Fast cluster finder for TPC detector

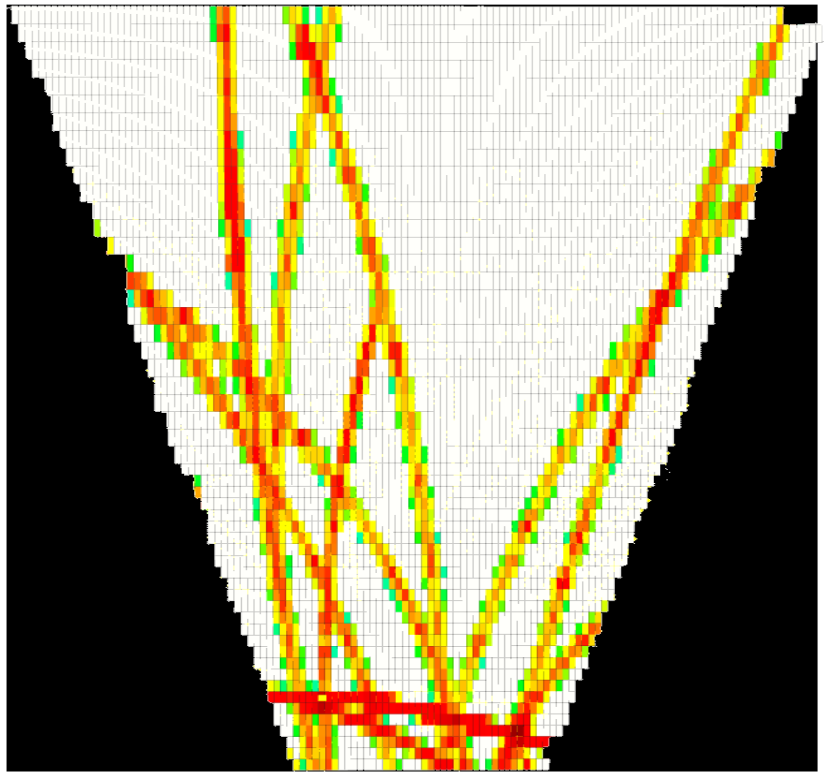
ALEXANDER KRYLOV (LHEP)

EMAIL: AVKRYLOV@JINR.RU

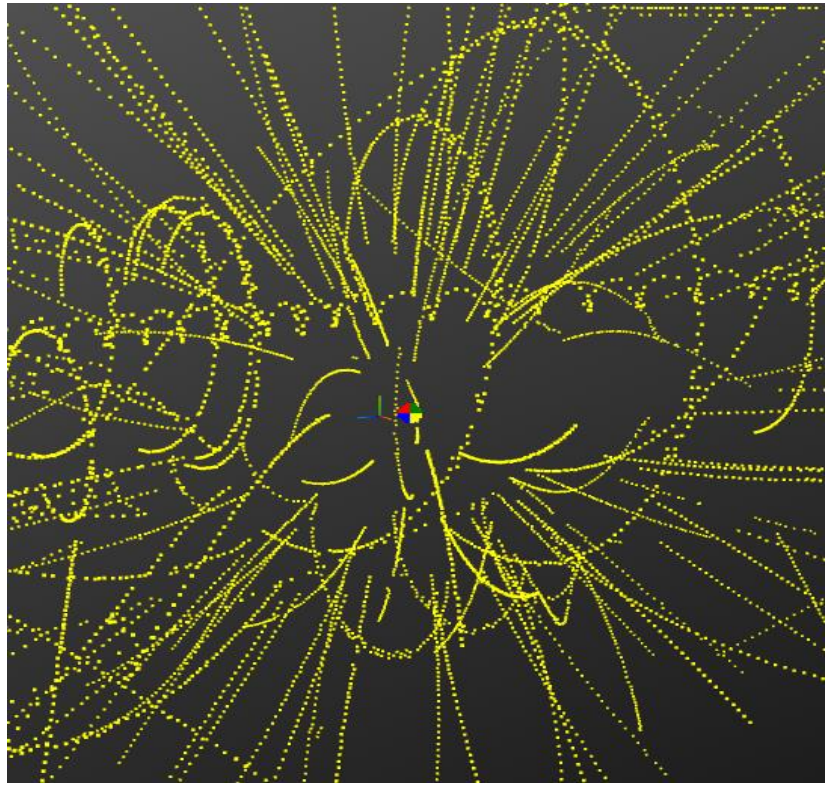
OLEG ROGACHEVSKY (LHEP)

