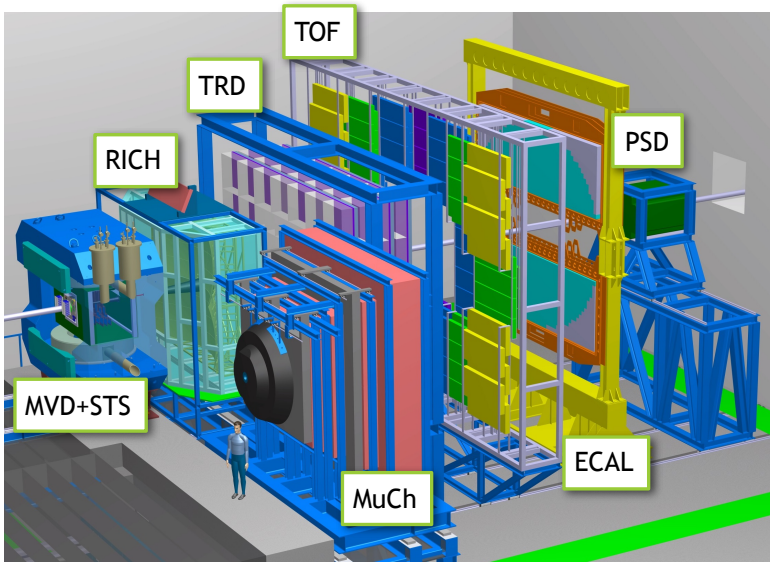
Application of Cellular Automaton track finder in TPC detectors

G.Kozlov[1,2], Y.Fisyak[3], I.Kisel[1,4], M.Zyzak[4]

[1]FIAS, Frankfurt, Germany
[2]JINR LIT, Dubna, Russia
[3]BNL, Upton, USA
[4]GSI, Darmstadt, Germany

