Time-based Track Finding for CBM

V. Akishina, I. Bezshyiko, I. Kisel, M. Zyzak

22 September 2014
FAIRNESS

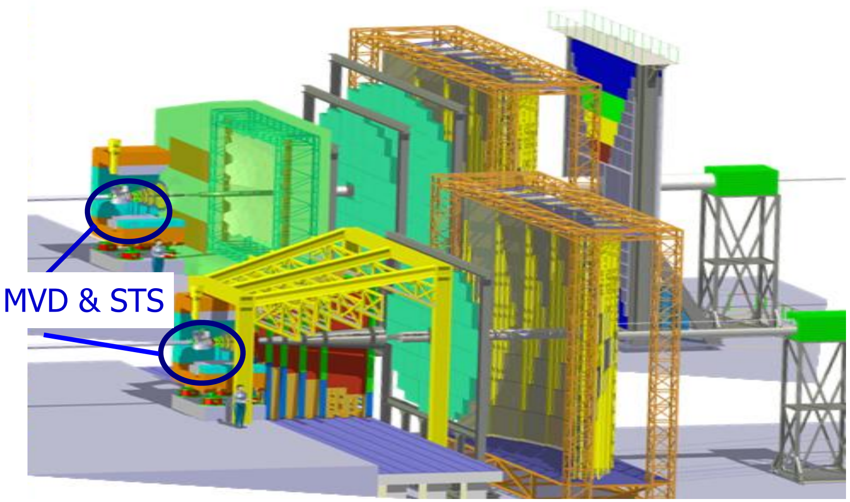
Outline

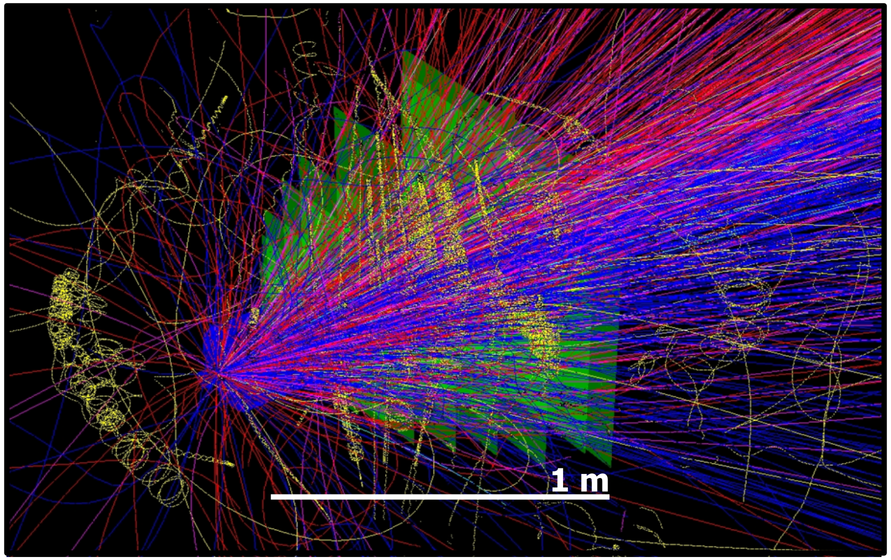
1. **Time-slices**
    - Why time-slices in reconstruction?
    - Event building
    - Time-slice length: border problem
2. **Time-based CA Track Finder**
    - 3D CA Track Finder at High Track Multiplicities
    - From 3D to 4D Track Reconstruction
    - Performance of 4D Track Reconstruction
    - Scalability

- **Results of Event building**

Tracking Challenge in the CBM Experiment

- ✓ Fixed-target heavy-ion experiment
- ✓ 10^7 Au+Au collisions/sec
- ✓ ~ 1000 charged particles/collision, only 1 m length
- ✓ Double-sided strip sensors - 85% fake combinatorial space points in STS
- ✓ Non-homogeneous magnetic field
- ✓ Reconstruction of the full event topology is required in the first level trigger