VIKTOR KRYLOV (LNP)

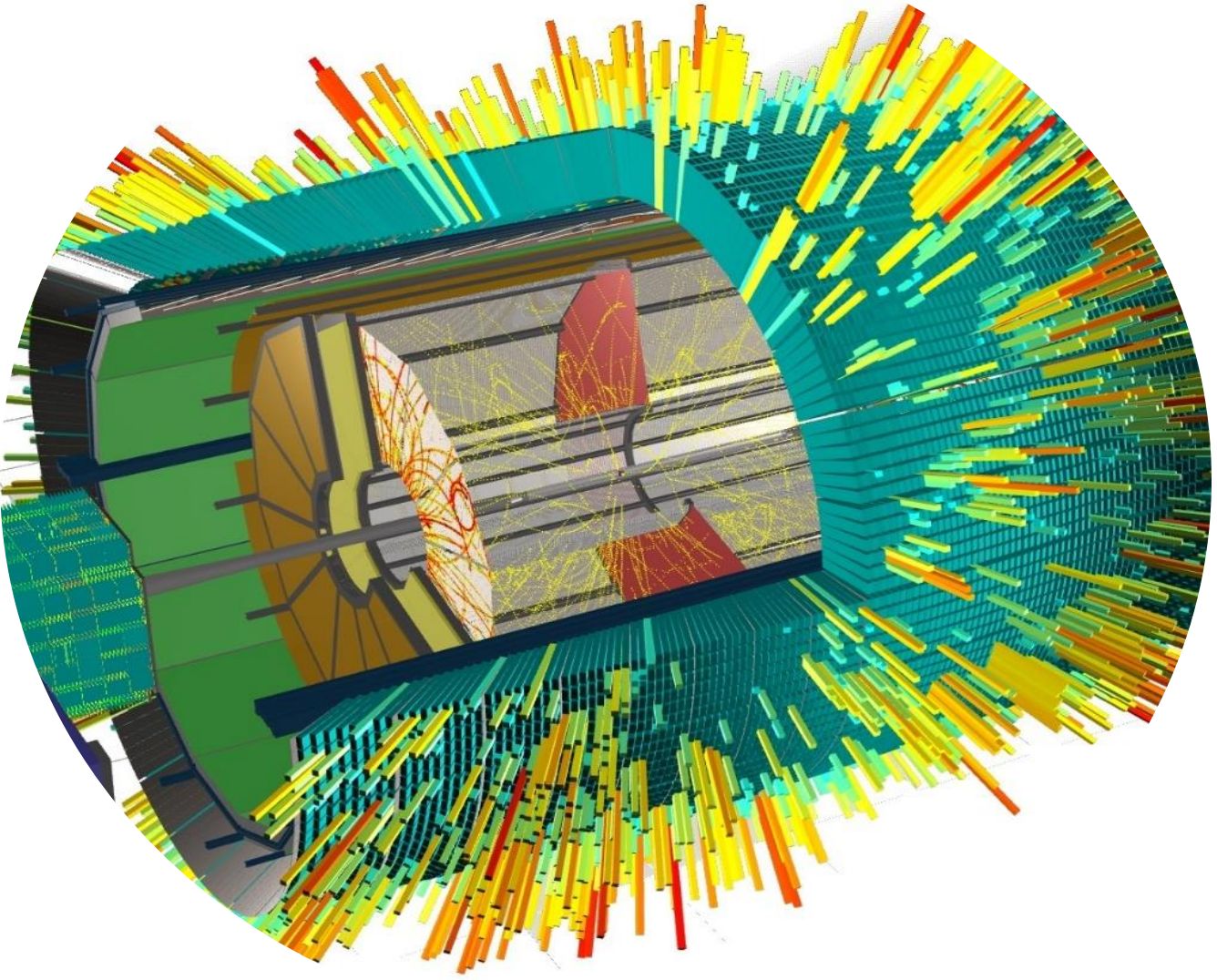
ClusterFinder Task



TPC Digits



TPC Hits



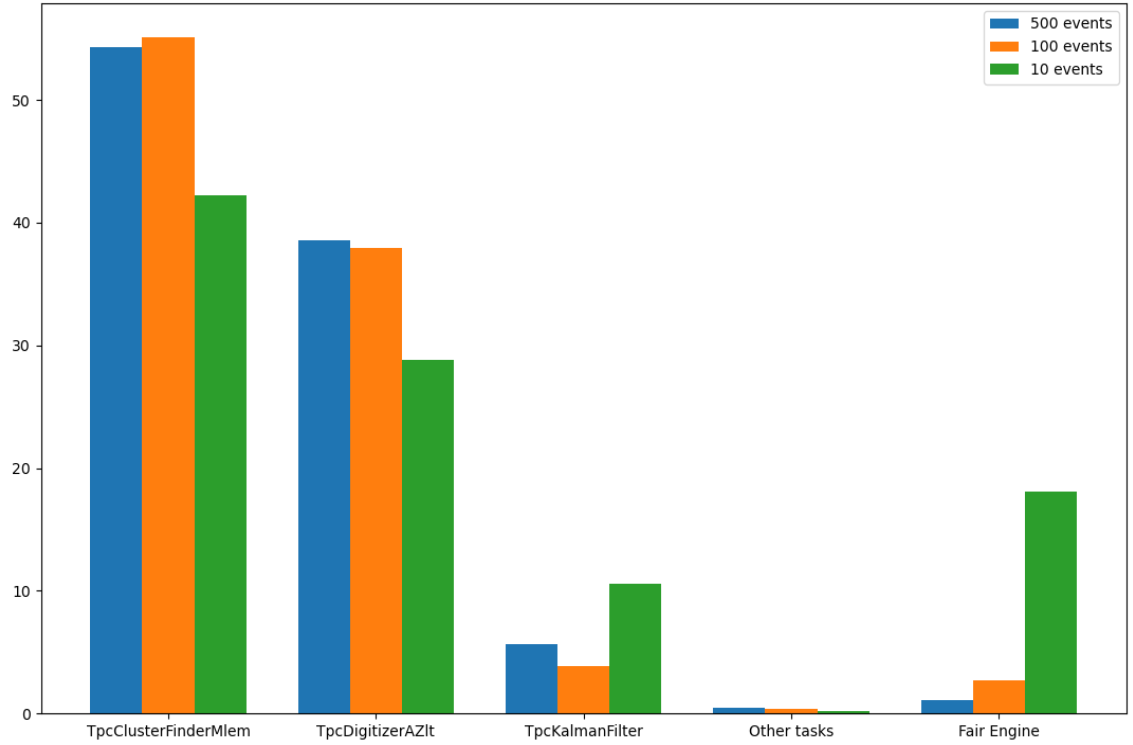
Event Display for the MPD

Event Display help to:

- visualize events online in the control room during run;
- checking reconstruction and physics analysis algorithms;
- look more closely in special events.

