ALICE HLT TPC Tracking on GPUs

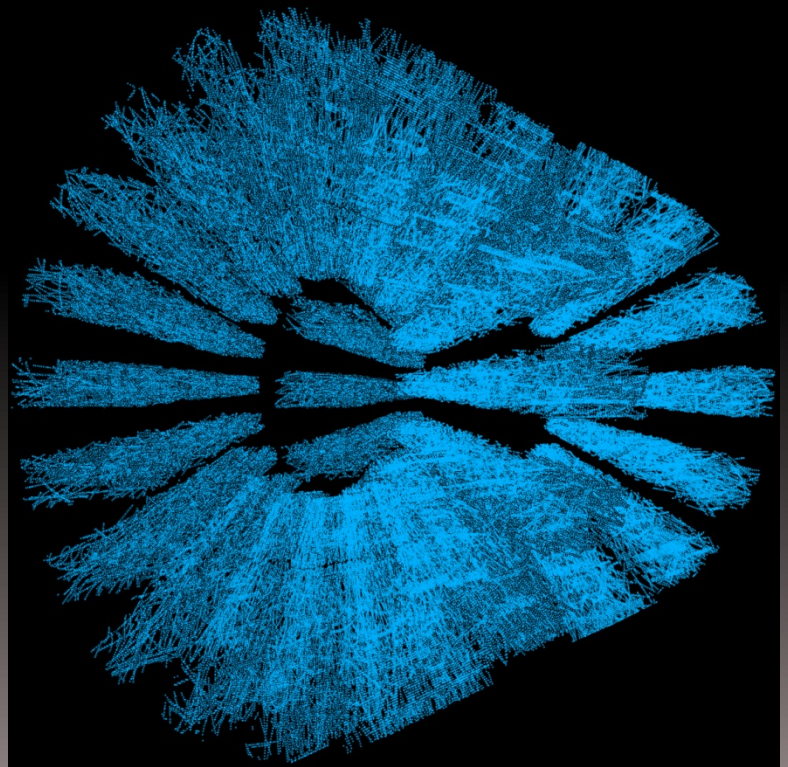
I: Introduction
II: Integration
III: GPU Tracker Performance
IV: CPU / GPU Tracker Comparison

David Rohr for the ALICE Collaboration
GSI - 29.1.2013

INTRODUCTION

Introduction

- ALICE HLT tracker divides the TPC in slices and processes the slices individually.
- Track segments from all slices are merged later.



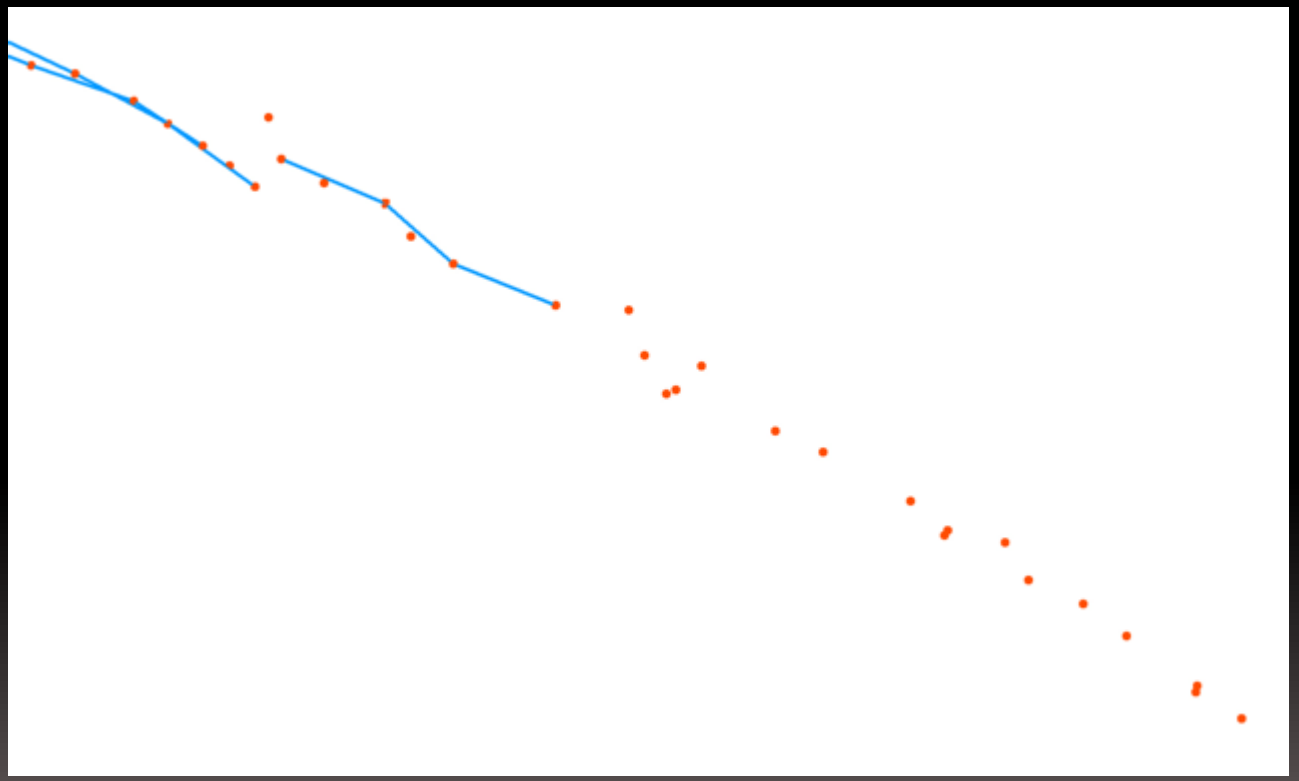
Introduction

Tracking algorithm

Category of task	Name of task	Description on task
	(Initialization)	
Combinatorial part (Cellular automation)	I: Neighbors finding	Construct seeds (Track candidates)
	II: Evolution	
Kalman filter part	III: Tracklet construction	Fit seed, extrapolate tracklet, find new clusters
	IV: Tracklet selection	Select good tracklets, sssign clusters to tracks
	(Tracklet output)	

