New approaches in Trackfinding / Trackfitting: Cellular Automaton, etc.

I. Kisel, I. Kulakov, M. Zyzak

Tracking workshop
CERN, Geneva,
July 07, 2011

Outline

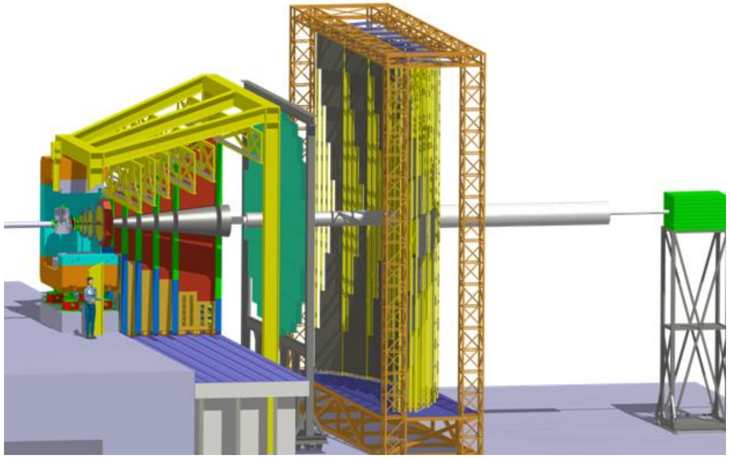
- CBM CA based track finder
- CBM CA track finder with detector inefficiency
- CBM CA track finder time optimization
- CBM CA track finder scalability on a many-core platform

- Kalman filter track fitter
- Alternative Kalman filter approaches
- CBM KF track fitter scalability on a many-core platform
- KF track fitter with Intel Array Building Blocks (ArBB)
- Deterministic Annealing Filter (DAF)

- STAR TPC CA based track finder
- STAR TPC CA track finder time optimization
- STAR TPC CA track finder with ArBB

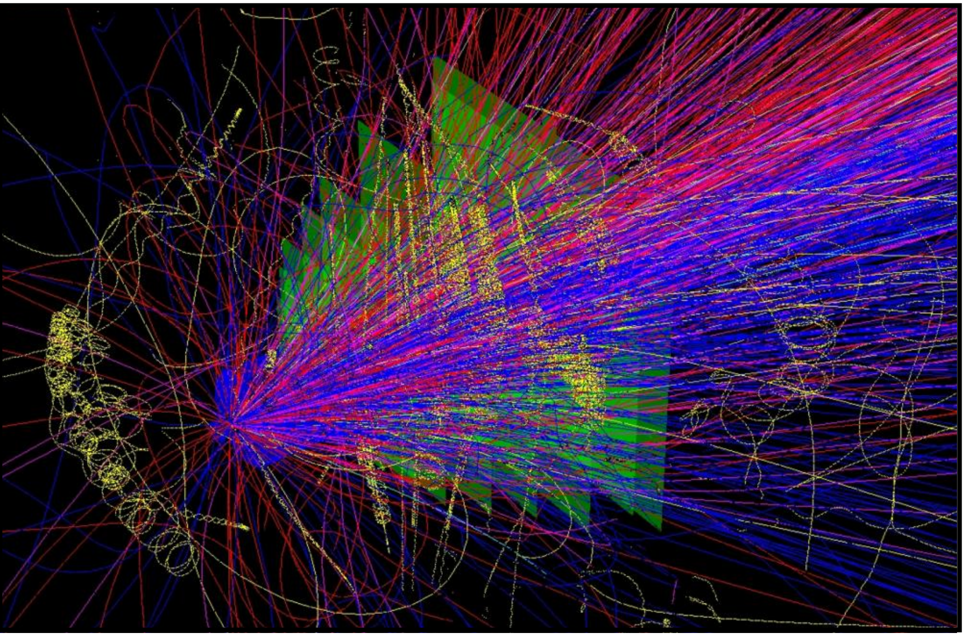
- Track reconstruction with
- Future: 4D reconstruction

Tracking Challenge in CBM

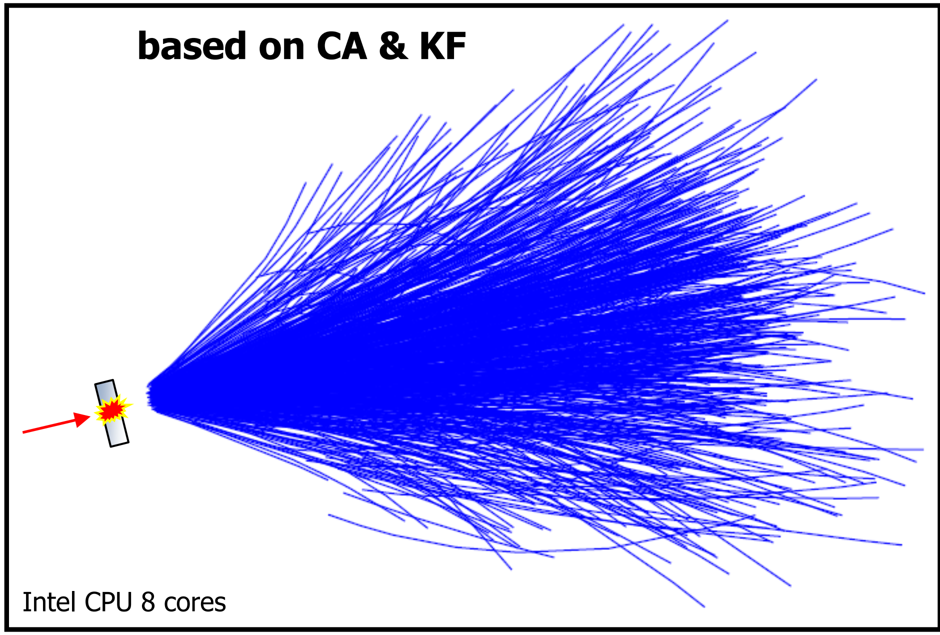


- 10^7 AuAu collisions/sec
- Double-sided strip detectors (85% fake space points)
- Non-homogeneous magnetic field
- 1000 charged particles/collision
- Track reconstruction in STS/MVD and displaced vertex search are required in the first level trigger

Simulation

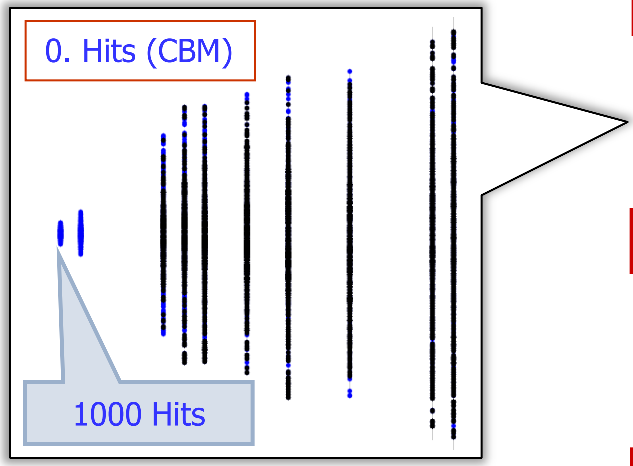


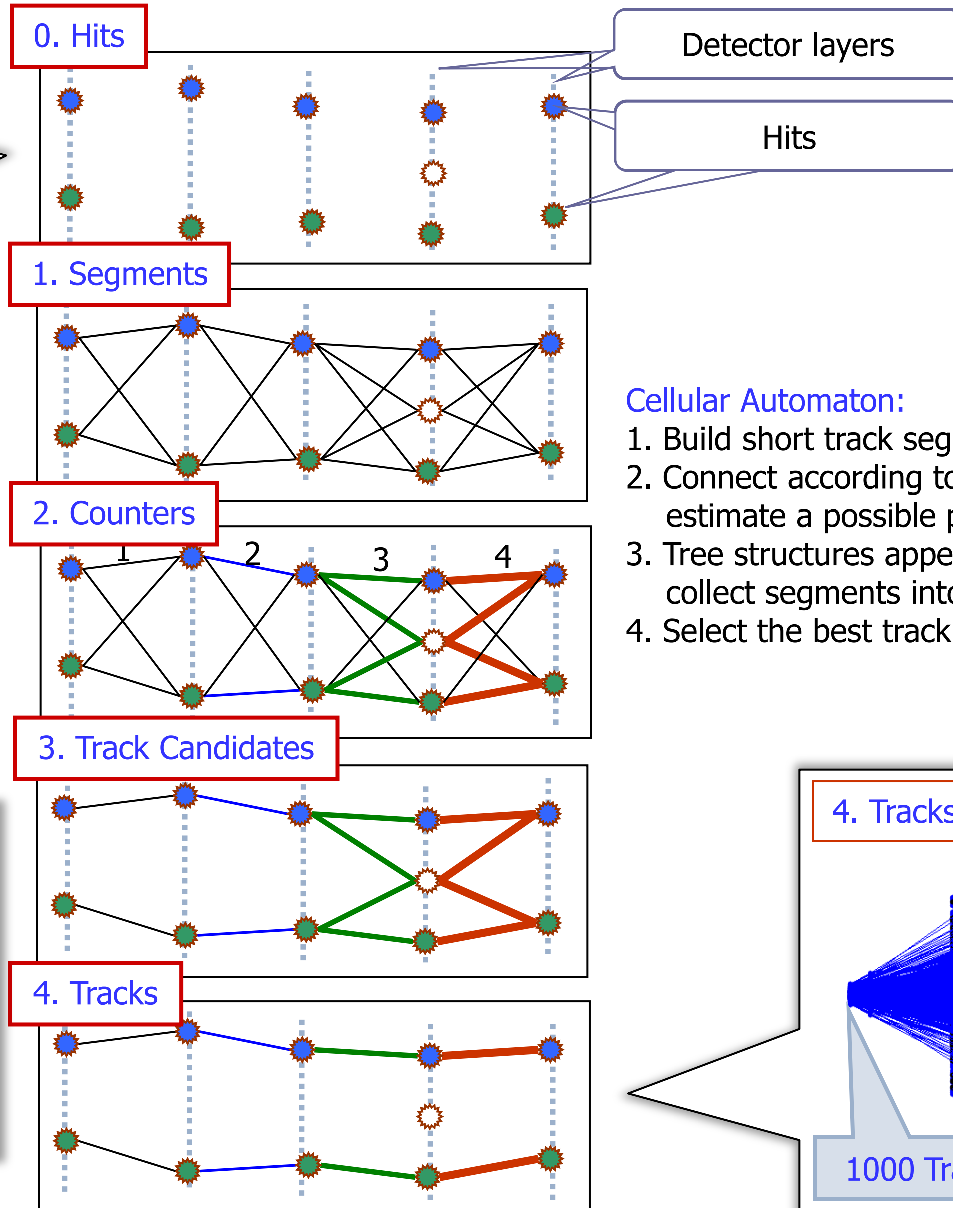
Reconstruction



Cellular Automaton (CA) as Track Finder

Track finding: Wich hits in detector belong to the same track? – Cellular Automaton (CA)