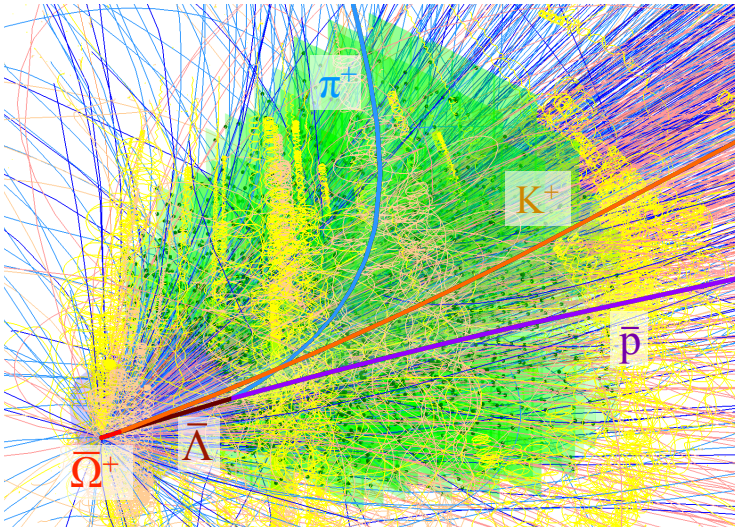
26.02.2018

Challenges in CBM

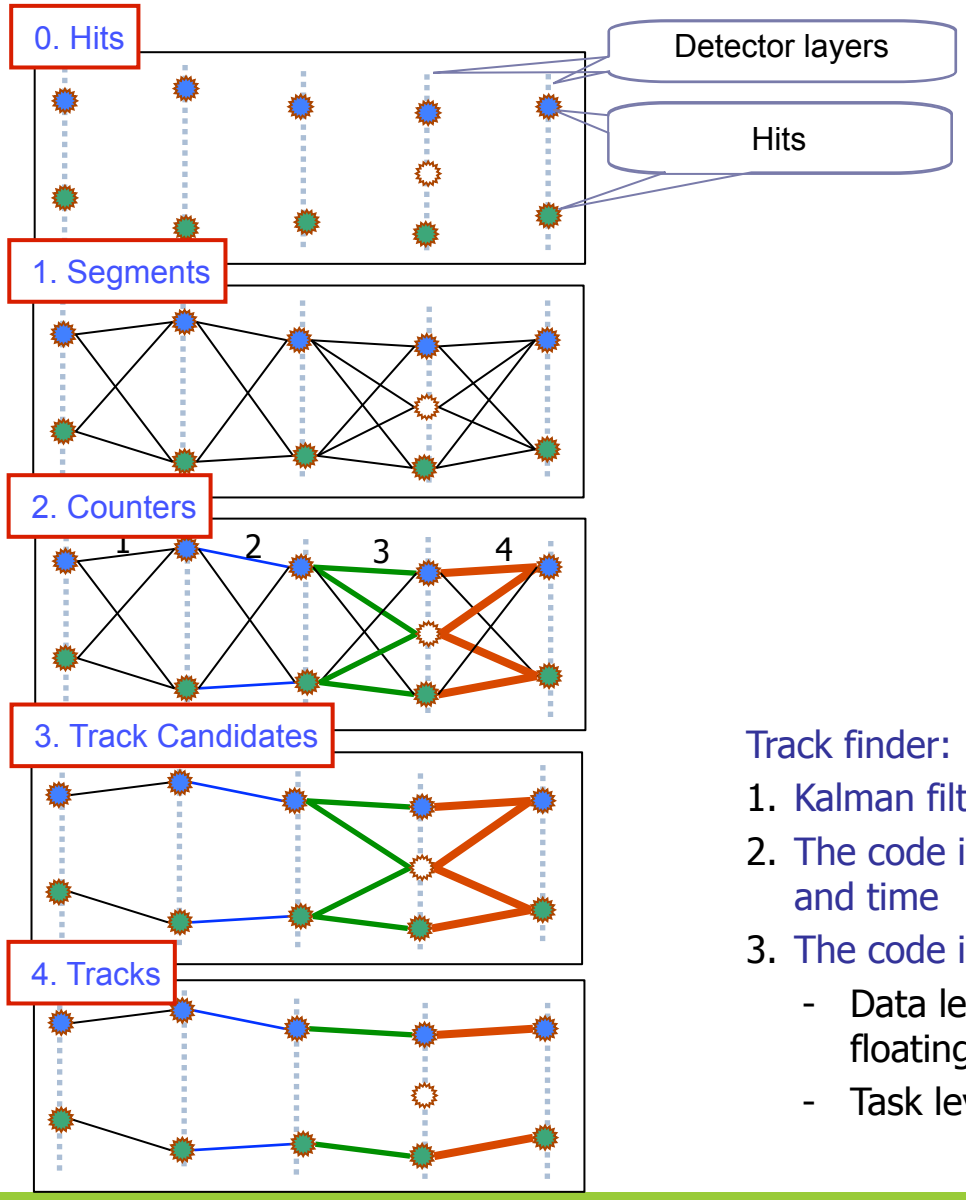


- CBM — future fixed-target heavy-ion experiment at FAIR, Darmstadt, Germany.
- Interaction rate: 10^5-10^7 collisions per second.
- Up to 1000 charged particles/collision.
- Free streaming data.
- No hardware triggers.
- On-line time-based event reconstruction and selection is required in the first trigger level.

- On-line reconstruction at the on-line farm with 60000 CPU equivalent cores.
- High speed and efficiency of the reconstruction algorithms are required.
- The algorithms have to be highly parallelised and scalable.
- CBM event reconstruction: Kalman Filter and Cellular Automaton.



Cellular Automaton 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:

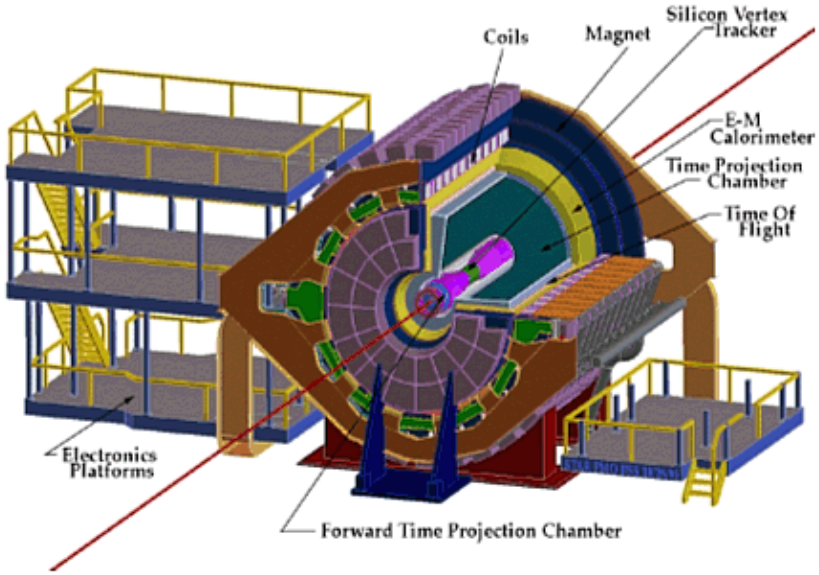
- local w.r.t. data
- intrinsically parallel
- extremely simple
- very fast