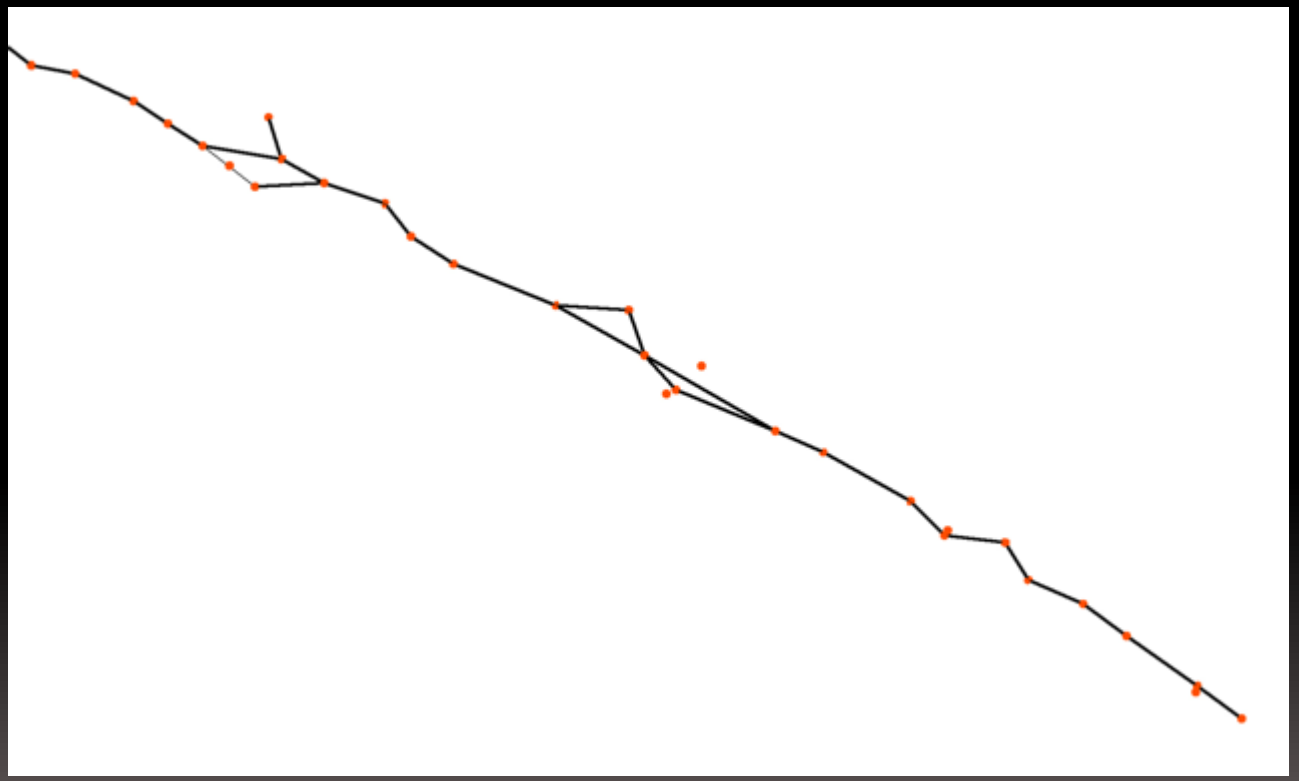
Introduction

Illustration of evolution step



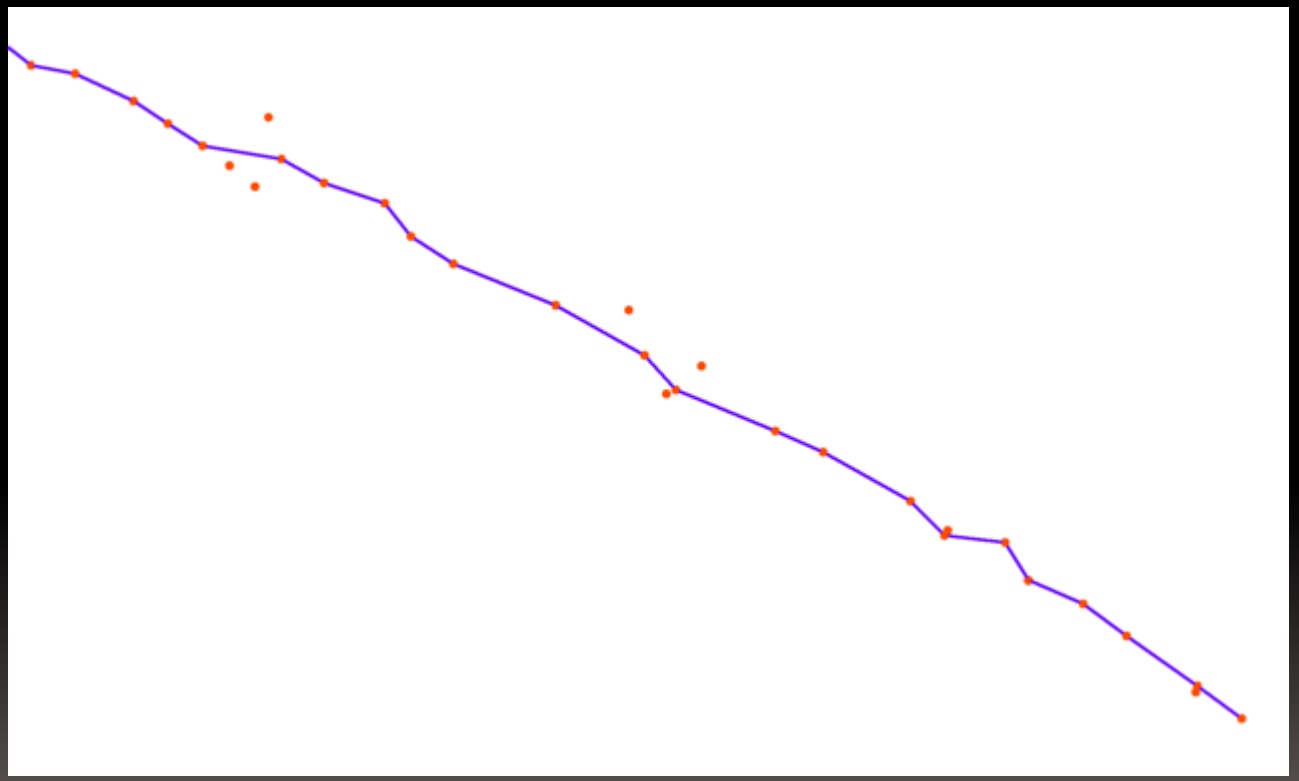
Introduction

Illustration of tracklet construction



