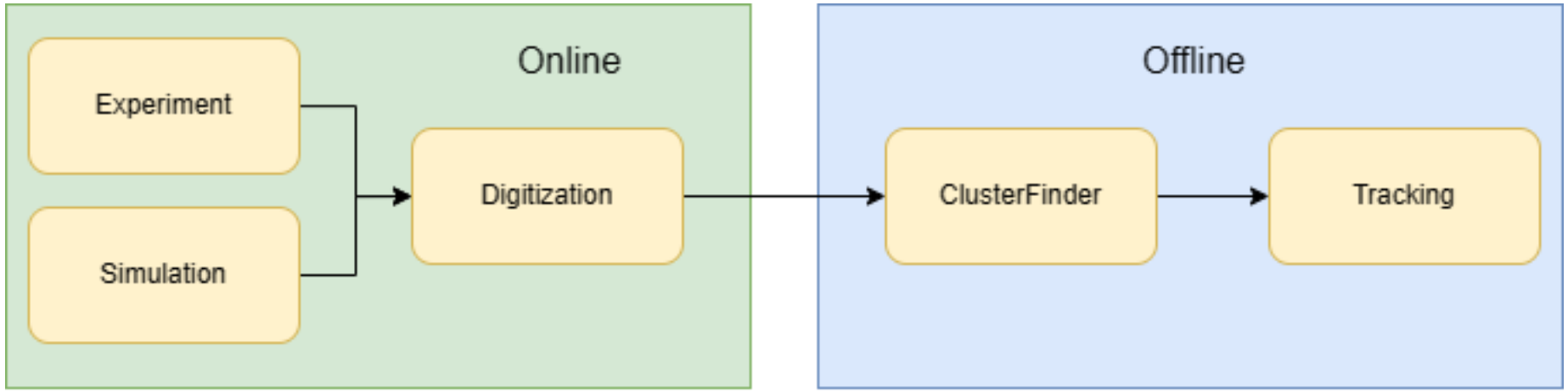
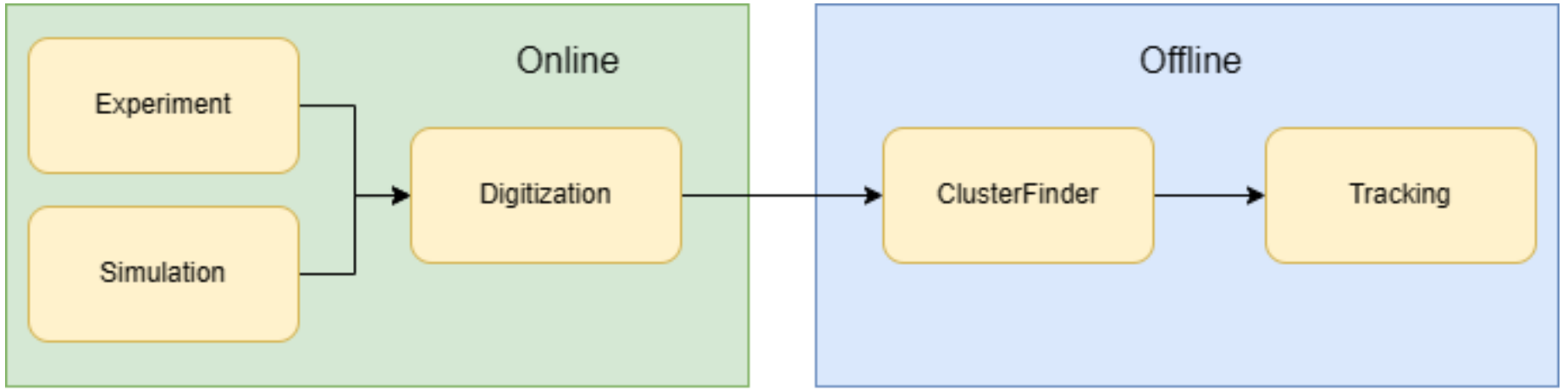
Cluster finder acceleration for TPC detector

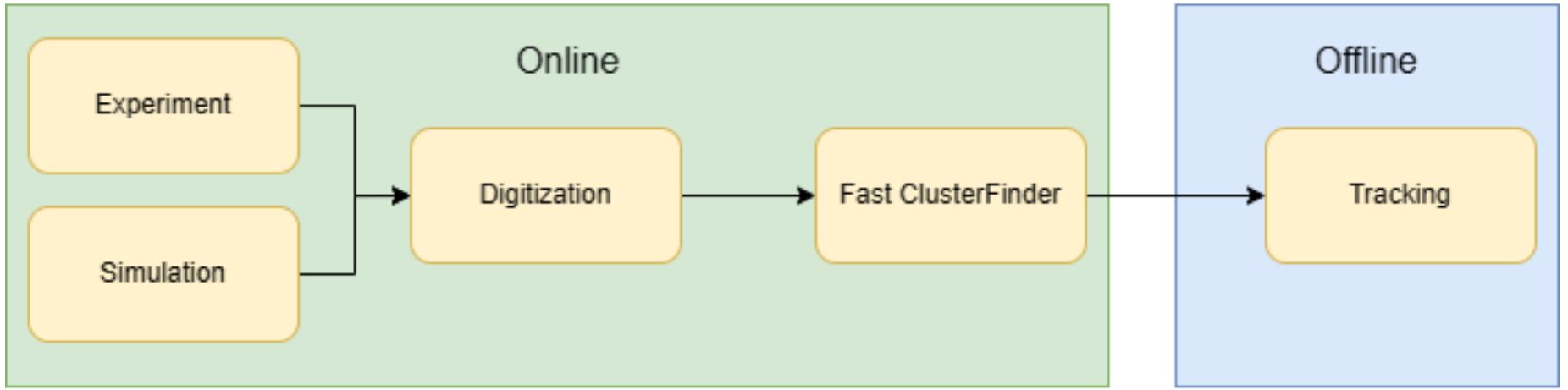
Alexander Krylov (LHEP)
Email: avkrylov@jinr.ru
O. Rogachevsky (LHEP)
V. Krylov (LNP)