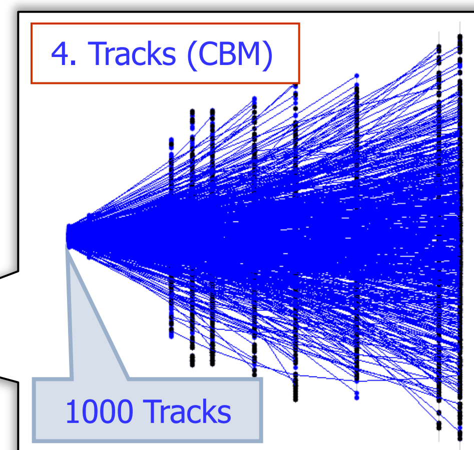




Cellular Automaton:
1. Build short track segments.
2. Connect according to the track model, estimate a possible position on a track.
3. Tree structures appear, collect segments into track candidates.
4. Select the best track candidates.

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

Perfect for many-core CPU/GPU !



CBM Track Finding Algorithm

The cellular automaton (CA) based track finder will be used both for off-line and for on-line track reconstruction in the CBM experiment.
Thus very efficient, fast and flexible realisation of the algorithm is required.

All algorithm divided on 3 stages:

- Fast (p > 0.5 GeV) primary tracks
- Slow (p < 0.5 GeV) primary tracks
- All secondary tracks

All hits (strips) which belong to the reconstructed tracks deleted from the further reconstruction

Each stage consist from 2 parts:

create «cells» →

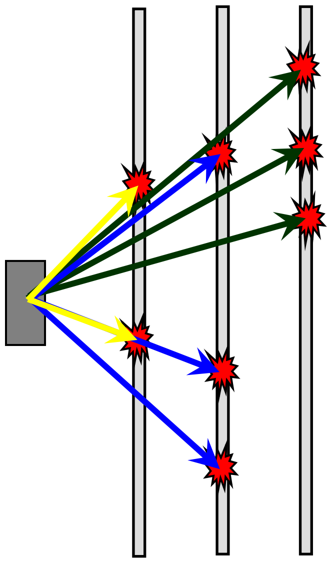
1. Finding tracklets (seeds)
   - Finding singlets
   - Finding doublets
   - Finding triplets (tracklets)
   - Selecting tracklets

apply CA rules of evolution →

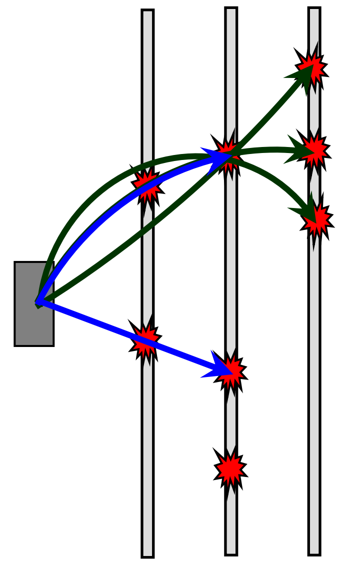
   - Finding pairs of neighbor tracklets
   - Count the "level" of tracklets (the lengths of the right connected chain of neighbors)
2. Collecting tracks
   - Collecting track candidates
   - Selecting track candidates

CBM Track Finding Algorithm (Continue): Finding Tracklets

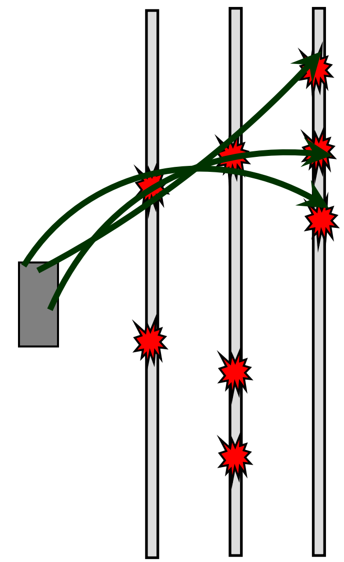
1.1. Singlets



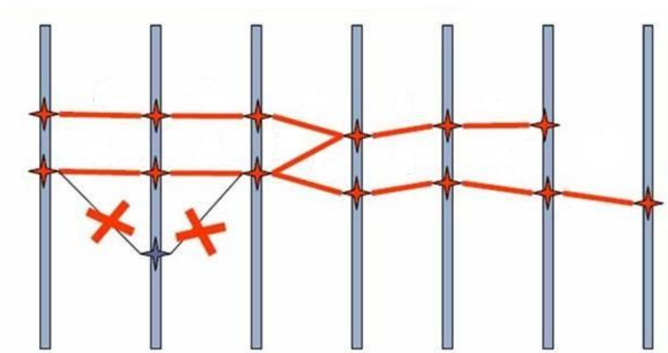
1.2. Doublets



1.3. Triplets



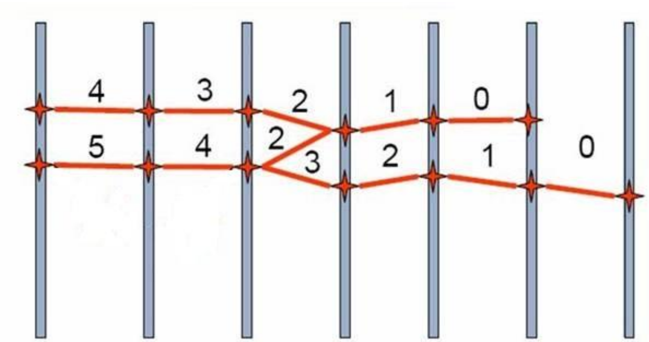
1.4. Selecting



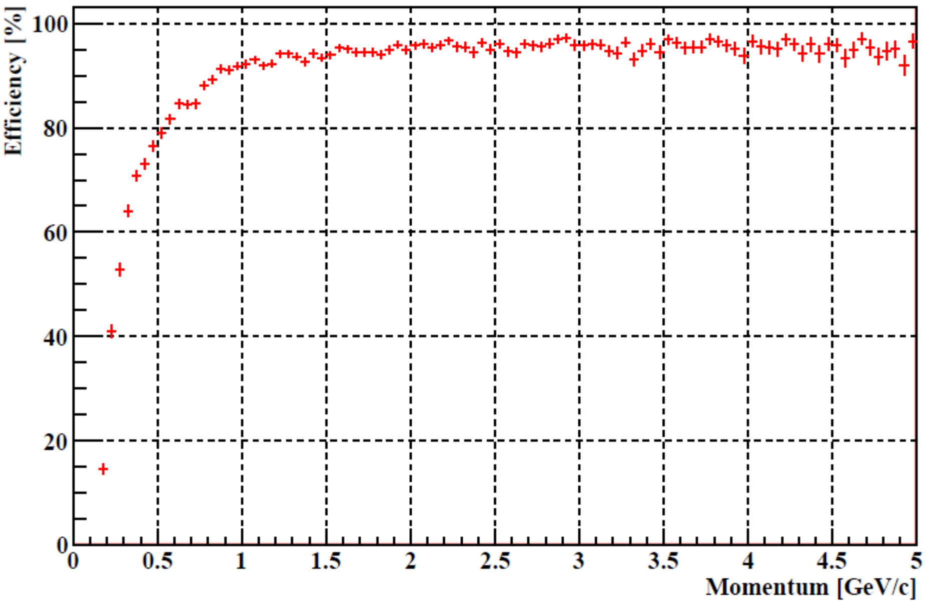
1.5. Neighbors

Neighbors:
- Have 2 common hits
- Have same momentum
  (accurate within errors)

1.6. “Level”



CBM CA Track Finder Status



AuAu 25 AGeV central; 2 MVD+8 STS; Statistic: 100 events

Efficiency and ratios, %	
Fast Prim Set	97.8
All Set	87.6
Clone	0.8
Ghost	12.8
Quality (reco hits)	88.6
Tracks/ev	733
Time/ev, s	1.4

Reconstructable track:
≥ 4 consecutive MC points

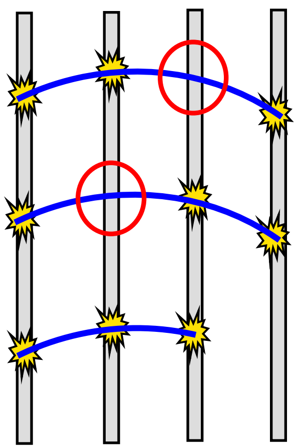
All set: $p \geq 0.1$ GeV/c
Reference set: $p \geq 1$ GeV/c
Ghost: purity < 70%

The CBM CA track finder shows high reconstruction efficiency.

