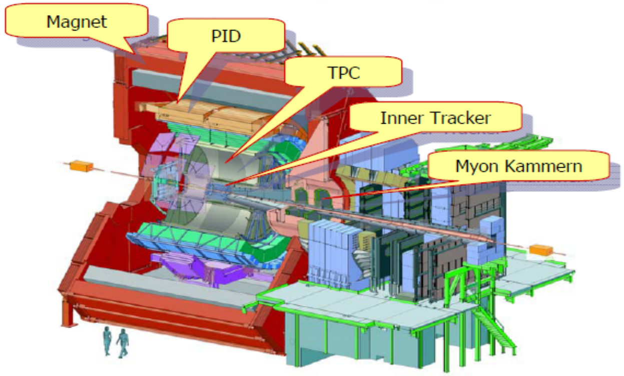
The FLES package is vectorized, parallelized, portable and scalable up to 3 200 CPU cores

A Common Reconstruction Approach/Package

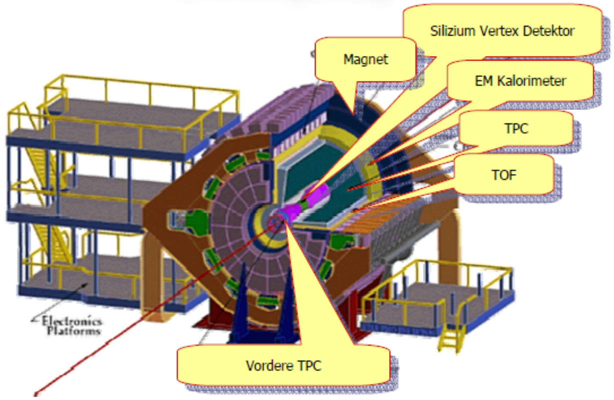
CBM (FAIR/GSI)



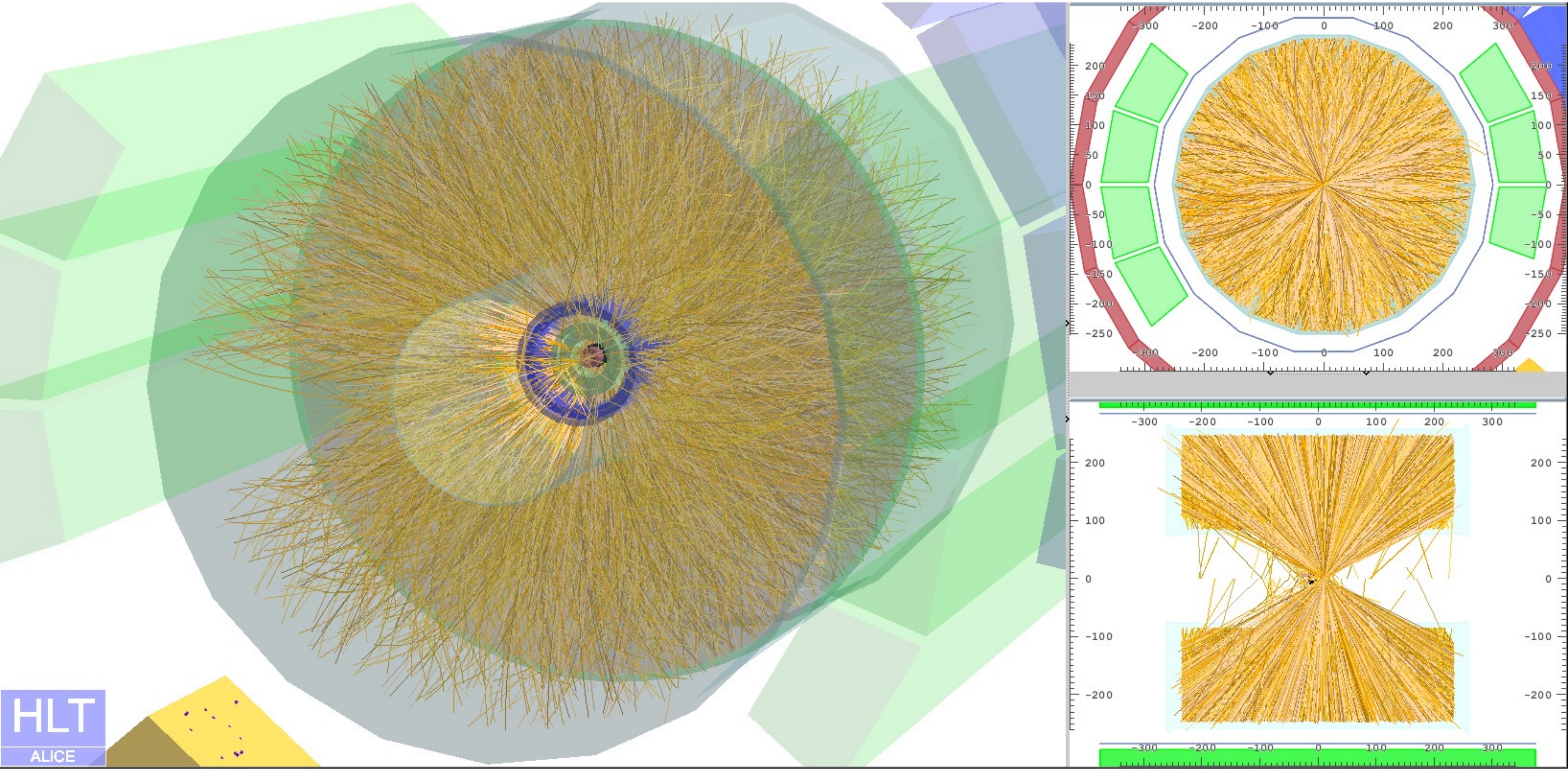
ALICE (CERN)