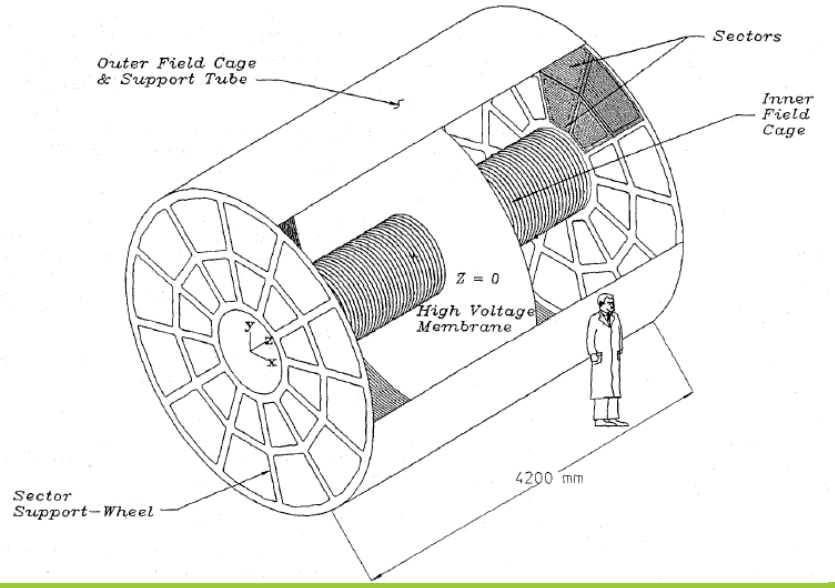
Perfect for many-core CPU/GPU !

Track finder:

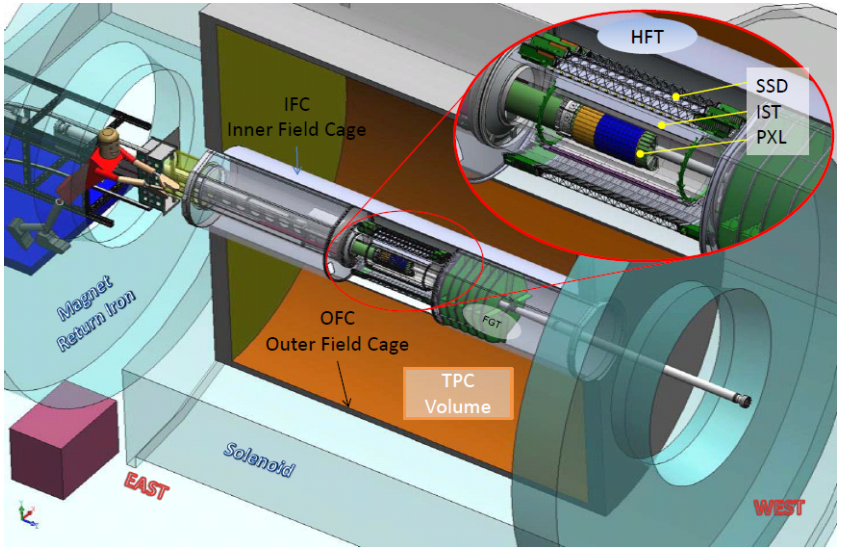
1. Kalman filter for track segments fit
2. The code is optimised with respect to both efficiency and time
3. The code is parallelised
   - Data level (SIMD instructions, 4 single-precision floating point calculations in parallel)
   - Task level (ITBB, parallelisation between cores)