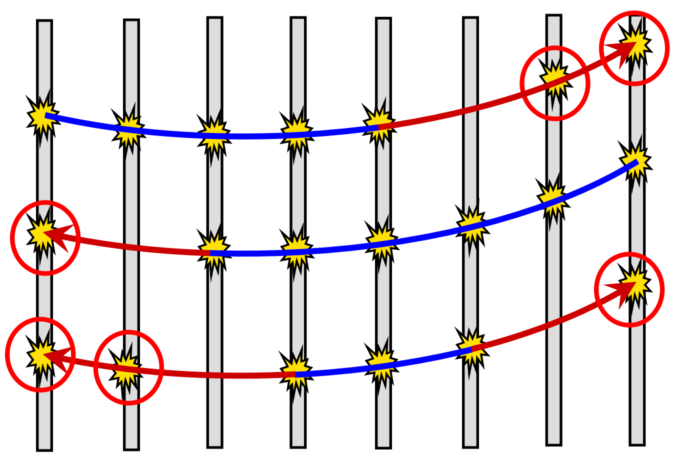
CBM CA Based Track Finder With Detector Inefficiency

The algorithm of STS track reconstruction had been developed in assumption of detector planes with 100% registration efficiency.
The investigation of stability of the track finder with respect to the detector inefficiency was required.

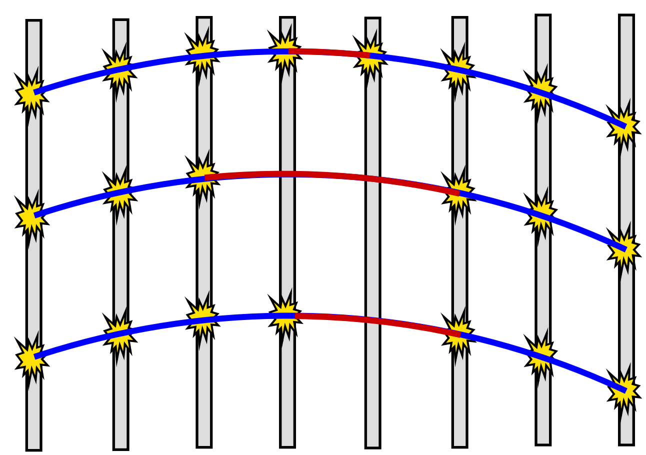
1. Triplets can skip one station with a missing hit



2. Gathering individual hits by track-candidates

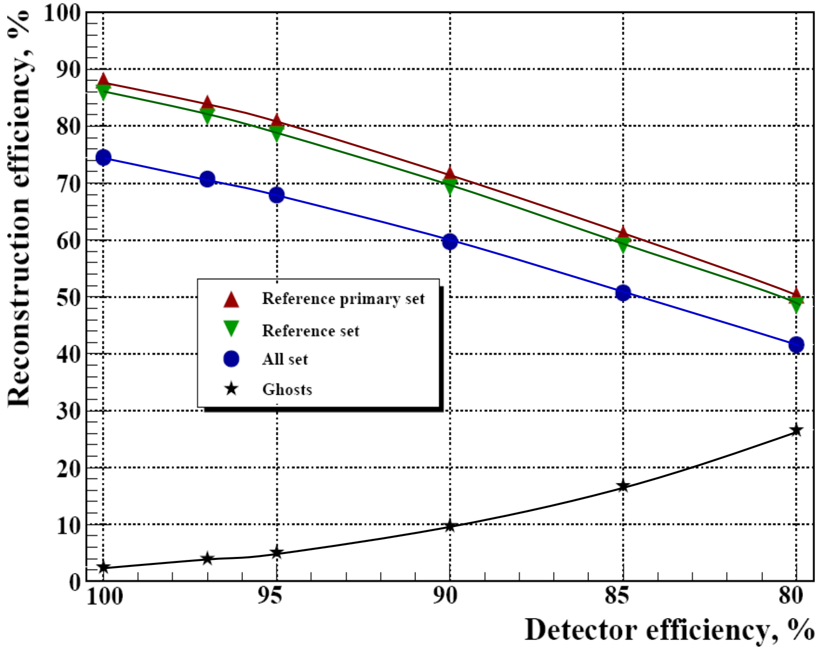


3. Merging separate parts of track

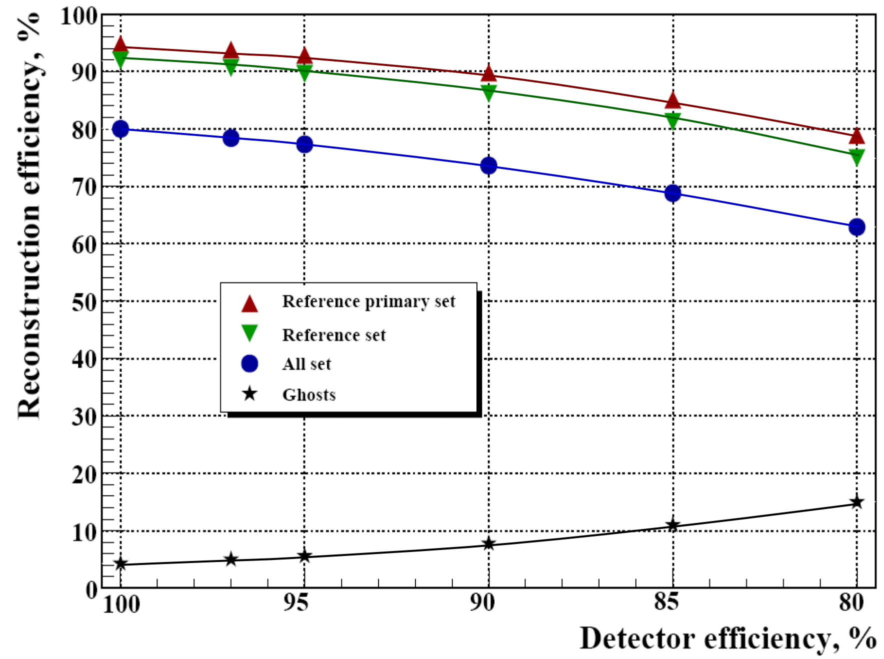


Reconstruction Efficiency

Sep 2010



Feb 2011



Reconstructable track:
≥ 4 consecutive MC points

Execution time (3% inefficiency), ms/ev	Sep 2010	Feb 2011
	564	591

All set: $p \geq 0.1$ GeV/c
Reference set: $p \geq 1$ GeV/c
Ghost: purity < 70%

Au+Au 25 AGeV central; 8 STS; 100 events;

CA track finder is stable with respect to the detector inefficiency.
The track reconstruction efficiency has been increased on 8% (in case of 3% inefficient detector).

Track Quality (Residuals)

Sep 2010

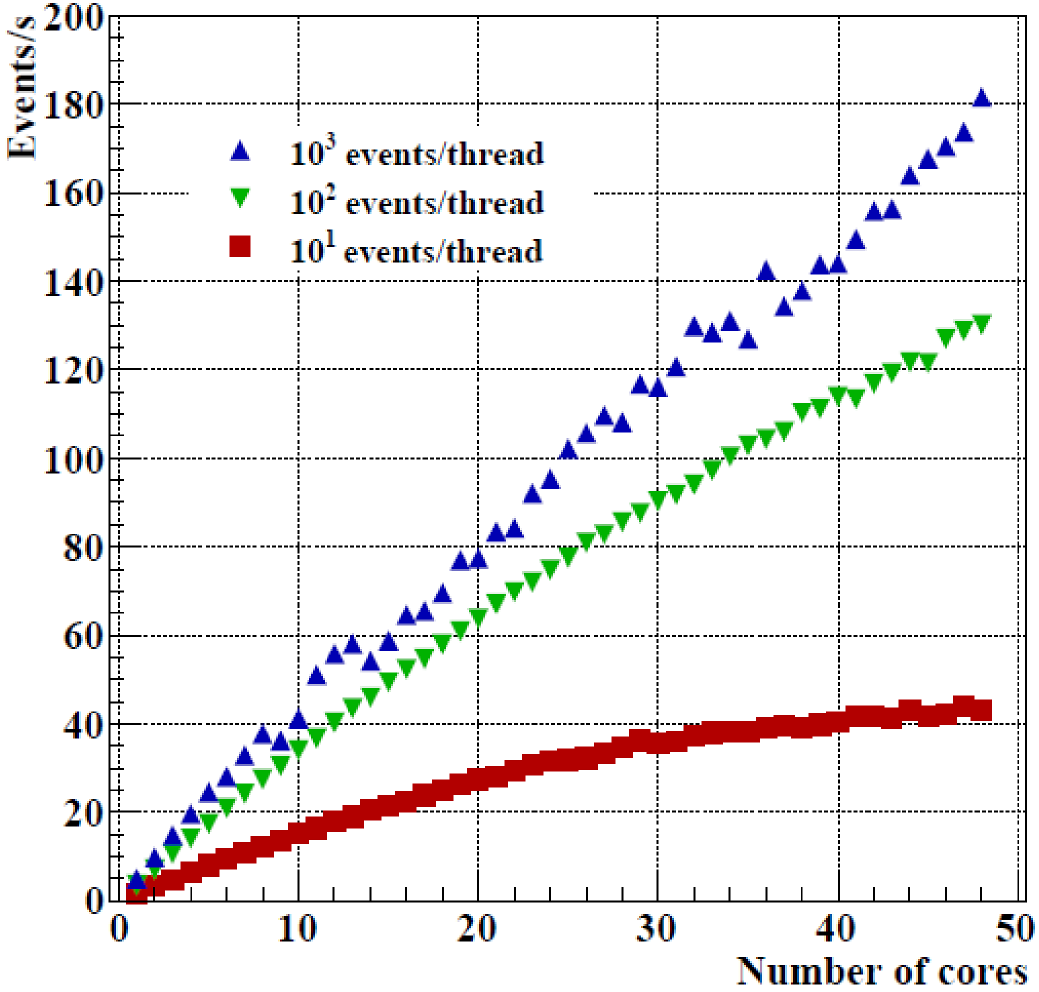
Detector efficiency	100	97	95	90	85	80
x, µm	12	12	13	14	14	15
y, µm	57	59	61	66	70	72
t_x, mrad	0.34	0.35	0.36	0.37	0.39	0.41
t_y, mrad	0.59	0.60	0.61	0.62	0.64	0.67
p, %	1.23	1.29	1.33	1.43	1.53	1.62

Feb 2011

100	97	95	90	85	80
12	13	13	14	14	15
57	60	61	65	69	73
0.35	0.36	0.37	0.38	0.40	0.42
0.60	0.61	0.61	0.63	0.64	0.66
1.22	1.25	1.28	1.34	1.41	1.48

Track momentum resolution has been improved with respect to STS detector inefficiency.