STAR (BNL)



TPC CA Track Finder in ALICE HLT

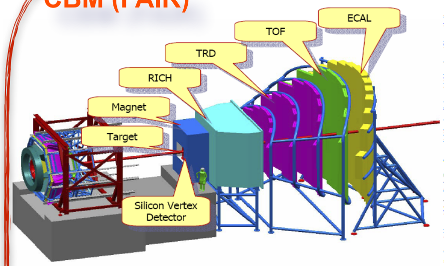


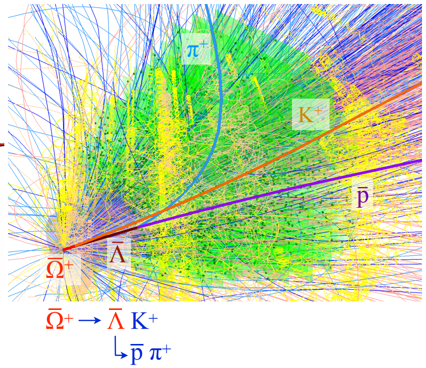
ALICE High-Level Trigger: Event of the first heavy-ion run reconstructed with the Cellular Automaton GPU tracker.

CBM -> STAR: Reconstruction and Analysis Software

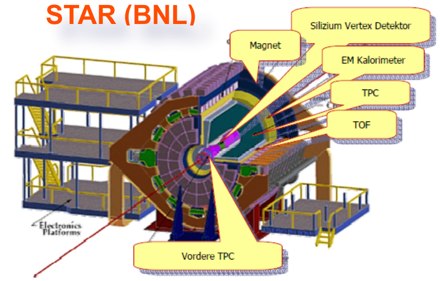
Within the FAIR Phase-0 program the CBM KF Particle Finder has been adapted to STAR and applied to real data of 2014, 2016 and BES-I.

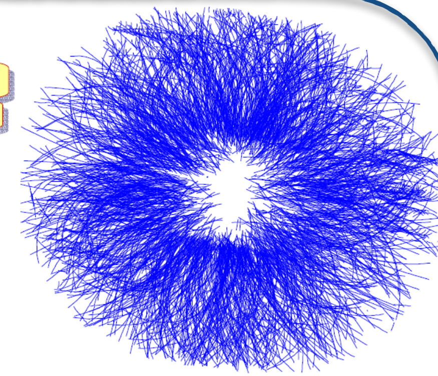
CBM (FAIR)