STAR at BNL



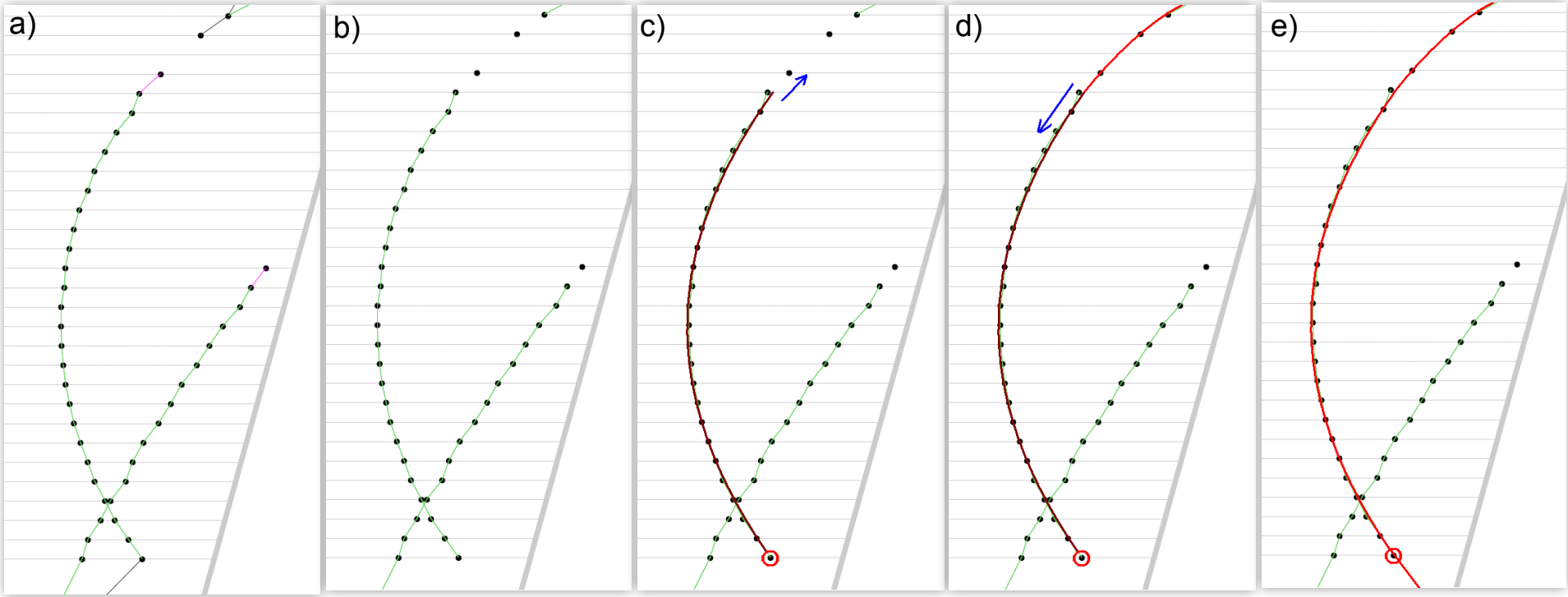


- Collider experiment at RHIC, BNL
- Up to 200 AGeV Au-Au collisions
- Main detector – TPC
- Standard Sti track reconstruction is based on track following
- Increased RIHC luminosity
- Upgrade the reconstruction algorithms for:
  - vectorization
  - multi-threading
  - many-core systems
- Study of the CA tracking algorithm within FAIR Phase 0



CA track reconstruction in STAR TPC

1. Reconstruction of track segments in each TPC sector:
   a) Find and link neighbours hits;
   b) Clean links;
   c) Create segments by fitting chains and adding outer hits;
   d) Refit tracks and add inner hits;
   e) Selection of tracks;
2. Merge sector tracks into TPC global tracks.