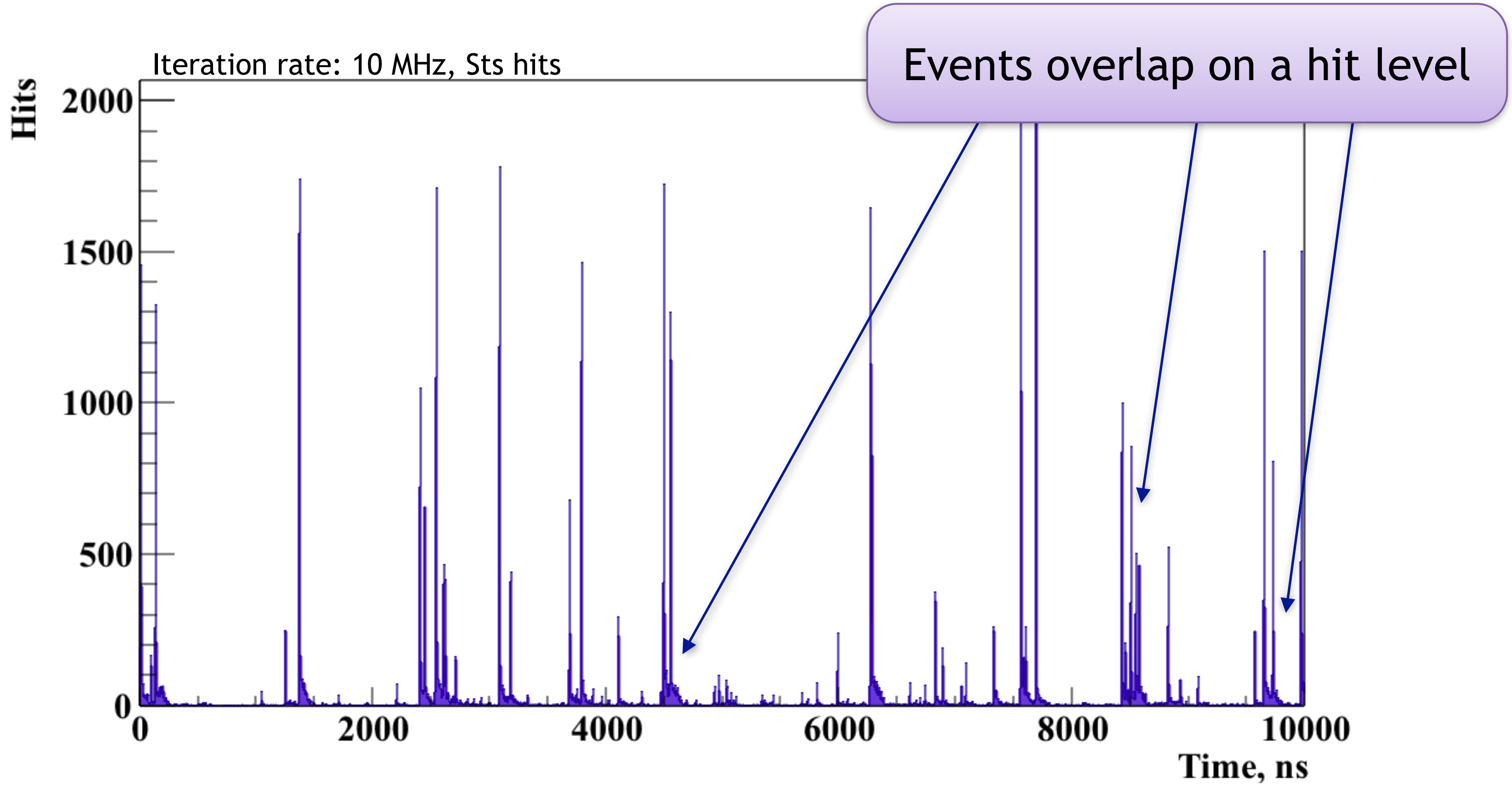




The speed and efficiency of the reconstruction algorithm are crucial

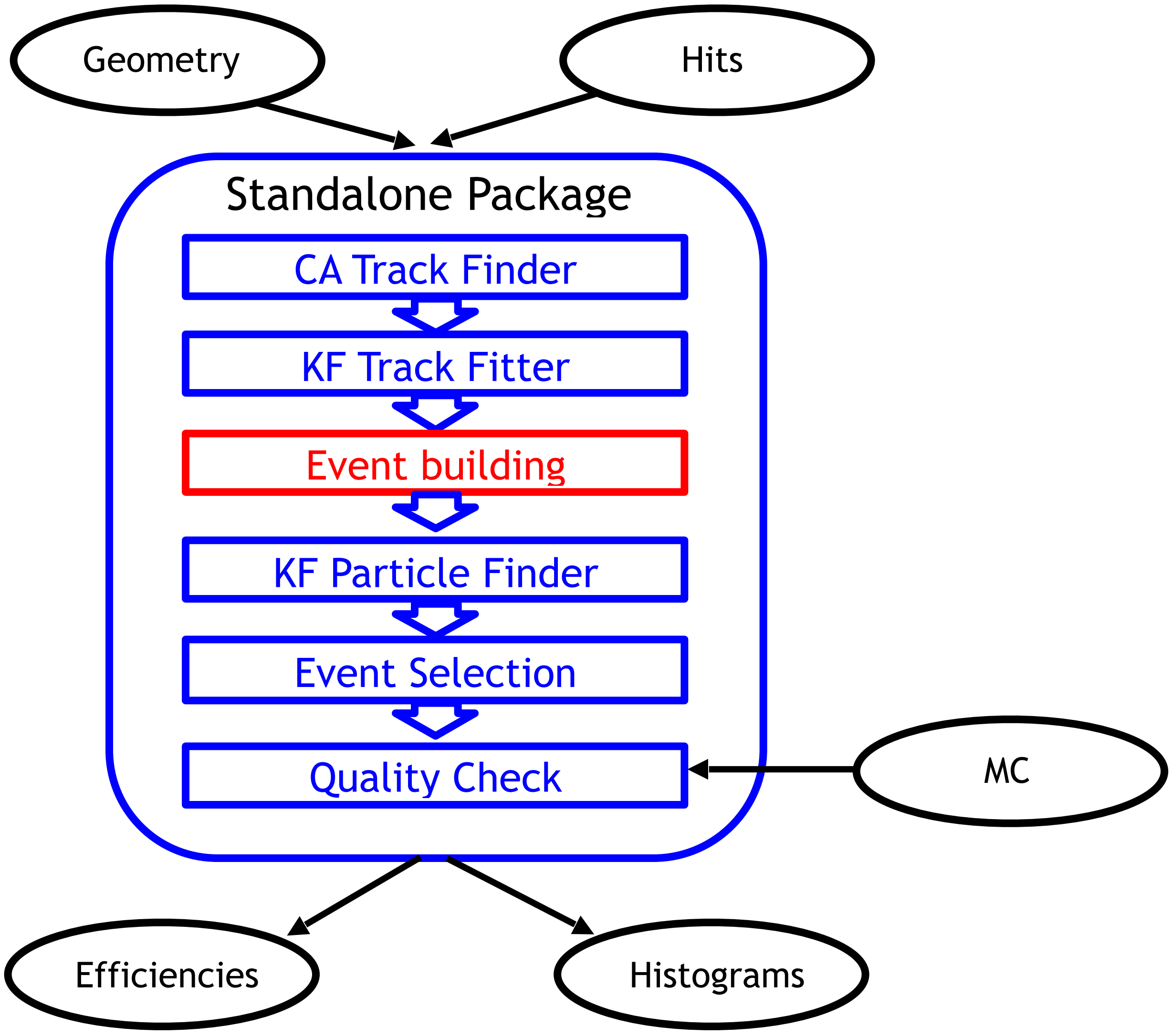
Time-slice based reconstruction

- The CBM beam: no bunch structure → Time-slice reconstruction rather than event-by-event
- Interaction rate up to 10 MHz — Time-based tracking: 4D (x, y, z, t)



Correct procedure of event building from time-slices is crucial for right physics interpretation

First Level Event Selection Package



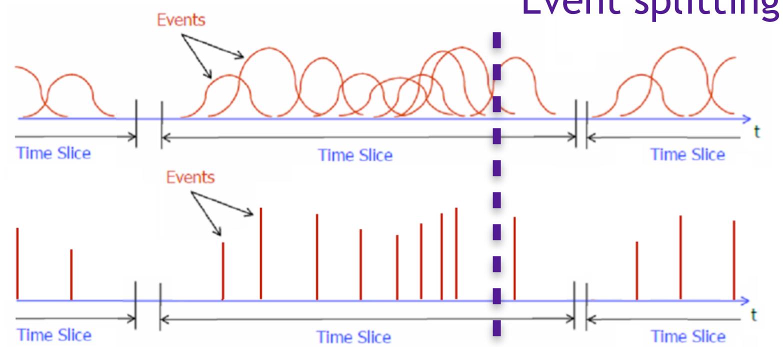
Event building as a part of the FLES package

Time-slice Length: Border Problem

The time-slice length should meet the requirements:

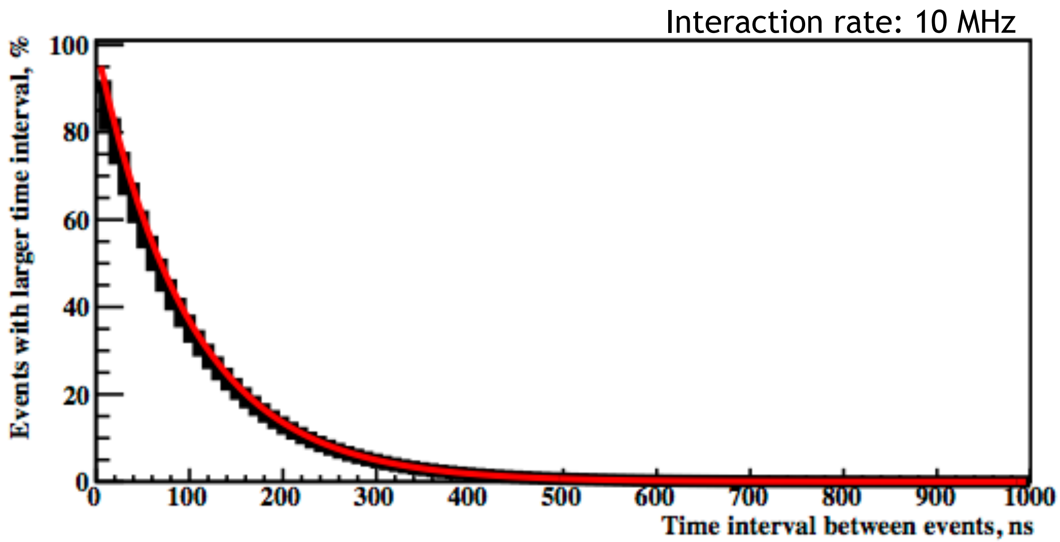
- avoid splitting of the event into two parts by the time-slice border
- maximise performance on modern many-core computing architectures