Scalability of the CBM CA Track Finder



Au+Au 25 AGeV; mbias; realistic STS

Measure tracks throughput rather than time per track.

Given n threads each filled with 10^m events, run them on specific n logical cores with 1 thread per 1 core.

For small groups of events the overhead becomes significant, while large groups of tracks use CPU more efficient.

opladev35 (CERN, Openlab) with 4 CPUs AMD E6164HE
12 cores per CPU, 1.7 GHz; TBB

A new Intel machine has been installed at GSI:
4 CPUs Intel Xeon Westmere E7-4860
in total 40 physical cores or 80 logical cores, 2.3 GHz

Strong many-core scalability for large groups of minimum bias events.
Reconstruction speed of 5 ms/event/node has been achieved.

CBM CA Track Finder Time Optimization

- Take into account additional information (acceptance, chi2)
- Resort hits
- Simplify computations where high precision is not needed
- Reduce copying of data
- Decrease number of finding iteration

Efficiency and ratios, %		
	Mar 2011	Apr 2011
Fast Prim Set	95.4	95.5
All Set	86.3	86.3
Clone	0.4	0.4
Ghost	5.1	4.4
Quality (reco hits)	89.9	90.3
Tracks/ev	718	717
Time/ev, ms	985	199

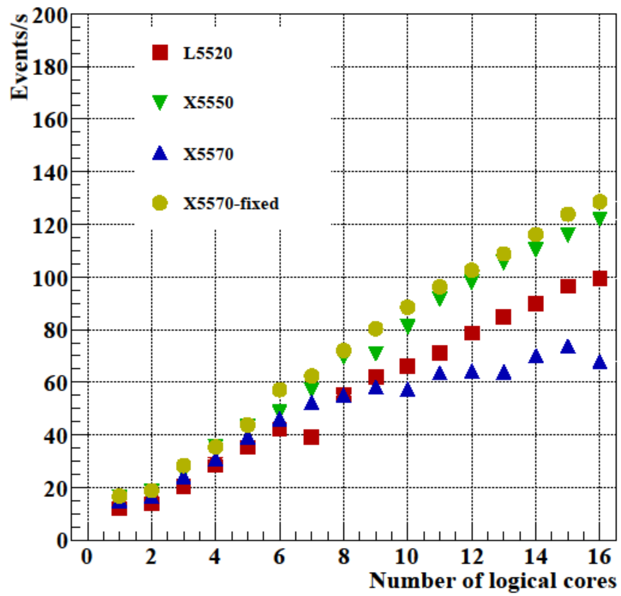
AuAu 25 AGeV central; 2 MVD+8 STS; Statistic: 100 events

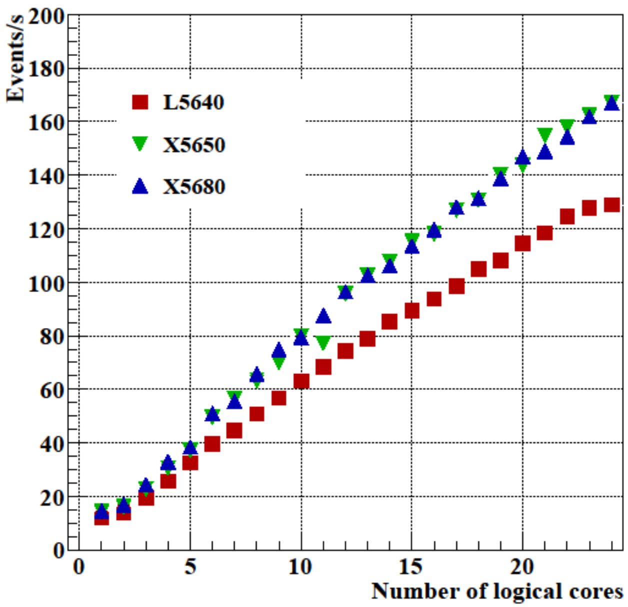
Reconstructable track:
$\geq$ 4 consecutive MC points

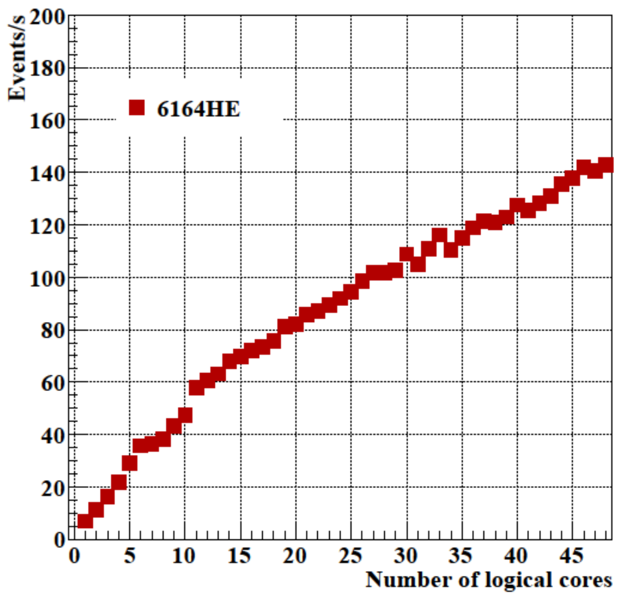
All set: $p \geq 0.1$ GeV/c
Reference set: $p \geq 1$ GeV/c
Ghost: purity < 70%

Time of track reconstruction has been improved by factor of 5.

Track Finder Scalability tests on different systems







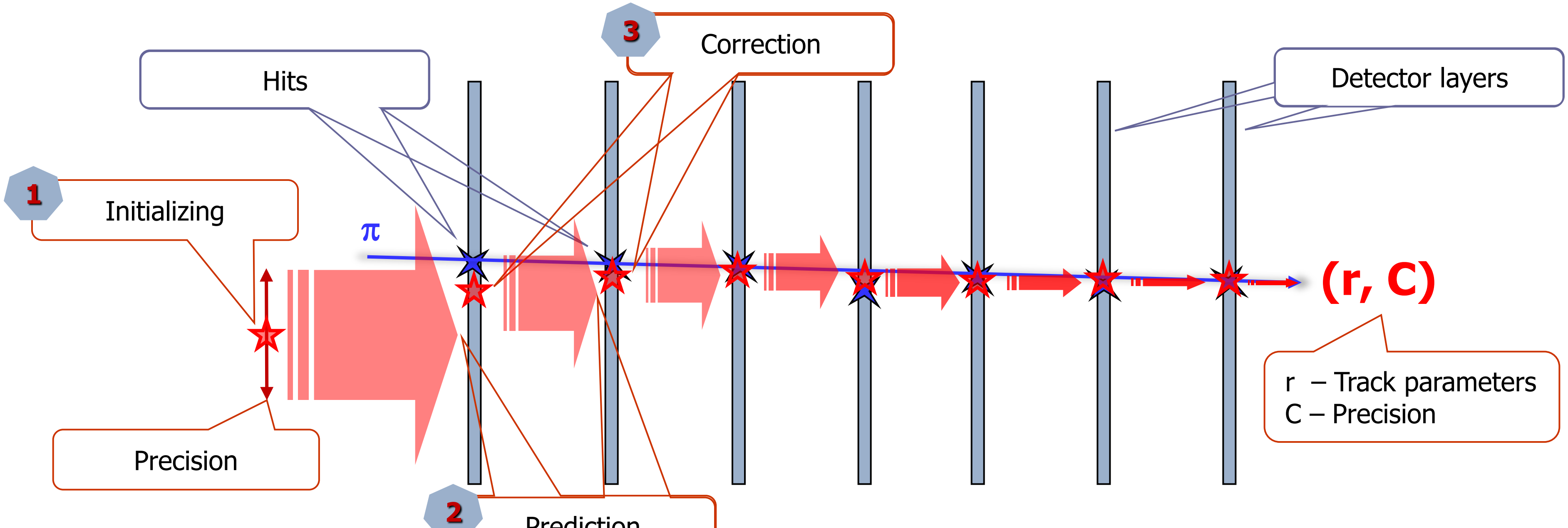
7 different systems have been tested.
On 5 of them CA has strong linear scalability.
One was fixed by CPU microcode modification.
For one a further investigation is needed.

100 central Au+Au 25 AGeV; realistic STS

with J. Leduc (CERN, openlab)

Kalman Filter (KF) based Track Fit

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



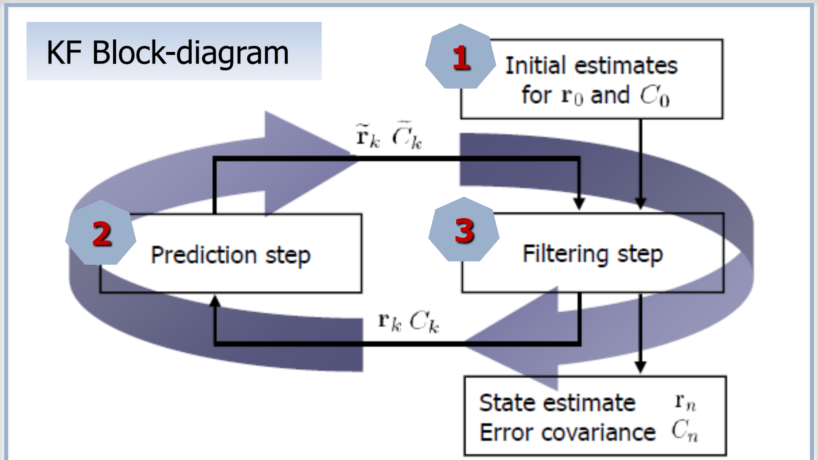
State vector

Position, direction and momentum

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

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.