TPC reconstruction workflow

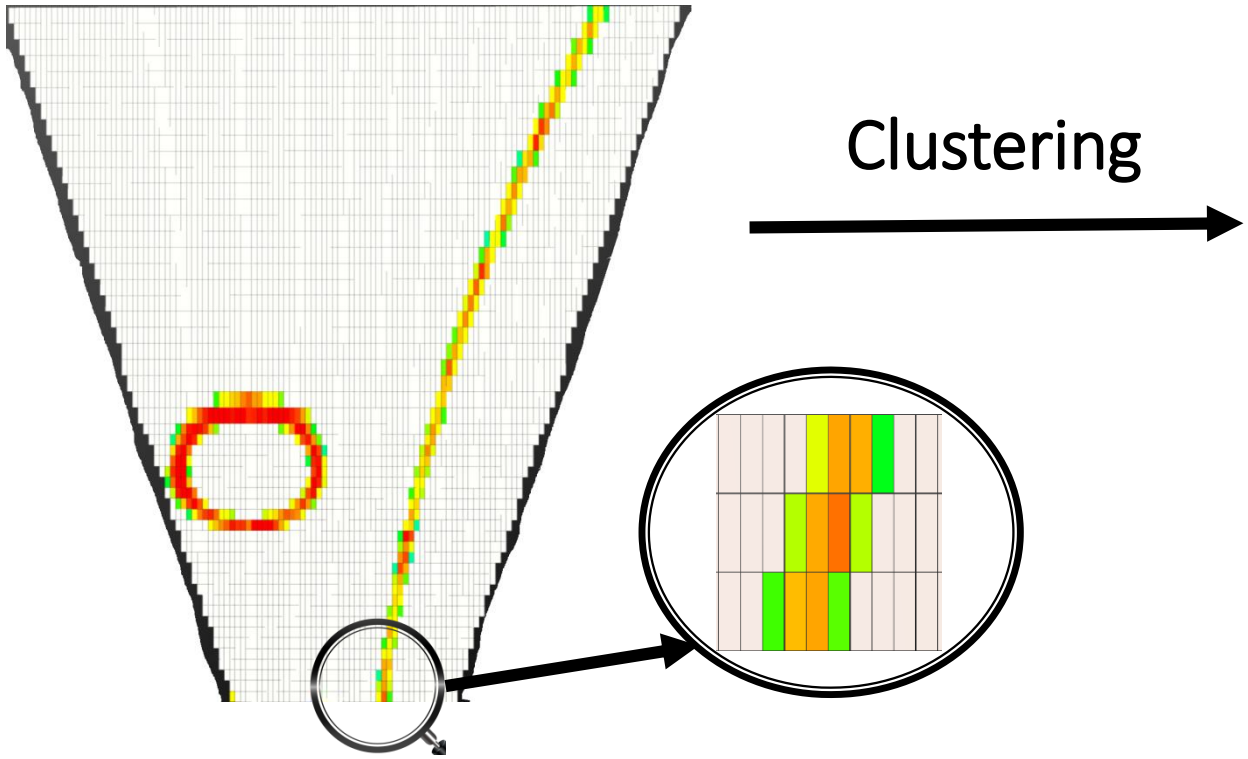


TPC reconstruction workflow



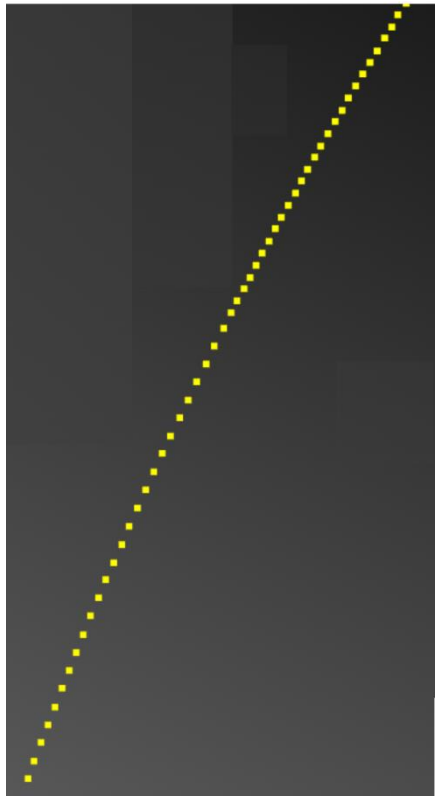


Clustering in TPC