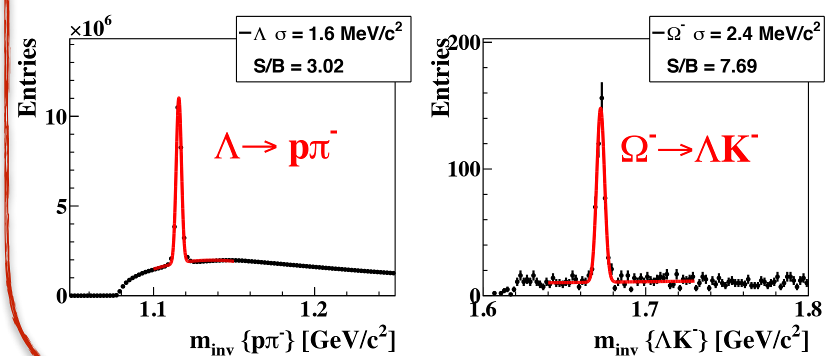


STAR (BNL)

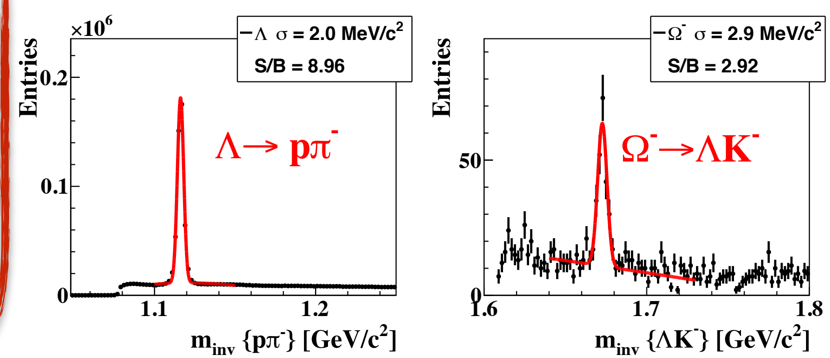




CBM, 5M central Au+Au, 10 AGeV, PHSD



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

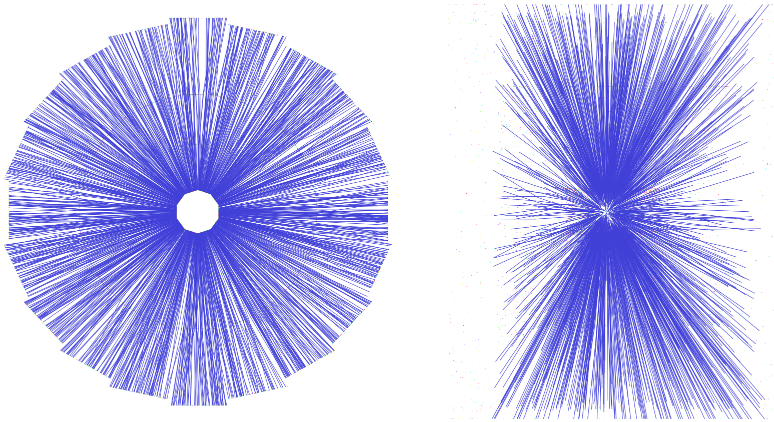


- ✓ Since 2013 (online) and 2016 (offline) the CA track finder is the standard STAR track finder for data production. Use of CA provides 25% more D^0 and 20% more W.
- ✓ The KF particle finder provides a factor 2 more signal particles than the standard approach in STAR. The integration of the KF particle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

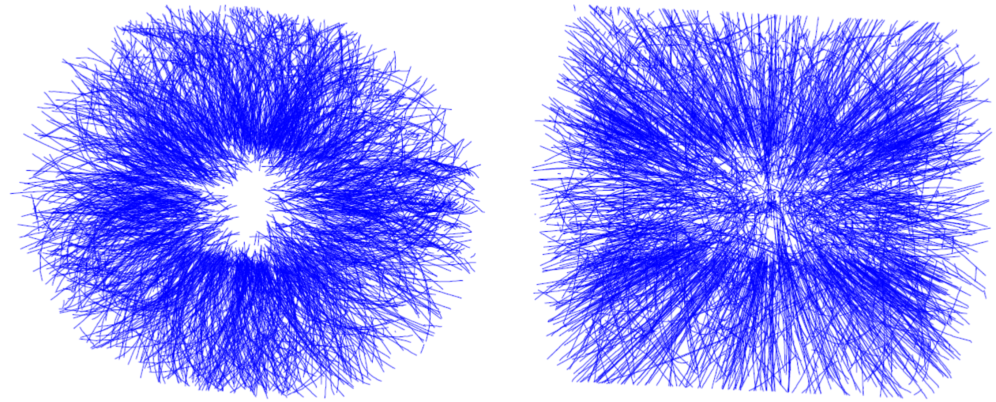
Real-time express physics analysis during BES-II runs (2019-2020)