KF as a recursive least squares method

Nowadays the Kalman Filter is used in almost all HEP experiments

KF Library

Track tools:

- KF track fitter
- KF track smoother
- Deterministic Annealing Filter

KF approaches:

- Conventional KF
- Double precision KF
- Square root KF (2 implementations)
- U-D-Filtering
- Gaussian sum filter

Track propagation:

- Runge-Kutta
- Analytic formula

Prediction step $\hat{x}_{k-1}^{+} \longrightarrow \hat{x}_k^{-}$ $\quad P_{k-1}^{+} \longrightarrow P_k^{-}$

$$P_k^- = F_{k-1} P_{k-1}^+ F_{k-1}^T + Q_{k-1}$$

$$\hat{x}_k^- = F_{k-1} \hat{x}_{k-1}^+$$

Filtering step $\hat{x}_k^{-} \longrightarrow \hat{x}_k^{+}$ $\quad P_k^{-} \longrightarrow P_k^{+}$

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$$

$$P_k^+ = (I - K_k H_k) P_k^-$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k (y_k - H_k \hat{x}_k^-)$$

$$P \rightarrow SS^T$$

Twice a precision in comparison with conventional,
but has more complicated computations = slower.

Prediction step $\hat{x}_{k-1}^{+} \longrightarrow \hat{x}_k^{-}$ $\quad S_{k-1}^{+} \longrightarrow S_k^{-}$

$$\begin{bmatrix} (S_k^-)^T \\ 0 \end{bmatrix} = T \begin{bmatrix} (S_{k-1}^+)^T F_{k-1}^T \\ Q_{k-1}^{T/2} \end{bmatrix}$$

$$\hat{x}_k^- = F_{k-1}\hat{x}_{k-1}^+$$

Filtering step $\hat{x}_k^{-} \longrightarrow \hat{x}_k^{+}$ $\quad S_k^{-} \longrightarrow S_k^{+}$

Implementation I

$$\phi_i = S_{i-1,k}^{+T} H_{ik}^T$$

$$a_i = \frac{1}{\phi_i^T \phi_i + R_{ik}}$$

$$\gamma_i = \frac{1}{1 \pm \sqrt{a_i R_{ik}}}$$

$$S_{ik}^+ = S_{i-1,k}^+ (I - a_i \gamma_i \phi_i \phi_i^T)$$

$$\hat{x}_{ik}^+ = \hat{x}_{i-1,k}^+ + K_{ik}(y_{ik} - H_{ik}\hat{x}_{i-1,k}^+)$$

Implementation II

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$$

$$\tilde{K}_k = K_k (R_k + H_k P_k^- H_k^T)^{T/2}$$

$$\begin{bmatrix} (R_k + H_k P_k^- H_k^T)^{T/2} & \tilde{K}_k^T \\ 0 & (S_k^+)^T \end{bmatrix} = \tilde{T} \begin{bmatrix} R_k^{T/2} & 0 \\ (S_k^-)^T H_k^T & (S_k^-)^T \end{bmatrix}$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k (y_k - H_k \hat{x}_k^-)$$

$P = UDU^T$

$$\begin{bmatrix} 1 & u_{12} & u_{13} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & 0 \\ 0 & 0 & d_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ u_{12} & 1 & 0 \\ u_{13} & u_{23} & 1 \end{bmatrix}$$

Increase precision in comparison with conventional.
Less number of computations than with square root.

Prediction step $U_{i-1} \longrightarrow U_i \quad D_{i-1} \longrightarrow D_i$

$$W = \begin{bmatrix} FU^+ & I \end{bmatrix}$$

$$\hat{D} = \begin{bmatrix} D^+ & 0 \\ 0 & Q \end{bmatrix}$$

$$W = U^- V \qquad D^- = V\hat{D}V^T$$

Filtering step $U^- \longrightarrow U^+ \qquad D^- \longrightarrow D^+$

$$\alpha_i \equiv H_i P_{i-1} H_i^T + R_i \qquad \bar{U}\bar{D}\bar{U}^T = \left[D_{i-1} - \frac{1}{\alpha_i}(D_{i-1}U_{i-1}^T H_i^T)(D_{i-1}U_{i-1}^T H_i^T)^T \right]$$

$$U_i = U_{i-1}\bar{U}$$

$$D_i = \bar{D}$$

Analytic Formula for Track Extrapolation

Allows to control precision and time consumptions.

$$
\begin{aligned}
x' &\equiv t_x \\
y' &\equiv t_y \\
t'_x &= \kappa \cdot (q/p) \cdot \sqrt{1 + t_x^2 + t_y^2} \cdot \left(t_x t_y \cdot B_x - (1 + t_x^2) \cdot B_y + t_y \cdot B_z \right) \\
t'_y &= \kappa \cdot (q/p) \cdot \sqrt{1 + t_x^2 + t_y^2} \cdot \left(\left(1 + t_y^2\right) \cdot B_x - t_x t_y \cdot B_y - t_x \cdot B_z \right) \\
(q/p)' &= 0
\end{aligned}
$$

Taylor expansion

$$
\begin{aligned}
t_x(z_e) &= t_x(z_0) + \sum_{k=1}^{n} \sum_{i_1,\dots,i_k = x,y,z} t_{x_{i_1 \dots i_k}}(z_0) \cdot \left(\int_{z_0}^{z_e} B_{i_1}(z_1) \dots \int_{z_0}^{z_{k-1}} B_{i_k}(z_k) \mathrm{d}z_k \dots \mathrm{d}z_1 \right) , \\
t_y(z_e) &= t_y(z_0) + \sum_{k=1}^{n} \sum_{i_1,\dots,i_k = x,y,z} t_{y_{i_1 \dots i_k}}(z_0) \cdot \left(\int_{z_0}^{z_e} B_{i_1}(z_1) \dots \int_{z_0}^{z_{k-1}} B_{i_k}(z_k) \mathrm{d}z_k \dots \mathrm{d}z_1 \right) , \\
x(z_e) &= x(z_0) + \int_{z_0}^{z_e} t_x(z) \mathrm{d}z , \\
y(z_e) &= y(z_0) + \int_{z_0}^{z_e} t_y(z) \mathrm{d}z ,
\end{aligned}
$$

General method.

$$\frac{\mathrm{d}\mathbf{r}(z)}{\mathrm{d}z} = \begin{pmatrix} t_x \\ t_y \\ \kappa \cdot (q/p) \cdot \sqrt{1+t_x^2+t_y^2} \cdot \left(t_x t_y \cdot B_x - (1+t_x^2) \cdot B_y + t_y \cdot B_z \right) \\ \kappa \cdot (q/p) \cdot \sqrt{1+t_x^2+t_y^2} \cdot \left(\left(1+t_y^2\right) \cdot B_x - t_x t_y \cdot B_y - t_x \cdot B_z \right) \\ 0 \end{pmatrix} \equiv \mathbf{f}(z,\mathbf{r})$$

$$\begin{aligned} \Delta\mathbf{r}_1 &= \mathbf{f}(z_0, \mathbf{r}_0) \cdot \Delta z \ , \\ \Delta\mathbf{r}_2 &= \mathbf{f}(z_0 + \tfrac{\Delta z}{2}, \mathbf{r}_0 + \tfrac{\Delta\mathbf{r}_1}{2}) \cdot \Delta z \ , \\ \Delta\mathbf{r}_3 &= \mathbf{f}(z_0 + \tfrac{\Delta z}{2}, \mathbf{r}_0 + \tfrac{\Delta\mathbf{r}_2}{2}) \cdot \Delta z \ , \\ \Delta\mathbf{r}_4 &= \mathbf{f}(z_0 + \Delta z, \mathbf{r}_0 + \Delta\mathbf{r}_3) \cdot \Delta z \ . \end{aligned}$$

$$\mathbf{r}(z_e) = \mathbf{r}_0 + \left(\frac{1}{6}\Delta\mathbf{r}_1 + \frac{1}{3}\Delta\mathbf{r}_2 + \frac{1}{3}\Delta\mathbf{r}_3 + \frac{1}{6}\Delta\mathbf{r}_4 \right) + O((\Delta z)^5)$$

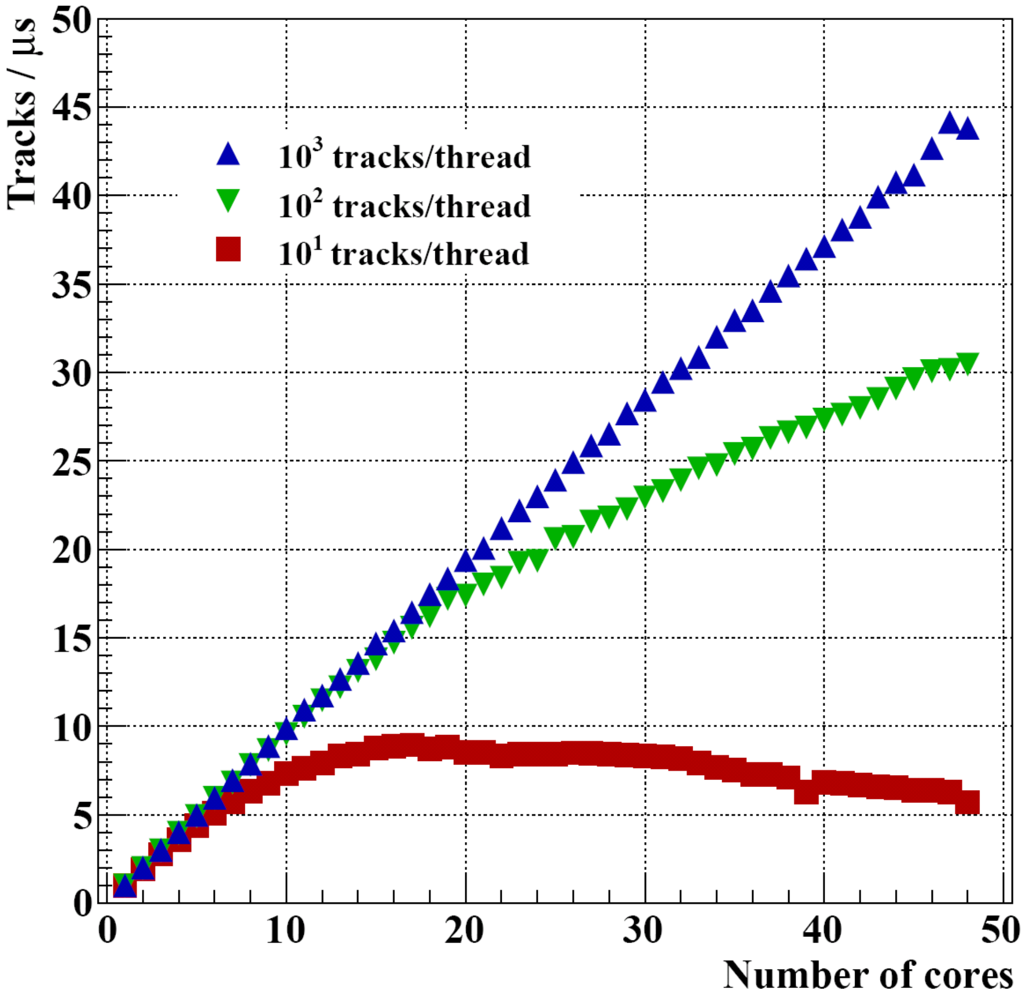
Comparison of Different KF Track Fit Procedures