Event splitting



The beam particle flow is close to Poisson process.
The number of events at any time defined by the Poisson distribution.
Thus, time intervals between neighbouring events at interaction rate I_N is distributed by:

$$f(t) = e^{-t \cdot I_N} \cdot I_N,$$



The probability to have a time interval between events large than a given value:

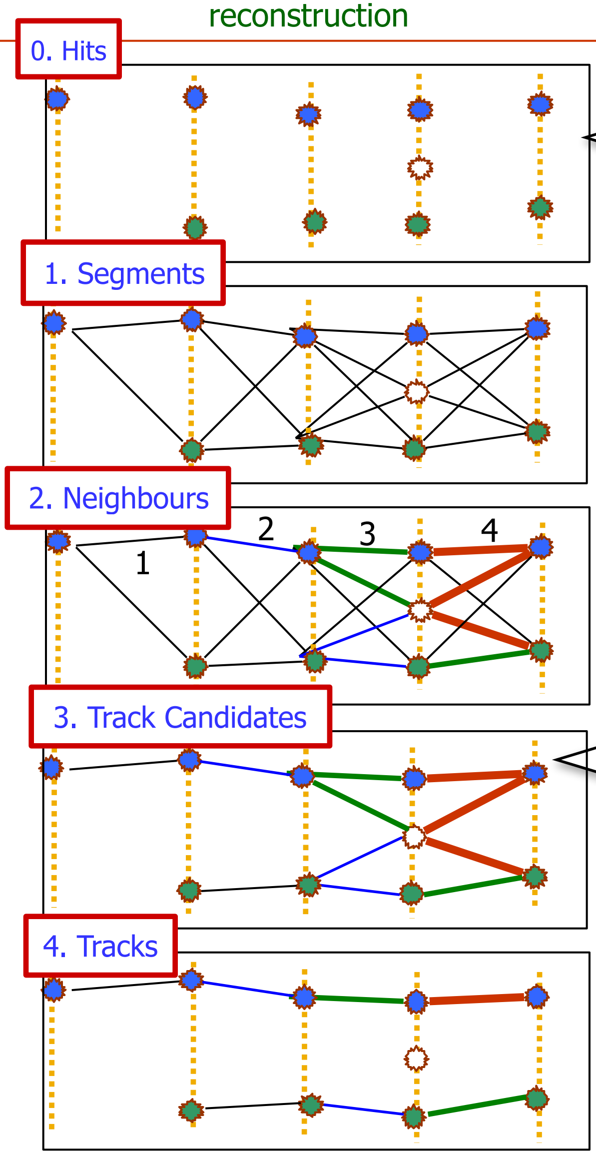
IR, Hz	100 ns	200 ns	300 ns
10	*1.0*	*1.0*	*1.0*
10	*0.99*	*0.98*	*0.97*
10	*0.91*	*0.82*	*0.74*
10	*0.37*	*0.14*	*0.05*

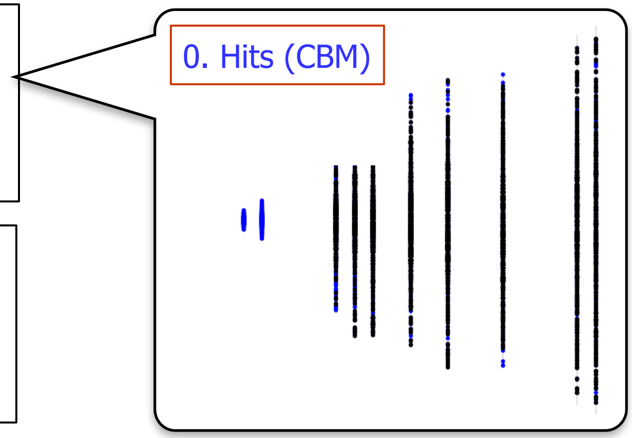
each 20th interval between events is larger than 300 ns at 10 MHz ⇒ safe cutting in time-slices starting from 100 events length without event splitting, no need to duplicate border regions

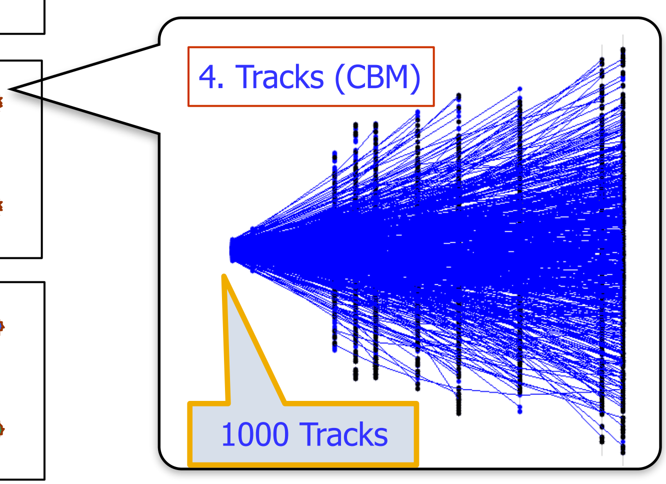
No time-slice border problem up to 10 MHz

Cellular Automaton Based Track Finder

CA illustration: Application to straight tracks reconstruction







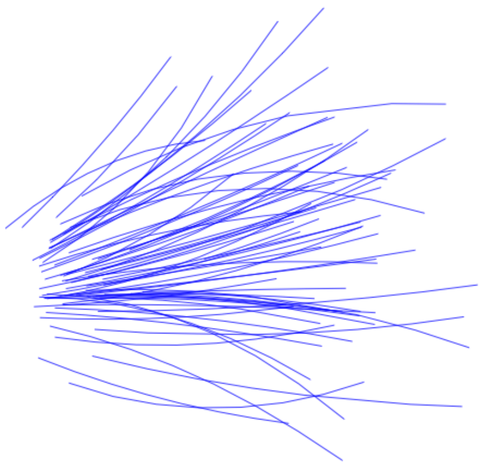
Cellular Automaton:
- local w.r.t. data
- intrinsically parallel
- simple
- very fast

Perfect for many-core CPU/GPU !

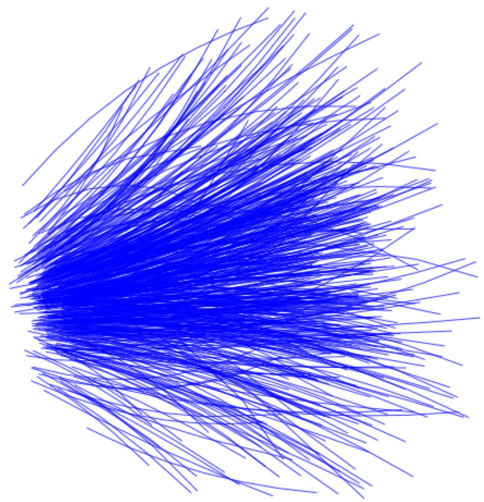
Cellular Automaton:
1. Build short track segments - triplets.
2. Mark possible neighbours while building triplets
3. Connect according to the track model, estimate a possible position on a track.
3. Tree structures appear, collect segments into track candidates.
4. Select the best track candidates.