CBM -> STAR: Reconstruction and Analysis Software

Front view **HFT CA** Side view



Front view **TPC CA** Side view



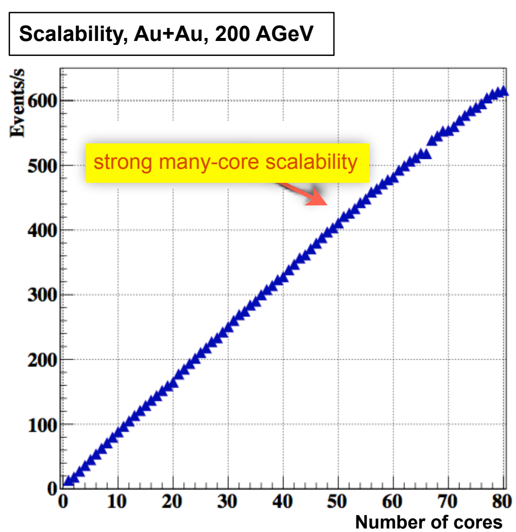
Au+Au event with 1446 reconstructed tracks in TPC

Online TPC **CA**: **simulated** Au+Au data, 200 GeV

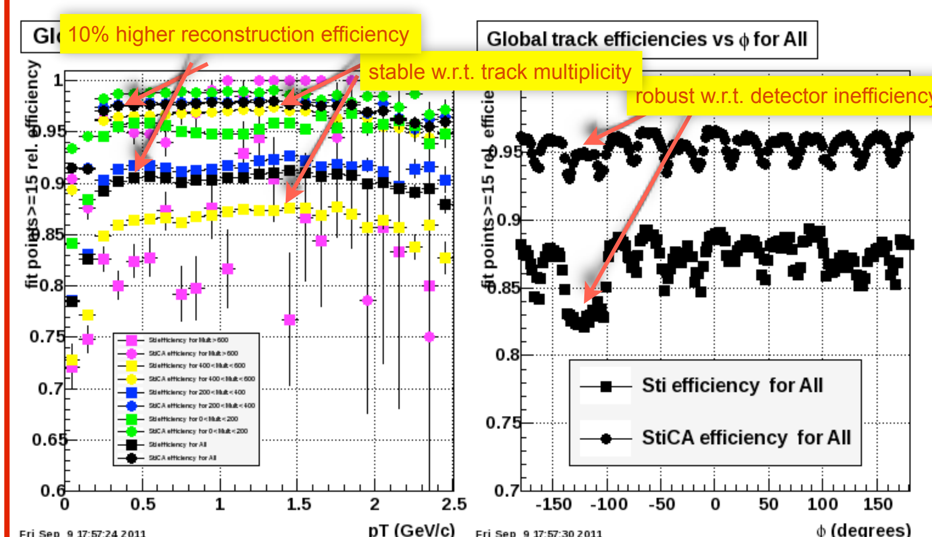
Efficiency and ratio, %	
Ref Set	96.6
All Set	88.6
Clone	10.6
Ghost	12.6
Tracks/ev	659
Time/ev, ms	47

since 2013 runs online in HLT

All set: p ≥ 0.05 GeV/c
Reference set: p ≥ 1 GeV/c
Ghost: purity < 90%



Offline TPC Sti+**CA**: **real** Au+Au data, 200 GeV, Year **2010**



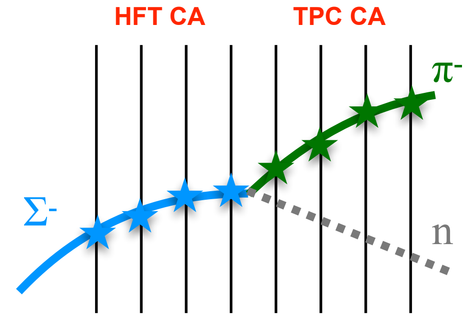
Since August **2016** the Sti+CA track finder is the standard STAR track finder for offline data production, providing **25% more D^0** and **20% more W**