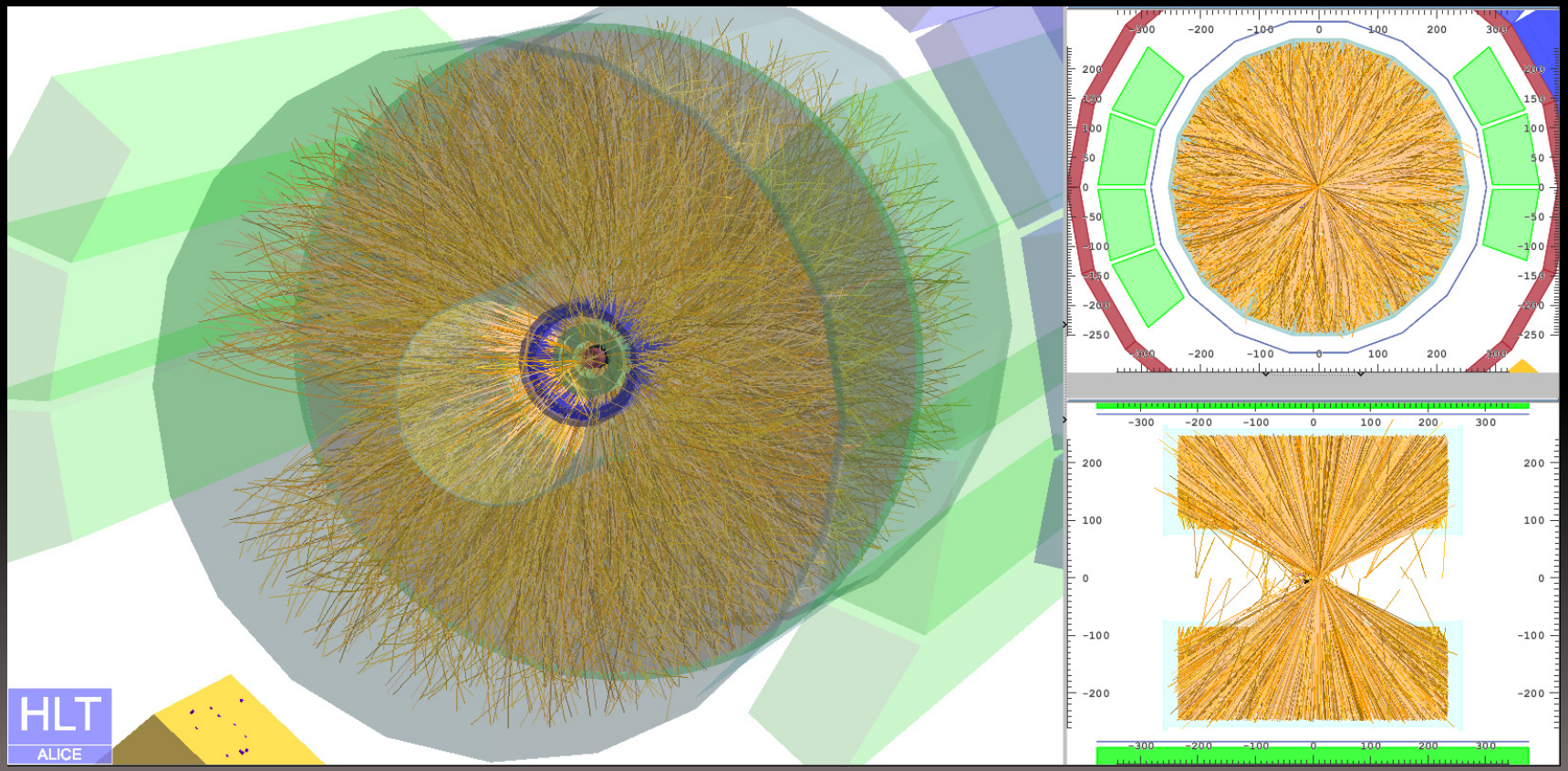
Introduction

Illustration of tracklet selection



Introduction

Screenshot of ALICE Online-Event-Display
during first physics-fill with active GPU Tracker