CA Track Finder at High Track Multiplicity

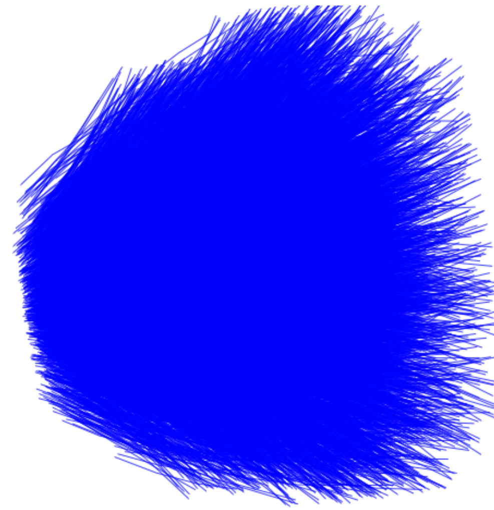
A number of minimum bias events is gathered into a group, which is then treated by the track finder as one event



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



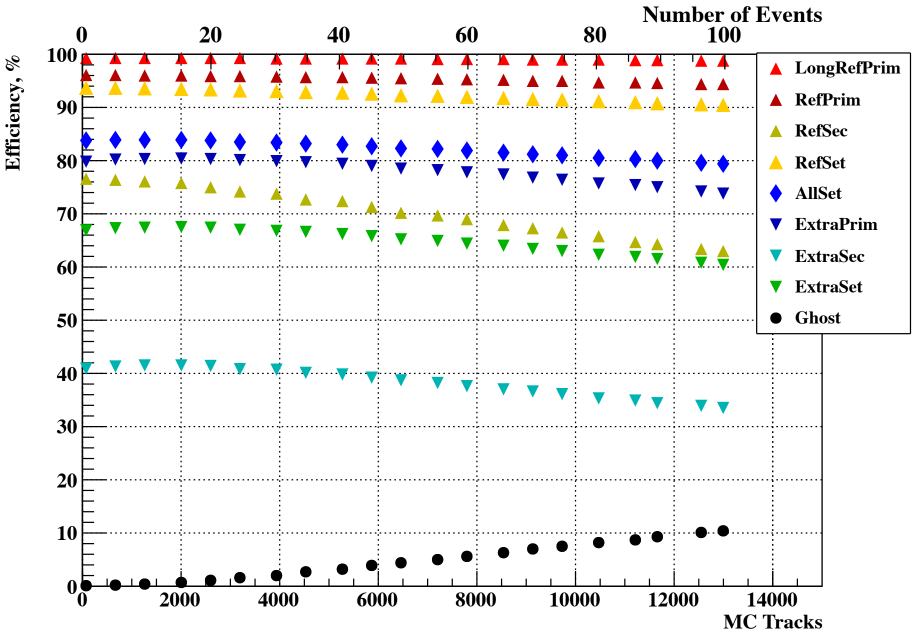
5 minimum bias events
$\langle N_{reco} \rangle = 572$

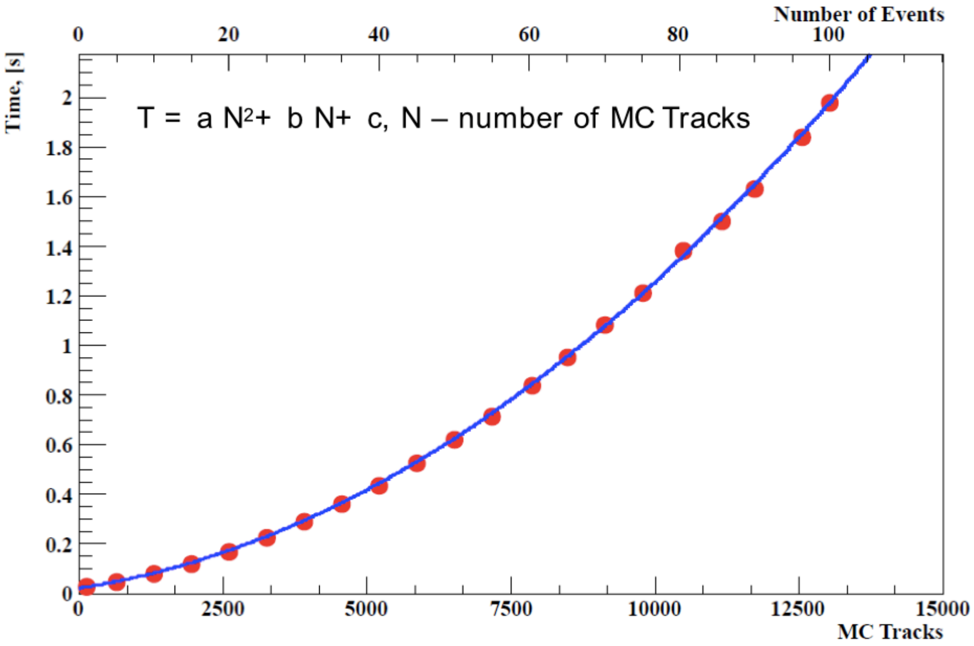


100 minimum bias events
$\langle N_{reco} \rangle = 10340$

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

Efficiency and Time vs. Track Multiplicities





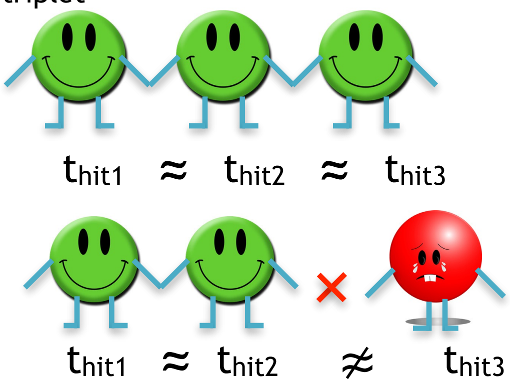
Stable reconstruction efficiency and time as a second order polynomial up to 100 minimum bias events in a group