Missing Mass Method

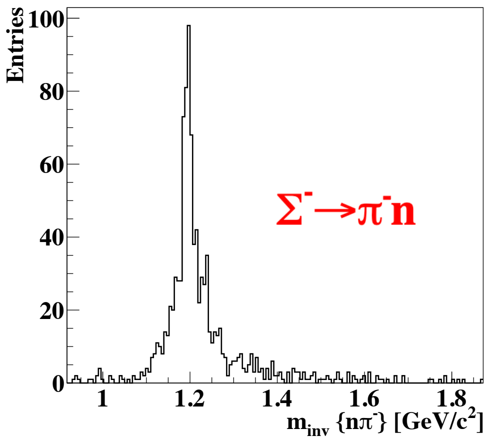
- Σ^+ and Σ^- have only channels with at least one neutral daughter.

$\Sigma^+ \to p\pi^0$	$\bar{\Sigma}^+ \to \bar{p}\pi^0$	BR = 51.6%
$\Sigma^+ \to n\pi^+$	$\bar{\Sigma}^+ \to \bar{n}\pi^-$	BR = 48.3%
$\Sigma^- \to n\pi^-$	$\bar{\Sigma}^- \to \bar{n}\pi^-$	BR = 99.8%

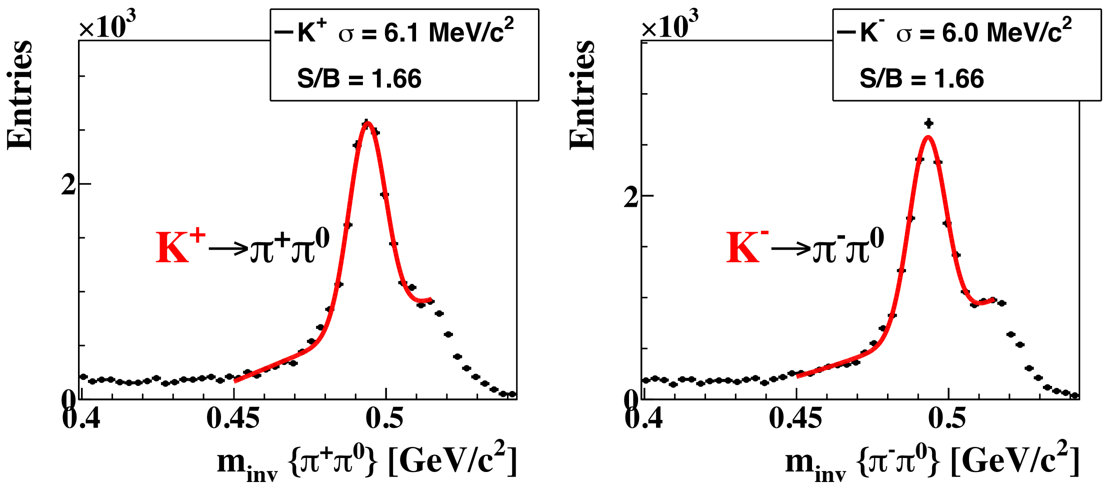
- Lifetime is sufficient to be registered by the tracking system: $c\tau$ = 2.4 cm for Σ^+ and $c\tau$ = 4.4 cm for Σ^-.
- Can not to be identified by the PID detectors. Identification is possible by the decay topology.



HFT CA + TPC CA
35K simulated Σ^- signal events



StiCA
700K mbias AuAu at 200 GeV, 2016



- The method is approbated with K^+ and K^- in 700K mbias AuAu at 200 GeV, 2016.
- For reconstruction of Σ^+ and Σ^- a standalone track finder in HFT is required.
- Use of HFT CA + TPC CA makes it possible to study Σ physics in STAR.