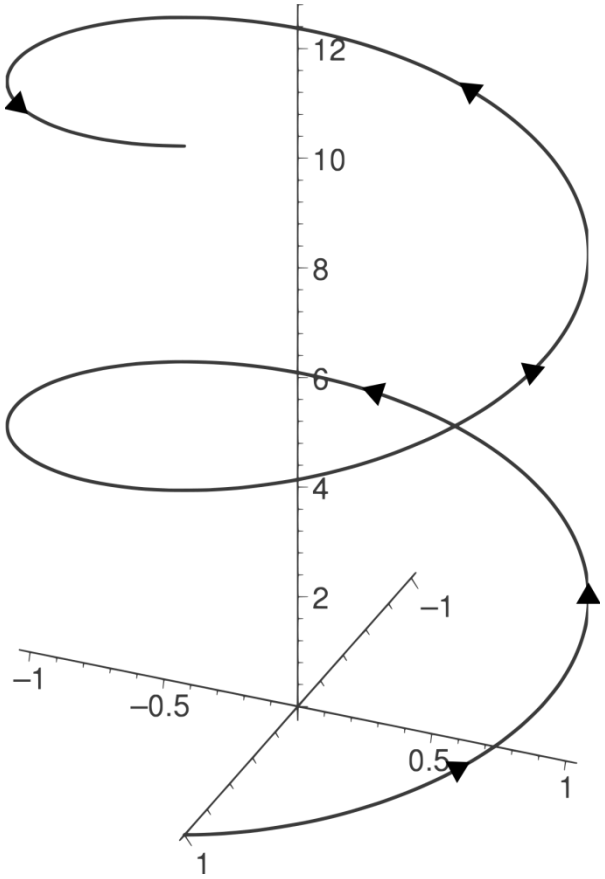


Clustering

Digits on PadPlane

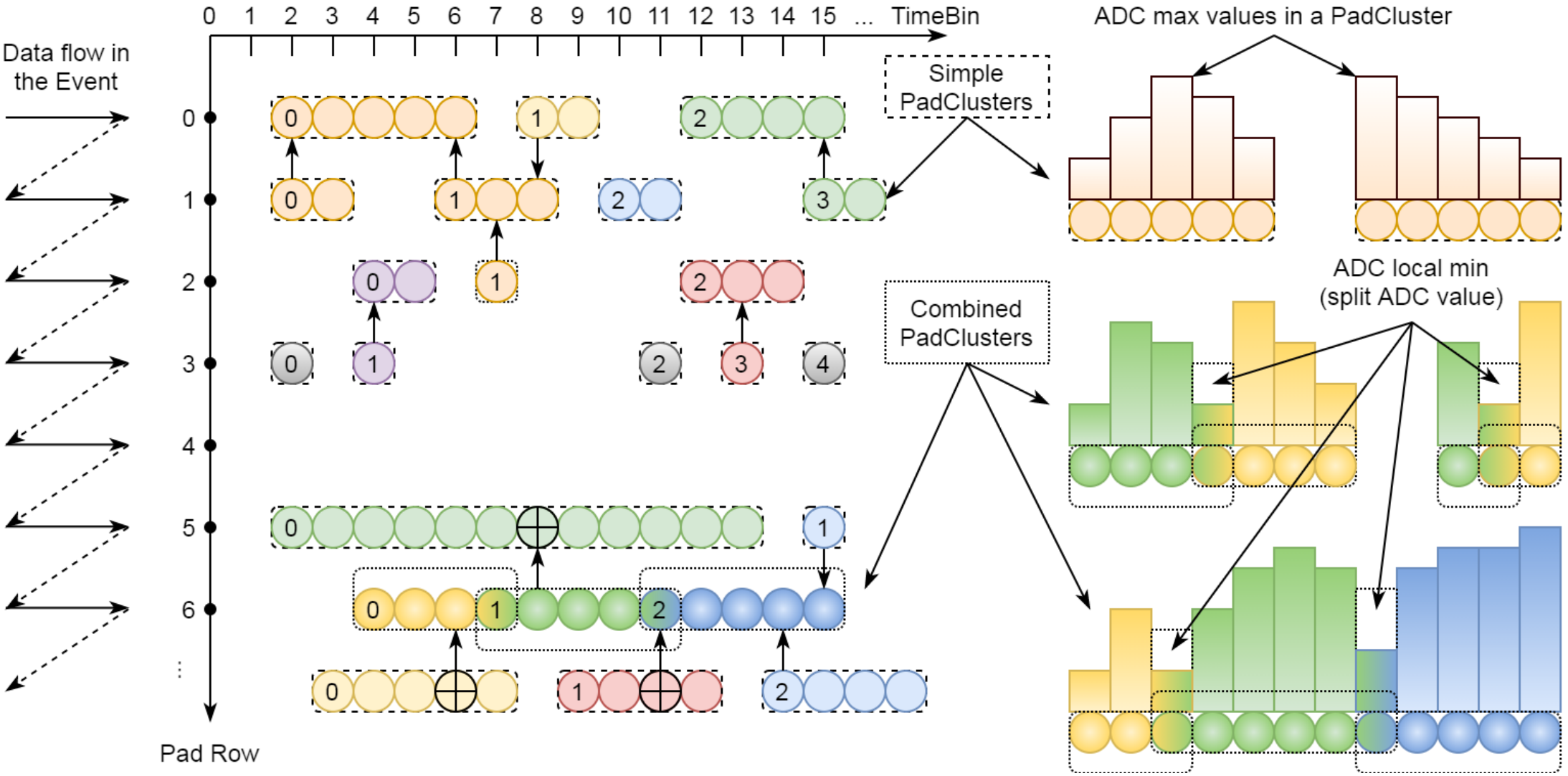


TPC hits



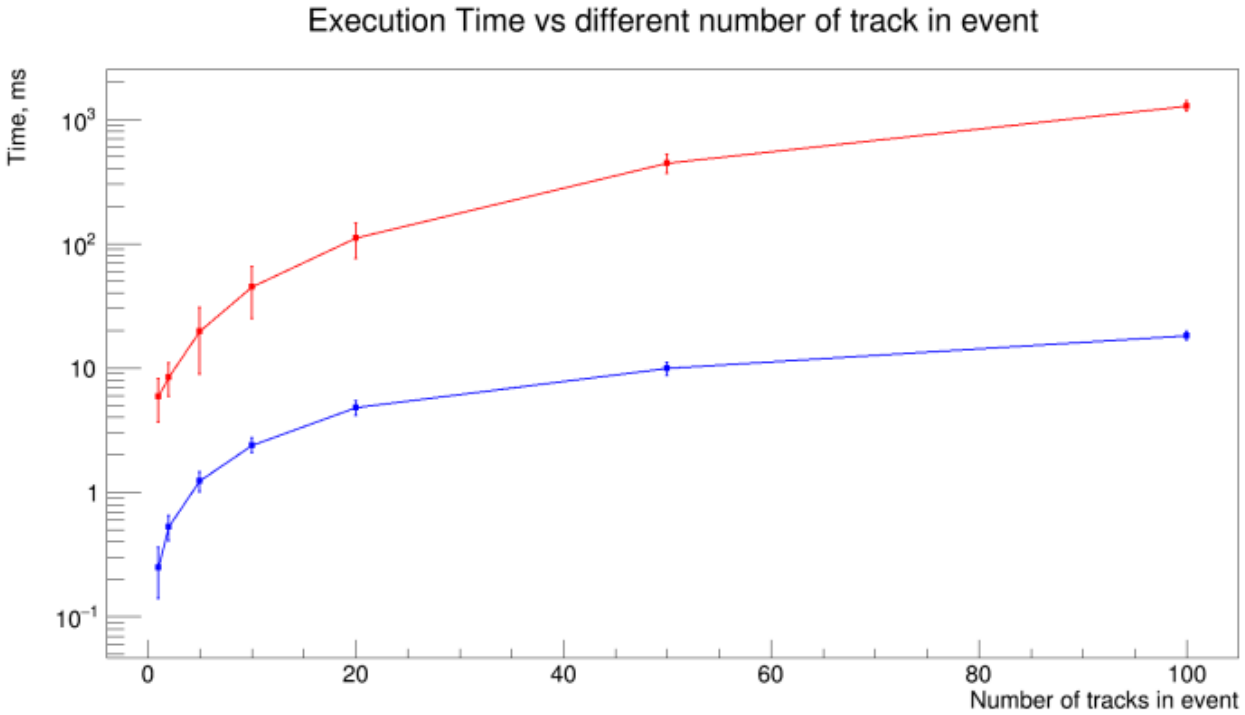
Helix