INTEGRATION

Integration

- GPU and CPU tracker share a common source files.
- Specialist wrappers for CPU and GPU exist, that include these common files.

common.cpp:

```
__DECL FitTrack(int n) {
….
}
```

cpu_wrapper.cpp:

```
#define __DECL void
#include ``common.cpp``

void FitTracks() {
  for (int i = 0;i < nTr;i++) {
    FitTrack(n);
  }
}
```

gpu_wrapper.cpp:

```
#define __DECL __device void
#include ``common.cpp``

__kernel void FitTracksGPU() {
  FitTrack(threadIdx.x);
}

void FitTracks() {
  FitTracksGPU<<<nTr>>>();
}
```

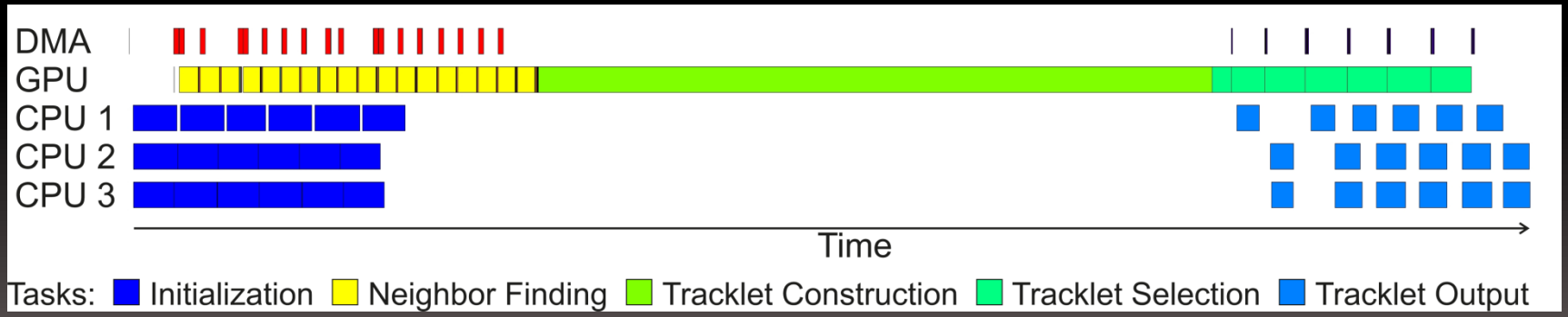
Integration

- The GPU Tracker is accessed via a virtual interface. The actual implementation is contained in a dedicated library (cagpu), which links against the CUDA runtime.
- AliRoot opens cagpu with dlopen, this creates a clear separation between AliRoot and CUDA.
- The same AliRoot binaries can be used on compute nodes with GPU and without GPU.
- This scheme is easily adoptable to other programming APIs, such as OpenCL.