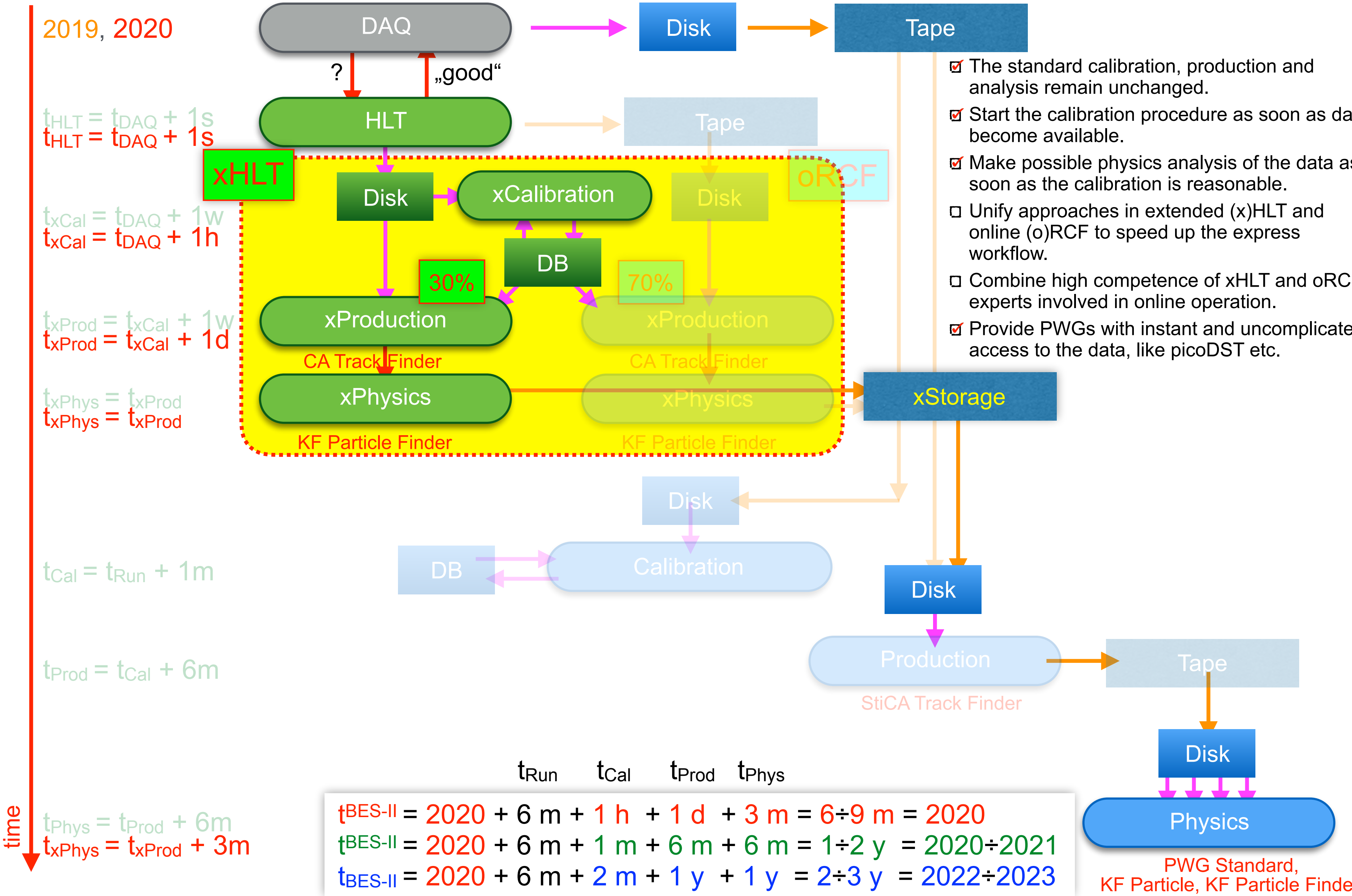
KF Particle Finder Comparison Summary

- Results of KF Particle Finder are compared with the STAR standard reconstruction approach.