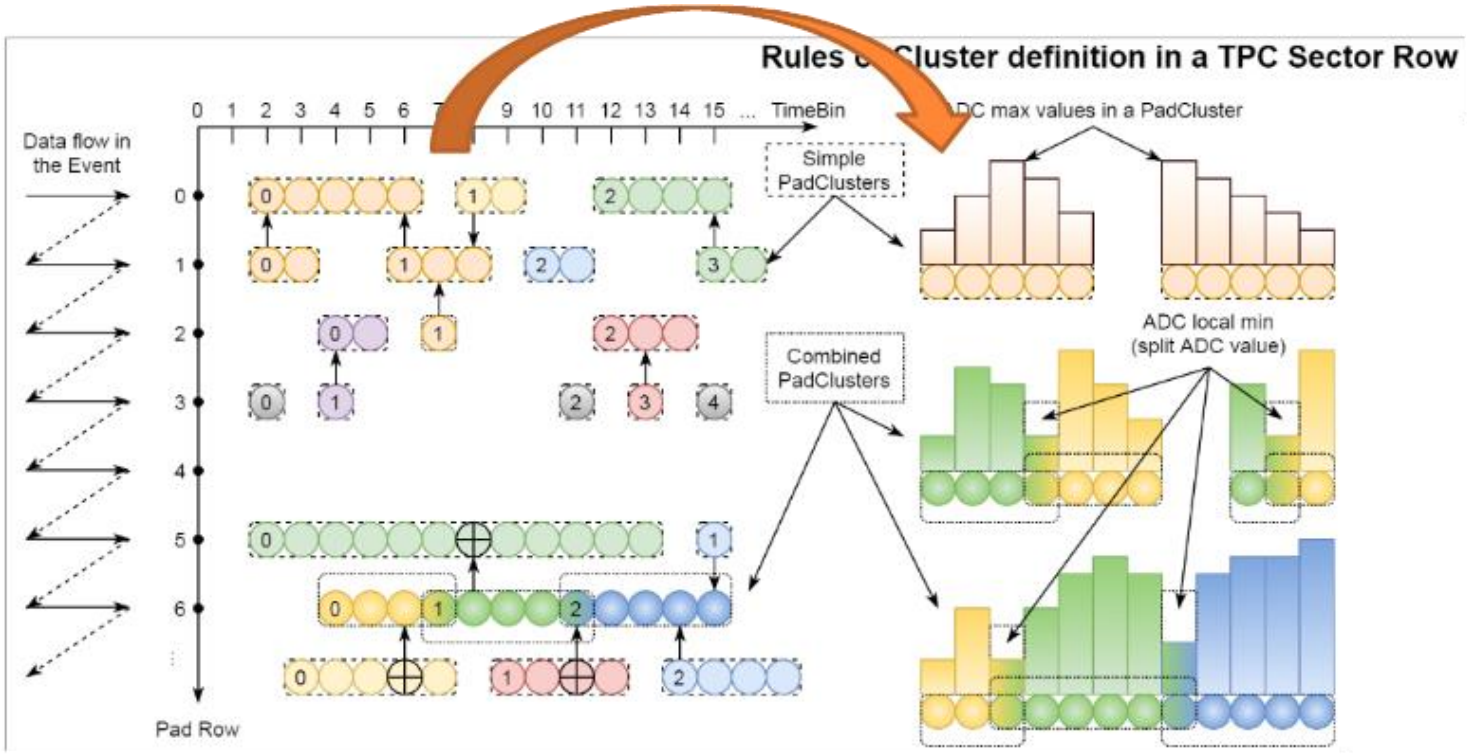
Reconstruction tasks' timing

- A reconstruction of high multiplicity events reaches timings above **1 minute**;
- ClusterFinder task takes half of that time on itself;
- We need fast ClusterFinder task for online visualization purposes.