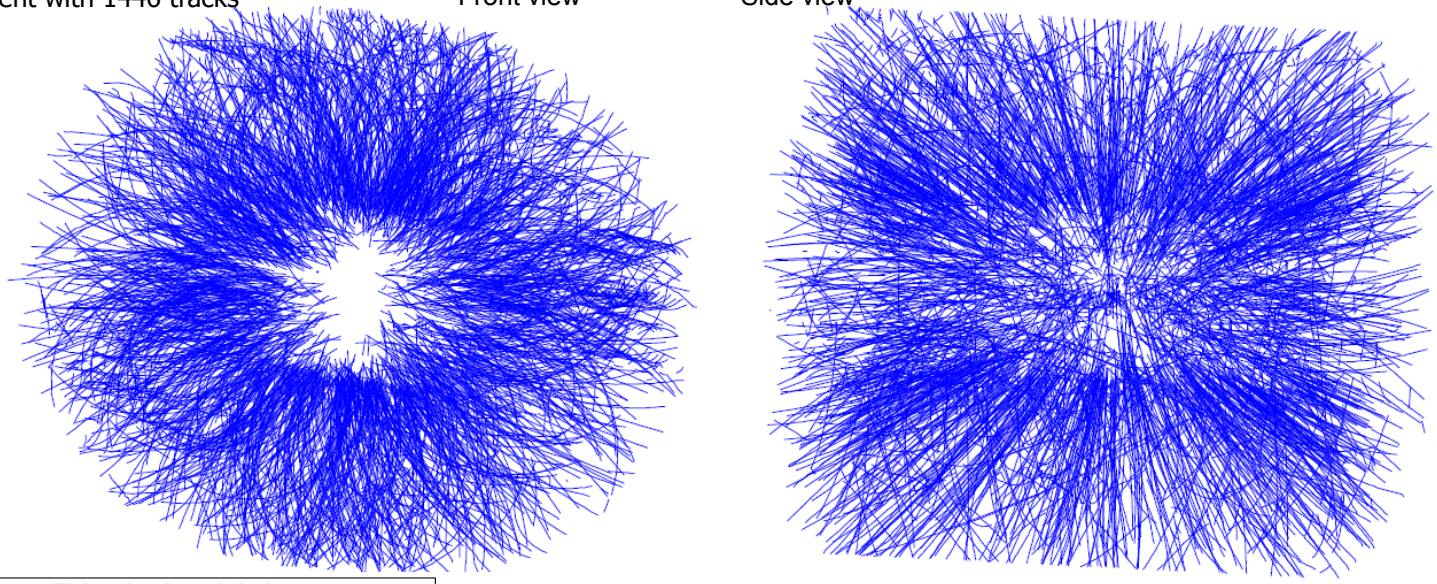
Based on the ALICE HLT TPC CA Track Finder