Decay	year	Signal	Significance	p_t
$D^0 \to K\pi$	2014	10393	70	0-10 GeV/c
		5774	45	
$D^\pm \to K\pi\pi$	2014	1357	30	1-10 GeV/c
		774	25	
$\Lambda_c^\pm \to pK\pi$	2014	261	11.0	3-10 GeV/c
		122	8.3	
	2016	459	9.6	
		337	7.6	

- KF Particle Finder provides 2 more signal with 1.5 better significance reconstructed in all compared channels due to better utilisation of data.

BES-II: eXpress+Standard Data Production and Analysis

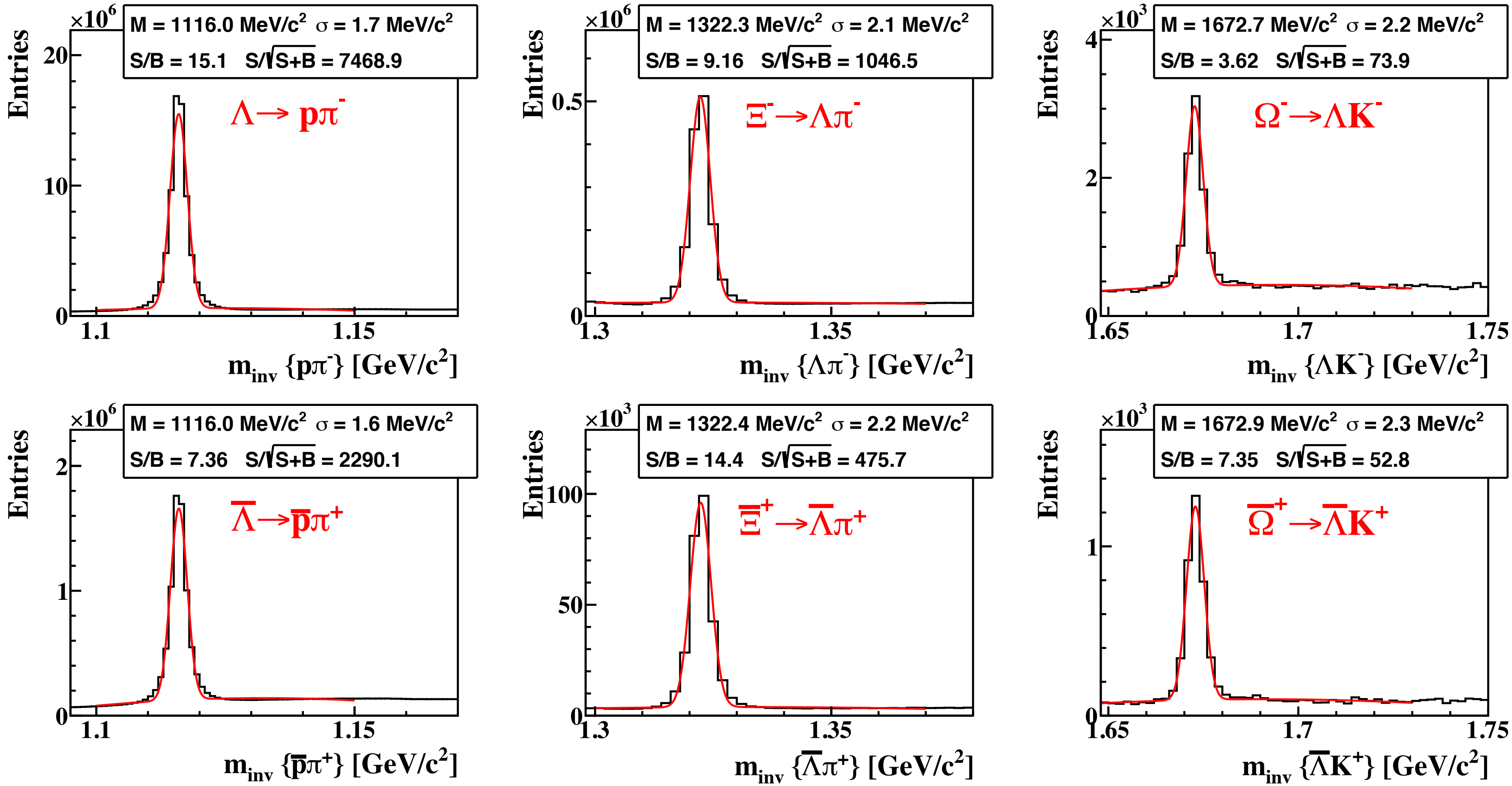


- ☑ The standard calibration, production and analysis remain unchanged.
- ☑ Start the calibration procedure as soon as data become available.
- ☑ Make possible physics analysis of the data as soon as the calibration is reasonable.
- ☐ Unify approaches in extended (x)HLT and online (o)RCF to speed up the express workflow.
- ☐ Combine high competence of xHLT and oRCF experts involved in online operation.