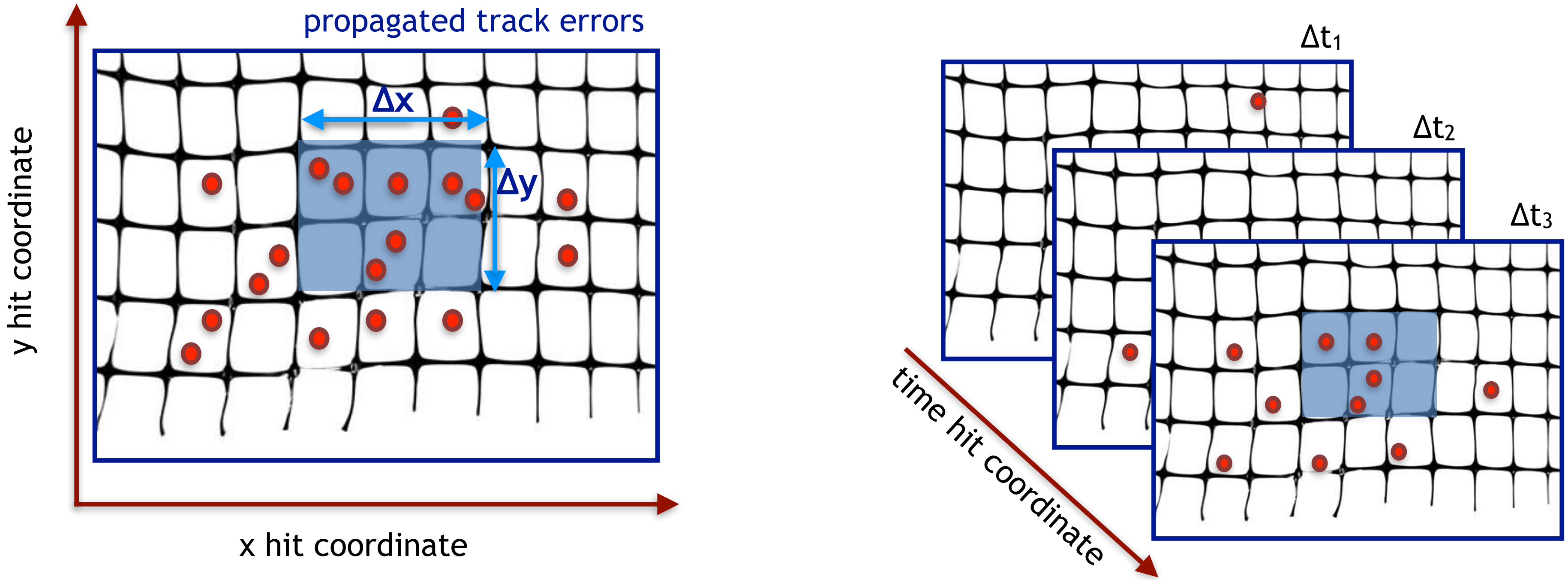
Time-based CA Track Finder

How to use time information in tracking?

- Triplets are build from the hits with the same time measurement within 3 σ of detector precision
- Fast access to the hits is provided by time-based structure: hits are sorted by time and space coordinates and stored into the time-based grid

Hits time measurement have to be the same within detector precision to build a triplet





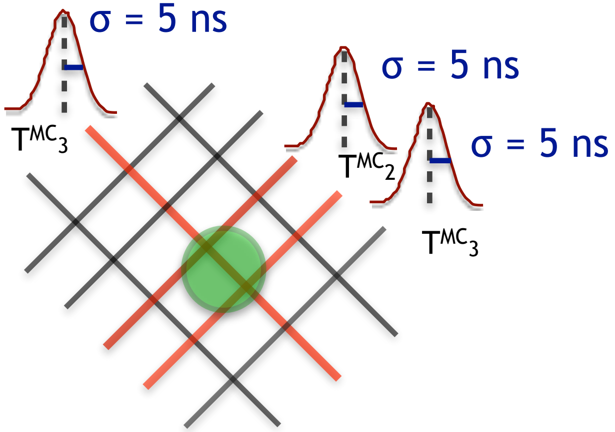
Variable time step of grid: 4D tracking in a 3D-style approach

Time Measurement Simulation

- The time is measured for each strip in cluster independently : $t_1, t_2, \ldots, t_N$
- The time measurement t_n is obtained by smearing the MC time stamp according Gauss to distribution with σ = 5 ns representing detector resolution
- The hit time measurement is defined as average over all strip measurements in front and back clusters:

$$t_{hit} = \sum_{n=1}^{N} t_{n\ strips}$$

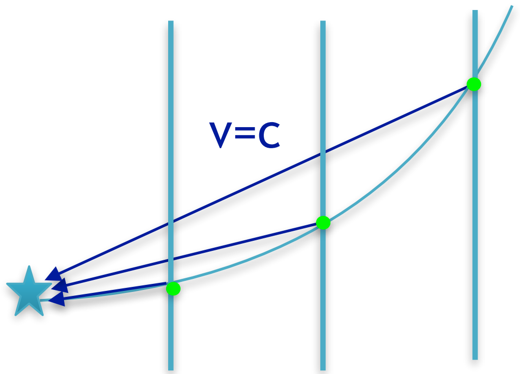
Hit time as an average over smeared strip time measurement of two clusters



- Hit time measurements are propagated to the primary vertex by a straight line assuming speed of light
- Track time is defined in the primary vertex as an average over its propagated hits time measurements

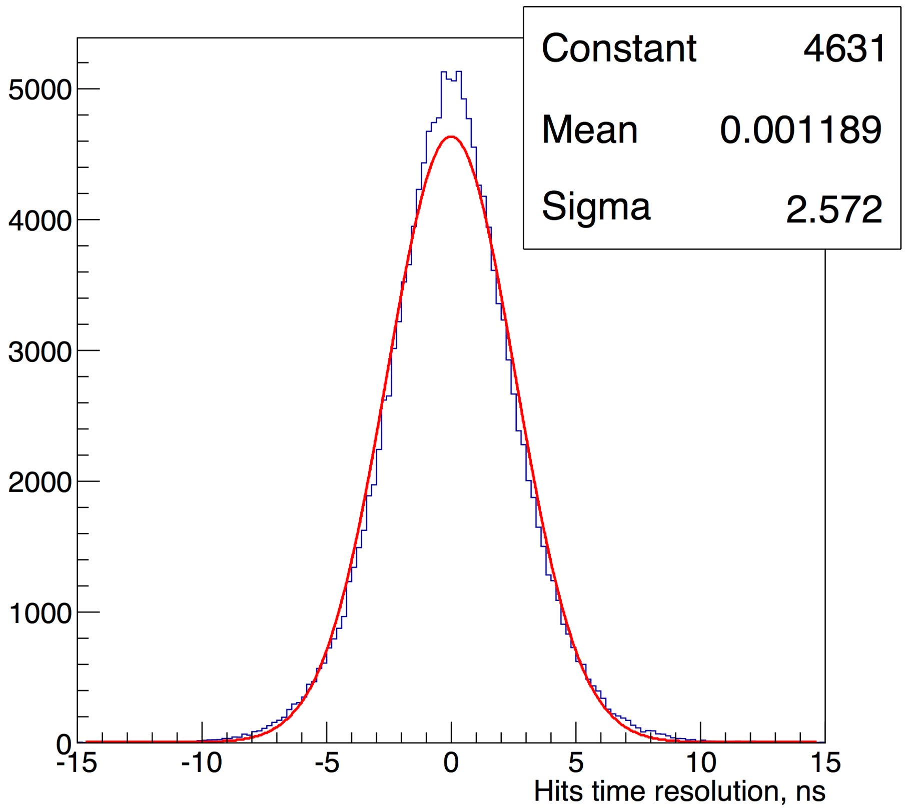
$$t_{track} = \sum_{n=1}^{N} t_{n\ hits}$$

Track time as an average over its hits time measurements propagated along straight line to primary vertex with v = c