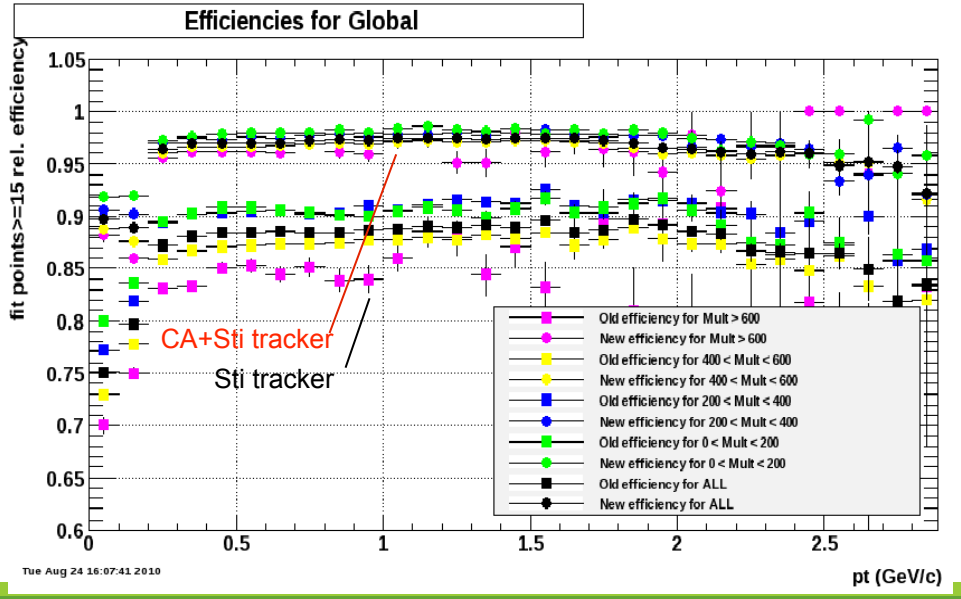
STAR TPC CA Track Finder

Au-Au event with 1446 tracks

Front view

Side view

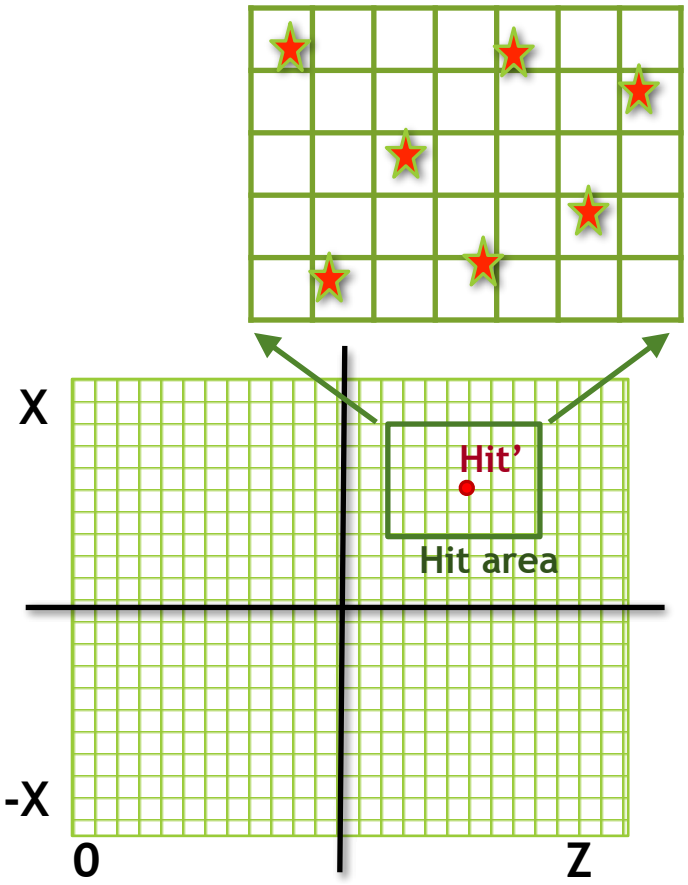




- More stable at high track multiplicity;
- Higher tracking efficiency;
- About 10 time faster than track following based Sti track finder.

Grid structure in TPC

Grid structure allows to establish compliance between hit coordinates and bins of this structure.

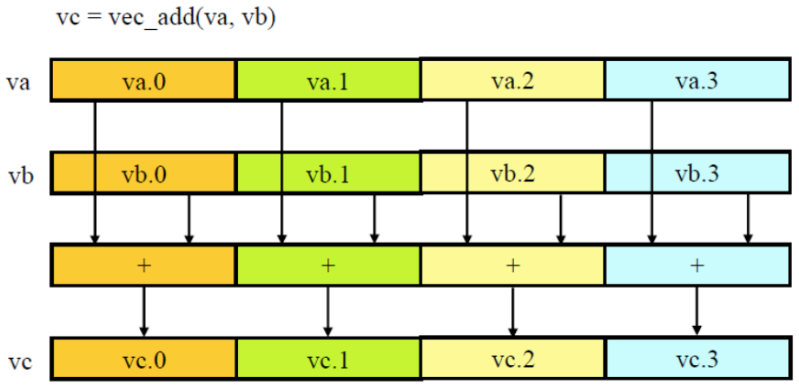


- Algorithm:
  - Calculate approximate hit' coordinates on the row;
  - Create *Hit area* around the hit';
  - Take hits only from the *Hit area*.

- Separate grid for each row in every slice;
- Based on X and Z coordinates;
- Able to work without information about primary vertex.

Vectorisation

- SIMD intrinsics with Vc headers are used.
- Faster calculations with the same hardware.
- Optimal for streaming calculations.



- Vectorisation of the grid structure: reduces the number of calls to the grid structure and creates data streams.
- Vectorisation of the track segment fitting and extrapolation.
- Vectorisation of the final fit in merger.