CPU/GPU PERFORMANCE

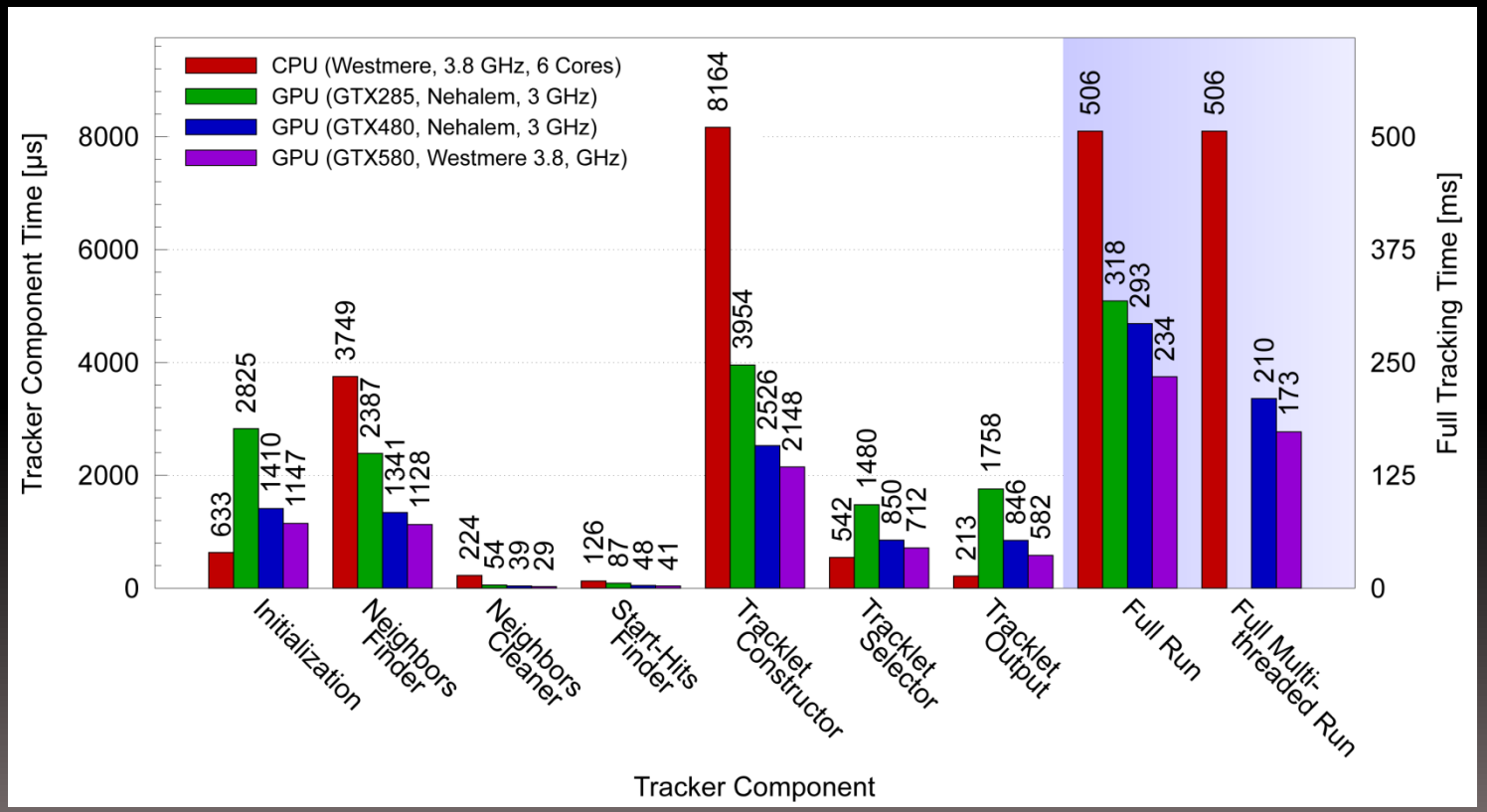
GPU Tracker Performance

- For good performance the GPU tracker pipelines the slices such that initialization on CPU, GPU tracking, and DMA transfer can overlap.
- A multithreaded pipeline ensures the CPU can keep step.



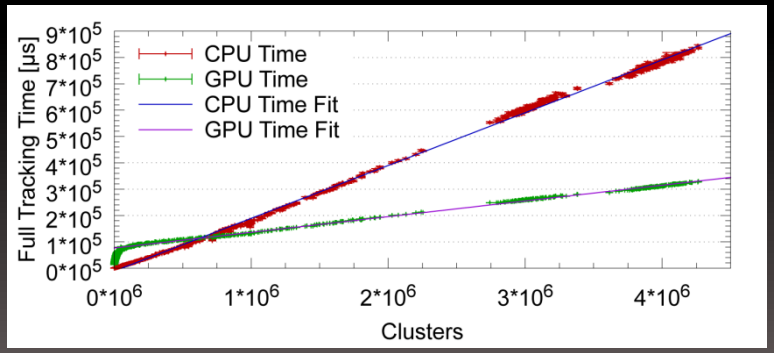
CPU / GPU Tracker Comparison

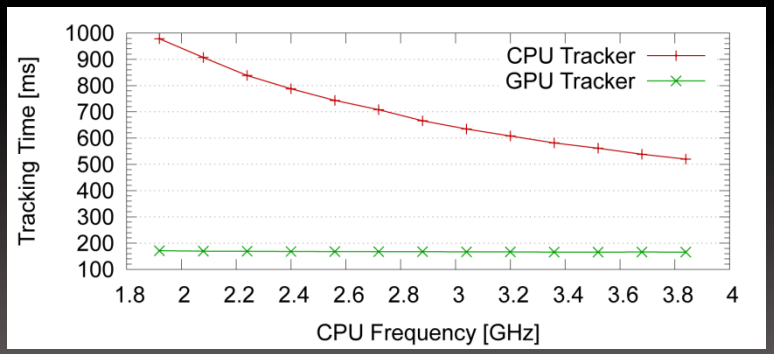
- Performance: GTX580 GPU almost three times as fast as 6-core processor.



GPU Tracker Performance

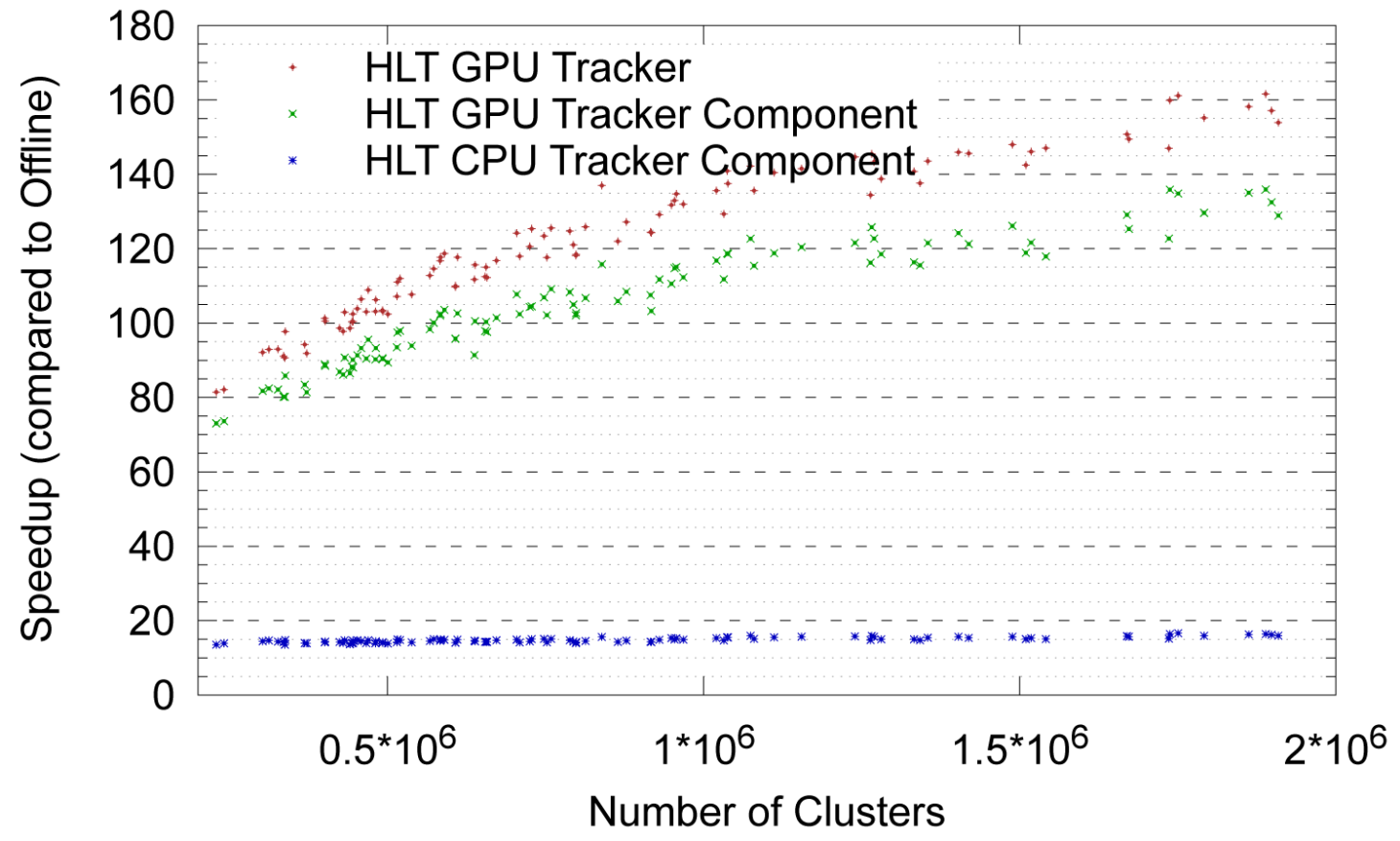
- Tracking time depends linearly on input data size.
- GPU tracking time independent from CPU performance (if initialization is fast enough).