Percent of reconstuction timing for different tasks

New Fast ClusterFinder algorithm



Fast ClusterFinder algorithm workflow

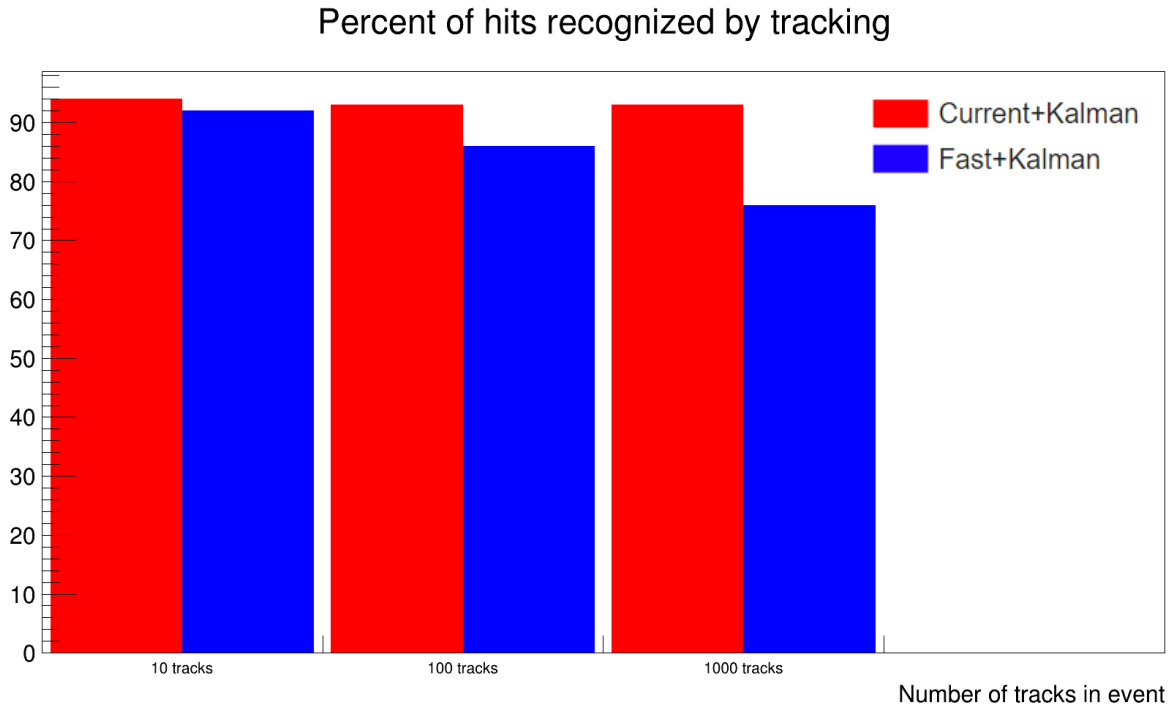
Tracking new hits with present Kalman filter

Quality of ClusterFinder with present Kalman tracking can be estimated by number of hits recognized as a track.

For 100 tracks in event tracking recognize:

- **93.55%** hits for current ClusterFinder
- **86.72%** hits for fast ClusterFinder

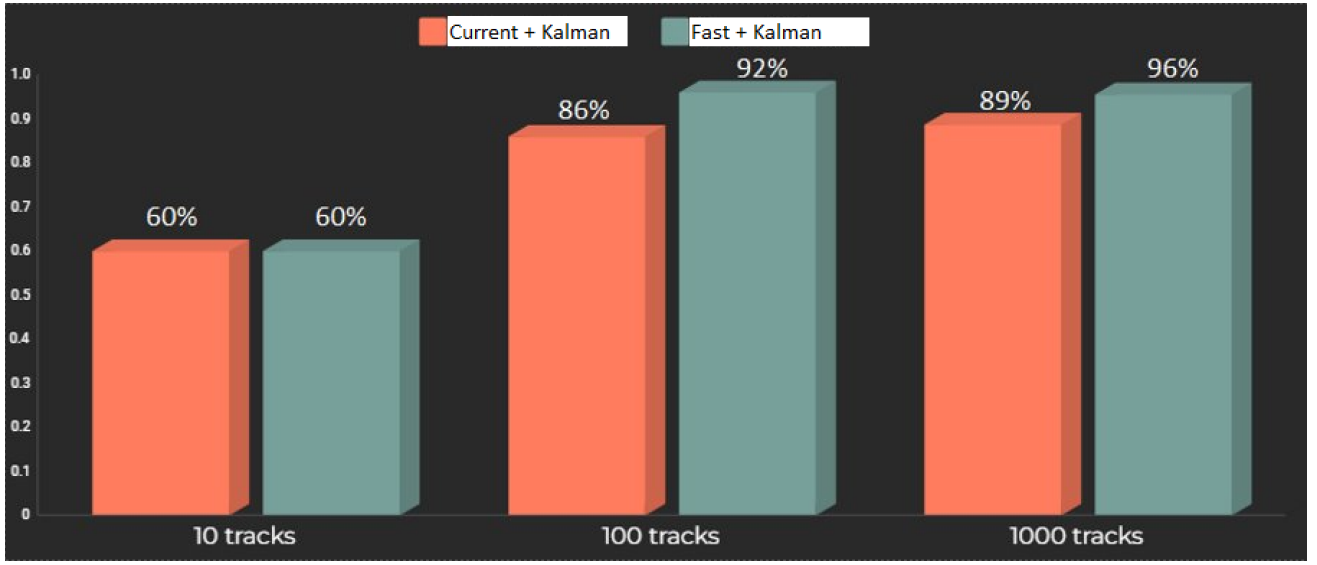
For all tests was used BOX generator with muons and Pt = 2.5 GeV/s



Number of reconstructed tracks

Also we can measure number of reconstructed tracks after ClusterFinder and KalmanFilter tasks.