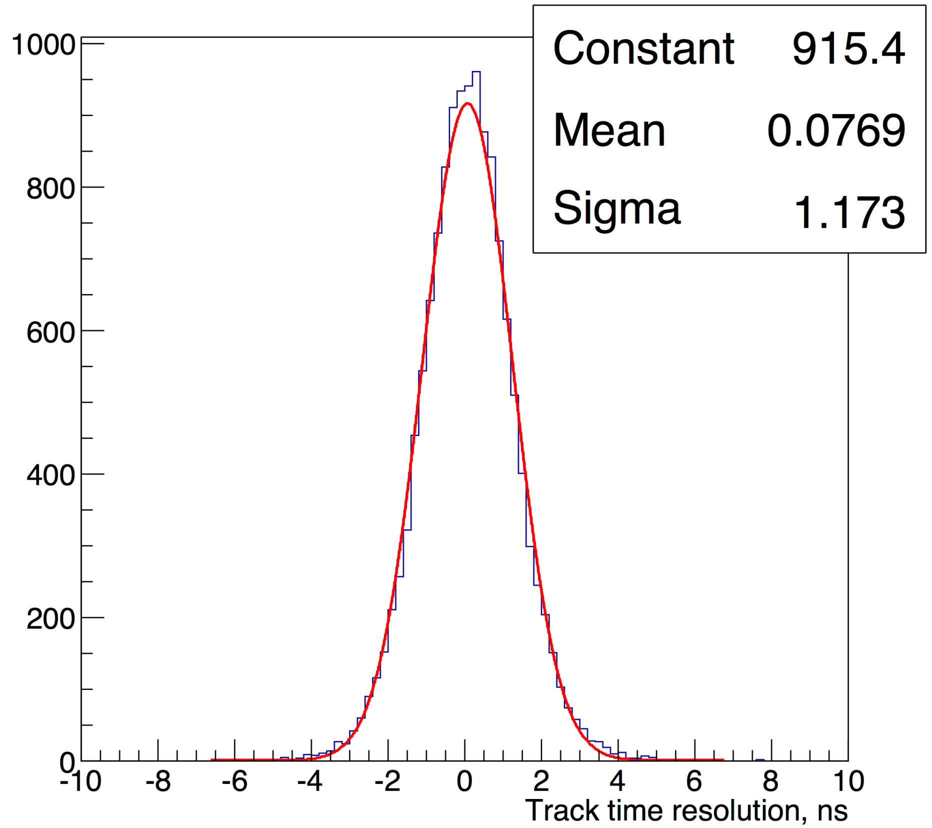


Time Resolution: Hits and Tracks

Hit time resolution: 2.6 ns



Track time resolution: 1.2 ns



Time-based CA Track Finder

Reconstruction of grouped event of 100 mbias AuAU 25AGeV

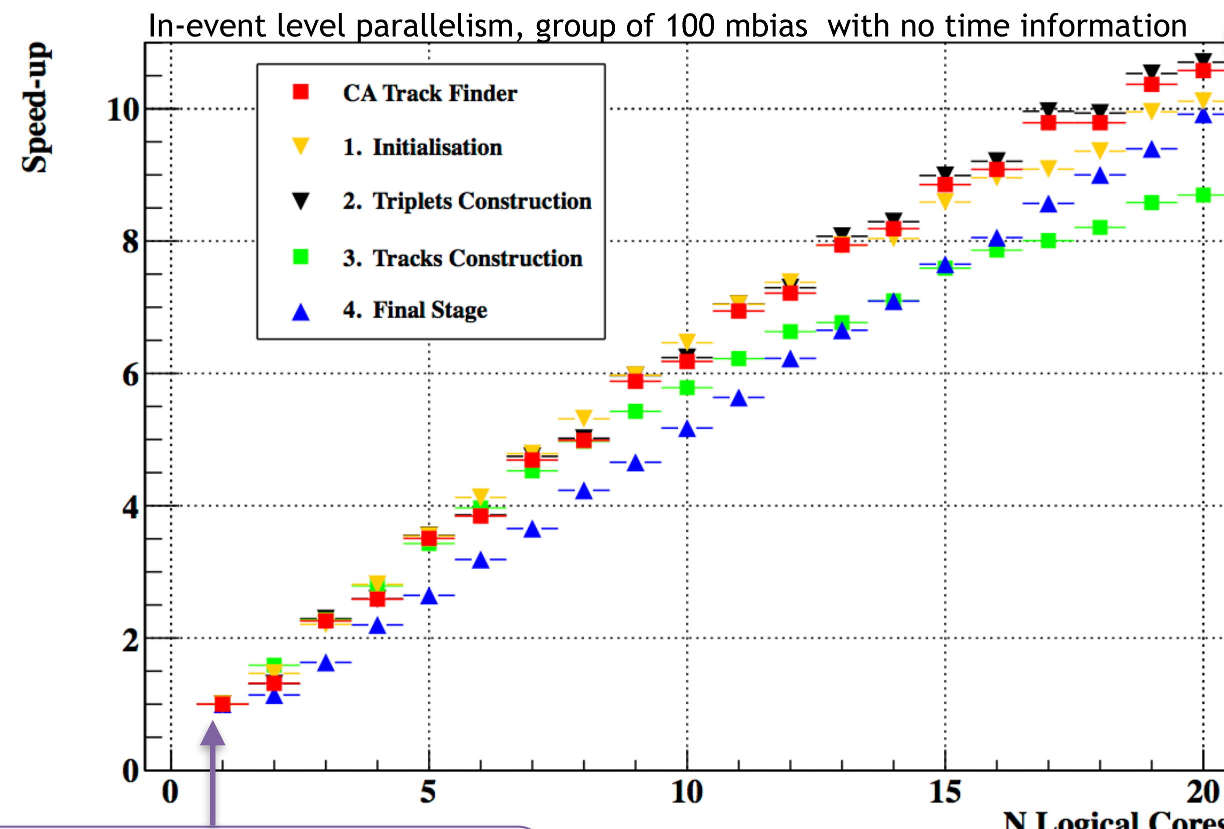
	3.5D CA Track Finder	4D CA Track Finder
Fast primary tracks, %	94%	94%
All tracks, %	80.6%	83%
Ghost, %	7.3%	0.2%
Reconstructed MC tracks	10 060	10 355
Time on one core	3 150 ms	850 ms

Time-based tracking shows high efficiency, kills ghosts and works 3.7 times faster

3D CA Track Finder Scalability within CPU

Algorithm Step	% of total execution time
Initialisation	2%
Triplets construction	90.4%
Tracks construction	4.1%
Final stage	3.4%