- ☑ Provide PWGs with instant and uncomplicated access to the data, like picoDST etc.

	t_{Run}		t_{Cal}		t_{Prod}		t_{Phys}		
t^{BES-II} = 2020	+ 6 m	+	1 h	+	1 d	+	3 m	= 6÷9 m	= 2020
t^{BES-II} = 2020	+ 6 m	+	1 m	+	6 m	+	6 m	= 1÷2 y	= 2020÷2021
t_{BES-II} = 2020	+ 6 m	+	2 m	+	1 y	+	1 y	= 2÷3 y	= 2022÷2023

BES-II: xHyperons

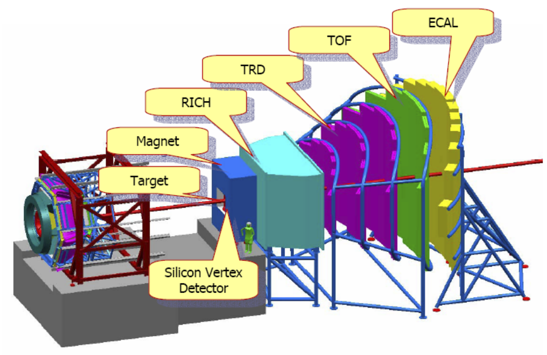
200M AuAu events at 14.5 GeV, 2019 BES-II express production



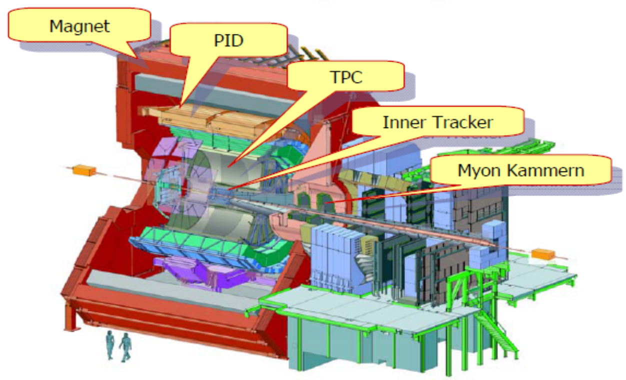
- With the express calibration and alignment we reconstruct hyperons with high significance and low level of background.
- Hyperons are clearly seen at all BES-II energies: 3, 3.2, 3.9, 7.7, 9.1, 14.5, 19.6, 27 GeV.
- High significance allows extraction of spectra.

Conclusion

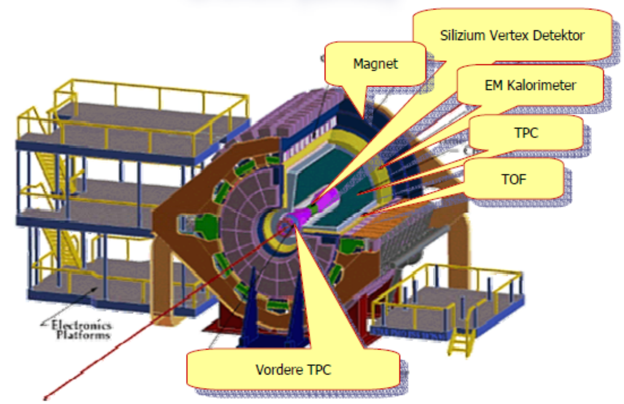
CBM (FAIR/GSI)



ALICE (CERN)



STAR (BNL)



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