Lack of work with "edge effect" sometimes gave us 2 different tracks for 1 primary track.



Number of recognized **primary** tracks for different multiplicity

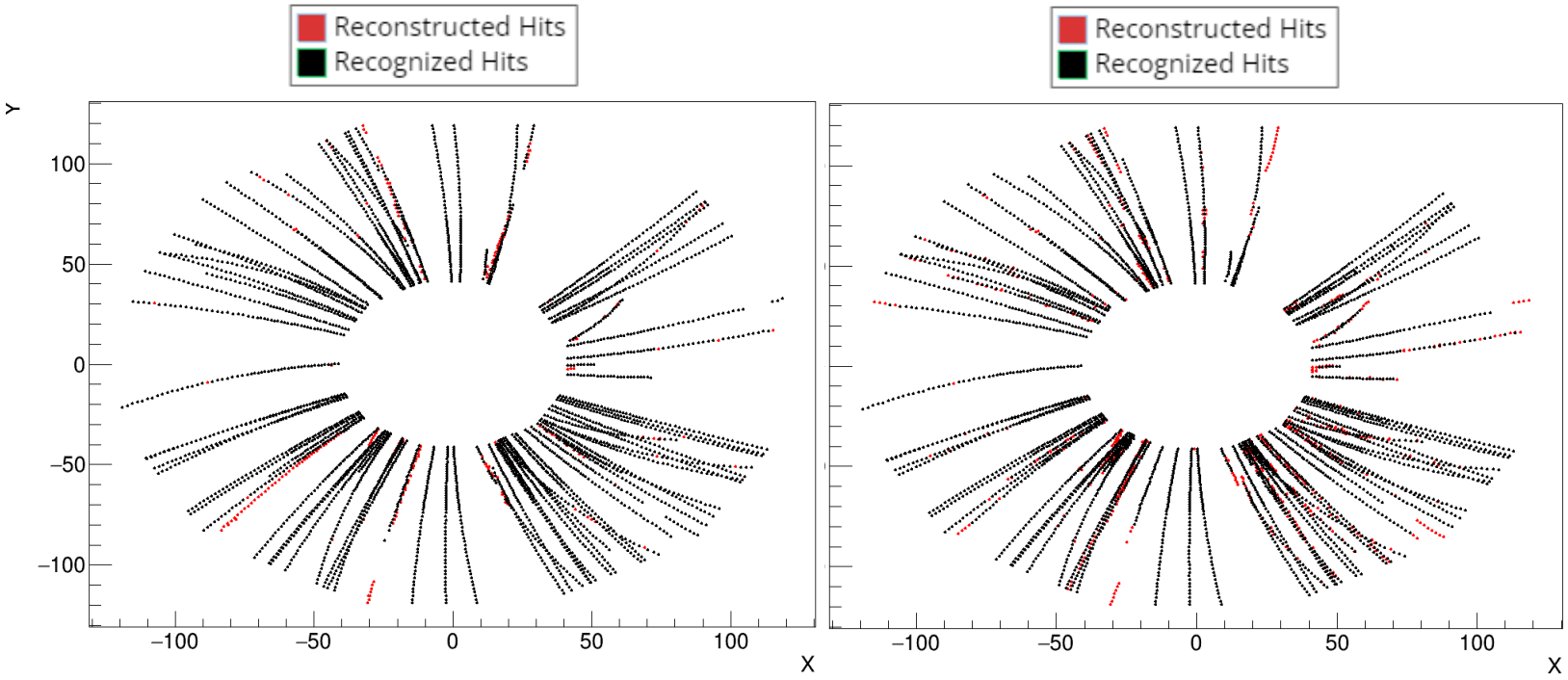
Hits recognized into a track

BOX generator was used:

- Multiplicity – 100 muon tracks
- Pt ≃ 2.5 GeV/c

Current Tasks:
ClusterFinderMlem + KalmanFilter

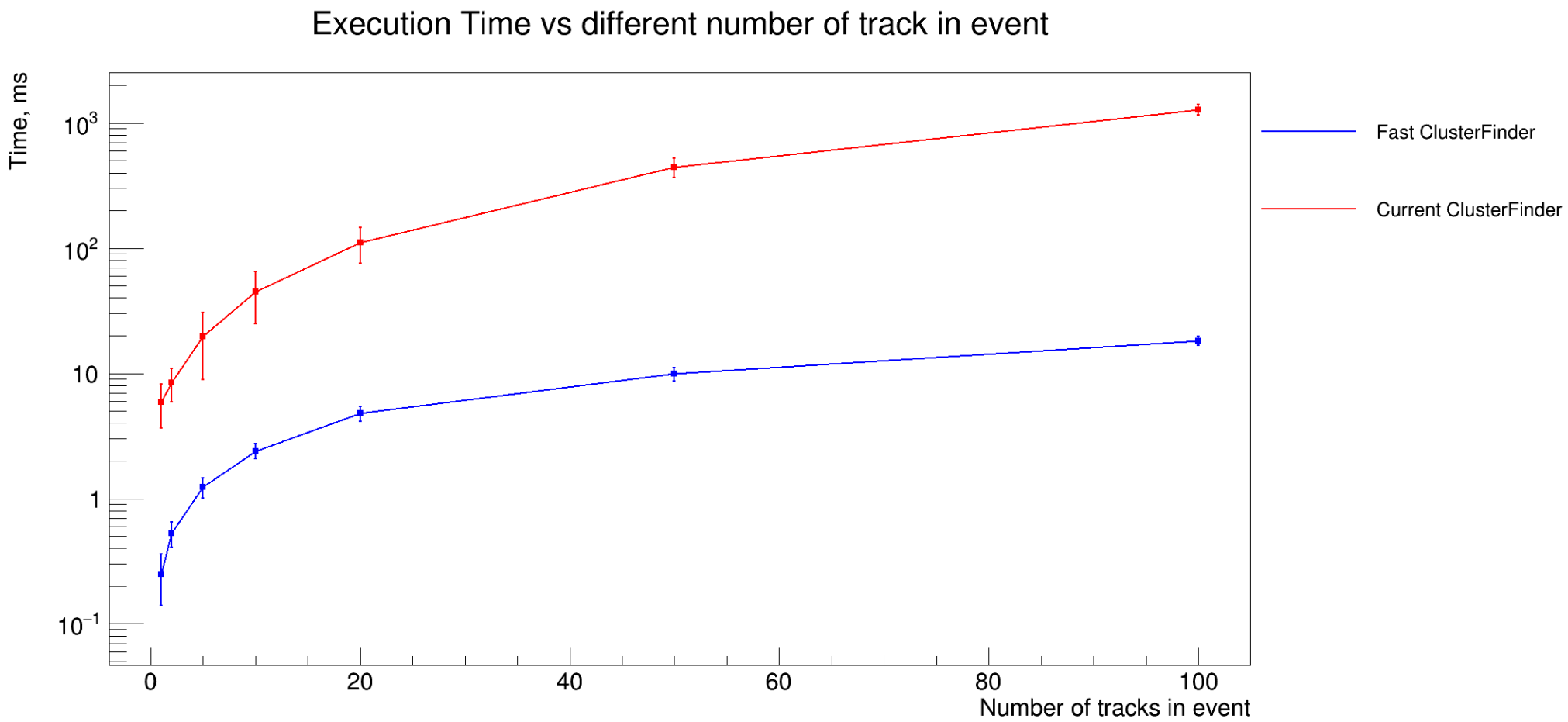
New Tasks:
ClusterFinder + KalmanFilter



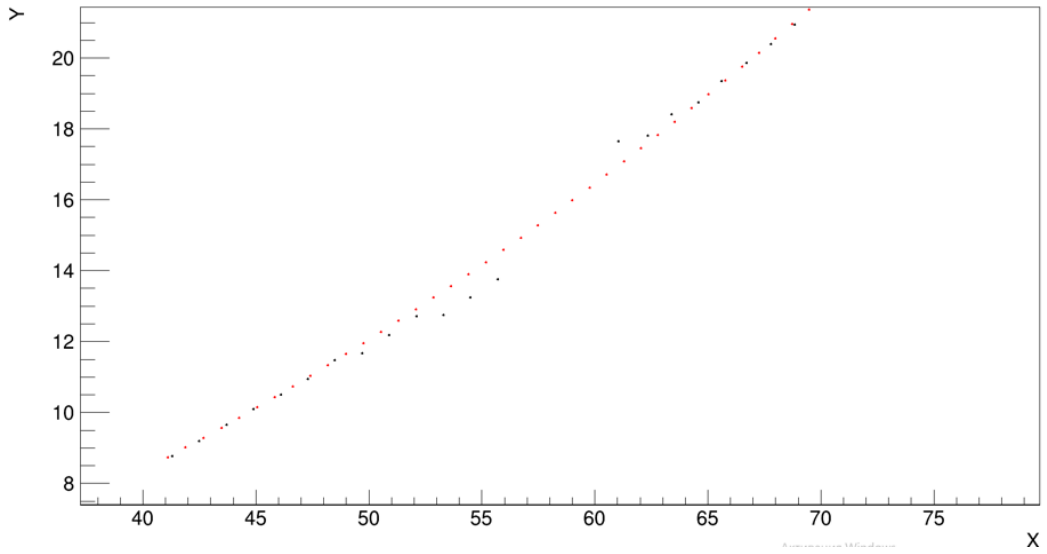
Current Cluster Finder

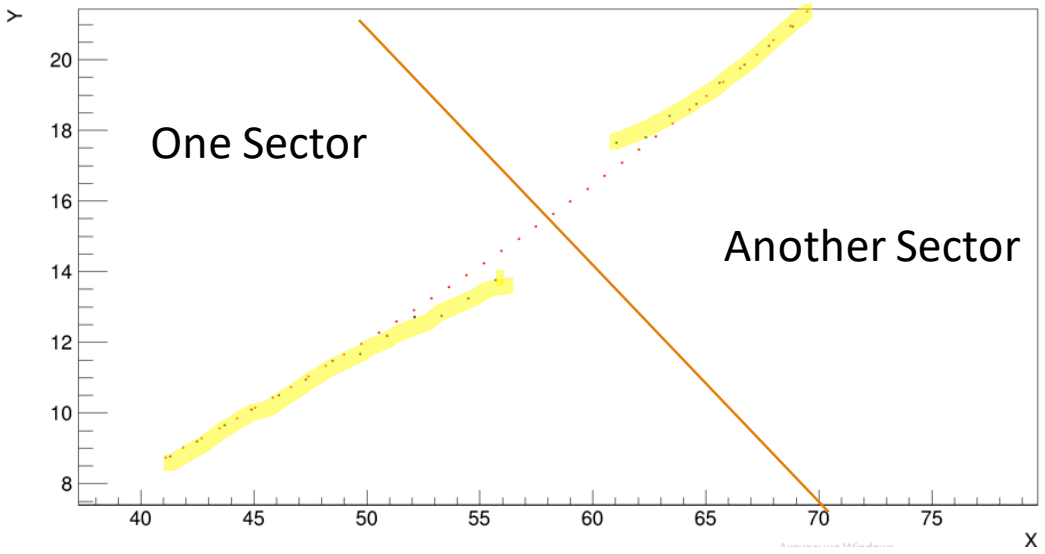
Fast Cluster Finder

Online ClusterFinder Execution time for one Sector



Unsolved problems





"Edge effect" problem

To Do

1. Take into account the edge effect
2. Check performance on real generators (UrQMD, LAQGSM etc.)
3. Get rid of cos() and sin() methods to speed up clustering
4. Add parallelization
5. Implement ClusterFinder into the Event Display