Total CA time = 298 ms



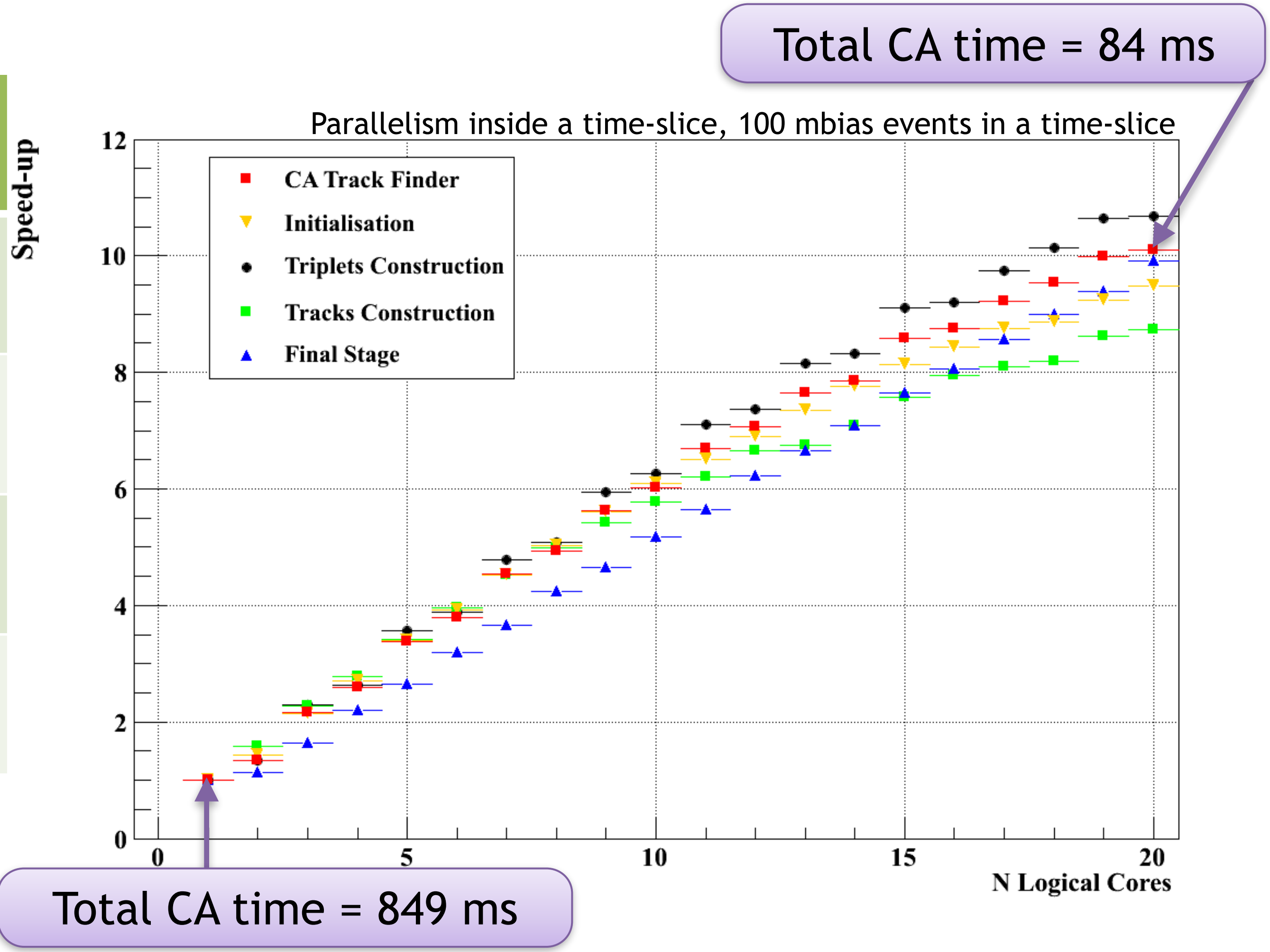
Total CA time = 3150 ms

Speed-up factor 10.6 Theoretically achievable factor: 13

4D CA Track Finder Scalability within CPU

100 mbias event time-slice reconstructed in parallel in 84 ms

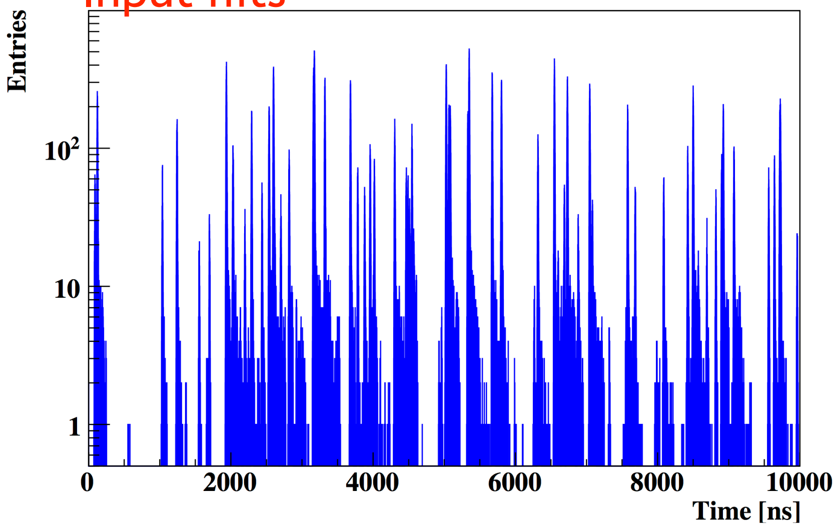
Algorithm Step	% of total execution time
Initialisation	8%
Triplets construction	64%
Tracks construction	15%
Final stage	13%

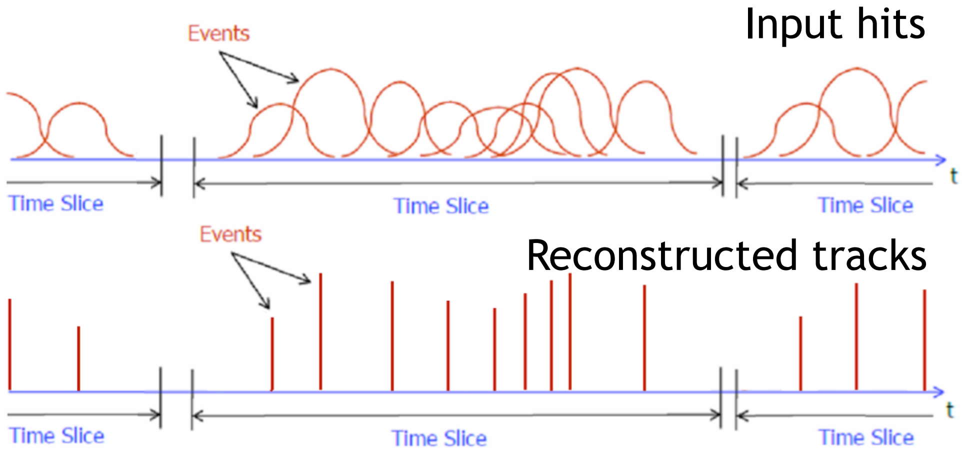


Speed-up factor 10.1 Theoretically achievable factor: 13

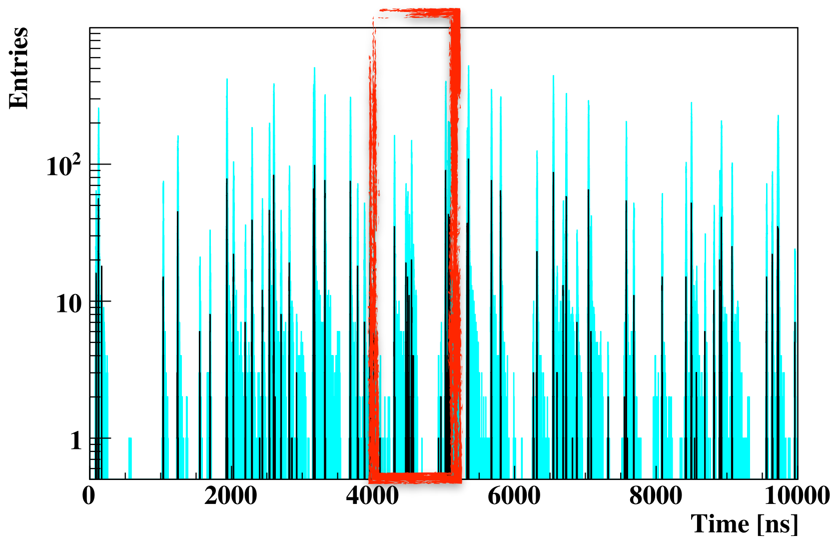
Time-based Track Reconstruction

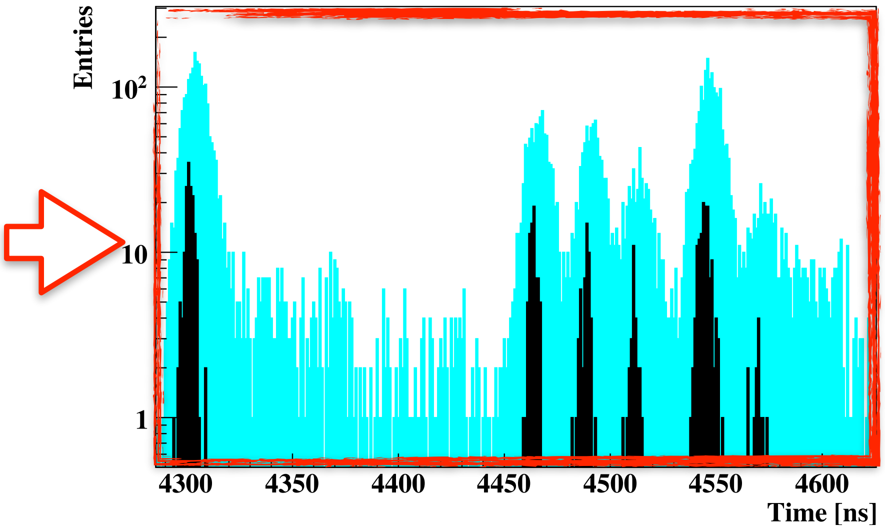
Input hits





Reconstructed tracks





Reconstructed tracks are clearly clustered in groups representing original events