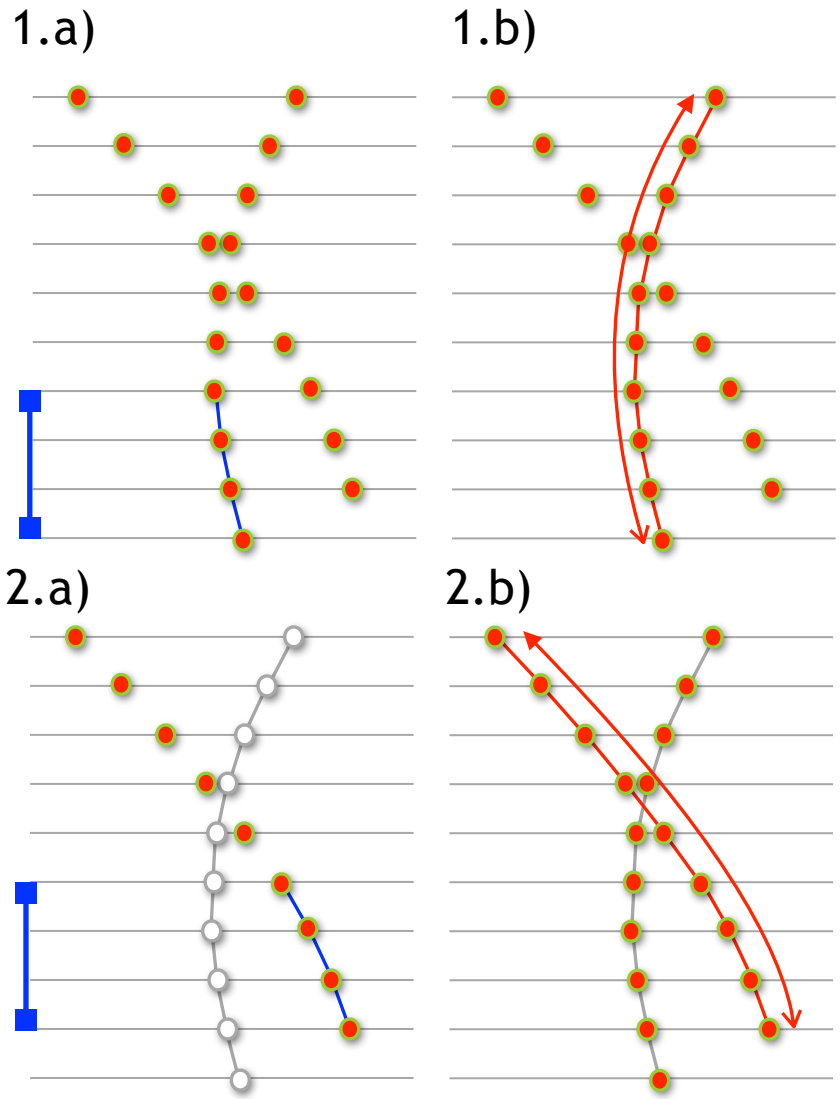
Scalar grid

Hit area	Hit1	Hit2	Hit3	…

Vector grid

Hit area 1	Hit1	Hit2	Hit3	Hit4
Hit area 2	Hit1	Hit2		
Hit area 3	Hit1	Hit2	Hit3	
Hit area 4	Hit1	Hit2	Hit3	…

Algorithm optimisation



1. a) Create 4-hit segments starting from the first row.
1. b) Fit and extend seeds to the outer row and back to inner row to get additional hits.
2. a) Create segments starting from the next row. Used hits are skipped.
2. b) Fit and extend new seeds.
3. Move to the next row…

- 4-hit seed is enough for track seeding.
- Less hits for seed finding.
- Hit area is much smaller.
- Allows to reduce the detector inefficiency.

Results

Calculation time (ms)

	Scalar	Vectorized	Optimized
Sector tracking	24.7	18.4	15.0
- Chains	15.7	12.2	11.5
- Tracklets	7.0	3.8	
Merging	18.4	12.0	11.2
Total	**43.1**	**30.4**	**26.2**

Intel Xeon X5550 at 2.7GHz

Efficiency (%)

	Vectorized	Optimized
Ref Set	97.5	97.6
AllSet	93.3	94.2
Clone	10.6	12.4
Ghost	13.7	18.6
Reco tracks/ev	650	656
Hits per track	22.4	23.0

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

Summary

- Cellular Automaton based track finder in STAR TPC detector is fast and efficient even in case of high track multiplicity.
- Vectorisation of basic steps of the algorithm allows to increase the calculation speed.
- Optimisation of the code gives additional speed up without losing of the efficiency.

Plans

- Optimise data structures and algorithm to reduce the overhead and vectorisation inefficiency.
- Create 3D grid.