	Conventional	Conventional double	U-D	Square root I impl	Square root II impl	Square root I impl + Runge-Kutta
P residual, %	1.25	1.09	1.10	1.08	1.09	1.07
Q/P pull	1.40	1.32	1.33	1.31	1.32	1.31
Bad C, 1/event	988.0	4.9	1014.4	9.2	7.8	5.5
Time, μs/track	1.7	3.5	< 3	< 2.5	< 3.5	< 2.5

Statistic: 10 central events
8 STS (no MVD)
Ideal STS
Ideal TrackFinder

Square root KF implementation in single precision significantly improves stability of track fitting in terms of incorrect covariance matrixes (diagonal elements less then zero).

Scalability of SIMD KF Fit Benchmark



Measure tracks throughput rather than time per track.

Given n threads each filled with 10^m tracks, run them on specific n logical cores with 1 thread per 1 core.

For small groups of tracks the overhead becomes significant, while large groups of tracks use CPU more efficient.

opladev35 (CERN, Openlab) with 4 CPUs AMD E6164HE, 12 cores per CPU, 1.7 GHz; TBB

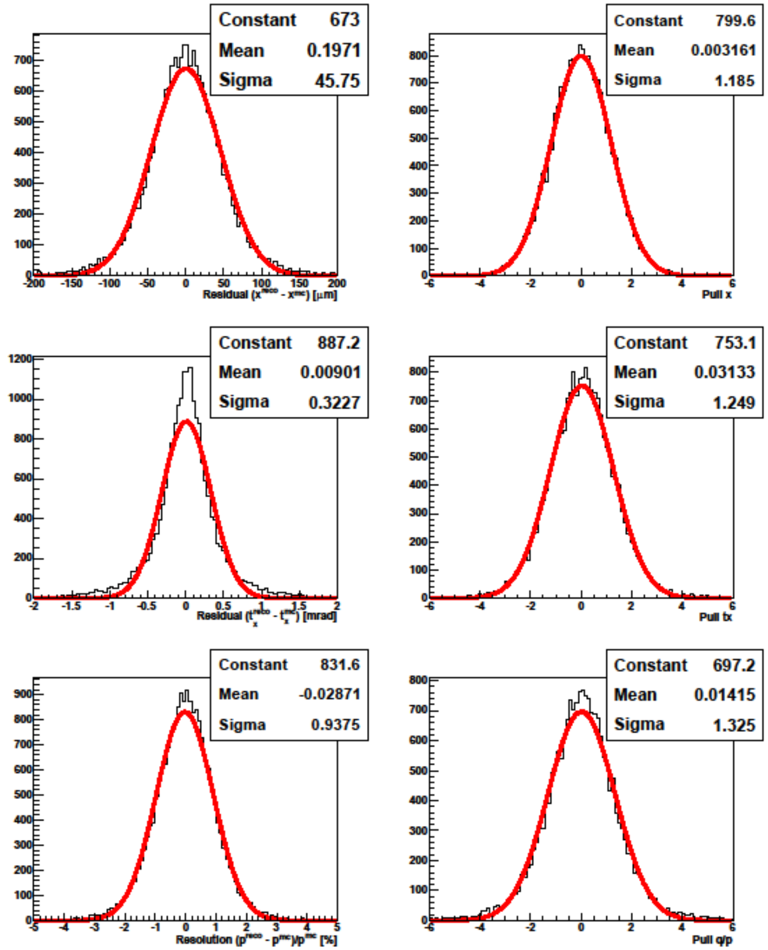
Strong many-core scalability for large groups of tracks.
Fitting speed of 22 ns/track/node has been achieved.

Array Building Blocks (ArBB) allows to avoid a lot of inconveniencies of parallel programming. It should be very useful for the event reconstruction.

Implementation of KF based on ArBB was the first step for the track finders ArBB-zation.

SIMD KF fit benchmark with ArBB has been implemented by Intel.
Comparison with SIMD KF fit benchmark based on Vector classes (Vc) was done.

	Vc		ArBB	
Cores	1	16	1	16
Time, µs	0.42	0.05	0.43	0.06



Tests were performed on the lxir039 computer with 2 Xeon X5550 processors having 8 cores in total at 2.7 GHz

With H. Pabst

KF track fit based on ArBB has been implemented by Intel.

Deterministic Annealing Filter (DAF)

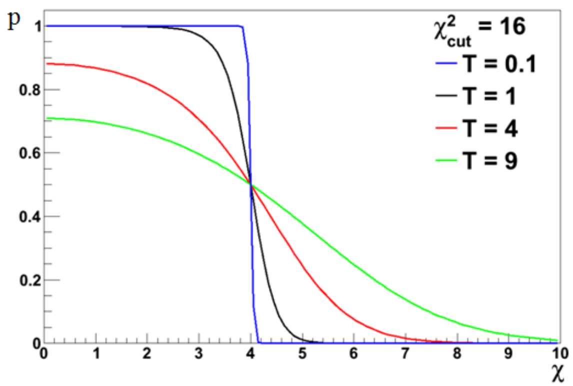
Task: reduce an influence of attached distorted or noise hits on the reconstructed track parameters.

- DAF has been implemented within SIMD KF track fit package
- The KF mathematics has been modified to include weights

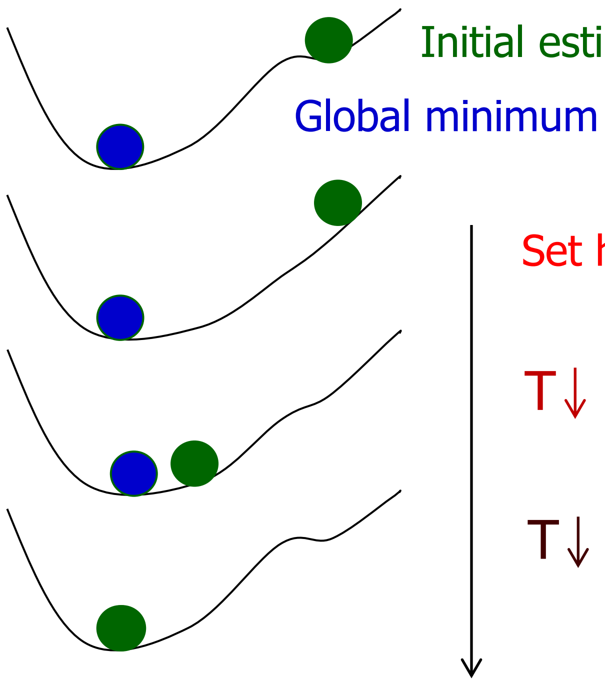
DAF algorithm:

- A weight is introduced to each hit

$$p = \frac{1}{1 + \exp\left((\chi^2 - \chi^2_{\text{cut}})/(2T)\right)}$$



- Algorithm is iterative, with each iteration T is decreasing, weight is recalculated using smoothed track parameters from the previous iteration



R. Frühwirth and A. Strandlie, Track Fitting with ambiguities and noise: a study of elastic tracking and nonlinear filters. Comp. Phys. Comm. 120 (1999) 197-214.

DAF and Noise Hits Rejection

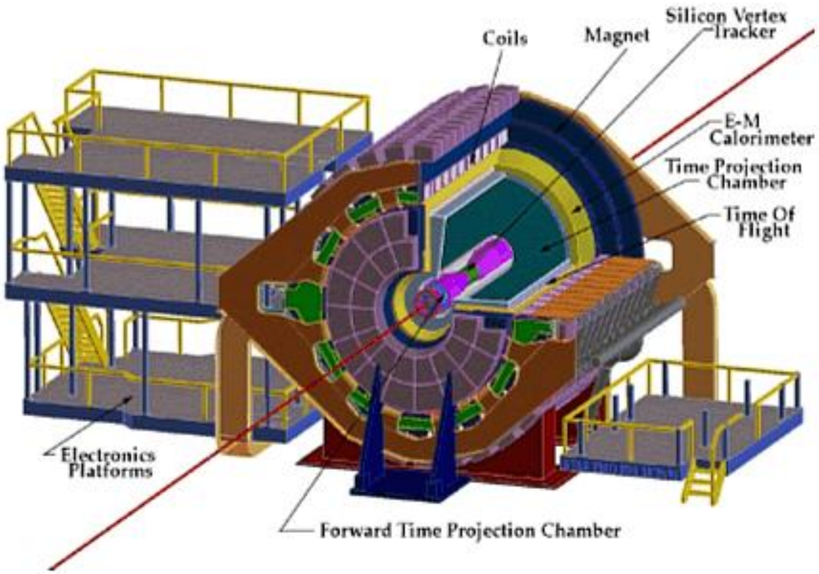
- The hit on the 4th STS station was displaced by a certain amount of the hit error (σ_{hit} = 17 μm) from the MC position

- The percentage of rejected hits was calculated. For the 4th station it should be 100%, for other – 0%