Event Building at IR = 10 MHz

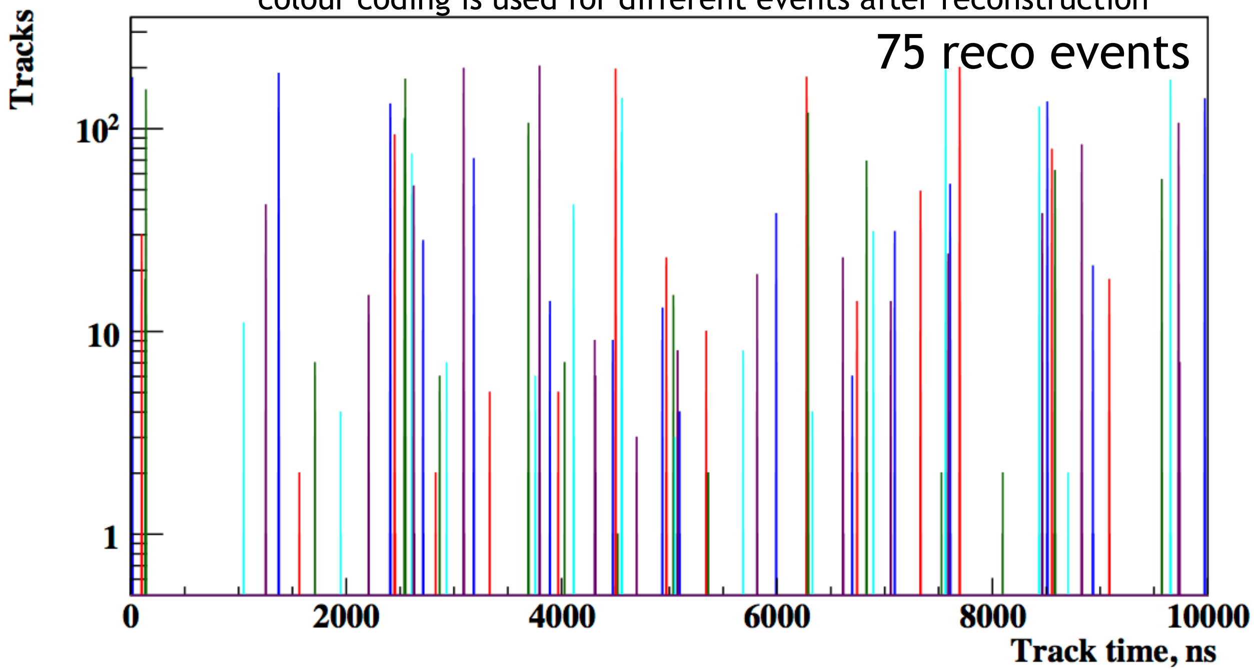
Reconstructed tracks are grouped in events using histogramming:

- all tracks are filled in a time histogram with bin width of 1 ns
- neighbouring not empty bins are called an event
- gap of a 4 empty bins is a sign for event end

	normalised to reconstructable MC events (83)	all MC events (100)
Eff.	100%	83%
Clones	0	0
Ghost	0	0

Event reconstruction efficiency: 100 %

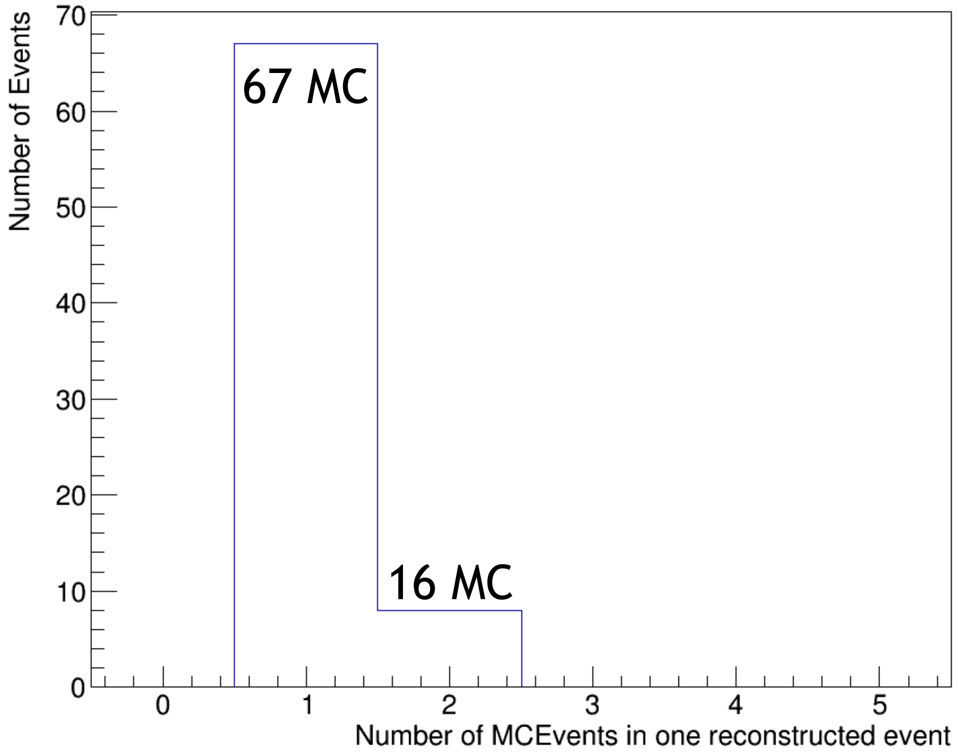
colour coding is used for different events after reconstruction



- The event is called reconstructable if it has at least one reconstructed track
- MC event is called reconstructed if all its reco tracks belong to this reco event only
- Thus reconstructed MC event is not splitted in several reco events, but can be merged with other MC event
- Clones are two or more reco events with tracks from one MC event (event splitting)
- Ghost is an event which tracks are not matched to any MC events

Event reconstruction efficiency 100 % with no event splitting

Primary Vertex Reconstruction



- About 16% of MC events are merged together, can not be separated with time info only
- Primary tracks can be separated using primary vertex information.
- Search of only one primary vertex per event using KF Particle Finder package is currently implemented.
- Multi-vertex reconstruction is in progress.

Merged events are the task for future multi-vertex analysis

Summary

Summary

- Event building is a necessary part of the FLES package
- Time-based 4D tracking allows to efficiently reconstruct tracks without ghost with speed of 84 ms for a time-slice of 100 bias while run in parallel on a 20 logical core CPU
- At interaction rates of 10 MHz it is possible to use time-slices with 100 events or more without data duplication of border region or event splitting
- A first version of event building was implemented based on the 4D tracking. The algorithm shows 100% efficiency without event splitting, although some events are merged, which is a task for future multi-vertex reconstruction

Future Plans

- Multi-vertex reconstruction for merged events
- Include 4D CA Track Finder in CBMROOT