GPU Tracker Performance

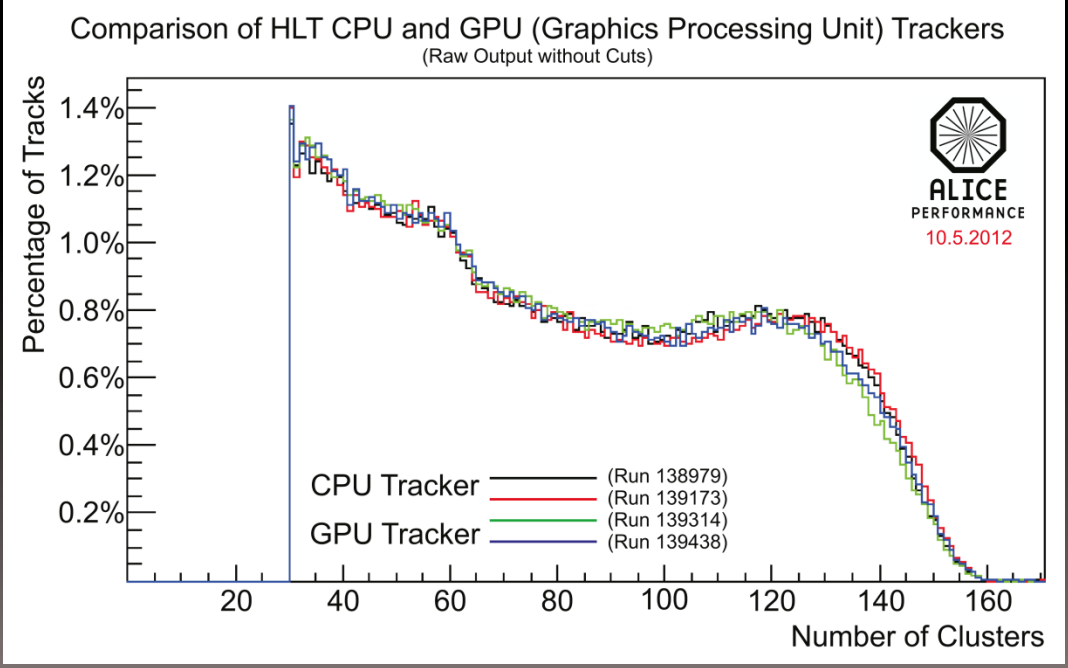
- Speedup of HLT GPU tracker v.s.offline and CPU Tracker (four CPU cores used each)



CPU / GPU TRACKER COMPARISON

CPU / GPU Tracker Consistency

- Inconsistencies during November 2010 run
  - Cluster to track assignment
  - Track Merger
  - Non-associative floating point arithmetics



CPU / GPU Tracker Consistency

- Cluster to track assignment
  - Problem: Cluster to track assignment was depending on the order of the tracks.
    - Each cluster was assigned to the longest possible track. Out of two tracks of the same length, the first one was chosen.
    - Concurrent GPU tracking processes the tracks in an undefined order.
  - Solution: Both the chi² and the track lenth are used as criteria. It is extremely unlikely that two tracks coincide in both values.

CPU / GPU Tracker Consistency

- Track merger
  - Problem: Result of the track merger depended on the order of input tracks.
  - Solution: Merger input is sorted.
    - Sorting is performed during a reformatting step.

    → No additional data copy.