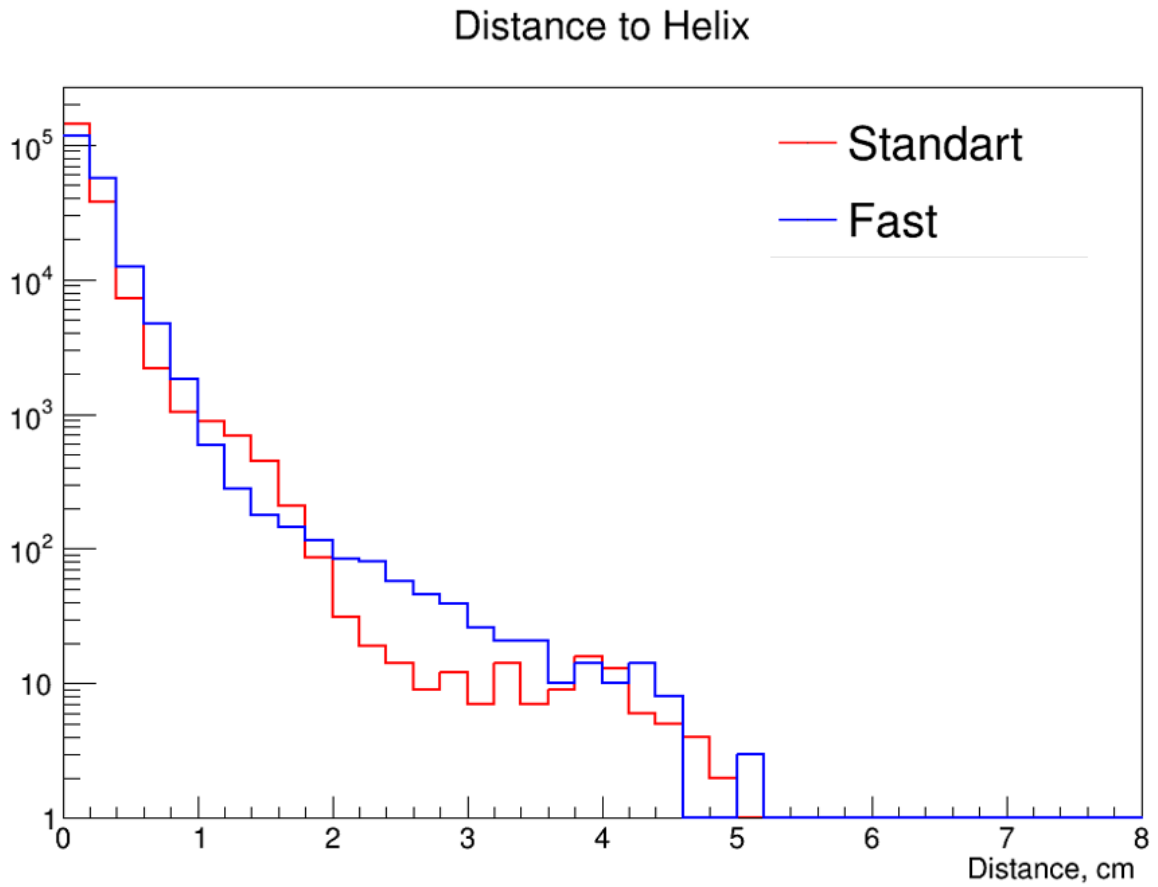
Fast Clustering workflow



Fast vs Standard

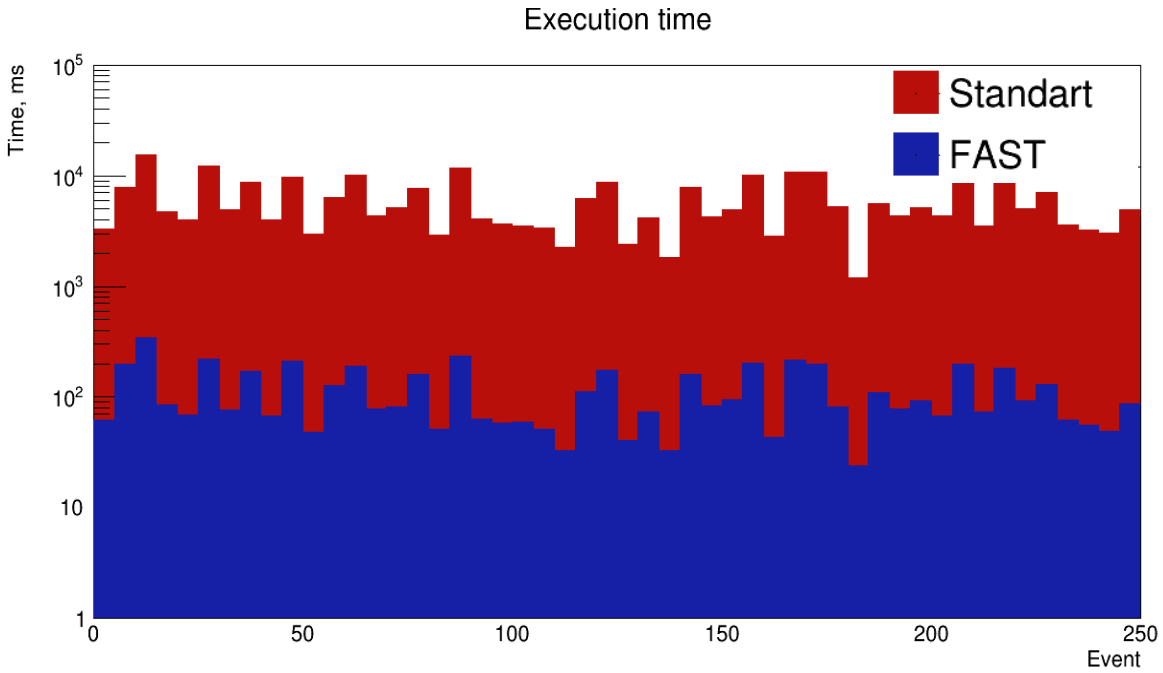
BOX generator (0.25 - 2.5 GeV/c), 100 muons with magnetic field