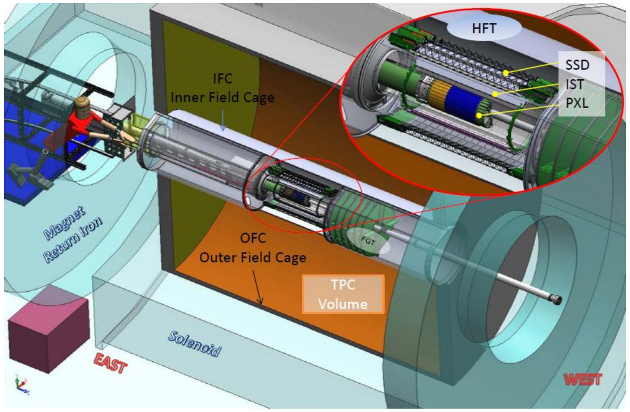
Rejection probability, %						
station		unshifted	5 σ_{hit}	10 σ_{hit}	20 σ_{hit}
MVD	1	0.4	0.4	0.4	0.4
	2	0.7	0.7	0.7	0.7
STS	1	0.3	0.3	0.3	0.3
	2	0.4	0.4	0.4	0.4
	3	0.4	0.7	0.8	0.5
	4	**0.5**	**43.9**	**85.0**	**98.7**
	5	0.5	1.6	1.6	0.8
	6	0.6	0.6	0.6	0.6
	7	0.6	0.6	0.6	0.6
	8	0.1	0.1	0.1	0.1

In collaboration with R. Frühwirth (HEPHY, Austria) and A. Strandlie (Uni-Oslo, Gjøvik University College, Norway)

The STAR experiment



- Collider experiment at RHIC, BNL
- Main detector – TPC
- 10^4 AuAu collisions/sec
- 5000 charged particles/collision
- New HFT detector (2014)



High Level Trigger (HLT):
- allows to pick out events of physics interest
- reduces the rate to tape
- reduces the time of offline processing
- plays a key role in online QA

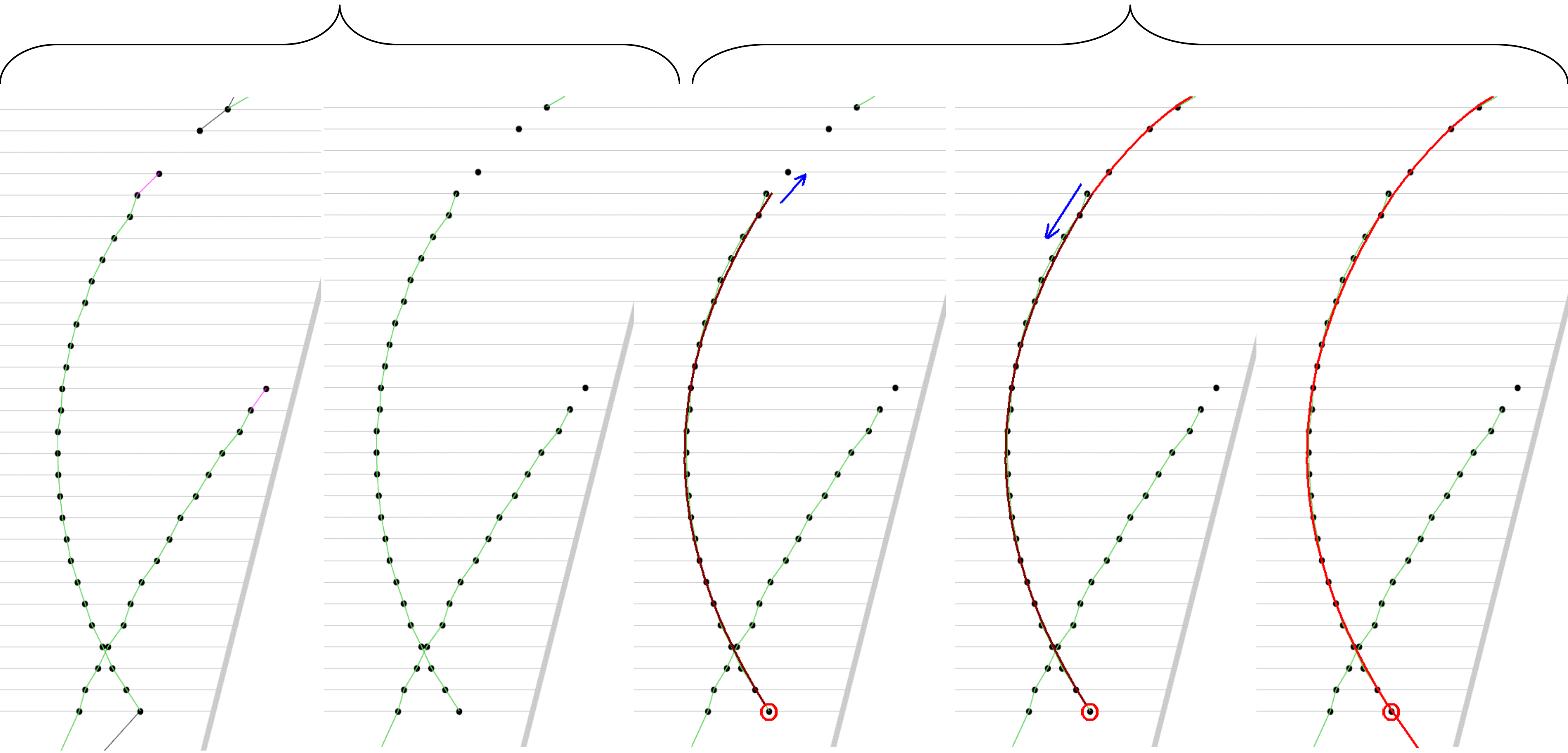
HLT farm:
- 24 PCs for TPC sector reconstruction
- 8 CPU cores per machine
- data acquisition and hit reconstruction
- tracking for HLT

Upgrade the reconstruction algorithms for:
- vectorization
- multi-threading
- many-core systems

STAR Reconstruction of Track Segments

Find neighbors   Clean neighbours   Fit chain   Extrapolate up   Extrapolate down

neighbours   segment

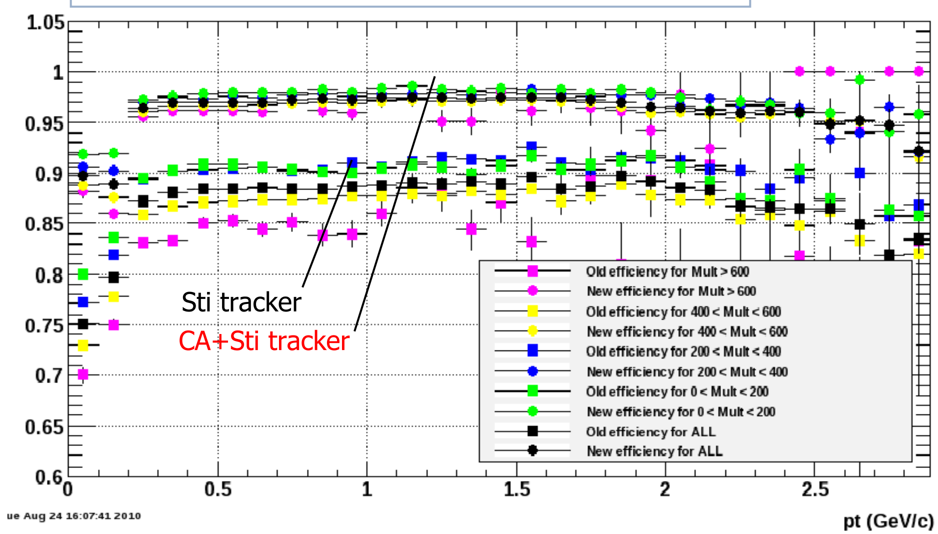


Comparison Baseline Reconstruction (Sti) with CA Tracking (CA+Sti)

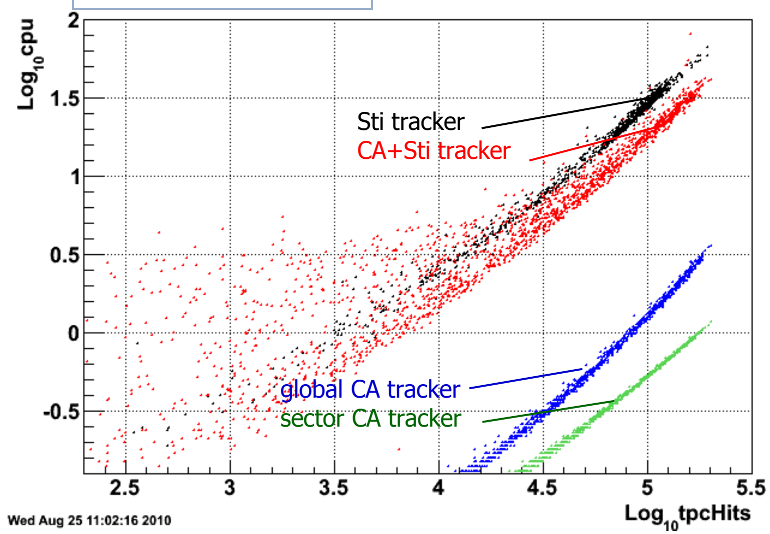
Real Au-Au 200 GeV/n data.

(fitted within 0.2–2.1 GeV)	Global tracks		Primary tracks	
	Sti	**CA+Sti**	Sti	**CA+Sti**
Mult < 200	90.3%	**97.7%**	97.3%	**99.3%**
200 < Mult < 400	90.2%	**97.5%**	97.0%	**99.1%**
400 < Mult < 600	86.9%	**96.6%**	96.0%	**98.9%**
Mult > 600	84.4%	**96.2%**	95.4%	**98.9%**
All	88.1%	**97.1%**	96.4%	**99.1%**

Track reconstruction efficiency for global tracks



CPU time per event



Efficiency for global tracks has been increased on 9%.
CA Tracker takes 10% of the full event reconstruction time. CA+Sti is ~50% faster than Sti alone.