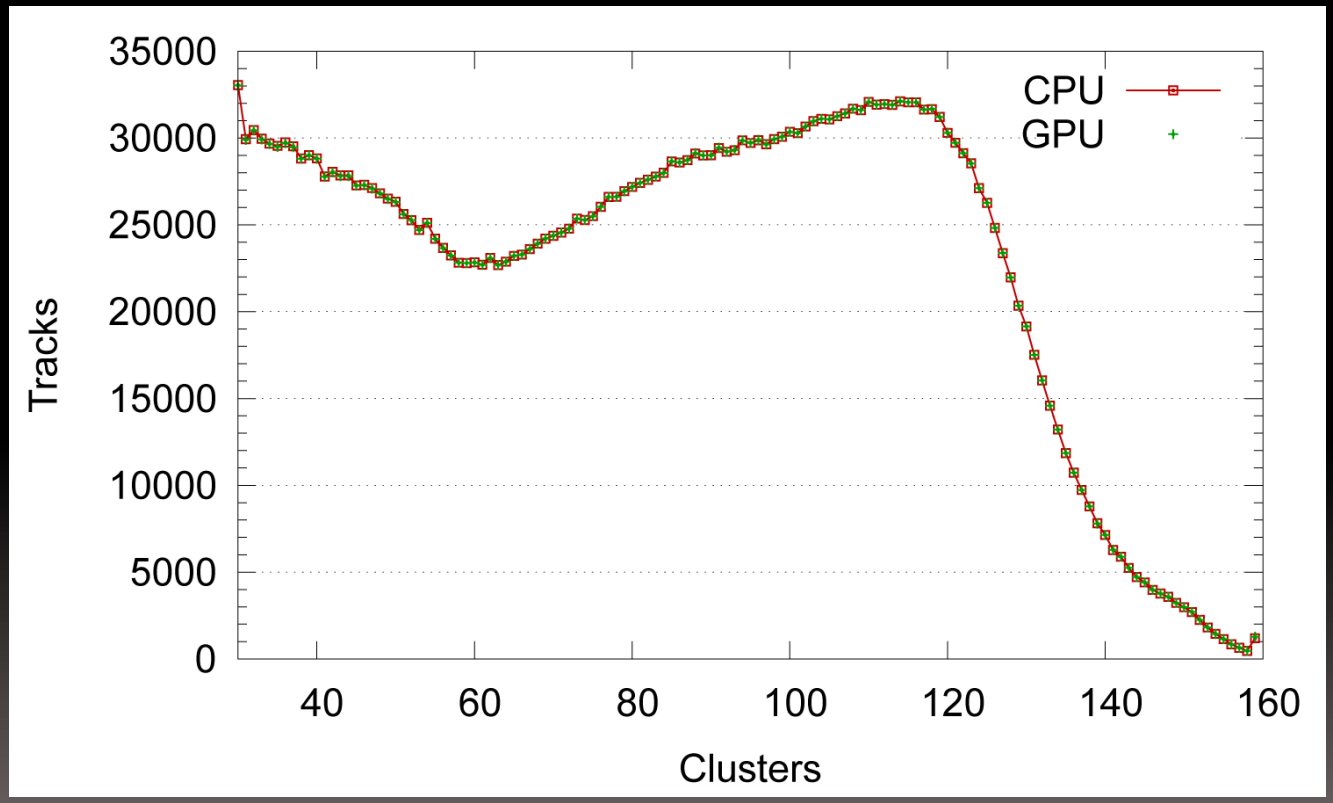
    → No performance penalty.

CPU / GPU Tracker Consistency

- Non associative floating point arithmetics
  - Problem: Different compilers perform the arithmetics in different order (also on the CPU).
  - Solution: Cannot be fixed, but…
    - Slight variations during the extrapolations do not matter as long as the clusters stay the same.
    - Inconsistent clusters: 0,00024%