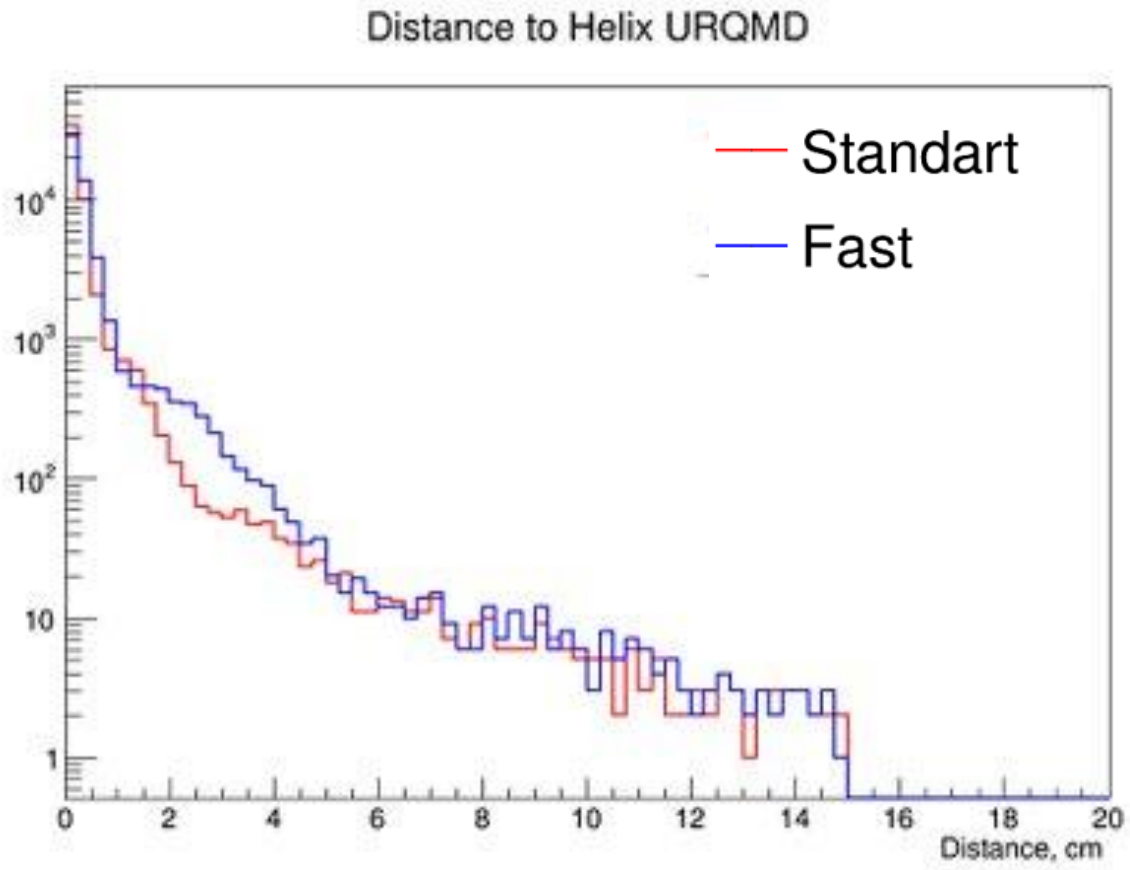




Fast vs Standard

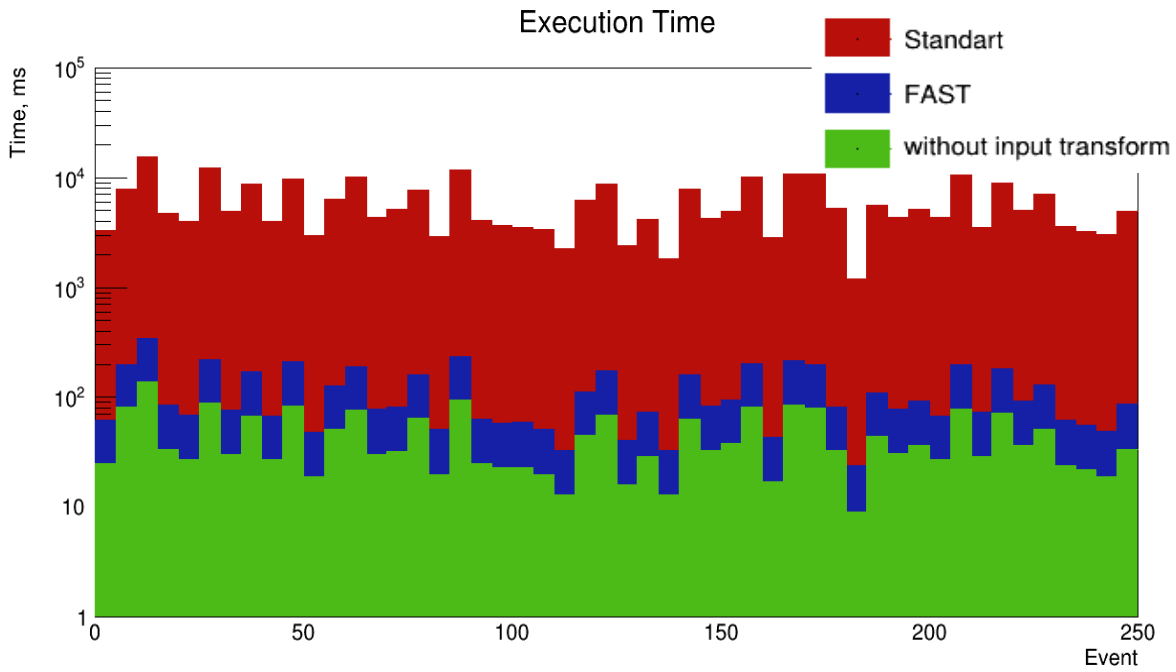
URQMD generator, $\sqrt{s}$= 7.0 GeV, mb, 250 events