STAR TPC CA Track Finder Time Optimization

HLT requires a track reconstruction algorithm with speed about 50 ms.

Reconstructable track:
Number of MC hits ≥ 10

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

Efficiency and ratios, %		
	Aug 2010	Dec 2010
Ref Set	96.7	96.6
All Set	88.6	88.6
Clone	9.9	10.6
Ghost	29.1	12.6
Tracks/ev	660	659
Time/ev, ms	178	47

Residuals and resolutions		
	Aug 2010	Dec 2010
x, mm	0.50	0.48
y, mm	0.96	0.92
sin φ, 10^{-3}	4.7	4.5
dz/ds, 10^{-3}	6.1	5.6
p_t, %	2.6	2.2

$tg\ \varphi = dy/dx$
$ds^2 = dy^2 + dx^2$

Au+Au 200 AGeV; 100 MC events

The execution time of STAR TPC CA track finder is 47 ms
(STAR HLT requires 50 ms).

STAR TPC CA Track Finder with ArBB

- Vector classes (Vc) has been replaced by ArBB
- There are still some issues:
  - The algorithm was simplified
  - Only a scalar execution works
  - The algorithm is not yet optimized at all
  - Data structure should be optimized for parallel implementation

Reconstructable track:
Number of MC hits ≥ 10

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

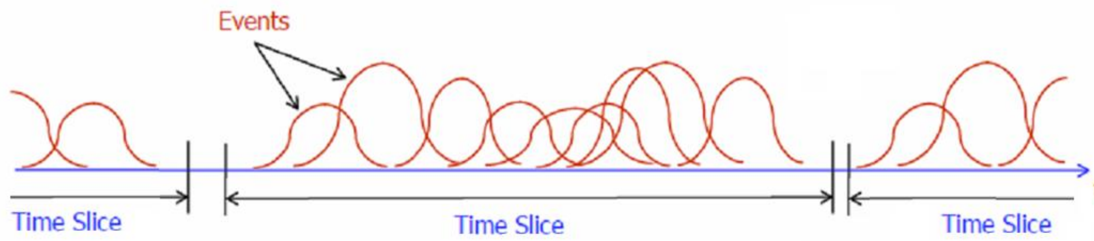
Efficiency and ratios, %		
	Vc	ArBB
Ref Set	94.8	95.1
All Set	82.3	82.5
Clone	1.8	1.6
Ghost	7.7	7.7
Tracks/ev	812	814
Time/ev, s	0.25	266.94

Au+Au 200 AGeV; 5 MC events

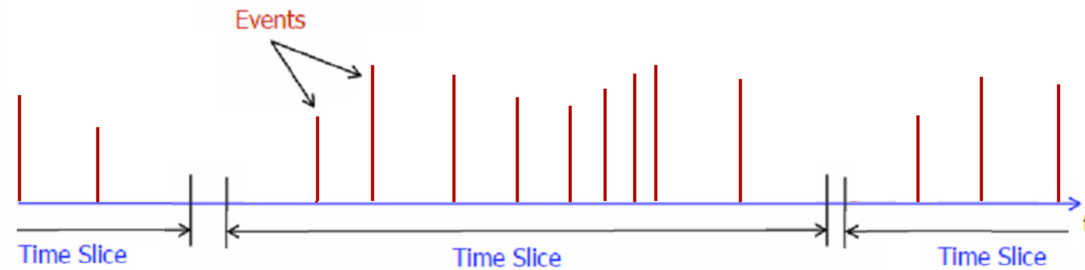
ArBB for track reconstruction algorithms is under investigation.

The beam in CBM will have no bunch structure, but continuous. Measurements in this case will be 4D (x, y, z, t).

Reconstruction rather time slices then events will be needed.



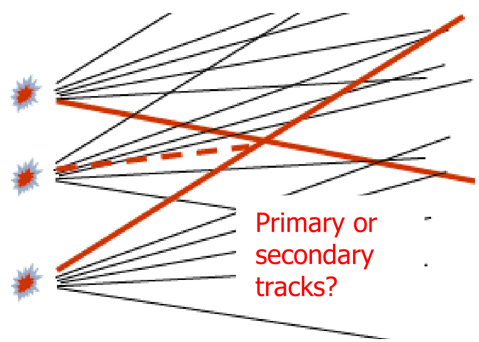
First idealized 4D STS reconstruction with CA track finder has been investigated. Discrete time was used.



✓The same efficiency
✓Slight increase of the processing time with larger size of the time slices

Next

- Reconstruction with more realistic simulation
- Event topology based on time and vertices
- Streaming data reconstruction



Summary

✓ CA track finder is applied both for the CBM and for the STAR experiment

✓ CBM CA track finder is stable with respect to the detector inefficiency

✓ The execution time of CBM CA track finder is 200 ms per central event
✓ The execution time of STAR TPC CA track finder is 47 ms (STAR HLT requires 50 ms)

✓ Strong scalabilities for the SIMD KF track fitter and the CBM CA track finder on the many-core platforms (up to 48 cores)

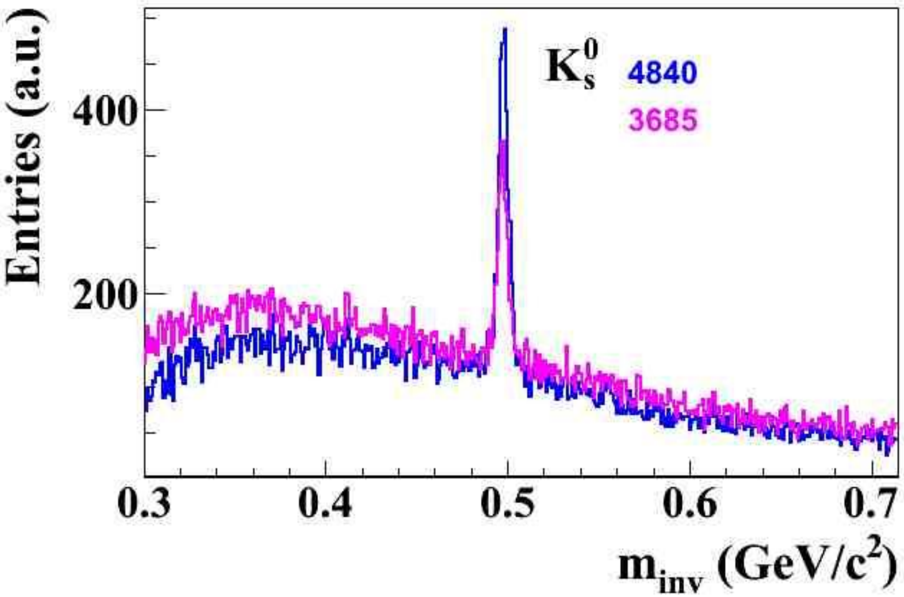
✓ ArBB for reconstruction algorithms is under investigation in collaboration with Intel

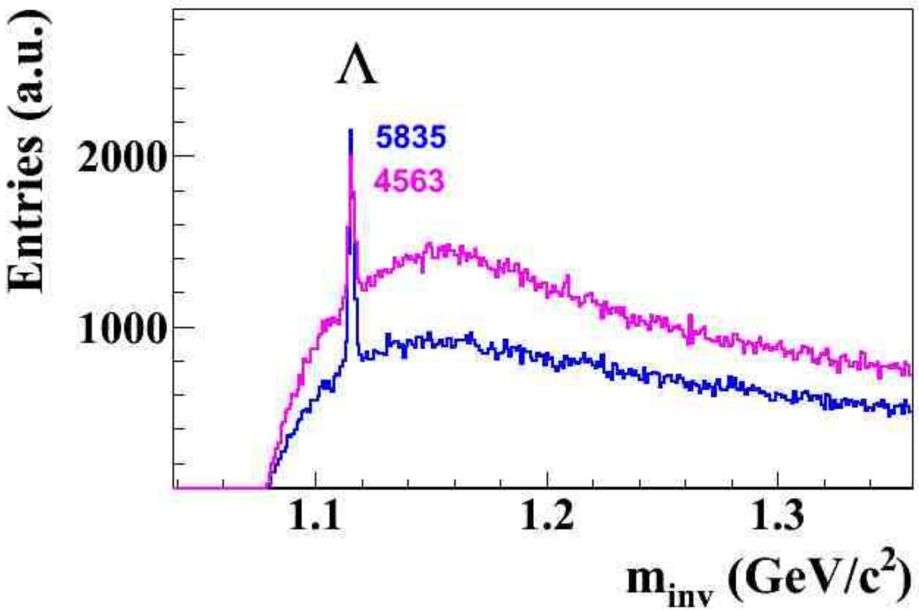
✓ Investigation of 4D reconstruction has been started

- Back up

Physics Tests

Physics tests of the improved algorithm were done with Λ baryons and K_s^0 mesons.
Inefficiencies of STS detector of 0% and 10% have been investigated.





Results are obtained by Y.Vassiliev.

With increasing of the detector inefficiency from 0% to 10% S/B ratio is decreased by a factor of 1.25 for K_s^0 and by a factor of 2.5 for Λ.

CA track finder has been investigated and improved with respect to STS detector inefficiency.

covariance matrix -> square root of covariance matrix

$$\mathbf{P = SS^T}$$

Transport example

$$\mathbf{P' = F\,P\,F^T}$$

$$P = \begin{pmatrix} 1 & 0 \\ 0 & 10^{10} \end{pmatrix} \quad F = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad P' = \begin{pmatrix} 10^{10}+1 & 10^{10} \\ 10^{10} & 10^{10} \end{pmatrix} = \begin{pmatrix} 10^{10} & 10^{10} \\ 10^{10} & 10^{10} \end{pmatrix}$$

Lose information!

$$\mathbf{S' = F\,S}$$

$$S = \begin{pmatrix} 1 & 0 \\ 0 & 10^{5} \end{pmatrix} \quad F = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad S' = \begin{pmatrix} 1 & 10^{5} \\ 0 & 10^{5} \end{pmatrix}$$

No problem with precision

- Further are just saved drafts