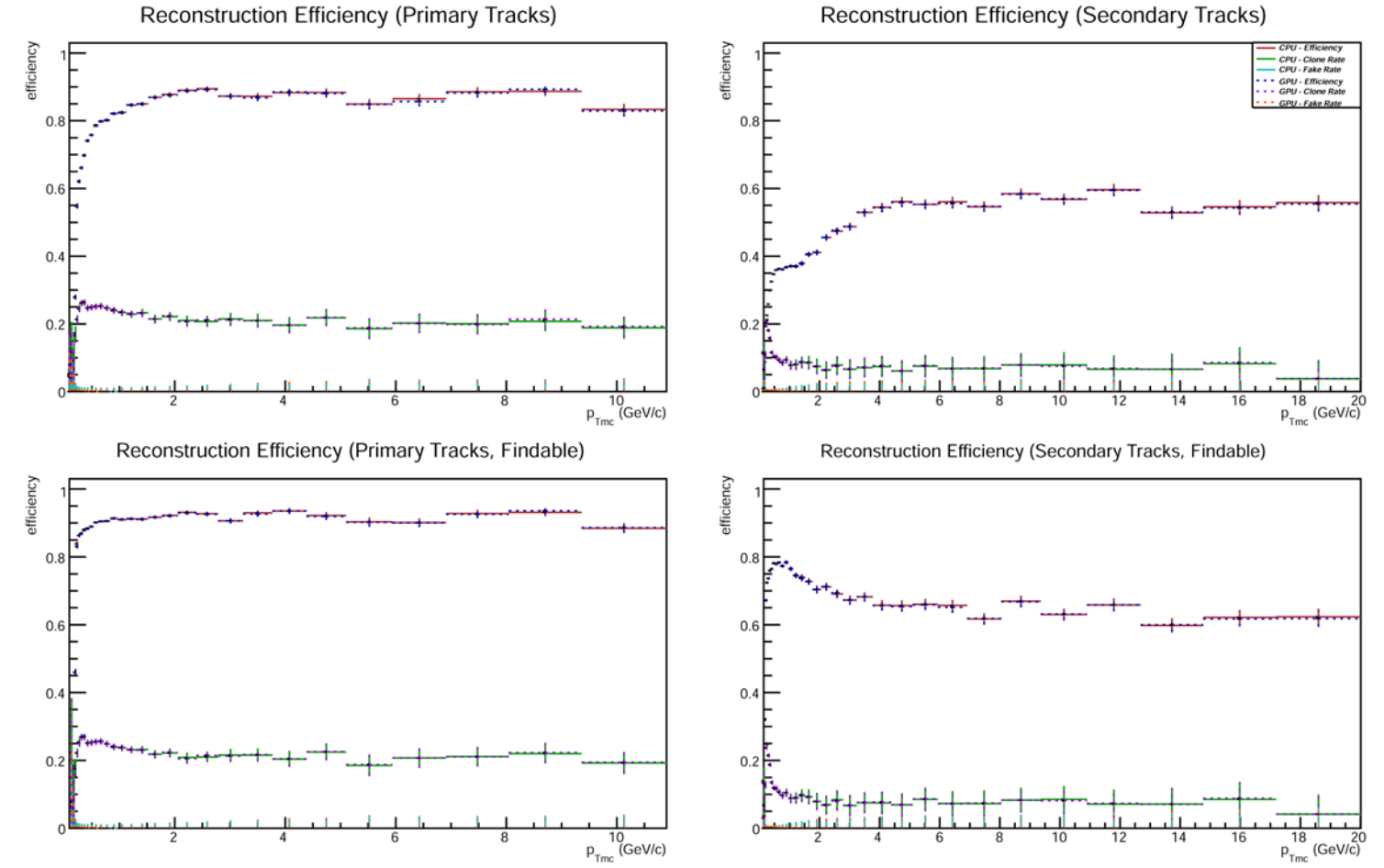
CPU / GPU Tracker Consistency

- Cluster per track statistic with improvements



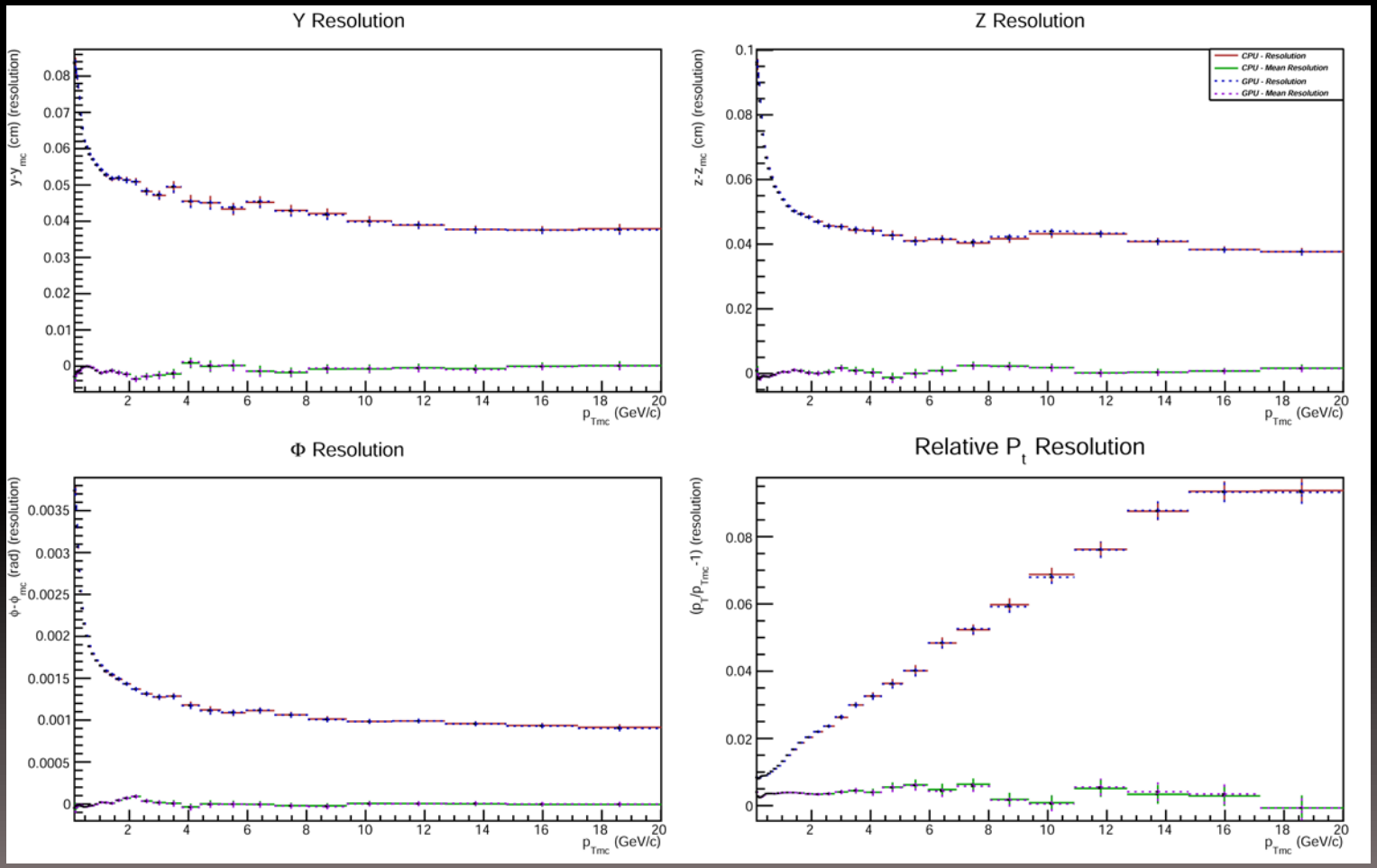
CPU / GPU Tracker Consistency

- Efficiency Comparison



CPU / GPU Tracker Consistency

- Resolution Comparison



Summary

- Threefold performance increase of GPU tracker compared to six-core CPU.
- GPU tracker performance is independent from CPU and depends linearly on data size.
- Results of GPU and CPU tracker match almost completely. Only 0.00024% of the clusters differ due to non-associative floating-point arithmetic