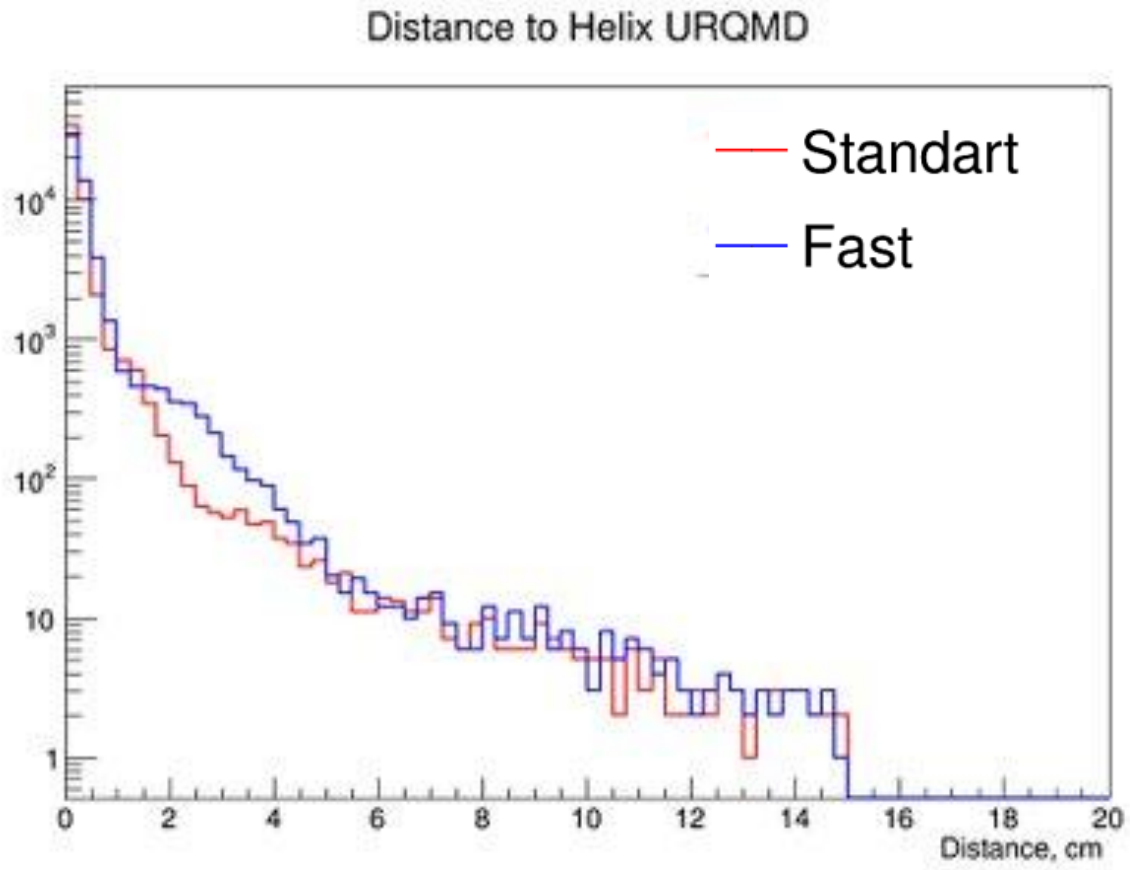




Fast vs Standard

URQMD generator, $\sqrt{s}$= 7.0 GeV, mb, 250 events

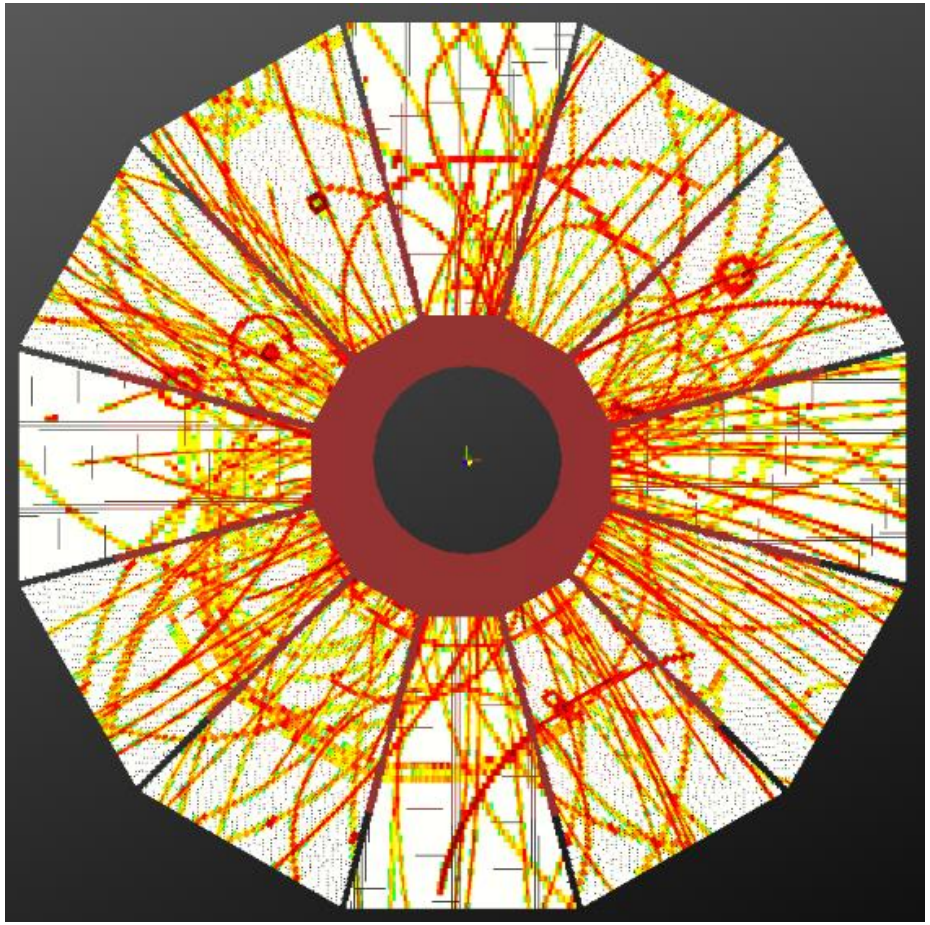




Parallel processing

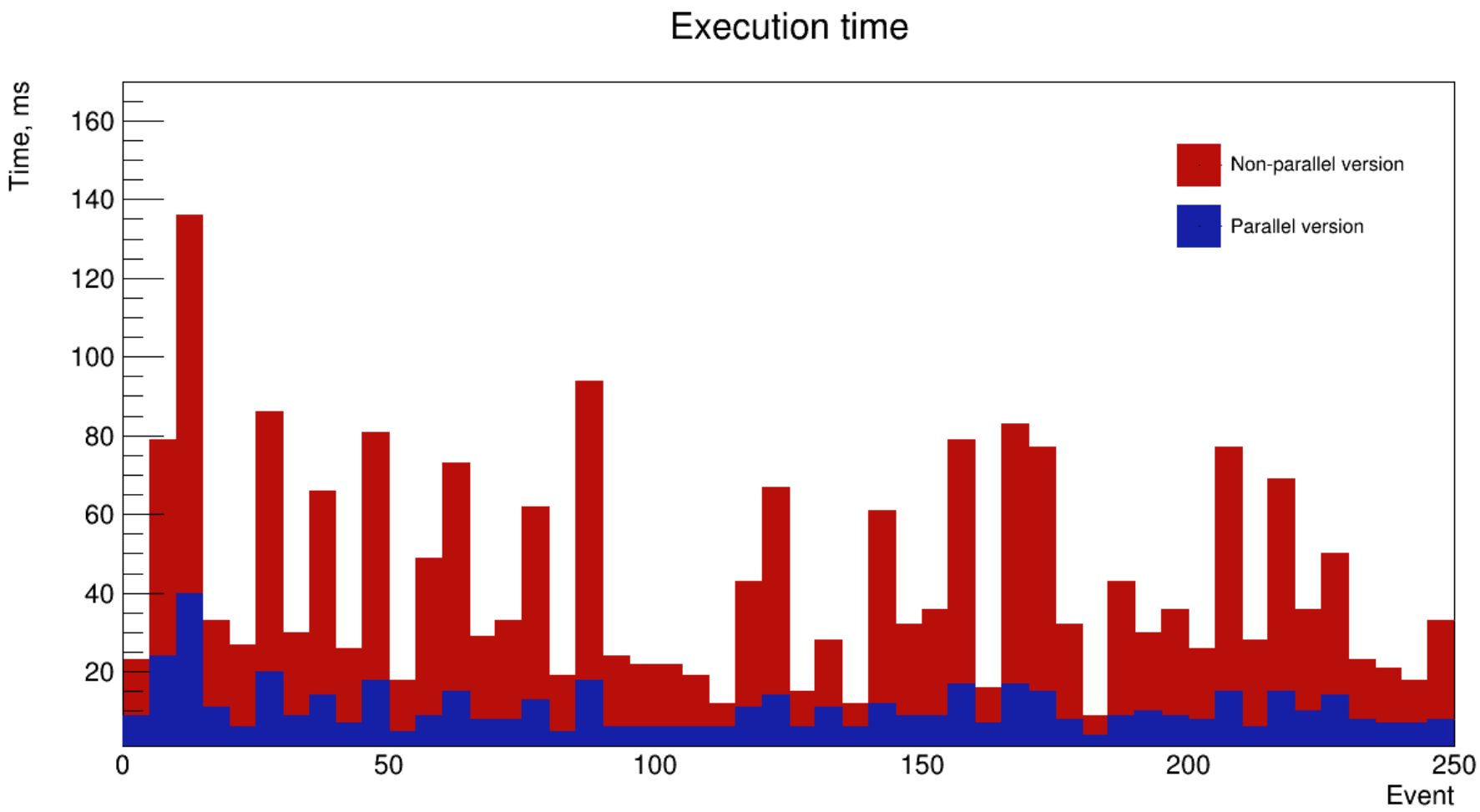
Parallel processing is performed for each of the 24 TPC sectors independently (one thread for each sector).

For that task POSIX Threads library was used.



Parallel processing results

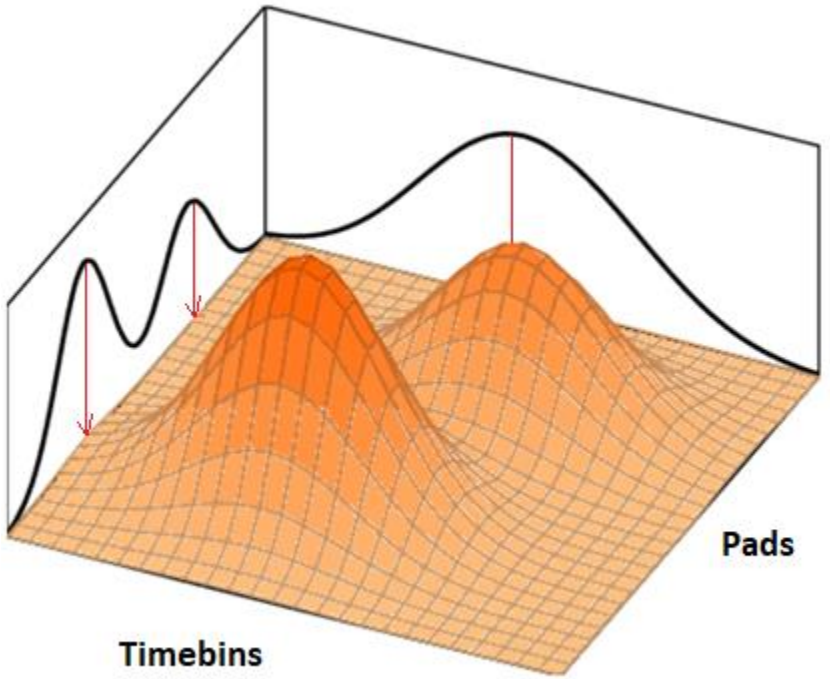
URQMD generator, 7.0GeV, mb, 250 events



Complex cluster disentanglement

A discrete **wavelet transform** (DWT) is a transform that decomposes a given signal into a number of sets, where each set is a series of coefficients describing the signal on a different scale.

This allows to "filter" the signal, cutting out the smal space structures. [*G.A.Ososkov "The growing neural gas and clustering of large amounts of data"*]

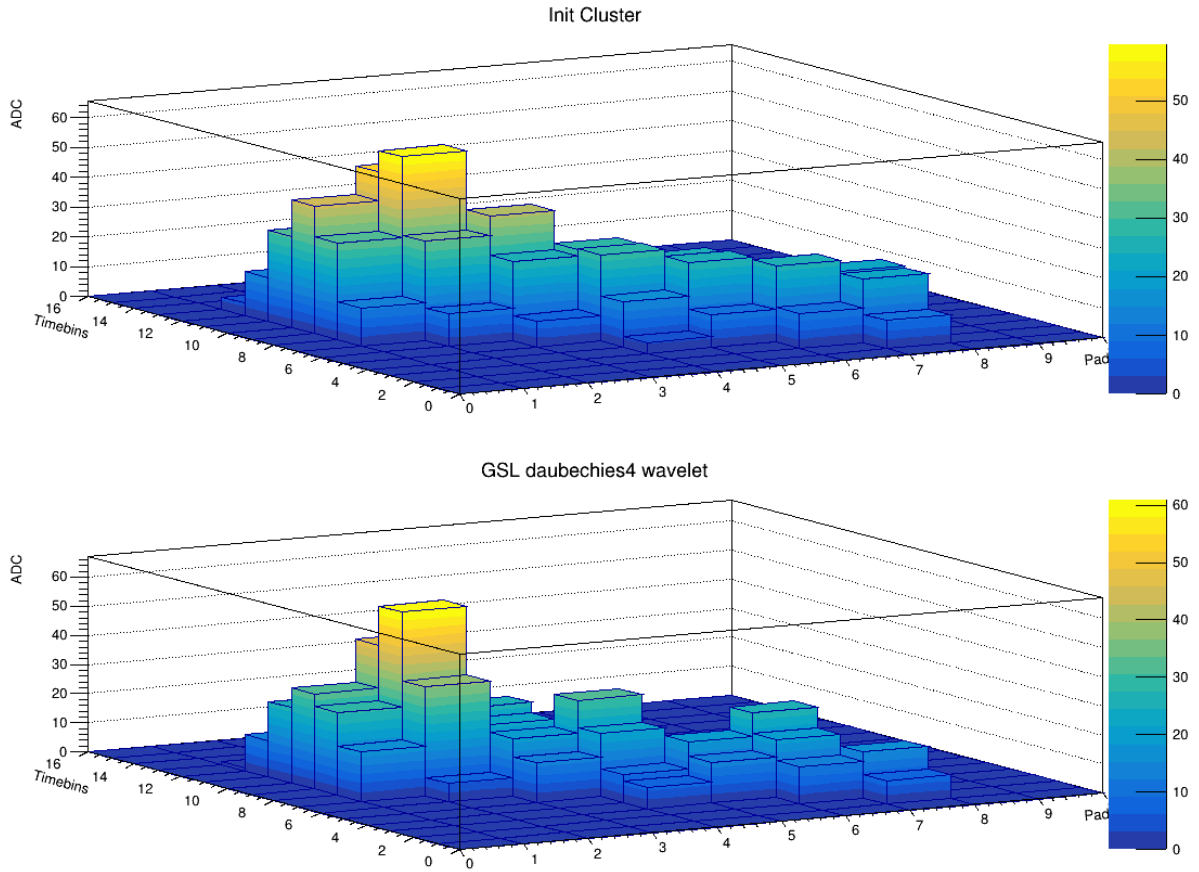


Signal with the translated wavelet

ClusterFinder with wavelet transform

The library provides functions to perform two-dimensional discrete wavelet transforms on square matrices.

Transform performs a complete discrete wavelet transform on the rows of the matrix, followed by a separate complete discrete wavelet transform on the columns.



Daubechie(4) wavelet transform

Wavelet transform on GPU

PDWT library is a parallel implementation of the Discrete Wavelet Transform (DWT). This implementation in CUDA targets Nvidia GPUs.

PDWT primarily aims at being fast, simple and versatile for an easy integration in a bigger project.

```
// Compute the wavelets coefficients with the "Daubechies 7" wavelet
Wavelets W(img, Nr, Nc, "db7", nlevels);
W.forward();

// Do some thresholding on the wavelets coefficients
float norm1 = W1.norm1();
printf("Before threshold : L1 = %e\n", norm1);
W.soft_threshold(10.0);
norm1 = W1.norm1();
printf("After threshold : L1 = %e\n", norm1);

// Inverse the DWT and retrieve the image from the GPU
W.inverse();
```

Git repository: https://github.com/pierrepaleo/PDWT

TODO

- Apply different wavelet transform algorithms on GPU
- Use GPU for other processes
- Apply different types of parallelization
- Using of FPGA cards for clustering (Alice)?

Conclusions

- Improvements:
- Increased task execution speed
- Added hits clusters proportional split
- Added parallelization by 24 sector
- Fixed some bugs

Thank you for your attention!

Introduction

- One of the tasks during the reconstruction of an event in the TPC d etector is the task of combining digits into clusters, followed by fin ding their most probable center. This task is called ClusterFinder a nd it is already present in the reconstruction chain.
- To work with real data from MPD detector the task of finding cluste rs must be completed in a minimum amount of time, but at the sa me time have satisfactory accuracy. This is why a library for quickly searching for clusters was written.

  For this task, methods to improve the speed and accuracy of cluste r determination, such as the use of GPU computing and wavelet tra nsform will be proposed

CUDA

CUDA is an architecture that allows the use of a graphics processing unit (GPU).

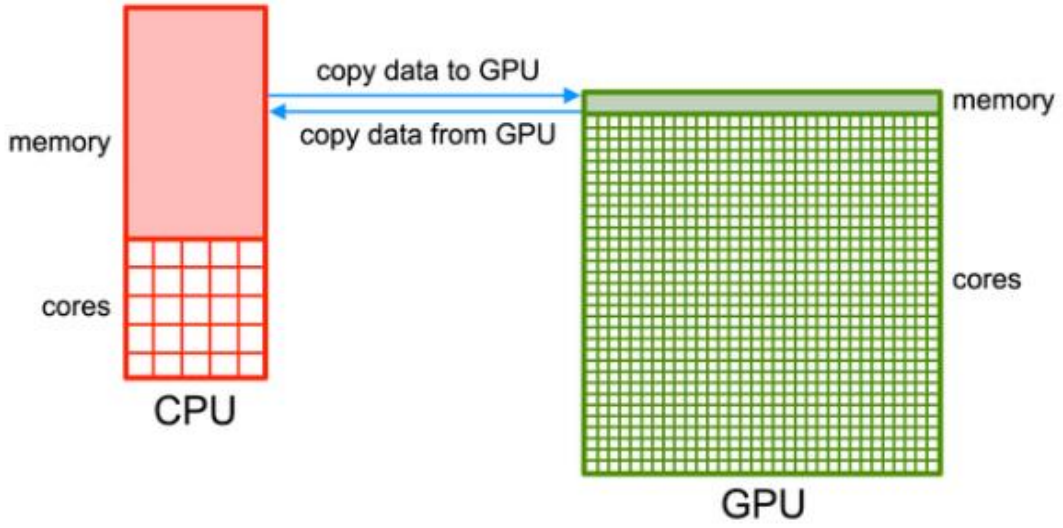
It is a set of tools and libraries for working with the GPU, based on GPGPU technology.

CUDA cores are better suited for tasks that require massive parallelism, such as matrix operations, machine learning and scientific simulations.

GPU computing

GPU computing enables applications to run with extreme efficiency by offloading series of computational scientific and technical tasks from the CPU.

GPUs process thousands of tasks in seconds through their hundreds of cores via parallel processing.



Operations for GPU

There are a number of processes for which processing using graphics cores is suitable.

For example, matrix calculations are used to search for the most probable cluster center.

Also GPU can be used to calculate wavelet transform for complex clusters.